



***Topic Modelling pada Sentimen terhadap *Headline*
Berita *Online* Berbahasa Indonesia***

Chairullah Naury

16917204

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Sistem Informasi Enterprise

Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

2020

Lembar Pengesahan Pembimbing

***Topic Modelling* pada Sentimen terhadap *Headline* Berita Online
Berbahasa Indonesia**

Chairullah Naury

16917204



Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

Lembar Pengesahan Penguji

**Topic Modelling pada Sentimen terhadap *Headline* Berita Online
Berbahasa Indonesia**

Chairullah Naury

16917204

Yogyakarta, 21 Desember 2020

Tim Penguji,

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

Ketua

Dr. Ing. Ridho Rahmadi

Anggota I

Ahmad Munasir Rafie Pratama, S.T., M.I.T, Ph.D.

Anggota II

Mengetahui,

Ketua Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia



Izzati Muhiimah, S.T., M.Sc., Ph.D.

Abstrak

***Topic Modelling* pada Sentimen terhadap *Headline* Berita *Online* Berbahasa Indonesia**

Media massa *online* adalah sumber informasi tercepat dan selalu menyajikan berita yang bersifat *up-to-date*. Sebuah model dapat memberikan pemetaan yang membantu dalam memilah informasi dengan lebih tepat. Pada penelitian ini, penulis menerapkan *topic modelling* terhadap hasil *sentiment analysis* pada *headline* berita *online* berbahasa Indonesia. Sumber data pada penelitian ini diperoleh dari media massa *online* berbahasa Indonesia. Data yang terkumpul dianalisis sentimennya dengan menggunakan metode *Long Short-term Memory* (LSTM), sehingga diperoleh tajuk-tajuk berita dengan sentimen positif, negatif, dan netral. Klasifikasi yang diperoleh dari hasil proses sentimen analisis tersebut dilanjutkan dengan proses pemodelan topik menggunakan metode *Latent Dirichlet Allocation* (LDA) dan divisualisasikan dalam bentuk *wordcloud* dan *intertopic distance map* (pyLDAVis) untuk mengetahui jarak antar topik dan keterkaitan satu topik dengan topik lainnya. Proses analisis sentimen menghasilkan model dengan berbagai tingkat akurasi dan semua model mengalami *overfitted* karena akurasi validasi lebih kecil dari akurasi pelatihan. Dengan mempertimbangkan nilai *overfitting* yang kecil dan prosentase akurasi diatas 70% diketahui model dengan nilai akurasi yang dipilih adalah model dengan nilai *training accuracy* 76.64%, *validation accuracy* 70.36%, dan *overfitted* 6.28%. Kecilnya akurasi yang diperoleh pada model LSTM menyebabkan terjadinya bias pada hasil *Topic Modelling*. Dari sisi *prototyping*, model yang dihasilkan dalam penelitian ini berhasil diimplementasikan dalam sebuah *prototype* aplikasi berbasis web yang mampu melakukan proses pemodelan topik pada hasil analisis sentimen terhadap *headline* berita *Online* berbahasa Indonesia.

Kata kunci

analisis sentimen; long short-term memory; headline berita online; pemodelan topik, latent dirichlet allocation, prototype.

Abstract

Topic modeling to the results of sentiment analysis on online news headlines in Indonesian

The online mass media is now become source of the fastest and up-to-date information. A model that can provide mapping will help in sorting out information more precisely. In this study, the authors applied topic modeling to the results of sentiment analysis on online news headlines in Indonesian. Sources of data in this study were obtained from online mass media in Indonesian. The data collected were analyzed for sentiment using the Long Short-term Memory (LSTM) method, in order to obtain news headlines with positive, negative, and neutral sentiments. The classification obtained from the results of the sentiment analysis process is continued with the topic modeling process using the Latent Dirichlet Allocation (LDA) method and visualized in the form of wordcloud and intertopic distance map (pyLDAVis) to determine the relationship between one topic and another. The sentiment analysis process produces models with various levels of accuracy and all models are overfitted. By considering the small overfitting value and the accuracy percentage above 70%, this research choose the model with a training accuracy value of 76.64%, validation accuracy 70.36%, and overfitted 6.28%. This small accuracy obtained in the LSTM model causes bias in the Topic Modeling results. The model which is obtained in this research successfully to be implemented into a prototype in form of web based application that able to handle the process of topic modelling to the results of sentiment analysis on online news headlines in Indonesian.

Keywords

sentiment analysis, long short-term memory, online news headline, topic modelling, latent dirichlet allocation, prototype.

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, November, 2020



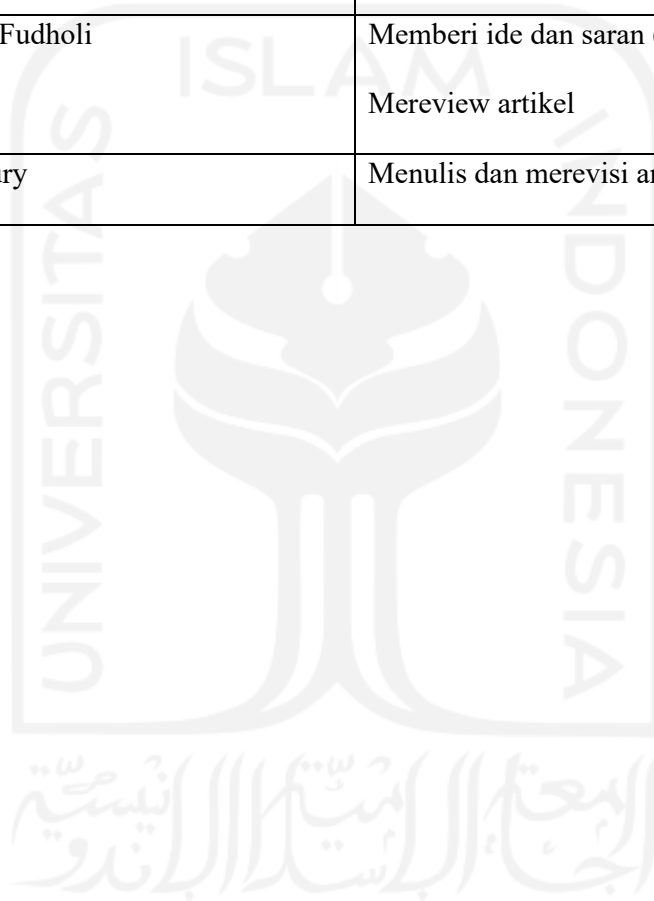
Chairullah Naury, S.Kom.

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Naury, C., Fudholi, D. H., Hidayatullah, A. F. (2021). *Topic Modelling* pada Sentimen terhadap *Headline Berita Online* Berbahasa Indonesia Menggunakan LDA dan LSTM. *Jurnal Media Informatika Budidarma* (Vol. 5, No. 1).

| Kontributor | Jenis Kontribusi |
|---------------------------|---|
| Ahmad Fathan Hidayatullah | Memberi ide dan saran (30%) |
| Dhomas Hatta Fudholi | Memberi ide dan saran (70%) Mereview artikel |
| Chairullah Naury | Menulis dan merevisi artikel (100%) |



Halaman Kontribusi

Terima kasih penulis sampaikan kepada Bapak Ahmad Fathan Hidayatullah atas ide awal penelitian tentang pemodelan topik dan analisis sentimen. Selanjutnya terima kasih untuk Bapak Dhomas Hatta Fudholi yang telah membimbing penulis di dalam melaksanakan penelitian ini.



Halaman Persembahan

Dengan mengucap rasa syukur Alhamdulillah, Saya persembahkan hasil penelitian ini kepada orang-orang yang selama ini selalu mendukung dan memotivasi di dalam menyelesaikan pendidikan magister di Universitas Islam Indonesia, khususnya kepada:

1. Ayah dan Ibu, terima kasih atas segala doa dan dukungannya.
2. Istri saya Rusmini, dan anak-anakku Muhammad Nur Adnan Na'im, Zaini Azka Dzulhaydar, Anis Adzka Dzulhaydar, Muhammad Ashfa Ar-Rayhan, dan Kamila Dian Hayuningtyas, kalian semua adalah penyemangat saya dalam hidup ini.
3. Instansi saya AMIK Harapan Bangsa Surakarta yang telah memberikan support positif kepada saya.



Kata Pengantar

Assalamu ‘alaikum Wr. Wb.

Bismillaahirrohmaanirrohiim, alhamdulillahirobbil ‘alamiin, assholaatu wassalaamu ‘alaa Nabiyyinaa Muhammadin wa ‘alaa aalihi wa shohbih. Penulis bersyukur kepada Allah Ta’aala yang telah memberikan segala nikmat sehingga penulis dapat menyelesaikan tesis yang berjudul “*Topic Modelling* pada Sentimen terhadap *Headline* Berita *Online* Berbahasa Indonesia”.

Penyusunan tesis ini dapat selesai berkat dukungan dari berbagai pihak, pada kesempatan ini dengan segala hormat penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Fathul Wahid, ST, M,Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
2. Ibu Izzati Muhimmah, ST., M.Sc., Ph.D., selaku Ketua Program Studi Informatika Program Magister Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Dhomas Hatta Fudholi, ST., M. Eng., Ph.D., selaku pembimbing penulisan tesis yang banyak memberikan saran dan masukan selama proses bimbingan.
4. Bapak dan Ibu Dosen Program Studi Informatika Program Magister Fakultas Teknologi Industri Universitas Islam Indonesia yang telah banyak berbagi ilmu pengetahuan dan pengalaman.
5. Teman-teman Magister Informatika Angkatan XV, terima kasih telah saling membantu dan memberikan motivasi.

Penulis menyadari bahwa laporan penelitian ini masih jauh dari sempurna, oleh karena itu penulis dengan senang hati menerima masukan dan saran yang membangun dari para pembaca semua agar pada penelitian selanjutnya akan menjadi lebih baik lagi.

Wassalamu ‘alaikum Wr. Wb.

Yogyakarta, November 2020

Penulis

Daftar Isi

| | |
|------------------------------------|------|
| Lembar Pengesahan Pembimbing | i |
| Lembar Pengesahan Penguji..... | ii |
| Abstrak | iii |
| Abstract..... | iv |
| Pernyataan Keaslian Tulisan | v |
| Daftar Publikasi..... | vi |
| Halaman Kontribusi..... | vii |
| Halaman Persembahan | viii |
| Kata Pengantar..... | ix |
| Daftar Isi..... | x |
| Daftar Tabel..... | xii |
| Daftar Gambar | xiii |
| Daftar <i>Code Snippet</i> | xvii |
| BAB 1 Pendahuluan | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Manfaat Penelitian | 3 |
| 1.6 Sistematika Penulisan | 4 |
| BAB 2 Tinjauan Pustaka | 5 |
| 2.1 Landasan Teori..... | 5 |
| 2.1.1 Topic Modelling | 5 |
| 2.1.2 Sentiment Analysis | 5 |
| 2.1.3 Hyperparameters Tuning | 7 |

| | | |
|----------------|---|-----|
| 2.1.4 | Prototyping | 7 |
| 2.2 | Kajian Pustaka | 8 |
| BAB 3 | Metodologi | 17 |
| 3.1 | Data Penelitian | 17 |
| 3.2 | Langkah-langkah Penelitian..... | 18 |
| 3.2.1 | Literature Review | 19 |
| 3.2.2 | Pembuatan Model dan Pengujian Model..... | 19 |
| 3.2.3 | Pengembangan <i>Prototype</i> Aplikasi Berbasis Web | 22 |
| BAB 4 | Hasil dan Pembahasan..... | 23 |
| 4.1 | Hasil dari Tahapan Pembuatan dan Pengujian Model | 23 |
| 4.1.1 | Pengumpulan Data..... | 23 |
| 4.1.2 | Pembuatan Model LSTM untuk Analisis Sentimen..... | 54 |
| 4.1.3 | Pelabelan <i>Unlabeled Dataset</i> menggunakan model LSTM terpilih..... | 83 |
| 4.1.4 | Pemodelan Topik..... | 89 |
| 4.1.5 | Visualisasi..... | 100 |
| 4.2 | <i>Prototype</i> Aplikasi Berbasis Web | 106 |
| 4.2.1 | Membuat Akun Pengguna | 107 |
| 4.2.2 | Masuk Sistem | 108 |
| 4.2.3 | Melihat Semua <i>Project</i> | 109 |
| 4.2.4 | Membuat <i>Project</i> Pemodelan Topik..... | 109 |
| 4.2.5 | Melihat Hasil Project Pemodelan Topik..... | 111 |
| BAB 5 | Kesimpulan dan Saran..... | 112 |
| 5.1 | Kesimpulan | 112 |
| 5.2 | Saran | 112 |
| Daftar Pustaka | | 114 |

Daftar Tabel

| | |
|---|-----|
| Tabel 2.1 Daftar <i>Literature Review</i> | 9 |
| Tabel 3.1 Daftar Situs Berita <i>Online</i> Berbahasa Indonesia..... | 17 |
| Tabel 3.2 Skenario <i>Hyperparameters Tuning</i> | 21 |
| Tabel 4.1 Nilai Akurasi Hasil Pelatihan Model LSTM..... | 82 |
| Tabel 4.2. Hasil Ekstraksi Topik Dataset <i>Headline</i> Bersentimen Positif..... | 99 |
| Tabel 4.3. Hasil Ekstraksi Topik Dataset <i>Headline</i> Bersentimen Netral | 100 |
| Tabel 4.4. Hasil Ekstraksi Topik Dataset <i>Headline</i> Bersentimen Negatif | 100 |
| Tabel 4.5. <i>Insight</i> Hasil Ekstraksi Topik pada Dataset <i>Headline</i> Bersentimen Positif | 102 |
| Tabel 4.6. <i>Insight</i> Hasil Ekstraksi Topik pada Dataset <i>Headline</i> Bersentimen Netral..... | 104 |
| Tabel 4.7. <i>Insight</i> Hasil Ekstraksi Topik pada Dataset <i>Headline</i> Bersentimen Negatif... | 106 |



Daftar Gambar

| | |
|--|----|
| Gambar 3.1 Langkah-langkah Penelitian | 18 |
| Gambar 4.1 Struktur Tabel `headlines` | 23 |
| Gambar 4.2 Struktur Tabel `master_account` | 24 |
| Gambar 4.3 Struktur Tabel `master_project` | 24 |
| Gambar 4.4 Struktur Tabel `cron` | 24 |
| Gambar 4.5 Situs Berita Online antaranews.com..... | 25 |
| Gambar 4.6 <i>Source</i> HTML Situs Berita antaranews.com | 25 |
| Gambar 4.7 Situs Berita Online cnnindonesia.com..... | 27 |
| Gambar 4.8. <i>Source</i> HTML Situs Berita cnnindonesia.com | 27 |
| Gambar 4.9 Situs Berita Online jpnn.com..... | 29 |
| Gambar 4.10 <i>Source</i> HTML Situs Berita jpnn.com | 30 |
| Gambar 4.11 Situs Berita <i>Online</i> kompas.com | 31 |
| Gambar 4.12 <i>Source</i> HTML Situs Berita kompas.com..... | 32 |
| Gambar 4.13 Situs Berita <i>Online</i> liputan6.com..... | 33 |
| Gambar 4.14 <i>Source</i> HTML Situs Berita liputan6.com | 34 |
| Gambar 4.15 Situs Berita <i>Online</i> merdeka.com | 35 |
| Gambar 4.16 <i>Source</i> HTML Situs Berita merdeka.com | 36 |
| Gambar 4.17 Situs Berita <i>Online</i> detik.com..... | 38 |
| Gambar 4.18 <i>Source</i> HTML Situs Berita detik.com | 38 |
| Gambar 4.19 Situs Berita <i>Online</i> okezone.com | 40 |
| Gambar 4.20 <i>Source</i> HTML Situs Berita okezone.com | 40 |
| Gambar 4.21 Situs Berita <i>Online</i> rmol.id..... | 42 |
| Gambar 4.22 <i>Source</i> HTML Situs Berita rmol.id | 43 |
| Gambar 4.23 Situs Berita <i>Online</i> sindonews.com..... | 44 |
| Gambar 4.24 <i>Source</i> HTML Situs Berita sindonews.com | 45 |
| Gambar 4.25 Situs Berita <i>Online</i> suara.com | 47 |
| Gambar 4.26 <i>Source</i> HTML Situs Berita suara.com..... | 47 |
| Gambar 4.27 Situs Berita <i>Online</i> tempo.co..... | 49 |
| Gambar 4.28 <i>Source</i> HTML Situs Berita tempo.co | 49 |
| Gambar 4.29 Menambahkan <i>Scheduled Cron Jobs</i> | 52 |

| | |
|---|----|
| Gambar 4.30 Data Headline Berita Hasil Scrapping Situs Berita Online | 52 |
| Gambar 4.31 Pelabelan Dataset Latih | 53 |
| Gambar 4.32 Arsitektur Model LSTM Eksperimen #1 | 54 |
| Gambar 4.33 <i>Training and Validation Accuracy of Model #1</i> | 55 |
| Gambar 4.34 Diagram Plot <i>Training and Validation Accuracy Model #1</i> | 55 |
| Gambar 4.35 Diagram Plot <i>Training and Validation Loss Model #1</i> | 55 |
| Gambar 4.36 Arsitektur Model LSTM Eksperimen #2 | 56 |
| Gambar 4.37 <i>Training and Validation Accuracy of Model #2</i> | 56 |
| Gambar 4.38 Diagram Plot <i>Training and Validation Accuracy Model #2</i> | 57 |
| Gambar 4.39 Diagram Plot <i>Training and Validation Loss Model #2</i> | 57 |
| Gambar 4.40 Arsitektur Model LSTM Eksperimen #3 | 58 |
| Gambar 4.41 <i>Training and Validation Accuracy of Model #3</i> | 58 |
| Gambar 4.42 Diagram Plot <i>Training and Validation Accuracy Model #3</i> | 59 |
| Gambar 4.43 Diagram Plot <i>Training and Validation Loss Model #3</i> | 59 |
| Gambar 4.44 Arsitektur Model LSTM Eksperimen #4 | 60 |
| Gambar 4.45 <i>Training and Validation Accuracy of Model #4</i> | 60 |
| Gambar 4.46 Diagram Plot <i>Training and Validation Accuracy Model #4</i> | 61 |
| Gambar 4.47 Diagram Plot <i>Training and Validation Loss Model #4</i> | 61 |
| Gambar 4.48 Arsitektur Model LSTM Eksperimen #5 | 62 |
| Gambar 4.49 <i>Training and Validation Accuracy of Model #5</i> | 62 |
| Gambar 4.50 Diagram Plot <i>Training and Validation Accuracy Model #5</i> | 63 |
| Gambar 4.51 Diagram Plot <i>Training and Validation Loss Model #5</i> | 63 |
| Gambar 4.52 Arsitektur Model LSTM Eksperimen #6 | 64 |
| Gambar 4.53 <i>Training and Validation Accuracy of Model #6</i> | 64 |
| Gambar 4.54 Diagram Plot <i>Training and Validation Accuracy Model #6</i> | 65 |
| Gambar 4.55 Diagram Plot <i>Training and Validation Loss Model #6</i> | 65 |
| Gambar 4.56 Arsitektur Model LSTM Eksperimen #7 | 66 |
| Gambar 4.57 <i>Training and Validation Accuracy of Model #7</i> | 66 |
| Gambar 4.58 Diagram Plot <i>Training and Validation Accuracy Model #7</i> | 67 |
| Gambar 4.59 Diagram Plot <i>Training and Validation Loss Model #7</i> | 67 |
| Gambar 4.60 Arsitektur Model LSTM Eksperimen #8 | 68 |
| Gambar 4.61 <i>Training and Validation Accuracy of Model #8</i> | 68 |
| Gambar 4.62 Diagram Plot <i>Training and Validation Accuracy Model #8</i> | 69 |

| | |
|---|-----|
| Gambar 4.63 Diagram Plot <i>Training and Validation Loss Model #8</i> | 69 |
| Gambar 4.64 Arsitektur Model LSTM Eksperimen #9 | 70 |
| Gambar 4.65 <i>Training and Validation Accuracy of Model #9</i> | 70 |
| Gambar 4.66 Diagram Plot <i>Training and Validation Accuracy Model #9</i> | 71 |
| Gambar 4.67 Diagram Plot <i>Training and Validation Loss Model #9</i> | 71 |
| Gambar 4.68 Arsitektur Model LSTM Eksperimen #10 | 72 |
| Gambar 4.69 <i>Training and Validation Accuracy of Model #10</i> | 72 |
| Gambar 4.70 Diagram Plot <i>Training and Validation Accuracy Model #10</i> | 73 |
| Gambar 4.71 Diagram Plot <i>Training and Validation Loss Model #10</i> | 73 |
| Gambar 4.72 Arsitektur Model LSTM Eksperimen #11 | 74 |
| Gambar 4.73 <i>Training and Validation Accuracy of Model #11</i> | 74 |
| Gambar 4.74 Diagram Plot <i>Training and Validation Accuracy Model #11</i> | 75 |
| Gambar 4.75 Diagram Plot <i>Training and Validation Loss Model #11</i> | 75 |
| Gambar 4.76 Arsitektur Model LSTM Eksperimen #12 | 76 |
| Gambar 4.77 <i>Training and Validation Accuracy of Model #12</i> | 76 |
| Gambar 4.78 Diagram Plot <i>Training and Validation Accuracy Model #12</i> | 77 |
| Gambar 4.79 Diagram Plot <i>Training and Validation Loss Model #12</i> | 77 |
| Gambar 4.80 Arsitektur Model LSTM Eksperimen #13 | 78 |
| Gambar 4.81 <i>Training and Validation Accuracy of Model #13</i> | 78 |
| Gambar 4.82 Diagram Plot <i>Training and Validation Accuracy Model #13</i> | 79 |
| Gambar 4.83 Diagram Plot <i>Training and Validation Loss Model #13</i> | 79 |
| Gambar 4.84 Arsitektur Model LSTM Eksperimen #14..... | 80 |
| Gambar 4.85 <i>Training and Validation Accuracy of Model #14</i> | 80 |
| Gambar 4.86 <i>Training and Validation Accuracy Model #14</i> | 81 |
| Gambar 4.87 <i>Training and Validation Loss Model #14</i> | 81 |
| Gambar 4.88 Contoh Sebagian <i>Dataset Headline</i> Berita Belum Memiliki Label | 88 |
| Gambar 4.89 Proses Jalannya Pelabelan Dataset `headlines` | 88 |
| Gambar 4.90 Hasil Pelabelan Dataset `headlines` | 89 |
| Gambar 4.91 Dataset <i>Headline</i> Berita Berlabel <i>Positive</i> | 90 |
| Gambar 4.92 Dataset <i>Headline</i> Berita Berlabel <i>Neutral</i> | 90 |
| Gambar 4.93 Dataset <i>Headline</i> Berita Berlabel <i>Negative</i> | 91 |
| Gambar 4.94 <i>Wordcloud</i> Topic #0 Sentimen Positif..... | 101 |
| Gambar 4.95 <i>Wordcloud</i> Topic #1 Sentimen Positif..... | 101 |

| | |
|--|-----|
| Gambar 4.96 <i>Wordcloud</i> Topic #2 Sentimen Positif..... | 101 |
| Gambar 4.97 <i>Wordcloud</i> Topic #3 Sentimen Positif..... | 101 |
| Gambar 4.98 <i>Wordcloud</i> Topic #4 Sentimen Positif..... | 101 |
| Gambar 4.99 <i>Wordcloud</i> Topic #5 Sentimen Positif..... | 101 |
| Gambar 4.100 <i>Intertopic Distance Map</i> Topik-topik <i>Headline</i> Bersentimen Positif..... | 102 |
| Gambar 4.101 <i>Wordcloud</i> Topic #0 Sentimen Netral..... | 103 |
| Gambar 4.102 <i>Wordcloud</i> Topic #1 Sentimen Netral..... | 103 |
| Gambar 4.103 <i>Wordcloud</i> Topic #2 Sentimen Netral..... | 103 |
| Gambar 4.104 <i>Wordcloud</i> Topic #3 Sentimen Netral..... | 103 |
| Gambar 4.105 <i>Wordcloud</i> Topic #4 Sentimen Netral..... | 103 |
| Gambar 4.106 <i>Wordcloud</i> Topic #5 Sentimen Netral..... | 103 |
| Gambar 4.107 <i>Intertopic Distance Map</i> Topik-topik <i>Headline</i> Bersentimen Netral..... | 104 |
| Gambar 4.108 <i>Wordcloud</i> Topic #0 Sentimen Negatif..... | 105 |
| Gambar 4.109 <i>Wordcloud</i> Topic #1 Sentimen Negatif..... | 105 |
| Gambar 4.110 <i>Wordcloud</i> Topic #2 Sentimen Negatif..... | 105 |
| Gambar 4.111 <i>Wordcloud</i> Topic #3 Sentimen Negatif..... | 105 |
| Gambar 4.112 <i>Wordcloud</i> Topic #4 Sentimen Negatif..... | 105 |
| Gambar 4.113 <i>Wordcloud</i> Topic #5 Sentimen Negatif..... | 105 |
| Gambar 4.114 <i>Intertopic Distance Map</i> Topik-topik <i>Headline</i> Bersentimen Negatif..... | 106 |
| Gambar 4.115 <i>Use Case Diagram</i> untuk <i>Prototype</i> Aplikasi topsenaindonesia.com..... | 107 |
| Gambar 4.116 Tampilan Awal Sistem | 107 |
| Gambar 4.117 Antarmuka Pembuatan Akun Pengguna..... | 108 |
| Gambar 4.118 <i>Dashboard</i> Sistem | 108 |
| Gambar 4.119 Antarmuka Halaman "Proyek Saya" | 109 |
| Gambar 4.120 Antarmuka Form "Membuat Proyek"..... | 110 |
| Gambar 4.121 List Proyek Pemodelan Topik | 110 |
| Gambar 4.122 Project Pemodelan Topic dengan Status Selesai | 111 |

Daftar Code Snippet

| | |
|---|----|
| <i>Code Snippet 4.1. Kode Python untuk Scrapping situs antaranews.com</i> | 26 |
| <i>Code Snippet 4.2 Kode Python untuk Scrapping situs cnnindonesia.com</i> | 28 |
| <i>Code Snippet 4.3. Kode Python untuk Scrapping situs jpnn.com</i> | 30 |
| <i>Code Snippet 4.4. Kode Python untuk Scrapping situs kompas.com</i> | 32 |
| <i>Code Snippet 4.5. Kode Python untuk Scrapping situs liputan6.com</i> | 34 |
| <i>Code Snippet 4.6. Kode Python untuk Scrapping situs merdeka.com</i> | 36 |
| <i>Code Snippet 4.7. Kode Python untuk Scrapping situs detik.com</i> | 39 |
| <i>Code Snippet 4.8. Kode Python untuk Scrapping situs okezone.com</i> | 41 |
| <i>Code Snippet 4.9. Kode Python untuk Scrapping situs rmol.id</i> | 43 |
| <i>Code Snippet 4.10. Kode Python untuk Scrapping situs sindonews.com</i> | 45 |
| <i>Code Snippet 4.11. Kode Python untuk Scrapping situs suara.com</i> | 48 |
| <i>Code Snippet 4.12. Kode Python untuk Scrapping situs tempo.co</i> | 50 |
| <i>Code Snippet 4.13 Bash script untuk mengeksekusi perintah scrapping situs</i> | 51 |
| <i>Code Snippet 4.14 Python script untuk pelabelan dataset</i> | 83 |
| <i>Code Snippet 4.15. Query Pembuatan Dataset Bersentimen Positif, Netral, dan Negatif</i> .. | 89 |
| <i>Code Snippet 4.16. Kode Python untuk Data Preprocessing</i> | 91 |
| <i>Code Snippet 4.17. Kode Python Pemodelan Topik LDA untuk Sentimen Positif</i> | 93 |
| <i>Code Snippet 4.18. Kode Python Pemodelan Topik LDA untuk Sentimen Netral</i> | 95 |
| <i>Code Snippet 4.19. Kode Python Pemodelan Topik LDA untuk Sentimen Negatif</i> | 97 |

BAB 1

Pendahuluan

1.1 Latar Belakang

Berita merupakan salah satu bentuk informasi yang banyak dicari oleh masyarakat (Wangsa et al., 2014). Berita yang pada awalnya dicetak, sekarang seiring dengan majunya teknologi informasi maka banyak media massa yang didigitalkan dan dipublikasikan secara *online* dalam bentuk *website* atau portal berita (Pratiwi, 2014). Secara umum publikasi berita di media massa memiliki banyak fungsi, yaitu: (1) penyebaran informasi, (2) penyampaian interpretasi atau opini terhadap suatu informasi, (3) pembentukan kesepakatan, (4) korelasi/penghubung antar kelompok masyarakat, (5) transmisi warisan budaya, (6) mengekspresikan nilai-nilai budaya untuk melestarikan identitas masyarakat (Yuniati, 2002).

Pesan pertama yang dibaca oleh pembaca berita adalah judul berita sebelum membaca keseluruhan informasi yang disajikan dalam sebuah berita. Ringkasan informasi pada sebuah berita pada umumnya terangkum di bagian judul berita. Struktur kalimat dalam judul berita dibuat efisien dengan penekanan pada unsur siapa dan apa yang dilakukan. Kata kerja, kata benda, kata sifat, dan kata keterangan dipilih yang singkat dan tajam untuk menyampaikan pesan kepada pembaca (Winiharti, 2011).

Headline atau tajuk adalah kalimat pendek atau frasa yang ditempatkan secara mencolok pada sebuah berita dengan menggunakan huruf yang menonjol (Badan Pengembangan Bahasa dan Perbukuan, 2016). *Headline news* atau berita utama adalah berita yang disepakati oleh dewan redaksi dan paling layak untuk ditampilkan di halaman depan surat kabar dan menggunakan judul yang dicetak dengan huruf lebih besar dari berita lainnya dan menarik mengundang rasa penasaran para pembaca (Wandik et al., 2017).

Dinamisnya berita-berita yang dipublikasikan di media massa *online*, menyebabkan media massa *online* menjadi salah satu sumber berita yang banyak dikonsumsi oleh masyarakat yang haus akan informasi terkini. Derasnya arus informasi dari berbagai media massa *online* dapat menjadi sumber data yang berlimpah untuk keperluan penelitian di dalam mengungkap tema atau topik apa saja yang sedang dibicarakan media massa *online*.

Topic Modelling adalah salah satu teknik di dalam *Natural Language Processing* (NLP) untuk menganalisis teks (Aggarwal & Zhai, 2013) yang berupa algoritma untuk

mengidentifikasi pola-pola tersembunyi dari serangkaian kata dengan menggunakan teknik pendistribusian kata-kata di dalam sekumpulan dokumen, *output* dari *topic modelling* berupa sekumpulan topik yang terdiri dari beberapa gugus kata yang muncul bersamaan di dalam dokumen berdasarkan pola tertentu (Jacobi et al., 2016). Metode *topic modelling* banyak ragamnya, misalnya *Latent Semantic Allocation* (Bergamaschi & Po, 2015), *Probabilistic Latent Semantic Analysis* (Lu et al., 2011), dan *Latent Dirichlet Allocation* (D. M. Blei et al., 2003). Setiap model memiliki beberapa tahapan, dan setiap tahapan akan berpengaruh terhadap hasil yang diharapkan (Hagen, 2018).

Sentiment analysis juga merupakan salah satu teknik di dalam NLP. *Sentiment analysis* dapat dilakukan dengan menggunakan dua pendekatan, yaitu *Machine Learning Approach* dan *Lexicon-based Approach* (Medhat et al., 2014). Pendekatan yang digunakan pada penelitian ini adalah pendekatan *Machine Learning*, yaitu dengan menggunakan metode *Recurrent Neural Network*.

Recurrent Neural Network (RNN) adalah salah satu dari metode *Neural Network* yang digunakan untuk menyelesaikan persoalan terkait dengan NLP. Metode RNN menggunakan perhitungan bobot secara berulang, sehingga nilai akurasi yang diperoleh lebih lebih baik dari jaringan saraf tiruan sederhana (Ivanedra & Mustikasari, 2019). Metode RNN dipakai dengan tujuan agar mesin mampu memahami bahasa manusia termasuk cara berkomunikasi, mendengarkan, mengenali percakapan, dan memahami tata bahasa tertentu (Primartha, 2018). Jenis RNN yang digunakan di dalam penelitian ini adalah *Long Short-term Memory* (LSTM). LSTM terbukti dapat menutupi kekurangan dari RNN terkait ketidakmampuannya di dalam menyimpan memori agar dapat diseleksi dan menambah mekanisme *attention* agar setiap kata sesuai dengan konteksnya (Ivanedra & Mustikasari, 2019).

Literatur-literatur yang penulis sebutkan menjadi dasar bagi penulis di dalam melakukan penelitian yang berhubungan dengan analisis sentimen dan pemodelan topik di bidang jurnalistik, khususnya pada media massa *online*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang penulis paparkan sebelumnya, rumusan masalah penelitian ini adalah bagaimana melakukan pemodelan topik pada hasil analisis sentimen terhadap *headline* berita *online* berbahasa Indonesia dan menerapkan model ke dalam sebuah *prototype* aplikasi berbasis web.

1.3 Batasan Masalah

Penelitian ini memerlukan pembatasan masalah agar pembahasannya menjadi lebih terarah dan tidak bias. Penulis membuat pembatasan-pembatasan di dalam melakukan penelitian ini yaitu:

- a. Obyek di dalam penelitian ini hanya berfokus kepada *headline* berita *online* berbahasa Indonesia.
- b. Penelitian ini tidak memilah antara judul berita yang bersifat *content representative* dan judul berita yang termasuk *click bait*.
- c. Metode klasifikasi pada penelitian ini menggunakan RNN (*Recurrent Neural Networks*) dan type RNN yang penulis pakai adalah LSTM (*Long Short-term Memory*) dengan melakukan perbandingan beberapa skenario *hyperparameters tuning* untuk memperoleh model terbaik. Metode *hyperparameters tuning* yang digunakan adalah metode *trial-and-error*.
- d. Metode *Topic Modelling* yang digunakan adalah *Latent Dirichlet Allocation* (LDA)
- e. Aplikasi yang dibangun berupa *prototype* yang mampu menunjukkan hasil pemodelan topik terhadap hasil sentimen analisis pada *headline* berita *online* berbahasa Indonesia.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membangun model sekaligus sistem yang dapat mengidentifikasi sentimen positif, netral, dan negatif pada *headline* berita *online* yang berbahasa Indonesia.

Klasifikasi *headline* berita hasil dari proses analisis sentimen tersebut selanjutnya akan digunakan sebagai data masukan pada proses pemodelan topik sehingga akan diperoleh topik-topik apa saja yang dibicarakan oleh media massa online dari masing-masing klasifikasi.

1.5 Manfaat Penelitian

Manfaat-manfaat yang diperoleh dari hasil penelitian ini dapat dilihat dari beberapa sudut pandang, yaitu bagi penulis, pengguna sistem, dan peneliti lain.

- a. Bagi penulis sendiri, dapat menambah wawasan di bidang *Machine Learning*.
- b. Bagi pengguna sistem dapat mengetahui topik-topik apa saja yang dibicarakan oleh media massa *online* hanya dengan memasukkan suatu kata kunci terkait isu berita tertentu. Pengguna sistem juga dapat memperoleh topik-topik berita yang sudah diklasifikasikan ke dalam sentimen positif, netral, dan negatif. Topik-topik pembicaraan media massa online disajikan menggunakan visualisasi *Wordcloud*, sehingga pengguna

sistem dapat lebih mudah menangkap *insight*/wawasan yang terkandung pada topik-topik yang dihasilkan oleh sistem. *Insight* yang diperoleh dapat digunakan oleh pengguna dengan berbagai macam tujuan, sebagai contoh bagi pemangku kebijakan dapat memanfaatkannya untuk bahan masukan di dalam pembuatan kebijakan terhadap suatu isu berita yang berkembang di masyarakat.

