**ANALISIS SENTIMEN PADA SOSIAL MEDIA X TERHADAP GENOSIDA ISRAEL KEPADA PALESTINA MENGGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)**

**SKRIPSI**

diajukan untuk menempuh ujian sarjan

pada Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Padjadjaran

WAFA TSABITA

NPM 140810200055



UNIVERSITAS PADJADJARAN  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM STUDI TEKNIK INFORMATIKA  
SUMEDANG  
2025

**SKRIPSI**

**ANALISIS SENTIMEN PADA SOSIAL MEDIA X TERHADAP GENOSIDA ISRAEL KEPADA PALESTINA MENGGUNAKAN METODE *LONG SHORT-TERM MEMEORY* (LSTM)**

***SENTIMEN ANALYSIS ON SOCIAL MEDIA X TOWARDS ISRAELS GENOCIDE AGAINST PALESTINE USING LONG SHORT-TERM MEMORY (LSTM) METHOD***

Telah dipersiapkan dan disusun oleh

WAFA TSABITA

NPM 140810200055

Telah dipertahankan di depan Tim Penguji

pada tanggal …………….

Susunan Tim Penguji

1. Dr. Juli Rejito, M.Kom. Ketua Tim Penguji

NIP. 19700615 199803 1 003 ……………………….

1. ……………………………… Pembimbing

NIP. ………………………... ……………………….

1. ……………………………… Co-Pembimbing

NIP. ………………………... ……………………….

1. ……………………………… Penguji

NIP. ………………………... ……………………….

1. ……………………………… Penguji

NIP. ………………………... ……………………….

1. ……………………………… Penguji

NIP. ………………………... ……………………….

**KATA PENGANTAR**

*Bismillahirahmanirrahim*. Dengan menyebut nama Allah yang Maha Pengasih dan Maha Penyayang. Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “ANALISIS SENTIMEN PADA SOSIAL MEDIA X TERHADAP GENOSIDA ISRAEL KEPADA PALESTINA MENGGUNAKAN METODE *LONG SHORT-TERM MEMEORY* (LSTM)” sebagai salah satu syarat menempuh sarjana pada Program Studi S-1 Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

Dalam proses penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis mengucapkan terima kasih kepada Ibu Afrida Helen , sebagai pembimbing utama, Ibu Mira Suryani, sebagai pembimbing pendamping yang telah meluangkan waktu dan pikirannya sehingga penulis dapat menyelesaikan skripsi ini. Ucapan terima kasih juga diberikan kepada keluarga penulis yang selalu memberikan motivasi dan doa yang menjadi pendorong dalam penyelesaian skripsi ini. Penulis juga mengucapkan terima kasih sebanyak-banyaknya kepada:

Prof. Dr. Iman Rahayu, S.Si., M.Si, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

Dr. Setiawan Hadi, M.Sc.CS, selaku Kepala Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

Dr. Akmal, S.Si., M.T., selaku Ketua Program Studi S-1 Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran

Dosen-dosen Teknik Informatika Unpad yang telah mengajar dan memberikan ilmu kepada penulis selama masa perkuliahan yang membawa penulis pada posisi sekarang ini.

Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi pembaca.

Jatinangor, Juli 2025

Penulis

ABSTRAK

Di era digital ini, banyak universitas yang menyediakan web repositori dimana mahasiswa dapat mengakses berkas digital dari skripsi atau karya ilmiah milik *civitas academica*. Digitalisasi ini tentunya memudahkan mahasiswa dalam memperoleh referensi dan sumber bacaan tanpa perlu mengunjungi perpustakaan. Pada saat ini, program studi Teknik Informatika Unpad belum memiliki repositori maupun sarana serupa. Hal inilah yang menjadi motivasi dasar pada penelitian ini untuk membangun repositori skripsi Teknik Informatika Unpad. Optimalnya, sebuah web harus responsif terhadap *smartphone* dan memiliki pengalaman pengguna yang baik.

Metoda penelitian yang digunakan dalam penelitian ini memanfaatkan *Extreme Programming* dengan model pengembangan perangkat lunak *Agile* sehingga bersifat gesit dan iteratif. Metoda *Extreme Programming* memiliki 6 fase yaitu fase eksplorasi, fase perencanaan, fase iterasi untuk dirilis, fase produksi, fase pemeliharaan, dan fase akhir. Dengan siklus singkat dan berfokus pada permintaan pengguna, aplikasi terus mendapat *feedback* sehingga meningkatkan kualitas aplikasi.

Dari hasil penelitian didapat nilai *usability testing* sebesar 88.53% dan *heuristic evaluation* sebesar 72%. Dari hasil tersebut dapat disimpulkan bahwa sistem repositori yang dibangun dengan keunggulan PWA dan metode *Extreme Programming* berjalan dengan baik dan memiliki nilai kepuasan yang baik pula.