- c. Bagi peneliti lain, dapat menjadi salah satu literatur acuan di dalam penelitian yang berkaitan dengan bidang pemodelan topik terhadap hasil analisis sentimen.

1.6 Sistematika Penulisan

Penulisan laporan penelitian ini disusun menggunakan sistematika penulisan berikut ini.

BAB 1 PENDAHULUAN

Bagian Pendahuluan ini berisi latar belakang masalah, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bagian Tinjauan Pustaka ini berisi landasan teori yang berupa teori pendukung pada penelitian ini dan berisi kajian Pustaka atau literature review terhadap penelitian-penelitian sebelumnya yang berkaitan dengan *topic modelling* dan *analysis sentiment*.

BAB 3 METODOLOGI

Bagian metodologi ini berisi langkah-langkah penulis di dalam melaksanakan penelitian dan menyelesaikan masalah yang diteliti.

BAB 4 HASIL DAN PEMBAHASAN

Bagian Hasil dan Pembahasan ini berisi hasil-hasil yang diperoleh dari tahapan penelitian dan pembahasannya.

BAB 5 PENUTUP

Bagian Penutup ini berisi kesimpulan dan saran terhadap hasil penelitian yang sudah dicapai.

BAB 2

Tinjauan Pustaka

2.1 Landasan Teori

2.1.1 Topic Modelling

Topic Modelling atau pemodelan topik adalah sebuah teknik yang dilengkapi berbagai macam algoritma yang bertujuan untuk mengungkap, menemukan, dan menjelaskan struktur tematik dari sekumpulan dokumen (D. Blei et al., 2010). *Topic modelling* juga digunakan untuk menemukan pola kelompok kata dalam suatu dokumen dan mengungkap topik tersembunyi yang terbentuk dari sekumpulan kata pada dokumen (Wang et al., 2011). *Output* yang dihasilkan dari *topic modelling* adalah sekumpulan topik yang terdiri dari beberapa kluster kata yang muncul bersamaan di dalam dokumen berdasarkan pola tertentu (Jacobi et al., 2016).

Model *topic modelling* yang digunakan penulis di dalam penelitian ini adalah model *Latent Dirichlet Allocation* (LDA). Model LDA ini dicetuskan oleh David M. Blei pada tahun 2003. Model LDA adalah model probabilitas generatif dari suatu *corpus*. Ide dasarnya adalah dokumen-dokumen pada suatu *corpus* direpresentasikan sebagai campuran berbagai topik tersembunyi, dan setiap topik dikarakteristikan berdasarkan distribusi berbagai kata (D. M. Blei et al., 2003).

Penulis mengaplikasikan *LDA Topic Modelling* pada penelitian ini dengan menggunakan bahasa pemrograman *python* dan *library* Gensim. Gensim adalah *library* untuk *python* yang khusus menangani permasalahan *topic modelling*. Gensim sudah memiliki model-model yang sudah *built-in* seperti *Latent Dirichlet Allocation (LDA)*, *Latent Semantic Analysis (LSA)*, *Random Projection (RP)*, dan *Hierarchical Dirichlet Process (HDP)* (Rehurek & Sojka, 2010).

2.1.2 Sentiment Analysis

Sentiment analysis adalah teknik untuk mengidentifikasi dan mengelompokkan polaritas teks-teks yang terdapat pada dokumen, kalimat, dan level frasa (Ankit & Saleena, 2018). Secara umum *sentiment analysis* adalah proses klasifikasi dokumen tekstual ke dalam beberapa kelas seperti sentimen positif dan sentimen negatif (Srivastava et al., 2019).

Recurrent Neural Network (RNN) adalah salah satu metode yang dapat digunakan untuk melakukan klasifikasi. Metode RNN dipakai dengan tujuan agar mesin mampu

memahami bahasa manusia termasuk cara berkomunikasi, mendengarkan, mengenali percakapan, dan memahami tata bahasa tertentu (Primartha, 2018). Metode RNN menggunakan perhitungan bobot secara berulang, sehingga nilai akurasi yang diperoleh lebih lebih baik dari jaringan saraf tiruan sederhana (Ivanedra & Mustikasari, 2019).

Long Short-term Memory (LSTM) adalah jenis RNN yang dikembangkan untuk menyempurnakan kekurangan RNN berkaitan dengan adanya *vanishing gradient* yang terjadi pada data masukan berupa *sequence* yang panjang (Chen, 2016). *Vanishing gradient* adalah suatu kondisi suatu *neural network* dengan banyak *layer* dilatih dengan optimasi *gradient descent* mengalami nilai *gradient* mendekati 0 atau sama dengan 0 (Suyudi; Djamal Emeranda C; Maspupah Asri, 2019).

Jaringan RNN-LSTM menambahkan beberapa komponen seperti *forget gate* (f_t), *input gate* (i_t), *input modulation gate* (\hat{c}_t), *output gate* (o_t), dan *cell state* (c_t) agar jaringan syaraf tiruan dapat mempelajari *sequence* yang Panjang. Komponen utama dari LSTM adalah *cell state* (c_t) yang berfungsi sebagai sel memori yang dapat menyimpan informasi sebelumnya dan informasi saat ini (Pasupa & Seneewong Na Ayutthaya, 2019). Persamaan untuk *cell state* (c_t) dapat dilihat pada persamaan 2.1.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \quad (2.1)$$

Cell state dihasilkan dari penjumlahan antara *cell state* sebelumnya (c_{t-1}) yang dikalikan dengan *forget gate* (f_t) dengan *input gate* (i_t) yang dikalikan dengan *input modulation gate* (\hat{c}_t).

Forget gate (f_t) berfungsi untuk menentukan informasi mana saja yang akan dilupakan dengan menggunakan fungsi aktivasi sigmoid (σ) yang ditunjukkan pada persamaan 2.2.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (2.2)$$

Sebuah *forget gate* (f_t) hanya menghasilkan nilai 0 dan 1, nilai 0 menunjukkan bahwa informasi dapat dilupakan, sedangkan nilai 1 menunjukkan bahwa informasi harus disimpan. Nilai 0 dan 1 diperoleh melalui fungsi aktivasi sigmoid (σ).

Input gate (i_t) menentukan informasi mana saja yang akan disimpan pada *cell state* (c_t) dengan menggunakan fungsi aktivasi sigmoid (σ) yang ditunjukkan pada persamaan 2.3.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (2.3)$$

Input modulation gate (\hat{c}_t) berperan sebagai kandidat *cell state* dengan menggunakan fungsi aktivasi *tangent tanh* yang ditunjukkan pada persamaan 2.4.

$$\hat{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \quad (2.4)$$

Output gate (o_t) menentukan informasi mana saja yang akan dipertimbangkan untuk berada pada *hidden state* (h_t) dengan menggunakan fungsi aktivasi sigmoid ditunjukkan pada persamaan 2.5.

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (2.5)$$

Sebuah *hidden state* (h_t) mengandung semua informasi yang akan digunakan pada proses *sequential* berikutnya. *Hidden state* (h_t) dihitung dengan mengalikan *output gate* (o_t) dengan *cell state* (c_t) di bawah fungsi aktivasi tanh. Persamaannya dapat dilihat pada persamaan 2.6.

$$h_t = o_t \cdot \tanh(c_t) \quad (2.6)$$

2.1.3 Hyperparameters Tuning

Hyperparameters adalah sekumpulan set parameter yang berada di luar model dan berguna untuk meningkatkan akurasi model di dalam melakukan prediksi. *Hyperparameters tuning* adalah proses pemilihan dan penerapan beberapa parameter seperti jumlah *layer*, *recurrent units*, *input size*, *batch size*, *epoch*, maupun nilai *dropout* (Reimers & Gurevych, 2017). Proses pemilihan *hyperparameter* pada praktiknya banyak dilakukan peneliti melalui pendekatan *trial-and-error*, *grid search*, dan *random search* (Luketina et al., 2016).

Metode *hyperparameters tuning* yang penulis gunakan di dalam penelitian ini adalah metode *trial-and-error*. *Hyperparameters tuning* dengan metode *trial-and-error* adalah metode mengatur nilai-nilai *hyperparameter* yang dilakukan secara manual berdasarkan pengalaman terhadap hasil dari uji coba berkali-kali dengan menerapkan nilai yang berbeda-beda dari tiap *hyperparameter* (Probst et al., 2019).

2.1.4 Prototyping

Prototype adalah model suatu sistem yang dapat berfungsi secara terbatas pada fungsi pokok sistem dan umumnya menjadi versi awal dari sebuah sistem yang akan dikembangkan lebih lanjut. *Prototyping* adalah suatu proses pembuatan model secara lebih cepat dan hemat biaya serta model yang dihasilkan dapat berfungsi meskipun secara terbatas untuk keperluan pengujian-pengujian.

Terdapat empat metode *prototyping*, yaitu:

1. *Illustrative*, yaitu metode pembuatan *prototype* yang berupa *mockup* tampilan baik tampilan *input* dan tampilan *output*.
2. *Simulated*, yaitu metode pembuatan *prototype* yang berupa simulasi suatu sistem tetapi tidak menggunakan data riil.

3. *Functional*, yaitu metode pembuatan *prototype* dengan menjalankan fungsi-fungsi pokok dari sistem dan menggunakan data riil.
4. *Evolutionary*, yaitu metode pengembangan *prototype* yang akan menghasilkan model sebagai bagian dari sistem yang final (Ogedebe & Jacob, 2012).

Pada penelitian ini, penulis menggunakan metode *functional prototyping* untuk menerapkan model LSTM dan LDA dengan merancang sebuah aplikasi yang dapat menjalankan fungsi-fungsi pokok dari proses analisis sentimen dan pemodelan topik dan mampu menghasilkan output berupa *wordcloud* dan *intertopic distance map*.

2.2 Kajian Pustaka

Kajian pustaka sangat diperlukan agar penelitian ini memiliki dasar dengan memaparkan penelitian-penelitian sebelumnya. Penulis mempelajari artikel-artikel ilmiah yang berkaitan dengan *topic modelling* dan *sentiment analysis*.



Tabel 2.1 Daftar *Literature Review*

| No | Judul | Penulis, Tahun | Permasalahan | Bidang NLP | Metode | Sumber Data | Hasil |
|----|--|-----------------------|--|------------------------|--|---|---|
| 1 | <i>Towards Predicting Trend of Scientific Research Topics using Topic Modeling</i> | (Abuhay et al., 2018) | <p>Penelitian ini berkaitan dengan prediksi tren topik-topik penelitian yang akan datang. Prediksi ini dapat diperoleh dari pendapat para pakar dan dari hasil analisis kuantitatif.</p> <p>Prediksi dari para pakar dapat menghasilkan kesimpulan yang bias. Oleh karena itu banyak penulis yang menggunakan analisis kuantitatif seperti <i>bibliometrics</i>, <i>scientometrics</i>, dan <i>informetrics</i>.</p> <p>Namun metode analisis kuantitatif tersebut</p> | <i>Topic Modelling</i> | <i>Non-negative Matrix Factorization (NMF)</i> | Penulis menggunakan data dari <i>proceeding International Conference on Computational Science (ICCS)</i> dari tahun 2001 hingga 2017. | Hasil penelitian ini berupa data terstruktur secara berurutan waktu dan dapat digunakan untuk memprediksi tren penelitian yang akan datang. |

| | | | | | | | |
|---|---|-----------------------------|--|------------------------|------------------------------------|---|---|
| | | | memiliki banyak keterbatasan, sehingga diperlukan metode alternatif lain seperti pemodelan topik. | | | | |
| 2 | <i>Quantitative Analysis of Large Amounts of Journalistic Texts Using Topic Modelling</i> | (Jacobi et al., 2016) | Konten berita online yang sangat banyak dan tersedia secara digital menjadi tantangan baru bagi penulis untuk menganalisis topik apa saja yang terdapat pada teks-teks konten berita tersebut. | <i>Topic Modelling</i> | <i>Latent Dirichlet Allocation</i> | Penulis menggunakan koran harian <i>New York Times</i> sebagai sumber data. | Penulis mendemonstrasikan manfaat dari penelitian jurnalisme. Penulis mengambil studi kasus pada koran <i>New York Times</i> berkaitan dengan teknologi nuklir dari tahun 1945 hingga sekarang. Penulis berkesimpulan bahwa dengan metode <i>Latent Dirichlet Allocation</i> (LDA) mampu menyelesaikan pemodelan topik dengan baik untuk menganalisis tren dan pola-pola pada arsip digital media <i>online</i> . |
| 3 | <i>Twitter Topic Modeling on Football News</i> | (Hidayatullah et al., 2018) | Twitter adalah salah satu platform media sosial yang dapat menjadi sumber informasi yang | <i>Topic Modelling</i> | <i>Latent Dirichlet Allocation</i> | Penulis menggunakan postingan tweet pada twitter sebagai sumber data. | Penulis menggunakan <i>Latent Dirichlet Allocation</i> untuk melakukan pemodelan topik terhadap |

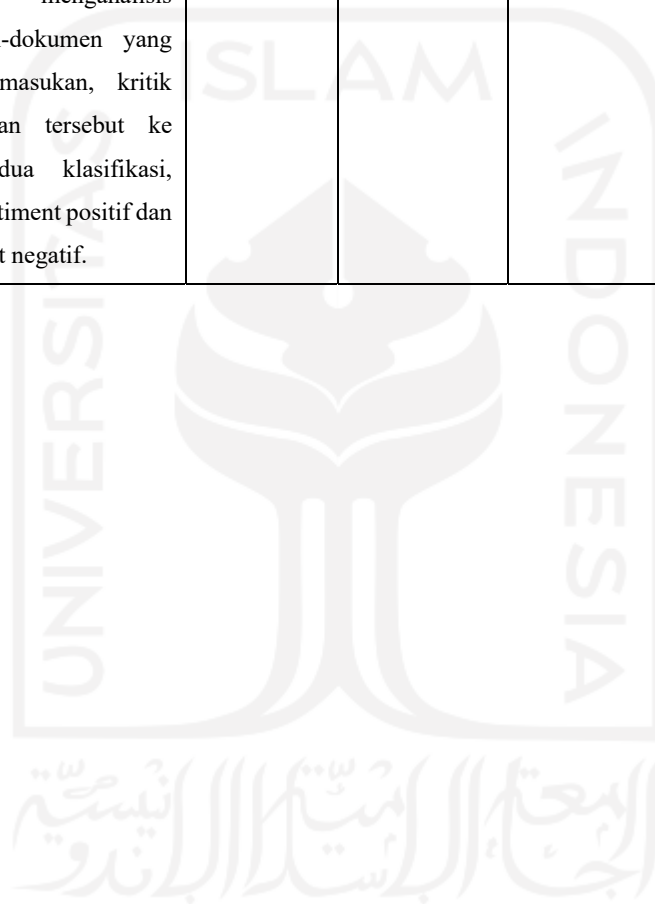
| | | | | | | | |
|---|---|---------------------------------------|---|------------------------|------------------------------------|--|--|
| | | | berkaitan dengan sepak bola. Sepak bola adalah olah raga yang populer di Indonesia. Masyarakat banyak yang ingin tahu berita sepak bola, khususnya terkait berita terbaru, prediksi, hasil pertandingan, transfer pemain, rumor dan lain-lain. | | | | tweet berbahasa Indonesia terkait dunia sepak bola. Hasil dari analisis konten, diperoleh beberapa topik yang dapat menjadi <i>insight</i> bermanfaat terkait analisis pra-tanding, update pertandingan live, prestasi klub sepak bola, dan lain-lain. |
| 4 | Analisis Topik Informasi Publik Media Sosial Di Surabaya Menggunakan Pemodelan <i>Latent Dirichlet Allocation (LDA)</i> | (Made Kusnanta Bramantya Putra, 2017) | Radio Suara Surabaya adalah radio dengan konsep <i>Radio News</i> dan informasi. Radio Suara Surabaya memiliki siaran interaktif berbasis jurnalistik masyarakat yang melibatkan partisipasi warga dalam melaporkan peristiwa kepada penyiar radio yang sedang bertugas. Laporan yang masuk disebarkan kembali oleh penyiar | <i>Topic Modelling</i> | <i>Latent Dirichlet Allocation</i> | Penulis menggunakan data kumpulan laporan dari masyarakat yang diterima oleh Radio Suara Surabaya. | Penulis menggunakan <i>Latent Dirichlet Allocation</i> untuk pemodelan topik. Eksperimen yang dilakukan menghasilkan 4 buah topik. Hasil eksperimen memperoleh nilai <i>perplexity</i> sebesar 213.41 dan diuji kemudahan interpretasinya menggunakan uji koherensi topik, dan dapat disimpulkan bahwa topik yang dihasilkan dapat |

| | | | | | | | |
|---|--|-------------------------|--|-------------------------------|---|--|---|
| | | | <p>maupun pengelola media sosial radio tersebut. Tingginya jumlah laporan yang masuk dan beragamnya topik laporan akan menimbulkan kesulitan dalam mengidentifikasi topik yang tepat. Dalam kondisi demikian, dibutuhkan pemodelan topik yang mampu secara otomatis mengklasifikasikan pesan media social ke dalam topik-topik yang muncul dari hasil pemodelan.</p> | | | | <p>diinterpretasikan dengan baik oleh manusia.</p> |
| 5 | <p>Pemodelan Topik Dengan <i>Latent Dirichlet Allocation</i></p> | <p>(Zulhanif, 2016)</p> | <p>Pemodelan topik pada tweet dengan kata kunci #bandung.</p> | <p><i>Topic Modelling</i></p> | <p><i>Latent Dirichlet Allocation</i></p> | <p>Penulis menggunakan postingan tweet pada twitter sebagai sumber data.</p> | <p>Hasil penelitian menunjukkan sebanyak 24 topik diperoleh berkaitan tweet dengan kata kunci #bandung. Topik-topik yang dihasilkan divisualisasikan menggunakan <i>wordcloud</i> dan <i>intertopic distance map</i>.</p> |

| | | | | | | | |
|---|---|---|---|---------------------------|--|---|--|
| 6 | <i>Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features</i> | (Pasupa & Seneewong Na Ayutthaya, 2019) | Studi komparasi efektifitas analisis sentimen menggunakan <i>word-embedding</i> , <i>POS-tag</i> , dan <i>sentic features</i> . | <i>Sentiment Analysis</i> | <i>LSTM, Bi-LSTM, dan CNN</i> | Penulis menggunakan kumpulan dongeng anak-anak negeri Thailand sebagai sumber data. | Penulis melakukan eksperimen klasifikasi sentiment terhadap kalimat-kalimat yang terdapat pada dongeng anak-anak Thailand menggunakan 3 model deep learning, yaitu LSTM, Bi-LSTM, dan CNN dengan mengombinasikan <i>word-embedding</i> , <i>POS-tag</i> , dan <i>sentic features</i> . Model dengan performa terbaik untuk klasifikasi sentiment pada dataset tersebut adalah model CNN. |
| 7 | <i>Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision</i> | (Fitri et al., 2019) | Penulis mengambil studi kasus kampanye Anti-LGBT di Indonesia. Penelitian ini bertujuan untuk mengetahui tendensi dari publik berkaitan kampanye anti-LGBT dari komentar- | <i>Sentiment Analysis</i> | <i>Naïve Bayes, Decision Tree, random Forest Algorithm</i> | Penulis menggunakan postingan tweet pada twitter sebagai sumber data. | Sentiment analysis yang diperoleh dalam penelitian ini menunjukkan bahwa pengguna Twitter di Indonesia cenderung lebih netral di dalam berkomentar terkait kampanye Anti-LGBT. Diperoleh tingkat keakuratan sebesar 86.43% |

| | | | | | | | |
|---|---|------------------------|--|---------------------------|-------------------------------------|---|--|
| | <i>tree, and random forest algorithm</i> | | komentar yang di-posting di Twitter. | | | | dari dataset uji menggunakan Naïve Bayes. Tingkat akurasi ini lebih tinggi dari pada algoritma lainnya yang dipakai yaitu Decision Tree dan Random Forest yang hanya sebesar 82.91%. |
| 8 | <i>Sentiment Analysis Berbasis Big Data</i> | (Nomleni et al., n.d.) | Pemerintah sebagai pelayan masyarakat idealnya mampu memberikan pelayanan yang optimal kepada masyarakat melalui kebijakan-kebijakan yang langsung menyentuh kepentingan masyarakat. Pemerintah didalam memberikan pelayanan juga tidak lepas dari kekurangan. Oleh karena itu masyarakat diperbolehkan memberikan masukan, kritik, dan saran. | <i>Sentiment Analysis</i> | <i>Support Vector Machine (SVM)</i> | Sumber data berasal dari facebook dan twitter Media Center Pemkot Surabaya Sapawarga. | Setiap keluhan yang masuk dianalisis menggunakan Teknik SVM, sehingga diperoleh polaritas dari keluhan yang dapat digunakan sebagai acuan bagi tim Media Center di dalam mengevaluasi dan meningkatkan pelayanan kepada masyarakat |

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | Pemerintah kota Surabaya memerlukan sistem yang mampu menganalisis dokumen-dokumen yang berupa masukan, kritik dan saran tersebut ke dalam dua klasifikasi, yaitu sentiment positif dan sentiment negatif. | | | | |
|--|--|--|--|--|--|--|--|



Setelah melaksanakan *literature review*, penulis mendapati bahwa pada penelitian-penelitian sebelumnya yang berkaitan dengan *topic modelling*, beberapa peneliti menggunakan metode *Non-negative Matrix Factorization (NMF)* dan *Latent Dirichlet Allocation (LDA)*. Sedangkan untuk *sentiment analysis*, beberapa peneliti menggunakan metode LSTM, *Naïve Bayes Classifier*, dan *Support Vector Machine*. Sumber dataset yang dipergunakan di dalam penelitian-penelitian sebelumnya adalah *proceeding* konferensi internasional, surat kabar, dan *tweet*.

Agar penelitian ini memiliki *gap* terhadap penelitian-penelitian sebelumnya, maka penulis melakukan penelitian yang tentang pemodelan topik terhadap hasil analisis sentimen pada *headline* berita *online* berbahasa Indonesia sekaligus model yang dihasilkan dari proses sentimen analisis dan pemodelan topik tersebut dibuat *prototype* berupa aplikasi berbasis web.



BAB 3

Metodologi

3.1 Data Penelitian

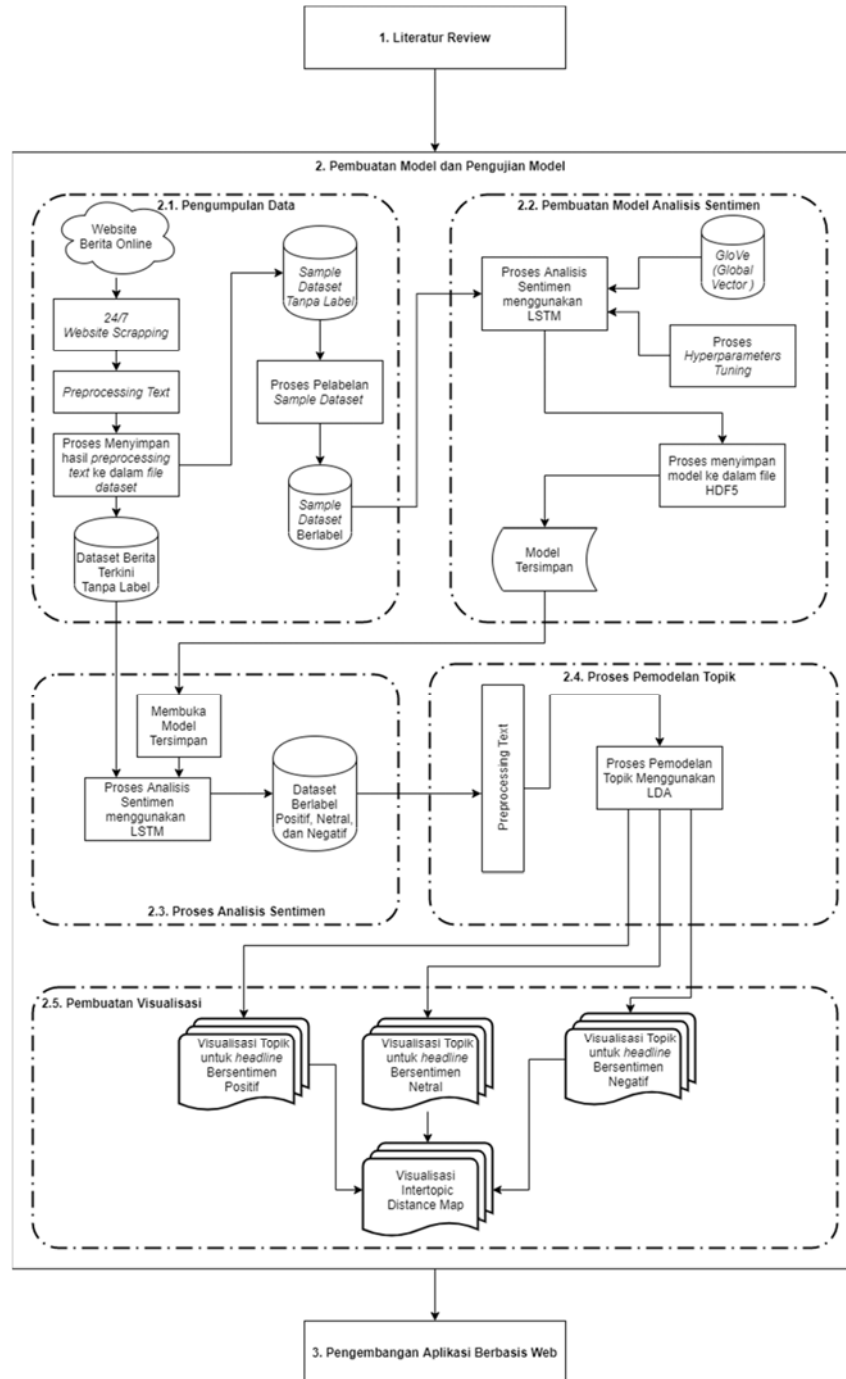
Data yang diperlukan di dalam penelitian ini adalah data tajuk berita yang penulis peroleh melalui proses *scrapping* situs-situs berita *online* berbahasa Indonesia. Tabel 3.1 menunjukkan daftar situs-situs berita online berbahasa Indonesia yang menjadi sumber data di dalam penelitian ini.

Tabel 3.1 Daftar Situs Berita *Online* Berbahasa Indonesia

| No | Situs Berita | Keterangan |
|----|------------------|--|
| 1 | antaranews.com | Situs berita online ini beroperasi di bawah naungan Perusahaan Umum Lembaga Kantor Berita Nasional Antara yang dimiliki oleh Pemerintah Indonesia. |
| 2 | cnnindonesia.com | Situs berita online ini beroperasi di bawah naungan Cable News Network Indonesia milik Trans Media bekerja sama dengan WarnerMedia, divisi dari AT&T. |
| 3 | detik.com | Situs berita online ini beroperasi di bawah naungan Grup CT Corp. |
| 4 | jpnn.com | Situs berita online ini beroperasi di bawah naungan PT. Jaringan Pemberitaan Nusantara Negeriku. |
| 5 | kompas.com | Situs berita online ini beroperasi di bawah naungan PT. Kompas Cyber Media (Kompas Gramedia Digital Group). |
| 6 | liputan6.com | Situs berita online ini beroperasi di bawah naungan PT Kreatif Media Karya (KMK). Perusahaan KMK ini merupakan anak perusahaan PT Elang Mahkota Teknologi (Emtek). |
| 7 | merdeka.com | Situs berita online ini beroperasi di bawah naungan PT KapanLagi Dot Com Networks. |
| 8 | okezone.com | Situs berita online ini beroperasi di bawah naungan PT Media Nusantara Citra Tbk (MNC). |
| 9 | rmol.id | Situs berita online ini beroperasi di bawah naungan PT. Muara Multi Media. |
| 10 | sindonews.com | Situs berita online ini beroperasi di bawah naungan PT Media Nusantara Citra Tbk (MNC). |
| 11 | suara.com | Situs berita online ini beroperasi di bawah naungan PT. Arkadia Media Nusantara. |
| 12 | tempo.co | Situs berita online ini beroperasi di bawah naungan PT Info Media Digital. |

3.2 Langkah-langkah Penelitian

Langkah-langkah penelitian diperlukan agar penelitian ini memiliki alur yang dapat dipahami dan mudah diikuti. Penulis menyusun langkah-langkah penelitian yang disajikan dalam bentuk diagram alir pada gambar 1.



Gambar 3.1 Langkah-langkah Penelitian

Berdasarkan gambar 1, tahapan-tahapan yang dilaksanakan pada penelitian ini adalah tahap pertama *literature review*, tahap kedua pembuatan model dan pengujian model, tahap ketiga proses pengembangan aplikasi berbasis web, dan tahap keempat proses implementasi.

3.2.1 Literature Review

Langkah pertama pada penelitian ini adalah menelaah literatur-literatur yang sesuai dengan topik penelitian yang penulis peroleh dari jurnal-jurnal ilmiah dan buku-buku tentang *machine learning*.

3.2.2 Pembuatan Model dan Pengujian Model

Langkah pembuatan model dan pengujian model ini dirinci lagi menjadi 5 sub proses, yaitu:

a. Pengumpulan Data

Penelitian ini memerlukan data tajuk berita dari berbagai media massa online. Untuk memperoleh data yang berupa tajuk berita online berbahasa Indonesia, penulis menggunakan cara scrapping pada situs-situs berita online. Proses scrapping situs berita online dilakukan dengan menanam kode dalam bahasa pemrograman python di Virtual Private Server (VPS). Eksekusi kode python tersebut dijadwalkan menggunakan CRON dengan interval setiap 1 menit.

Pada penelitian ini, untuk tujuan uji coba sistem, penulis menggunakan data hasil dari query data tajuk berita yang mengandung term “coronavirus disease 2019” atau “Covid-19” dari dataset, namun pada implementasi di aplikasi berbasis web yang dikembangkan penulis, pengguna sistem dapat memasukkan berbagai *keyword* atau kata kunci yang diinginkan.

Coronavirus disease 2019 adalah penyakit yang disebabkan virus corona yang pertama kali menjangkiti warga Wuhan di Tiongkok pada akhir tahun 2019 yang lalu dan menyebabkan pandemi (wabah mendunia) serta menjadi masalah utama kesehatan masyarakat dunia (Lai et al., 2020). Coronavirus ini dapat menginfeksi masyarakat dari berbagai tingkatan usia. Kelompok masyarakat yang paling beresiko terhadap dampak infeksi Covid-19 ini adalah kelompok usia tua (usia 60 tahun keatas) dan kelompok masyarakat yang memiliki riwayat penyakit kronis seperti diabetes, gangguan jantung, paru-paru, dan kanker (WHO, 2020).

Data yang diperoleh dari hasil proses *scrapping* situs berita online penulis bagi menjadi dua bagian. Dataset bagian pertama penulis gunakan sebagai *sample dataset* atau data latih untuk pembuatan model *sentiment analysis*, dan bagian kedua adalah data yang selalu mengalir dan terus menerus tersimpan ke dalam dataset yang berisi *headline* berita terkini. Dataset bagian kedua inilah yang nantinya akan digunakan di dalam proses analisis sentimen, pemodelan topik dan visualisasi.

Dataset bagian pertama hasil dari *scrapping* situs berita online masih berupa *unlabeled dataset*, yaitu dataset yang belum memiliki label positif, negatif, dan netral. Proses pelabelan *sample dataset* ini dilakukan secara manual sehingga hasil akhirnya dihasilkan *sample dataset* yang berlabel positif, negatif, dan netral.

b. Pembuatan Model LSTM untuk Analisis Sentimen

Tahapan selanjutnya yang dilakukan penulis adalah pembuatan model analisis sentimen. Model ini perlu dibuat agar suatu saat melakukan proses analisis sentimen maka tidak perlu lagi melakukan training dataset yang cukup memakan waktu. Hanya dengan menggunakan model yang sudah dibuat sebelumnya, maka dapat dilakukan proses analisis sentimen sewaktu-waktu dan berkali-kali dengan lebih cepat.

Pembuatan model yang baik tentunya memerlukan kombinasi konfigurasi hyperparameter yang optimal. Proses pembuatan model ini dilakukan dengan menyesuaikan berbagai parameter seperti jumlah layer, recurrent units, input size, batch size, epoch, maupun dropout (Reimers & Gurevych, 2017).

Proses pembuatan model pada penelitian ini juga memerlukan *pre-trained global vector for word representation* berbahasa Indonesia yang penulis peroleh dari <https://fasttext.cc/docs/en/crawl-vectors.html> yang memuat 2.000.000 term dengan dimensi sebanyak 50 dimensi.

Pelatihan dataset dilakukan dengan menguji beberapa skenario hyperparameters tuning berikut ini:

Tabel 3.2 Skenario *Hyperparameters Tuning*

| Eksperimen | Jumlah LSTM Layer | Recurrent Units | Dropouts | Batch Size | Max Epoch |
|------------|-------------------|-----------------|----------|------------|-----------|
| #1 | 1 | 128 | - | 64 | 16 |
| #2 | 1 | 128 | - | 64 | 32 |
| #3 | 1 | 128 | - | 128 | 16 |
| #4 | 1 | 128 | - | 128 | 32 |
| #5 | 2 | 32;16 | 0.2 | 64 | 16 |
| #6 | 2 | 32;16 | 0.2 | 64 | 32 |
| #7 | 2 | 32;16 | 0.2 | 128 | 16 |
| #8 | 2 | 32;16 | 0.2 | 128 | 32 |
| #9 | 2 | 64;16 | 0.2 | 64 | 16 |
| #10 | 2 | 256;16 | 0.2 | 64 | 16 |
| #11 | 3 | 32;16;8 | 0.5 | 64 | 16 |
| #12 | 3 | 32;16;8 | 0.5 | 64 | 32 |
| #13 | 3 | 32;16;8 | 0.5 | 128 | 16 |
| #14 | 3 | 32;16;8 | 0.5 | 128 | 32 |

Hasil pelatihan *dataset* yang dilakukan dalam beberapa eksperimen tersebut selanjutnya dievaluasi untuk diketahui tingkat akurasi modelnya. Proses pelatihan dataset ini dilakukan berkali-kali dengan mengubah-ubah parameter-parameter sesuai tabel 1 sehingga diperoleh model dengan tingkat akurasi yang tinggi. Model dengan tingkat akurasi yang tinggi disimpan sebagai model yang akan digunakan untuk proses analisis sentimen berikutnya. Model disimpan dalam bentuk file dengan format HDF5 atau file dengan ekstensi h5.

- c. Proses Pelabelan terhadap Dataset yang belum memiliki Label menggunakan model LSTM

Pada tahapan ini, penulis melakukan proses analisis sentimen terhadap unlabeled dataset yang berupa data tajuk berita yang belum ada label positif, negatif, maupun netral.

Proses analisis sentimen pada tahap ini menggunakan model yang sudah tersimpan sebelumnya, sehingga diperoleh dataset baru yang sudah memiliki label positif, negatif, dan netral. Dataset yang sudah berlabel ini selanjutnya akan dipakai untuk proses pemodelan topik.

- d. Proses Pemodelan Topik

Proses pemodelan topik dilakukan dengan menggunakan metode *Latent Dirichlet Allocation* (LDA). Proses ini dilakukan untuk masing-masing dataset berlabel positif, dataset berlabel negatif, dan dataset berlabel netral. Sehingga diperoleh topik-topik apa

saja yang dibicarakan media massa *online* terkait suatu isu berita, baik itu topik yang berasal dari tajuk berita dengan sentimen positif, negatif, dan netral.

e. Proses Pembuatan Visualisasi

Topik-topik yang diperoleh dari hasil pemodelan topik selanjutnya divisualisasikan dalam bentuk wordcloud (Hidayatullah et al., n.d.) agar lebih mudah untuk melihat *insight* atau wawasan dari topik-topik yang dibicarakan media massa online dan diagram intertopic distance map (Hidayatullah et al., 2018) untuk melihat jarak dan irisan keterkaitan antar topik.

3.2.3 Pengembangan *Prototype* Aplikasi Berbasis Web

Model analisis sentimen yang dihasilkan dari proses pembuatan model akan diterapkan di sebuah aplikasi berbasis web yang dapat melakukan analisis sentimen dan pemodelan topik dari berbagai *keyword* yang diinginkan pengguna aplikasi. Pada penelitian ini, aplikasi berbasis web tersebut masih berupa *prototype* yang dirancang sebagai media simulasi fungsional pemodelan topik pada hasil analisis sentimen terhadap *headline* berita *online* berbahasa Indonesia.

BAB 4

Hasil dan Pembahasan

Berdasarkan Langkah-langkah pada metodologi penelitian, maka pada bagian ini akan dibahas tentang hasil dari tahapan pembuatan model, pengujian model, dan penerapan model pada *prototype* aplikasi berbasis web.

4.1 Hasil dari Tahapan Pembuatan dan Pengujian Model

Tahapan pembuatan dan pengujian model ini terdiri dari 5 sub proses, yaitu: pengumpulan data, pembuatan model LSTM untuk analisis sentimen, pelabelan *unlabeled dataset* menggunakan model LSTM, pemodelan topik, dan visualisasi.


4.1.1 Pengumpulan Data

Penulis menggunakan 2 buah *Virtual Private Server* (VPS) pada penelitian ini. VPS pertama penulis menggunakan VPS layanan dari Amazon Web Service (AWS) dan VPS ini penulis *setup* sebagai server khusus untuk proses *scrapping* situs berita *online* dan proses *sentiment analysis*. VPS kedua penulis menggunakan layanan dari Digital Ocean dan penulis *setup* sebagai *server* penyimpanan *database MySQL* dan *script* pemrograman PHP yang diperlukan agar *prototype* sistem ini dapat berjalan dengan baik.

Sebelum melakukan pengumpulan data, perlu dipersiapkan terlebih dahulu sebuah *database* dan tabel-tabel yang diperlukan untuk sistem ini di *server*.

a. Tabel `'headlines'`

Tabel `'headlines'` ini digunakan untuk menampung data hasil *scrapping* situs-situs berita *online*. Struktur tabel `'headlines'` ditunjukkan pada gambar 4.1.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|--|-------------|-------------------|------------|------|---------|----------|----------------|
| 1 | id  | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | media | varchar(30) | latin1_swedish_ci | | No | None | | |
| 3 | headline_id | text | latin1_swedish_ci | | No | None | | |
| 4 | date | varchar(50) | latin1_swedish_ci | | No | None | | |
| 5 | sentiment_result | varchar(15) | latin1_swedish_ci | | Yes | NULL | | |

Gambar 4.1 Struktur Tabel `'headlines'`

b. Tabel `'master_account'`

Tabel `'master_account'` ini digunakan untuk menampung data pengguna yang melakukan registrasi ke sistem. Struktur tabel `'master_account'` ditunjukkan pada gambar 4.2.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|------------------|-------------|--------------------|------------|------|---------|----------|----------------|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | account_email | varchar(40) | utf8mb4_general_ci | | No | None | | |
| 3 | account_fullname | varchar(70) | utf8mb4_general_ci | | No | None | | |
| 4 | account_password | varchar(20) | utf8mb4_general_ci | | No | None | | |

Gambar 4.2 Struktur Tabel `master_account`

c. Tabel `master_project`

Tabel `master_project` ini digunakan untuk menampung data *project* pemodelan topik terkait *keyword* tertentu yang dibuat oleh pengguna sistem. Struktur tabel `master_project` ditunjukkan pada gambar 4.3.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|----|---------------------------|----------|--------------------|------------|------|---------|---|----------------|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | account_id | int(11) | | | No | None | | |
| 3 | project_name | text | utf8mb4_general_ci | | No | None | | |
| 4 | project_description | text | utf8mb4_general_ci | | No | None | | |
| 5 | project_keywords | text | utf8mb4_general_ci | | No | None | | |
| 6 | project_all_periods | int(1) | | | No | None | 1=All Periods; 0=Use Period | |
| 7 | project_datasets_period_1 | date | | | Yes | NULL | | |
| 8 | project_datasets_period_2 | date | | | Yes | NULL | | |
| 9 | project_datasets | longtext | utf8mb4_general_ci | | Yes | NULL | | |
| 10 | project_date_created | datetime | | | No | None | | |
| 11 | project_date_finished | datetime | | | Yes | NULL | | |
| 12 | project_status | int(1) | | | No | None | -1=FAILED; 0=PENDING; 1=ON PROGRESS; 2=FINISHED | |
| 13 | email_sent | int(1) | | | No | 0 | 0=Email not sent; 1 = Email sent | |

Gambar 4.3 Struktur Tabel `master_project`

d. Tabel `cron`

Tabel `cron` adalah tabel yang berfungsi untuk menyimpan status eksekusi *cron* terkait proses pemodelan topik pada suatu project di server apakah berstatus sudah selesai atau belum. Struktur tabel `cron` ditunjukkan pada gambar 4.4.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|--------|--------|-----------|------------|------|---------|----------|-------|
| 1 | status | int(1) | | | No | None | | |

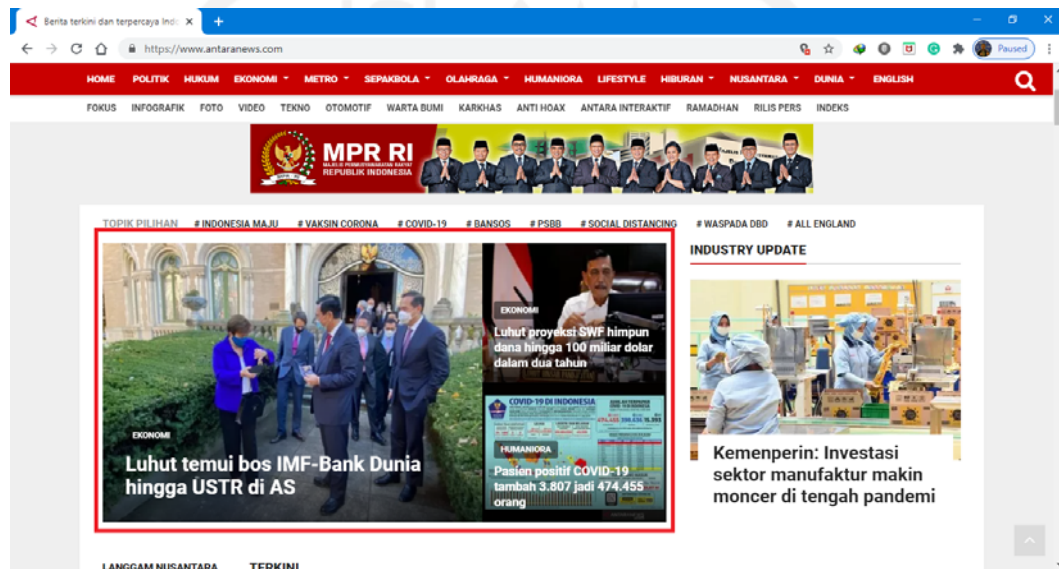
Gambar 4.4 Struktur Tabel `cron`

Setelah database dan tabel-tabelnya selesai dipersiapkan, maka proses pengumpulan data dilakukan dengan cara *scrapping* situs-situs berita *online* berbahasa Indonesia sesuai daftar situs berita yang tercantum pada tabel 3.1. Penulis membuat kode program dengan menggunakan bahasa *Python* dan dengan memanfaatkan *library BeautifulSoup4* untuk mempermudah melakukan *scrapping* situs berita.

Proses *scrapping* situs pada dasarnya membaca kode html dari suatu situs. Pada penelitian ini, *scrapping* situs hanya menasar pada elemen tag div, h1, dan h2 dari situs berita *online*. Karena setiap situs tentunya memiliki nama class yang berbeda-beda untuk area *headline*, khususnya tag div, tag h1, dan tag h2, maka kode python untuk *scrapping* situs dibuat berbeda-beda karena harus menyesuaikan dengan elemen *headline news* pada masing-masing situs berita *online*.

a. *Scrapping* situs berita antaranews.com

Situs berita *online* antaranews.com memiliki area *headline news* seperti yang ditunjukkan pada gambar 4.5.



Gambar 4.5 Situs Berita Online antaranews.com

Apabila situs tersebut dilihat kode sumber htmlnya, maka akan tampak bahwa elemen lokasi *headline news* berada di antara pasangan tag div dengan class *featured-slider* seperti yang ditunjukkan pada gambar 4.6.



Gambar 4.6 Source HTML Situs Berita antaranews.com

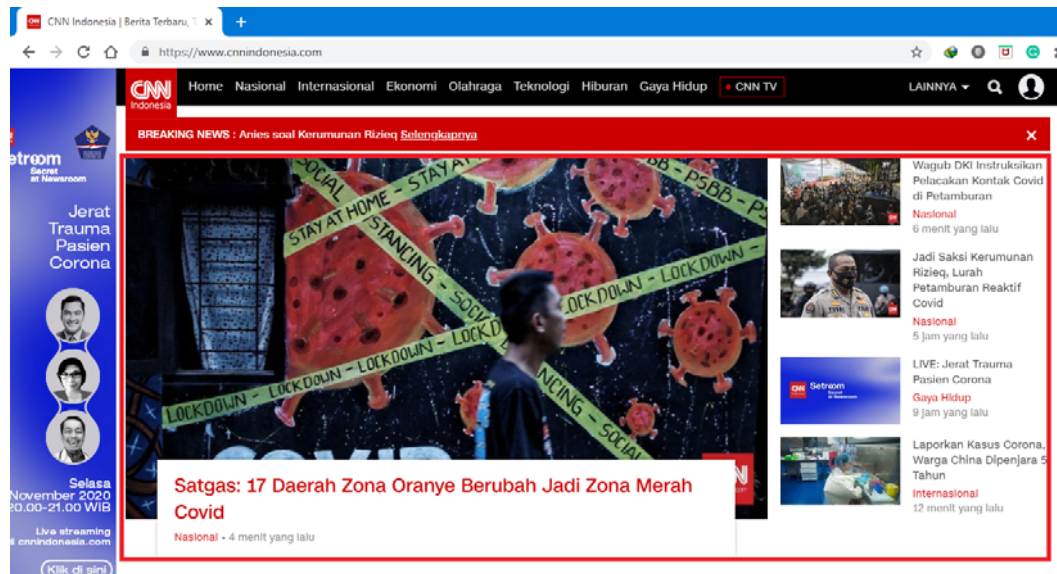
Setelah mengetahui lokasi berita utama ada di elemen div dengan *class featured-slider*, langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet 4.1*.

Code Snippet 4.1. Kode *Python* untuk *Scrapping* situs *antaranews.com*

```
1. '''
2. script python untuk melakukan scrapping situs berita
   antaranews.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'antaranews.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("div", class_='featured-
   slider')
24.     for obj in div_headlines:
25.         headline_titles = obj.find_all('h2')
26.         for h in headline_titles:
27.             headline_title = h.text
28.             dateTimeObj = datetime.now()
29.             headline_date = dateTimeObj
30.
31.             sql = "SELECT * FROM `headlines` WHERE
   `media` = %s AND `headline_id` = %s"
32.             val = (media, headline_title)
33.
34.             cursor.execute(sql, val)
35.             cursor.fetchall()
36.             numRows=cursor.rowcount
37.
38.             if numRows==0:
39.                 sql = "INSERT INTO `headlines`
   (`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
40.                 val = (media, headline_title,
   headline_date)
41.                 cursor.execute(sql, val)
42.                 db.commit()
43.             else:
44.                 print('No data to be inserted!')
45. cursor.close()
46. db.close()
```

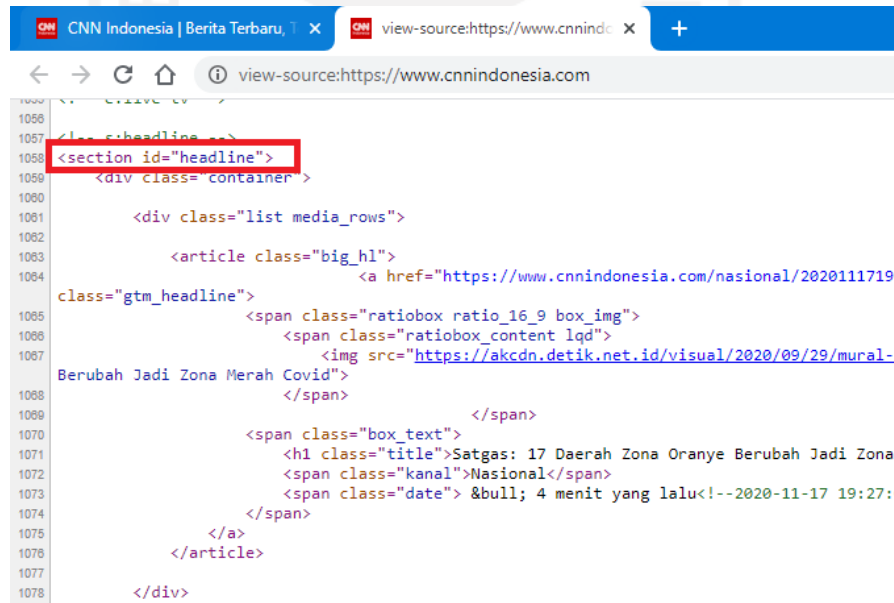

b. *Scrapping* situs berita cnnindonesia.com

Situs berita *online* cnnindonesia.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.7.



Gambar 4.7 Situs Berita Online cnnindonesia.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<section id="headline">` seperti yang ditunjukkan pada gambar 4.8.



Gambar 4.8. *Source* HTML Situs Berita cnnindonesia.com

Setelah mengetahui lokasi berita utama ada di elemen *section* dengan `id=headline`, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet* 4.2.

Code Snippet 4.2 Kode Python untuk Scrapping situs cnnindonesia.com

```
1. '''
2. script python untuk melakukan scrapping situs berita
   cnnindonesia.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'cnnindonesia.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("section", id='headline')
24.
25.     for obj in div_headlines:
26.         headline_titles = obj.find_all('h1', class_='title')
27.         for h in headline_titles:
28.             headline_title = h.text
29.             dateTimeObj = datetime.now()
30.             headline_date = dateTimeObj
31.
32.             sql = "SELECT * FROM `headlines` WHERE `media` =
   %s AND `headline_id` = %s"
33.             val = (media, headline_title)
34.
35.             cursor.execute(sql, val)
36.             cursor.fetchall()
37.             numrows=cursor.rowcount
38.
39.             if numrows==0:
40.                 sql = "INSERT INTO `headlines` (`media`,
   `headline_id`, `date`) VALUES (%s, %s, %s) "
41.                 val = (media, headline_title,
   headline_date)
42.                 cursor.execute(sql, val)
43.                 db.commit()
44.             else:
45.                 print('No data to be inserted!')
46.
47.     for obj in div_headlines:
48.         headline_titles = obj.find_all('h2', class_='title')
49.         for h in headline_titles:
50.             headline_title = h.text
51.             dateTimeObj = datetime.now()
52.             headline_date = dateTimeObj
53.
54.             sql = "SELECT * FROM `headlines` WHERE `media` =
   %s AND `headline_id` = %s"
```

```

55.         val = (media, headline_title)
56.
57.         cursor.execute(sql, val)
58.         cursor.fetchall()
59.         numrows=cursor.rowcount
60.
61.         if numrows==0:
62.             sql = "INSERT INTO `headlines` (`media`,
`headline_id`, `date`) VALUES (%s, %s, %s) "
63.             val = (media, headline_title,
headline_date)
64.             cursor.execute(sql, val)
65.             db.commit()
66.         else:
67.             print('No data to be inserted!')
68.     cursor.close()
69.     db.close()

```

c. *Scrapping* situs berita jpnn.com

Situs berita online jpnn.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.9.



Gambar 4.9 Situs Berita Online jpnn.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="content headline-3">` seperti yang ditunjukkan pada gambar 4.10.

```

35     googletag.defineSlot('/51205855/jp_right_rectangle', [300, 100], 'div-gpt-ad-1582624506229-
0').addService(googletag.pubads());
36     googletag.defineSlot('/51205855/cb_one', [680, 100], 'div-gpt-ad-1600010581786-0').addService(goog
37     googletag.pubads().enableSingleRequest();
38     googletag.pubads().collapseEmptyDivs();
39     googletag.enableServices();
40 });
41 </script></div><div class="clear h10"></div> <!--Top Section--> <section class="home"> <div id="content-ut
class="container"> <!--Leftside--> <div class="kolom-kiri"> <!-- <div class="border-box"> </div> <div clas
</div> --> <div class="border-box"> <div class="content headline-3"> <div id="headline-3" class="owl-carou
class="item" data-thumb="Penyidik Ditreskrimum Polda Metro Jaya saat ini masih melakukan pemeriksaan terha
Jakarta Anies Baswedan"> <a href="https://www.jpnn.com/news/penyidik-periksa-anies-baswedan-sudah-lebih-da
apa-saja" title="Penyidik Periksa Anies Baswedan Sudah Lebih dari Lima Jam, Tanya Apa Saja?">  <
<div class="bg-shadow"></div> <div class="title-teaser"> <h2> <a href="https://www.jpnn.com/news/penyidik-
baswedan-sudah-lebih-dari-lima-jam-tanya-apa-saja" title="Penyidik Periksa Anies Baswedan Sudah Lebih dari

```

Gambar 4.10 Source HTML Situs Berita jpnn.com

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan `class=content headline-3`, maka langkah selanjutnya adalah menerapkannya ke dalam kode `python`. Kode program ditunjukkan pada *Code Snippet 4.3*.

Code Snippet 4.3. Kode `Python` untuk *Scrapping* situs jpnn.com

```

1. '''
2. script python untuk melakukan scrapping situs berita jpnn.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'jpnn.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://' + media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("div", class_='content
headline-3')
24.
25. for obj in div_headlines:
26.     headline_titles = obj.find_all('h2')
27.     for h in headline_titles:
28.         headline_title = h.text
29.         dateTimeObj = datetime.now()
30.         headline_date = dateTimeObj
31.
32.         sql = "SELECT * FROM `headlines` WHERE `media` =
%s AND `headline_id` = %s"

```

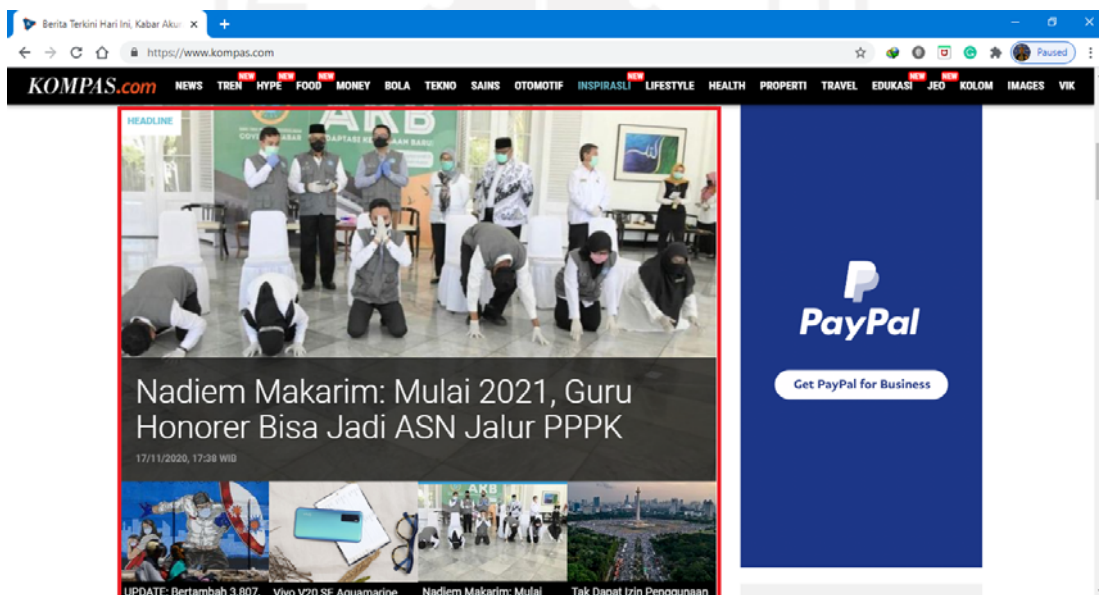
```

33.         val = (media, headline_title)
34.
35.         cursor.execute(sql, val)
36.         cursor.fetchall()
37.         numrows=cursor.rowcount
38.
39.         if numrows==0:
40.             sql = "INSERT INTO `headlines` (`media`,
`headline_id`, `date`) VALUES (%s, %s, %s) "
41.             val = (media, headline_title,
headline_date)
42.             cursor.execute(sql, val)
43.             db.commit()
44.         else:
45.             print('No data to be inserted!')
46.     cursor.close()
47.     db.close()

```

d. *Scrapping* situs berita kompas.com

Situs berita online kompas.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.11.



Gambar 4.11 Situs Berita *Online* kompas.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class=" headline_big_box">` seperti yang ditunjukkan pada gambar 4.12.

Gambar 4.12 Source HTML Situs Berita Kompas.com

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan `class=headline_big_box`, maka langkah selanjutnya adalah menerapkannya ke dalam kode `python`. Kode program ditunjukkan pada *Code Snippet 4.4*.

Code Snippet 4.4. Kode Python untuk Scapping situs Kompas.com

```

1. '''
2. script python untuk melakukan scapping situs berita Kompas.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'kompas.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17. cursor = db.cursor()
18.
19. page = requests.get('https://'+media)
20. soup = BeautifulSoup(page.text, 'html.parser')
21. if page.status_code==200:
22.     div_headlines = soup.find_all("div",
23.     class_='headline_big_box')
24.
25.     for obj in div_headlines:
26.         if obj.find('h1'):
27.
28.             dateTimeObj = datetime.now()
29.             headline_title = obj.find('h1',
30.             class_='headline_big_title').text

```



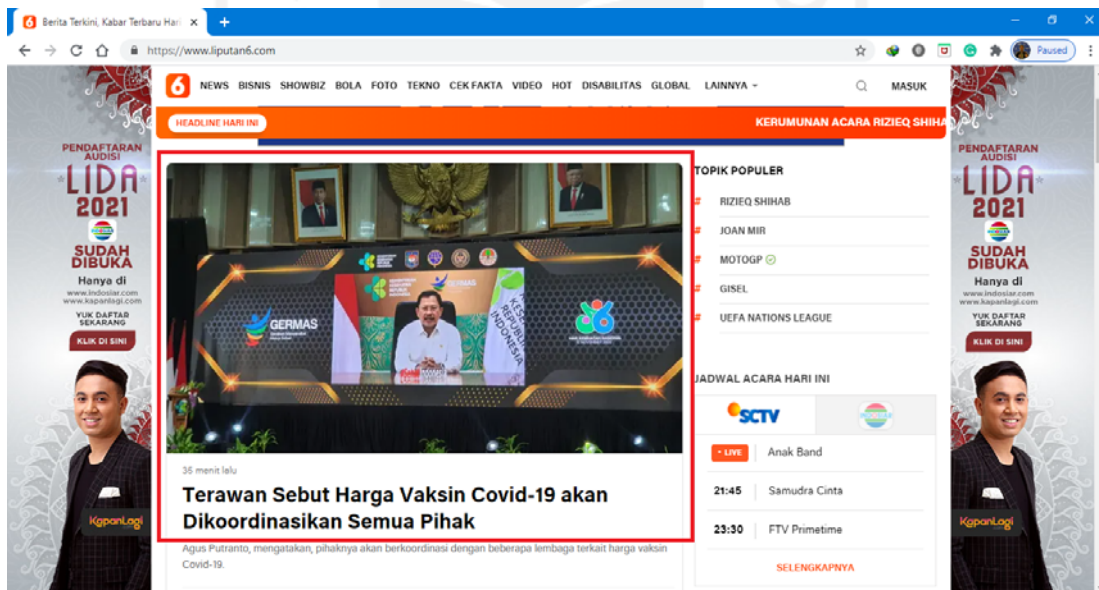
```

29.         headline_date = dateTimeObj
30.
31.         sql = "SELECT * FROM `headlines` WHERE `media` =
    %s AND `headline_id` = %s"
32.         val = (media, headline_title)
33.
34.         cursor.execute(sql, val)
35.         cursor.fetchall()
36.         numrows=cursor.rowcount
37.
38.         if numrows==0:
39.             sql = "INSERT INTO `headlines` (`media`,
    `headline_id`, `date`) VALUES (%s, %s, %s) "
40.             val = (media, headline_title,
    headline_date)
41.             cursor.execute(sql, val)
42.             db.commit()
43.         else:
44.             print('No data to be inserted!')
45.     cursor.close()
46.     db.close()

```

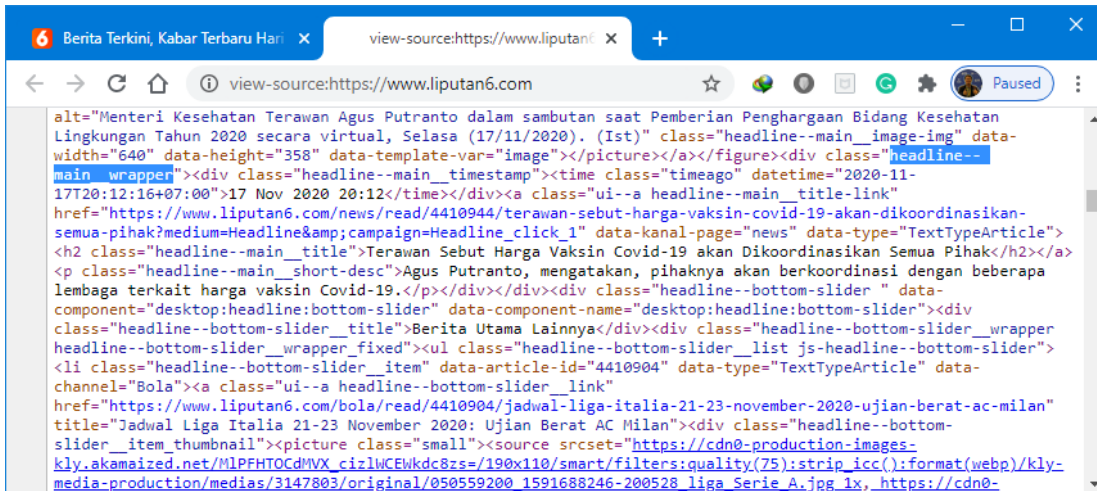
e. *Scrapping* situs berita liputan6.com

Situs berita *online* liputan6.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.13.



Gambar 4.13 Situs Berita *Online* liputan6.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class=" headline--main__wrapper">` seperti yang ditunjukkan pada gambar 4.14.



Gambar 4.14 Source HTML Situs Berita liputan6.com

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan `class=headline--main__wrapper`, maka langkah selanjutnya adalah menerapkannya ke dalam kode `python`. Kode program ditunjukkan pada *Code Snippet 4.5*.

Code Snippet 4.5. Kode `Python` untuk *Scrapping* situs liputan6.com

```

1. '''
2. script python untuk melakukan scrapping situs berita liputan6.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'liputan6.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("div", class_='headline--
main__wrapper')
24.
25. for obj in div_headlines:
26.     if obj.find('h2'):
27.         dateTimeObj = datetime.now()
28.         headline_title = obj.find('h2',
class_='headline--main__title').text
29.         headline_date = dateTimeObj
30.
31.         sql = "SELECT * FROM `headlines` WHERE `media` =
%s AND `headline_id` = %s"

```



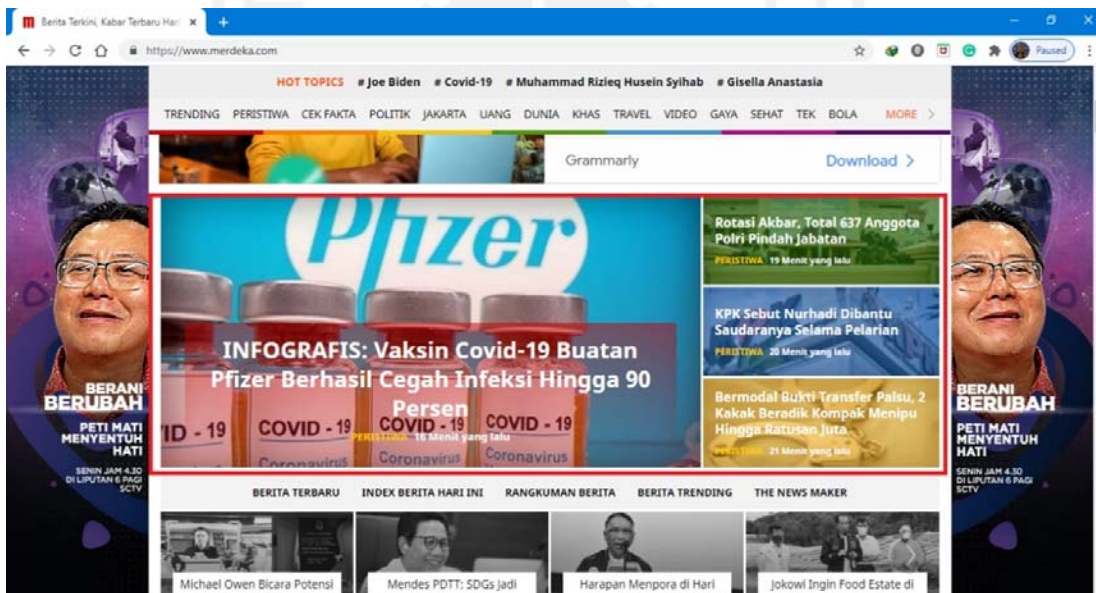
```

32.         val = (media, headline_title)
33.
34.         cursor.execute(sql, val)
35.         cursor.fetchall()
36.         numrows=cursor.rowcount
37.
38.         if numrows==0:
39.             sql = "INSERT INTO `headlines` (`media`,
`headline_id`, `date`) VALUES (%s, %s, %s) "
40.             val = (media, headline_title,
headline_date)
41.             cursor.execute(sql, val)
42.             db.commit()
43.         else:
44.             print('No data to be inserted!')
45.     cursor.close()
46.     db.close()

```

f. *Scrapping* situs berita merdeka.com

Situs berita online merdeka.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.15.



Gambar 4.15 Situs Berita *Online* merdeka.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="mdk-headline-left">` dan `<div class="mdk-headline-right">` seperti yang ditunjukkan pada gambar 4.16.

```
352 <div class="mdk-headline-left">
353 <!--
354 <a href="/peristiwa/infografis-vaksin-covid-19-buatan-pfizer-berhasil-cegah-infeksi-hingga-90-persen.html"
class="deskrip-h1 bg-red" style="background-
image:url(https://cdns.kling.com/merdeka.com/i/w/news/2020/11/17/1243633/640x320/infografis-vaksin-covid-19-
buatan-pfizer-berhasil-cegah-infeksi-hingga-90-persen.jpg)">
355 <h1 class="title-h">INFOGRAFIS: Vaksin Covid-19 Buatan Pfizer Berhasil Cegah Infeksi Hingga 90
Persen</h1>
356 <div class="kategori-time">
357 <span class="link-kh">PERISTIWA</span>
358 <span class="time-h"> 16 Menit yang lalu</span>
359 </div>
360 </a-->
361 <a href="/peristiwa/infografis-vaksin-covid-19-buatan-pfizer-berhasil-cegah-infeksi-hingga-90-persen.html"
class="deskrip-h1 bg-red" style="background-
image:url(https://cdns.kling.com/merdeka.com/i/w/news/2020/11/17/1243633/640x320/infografis-vaksin-covid-19-
buatan-pfizer-berhasil-cegah-infeksi-hingga-90-persen.jpg)">
362 <div class="headline-red">
363 <h1 class="title-h">INFOGRAFIS: Vaksin Covid-19 Buatan Pfizer Berhasil Cegah Infeksi Hingga 90
Persen</h1>
364 <div class="kategori-time">
365 <span class="link-kh">PERISTIWA</span>
366 <span class="time-h"> 16 Menit yang lalu</span>
367 </div>
368 </div>
369 </a>
370
371 </div>
372 <div class="mdk-headline-right">
373 <div class="mdk-item">
```

Gambar 4.16 Source HTML Situs Berita merdeka.com

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan *class= mdk-headline-left* dan *mdk-headline-right*, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet 4.6*.

Code Snippet 4.6. Kode *Python* untuk *Scraping* situs merdeka.com

```
1. '''
2. script python untuk melakukan scrapping situs berita merdeka.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'merdeka.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("div", class_='mdk-
headline-left')
24.     for obj in div_headlines:
25.         headline_titles = obj.find_all('h1',
class_='title-h')
```

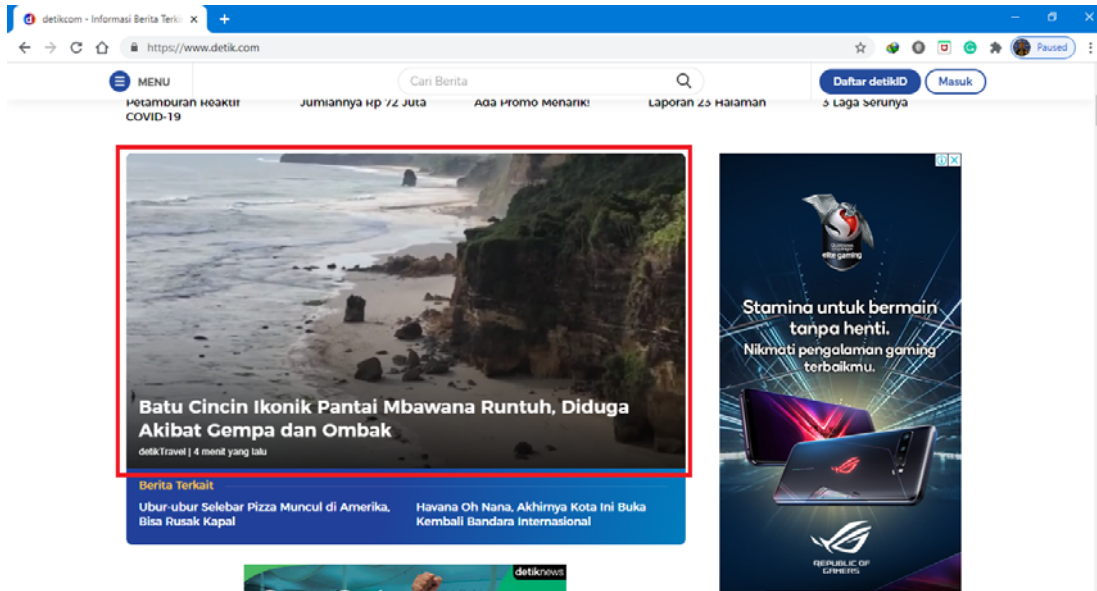
```

26.         for h in headline_titles:
27.             headline_title = h.text
28.             dateTimeObj = datetime.now()
29.             headline_date = dateTimeObj
30.
31.             sql = "SELECT * FROM `headlines` WHERE
`media` = %s AND `headline_id` = %s"
32.             val = (media, headline_title)
33.
34.             cursor.execute(sql, val)
35.             cursor.fetchall()
36.             numrows=cursor.rowcount
37.
38.             if numrows==0:
39.                 sql = "INSERT INTO `headlines`
(`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
40.                 val = (media, headline_title,
headline_date)
41.                 cursor.execute(sql, val)
42.                 db.commit()
43.             else:
44.                 print('No data to be inserted!')
45.
46.         div_headlines = soup.find_all("div", class_='mdk-
headline-right')
47.         for obj in div_headlines:
48.             headline_titles = obj.find_all('h1',
class_='title-h')
49.             for h in headline_titles:
50.                 headline_title = h.text
51.                 dateTimeObj = datetime.now()
52.                 headline_date = dateTimeObj
53.
54.                 sql = "SELECT * FROM `headlines` WHERE
`media` = %s AND `headline_id` = %s"
55.                 val = (media, headline_title)
56.
57.                 cursor.execute(sql, val)
58.                 cursor.fetchall()
59.                 numrows=cursor.rowcount
60.
61.                 if numrows==0:
62.                     sql = "INSERT INTO `headlines`
(`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
63.                     val = (media, headline_title,
headline_date)
64.                     cursor.execute(sql, val)
65.                     db.commit()
66.                 else:
67.                     print('No data to be inserted!')
68.     cursor.close()
69.     db.close()

```

g. *Scrapping* situs berita detik.com

Situs berita online detik.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.17.