**Kata Kunci**: ExpressJs*, Extreme Programming*, PWA, ReactJs, Repositori

*ABSTRACT*

*In this digital era, many universities provide a web repository where students can access digital files from theses or scientific works belonging to the academic community. This digitization certainly makes it easier for students to obtain references and reading sources without the need to visit the library. At this time, Unpad Informatics Engineering study program does not yet have a repository or similar facilities. This is the basic motivation for this research to build a thesis repository for Informatics Engineering Unpad. Optimally, a web should be responsive to smartphones and have a good user experience.*

*The research method used in this study utilizes Extreme Programming with the Agile software development model so that it is agile and iterative. The Extreme Programming method has 6 phases, namely the exploration phase, the planning phase, the iteration phase for release, the production phase, the maintenance phase, and the final phase. With a short cycle and focused on user requests, the application continues to receive feedback, thereby improving application quality.*

*The results showed that the usability testing value was 88.53% and the heuristic evaluation was 72%. From these results it can be concluded that the repository system built with the advantages of PWA and the Extreme Programming method runs well and has a good satisfaction score as well.*

***Keywords:*** *ExpressJs, Extreme Programming, PWA, ReactJs, Repository*

DAFTAR TABEL

[Tabel ‎2.1 Simbol-simbol pada *Use Case Diagram* 10](#_Toc203550518)

[Tabel ‎2.2 Simbol-Simbol pada *Activity Diagram* 12](#_Toc203550519)

[Tabel ‎4.1 *Black Box Testing* pada Fitur Registrasi 23](#_Toc203550520)

DAFTAR ISI

[ABSTRAK v](#_Toc203562575)

[*ABSTRACT* vi](#_Toc203562576)

[DAFTAR ISI vii](#_Toc203562577)

[DAFTAR TABEL x](#_Toc203562578)

[DAFTAR GAMBAR xi](#_Toc203562579)

[DAFTAR LAMPIRAN xiii](#_Toc203562580)

[BAB I 1](#_Toc203562581)

[1.1 Latar Belakang 1](#_Toc203562582)

[1.2 Identifikasi Masalah 2](#_Toc203562583)

[1.3 Batasan Masalah 3](#_Toc203562584)

[1.4 Maksud dan Tujuan Penelitian 4](#_Toc203562585)

[1.5 Manfaat Penelitian 4](#_Toc203562586)

[1.6 Metodologi Penelitian 4](#_Toc203562587)

[1.7 Sistematika Penulisan 5](#_Toc203562588)

[BAB II 7](#_Toc203562589)

[2.1 Analisis Sentimen 7](#_Toc203562590)

[2.2 Text Preprocessing 8](#_Toc203562591)

[2.3 *Long Short-Term Memory* 8](#_Toc203562592)

[2.4 *Unified Modeling Language* (UML) 9](#_Toc203562593)

[2.4.1 *Use Case Diagram* 10](#_Toc203562594)

[2.4.2 *Activity Diagram* 12](#_Toc203562595)

[BAB III 15](#_Toc203562596)

[3.1 Objek Penelitinan 15](#_Toc203562597)

[3.2 Tahapan Penelitian 15](#_Toc203562598)

[3.2.1 Pengumpulan Data 16](#_Toc203562599)

[3.2.2 Pelabelan Data 16](#_Toc203562600)

[3.2.3 Data Balancing 16](#_Toc203562601)

[3.2.4 Data Preprocessing 16](#_Toc203562602)

[3.2.5 Data Training, Validation dan Predictiing 16](#_Toc203562603)

[3.2.6 Lapisan Word Embedding (Word2vec) 16](#_Toc203562604)

[3.2.7 Lapisan Long Short-Term Memory 16](#_Toc203562605)

[3.2.8 Proses Pembuatan Model LSTM 16](#_Toc203562606)

[3.3 Perancangan Aplikasi 16](#_Toc203562607)

[3.3.1 Use Case Diagram 17](#_Toc203562608)

[3.3.2 Activity Diagram 19](#_Toc203562609)

[3.3.3 Sequence Diagram 19](#_Toc203562610)

[3.3.4 Deployment Diagram 19](#_Toc203562611)

[3.3.5 Perancangan Antarmuka 19](#_Toc203562612)

[3.4 Analisis Kebutuhan 19](#_Toc203562613)

[3.4.1 Kebutuhan Perangkat Lunak 19](#_Toc203562614)

[3.4.2 Kebutuhan Perangkat Keras 19](#_Toc203562615)

[BAB IV 21](#_Toc203562616)

[4.1 Analisis Pengujian *Hyperparameter* 21](#_Toc203562617)

[4.1.1 Word2vec 21](#_Toc203562618)

[4.1.2 Analsis Jumlah Lapisan LSTM 21](#_Toc203562619)

[4.1.3 Analsisi Unit LSTM 21](#_Toc203562620)

[4.1.4 Analsis *Optimizer* 21](#_Toc203562621)

[4.1.5 Analisis *Dropout* 21](#_Toc203562622)

[4.2 Implementasi Aplikasi 21](#_Toc203562623)

[BAB V 22](#_Toc203562624)