Gambar 4.17 Situs Berita *Online* detik.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="headline">` seperti yang ditunjukkan pada gambar 4.18.

```

97 <div class="headline">
98 <article class="media media--text-overlay block-link">
99 <div class="media_image">
100 <a onclick='_pt(this, "headline", "Fakta Baru Brigadir AN Ngamuk di BNN Bone Ternyata Positif Sabu
href="https://news.detik.com/berita/d-5259200/fakta-baru-brigadir-an-ngamuk-di-bnn-bone-ternyata-positif-sabu?
class="media_link">
101 <span class="ratiobox ratiobox--16-9 lqd">
102  </span>
103 </div>
104 <div class="media_text">
105 <div class="media_subtitle">Round-Up</div>
106 <h2 class="media_title">
107 <a onclick='_pt(this, "headline", "Fakta Baru Brigadir AN Ngamuk di BNN Bone Ternyata Positif
href="https://news.detik.com/berita/d-5259200/fakta-baru-brigadir-an-ngamuk-di-bnn-bone-ternyata-positif-sabu?
class="media_link">
108 Fakta Baru Brigadir AN Ngamuk di BNN Bone Ternyata Positif Sabu </a>
109 </h2>
110 <div class="media_date mgt-4"><span d-time="1605623438" title="Selasa, 17 Nov 2020 21:30 WIB">12
</div>
111 </div>
112 </article>
  
```

Gambar 4.18 Source HTML Situs Berita detik.com

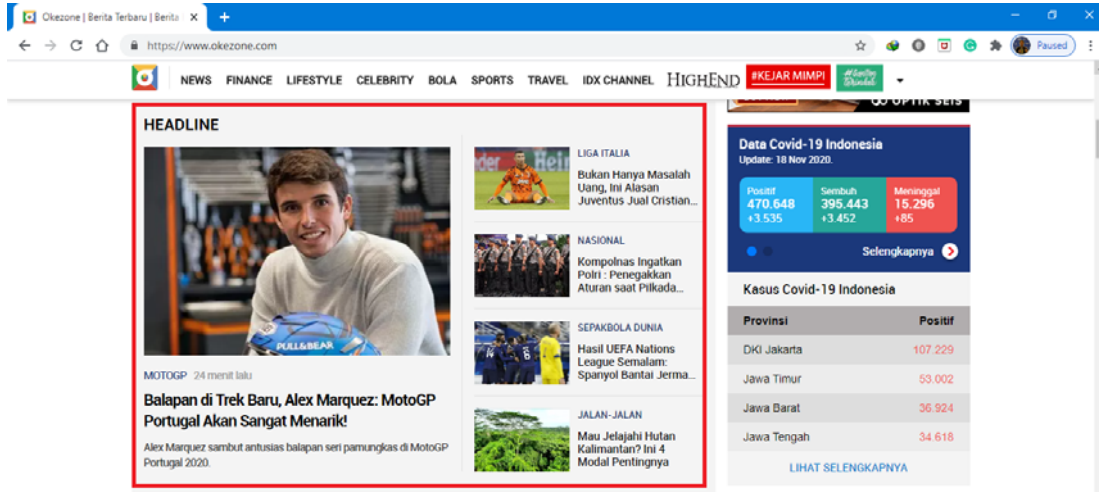
Setelah mengetahui lokasi berita utama ada di elemen tag div dengan *class=headline*, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet 4.7*.

Code Snippet 4.7. Kode Python untuk Scrapping situs detik.com

```
1. '''
2. script python untuk melakukan scrapping situs berita detik.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'detik.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("article", class_='media
24.     media--text-overlay block-link')
25.
26.     i = 1
27.
28.     for obj in div_headlines:
29.         if obj.find('h2', class_='media__title'):
30.             dateTimeObj = datetime.now()
31.
32.             headline_title = (obj.find('h2',
33.             class_='media__title').text).replace("
34.             ").strip()
35.
36.             headline_date = dateTimeObj
37.
38.             sql = "SELECT * FROM `headlines` WHERE `media` =
39.             %s AND `headline_id` = %s"
40.             val = (media, headline_title)
41.
42.             cursor.execute(sql, val)
43.             cursor.fetchall()
44.             numrows=cursor.rowcount
45.
46.             if numrows==0:
47.                 sql = "INSERT INTO `headlines` (`media`,
48.                 `headline_id`, `date`) VALUES (%s, %s, %s) "
49.                 val = (media, headline_title,
50.                 headline_date)
51.
52.                 cursor.execute(sql, val)
53.                 db.commit()
54.             else:
55.                 print('No data to be inserted!')
56.
57.     cursor.close()
58.     db.close()
```

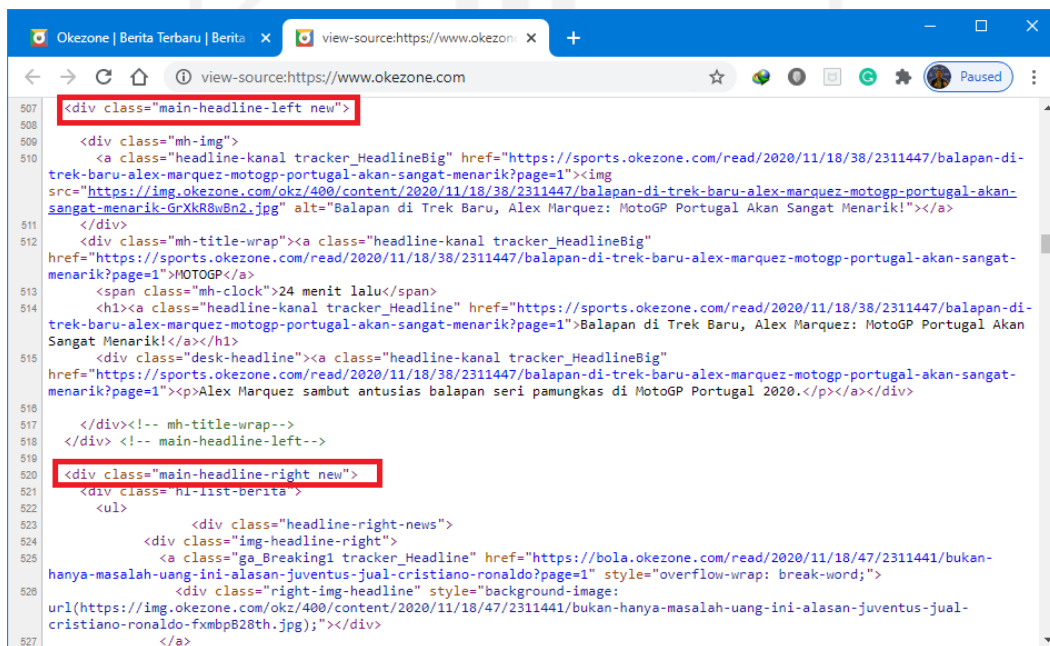
h. *Scrapping* situs berita okezone.com

Situs berita *online* okezone.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.19.



Gambar 4.19 Situs Berita *Online* okezone.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="main-headline-left">` dan `<div class="main-headline-right">` seperti yang ditunjukkan pada gambar 4.20.



Gambar 4.20 *Source* HTML Situs Berita okezone.com

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan *class= main-headline-left* dan *class=main-headline-right*, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet 4.8*.

Code Snippet 4.8. Kode *Python* untuk *Scrapping* situs okezone.com

```
1. '''
2. script python untuk melakukan scrapping situs berita okezone.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'okezone.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19. page = requests.get('https://'+media)
20. soup = BeautifulSoup(page.text, 'html.parser')
21. if page.status_code==200:
22.     div_headlines = soup.find_all("div", class_='main-
23. headline-left new')
24.     for obj in div_headlines:
25.         headline_titles = obj.find_all('h1')
26.         for h in headline_titles:
27.             headline_title = h.text
28.             dateTimeObj = datetime.now()
29.             headline_date = dateTimeObj
30.
31.             sql = "SELECT * FROM `headlines` WHERE
32. `media` = %s AND `headline_id` = %s"
33.             val = (media, headline_title)
34.
35.             cursor.execute(sql, val)
36.             cursor.fetchall()
37.             numrows=cursor.rowcount
38.
39.             if numrows==0:
40.                 sql = "INSERT INTO `headlines`
41. (`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
42.                 val = (media, headline_title,
43. headline_date)
44.                 cursor.execute(sql, val)
45.                 db.commit()
46.             else:
47.                 print('No data to be inserted!')
```

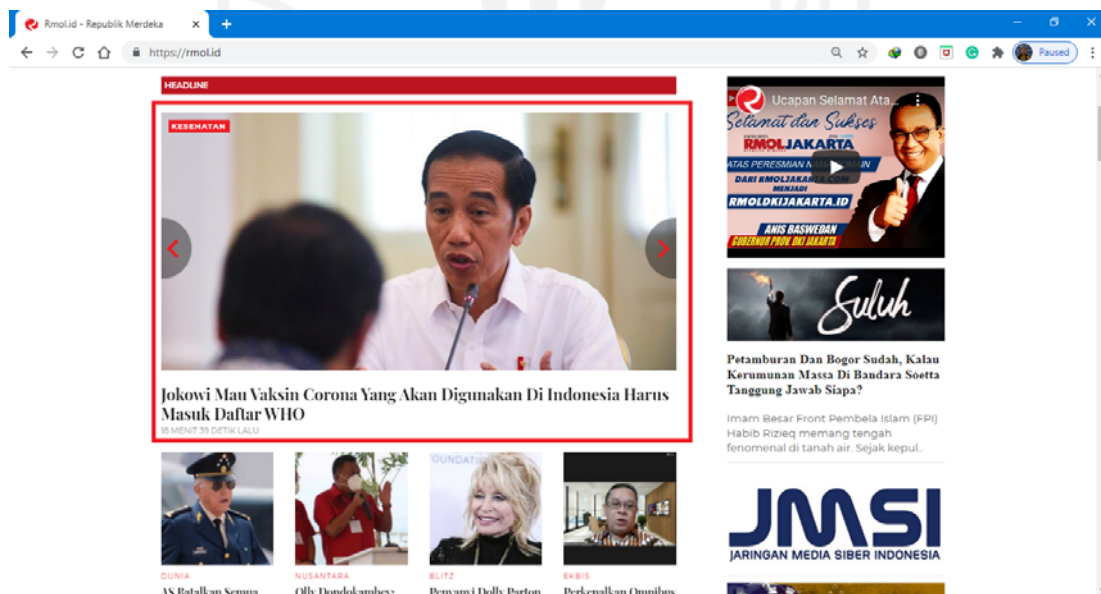
```

47.         headline_titles = obj.find_all('div',
48.         class_='jdl-right-headline')
49.         for h in headline_titles:
50.             headline_title = h.text
51.             dateTimeObj = datetime.now()
52.             headline_date = dateTimeObj
53.             sql = "SELECT * FROM `headlines` WHERE
54.             `media` = %s AND `headline_id` = %s"
55.             val = (media, headline_title)
56.             cursor.execute(sql, val)
57.             cursor.fetchall()
58.             numRows=cursor.rowcount
59.
60.             if numRows==0:
61.                 sql = "INSERT INTO `headlines`
62.                 (`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
63.                 val = (media, headline_title,
64.                 headline_date)
65.                 cursor.execute(sql, val)
66.                 db.commit()
67.             else:
68.                 print('No data to be inserted!')
69.         cursor.close()
70.         db.close()

```

i. *Scrapping* situs berita rmol.id

Situs berita *online* rmol.id memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.21.



Gambar 4.21 Situs Berita *Online* rmol.id

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="home_headline_wrap">` seperti yang ditunjukkan pada gambar 4.22.


```

265 <div class="container margin-top-20">
266 <div class="row">
267 <div class="col-md-8 col-sm-8">
268 <!-- Headline -->
269 <div class="home_headline_wrap">
270 <div class="layout_1 margin-bottom-0">
271 <div class="row">
272 <div class="col-md-12">
273 <h3 class="heading-1"><span>Headline</span></h3>
274 <div class="post-carousel-wrap">
275 <div class="post-carousel">
276 <div class="layout_1--
item">
277 <a href="https://politik.rmol.id/read/2020/11/18/461725/puan-
maharani-selamat-hut-ke-108-muhammadiyah-terus-konsisten-perjuangkan-islam-berkemajuan">
278 <span class="badge text-uppercase badge-overlay badge-
lifestyle">Politik</span>
279 <div class="overlayx"></div>
280 

```

Gambar 4.22 Source HTML Situs Berita rmol.id

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan `class=home_headline_wrap`, maka langkah selanjutnya adalah menerapkannya ke dalam kode `python`. Kode program ditunjukkan pada *Code Snippet 4.9*.

Code Snippet 4.9. Kode `Python` untuk *Scrapping* situs rmol.id

```

1. '''
2. script python untuk melakukan scrapping situs berita rmol.id
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'rmol.id'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://' + media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code == 200:
23.     div_headlines = soup.find_all("div",
class_='home_headline_wrap')
24.
25. for obj in div_headlines:
26.     headline_titles = obj.find_all('h4')
27.     for h in headline_titles:
28.         headline_title = h.text
29.         dateTimeObj = datetime.now()
30.         headline_date = dateTimeObj

```

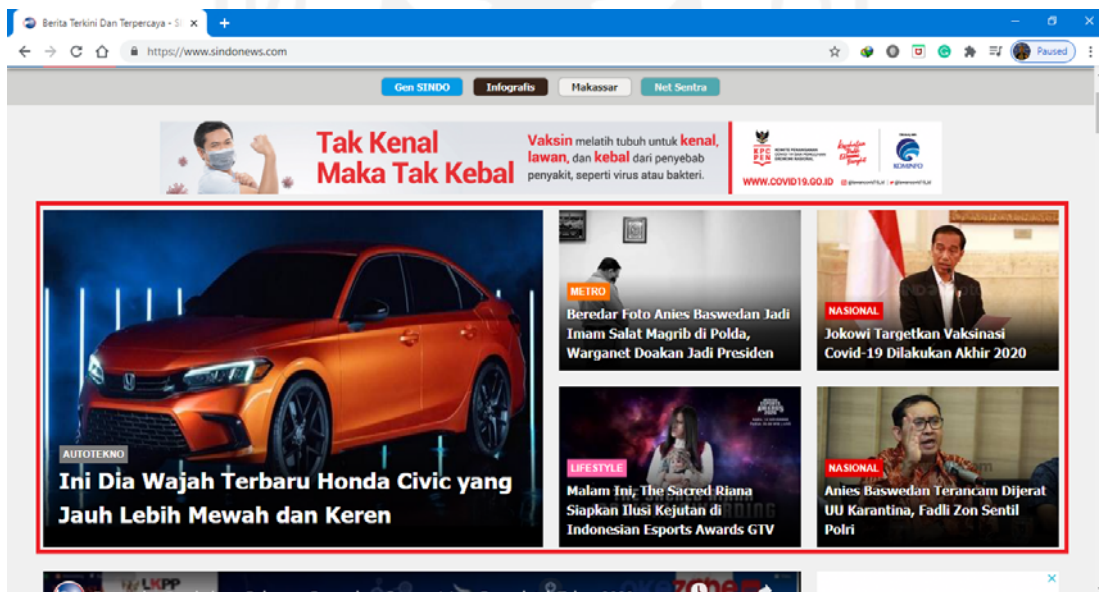
```

31.
32.         sql = "SELECT * FROM `headlines` WHERE `media` =
    %s AND `headline_id` = %s"
33.         val = (media, headline_title)
34.
35.         cursor.execute(sql, val)
36.         cursor.fetchall()
37.         numRows=cursor.rowcount
38.
39.         if numRows==0:
40.             sql = "INSERT INTO `headlines` (`media`,
    `headline_id`, `date`) VALUES (%s, %s, %s) "
41.             val = (media, headline_title,
    headline_date)
42.             cursor.execute(sql, val)
43.             db.commit()
44.         else:
45.             print('No data to be inserted!')
46.     cursor.close()
47.     db.close()

```

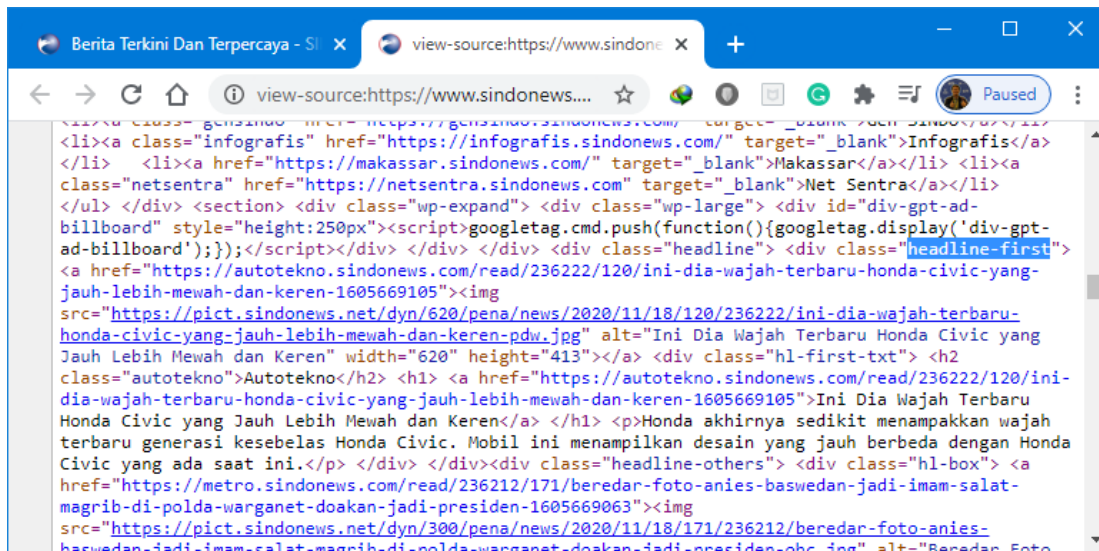
j. *Scrapping* situs berita sindonews.com

Situs berita *online* sindonews.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.23.



Gambar 4.23 Situs Berita *Online* sindonews.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="headline-first">` dan `<div class="headline-others">` seperti yang ditunjukkan pada gambar 4.24.



Gambar 4.24 Source HTML Situs Berita sindonews.com

Setelah mengetahui lokasi berita utama ada di elemen tag div dengan *class=headline_first* dan *headline_others*, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet 4.10*.

Code Snippet 4.10. Kode *Python* untuk *Scraping* situs sindonews.com

```

1. '''
2. script python untuk melakukan scrapping situs berita
   sindonews.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'sindonews.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("div", class_='headline-
   first')
24.     for obj in div_headlines:
25.         headline_titles = obj.find_all('h1')
26.         for h in headline_titles:
27.             headline_title = h.text
28.             dateTimeObj = datetime.now()
29.             headline_date = dateTimeObj
30.

```

```

31.         sql = "SELECT * FROM `headlines` WHERE
`media` = %s AND `headline_id` = %s"
32.         val = (media, headline_title)
33.
34.         cursor.execute(sql, val)
35.         cursor.fetchall()
36.         numrows=cursor.rowcount
37.
38.         if numrows==0:
39.             sql = "INSERT INTO `headlines`
(`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
40.             val = (media, headline_title,
headline_date)
41.             cursor.execute(sql, val)
42.             db.commit()
43.         else:
44.             print('No data to be inserted!')
45.
46.     div_headlines = soup.find_all("div", class_='headline-
others')
47.     for obj in div_headlines:
48.         headline_titles = obj.find_all('div',
class_='hl-title')
49.         for h in headline_titles:
50.             headline_title = h.text
51.             dateTimeObj = datetime.now()
52.             headline_date = dateTimeObj
53.
54.             sql = "SELECT * FROM `headlines` WHERE
`media` = %s AND `headline_id` = %s"
55.             val = (media, headline_title)
56.
57.             cursor.execute(sql, val)
58.             cursor.fetchall()
59.             numrows=cursor.rowcount
60.
61.             if numrows==0:
62.                 sql = "INSERT INTO `headlines`
(`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
63.                 val = (media, headline_title,
headline_date)
64.                 cursor.execute(sql, val)
65.                 db.commit()
66.             else:
67.                 print('No data to be inserted!')
68.     cursor.close()
69.     db.close()

```

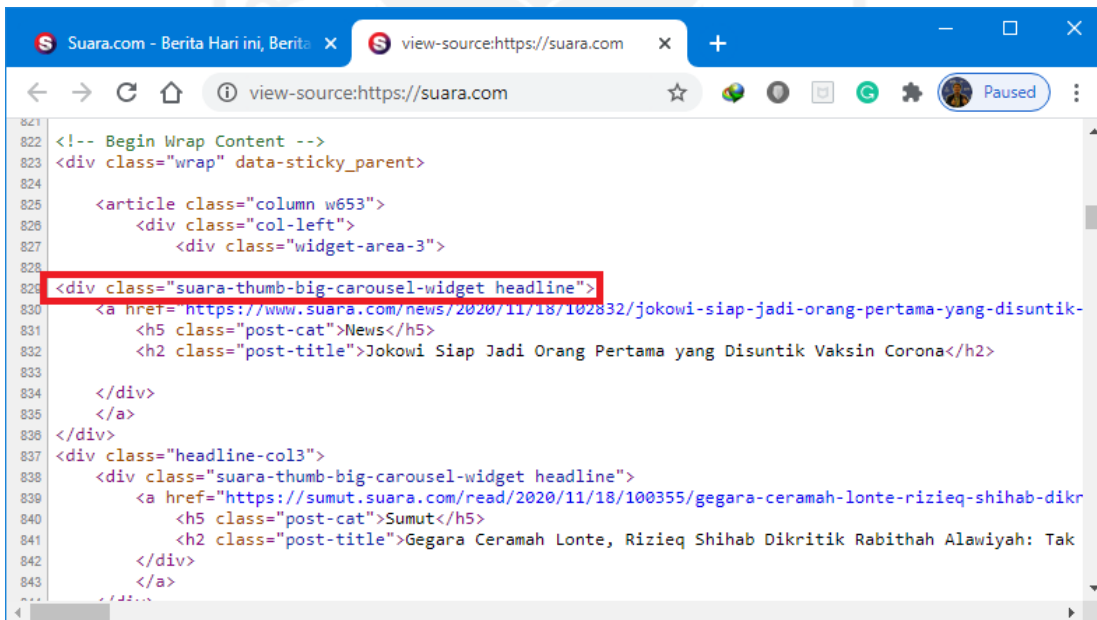
k. *Scrapping* situs berita suara.com

Situs berita *online* suara.com memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.25.



Gambar 4.25 Situs Berita *Online* suara.com

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<div class="suara-thumb-big-carousel-widget headline">` seperti yang ditunjukkan pada gambar 4.26.



Gambar 4.26 *Source* HTML Situs Berita suara.com

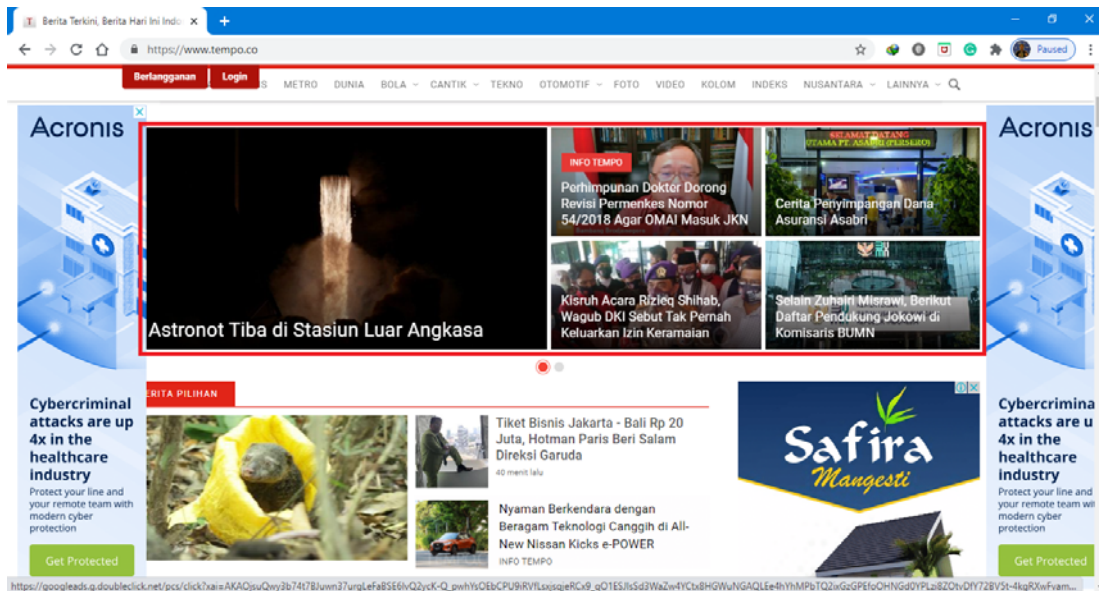
Setelah mengetahui lokasi berita utama ada di elemen tag `div` dengan atribut `class=suara-thumb-big-carousel-widget headline`, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*. Kode program ditunjukkan pada *Code Snippet* 4.11.

Code Snippet 4.11. Kode Python untuk Scrapping situs suara.com

```
1. '''
2. script python untuk melakukan scrapping situs berita suara.com
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'suara.com'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://'+media)
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("div", class_='suara-
thumb-big-carousel-widget headline')
24.
25.     for obj in div_headlines:
26.         headline_titles = obj.find_all('h2',
class_='post-title')
27.         for h in headline_titles:
28.             headline_title = h.text
29.             dateTimeObj = datetime.now()
30.             headline_date = dateTimeObj
31.
32.             sql = "SELECT * FROM `headlines` WHERE
`media` = %s AND `headline_id` = %s"
33.             val = (media, headline_title)
34.
35.             cursor.execute(sql, val)
36.             cursor.fetchall()
37.             numRows=cursor.rowcount
38.
39.             if numRows==0:
40.                 sql = "INSERT INTO `headlines`
(`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
41.                 val = (media, headline_title,
headline_date)
42.                 cursor.execute(sql, val)
43.                 db.commit()
44.                 print('This data has been
inserted: ')
45.                 print(headline_title)
46.                 print(headline_date)
47.             else:
48.                 print('No data to be inserted!')
49. cursor.close()
50. db.close()
```

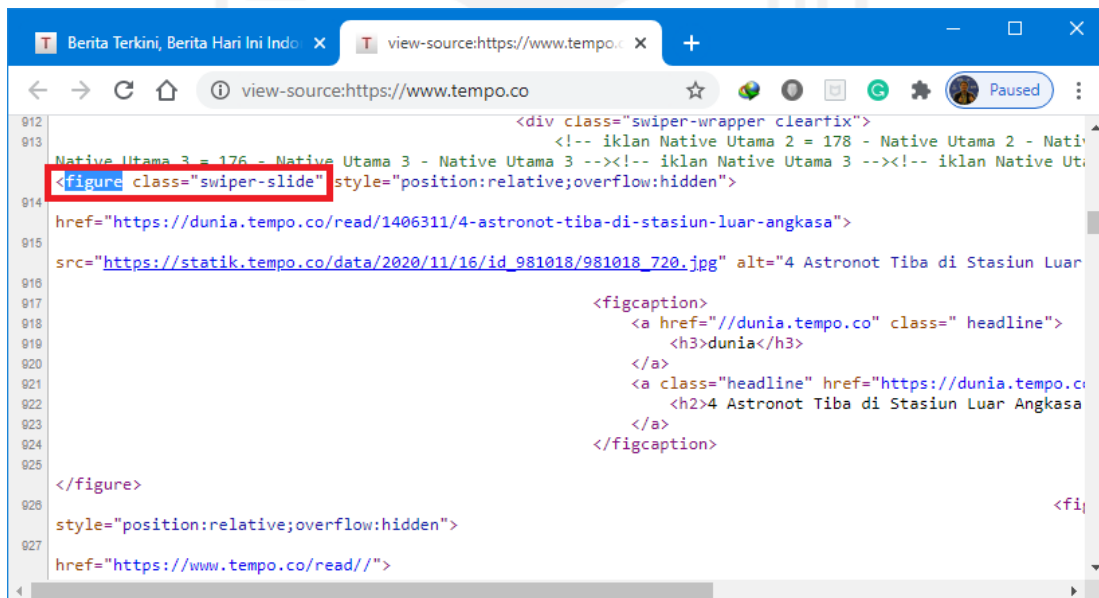

1. *Scrapping* situs berita tempo.com

Situs berita *online* tempo.co memiliki area *headline news* pada lokasi yang ditunjukkan pada gambar 4.27.



Gambar 4.27 Situs Berita *Online* tempo.co

Dengan melihat kode sumber htmlnya, diketahui lokasi *headline news* berada di elemen tag `<figure class="swiper-slide">` seperti yang ditunjukkan pada gambar 4.28.



Gambar 4.28 *Source* HTML Situs Berita tempo.co

Setelah mengetahui lokasi berita utama ada di elemen tag figure dengan atribut `class=swiper-slide`, maka langkah selanjutnya adalah menerapkannya ke dalam kode *python*.

Code Snippet 4.12. Kode Python untuk Scrapping situs tempo.co

```
1. '''
2. script python untuk melakukan scrapping situs berita tempo.co
3. '''
4. from bs4 import BeautifulSoup
5. from datetime import datetime
6. import requests
7. import mysql.connector
8.
9. media = 'tempo.co'
10.
11. db = mysql.connector.connect(
12.     host="topsenaindonesia.com",
13.     user="naury",
14.     passwd="T0ps3na",
15.     database="thesis"
16. )
17.
18. cursor = db.cursor()
19.
20. page = requests.get('https://tempo.co')
21. soup = BeautifulSoup(page.text, 'html.parser')
22. if page.status_code==200:
23.     div_headlines = soup.find_all("figure", class_='swiper-
    slide')
24.     for obj in div_headlines:
25.         headline_titles = obj.find_all('h2')
26.         for h in headline_titles:
27.             headline_title = h.text
28.             dateTimeObj = datetime.now()
29.             headline_date = dateTimeObj
30.
31.             sql = "SELECT * FROM `headlines` WHERE
    `media` = %s AND `headline_id` = %s"
32.             val = (media, headline_title)
33.
34.             cursor.execute(sql, val)
35.             cursor.fetchall()
36.             numRows=cursor.rowcount
37.
38.             if numRows==0:
39.                 sql = "INSERT INTO `headlines`
    (`media`, `headline_id`, `date`) VALUES (%s, %s, %s) "
40.                 val = (media, headline_title,
    headline_date)
41.                 cursor.execute(sql, val)
42.                 db.commit()
43.             else:
44.                 print('No data to be inserted!')
45. cursor.close()
46. db.close()
```

Script-script *python* tersebut selanjutnya dimasukkan ke dalam *scheduled cron jobs* menggunakan fitur *cron* pada server. Untuk keperluan membuat *scheduled cron jobs*, penulis memanfaatkan *bash script* sebagai file pemicu.

Code Snippet 4.13 Bash script untuk mengeksekusi perintah *scrapping* situs

```
1. #!/bin/bash
2.
3. echo "Scrapping Situs antaranews.com"
4. python3 /home/topsena/scrapping/antaranews.py
5.
6. echo "Scrapping Situs cnnindonesia.com"
7. python3 /home/topsena/scrapping/cnnindonesia.py
8.
9. echo "Scrapping Situs jpnn.com"
10. python3 /home/topsena/scrapping/jpnn.py
11.
12. echo "Scrapping Situs kompas.com"
13. python3 /home/topsena/scrapping/kompas.py
14.
15. echo "Scrapping Situs liputan6.com"
16. python3 /home/topsena/scrapping/liputan6.py
17.
18. echo "Scrapping Situs merdeka.com"
19. python3 /home/topsena/scrapping/merdeka.py
20.
21. echo "Scrapping Situs detik.com"
22. python3 /home/topsena/scrapping/detik.py
23.
24. echo "Scrapping Situs okezone.com"
25. python3 /home/topsena/scrapping/okezone.py
26.
27. echo "Scrapping Situs rmol.id"
28. python3 /home/topsena/scrapping/rmol.py
29.
30. echo "Scrapping Situs sindonews.com"
31. python3 /home/topsena/scrapping/sidonews.py
32.
33. echo "Scrapping Situs suara.com"
34. python3 /home/topsena/scrapping/suara.py
35.
36. echo "Scrapping Situs tempo.co"
37. python3 /home/topsena/scrapping/tempo.py
```

Selanjutnya file bash script tersebut dimasukkan ke dalam *list* crontab dengan pengaturan eksekusi setiap menit melalui perintah *crontab -e* pada terminal dan menambahkan baris `* * * * * /bin/sh /home/topsena/scrapping/site_scrapping.sh`. Gambar 4.29 menunjukkan penambahan *scheduled cron jobs*.

```

topsena.tlp - topsena@topsenaIndonesia.com:22 - Bitwise xterm - topsena@naury: ~/scrapping
GNU nano 2.9.3 /tmp/crontab.DiGkSM/crontab
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * * /bin/sh /home/topsena/scrapping/site_scrapping.sh

```

Gambar 4.29 Menambahkan *Scheduled Cron Jobs*

Dengan menambahkan *entry* pada *scheduled cron jobs*, maka bash script yang sudah dibuat sebelumnya akan selalu dieksekusi setiap menit, artinya proses *scrapping* situs-situs berita *online* dilakukan setiap menit dan setiap ada headline berita baru maka akan tersimpan ke *database* di *server VPS*. Dengan menggunakan *scheduled cron jobs*, maka data yang disimpan ke *database* bersifat *streaming*, selalu mengalir dan tersimpan setiap ada *headline* berita baru.

Data yang diperoleh dari *scrapping* situs-situs berita *online* berbahasa Indonesia tersimpan di tabel `'headlines'`. Gambar 4.30 menunjukkan sebagian data *headline* berita yang sudah tersimpan di *database* sebagai hasil dari proses *scrapping* situs.

| id | media | headline_id | date | sentiment_result |
|----|----------|--|---------------------------|------------------|
| 10 | tempo.co | Wahyu Setiawan Mengundurkan Diri sebagai Komisioner... | 2020-01-10 19:11:08.95205 | |
| 11 | tempo.co | Utara Kemayoran Masih Ditutup, Pengelola Kemayoran ... | 2020-01-10 19:11:08.95441 | |
| 13 | tempo.co | BEI Temukan Transaksi Saham Gorengan Rp 755 Miliar... | 2020-01-10 19:11:08.95681 | |
| 14 | tempo.co | Prancis Investigasi Jatuhnya Pesawat Ukraine Inter... | 2020-01-10 19:11:08.95792 | |
| 15 | tempo.co | KPU Sebut Permohonan PAW Harun Masiku Diteken Mega... | 2020-01-10 19:11:08.95882 | |
| 16 | tempo.co | Awal 2020, Modal Asing Masuk RI Sudah Rp 10 Triliu... | 2020-01-10 19:11:08.96011 | |
| 17 | tempo.co | Berkas Sudah P21, Kasus Sabu AKBP Benny Alamsyah S... | 2020-01-10 19:11:08.96129 | |
| 18 | tempo.co | Penyidik KPK Dapat Izin Pengegeledahan Kasus Suap W... | 2020-01-10 19:11:08.96233 | |
| 19 | tempo.co | Mani Erika Pangestu Direktur Bank Dunia, Bos BI Uca... | 2020-01-10 19:11:08.96342 | |

Gambar 4.30 Data Headline Berita Hasil Scrapping Situs Berita Online

Setelah data terkumpul banyak, sebagai informasi pada saat penelitian ini dilaksanakan, diperoleh data secara *streaming* sebanyak 201220 record, selanjutnya dipilih

2500 *record* untuk dijadikan dataset latihan dengan menggunakan query “CREATE TABLE dataset_latih SELECT * FROM headlines LIMIT 2500”. Tabel baru yang bernama `dataset_latih` tersebut selanjutnya dikonversi menjadi file CSV dan selanjutnya dilakukan proses pelabelan secara manual menggunakan *software spreadsheet*. Proses pelabelan dataset secara manual ini dilakukan sendiri oleh penulis dalam beberapa sesi waktu yang berbeda dan tentunya subyektifitas penulis terhadap sentimen suatu *headline* akan mendominasi. Secara umum pelabelan suatu headline apakah termasuk bersentimen positif, negatif, dan netral dengan mempertimbangkan makna yang dirasakan dari suatu kalimat. Kalimat dengan makna positif secara umum masuk ke dalam klasifikasi sentimen positif, sebaliknya kalimat dengan makna negatif secara umum masuk ke dalam klasifikasi sentimen negatif, sedangkan kalimat yang tidak termasuk kedua kelompok tersebut akan diklasifikasikan ke dalam sentimen netral. Gambar 4.31 menunjukkan proses pelabelan dataset latihan menggunakan *software spreadsheet* Microsoft Excel.

| | A | B | C |
|------|---|----------|---|
| 2486 | Wabah Virus Corona, AS Rilis Peringatan Perjalanan ke China | Neutral | |
| 2487 | Wabah Virus Corona, China Kirimkan Tambahan 2.500 Pekerja Medis ke Wuhan | Positive | |
| 2488 | Wabah Virus Corona, DPR Minta Pemerintah Lindungi Mahasiswa Aceh di Wuhan | Neutral | |
| 2489 | Wabah Virus Corona, Host Minta Maaf Video Makan Sup Kelelawar | Positive | |
| 2490 | Wabah Virus Corona, Ibu Hamil 9 Bulan Minta Dievakuasi dari China | Neutral | |
| 2491 | Wabah Virus Corona, Jokowi Jelaskan Alasan Belum Evakuasi WNI di China | Neutral | |
| 2492 | Wabah Virus Corona, KBRI Beijing Bentuk Posko Pantau WNI di Cina | Positive | |
| 2493 | Wabah Virus Corona, Mahasiswa Nigeria Klaim Diabaikan Pemerintah | Negative | |
| 2494 | Wabah Virus Corona, Malaysia akan Evakuasi Warganya dari Wuhan | Neutral | |
| 2495 | Wabah Virus Corona, Malaysia Kirim 18 Juta Sarung Tangan Medis ke China | Neutral | |
| 2496 | Wabah Virus Corona, Maskapai di Asia Pasifik Rugi 27,8 Miliar Dolar | Negative | |
| 2497 | Wabah Virus Corona, Pemkot Bandung Tak Berencana Larang Turis Asal China | Neutral | |
| 2498 | Wabah Virus Corona, Presiden Jokowi Minta Masyarakat Tak Cemas Berlebihan | Positive | |
| 2499 | Wabah Virus Corona, Seorang Kakek Jadi Korban Penusukkan Gara-gara Rebutan Cairan | Negative | |
| 2500 | Wabah Virus Corona, 'Shan Cai' Sebar 10 Ribu Masker di Wuhan | Neutral | |

Gambar 4.31 Pelabelan Dataset Latihan

Dataset latihan dalam format CSV ini akan dipakai sebagai dataset untuk proses pembuatan model analisis sentimen menggunakan metode *Long Short-term Memory* (LSTM).

4.1.2 Pembuatan Model LSTM untuk Analisis Sentimen

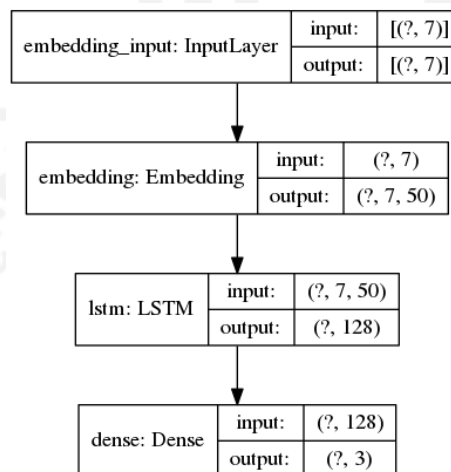
Pembuatan model LSTM untuk menganalisis sentimen tajuk berita *online* memerlukan *dataset* latih, *data preprocessing*, menentukan arsitektur model LSTM, dan melakukan *training* terhadap *dataset* latih. Hasil training *dataset* akan dievaluasi nilai akurasi dengan memperhatikan apakah ada *underfitting* atau *overfitting*.

Dataset latih adalah dataset dalam format CSV yang sudah diberi label *positive*, *neutral*, dan *negative*. Tahapan *data preprocessing* meliputi pembuangan *stopwords* pada kalimat, pembuangan kalimat yang hanya terdiri dari satu kata, pembuangan karakter *non ascii*, pembuangan tanda baca, pembuangan *double spaces*, pembuangan angka-angka, dan mengubah teks menjadi *lowercase*. Untuk mendapatkan model LSTM yang optimal, dilakukan dengan beberapa skenario eksperimen *hyperparameters tuning* sesuai tabel 3.2 dan dipilih model dengan tingkat akurasi tertinggi.

Semua eksperimen menggunakan lapisan embedding dengan parameter-parameter $input_dim=4312$, $output_dim=50$, $input_length=7$, dan *weights* yang diperoleh dari nilai pembobotan pada *embedding matrix* pada *Word Vector*.

a. Hasil Eksperimen #1

Pada Eksperimen #1, pembuatan model LSTM menggunakan parameter-parameter Jumlah layer = 1, Jumlah Units = 128, Ukuran Batch = 64, dan Maksimum epoch = 16. Berdasarkan parameter jumlah layer dan jumlah units, maka arsitektur model LSTM ditunjukkan pada gambar 4.32.



Gambar 4.32 Arsitektur Model LSTM Eksperimen #1

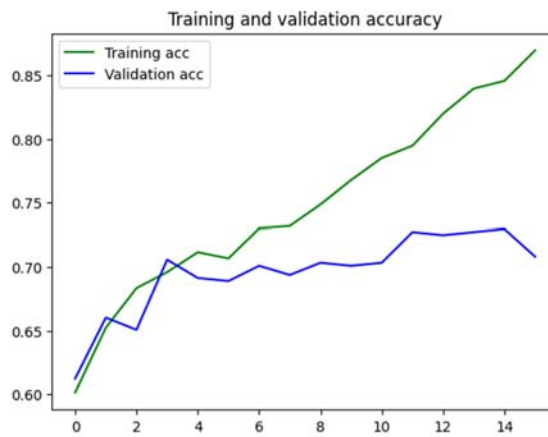
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 81.52 % dan akurasi *validation* 66.35 % seperti yang ditunjukkan pada

Gambar 4.33. Model ini juga memiliki tingkat *training loss* 46.12% dan *validation loss* 99.03%.

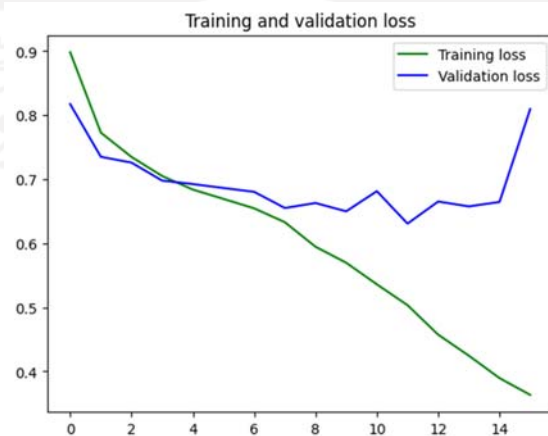
```
66/66 [=====] - 0s 2ms/step - loss: 0.4612 - acc: 0.8152
Training Accuracy: 0.8152
17/17 [=====] - 0s 2ms/step - loss: 0.9903 - acc: 0.6635
Validation Accuracy: 0.6635
```

Gambar 4.33 *Training and Validation Accuracy of Model #1*

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses training dalam 16 *epoch* ditunjukkan pada Gambar 4.34, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 81.52% dan *validation accuracy* 66.35%. Pada Gambar 4.35 menunjukkan *training loss* yang cenderung menurun, namun pada *validation loss* cenderung fluktuatif, dan diakhir epoch justru meningkat.



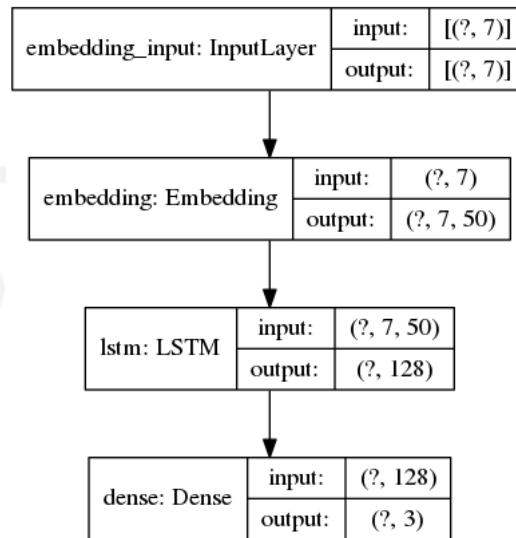
Gambar 4.34 Diagram Plot *Training and Validation Accuracy Model #1*



Gambar 4.35 Diagram Plot *Training and Validation Loss Model #1*

b. Hasil Eksperimen #2

Pada Eksperimen #2, pembuatan model LSTM menggunakan parameter-parameter Jumlah layer = 1, Jumlah Units = 128, Ukuran Batch = 64, dan Maksimum epoch = 32. Berdasarkan parameter jumlah layer dan jumlah units, maka arsitektur model LSTM ditunjukkan pada gambar 4.36.



Gambar 4.36 Arsitektur Model LSTM Eksperimen #2

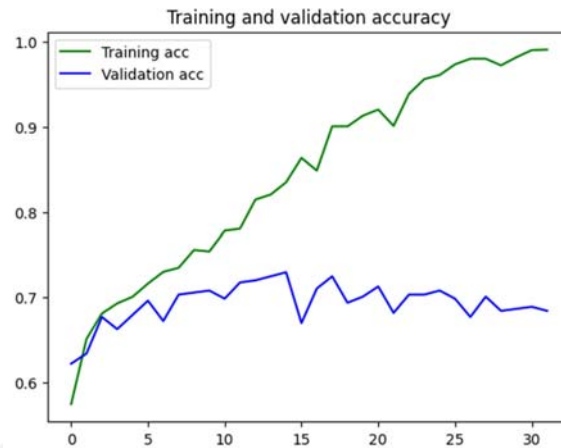
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 93.35% dan akurasi *validation* 67.30% seperti ditunjukkan pada Gambar 4.37. Model ini juga memiliki tingkat *training loss* 29.17% dan *validation loss* 158.7%.

```

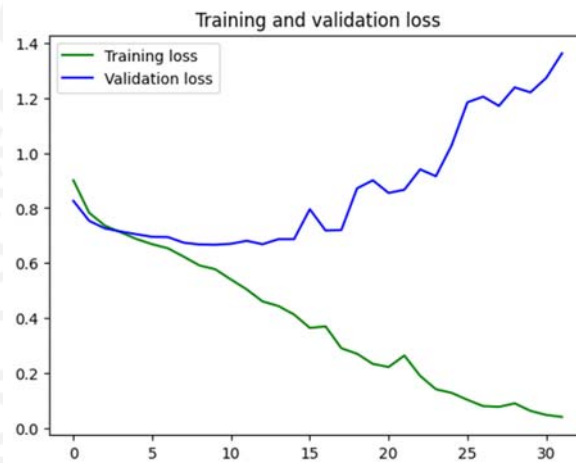
66/66 [=====] - 0s 2ms/step - loss: 0.2917 - acc: 0.9335
Training Accuracy: 0.9335
17/17 [=====] - 0s 2ms/step - loss: 1.5870 - acc: 0.6730
Validation Accuracy: 0.6730
  
```

Gambar 4.37 Training and Validation Accuracy of Model #2

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 32 *epoch* ditunjukkan pada Gambar 4.38, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 93.35% dan *validation accuracy* 67.3%. Gambar 4.39 menunjukkan *training loss* yang cenderung menurun, dan *validation loss* cenderung meningkat.



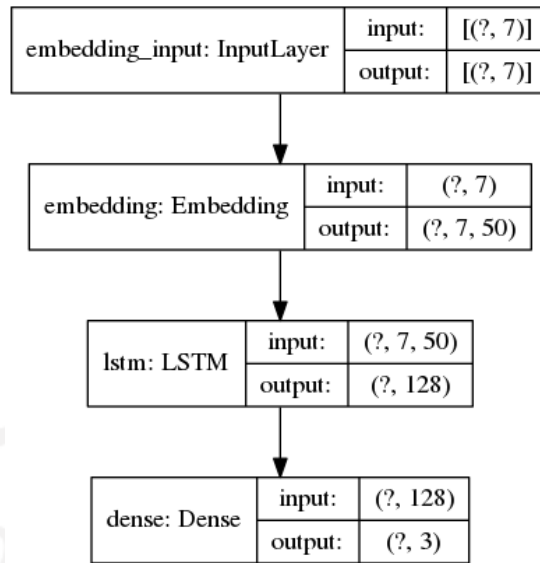
Gambar 4.38 Diagram Plot *Training and Validation Accuracy Model #2*



Gambar 4.39 Diagram Plot *Training and Validation Loss Model #2*

c. Hasil Eksperimen #3

Pada Eksperimen #3, pembuatan model LSTM menggunakan parameter-parameter jumlah layer = 1, jumlah *Units* = 128, ukuran *Batch* = 128, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer* dan jumlah *units*, maka arsitektur model LSTM ditunjukkan pada gambar 4.40.



Gambar 4.40 Arsitektur Model LSTM Eksperimen #3

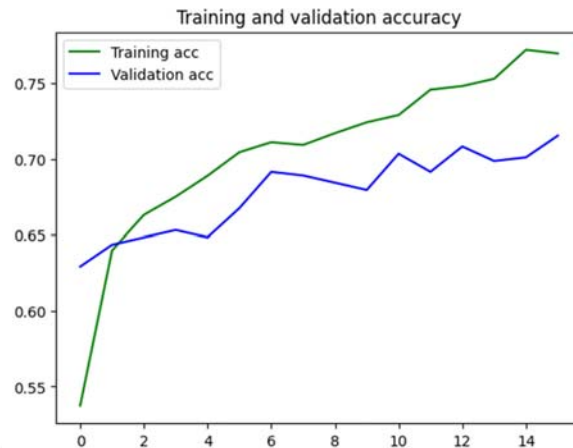
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 76.26 % dan akurasi *validation* 67.69 % seperti ditunjukkan pada Gambar 4.41. Model ini juga memiliki tingkat *training loss* 55.5% dan *validation loss* 76.8%.

```

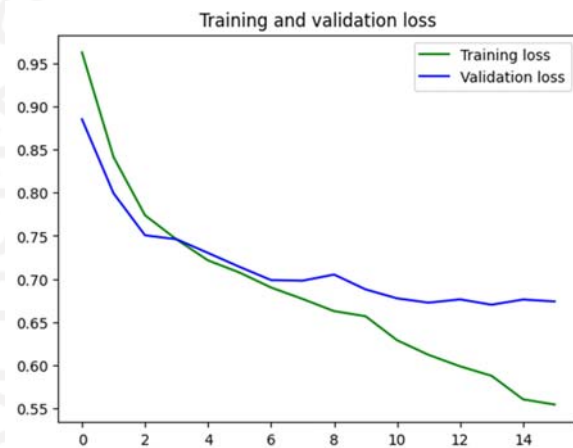
66/66 [=====] - 0s 2ms/step - loss: 0.5550 - acc: 0.7626
Training Accuracy: 0.7626
17/17 [=====] - 0s 2ms/step - loss: 0.7680 - acc: 0.6769
Validation Accuracy: 0.6769
  
```

Gambar 4.41 Training and Validation Accuracy of Model #3

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 epoch ditunjukkan pada Gambar 4.42, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 76.26% dan *validation accuracy* 67.69%. Gambar 4.43 menunjukkan *training loss* dan *validation loss* yang cenderung menurun, namun *validation loss* terlihat stagnan di sekitar 68%.



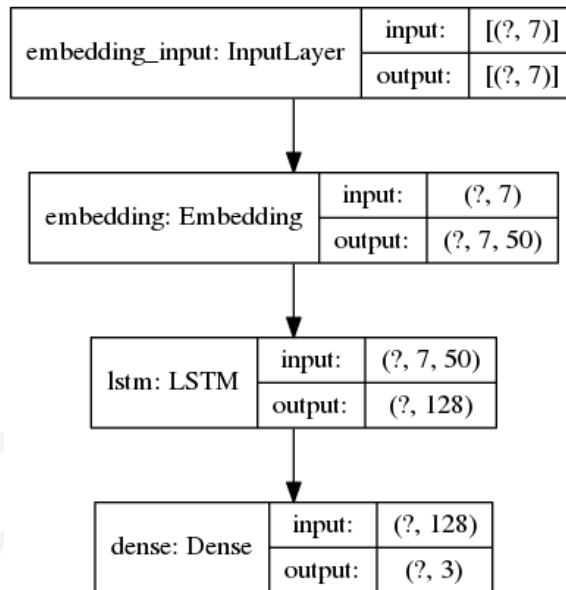
Gambar 4.42 Diagram Plot *Training and Validation Accuracy Model #3*



Gambar 4.43 Diagram Plot *Training and Validation Loss Model #3*

d. Hasil Eksperimen #4

Pada Eksperimen #4, pembuatan model LSTM menggunakan parameter-parameter Jumlah layer = 1, Jumlah Units = 128, Ukuran Batch = 128, dan Maksimum epoch = 32. Berdasarkan parameter jumlah *layer* dan jumlah *units*, maka arsitektur model LSTM ditunjukkan pada gambar 4.44.



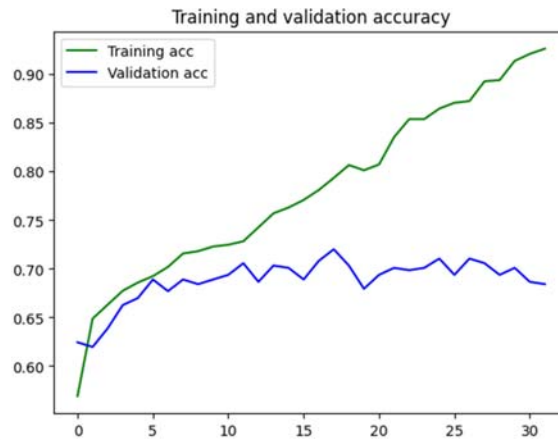
Gambar 4.44 Arsitektur Model LSTM Eksperimen #4

Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 88.18 % dan akurasi *validation* 69.41 % seperti yang ditunjukkan pada Gambar 4.45. Model ini juga memiliki tingkat *training loss* 35.04% dan *validation loss* 103.4%.

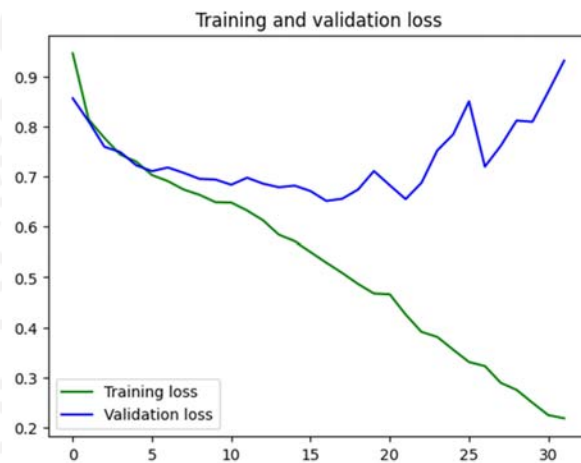
```
66/66 [=====] - 0s 2ms/step - loss: 0.3504 - acc: 0.8818
Training Accuracy: 0.8818
17/17 [=====] - 0s 2ms/step - loss: 1.0340 - acc: 0.6941
Validation Accuracy: 0.6941
```

Gambar 4.45 Training and Validation Accuracy of Model #4

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 32 *epoch* ditunjukkan pada Gambar 4.46, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 88.18% dan *validation accuracy* 69.41%. Gambar 4.47 menunjukkan *training loss* yang cenderung menurun, namun *validation loss* terlihat turun di awal 16 *epoch*, namun mendadak menjadi meningkat secara fluktuatif setelah 16 *epoch* hingga 32 *epoch*.



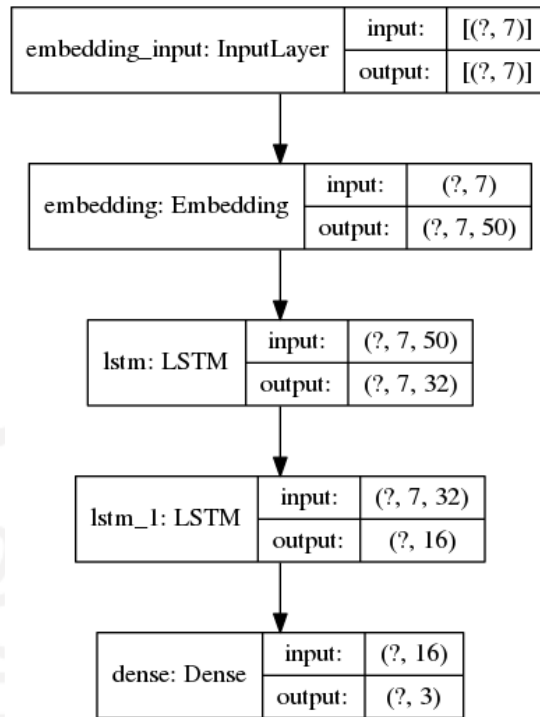
Gambar 4.46 Diagram Plot *Training and Validation Accuracy* Model #4



Gambar 4.47 Diagram Plot *Training and Validation Loss* Model #4

e. Hasil Eksperimen #5

Pada Eksperimen #5, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 2, Jumlah *Units* = 32 dan 16, *Dropouts* = 0.2, Ukuran *Batch* = 64, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.48.



Gambar 4.48 Arsitektur Model LSTM Eksperimen #5

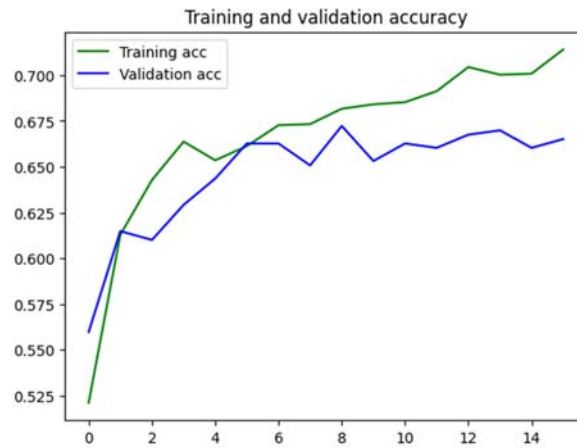
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 73.72 % dan akurasi *validation* 68.64 % seperti ditunjukkan pada Gambar 4.49. Model ini juga memiliki tingkat *training loss* 62.51% dan *validation loss* 72.98%.

```

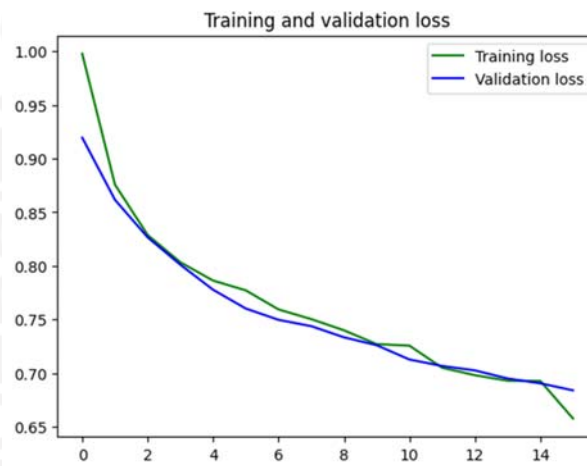
66/66 [=====] - 0s 2ms/step - loss: 0.6251 - acc: 0.7372
Training Accuracy: 0.7372
17/17 [=====] - 0s 2ms/step - loss: 0.7298 - acc: 0.6864
Validation Accuracy: 0.6864
  
```

Gambar 4.49 Training and Validation Accuracy of Model #5

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 *epoch* ditunjukkan pada Gambar 4.50, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 73.72% dan *validation accuracy* 68.64%. Gambar 4.51 menunjukkan *training loss* dan *validation loss* yang sama-sama cenderung menurun.



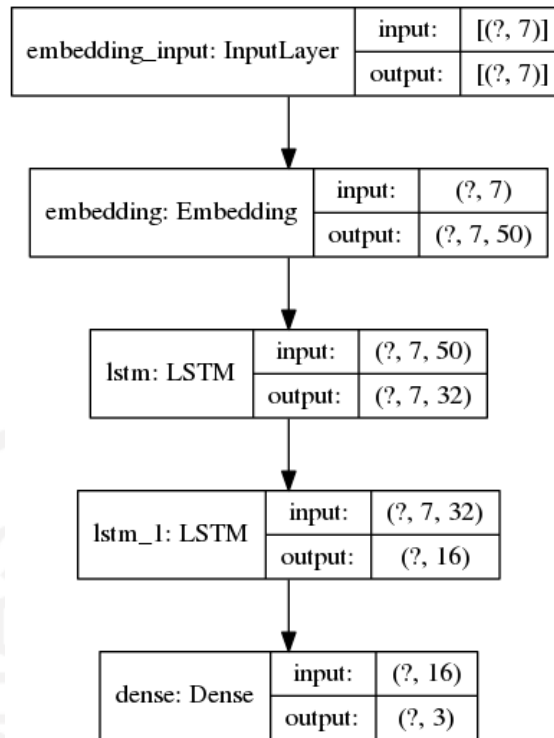
Gambar 4.50 Diagram Plot *Training and Validation Accuracy Model #5*



Gambar 4.51 Diagram Plot *Training and Validation Loss Model #5*

f. Hasil Eksperimen #6

Pada Eksperimen #6, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 2, Jumlah *Units* =32 dan 16, *Dropouts* = 0.2, Ukuran *Batch* = 64, dan Maksimum *epoch* = 32. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.52.



Gambar 4.52 Arsitektur Model LSTM Eksperimen #6

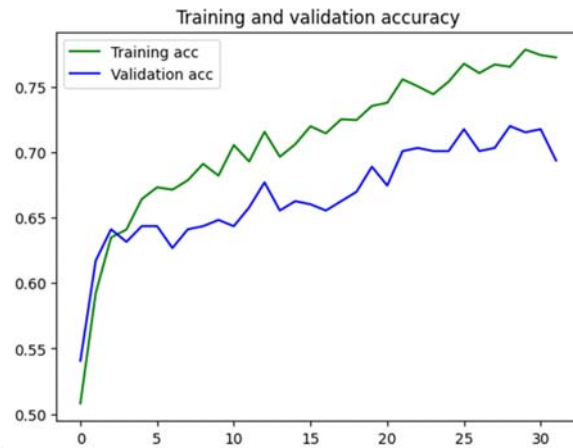
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 79.94 % dan akurasi *validation* 68.64 % seperti ditunjukkan pada Gambar 4.53. Model ini juga memiliki tingkat *training loss* 48.79% dan *validation loss* 77.22%.

```

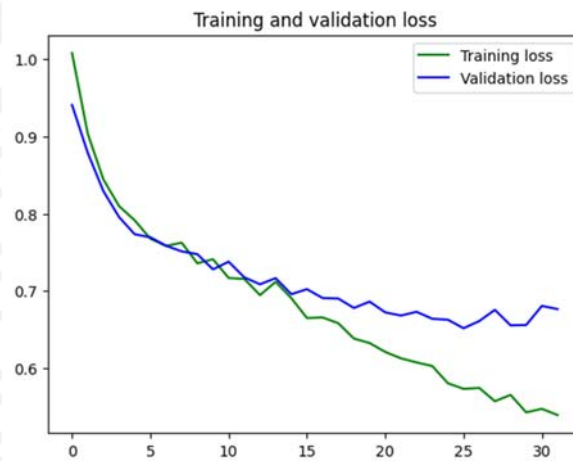
66/66 [=====] - 0s 2ms/step - loss: 0.4879 - acc: 0.7994
Training Accuracy: 0.7994
17/17 [=====] - 0s 2ms/step - loss: 0.7722 - acc: 0.6864
Validation Accuracy: 0.6864
  
```

Gambar 4.53 Training and Validation Accuracy of Model #6

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 32 *epoch* ditunjukkan pada Gambar 4.54, diketahui bahwa selama proses pelatihan model dalam 32 *epoch* menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 79.94% dan *validation accuracy* 68.64%. Gambar 4.55 menunjukkan *training loss* yang cenderung menurun, dan *validation loss* yang mulai stagnan di sekitar 68% di *epoch* 25 hingga *epoch* 32.



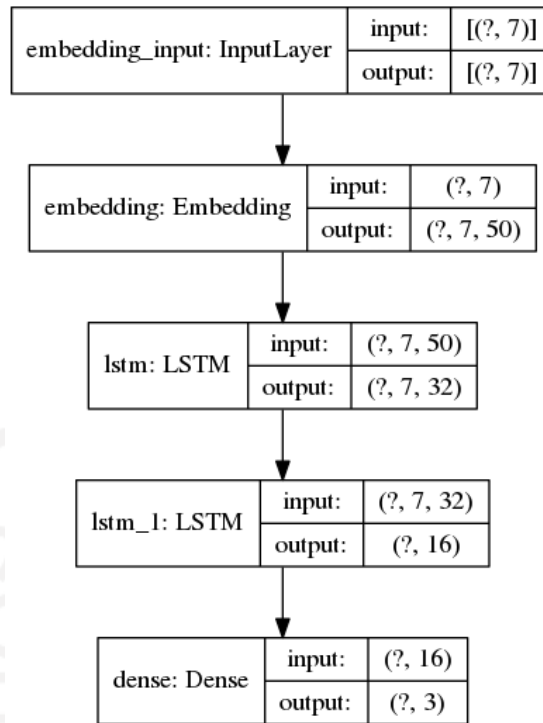
Gambar 4.54 Diagram Plot *Training and Validation Accuracy Model #6*



Gambar 4.55 Diagram Plot *Training and Validation Loss Model #6*

g. Hasil Eksperimen #7

Pada Eksperimen #7, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 2, Jumlah *Units* = 32 dan 16, *Dropouts* = 0.2, Ukuran *Batch* = 128, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.56.



Gambar 4.56 Arsitektur Model LSTM Eksperimen #7

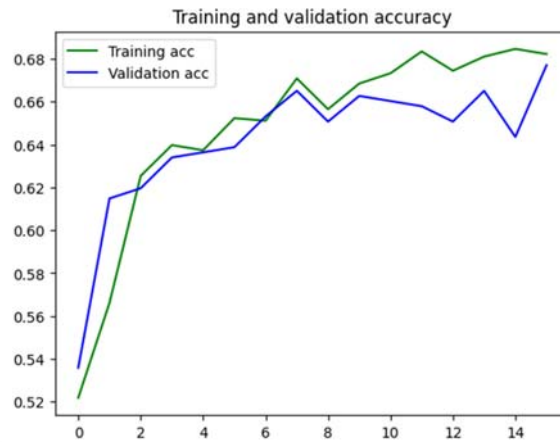
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 69.27 % dan akurasi *validation* 66.35 % seperti ditunjukkan pada Gambar 4.57. Model ini juga memiliki tingkat *training loss* 70.04% dan *validation loss* 75.23%.

```

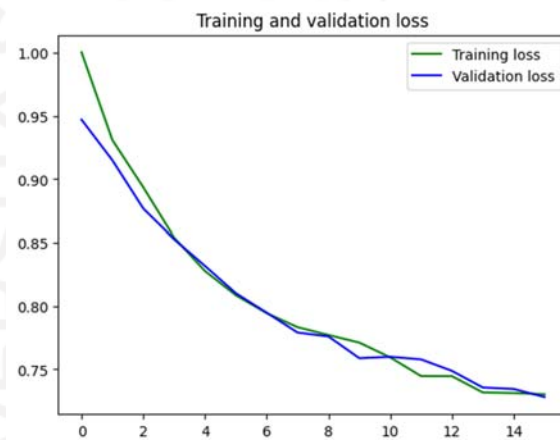
66/66 [=====] - 0s 2ms/step - loss: 0.7004 - acc: 0.6927
Training Accuracy: 0.6927
17/17 [=====] - 0s 2ms/step - loss: 0.7523 - acc: 0.6635
Validation Accuracy: 0.6635
  
```

Gambar 4.57 Training and Validation Accuracy of Model #7

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 *epoch* ditunjukkan pada Gambar 4.58, diketahui bahwa selama proses pelatihan model menunjukkan *training accuracy* dan *validation accuracy* yang sama-sama cenderung meningkat namun fluktuatif. Selisih nilai *training accuracy* dan *validation accuracy* cukup kecil yaitu 2.92%. Gambar 4.59 menunjukkan *training loss* dan *validation loss* sama-sama cenderung menurun secara *smooth*.



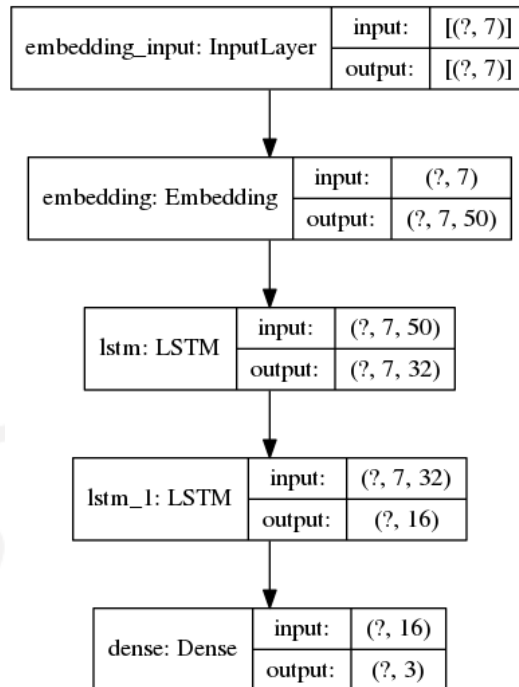
Gambar 4.58 Diagram Plot *Training and Validation Accuracy Model #7*



Gambar 4.59 Diagram Plot *Training and Validation Loss Model #7*

h. Hasil Eksperimen #8

Pada Eksperimen #8, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 2, Jumlah *Units* = 32 dan 16, *Dropouts* = 0.2, Ukuran *Batch* = 128, dan Maksimum *epoch* = 32. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.60.