[5.1 Kesimpulan 22](#_Toc203562625)

[**5.1** **Saran** 22](#_Toc203562626)

[DAFTAR PUSTAKA 23](#_Toc203562627)

[LAMPIRAN 25](#_Toc203562628)

[RIWAYAT HIDUP 40](#_Toc203562629)

DAFTAR GAMBAR

[Gambar ‎2.1 Contoh *Use Case Diagram* (Suriya Sundramoothy, 2022) 11](#_Toc203550540)

[Gambar ‎2.2 Contoh *Activity Diagram* (Suriya Sundramoothy, 2022) 13](#_Toc203550541)

[Gambar ‎3.1 Alur Tahapan Penelitian 15](#_Toc203550542)

[Gambar ‎3.2 *Use Case Diagram* Aplikasi 17](#_Toc203550543)

[Gambar ‎4.1 Halaman Utama pada *Desktop* (a) dan *Mobile* (b) 21](#_Toc203550544)

# DAFTAR LAMPIRAN

[Lampiran 1 Ringkasan Hasil *Usability Testing* 130](#_Toc35650951)

[Lampiran 2 Kode Web Aplikasi 136](#_Toc35650952)

# BAB I

**PENDAHULUAN**

## Latar Belakang

Tindakan kekerasan penjajahan Isreal terhadap Palestina peningkatan secara signifikan sejak 7 Oktober 2023, penjajahan ini telah berlangsung selama kebih dari 77 tahun sejak tanggal 14 Mei 1948. Penjajahan ini diawali dengan perpecahan ideologi antara pemukim Israel dan Palestina, yang kemudian berkembang menjadi isu global dan geopolitik international. bahkan termasuk dalam ranah ekonomi dan kemanusiaan.

Serangan isreal pada tanggal 7 Oktober 2023 disebut sebagai Operasi Banjir Al-Aqsa, yang memicu eskalasi kekerasan terbesar dalam sejarah penjajahan Israel terhadap Palestina. Serangan dimulai dengan pengeboman pemukiman sipil di wilayah Gaza oleh Israel, yang kemudian dibalas oleh kelompok Hamas. Pada hari berikutnya, Israel meluncurkan lebih dari 2.000 bom ke wilayah Palestina dalam satu hari, dengan dalih membela diri.

Narasi “pembelaan diri” tersebut menyebar luas melalui media, Dimana banyak pemberitaan yang menyebutkan pejuang Palestina sebagai teroris sedangkan Isreal digambarkan sebagai pihak yang hanya membela diri setelah melakukan pengeboman tersebut. Kondisi ini mencerminkan fenomena yang disebut sebagai *Hasbara Project*, yaitu strategi propaganda digital yang dilakukan Israel untuk menyetir opini publik melalui media informasi (Jędrzejewska, 2020)

Media sosial mulai mengangkat isu penyerangan ini secara massif sejak serangan 7 oktober 2023. Pengguna sosial media secara global mulai melontarkan opini mereka terkait ‘konflik’ ini, ada yang mendukung bahkan membenarkan propaganda yang dibuat oleh isreal adapula yang menolak dan mengutuk keras atas kebohongan tersebut.

Media sosial menjadi wadah untuk mengungkapkan banyak hal, termasuk perihal yang mengganaskan yang terjadi di palestina. Terutama pada media sosial twitter atau yang disebut juga sebagai X. pengguna twitter lebih massif dalam menyeruakan keberpihakan serta lebih vokal dalam menyeruakan kebenaran. Oleh karena itu, untuk mengetahui perkembangan opini public terhadap aksi geosida ini pada sosial media tertuama pada X penelitian ini akan menggunakan *machine learning* dan model *long short term memory* (LSTM) untuk melakukan analisi terhadap perkembangan sentimen pengguna X sejak serangan 7 oktober 2023.

Dengan demikian, penelitian ini bertujuan untuk memahami perkembangan sentimen publik terhadap isu genosida Palestina di media sosial X sejak 7 Oktober 2023. Melalui pendekatan analisis sentimen berbasis LSTM, diharapkan hasil penelitian ini dapat memberikan kontribusi dalam memetakan opini global.

## Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, masalah yang akan dicari solusinya dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara kerja algoritma *Long Short-Term Memory* di dalam analisis sentimen?
2. Bagaimana mendapatkan kinerja *Long Short-Term Memory* yang optimal dalam mengklasifikasikan sentimen *tweet* terhadap isu genosida Israel terhadap Palestina?
3. Bagaimana mengembangkan aplikasi berbasis web yang dengan menggunakan untuk medisemasikan hasil klasifikasi sentimen *tweet* isu genosida Israel terhadap Palestina menggunakan *Unified Modelling Language*?

## Batasan Masalah

Dari identifikasi masalah tersebut, maka dalam penelitian ini terdapat beberapa batasan masalah pada sebagai berikut:

1. Mempelajari cara kerja algoritma *Long Short-Term Memory* untuk analisis sentimen *tweet* terhadap isu genosida Israel terhadap Palestina.
2. Memperoleh bentuk model *Long Short-Term Memory* yang optimal untuk mengklasifikasikan sentimen berdasarkan *hyperparameter* yang diuji.
3. *Tweet* yang digunakan adalah *tweet* yang berasal dari aplikasi media sosial X (Twitter) dengan kata kunci berupa ‘*genocide*’,’*gaza*’,’*israel*’, dan ’*palestine*’ dalam bahasa Inggris dengan data yang dikumpulkan dari tanggal 7 Oktober 2023 hingga tanggal 7 April 2025.
4. *Tweet* yang mengandung kalimat pendek atau hanya berisi hashtag saja tidak dimasukan dalam dataset.
5. Implementasi berbasis *Web* menggunakan Python dan HTML.

## Maksud dan Tujuan Penelitian

Berdasarkan identifikasi masalah, penelitian ini memiliki maksud untuk menganalisis sentimen dari opini pengguna medial sosial X (Twitter) terhadap isu genosida Israel kepada Palestina menggunakan *Long Short-Term Memory*.

Tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah:

1. Mempelajari cara kerja algoritma *Long Short-Term Momory* pada analisis sentimen.
2. Memperoleh model *Long Short-Term Memory* yang optimal untuk mengklasifikasikan sentimen *tweet* terhadap isu genosida Israel kepada Palestina.
3. Mengembangkan aplikasi berbasis *web* yang mendesiminasikan hasil klasifikasi sentimen *tweet* isu genosida Israel terhadap Palestina menggunakan *Unified Modelling Language*.

## Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah untuk mengetahui opini masyarakat terhadap isu genosida Israel kepada Palestina melalui hasil analisis sentimen. Penelitian ini juga dapat dimanfaatkan oleh pihak lain dalam penelitian lebih lanjut. Selain itu, penelitian ini dapat dimanfaatkan oleh pihak maupun komunitas Palestina sebagai preferensi dalam mengklasifikasi sentimen *tweet*.

## Metodologi Penelitian

Jenis penelitian yang akan digunakan ialah Kuantitatif Eksperimental dengan pendekatan *text mining* dan *machine learning,* khususnya *deep learning* menggunakan model *Long Short-Term Memory* (LSTM). Penelitian ini menggunakan metodologi CRISP-DM dengan rincian tahapan sebagai berikut:

* 1. *Business Understanding*, yaitu proses memahami tujuan dan persyaratan proyek dari prespektif bisnis. Proses ini terdiri dari menentukan tujuan, mengumpulkan persyaratan dan Menyusun rancangan proyek.
  2. *Data Understanding*, yaitu proses memahami data yang akan digunakan. Proses ini terdiri dari pengumpulan data, mempelajari data dan memastikan kualitas dari data yang telah dikumpulkan.
  3. *Data Preparation*, yaitu proses mempersiapkan data sebelum digunakan untuk membuat model. Proses ini terdiri dari mengeleksi atribut dan data yang akan digunakan, membersihkan data dan mentransformasi data ke dalam format yang sama.
  4. *Modeling*, yaitu proses membangung model-model dari lagoritma yang telah ditentukan dengan mencari parameter yang optimal.
  5. *Evaluation*, yaitu proses yang dilakukan untuk mengevaluasi model yang telah dibangun, seperti menguji model yang dibangun kemudian melakukan analisis terhadap hasil pengujian model tersebut.
  6. *Deployment*, yaitu proses melakukan *deploy* dari model optimal terpilih sehingga dapat digunakan pada proses bisnis.

## Sistematika Penulisan

Untuk memberi gambaran yang jelas tentang penelitian ini, maka disusunlah sistematika penulisan yang berisi materi yang akan dibahas pada setiap bab. Sistematika dalam penulisan tugas akhir ini adalah sebagai berikut:

**BAB I PENDAHULUAN**

Pada bab ini dijelaskan tentang latar belakang dari topik penulisan skripsi, pokok permasalahan berupa identifikasi dan batasan masalah, tujuan dan manfaat yang diharapkan dari penulisan skripsi, metodologi yang digunakan serta sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Pada bab ini berisi penjelasan mengenai teori dan penelitian terkait yang menjadi dasaar penelitian ini, seperti mengenai analisis sentimen, *data preprocessing*, dan metode LSTM.

**BAB III ANALISIS DAN PERANCANGAN**

Pada bab ini dijelaskan secara rinci mengenai tahapan penelitian, seperti tahap menyiapkan data, komputasi model dan analisis kebutuhan aplikasi.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini dibahas hasil dari penelitian yang dilakukan dan analisa terhadap hasil implementasi tersebut.

**BAB V KESIMPULAN DAN SARAN**

Bab ini merupakan penutup yang berisi kesimpulan dan saran dari penelitian yang sudah dilakukan.