Gambar 4.60 Arsitektur Model LSTM Eksperimen #8

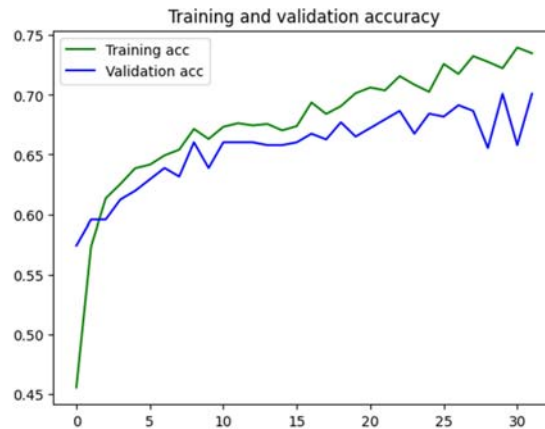
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 75.54 % dan akurasi *validation* 67.30 % seperti ditunjukkan pada Gambar 4.61. Model ini juga memiliki tingkat *training loss* 59.53% dan *validation loss* 75.58%.

```

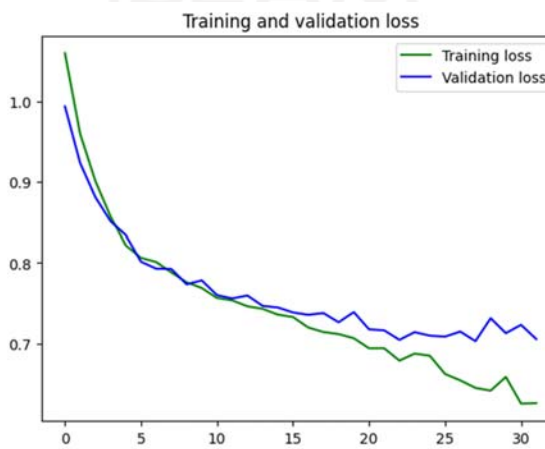
66/66 [=====] - 0s 2ms/step - loss: 0.5953 - acc: 0.7554
Training Accuracy: 0.7554
17/17 [=====] - 0s 2ms/step - loss: 0.7558 - acc: 0.6730
Validation Accuracy: 0.6730
  
```

Gambar 4.61 Training and Validation Accuracy of Model #8

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 32 *epoch* ditunjukkan pada Gambar 4.62, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 75.54% dan *validation accuracy* 67.3%. Gambar 4.63 menunjukkan *training loss* dan *validation loss* yang cenderung menurun secara fluktuatif.



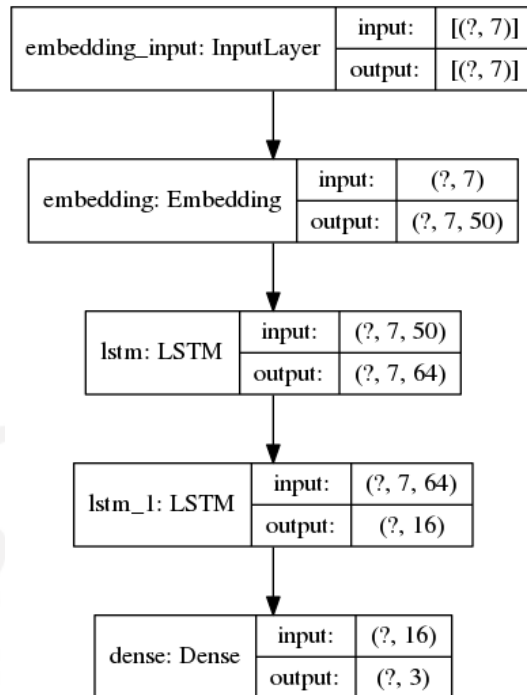
Gambar 4.62 Diagram Plot *Training and Validation Accuracy Model #8*



Gambar 4.63 Diagram Plot *Training and Validation Loss Model #8*

i. Hasil Eksperimen #9

Pada Eksperimen #9, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 2, Jumlah *Units* =64 dan 16, *Dropouts* = 0.2, Ukuran *Batch* = 64, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.64.



Gambar 4.64 Arsitektur Model LSTM Eksperimen #9

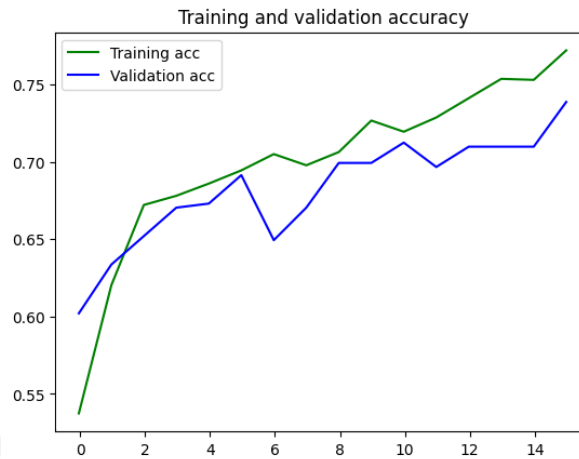
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 76.64 % dan akurasi *validation* 70.36 % seperti ditunjukkan pada Gambar 4.65. Model ini juga memiliki tingkat *training loss* 56.97% dan *validation loss* 74.04%.

```

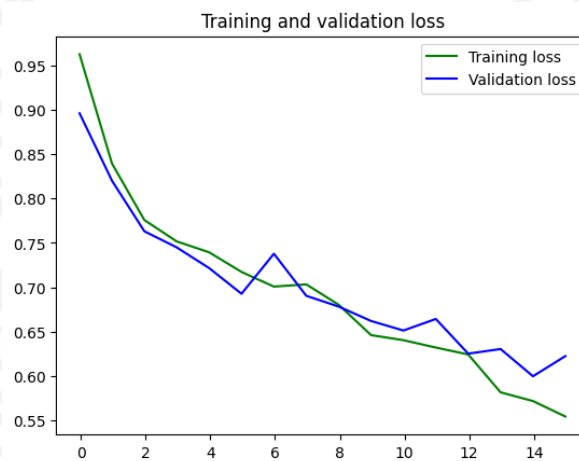
66/66 [=====] - 0s 3ms/step - loss: 0.5697 - acc: 0.7664
Training Accuracy: 0.7664
17/17 [=====] - 0s 3ms/step - loss: 0.7404 - acc: 0.7036
Validation Accuracy: 0.7036
  
```

Gambar 4.65 Training and Validation Accuracy of Model #9

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 *epoch* ditunjukkan pada Gambar 4.66, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 76.64% dan *validation accuracy* 70.36%. Gambar 4.67 menunjukkan *training loss* dan *validation loss* yang cenderung menurun secara fluktuatif.



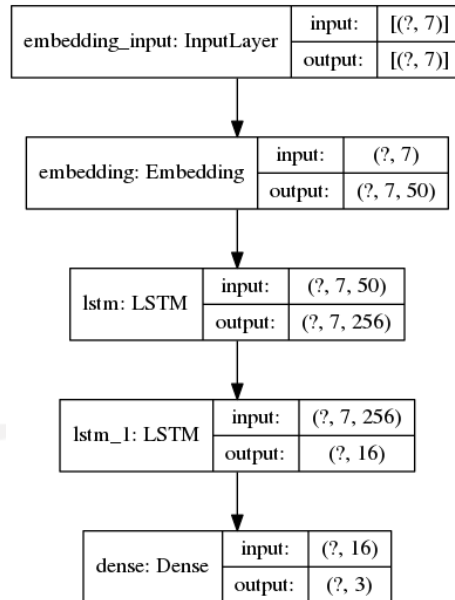
Gambar 4.66 Diagram Plot *Training and Validation Accuracy Model #9*



Gambar 4.67 Diagram Plot *Training and Validation Loss Model #9*

j. Hasil Eksperimen #10

Pada Eksperimen #10, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 2, Jumlah *Units* = 256 dan 16, *Dropouts* = 0.2, Ukuran *Batch* = 64, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.68.



Gambar 4.68 Arsitektur Model LSTM Eksperimen #10

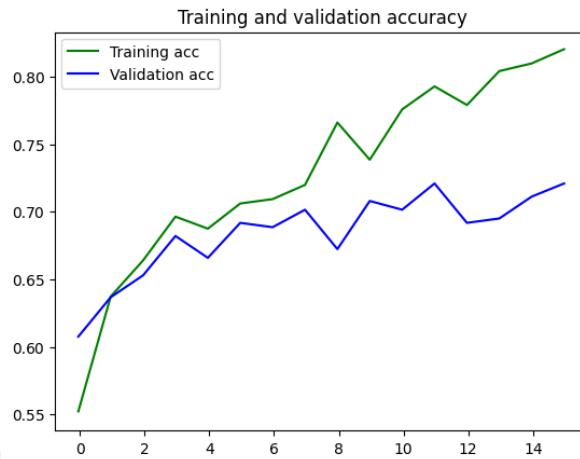
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 80.23 % dan akurasi *validation* 71.13 %. Model ini juga memiliki tingkat *training loss* 50.01% dan *validation loss* 75.33%.

```

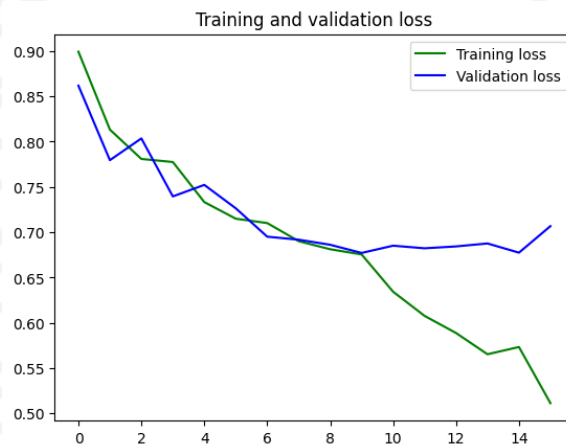
66/66 [=====] - 1s 9ms/step - loss: 0.5001 - acc: 0.8023
Training Accuracy: 0.8023
17/17 [=====] - 0s 9ms/step - loss: 0.7533 - acc: 0.7113
Validation Accuracy: 0.7113
  
```

Gambar 4.69 Training and Validation Accuracy of Model #10

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 *epoch* ditunjukkan pada Gambar 4.70, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 80.23% dan *validation accuracy* 71.13%. Gambar 4.71 menunjukkan *training loss* yang cenderung menurun dan *validation loss* yang mulai stagnan pada epoch 9 hingga epoch 16.



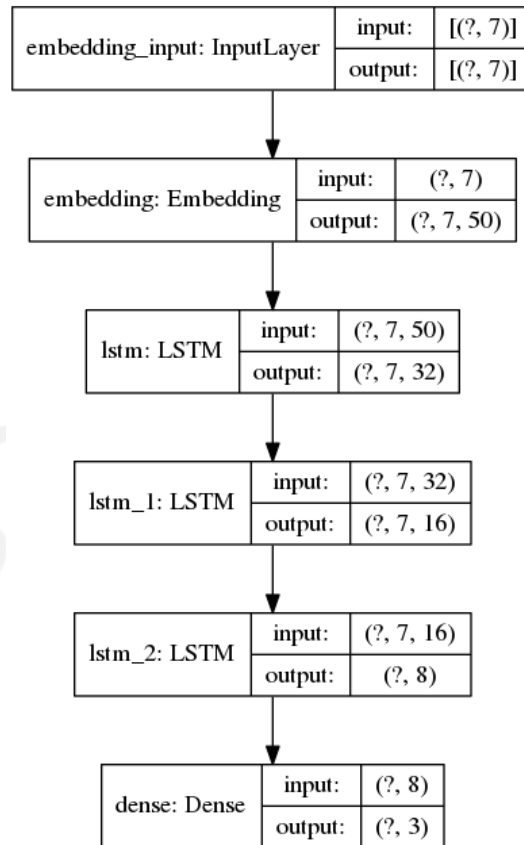
Gambar 4.70 Diagram Plot *Training and Validation Accuracy Model #10*



Gambar 4.71 Diagram Plot *Training and Validation Loss Model #10*

k. Hasil Eksperimen #11

Pada Eksperimen #11, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 3, Jumlah *Units* = 32, 16 dan 8, *Dropouts* = 0.5, Ukuran *Batch* = 64, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.72.



Gambar 4.72 Arsitektur Model LSTM Eksperimen #11

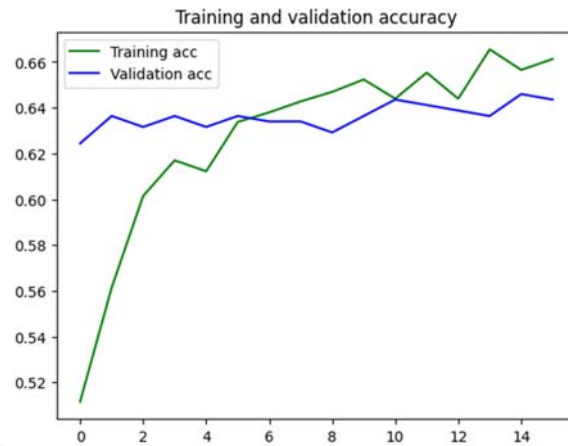
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 67.64 % dan akurasi *validation* 64.82 % seperti ditunjukkan pada Gambar 4.73. Model ini juga memiliki tingkat *training loss* 76.12% dan *validation loss* 79.75%.

```

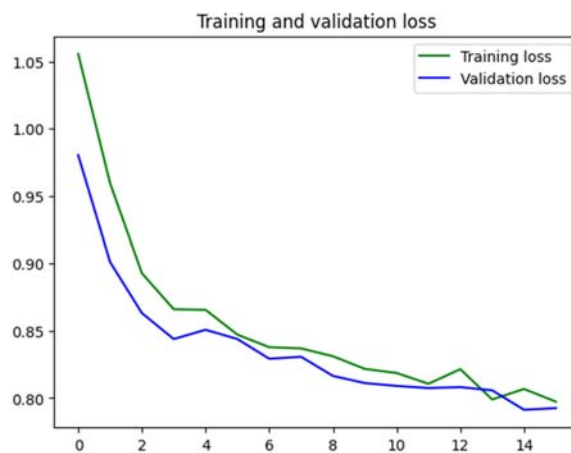
66/66 [=====] - 0s 3ms/step - loss: 0.7612 - acc: 0.6764
Training Accuracy: 0.6764
17/17 [=====] - 0s 2ms/step - loss: 0.7975 - acc: 0.6482
Validation Accuracy: 0.6482
  
```

Gambar 4.73 Training and Validation Accuracy of Model #11

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 *epoch* ditunjukkan pada Gambar 4.74, diketahui bahwa selama proses pelatihan model menunjukkan *training accuracy* yang fluktuatif meningkat dan *validation accuracy* yang cenderung stagnan di sekitar 62% hingga 64%. Selisih nilai *training accuracy* dan *validation accuracy* cukup kecil yaitu 2.82%. Gambar 4.75 menunjukkan *training loss* dan *validation loss* sama-sama cenderung menurun secara *smooth*.



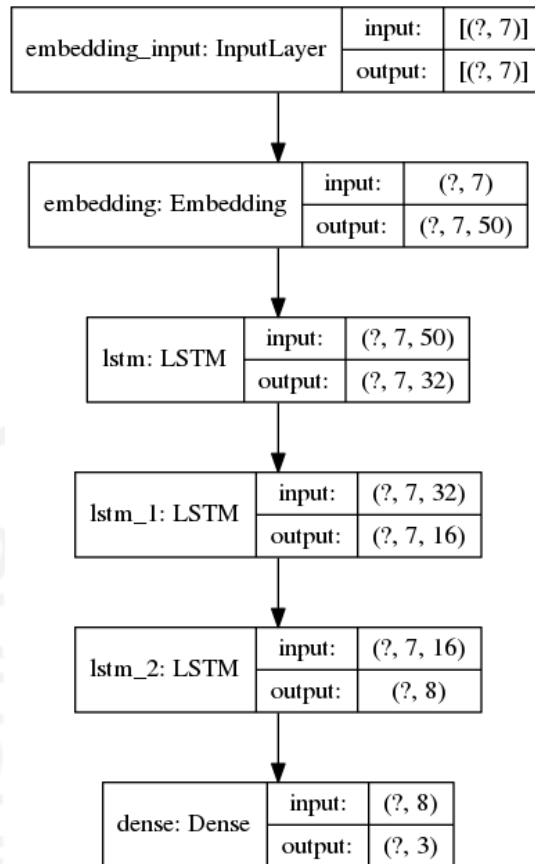
Gambar 4.74 Diagram Plot *Training and Validation Accuracy Model #11*



Gambar 4.75 Diagram Plot *Training and Validation Loss Model #11*

1. Hasil Eksperimen #12

Pada Eksperimen #12, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 3, Jumlah *Units* = 32, 16 dan 8, *Dropouts* = 0.5, Ukuran *Batch* = 64, dan Maksimum *epoch* = 32. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.76.



Gambar 4.76 Arsitektur Model LSTM Eksperimen #12

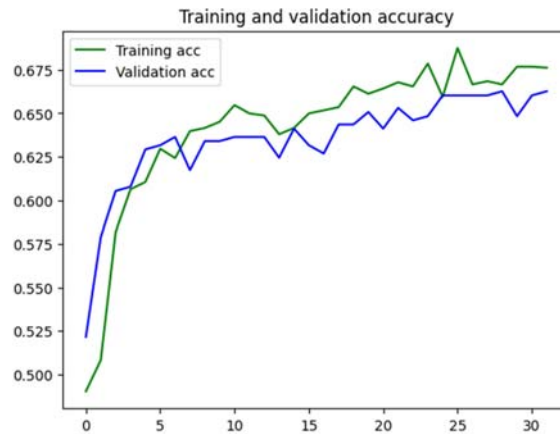
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 70.66 % dan akurasi *validation* 65.97 % seperti ditunjukkan pada Gambar 4.77. Model ini juga memiliki tingkat *training loss* 68.67% dan *validation loss* 78.25%.

```

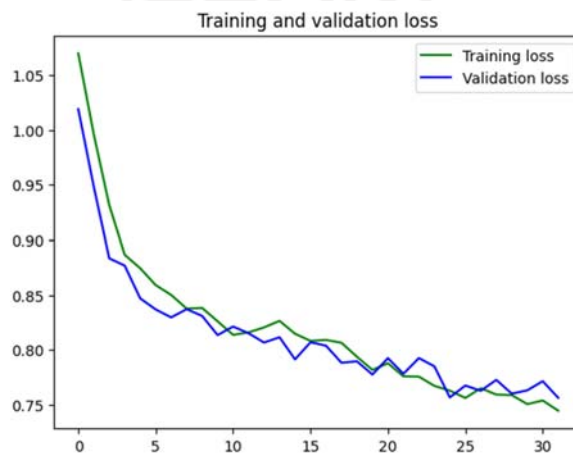
66/66 [=====] - 0s 3ms/step - loss: 0.6867 - acc: 0.7066
Training Accuracy: 0.7066
17/17 [=====] - 0s 2ms/step - loss: 0.7825 - acc: 0.6597
Validation Accuracy: 0.6597
  
```

Gambar 4.77 Training and Validation Accuracy of Model #12

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 32 *epoch* ditunjukkan pada Gambar 4.78, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 70.66% dan *validation accuracy* 65.97%. Gambar 4.79 menunjukkan *training loss* dan *validation loss* yang cenderung menurun secara fluktuatif.



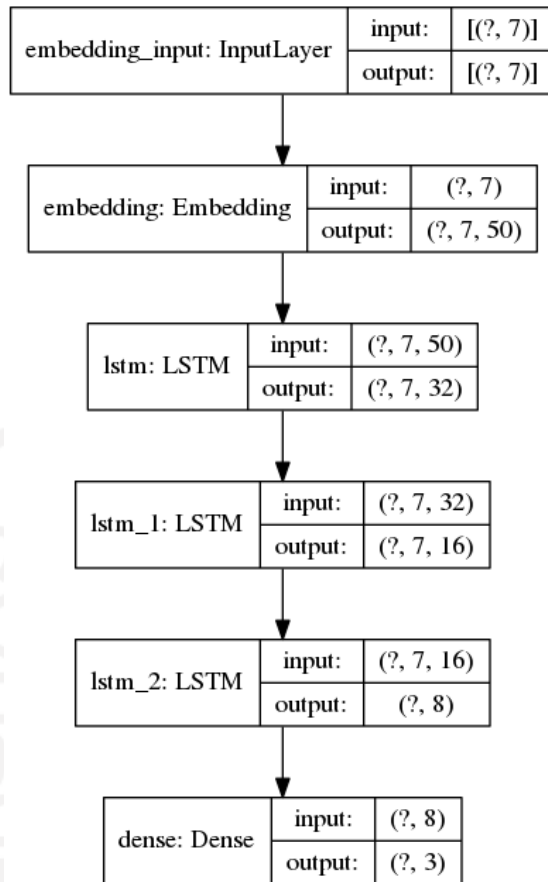
Gambar 4.78 Diagram Plot *Training and Validation Accuracy Model #12*



Gambar 4.79 Diagram Plot *Training and Validation Loss Model #12*

m. Hasil Eksperimen #13

Pada Eksperimen #13, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 3, Jumlah *Units* = 32, 16 dan 8, *Dropouts* = 0.5, Ukuran *Batch* = 128, dan Maksimum *epoch* = 16. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.80.



Gambar 4.80 Arsitektur Model LSTM Eksperimen #13

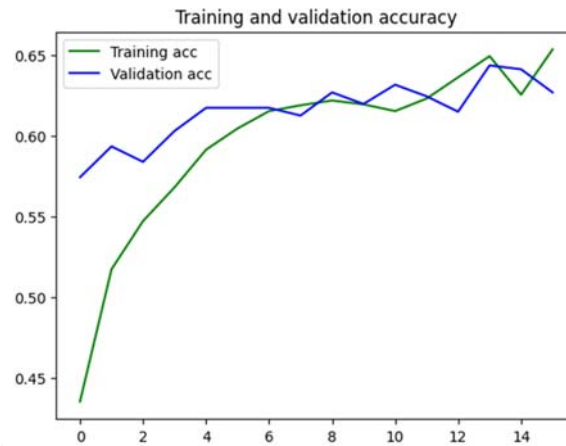
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 66.06 % dan akurasi *validation* 63.48 % seperti ditunjukkan pada Gambar 4.81. Model ini juga memiliki tingkat *training loss* 79.91% dan *validation loss* 82.46%.

```

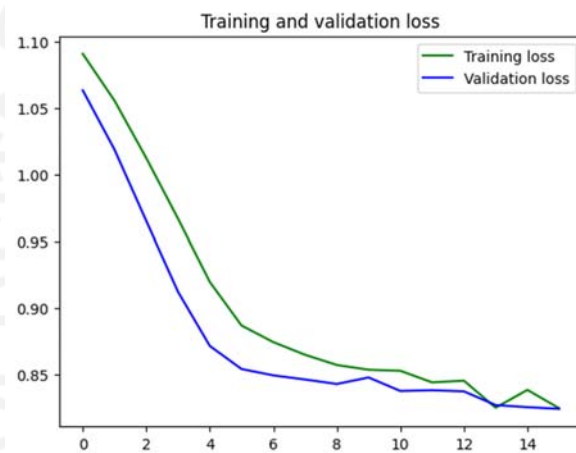
66/66 [=====] - 0s 3ms/step - loss: 0.7991 - acc: 0.6606
Training Accuracy: 0.6606
17/17 [=====] - 0s 2ms/step - loss: 0.8246 - acc: 0.6348
Validation Accuracy: 0.6348
  
```

Gambar 4.81 Training and Validation Accuracy of Model #13

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 16 *epoch* ditunjukkan pada Gambar 4.82, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 66.06% dan *validation accuracy* 63.48%. Gambar 4.83 menunjukkan *training loss* dan *validation loss* yang cenderung menurun secara fluktuatif.



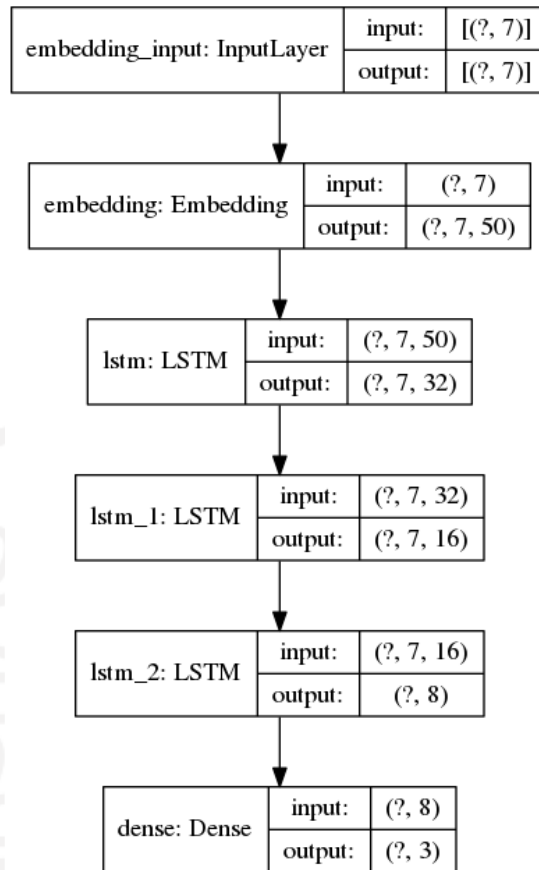
Gambar 4.82 Diagram Plot *Training and Validation Accuracy Model #13*



Gambar 4.83 Diagram Plot *Training and Validation Loss Model #13*

n. Hasil Eksperimen #14

Pada Eksperimen #14, pembuatan model LSTM menggunakan parameter-parameter Jumlah *layer* = 3, Jumlah *Units* = 32, 16 dan 8, *Dropouts* = 0.5, Ukuran *Batch* = 128, dan Maksimum *epoch* = 32. Berdasarkan parameter jumlah *layer*, jumlah *units*, dan nilai *Dropouts*, maka arsitektur model LSTM ditunjukkan pada gambar 4.84.



Gambar 4.84 Arsitektur Model LSTM Eksperimen #14

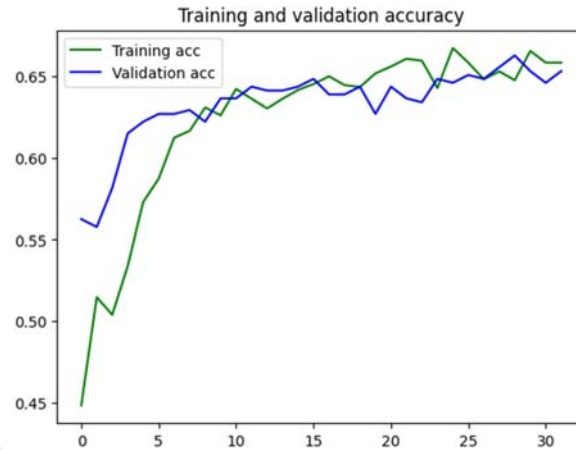
Pelatihan model menggunakan dataset latih menghasilkan model dengan tingkat akurasi *training* 68.69 % dan akurasi *validation* 65.77 % seperti tampak pada Gambar 4.85. Model ini juga memiliki tingkat *training loss* 72.38% dan *validation loss* 77.97%.

```

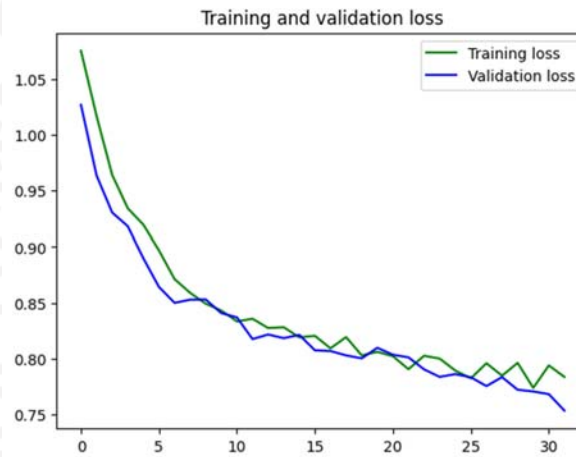
66/66 [=====] - 0s 3ms/step - loss: 0.7238 - acc: 0.6869
Training Accuracy: 0.6869
17/17 [=====] - 0s 2ms/step - loss: 0.7797 - acc: 0.6577
Validation Accuracy: 0.6577
  
```

Gambar 4.85 Training and Validation Accuracy of Model #14

Nilai *training accuracy* dan *validation accuracy* tiap *epoch* selama proses *training* dalam 32 *epoch* ditunjukkan pada Gambar 4.86, diketahui bahwa selama proses pelatihan model menunjukkan terjadinya *overfitting* pada model, terdapat selisih yang cukup besar antara *training accuracy* 68.69% dan *validation accuracy* 65.77%. Gambar 4.87 menunjukkan *training loss* dan *validation loss* yang cenderung menurun secara fluktuatif.

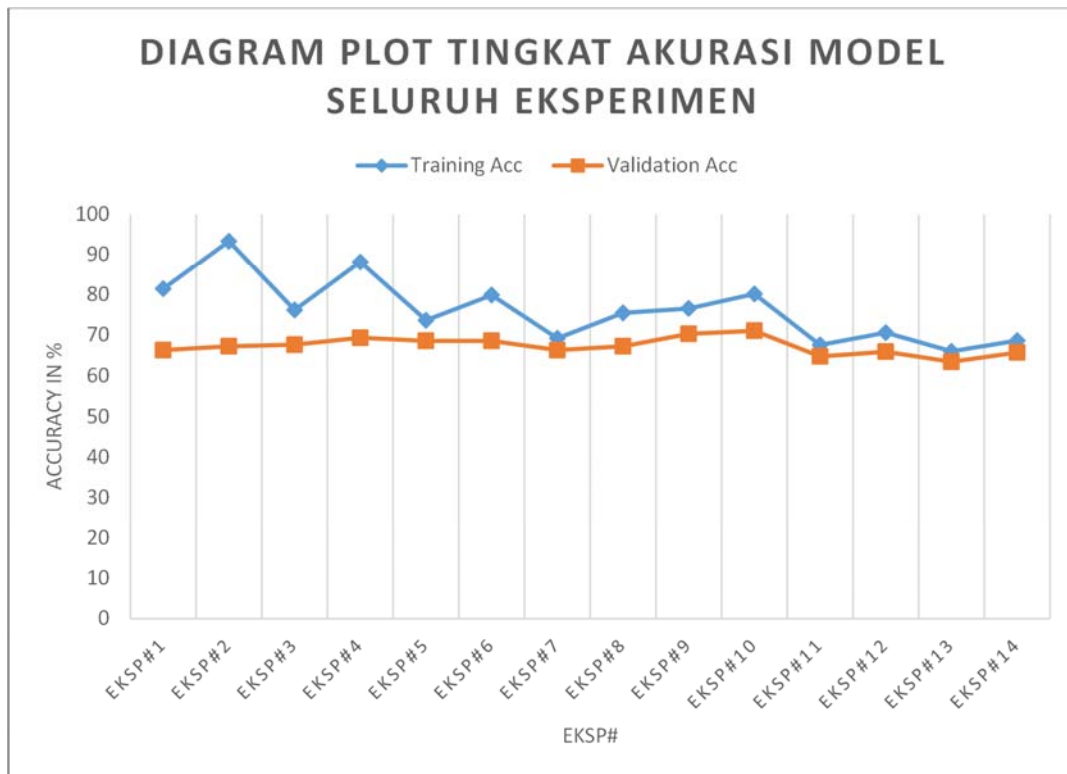


Gambar 4.86 *Training and Validation Accuracy Model #14*



Gambar 4.87 *Training and Validation Loss Model #14*

Untuk mempermudah mengevaluasi hasil semua eksperimen, penulis sajikan juga diagram plot untuk keseluruhan eksperimen yang ditunjukkan pada Gambar 4.88. Dengan memperhatikan diagram plot ini, diketahui bahwa nilai akurasi minimal atau diatas 70% terdapat pada eksperimen #9 dan eksperimen #10.



Gambar 4.88 Plot Diagram Akurasi Model dari Keseluruhan Eksperimen

Rangkuman nilai akurasi hasil pelatihan model LSTM berdasarkan skenario *hyperparameters tuning* disajikan pada Tabel 4.1 dengan detail spesifikasi masing-masing model mengacu pada Tabel 3.2.

Tabel 4.1 Nilai Akurasi Hasil Pelatihan Model LSTM

| Eksp # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Trn Acc % | 81.52 | 93.35 | 76.26 | 88.18 | 73.72 | 79.94 | 69.27 | 75.54 | 76.64 | 80.23 | 67.64 | 70.66 | 66.06 | 68.69 |
| Val Acc % | 66.35 | 67.30 | 67.69 | 69.41 | 68.64 | 68.64 | 66.35 | 67.30 | 70.36 | 71.13 | 64.82 | 65.97 | 63.48 | 65.77 |
| Over fitting | 15.17 | 26.05 | 8.57 | 18.77 | 5.08 | 11.30 | 2.92 | 8.24 | 6.28 | 9.1 | 2.82 | 4.69 | 2.58 | 2.92 |

Berdasarkan Tabel 4.1, semua model mengalami *overfitted* karena akurasi validasi lebih kecil dari akurasi pelatihan. Dengan mempertimbangkan nilai *overfitting* yang kecil dan prosentase akurasi minimal 70% diketahui model dengan nilai akurasi yang akan dipilih adalah Model #9 dengan nilai *training accuracy* 76.64%, *validation accuracy* 70.36%, dan *overfitted* 6.28%. Sehingga model #9 ini akan menjadi model yang dipakai pada proses pelabelan seluruh dataset tajuk berita *Online* berbahasa Indonesia.

Rendahnya tingkat akurasi semua model LSTM hasil dari beberapa skenario *hyperparameters tuning* diakibatkan dari beberapa sebab, yaitu adanya kemungkinan judul *headline* yang berupa click bait sehingga hasil analisis sentimen menjadi bias dan karena adanya unsur subyektifitas di dalam proses pelabelan manual saat pembuatan dataset latih.

4.1.3 Pelabelan *Unlabeled Dataset* menggunakan model LSTM terpilih

Unlabeled dataset adalah keseluruhan dataset yang belum memiliki label *positive*, *neutral*, dan *negative*. Dataset ini diperoleh dari *scrapping* situs berita online, sehingga data yang terekam adalah data *headline* berita *ter-up-to-date* dari situs-situs berita yang terdapat pada tabel 3.1.

Proses pelabelan *unlabeled dataset* ini diimplementasikan pada kode *python* yang eksekusinya dijadwalkan pada *crontab* setiap 5 menit. Pada prinsipnya script ini membaca dataset yang belum memiliki label *positive*, *neutral*, dan *negative* selanjutnya dilakukan analisis sentimen menggunakan model terpilih untuk mendapatkan label *positive*, *neutral*, atau *negative* dari tiap-tiap *headline* berita. Kode *python* untuk proses pelabelan penulis sajikan dalam *code snippet* berikut ini.