# BAB II

**TINJAUAN PUSTAKA**

## Analisis Sentimen

Sentimen adalah perasaan yang mendasari, sikap, evaluasi, atau emosi yang terkait dengan opini. Orientasi sentimen bisa positif, negatif, atau netral. Target sentimen atau dikenal juga sebagai target opini, adalah entitas atau aspek entitas tempat sentimen tersebut diungkapkan (Liu, 2010). Analisis sentimen atau *opinion mining* adalah studi komputasional tentang opini orang, sentimen, penilaian, sikap, dan emosi terhadap entitas dan aspek-aspeknya yang diekspresikan melalui teks (Bounds, 2017).

Secara umum terdapat dua metode yang biasa digunakan dalam melakukan analisis sentiment, yaitu dengan pendekatan *tule based/lexicon based* dan pendekatan dengan *machine learning/supervised learning* (Liu, 2010). Keuntungan utama dari pendekatan *supervised learning* untuk klasifikasi sentimen adalah dapat belajar secara otomatis dari semua jenis fitur untuk klasifikasi melalui optimasi. Sebagian besar fitur ini sulit digunakan dengan metode *lexicon based*. Namun, metode *supervised learning* bergantung pada data pelatihan, yang perlu diberi label secara manual untuk setiap domain. Kelemahan dari endekatan ini adalah *supervised learning* yang dilatih dari data berlabel di satu domain sering kali tidak berfungsi di domain lainnya. Sehingga untuk setiap domain, data pelatihan beru perlu diberi label yang memakan waktu lama (Bounds, 2017).

## Text Preprocessing

*Text preprocessing* adalah sebuah tahap awal dari *text mining* utnuk mempersiapkan data teks, seperti membersihkan data dari *noise*, ataupun mengubah format data sehingga mempermudah dalam melakukan pemrosesan data. *Text preprocessing* merupakan salah satu bagian yang penting dalam melakukan analisis sentimen, karena hasilnya akan berpengaruh terhadap performa klasifikasi sentimen yang akan dilakukan (Indurkhya and Damerau, 2010).

Berikut ini merupakan beberapa tahap yang umumnya dilakukan pada proses *text preprocessing* :

1. Case Folding
2. Noise Removal
3. Tokenizing
4. Normalizing
5. Stemming
6. Stopword Removal

## *Long Short-Term Memory*

*Long Short-Term Memory* (LSTM) merupakan salah satu arsitektur jaringan saraf tiruan yang termasuk dalam kategori *Recurrent Neural Netweok* (RNN). LSTM dirancang secara khusus untuk mengatasi kelemahab utama RNN yaitu *vanishin gradient* ketika memproses data berurutan dalam rentang waktu yang panjang (Hochreiter & Schmidhuber, 1997).

LSTM memperkenalkan struktur memori internal yang disebut *cell state*, serta tiga gerbang utama yaitu *input gate, forget gate*, dan *output gate* yang memungkinkan model untuk menyimpan, menghapus, atau meneruskan informasi tertentu pada setiap waktu (Staudemeyer & Morris, 2019).

Dalam konteks pemrosesan bahasa alami, LSTM banyak digunakan untuk menangkap hubungan urutan kata dalam kalimat karena kemampuannya dalam memahami konteks jangka panjang. Salah satu penerapan paling umum adalah pada analisis sentimen, LSTM dapat mengklasifikasikan ekspresi opini dalam bentuk teks menjadi berbagai macam kategori sentimen. LSTM bekerja dengan menerima *input* berupa data teks yang sudah di *word embedding* dan kemudian memproses urutan kata tersebut untuk mengenali pola-pola yang menunjukkan arah sentiemn tertentu (Zhang et al., 2018).

Keunggulan utama dari penggunaan LSTM dalam analisis sentimen adalah kemampuannya untuk mempertahankan informasi penting dalam urutan kata yang panjang. Namun, model ini tetap bergantung pada kualitas *embedding* dan *training data* yang digunakan. Model LSTM akan bekerja optimal ketika dikombinasikan dengan teknik *word embedding* dan ketika dilatih *data training* yang digunakan adalah data yang sesuai dengan domain analisisnya (Vijayakumar, 2024)

## *Unified Modeling Language* (UML)

*Unified Modelling Language* atau disingkat UML adalah notasi grafis yang digunakan untuk menggambarkan, merancang dan mendokumentasikan sistem perangkat lunak (Fowler, 2004). UML terdiri dari berbagai jenis diagram yang digunakan untuk merepresentasikan berbagai aspek dari perangkat lunak seperti struktur, aktivitas, dan interaksi.

Pada umumnya UML digunakan sebagai sketsa atau *blueprint* dari sistem perangkat lunak yang dikembangkan. Diagram UML dapat digunakan sebagai media komunikasi pengembang baik dalam tim yang sama maupun berbeda. Beberapa diagram UML yang umum digunakan dalam pengembangan perangkat lunak adalah *Use Case Diagram, Activity Diagram, Sequence Diagram* dan *Class Diagram.*

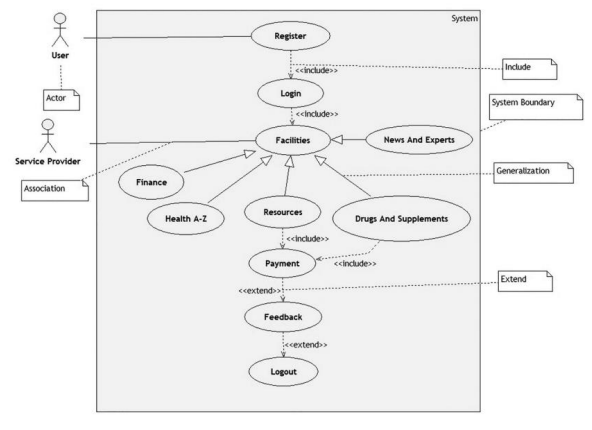
### *Use Case Diagram*

*Use Case Diagram* meruapakan tampilan grafis yang daftar *use case* (fitur/kasus penggunaan). Diagram ini menunjukkan batasan fitur dan interaksi yang disediakan aplikasi. *Use case diagram* memiliki 3 komponen yaitu aktor, *use cases*, dan hubungan antara aktor dan *use case*.

*Use case diagram* digunakan untuk memudahkan pemahaman kebutuhan fungsional aplikasi. *Use case diagram* sebaiknya dibuat tampilan yang sederhana dan tidak terlalu rumit untuk memudahkan ketika membaca dan memahaminya. Masing-masing simbol dalam *use case diagram* dijelaskan pada Tabel 2.1.

Tabel ‎2.1 Simbol-simbol pada *Use Case Diagram*

| **No** | **Simbol** | **Keterangan** |
| --- | --- | --- |
| 1 | *Use Case*  *https://lh4.googleusercontent.com/aGoWgN9QLWt6cYhoD6If-1pVWdIGSDthlP-JHI4l_otU89K6rVLBWwUy1SpKgWMi8hH4mOUUioPHpG1pAJWEens_kDuXbhkl-SkRL405imaUUSx9_WfNtOM03LNWzAgdoMUaGqcX* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor |
| 2 | *Actor*  *https://lh3.googleusercontent.com/tSEAg0SgSnqtT21tCniUCUu9Z7WocIUCePUXRQsbkimjod8vkF3FvQDwet-ag9UKfGbmo8NsF1Y6QN4bXYrCJgWZfS46QhtqQfOUqfl9MHCGdD7YQ1b-MjVUq4fSWu66r0gh37-b* | Orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Biasanya dinyatakan menggunakan kata benda atau nama aktor. |
| 3 | *Association*  *https://lh4.googleusercontent.com/SViGOIxkm4T82HeVAuSiZmc780safOlfVR22PvNtnWcDiPQzc4VCwLBlYcRJdTk_eSTupCQeZTIpVJe2AqMBzkaLi3ADmPAcvI_36DfEzuuvUl7YRIChlYH9mGJ23q5XR3seySmL* | Komunikasi antara aktor dengan *use case* yang berpartisipasi pada diagram |
| 4 | *Extend*  *https://lh6.googleusercontent.com/J3JBlXN1lK5hwy3QXjzpQ3k1G4oWSpznif1uRJPeB0MEmq9lMIgVZKW4nhjF5J1QYjikKo8stGbeXPfwkLzi-8rrkyPrH2-yTSUSSf_FBtb_K_1Nm-3QLZSF2sG-i5n27iGG2pf-* | Relasi *use case* tambahan ke sebuah *use case*, dimana *use case* yang ditambahkan dapat berdiri sendiri meski tanpa *use case* tambahan itu. Arah panah mengarah pada *use case* yang ditambahkan. |
| 5 | *Generalization*  *https://lh3.googleusercontent.com/XwQKVQuzFNE-E2-BKEUf4hEvU_t77aWyx_JTwycI-lJcHPdabIEBYD_LRX75KGC7nQ7y76qzoSaMQEPOKqZMqgN4ahBnOqsmq2LUCrL8tNtQYNrY8sYhtJ8KmJSQSeawCdIccmvd* | Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 6 | *Include*  *https://lh6.googleusercontent.com/Pq5A44NAZ6zjy1qTIGxJ_ndVgw42YUy2zmxCRxJP4KGR3vX2n-J0XJJBmTNFR9VT8GnlGNTWtc7Wu8zZ5aC4lX7t1maDY0rddsJnBmNCrgNzZ8q8aAbtLgfOKeifGlLUPNtSAlU7* | Relasi *use case* tambahan ke sebuah *use case,* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. Arah panah *include* mengarah pada *use case* yang dibutuhkan |



Gambar ‎2.1 Contoh *Use Case Diagram* (Suriya Sundramoothy, 2022)

Gambar 2.1 merupakan salah satu contoh penggunaan *Use Case Diagram* pada aplikasi. Pada diagram tersebu, terdapat dua aktor yang dapat melakukan berbagai aktivitas dengan ketentuan aturan-aturan yang terdapat pada tabel penggunaan *Use Case Diargam*.