Code Snippet 4.14 Python script untuk pelabelan dataset

```
1. #!/usr/bin/env python
2. # coding: utf-8
3. model_name = "/home/ec2-
  user/results/models/lstm_model_eksperimen09.h5"
4.
5. import pandas as pd
6. import numpy as np
7. import matplotlib.pyplot as plt
8. import os
9. from sklearn.model_selection import train_test_split
10. from tensorflow.keras.utils import to_categorical
11. import math
12.
13. import tensorflow as tf
14.
15. from keras.models import load_model
16.
17. #preprocessing text
18. #membuang stopwords
19. import nltk
20. from nltk.corpus import stopwords
21. import re, string, unicodedata
22. from nltk import word_tokenize, sent_tokenize
23.
24. def removeStopword(str):
25.     stop_words = set(stopwords.words('indonesian'))
26.     word_tokens = word_tokenize(str)
27.     filtered_sentence = [w for w in word_tokens if not w in
  stop_words]
28.     return ' '.join(filtered_sentence)
29.
```

```

30. #membuang kalimat yang hanya terdiri dari satu kata
31. def removeSentence(str):
32.     word = str.split()
33.     wordCount = len(word)
34.     if(wordCount<=1):
35.         str = ''
36.     return str
37.
38. def cleaning(str):
39.     #buang non-ascii
40.     str = unicodedata.normalize('NFKD', str).encode('ascii',
'ignore').decode('utf-8', 'ignore')
41.     #buang URLs
42.     str = re.sub(r'(?i)\b(?:https?://|www\d{0,3}[.][a-z0-9.\-
]+[.][a-
z]{2,4})/?(?:[^\s()<>+|\\(|([^\s()<>+|\\(|([^\s()<>+|\\(|
s()<>+|\\(|([^\s()<>+|\\(|
str)
43.     #buang punctuations
44.     str = re.sub(r'[\W_]', ' ', str)
45.     #buang digit from string
46.     str = re.sub("\S*\d\S*", "", str).strip()
47.     #buang digit or numbers
48.     str = re.sub(r"\b\d+\b", " ", str)
49.     #buang double spaces
50.     str = re.sub('[\s]+', ' ', str)
51.     #ubah menjadi lowercase
52.     str = str.lower()
53.     return str
54.
55. def preprocessing(str):
56.     str = removeSentence(str)
57.     str = cleaning(str)
58.     str = removeStopword(str)
59.
60.     return str
61.
62.
63. # PRETRAINED WORD EMBEDDING
64. # Membaca Fasttext Pretrained Word Embedding
65. # 50 dimensional version (embedding dimension)
66.
67. embedding_dim = 50
68.
69. from datetime import datetime
70.
71. # current date and time
72. now = datetime.now()
73.
74. timestamp = datetime.timestamp(now)
75. dt_object = datetime.fromtimestamp(timestamp)
76.
77. print("Start Open Wordvec =", dt_object)
78. embeddings_index = dict()
79. f = open('/home/ec2-user/cc.id.50.vec.ori', encoding="utf8")
80. for line in f:
81.     values = line.split()
82.     word = values[0]
83.     coefs = np.asarray(values[1:], dtype='float32')
84.     embeddings_index[word] = coefs
85. f.close()

```

```

86. print('Loaded %s word vectors.' % len(embeddings_index))
87. now = datetime.now()
88.
89. timestamp = datetime.timestamp(now)
90. dt_object = datetime.fromtimestamp(timestamp)
91.
92. print("End =", dt_object)
93.
94. # Membuka data headline berita
95. nama_file = "/home/ec2-user/dataset_latih.csv"
96. df = pd.read_csv(nama_file,delimiter=";",encoding='ISO-8859-1')
97.
98. df.isnull().sum()
99.
100. df[['headline_id', 'sentiment_result']].head()
101.
102. df['sentiment_result'].value_counts()
103.
104. df['sentiment_result'] = df['sentiment_result'].replace('Negative',
0)
105. df['sentiment_result'] = df['sentiment_result'].replace('Neutral',
1)
106. df['sentiment_result'] = df['sentiment_result'].replace('Positive',
2)
107.
108.
109. X = df['headline_id'] # data
110. y = df['sentiment_result'] # labels
111.
112. #Preprocess data
113. Sentences = []
114. for st in X:
115.     r = preprocessing(st)
116.     Sentences.append(r)
117. X = Sentences
118.
119. from tensorflow.keras.preprocessing.text import Tokenizer
120. from tensorflow.keras.preprocessing.text import
text_to_word_sequence
121.
122. t = Tokenizer()
123. t.fit_on_texts(X)
124.
125. vocab_size = len(t.word_index) + 1
126.
127.
128. sequences = t.texts_to_sequences(X)
129.
130. def max_headline():
131.     for i in range(1, len(sequences)):
132.         max_length = len(sequences[0])
133.         if len(sequences[i]) > max_length:
134.             max_length = len(sequences[i])
135.     return max_length
136.
137. headline_num = max_headline()
138.
139. from tensorflow.keras.preprocessing.sequence import pad_sequences
140. maxlen = headline_num
141.
142. padded_X = pad_sequences(sequences, padding='post', maxlen=maxlen)

```

```

143.
144. labels = to_categorical(np.asarray(y))
145.
146. from sklearn.model_selection import train_test_split
147. X_train, X_test, y_train, y_test = train_test_split(padded_X,
    labels, test_size = 0.2, random_state = 0)
148.
149. embedding_dim = 50
150. embedding_matrix = np.zeros((vocab_size, embedding_dim))
151.
152. for word, i in t.word_index.items():
153.     embedding_vector = embeddings_index.get(word)
154.     if embedding_vector is not None:
155.         embedding_matrix[i] = embedding_vector
156.
157. from tensorflow.keras.layers import Embedding
158.
159. embedding_layer = Embedding(input_dim=vocab_size,
    output_dim=embedding_dim, weights=[embedding_matrix], input_length =
    headline_num, trainable=False)
160.
161. #BUILD NEURAL NETWORK
162. from tensorflow.keras.models import Sequential
163. from tensorflow.keras.layers import Dense
164. from tensorflow.keras.layers import Flatten
165. from tensorflow.keras.layers import LSTM
166. from tensorflow.keras.layers import GRU
167. from tensorflow.keras.layers import BatchNormalization
168.
169. # Convert labels
170. labels = to_categorical(np.asarray(y))
171.
172. #membuka model
173. model = load_model(model_name)
174. model.get_weights()
175. model.optimizer
176.
177. import mysql.connector
178. db = mysql.connector.connect(
179.     host="topsenaindonesia.com",
180.     user="naury",
181.     passwd="T0ps3na",
182.     database="thesis"
183. )
184.
185. cursor = db.cursor()
186. cursor.execute("SELECT * FROM headlines WHERE sentiment_result =
    ''")
187.
188. result = cursor.fetchall()
189. numrows = cursor.rowcount
190.
191. nomor = 1
192. hs = open("log_sentina.txt", "w")
193. for x in result:
194.     print('Sedang memroses data ke-' + str(nomor) + ' dari ' +
        str(numrows) + ' baris data')
195.     if x[2] != "":
196.         ids = x[0]
197.         media = x[1]
198.         headline_title = x[2]

```



```

199.     headline_date = x[4]
200.
201.     teks = [x[2]]
202.
203.     instance = teks
204.     instance = t.texts_to_sequences(instance)
205.
206.     flat_list = []
207.     for sublist in instance:
208.         for item in sublist:
209.             flat_list.append(item)
210.
211.     flat_list = [flat_list]
212.
213.     instance = pad_sequences(flat_list, padding='post',
        maxlen=maxlen)
214.
215.     sentiment=model.predict(instance)
216.
217.     print(teks)
218.     print(sentiment)
219.     print()
220.
221.     hs.write("{}\n".format(teks))
222.     hs.write("{}\n".format(sentiment))
223.     if(np.argmax(sentiment)==0):
224.         hs.write("Negative" + "\n\n")
225.         sentiment_result = "Negative"
226.     elif(np.argmax(sentiment)==1):
227.         hs.write("Neutral" + "\n\n")
228.         sentiment_result = "Neutral"
229.     elif(np.argmax(sentiment)==2):
230.         hs.write("Positive" + "\n\n")
231.         sentiment_result = "Positive"
232.     sql = "UPDATE `headlines` SET `sentiment_result` = '" +
        sentiment_result + "' WHERE `id` = '" + str(ids) + "'"
233.
234.     cursor.execute(sql)
235.     db.commit()
236.     print(sentiment_result)
237.     print()
238.     nomor+=1
239.
240. print("Selesai...")
241. hs.close()

```

Gambar 4.89 menunjukkan *dataset* `headlines` yang belum memiliki label *positive*, *neutral*, atau *negative* pada kolom `sentiment_result`. Dataset ini selanjutnya akan dilakukan proses pelabelan dengan menggunakan kode *python* seperti ditunjukkan pada *code snippet* 4.14.

| Options | | | | | | | sentiment_result |
|--------------------------|----|----------|--|------------------------------|--|--|------------------|
| | id | media | headline_id | date | | | |
| <input type="checkbox"/> | 10 | tempo.co | Wahyu Setiawan Mengundurkan Diri sebagai Komisione... | 2020-01-10 19:11:08.95205 | | | |
| <input type="checkbox"/> | 11 | tempo.co | Utam Kemayoran Masih Ditutup, Pengelola Kemayoran ... | 2020-01-10 19:11:08.95441 | | | |
| <input type="checkbox"/> | 13 | tempo.co | BEI Temukan Transaksi Saham Gorengan Rp 755 Milliar... | 2020-01-10 19:11:08.95681 | | | |
| <input type="checkbox"/> | 14 | tempo.co | Prancis Investigasi Jatuhnya Pesawat Ukraine Inter... | 2020-01-10 19:11:08.95792 | | | |
| <input type="checkbox"/> | 15 | tempo.co | KPU Sebut Permohonan PAW Harun Masiku Diteken Mega... | 2020-01-10 19:11:08.95882 | | | |
| <input type="checkbox"/> | 16 | tempo.co | Awal 2020, Modal Asing Masuk RI Sudah Rp 10 Triliu... | 2020-01-10 19:11:08.96011 | | | |
| <input type="checkbox"/> | 17 | tempo.co | Berkas Sudah P21, Kasus Sabu AKBP Benny Alamsyah S... | 2020-01-10 19:11:08.96129 | | | |
| <input type="checkbox"/> | 18 | tempo.co | Penyidik KPK Dapat Izin Penggeledahan Kasus Suap W... | 2020-01-10 19:11:08.96233 | | | |

Gambar 4.89 Contoh Sebagian *Dataset Headline* Berita Belum Memiliki Label

Jalannya proses pelabelan dataset *'headlines'* yang tersimpan di database ditunjukkan pada Gambar 4.90. Pada saat penelitian ini dilaksanakan, jumlah dataset yang dilabeli sebanyak 201.213 record.

```

ec2-user@ec2-3-89-62-177.compute-1.amazonaws.com:22 - Bitvise xterm - ec2-user@ip-172-31-24-6:~
['KPK: Nggak Usah Ditantang, Kalau Dirasa Perlu ya Hasto Dipanggil']
[[0.1004799 0.02030521 0.87921494]]

Positive

Sedang memproses data ke-913 dari 201213 baris data
['Andi Arief: Mereka Bisa Lawan Penggeledahan KPK, Tapi Tidak Bisa Ubah Fakta']
[[0.71938086 0.00334138 0.27727786]]

Negative

Sedang memproses data ke-914 dari 201213 baris data
[' Liverpool Memang Kejam ']
[[0.05508967 0.00374052 0.9411698 ]]

```

Gambar 4.90 Proses Jalannya Pelabelan Dataset *'headlines'*

Hasil proses pelabelan berupa dataset *'headlines'* yang sudah memiliki label ditunjukkan pada Gambar 4.91. Sebanyak 201.213 record berhasil dilabeli dengan label *Positive*, *Neutral*, dan *Negative*.

| | id | media | headline_id | date | sentiment_result |
|---|----|----------|--|------------------------------|------------------|
| <input type="checkbox"/> Edit Copy Delete | 10 | tempo.co | Wahyu Setiawan Mengundurkan Diri sebagai Komisioner... | 2020-01-10 19:11:08.95205 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 11 | tempo.co | Utara Kemayoran Masih Ditutup, Pengelola Kemayoran ... | 2020-01-10 19:11:08.95441 | Negative |
| <input type="checkbox"/> Edit Copy Delete | 13 | tempo.co | BEI Temukan Transaksi Saham Gorengan Rp 755 Miliar... | 2020-01-10 19:11:08.95681 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 14 | tempo.co | Prancis Investigasi Jatuhnya Pesawat Ukraine Inter... | 2020-01-10 19:11:08.95792 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 15 | tempo.co | KPU Sebut Permohonan PAW Harun Masiku Diteken Mega... | 2020-01-10 19:11:08.95882 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 16 | tempo.co | Awal 2020, Modal Asing Masuk RI Sudah Rp 10 Triliu... | 2020-01-10 19:11:08.96011 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 17 | tempo.co | Berkas Sudah P21, Kasus Sabu AKBP Benny Alamsyah S... | 2020-01-10 19:11:08.96129 | Negative |
| <input type="checkbox"/> Edit Copy Delete | 18 | tempo.co | Penyidik KPK Dapat Izin Pengeledahan Kasus Suap W... | 2020-01-10 19:11:08.96233 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 19 | tempo.co | Mari Elka Pangestu Direktur Bank Dunia, Bos BI Uca... | 2020-01-10 19:11:08.96342 | Positive |
| <input type="checkbox"/> Edit Copy Delete | 20 | tempo.co | Saiful Ilah Ditahan, KPK Geledah Kantor Dinas PUBM... | 2020-01-10 19:11:08.96435 | Positive |

Gambar 4.91 Hasil Pelabelan Dataset `headlines`

4.1.4 Pemodelan Topik

Pemodelan topik LDA dijalankan untuk semua hasil analisis sentimen negatif, netral, dan positif. Untuk keperluan penelitian, penulis melakukan query pada dataset `headlines` dengan memilih data `headline` berita yang mengandung keyword “covid” dan “corona” dari rentang tanggal 5 April 2020 hingga 12 April 2020. Kode query penulis sajikan pada *Code Snippet 4.13*.

Code Snippet 4.15. Query Pembuatan Dataset Bersentimen Positif, Netral, dan Negatif

```
CREATE TABLE IF NOT EXISTS `dataset_positif`
SELECT * FROM `headlines` WHERE (`headline_id` LIKE '%corona%' OR
`headline_id` LIKE '%covid%')
AND (`date` >= '2020-04-05' AND `date` <='2020-04-12') AND
(`sentiment_result`='Positive');

CREATE TABLE IF NOT EXISTS `dataset_netral`
SELECT * FROM `headlines` WHERE (`headline_id` LIKE '%corona%' OR
`headline_id` LIKE '%covid%')
AND (`date` >= '2020-04-05' AND `date` <='2020-04-12') AND
(`sentiment_result`='Neutral');

CREATE TABLE IF NOT EXISTS `dataset_negatif`
SELECT * FROM `headlines` WHERE (`headline_id` LIKE '%corona%' OR
`headline_id` LIKE '%covid%')
AND (`date` >= '2020-04-05' AND `date` <='2020-04-12') AND
(`sentiment_result`='Negative');
```

Hasil dari query tersebut berupa 3 buah tabel bernama `dataset_positif`, `dataset_netral`, dan `dataset_negatif`. Ketiga dataset tersebut dikonversi menjadi format CSV yang dapat dibuka di aplikasi spreadsheet seperti yang ditunjukkan pada Gambar 4.92, Gambar 4.93, dan Gambar 4.94.

| id | media | headline_id | date | sentiment_result |
|-------|------------|-----------------------------------|-----------------|------------------|
| 85086 | liputan6.c | 18 Pasien Positif Corona Covid-11 | 2020-04-05 01:3 | Positive |
| 85089 | cnnindon | Sosial Distancing Corona Tak Teki | 2020-04-05 01:4 | Positive |
| 85092 | rmol.id | Penyanyi Pink Dan Anaknya Sem | 2020-04-05 02:2 | Positive |
| 85094 | liputan6.c | RS Rujukan Corona Covid-19 di Ki | 2020-04-05 02:3 | Positive |
| 85095 | rmol.id | Kabinet Gaduh Dan Berseminya C | 2020-04-05 02:4 | Positive |
| 85098 | rmol.id | Bantu Pemda Tangani Pandemi | 2020-04-05 02:5 | Positive |
| 85099 | detik.com | Yasonna Bantah Akan Bebasan k | 2020-04-05 03:0 | Positive |
| 85117 | merdeka.i | Kasad Sediakan Toilet Portabel u | 2020-04-05 04:1 | Positive |
| 85125 | cnnindon | Wabah Corona Membayang Toky | 2020-04-05 04:5 | Positive |
| 85128 | cnnindon | Corona, Pembukaan Legoland Te | 2020-04-05 05:1 | Positive |
| 85137 | cnnindon | Pasien Covid-19 Spanyol Jalani Ki | 2020-04-05 05:3 | Positive |
| 85142 | tempo.co | Giliran Ratusan Satpol PP Tang | 2020-04-05 05:5 | Positive |
| 85146 | tempo.co | Spanyol Lampau Italia Jumlah Kc | 2020-04-05 05:5 | Positive |
| 85159 | cnnindon | WHO Kini Dukung Semua Orang f | 2020-04-05 06:2 | Positive |
| 85169 | detik.com | Pasien Termula Borah 5 Tahun c | 2020-04-05 06:5 | Positive |

Gambar 4.92 Dataset *Headline* Berita Berlabel *Positive*

| id | media | headline_id | date | sentiment_result |
|-------|-----------|--------------------------|-----------------|------------------|
| 85073 | rmol.id | Tunjang Pelayanan Bagi | 2020-04-05 00:1 | Neutral |
| 85074 | cnnindon | Drogba Marah Afrika Dis | 2020-04-05 00:2 | Neutral |
| 85076 | rmol.id | Termasuk Optimalkan D | 2020-04-05 00:4 | Neutral |
| 85077 | detik.com | Camat Bekasi Utara yang | 2020-04-05 00:4 | Neutral |
| 85101 | cnnindon | Wakil Presiden Barcelon | 2020-04-05 03:2 | Neutral |
| 85103 | cnnindon | Google Catat Mobilitas F | 2020-04-05 03:2 | Neutral |
| 85106 | rmol.id | Salut, Warga Lampung G | 2020-04-05 03:3 | Neutral |
| 85120 | rmol.id | Bantu Warga Terdampak | 2020-04-05 04:3 | Neutral |
| 85122 | cnnindon | Kemenkumham Telah Br | 2020-04-05 04:4 | Neutral |
| 85127 | merdeka.i | Dua Pasien Positif Covid | 2020-04-05 05:0 | Neutral |
| 85132 | detik.com | Heboh di Prabumulih: W | 2020-04-05 05:2 | Neutral |
| 85141 | cnnindon | VIDEO: PBB Serukan Gen | 2020-04-05 05:4 | Neutral |
| 85144 | tempo.co | KPK Tolak Covid-19 Jadi | 2020-04-05 05:5 | Neutral |

Gambar 4.93 Dataset *Headline* Berita Berlabel *Neutral*


```

27.
28.     #hapus punctuations
29.     str = re.sub(r'[\W_]', ' ', str)
30.
31.     #hapus digit
32.     str = re.sub("\S*\d\S*", "", str).strip()
33.
34.     #hapus angka
35.     str = re.sub(r"\b\d+\b", " ", str)
36.
37.     #Remove additional white spaces
38.     str = re.sub('[\s]+', ' ', str)
39.
40.     #jadikan lowercase
41.     str = str.lower()
42.     return str
43.
44. def preprocessing(str):
45.     str = removeSentence(str)
46.     str = cleaning(str)
47.     str = removeStopword(str)
48.
49.     return str
50.
51. #mulai data preprocessing
52. import pandas as pd
53. import xlswriter
54. fo = pd.read_excel('../dataset/dataset_positif.xlsx',
55.     sheet_name='dataset_positif') #read excel file
56. txt = fo['headline_id']
57. workbook =
58.     xlswriter.Workbook('../dataset/clean_data_positif.xlsx')
59. worksheet = workbook.add_worksheet()
60. row = 0
61. col = 0
62. rowHeaders = ['text']
63. worksheet.write_row(row, col, tuple(rowHeaders))
64.
65. for t in txt:
66.     new_txt = preprocessing(t)
67.     rowValues = [new_txt]
68.     row += 1
69.     worksheet.write_row(row, col, tuple(rowValues))
70.
71. workbook.close()
72.
73. fo = pd.read_excel('../dataset/dataset_netral.xlsx',
74.     sheet_name='dataset_netral') #read excel file
75. txt = fo['headline_id']
76. workbook =
77.     xlswriter.Workbook('../dataset/clean_data_netral.xlsx')
78. worksheet = workbook.add_worksheet()
79. row = 0
80. col = 0
81. rowHeaders = ['text']
82. worksheet.write_row(row, col, tuple(rowHeaders))
83.
84. for t in txt:
85.     new_txt = preprocessing(t)
86.     rowValues = [new_txt]

```

```

84.     row += 1
85.     worksheet.write_row(row, col, tuple(rowValues))
86.
87. workbook.close()
88.
89. fo = pd.read_excel('../dataset/dataset_negatif.xlsx',
    sheet_name='dataset_negatif') #read excel file
90. txt = fo['headline_id']
91. workbook =
    xlsxwriter.Workbook('../dataset/clean_data_negatif.xlsx')
92. worksheet = workbook.add_worksheet()
93. row = 0
94. col = 0
95. rowHeaders = ['text']
96. worksheet.write_row(row, col, tuple(rowHeaders))
97.
98. for t in txt:
99.     new_txt = preprocessing(t)
100.    rowValues = [new_txt]
101.    row += 1
102.    worksheet.write_row(row, col, tuple(rowValues))
103.
104. workbook.close()
105.

```

Masing-masing dataset hasil dari *data preprocessing* menjadi sumber data pada proses pemodelan topik menggunakan model LDA. Kode python untuk pemodelan topik LDA terhadap hasil sentimen analisis klasifikasi positif ditunjukkan pada *Code Snippet 4.15*, Kode python untuk pemodelan topik LDA terhadap hasil sentimen analisis klasifikasi netral ditunjukkan pada *Code Snippet 4.16*, dan Kode python untuk pemodelan topik LDA terhadap hasil sentimen analisis klasifikasi negatif ditunjukkan pada *Code Snippet 4.17*.

Code Snippet 4.17. Kode Python Pemodelan Topik LDA untuk Sentimen Positif

```

1. from helper import *
2. import warnings
3. warnings.filterwarnings('ignore')
4. import pandas as pd
5.
6. import gensim
7. import pyLDAvis.gensim
8.
9. fo = pd.ExcelFile('../dataset/clean_data_positif.xlsx')
10. df = pd.read_excel(fo, 'Sheet1')
11. text = df['text']
12. text_list = [i.split() for i in text]
13.
14. #Bigram & Trigram Models
15. from gensim.models import Phrases
16. bigram = Phrases(text_list, min_count=10)
17. trigram = Phrases(bigram[text_list])
18. for idx in range(len(text_list)):
19.     for token in bigram[text_list[idx]]:
20.         if '_' in token:
21.             # jika Token bigram, masukkan ke document.
22.             text_list[idx].append(token)

```



```

23.     for token in trigram[text_list[idx]]:
24.         if '_' in token:
25.             text_list[idx].append(token)
26.
27. from gensim import corpora, models
28. #buat dictionary
29. dictionary = corpora.Dictionary(text_list)
30. dictionary.filter_extremes(no_below=5, no_above=0.2)
31.
32. #buat corpus
33. doc_term_matrix = [dictionary.doc2bow(doc) for doc in text_list]
34.
35. #buat TF-IDF model
36. tfidf = models.TfidfModel(doc_term_matrix)
37. corpus_tfidf = tfidf[doc_term_matrix]
38.
39. from gensim.models.coherencemodel import CoherenceModel
40. from gensim.models.ldamodel import LdaModel
41. from gensim.corpora.dictionary import Dictionary
42. from numpy import array
43.
44. from multiprocessing import Process, freeze_support
45.
46. #menghitung nilai coherence
47. def compute_coherence_values(dictionary, corpus, texts, limit,
48. start, step):
49.     coherence_values = []
50.     model_list = []
51.     for num_topics in range(start, limit, step):
52.         model = LdaModel(corpus=corpus, id2word=dictionary,
53. num_topics=num_topics, iterations=100)
54.         model_list.append(model)
55.         coherencemodel = CoherenceModel(model=model,
56. texts=texts, dictionary=dictionary, coherence='c_v')
57.         coherence_values.append(coherencemodel.get_coherence())
58.     return model_list, coherence_values
59.
60. start=1
61. limit=20
62. step=1
63. model_list, coherence_values =
64. compute_coherence_values(dictionary, corpus=corpus_tfidf,
65. texts=text_list, start=start, limit=limit, step=step)
66. x = range(start, limit, step)
67.
68. for m, cv in zip(x, coherence_values):
69.     print("Num Topics =", m, " has Coherence Value of",
70. round(cv, 6))
71.
72. #buat diagram plot
73. import matplotlib.pyplot as plt
74.
75. plt.plot(x, coherence_values)
76. plt.xlabel("Num Topics")
77. plt.ylabel("Coherence score")
78. plt.legend(("coherence_values"), loc='best')
79. plt.show()
80.
81. model = LdaModel(corpus=corpus_tfidf, id2word=dictionary,
82. num_topics=6)

```



```

77. for idx, topic in model.print_topics(-1):
78.     print('Topic: {} Word: {}'.format(idx, topic))
79.
80. top_words_per_topic = []
81. for t in range(model.num_topics):
82.     top_words_per_topic.extend([(t, ) + x for x in
model.show_topic(t, topn = 10)])
83.
84. df = pd.DataFrame(top_words_per_topic, columns=['Topic',
'Word', 'P']).to_csv("top_words_positif.csv")
85. print(df)
86.
87. data = pyLDAvis.gensim.prepare(model, corpus_tfidf, dictionary)
88. print(data)
89. pyLDAvis.save_html(data, '../dataset/pyLDAvis_positif.html')
90.
91. from wordcloud import WordCloud as wd
92. for t in range(model.num_topics):
93.     plt.figure(figsize=(7,6))
94.     plt.imshow(wd(max_font_size=50,
min_font_size=6).fit_words(dict(model.show_topic(t, 200))))
95.     plt.axis("off")
96.     plt.title("Topic #" + str(t))
97.     plt.savefig("../dataset/pos-wcld-topic-#" + str(t) + ".png",
facecolor='k', bbox_inches='tight')
98. plt.show()

```

Code Snippet 4.18. Kode Python Pemodelan Topik LDA untuk Sentimen Netral

```

1. from helper import *
2. import warnings
3. warnings.filterwarnings('ignore')
4. import pandas as pd
5.
6. import gensim
7. import pyLDAvis.gensim
8.
9. fo = pd.ExcelFile('../dataset/clean_data_netral.xlsx')
10. df = pd.read_excel(fo, 'Sheet1')
11. text = df['text']
12. text_list = [i.split() for i in text]
13.
14. #Bigram & Trigram Models
15. from gensim.models import Phrases
16. bigram = Phrases(text_list, min_count=10)
17. trigram = Phrases(bigram[text_list])
18. for idx in range(len(text_list)):
19.     for token in bigram[text_list[idx]]:
20.         if '_' in token:
21.             # jika Token bigram, masukkan ke document.
22.             text_list[idx].append(token)
23.     for token in trigram[text_list[idx]]:
24.         if '_' in token:
25.             text_list[idx].append(token)
26.
27. from gensim import corpora, models
28. #buat dictionary
29. dictionary = corpora.Dictionary(text_list)
30. dictionary.filter_extremes(no_below=5, no_above=0.2)
31.

```

```

32. #buat corpus
33. doc_term_matrix = [dictionary.doc2bow(doc) for doc in
    text_list]
34.
35. #buat TF-IDF model
36. tfidf = models.TfidfModel(doc_term_matrix)
37. corpus_tfidf = tfidf[doc_term_matrix]
38.
39. from gensim.models.coherencemodel import CoherenceModel
40. from gensim.models.ldamodel import LdaModel
41. from gensim.corpora.dictionary import Dictionary
42. from numpy import array
43.
44. from multiprocessing import Process, freeze_support
45.
46. #menghitung nilai coherence
47. def compute_coherence_values(dictionary, corpus, texts, limit,
    start, step):
48.     coherence_values = []
49.     model_list = []
50.     for num_topics in range(start, limit, step):
51.         model = LdaModel(corpus=corpus, id2word=dictionary,
    num_topics=num_topics, iterations=100)
52.         model_list.append(model)
53.         coherencemodel = CoherenceModel(model=model,
    texts=texts, dictionary=dictionary, coherence='c_v')
54.         coherence_values.append(coherencemodel.get_coherence())
55.     return model_list, coherence_values
56.
57.
58. start=1
59. limit=20
60. step=1
61. model_list, coherence_values =
    compute_coherence_values(dictionary, corpus=corpus_tfidf,
    texts=text_list, start=start, limit=limit, step=step)
62. x = range(start, limit, step)
63.
64. for m, cv in zip(x, coherence_values):
65.     print("Num Topics =", m, " has Coherence Value of",
    round(cv, 6))
66.
67. #buat diagram plot
68. import matplotlib.pyplot as plt
69.
70. plt.plot(x, coherence_values)
71. plt.xlabel("Num Topics")
72. plt.ylabel("Coherence score")
73. plt.legend(("coherence_values"), loc='best')
74. plt.show()
75.
76. model = LdaModel(corpus=corpus_tfidf, id2word=dictionary,
    num_topics=6)
77. for idx, topic in model.print_topics(-1):
78.     print('Topic: {} Word: {}'.format(idx, topic))
79.
80. top_words_per_topic = []
81. for t in range(model.num_topics):
82.     top_words_per_topic.extend([(t, ) + x for x in
    model.show_topic(t, topn = 10)])
83.

```

```

84. df = pd.DataFrame(top_words_per_topic, columns=['Topic',
'Word', 'P']).to_csv("top_words_positif.csv")
85. print(df)
86.
87. data = pyLDAvis.gensim.prepare(model, corpus_tfidf, dictionary)
88. print(data)
89. pyLDAvis.save_html(data, '../dataset/pyLDAvis_netral.html')
90.
91. from wordcloud import WordCloud as wd
92. for t in range(model.num_topics):
93.     plt.figure(figsize=(7,6))
94.     plt.imshow(wd(max_font_size=50,
min_font_size=6).fit_words(dict(model.show_topic(t, 200))))
95.     plt.axis("off")
96.     plt.title("Topic #" + str(t))
97.     plt.savefig("../dataset/neu-wcld-topic-#" + str(t) + ".png",
facecolor='k', bbox_inches='tight')
98. plt.show()

```

Code Snippet 4.19. Kode Python Pemodelan Topik LDA untuk Sentimen Negatif

```

1. from helper import *
2. import warnings
3. warnings.filterwarnings('ignore')
4. import pandas as pd
5.
6. import gensim
7. import pyLDAvis.gensim
8.
9. fo = pd.ExcelFile('../dataset/clean_data_negatif.xlsx')
10. df = pd.read_excel(fo, 'Sheet1')
11. text = df['text']
12. text_list = [i.split() for i in text]
13.
14. #Bigram & Trigram Models
15. from gensim.models import Phrases
16. bigram = Phrases(text_list, min_count=10)
17. trigram = Phrases(bigram[text_list])
18. for idx in range(len(text_list)):
19.     for token in bigram[text_list[idx]]:
20.         if '_' in token:
21.             # jika Token bigram, masukkan ke document.
22.             text_list[idx].append(token)
23.     for token in trigram[text_list[idx]]:
24.         if '_' in token:
25.             text_list[idx].append(token)
26.
27. from gensim import corpora, models
28. #buat dictionary
29. dictionary = corpora.Dictionary(text_list)
30. dictionary.filter_extremes(no_below=5, no_above=0.2)
31.
32. #buat corpus
33. doc_term_matrix = [dictionary.doc2bow(doc) for doc in
text_list]
34.
35. #buat TF-IDF model
36. tfidf = models.TfidfModel(doc_term_matrix)
37. corpus_tfidf = tfidf[doc_term_matrix]
38.

```

```

39. from gensim.models.coherencemodel import CoherenceModel
40. from gensim.models.ldamodel import LdaModel
41. from gensim.corpora.dictionary import Dictionary
42. from numpy import array
43.
44. from multiprocessing import Process, freeze_support
45.
46. #menghitung nilai coherence
47. def compute_coherence_values(dictionary, corpus, texts, limit,
    start, step):
48.     coherence_values = []
49.     model_list = []
50.     for num_topics in range(start, limit, step):
51.         model = LdaModel(corpus=corpus, id2word=dictionary,
    num_topics=num_topics, iterations=100)
52.         model_list.append(model)
53.         coherencemodel = CoherenceModel(model=model,
    texts=texts, dictionary=dictionary, coherence='c_v')
54.         coherence_values.append(coherencemodel.get_coherence())
55.     return model_list, coherence_values
56.
57.
58. start=1
59. limit=20
60. step=1
61. model_list, coherence_values =
    compute_coherence_values(dictionary, corpus=corpus_tfidf,
    texts=text_list, start=start, limit=limit, step=step)
62. x = range(start, limit, step)
63.
64. for m, cv in zip(x, coherence_values):
65.     print("Num Topics =", m, " has Coherence Value of",
    round(cv, 6))
66.
67. #buat diagram plot
68. import matplotlib.pyplot as plt
69.
70. plt.plot(x, coherence_values)
71. plt.xlabel("Num Topics")
72. plt.ylabel("Coherence score")
73. plt.legend(("coherence_values"), loc='best')
74. plt.show()
75.
76. model = LdaModel(corpus=corpus_tfidf, id2word=dictionary,
    num_topics=6)
77. for idx, topic in model.print_topics(-1):
78.     print('Topic: {} Word: {}'.format(idx, topic))
79.
80. top_words_per_topic = []
81. for t in range(model.num_topics):
82.     top_words_per_topic.extend([(t, ) + x for x in
    model.show_topic(t, topn = 10)])
83.
84. df = pd.DataFrame(top_words_per_topic, columns=['Topic',
    'Word', 'P']).to_csv("top_words_positif.csv")
85. print(df)
86.
87. data = pyLDAvis.gensim.prepare(model, corpus_tfidf, dictionary)
88. print(data)
89. pyLDAvis.save_html(data, '../dataset/pyLDAvis_negatif.html')
90.

```

```

91. from wordcloud import WordCloud as wd
92. for t in range(model.num_topics):
93.     plt.figure(figsize=(7,6))
94.     plt.imshow(wd(max_font_size=50,
min_font_size=6).fit_words(dict(model.show_topic(t, 200))))
95.     plt.axis("off")
96.     plt.title("Topic #" + str(t))
97.     plt.savefig("../dataset/neg-wcld-topic-#" + str(t) + ".png",
facecolor='k', bbox_inches='tight')
98. plt.show()

```

Hasil pemodelan topik LDA berupa ekstraksi topik pada dataset *headline* berita bersentimen positif ditunjukkan pada Tabel 4.2, ekstraksi topik pada dataset *headline* berita bersentimen netral ditunjukkan pada Tabel 4.3, dan ekstraksi topik pada dataset *headline* berita bersentimen negatif ditunjukkan pada Tabel 4.4.

Tabel 4.2. Hasil Ekstraksi Topik Dataset *Headline* Bersentimen Positif

| Topic ID | Word Distribution |
|-----------------|---|
| Topic 0 | virus, gugus_tugas, warga, pandemi, penanganan, lawan, jokowi, indonesia, masker, dpr. |
| Topic 1 | gugus_tugas, pasien, pemerintah, terdampak, sembuh, virus, bantuan, psbb, rs, lawan. |
| Topic 2 | cegah_penyebaran, pandemi, indonesia, cegah, rumah, lawan, polri, sembuh, dki_jakarta, jubar. |
| Topic 3 | rapid_test, positif, virus, pandemi, pasien, pemerintah, pakai_masker, penanganan, rapid, sembuh. |
| Topic 4 | positif, tenaga_medis, pasien_positif, pemerintah, update, sembuh, jakarta, pandemi, pasien, rumah_sakit. |
| Topic 5 | wabah, positif, alat_tes, virus, pasien_positif, pemerintah, cegah, tenaga_medis, sri_mulyani, sembuh. |

Tabel 4.3. Hasil Ekstraksi Topik Dataset *Headline* Bersentimen Netral

| <i>Topic ID</i> | <i>Word Distribution</i> |
|-----------------|--|
| Topic 0 | pasien, as, cegah_penyebaran, dampak, virus, akibat, video, jakarta, menag, orang. |
| Topic 1 | wabah, virus, jokowi, rumah_sakit, imbas, tangani, sembuh, sembuh meninggal, anggaran, china. |
| Topic 2 | pandemi, meninggal, sembuh, indonesia, orang, pasien, virus, april, update, positif. |
| Topic 3 | pandemi, positif, kota, virus, sembuh, rumah, ribu, daftar, indonesia, april. |
| Topic 4 | positif, gugus_tugas, pasien, bantuan, warga, terdampak, virus, online, sembuh, menteri. |
| Topic 5 | pemerintah, pandemi, meninggal_dunia, tenaga_medis, pasien, anak, dunia, cerita, nasib, darurat. |

Tabel 4.4. Hasil Ekstraksi Topik Dataset *Headline* Bersentimen Negatif

| <i>Topic ID</i> | <i>Word Distribution</i> |
|-----------------|--|
| Topic 0 | dinyatakan_sembuh, wabah, as, kena_phk, warga, april, bertambah, pdp, phk, pandemi. |
| Topic 1 | update, april, meninggal_dunia, dunia, pm_inggris, video, jateng, angka kematian, pandemi, akibat. |
| Topic 2 | indonesia, meninggal_dunia, update, akibat, tenaga_medis, warga, april, tembus, odp, as. |
| Topic 3 | april, bertambah, akibat, dokter, pandemi, indonesia, korsel, rs, update, pdp. |
| Topic 4 | jatim, indonesia, wabah, tenaga_medis, update, rumah_sakit, pdp, umumkan, korban, ribu. |
| Topic 5 | jakarta, ribu, bertambah, kena_phk, tenaga_medis, juta, angka_kematian, sumbar, pandemi, update. |

Dengan memperhatikan Tabel 4.2, Tabel 4.3, dan Tabel 4.4, topik-topik yang diperoleh tampak mengalami bias, sebagai contoh pada hasil ekstraksi topik bersentimen positif, term “positif” dan “pasien_positif” idealnya adalah bersentimen negatif namun term ini muncul di topik bersentimen positif dan tidak muncul di hasil ekstraksi topik bersentimen negatif. Hal yang demikian ini disebabkan karena model LSTM masih memiliki tingkat akurasi yang kurang tinggi dan mengalami *overfitting*.