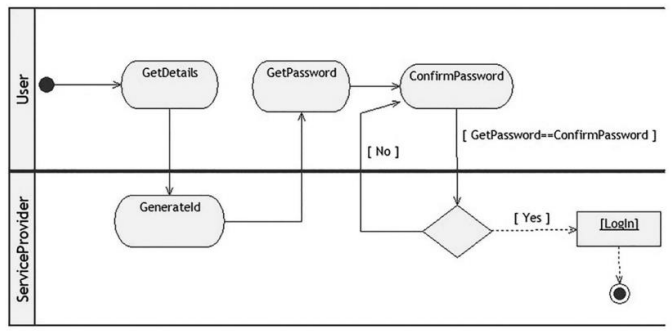
### *Activity Diagram*

*Activity Diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan alur kerja aplikasi, Diagram ini menampilkan rangkaian aktivitas yang terjadi dalam aplikasi. *Activity diagram* digunakan untuk membantu pengembang dan pihak yang terlibat dalam pembangunan aplikasi dalam memahami proses atau alur kerja yang terjadi dalam pembangunan aplikasi dalam memahami proses atau alur kerja yang terjadi dalam aplikasi, bagaimana interaksi antar tugas terjadi, dan bagaimana keputusan akan memengaruhi alur kerja.

*Activity diagram* memiliki beberapa komponen utama yang ditampilkan dalam bentuk simbol. Masing-masing simbol mewakili suatu elemen atau aktivitas yang terjadi pada sistem. Penjelasan dari masing-masing simbol pada *activity diagram* dijelaskan pada Tabel 2.2.

Tabel ‎2.2 Simbol-Simbol pada *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1 | Status Awal | Menandakan tindakan awal atau titik awal aktivitas untuk setiap diagram aktivitas. |
| 2 | Aktivitas | Menunjukkan aktivitas yang dilakukan sistem |
| 3 | Percabangan */ decision* | Asosiasi percabangan adalah keadaan dimana terdapat pilihan aktivitas lebih dari satu. |
| 4 | Penggabungan */ join* | Asosiasi penggabungan adalah keadaan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 | *Status akhir* | Menunjukan bagian akhir dari aktivitas. |
| 6 | *Swimlane* | *Swimlane* memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |



Gambar ‎2.2 Contoh *Activity Diagram* (Suriya Sundramoothy, 2022)

Pada gambar 2.2 merupakan alur *Activity Diagram* yang digunakan oleh *User* ketika melakukan *login* pada suatu aplikasi. Pada *Activity Diagram* tersebut, terdapat interaksi pada *User* dan *Service Provider* dimana user akan menginputkan informasi seperti *username*, email, dan *password* untuk melakukan *login.*

## Penelitian Terdahulu

# BAB III

**ANALISIS DAN PERANCANGAN**

Sebagaimana yang telah dijelaskan pada bab sebelumnya, telah disajikan tinjauan pustaka yang berkaitan dengan penelitian ini. Metode yang dipakai dalam penelitian ini yaitu menggunakan metode LSTM. Selanjutkan akan dilakukan proses analisis mulai dari data penelitian yang digunakan, langkah-langkah penelitian, hingga evaluasi kinerja model

## Objek Penelitinan

Data yang penulis gunakan dalam penelitian ini merupakan data sekunder berupa tweet pengguna pada aplikasi X (twitter) yang didapatkan dari Twitter dengan rentang waktu 7 oktober 2023 hingga 7 April 2025. Data yang berhasil dikumpulkan dalam penilitian ini sebanyak ...  *tweet* dengan fokus utama pada isu aksi genosida Israel terhadap Palestina.

* + - * Asal-usul sentimen pro-konta-netral
      * Contoh sentimen pro-kontra-netral

## Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahapan yang meliputi pengumpulan data, pelabelan data, *data balancing*, *data preprocessing*, pengelompokkan data *training* dan *validation*, pembobotan fitur menggunakan Word2vec, proses perancangan dan pembangunan model LSTM. Gambar 3.1 merupakan diagram alur tahapan penelitian yang dilakukan.

Gambar ‎3.1 Alur Tahapan Penelitian

### Pengumpulan Data

Data yang dikumpulkan pada penelitian ini adalah data teks berupa *tweet* dari media sosial X (Twitter) dengan menggunakan teknik *crawling* secara *online*