4.1.5 Visualisasi

Hasil ekstraksi topik-topik dari proses pemodelan topik selanjutnya akan divisualisasikan dalam bentuk *wordcloud* dan *intertopic distance map*. Visualisasi *Wordcloud* digunakan untuk mempermudah memperoleh *insight* dari topik-topik yang dihasilkan, dan *intertopic distance map* digunakan untuk mengetahui jarak antar topik dan keterkaitan satu topik dengan topik lainnya.

a. Visualisasi topik pada hasil analisis sentimen positif

Visualisasi *wordcloud* untuk topik *headline* bersentimen positif disajikan pada gambar 4.94, gambar 4.95, gambar 4.96, gambar 4.97, gambar 4.98, dan gambar 4.99. Sedangkan visualisasi *intertopic distance map* disajikan pada gambar 4.100.



Gambar 4.95 *Wordcloud* Topic #0 Sentimen Positif



Gambar 4.98 *Wordcloud* Topic #3 Sentimen Positif



Gambar 4.96 *Wordcloud* Topic #1 Sentimen Positif



Gambar 4.99 *Wordcloud* Topic #4 Sentimen Positif



Gambar 4.97 *Wordcloud* Topic #2 Sentimen Positif



Gambar 4.100 *Wordcloud* Topic #5 Sentimen Positif

Insight yang dapat diperoleh dari visualisasi *wordcloud* topik *headline* bersentimen positif ditunjukkan pada Tabel 4.5.



Gambar 4.102 Wordcloud Topic #0
Sentimen Netral



Gambar 4.105 Wordcloud Topic #3
Sentimen Netral



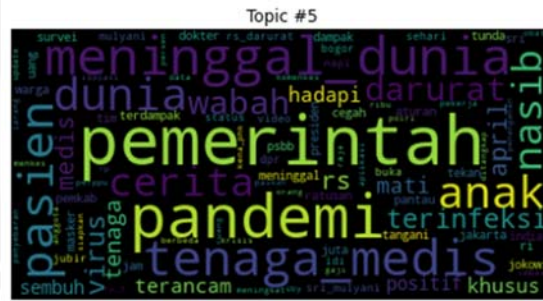
Gambar 4.103 Wordcloud Topic #1
Sentimen Netral



Gambar 4.106 Wordcloud Topic #4
Sentimen Netral



Gambar 4.104 Wordcloud Topic #2
Sentimen Netral

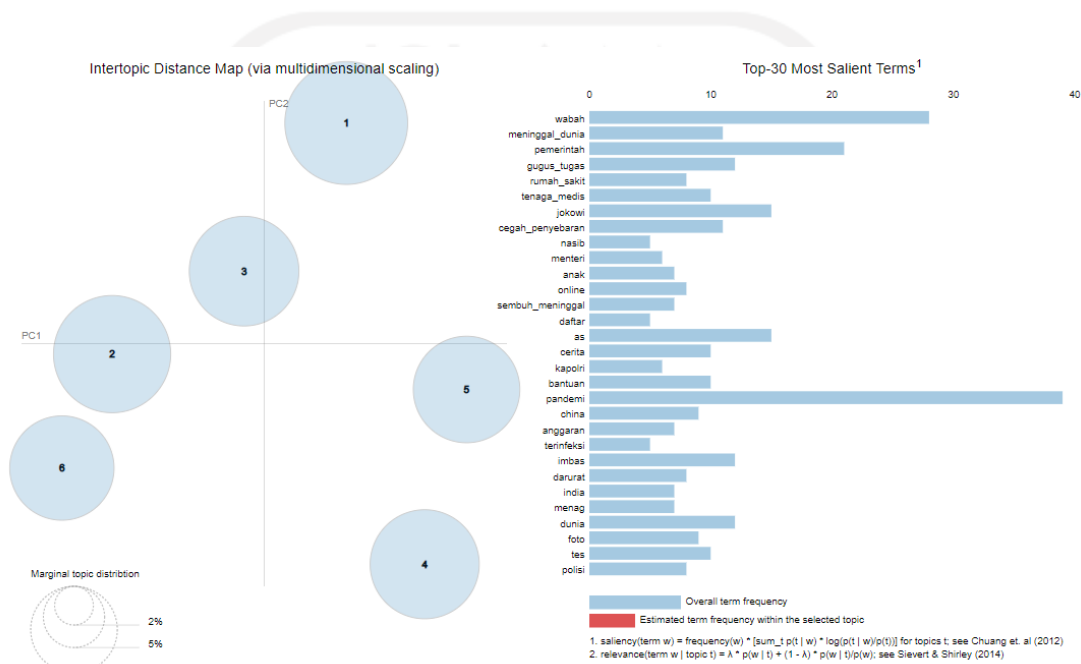


Gambar 4.107 Wordcloud Topic #5
Sentimen Netral

Insight yang dapat diperoleh dari visualisasi *wordcloud* topik *headline* bersentimen netral ditunjukkan pada Tabel 4.6.

Tabel 4.6. *Insight* Hasil Ekstraksi Topik pada Dataset *Headline* Bersentimen Netral

| <i>Topic ID</i> | <i>Insight</i> |
|-----------------|--|
| Topic 0 | Topik 0 berkaitan dengan berita pencegahan penyebaran virus. |
| Topic 1 | Topik 1 berkaitan dengan berita penanganan wabah virus. |
| Topic 2 | Topik 2 berkaitan dengan berita update data pasien meninggal dan sembuh bulan April. |
| Topic 3 | Topik 3 berkaitan dengan berita tentang kota-kota di Indonesia yang terkena pandemi virus. |
| Topic 4 | Topik 4 berkaitan dengan berita upaya gugus tugas di dalam mencegah dampak virus. |
| Topic 5 | Topik 5 berkaitan dengan berita pemerintah hadapi pandemi |



Gambar 4.108 *Intertopic Distance Map* Topik-topik *Headline* Bersentimen Netral

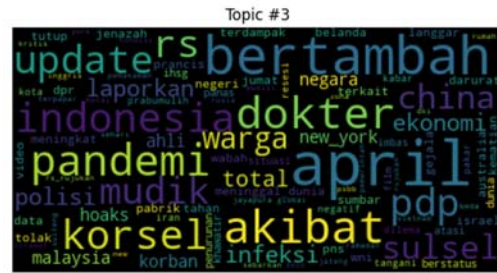
Berdasarkan *Intertopic Distance Map*, secara global diketahui tidak ada topik-topik yang beririsan. Topic #5 dan Topic #1 memiliki jarak yang relatif dekat, Topic #0 dan Topic #2 memiliki jarak yang relatif dekat namun berbeda radian. Topic #3 dan Topic #4 tidak terlalu dekat tetapi masih dalam radian yang sama.

c. Visualisasi topik pada hasil analisis sentimen negatif

Visualisasi *wordcloud* untuk topik *headline* bersentimen negatif disajikan pada gambar 4.108, gambar 4.109, gambar 4.110, gambar 4.111, gambar 4.112, dan gambar 4.113. Sedangkan visualisasi *intertopic distance map* disajikan pada gambar 4.114.



Gambar 4.109 *Wordcloud* Topic #0
Sentimen Negatif



Gambar 4.112 *Wordcloud* Topic #3
Sentimen Negatif



Gambar 4.110 *Wordcloud* Topic #1
Sentimen Negatif



Gambar 4.113 *Wordcloud* Topic #4
Sentimen Negatif



Gambar 4.111 *Wordcloud* Topic #2
Sentimen Negatif

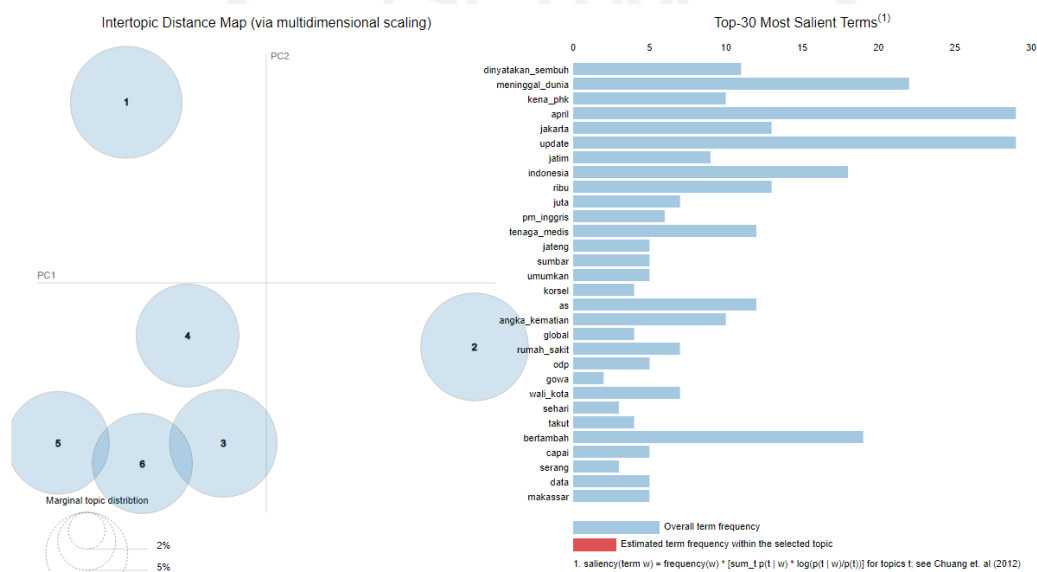


Gambar 4.114 *Wordcloud* Topic #5
Sentimen Negatif

Insight yang dapat diperoleh dari visualisasi *wordcloud* topik *headline* bersentimen negatif ditunjukkan pada Tabel 4.7.

Tabel 4.7. *Insight* Hasil Ekstraksi Topik pada Dataset *Headline* Bersentimen Negatif

| <i>Topic ID</i> | <i>Insight</i> |
|-----------------|--|
| Topic 0 | Topik 0 berkaitan dengan berita tentang karyawan terkena phk. |
| Topic 1 | Topik 1 berkaitan dengan berita update data meninggal dunia akibat virus covid-19. |
| Topic 2 | Topik 2 berkaitan dengan berita update tenaga medis meninggal akibat virus. |
| Topic 3 | Topik 3 berkaitan dengan berita tentang pdp yang bertambah secara nasional di Indonesia. |
| Topic 4 | Topik 4 berkaitan dengan berita tentang pdp di Jatim. |
| Topic 5 | Topik 5 berkaitan dengan berita tentang pekerja yang terkena phk di Jakarta |



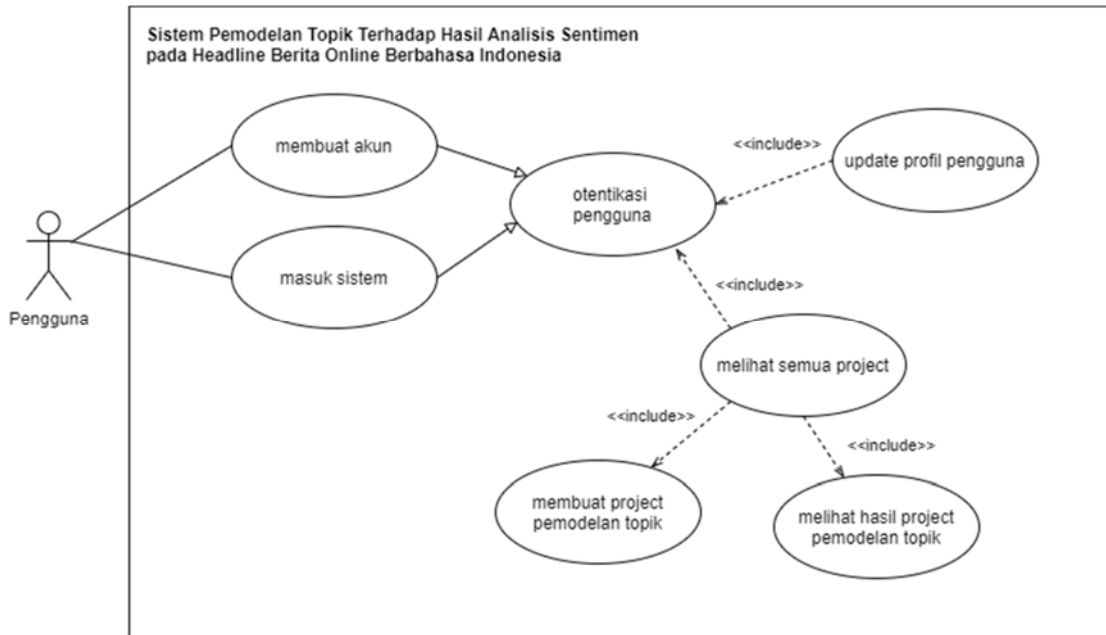
Gambar 4.115 *Intertopic Distance Map* Topik-topik *Headline* Bersentimen Negatif

Berdasarkan *Intertopic Distance Map*, Topic #3, Topic #4, Topic #5, Topic #6 berada dalam *cluster* yang sama, sedangkan Topic #1 dan Topic #2 saling berjauhan dan menyendiri.

4.2 *Prototype* Aplikasi Berbasis Web

Model LSTM dan LDA yang dihasilkan pada penelitian ini diaplikasikan ke sebuah *prototype* aplikasi berbasis web yang memungkinkan bagi para pengguna aplikasi untuk melakukan pemodelan topik terhadap suatu *headline* berita dengan menggunakan *keyword* tertentu dan rentang tanggal tertentu.

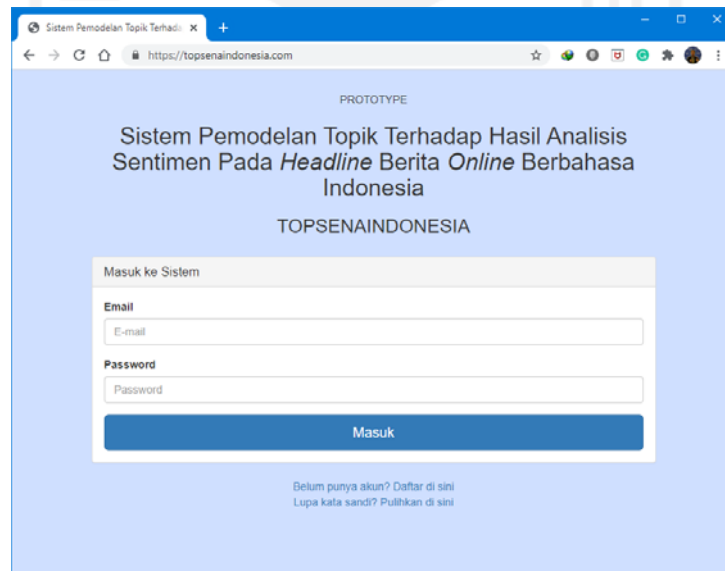
Prototype aplikasi berbasis web ini penulis unggah di server VPS dan dapat diakses menggunakan alamat <https://topsenaindonesia.com>. *Use case diagram* *prototype* aplikasi ini ditunjukkan pada gambar 4.116.



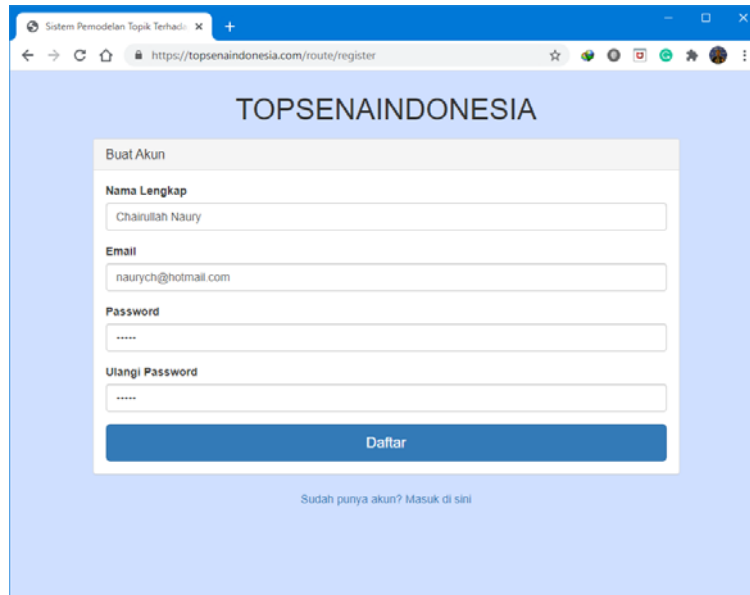
Gambar 4.116 Use Case Diagram untuk Prototype Aplikasi topsenaindonesia.com

4.2.1 Membuat Akun Pengguna

Pengguna dapat membuat akun untuk mengakses sistem dengan melakukan klik pada link bertuliskan “Belum punya akun? Daftar di sini” seperti ditunjukkan pada gambar 4.117 dan selanjutnya mengisi form “Buat Akun” seperti ditunjukkan pada gambar 4.118.



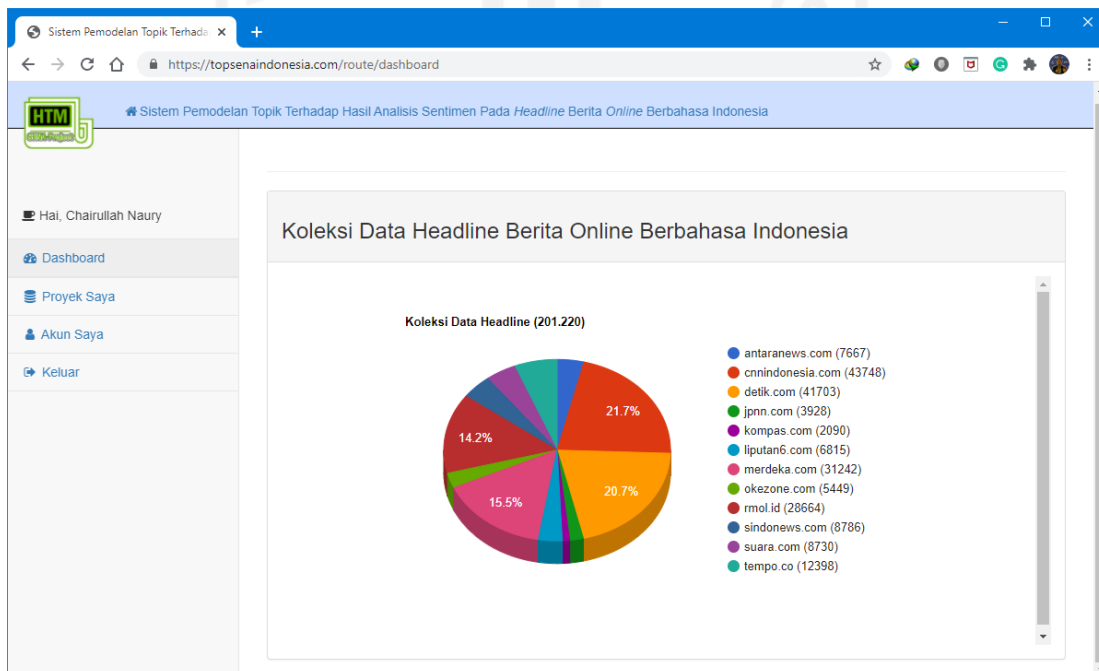
Gambar 4.117 Tampilan Awal Sistem



Gambar 4.118 Antarmuka Pembuatan Akun Pengguna

4.2.2 Masuk Sistem

Pengguna dapat masuk ke sistem melalui halaman awal sistem seperti tampak pada gambar 4.116 dengan memasukkan alamat email dan password akun pengguna topsenainonesia.com. Apabila otentikasi kredensial pengguna berhasil, maka pengguna akan diantarkan ke halaman dashboard sistem milik pengguna seperti ditunjukkan pada gambar 4.119.



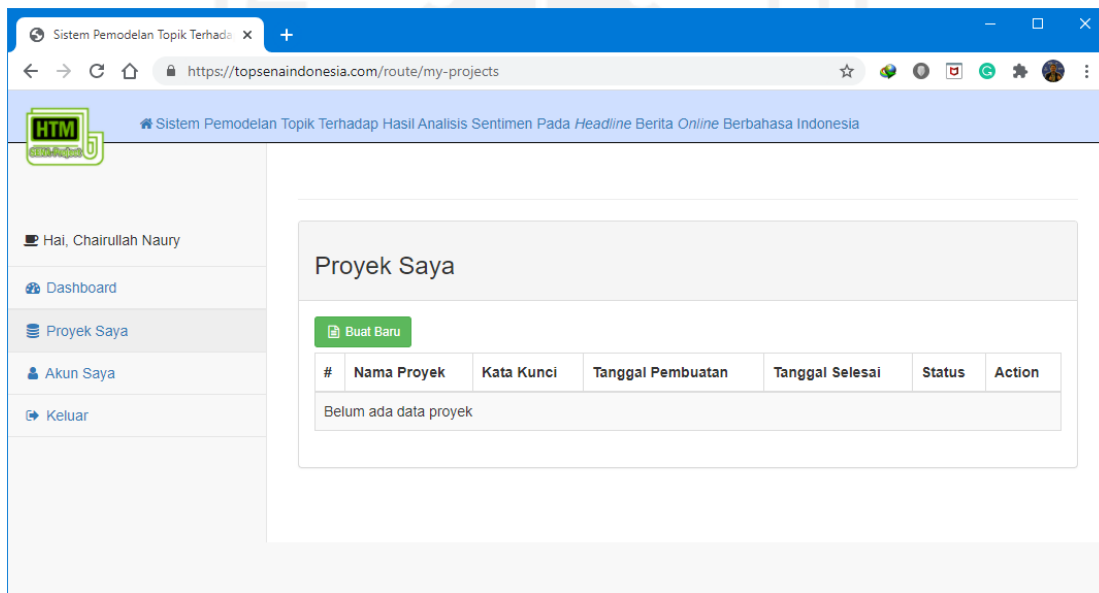
Gambar 4.119 Dashboard Sistem

Halaman *dashboard* sistem ini memiliki 4 menu yang terletak pada *sidebar* kiri, yaitu:

- a. Menu “*Dashboard*” yang berfungsi sebagai halaman beranda pengguna.
- b. Menu “*Proyek Saya*” yang berfungsi untuk melihat proyek pemodelan topik, membuat proyek pemodelan topik, melihat hasil proyek pemodelan topik, dan menghapus proyek pemodelan topik.
- c. Menu “*Akun Saya*” yang berfungsi untuk mengubah profil pengguna.
- d. Menu “*Keluar*” yang berfungsi untuk keluar dari sistem.

4.2.3 Melihat Semua *Project*

Pengguna dapat melihat semua *project* pemodelan topik yang sudah pernah dibuat sebelumnya dengan mengakses menu “*Proyek Saya*” pada *dashboard*. Apabila belum pernah membuat *project*, maka pada halaman ini tertera informasi “*Belum ada data proyek*” seperti yang ditunjukkan pada Gambar 4.120.



Gambar 4.120 Antarmuka Halaman "Proyek Saya"

4.2.4 Membuat *Project* Pemodelan Topik

Pengguna dapat membuat *project* pemodelan topik dengan mengakses menu “*Proyek Saya*” dan klik pada tombol “*Buat Baru*”, kemudian mengisi form “*Membuat Proyek*” seperti ditunjukkan pada gambar 4.121.

Sistem Pemodelan Topik Terhadap Hasil Analisis Sentimen Pada *Headline* Berita *Online* Berbahasa Indonesia

Hai, Chairullah Naury

Dashboard

Proyek Saya

Akun Saya

Keluar

Membuat Proyek

Nama Proyek

Topic Modelling Kasus Jiwasraya

Deskripsi

Mencari insight dari berita yang berkaitan dengan topik kasus Jiwasraya

Kata Kunci^{*)}

jiwasraya

*) Jika kata kunci lebih dari 1, gunakan pemisah semicolon (;) untuk masing-masing kata kunci

Periode Berita: Semua Periode Rentang Tanggal

Save Cancel

Gambar 4.121 Antarmuka Form "Membuat Proyek"

Setelah mengisi semua isian, pengguna klik pada tombol “Save” dan pengguna akan diantarkan ke halaman “Proyek Saya” yang sudah tercantum daftar proyek yang baru saja dibuat seperti tampak pada gambar 4.122.

Sistem Pemodelan Topik Terhadap Hasil Analisis Sentimen Pada *Headline* Berita *Online* Berbahasa Indonesia

Hai, Chairullah Naury

Dashboard

Proyek Saya

Akun Saya

Keluar

Proyek Saya

Buat Baru

| # | Nama Proyek | Kata Kunci | Tanggal Pembuatan | Tanggal Selesai | Status | Action |
|---|---|------------|---------------------|-----------------|----------|--------|
| 1 | Topic Modelling Kasus Jiwasraya Mencari insight dari berita yang berkaitan dengan topik kasus Jiwasraya Periode Data: Tanpa Periode | jiwasraya | 2020-11-21 17:49:43 | | Menunggu | Delete |

Gambar 4.122 List Proyek Pemodelan Topik

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Analisis sentimen yang dilaksanakan pada penelitian ini menggunakan metode *Long Short-term Memory* (LSTM) dengan memanfaatkan *pre-trained global vector for word representation* berbahasa Indonesia yang diperoleh dari *fasttext* menghasilkan model LSTM dengan berbagai tingkat akurasi dan semua model mengalami *overfitted* karena akurasi validasi lebih kecil dari akurasi pelatihan. Dengan mempertimbangkan nilai *overfitting* yang kecil dan prosentase akurasi diatas 70% diketahui model dengan nilai akurasi yang dipilih adalah model dengan nilai training accuracy 76.64%, validation accuracy 70.36%, dan *overfitted* 6.28%. Kecilnya akurasi yang diperoleh pada model LSTM menyebabkan terjadinya bias pada hasil Topic Modelling. Pada setiap eksperimen *hyperparameters tuning*, proses ini menghasilkan nilai akurasi yang berbeda-beda sesuai dengan parameter-parameter yang dipasang, yaitu jumlah lapisan LSTM, jumlah *recurrent units*, *dropouts*, ukuran *batch*, dan maksimum *epoch*. Pemodelan topik pada penelitian ini menggunakan metode *Latent Dirichlet Allocation* (LDA) dapat menghasilkan topik-topik yang dibicarakan oleh media *online*, baik itu topik-topik yang bersentimen positif, negatif, maupun netral. Topik-topik yang diperoleh dari pemodelan topik terhadap hasil analisis sentimen pada tajuk berita *online* berbahasa Indonesia ini mengalami bias.

Model yang dihasilkan dapat diterapkan pada *prototype* aplikasi pemodelan topik berbasis web. Kinerja *prototype* ditinjau dari sudut pandang fungsional dapat menyajikan hasil pemodelan topik terhadap hasil sentimen analisis berupa visualisasi *Wordcloud* dan *Intertopic Distance Map* dari masing-masing topik-topik bersentimen positif, netral, dan negatif.

5.2 Saran

Model LSTM yang dihasilkan dapat ditingkatkan lagi akurasinya dengan mengimplementasikan lebih banyak skenario *hyperparameters tuning* dengan metode *trial-and-error* yaitu mengubah-ubah *hyperparameter* jumlah *layer* LSTM, *units*, dimensi *embedding layer*, ukuran *batch*, dan maksimum *epoch* sampai diperoleh model yang optimal.

Penelitian ini membuka peluang *future research* berupa penerapan *hyperparameters tuning* dengan metode *Grid Search* dan *Random Search* untuk mendapatkan nilai *hyperparameter* yang optimal agar dapat memperoleh model dengan tingkat akurasi yang lebih baik.

Prototype aplikasi pemodelan topik dapat dikembangkan lagi menjadi aplikasi jadi siap pakai dengan menambahkan fitur-fitur lain seperti menampilkan hasil pemodelan topik yang dapat difilter berdasarkan media berita *online* tertentu, misalnya detik.com membicarakan topik a, b, dan c terkait suatu isu berita, namun media yang lain misalnya tempo.co bisa jadi membicarakan topik yang berbeda x, y, dan z meskipun dengan isu yang sama.



Daftar Pustaka

- Abuhay, T. M., Nigatie, Y. G., & Kovalchuk, S. V. (2018). Towards Predicting Trend of Scientific Research Topics using Topic Modeling. *Procedia Computer Science*, 136, 304–310. <https://doi.org/10.1016/j.procs.2018.08.284>
- Aggarwal, C. C., & Zhai, C. X. (2013). Mining text data. In *Mining Text Data* (Vol. 9781461432). <https://doi.org/10.1007/978-1-4614-3223-4>
- Ankit, & Saleena, N. (2018). An Ensemble Classification System for Twitter Sentiment Analysis. *Procedia Computer Science*, 132(Iccids), 937–946. <https://doi.org/10.1016/j.procs.2018.05.109>
- Badan Pengembangan Bahasa dan Perbukuan. (2016). *Kamus Besar Bahasa Indonesia*. Kementerian Pendidikan Dan Kebudayaan. <https://kbbi.kemdikbud.go.id/>
- Bergamaschi, S., & Po, L. (2015). Comparing LDA and LSA topic models for content-based movie recommendation systems. *Lecture Notes in Business Information Processing*, 226, 247–263. https://doi.org/10.1007/978-3-319-27030-2_16
- Blei, D., Carin, L., & Dunson, D. (2010). Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6), 55–65. <https://doi.org/10.1109/MSP.2010.938079>
- Blei, D. M., Edu, B. B., Ng, A. Y., Edu, A. S., Jordan, M. I., & Edu, J. B. (2003). Latent Dirichlet Allocation. *CrossRef Listing of Deleted DOIs*, 1, 993–1022. <https://doi.org/10.1162/jmlr.2003.3.4-5.993>
- Chen, G. (2016). *A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation*. 1–10. <http://arxiv.org/abs/1610.02583>
- Fitri, V. A., Andreswari, R., & Hasibuan, M. A. (2019). Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm. *Procedia Computer Science*, 161, 765–772. <https://doi.org/10.1016/j.procs.2019.11.181>
- Hagen, L. (2018). Content analysis of e-petitions with topic modeling: How to train and evaluate LDA models? *Information Processing and Management*, 54(6), 1292–1307. <https://doi.org/10.1016/j.ipm.2018.05.006>
- Hidayatullah, A. F., Hakim, A. M., Cahyaningtyas, S., & Aulia, W. P. (n.d.). *TOPIC MODELING DATA TWITTER TERHADAP CALON PRESIDEN REPUBLIK INDONESIA 2019 MENGGUNAKAN METODE LATENT DIRICHLET ALLOCATION (LDA)*.

- Hidayatullah, A. F., Pembrani, E. C., Kurniawan, W., Akbar, G., & Pranata, R. (2018). Twitter Topic Modeling on Football News. *2018 3rd International Conference on Computer and Communication Systems, ICCCS 2018*, 94–98.
<https://doi.org/10.1109/CCOMS.2018.8463231>
- Ivanedra, K., & Mustikasari, M. (2019). *Implementasi Metode Recurrent Neural Network Pada Text Summarization Dengan Teknik Abstraktif the Implementation of Text Summarization With Abstractive Techniques Using Recurrent Neural Network Method*. 6(4), 377–382. <https://doi.org/10.25126/jtiik.201961067>
- Jacobi, C., Van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, 4(1), 89–106.
<https://doi.org/10.1080/21670811.2015.1093271>
- Lai, C. C., Shih, T. P., Ko, W. C., Tang, H. J., & Hsueh, P. R. (2020). Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges. *International Journal of Antimicrobial Agents*, 55(3), 105924.
<https://doi.org/10.1016/j.ijantimicag.2020.105924>
- Lu, Y., Mei, Q., & Zhai, C. X. (2011). Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA. *Information Retrieval*, 14(2), 178–203. <https://doi.org/10.1007/s10791-010-9141-9>
- Luketina, J., Berglund, M., Greff, K., & Raiko, T. (2016). Scalable gradient-based tuning of continuous regularization hyperparameters. *33rd International Conference on Machine Learning, ICML 2016*, 6, 4333–4341.
- Made Kusnanta Bramantya Putra, I. (2017). Analisis Topik Informasi Publik Media Sosial Di Surabaya Menggunakan Pemodelan Latent Dirichlet Allocation (Lda) Topic Analysis of Public Information in Social Media in Surabaya Based on Latent Dirichlet Allocation (Lda) Topic Modelling. *Jurnal Teknik Its*, 6(2), 2–7.
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093–1113.
<https://doi.org/10.1016/j.asej.2014.04.011>
- Nomleni, P., Hariadi, M., & Purnama, K. E. (n.d.). Sentiment Analysis Berbasis Big Data Sentiment Analysis Based Big Data. *Seminar Nasional Ke, 9*, 142.
- Ogedebe, P. M., & Jacob, B. P. (2012). Software Prototyping: A Strategy to Use When User Lacks Data Processing Experience. *ARPN Journal of Systems and Software*,

2(6), 219–224.

- Pasupa, K., & Seneewong Na Ayutthaya, T. (2019). Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features. *Sustainable Cities and Society*, 50(May), 101615. <https://doi.org/10.1016/j.scs.2019.101615>
- Pratiwi, M. R. (2014). Peran ICT bagi Organisasi Media Massa dan Budaya Masyarakat. *Komunikator*, 6(5), 21. <http://journal.umy.ac.id/index.php/jkm/article/view/212/174>
- Primartha, R. (2018). *Belajar Machine Learning Teori dan Praktik*. Penerbit INFORMATIKA.
- Probst, P., Boulesteix, A. L., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20, 1–22.
- Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.
- Reimers, N., & Gurevych, I. (2017). *Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks*. <http://arxiv.org/abs/1707.06799>
- Srivastava, A., Singh, V., & Drall, G. S. (2019). Sentiment Analysis of Twitter Data. *International Journal of Healthcare Information Systems and Informatics*, 14(2), 1–16. <https://doi.org/10.4018/ijhisi.2019040101>
- Suyudi; Djamel Emeranda C; Maspupah Asri. (2019). Prediksi Harga Saham menggunakan Metode Recurrent Neural Network. *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, A-33-A-38.
- Wandik, M., M.Golung, A., & H.Mulyono. (2017). Proses Penentuan Headline Surat Kabar (Studi Pada Surat Kabar Harian Manado Post). *E-Journal Acta Diurna*, VI(2).
- Wang, H., Zhang, D., & Zhai, C. X. (2011). Structural topic model for latent topical structure analysis. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1, 1526–1535.
- Wangsa, B. K., Utomo, D., & Nugroho, S. (2014). Sistem Peringkat Berita Otomatis berbasis Text Mining menggunakan Generalized Vector Space Model: Studi Kasus Berita diambil dari Media Massa Online. *Techne Jurnal Ilmiah Elektroteknika*, 13, 231–241.

- WHO. (2020). Coronavirus disease 2019. *World Health Organization*, 2019(March), 2633.
<https://doi.org/10.1001/jama.2020.2633>
- Winiharti, M. (2011). Analisis Diksi Pada Judul Berita Utama Surat Kabar yang Memberitakan Rapat Pansus DPR RI untuk Kasus Bank Century. *Jurnal Penelitian Humaniora*, 12(1), 19–31.
- Yuniati, Y. (2002). *Pengaruh Berita di Surat Kabar terhadap Persepsi Mahasiswa tentang Politik* (Vol. 3, Issue 1, pp. 1–17).
- Zulhanif. (2016). Pemodelan Topik Dengan Latent Dirichlet Allocation. *Seminar Nasional Pendidikan Matematika*, 1–8.