### Pelabelan Data

### *Data Balancing*

### *Data Preprocessing*

### Data *Training*, *Validation* dan *Predictiing*

### Lapisan *Word Embedding* (Word2vec)

### Lapisan Long Short-Term Memory

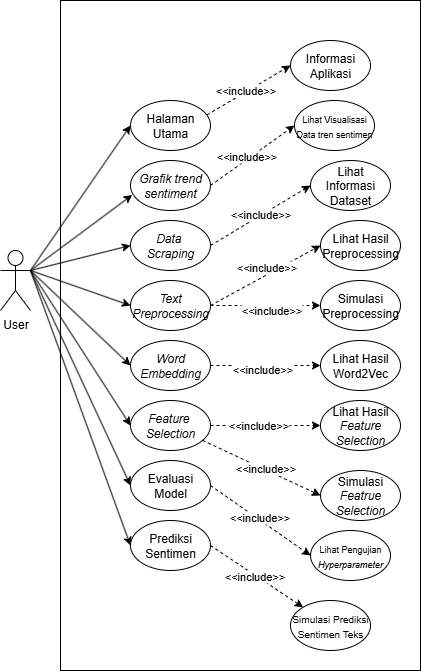
### Proses Pembuatan Model LSTM

Proses tuning dan training model

## Perancangan Aplikasi

Perancangan sistem merupakan proses pembuatan rancangan dari sistem yang akan dibangun dalam bentuk aplikasi. Pada penelitian ini, perancangan desain sistem dibuat dalam bentuk diagram UML yang terbagi menjad

### Use Case Diagram



Gambar ‎3.2 *Use Case Diagram* Aplikasi

*Use Case Diagram* digunakan dalam mengvisualisasikan aktivitas yang dapat dilakukan oleh pengguna sistem. Pada sistem ini terdapat menu utama, yaitu halaman depan, halaman tren sentimen, halaman data scraping, halaman text preprocessing, halaman word embedding, halaman feature selection, halaman hasil training dan evaluasi model, dan halaman prediksi sentimen. Seluruh fitur tersebut dapat digambarkan pada *use case diagram* yang dapat dilihat pada gambar 3.1 yang akan dijabarkan sebagai berikut.

Halaman depan menampilkan deskripsi terkait aplikasi yang dibangun beserta pengembanganya.

Halaman tren sentimen menampilkan visualisasi dari perkembangan tren sentimen pada dataset.

Halaman *data scraping* menampilkan deskripsi terkait data yang digunakan serta metode pengumpulannya. Pada halaman ini pengguna dapat melihat sampel dataset yang telah dikumpulkan peneliti.

Halaman *text preprocessing* menampilkan tahapan-tahapan text preprocessing yang digunakan pada penelitian ini secara berururtan. Pengguna dapat melihat hasil text preprocessing maupun melakukan simulasi *text preprocessing*.

Halaman *word embedding* menampilkan hasil dari komputasi word ke vektor. Pengguna dapat melihat hasil dari *word embedding* yang dilakukan pada dataset.

Halaman *Feature Selection* menampilan hasil dari

### Activity Diagram

### Deployment Diagram

### Perancangan Antarmuka

## Analisis Kebutuhan

Analisis kebutuhan pada penelitian ini terdiri dari kebutuhan perangkat lunak (*software*) dan kebutuhan perangkat keras (*hardware*).

### Kebutuhan Perangkat Lunak

Berikut merupakan perangkat lunak yang digunakan dalam penelitian ini, yaitu:

1. Sistem Operaasi Microsoft Windows 11.
2. Visual Studio Code
3. Google Collab
4. *Browser* Google Chrome
5. Microsoft Excel
6. Bahasa Pemrograman Python
7. Streamlit

### Kebutuhan Perangkat Keras

Dalam melakukan penelitian ini dibutuhkan perangkat keras yaitu sebuah *laptop* dengan spesifikasi sebagai berikut:

1. Manufaktur : Asus M415D
2. *Processor*  : AMD Ryzen 3 3250U CPU 2.5 GHz
3. RAM : 16 GB
4. *Storage* : SSD 500 GB
5. *Graphic Processor* : Radeon Graphics

# BAB IV

**HASIL DAN PEMBAHASAN**

Pada bab ini akan dijelaskan hasil dari implementasi model LSTM termasuk analisis pengujian berdasarkan *Hyperparameter* model, kemudian dalam bab ini akan dijelaskan bagaimana aplikasi analsis sentimen diimplementasikan dalam bentuk *website*.

## Analisis Pengujian *Hyperparameter*

Dalam menentukan model yang paling optimal dalam mengklasifikasi sentiemen diperlukan beberapa pengujian terhadap *hyperparameter* yang digunakan pada model LSTM tersebut. Diantara *hyperparameter* yang akan dianalisis adalah Word2vec, jumlah lapisan LSTM, *optimizer, dropout* dan lain sebagainya.

### Word2vec

### Analsis Jumlah Lapisan LSTM

### Analsisi Unit LSTM

### Analsis *Optimizer*

### Analisis *Dropout*

## Implementasi Aplikasi

## Limitasi Penelitian

Proses pembanguna model, kendala dan cara mengoptimalkan model

Saran untuk penelitian ini.

# BAB V

**KESIMPULAN DAN SARAN**

## Kesimpulan

Berdasarkan berbagai hal yang telah disampaikan pada bab sebelumnya serta perancangan dan implementasi web repositori yang sudah dilakukan penulis, maka dapat diambil simpulan sebagai berikut:

1. ……….. ………….. . ……….. ……….. ……… ………… …………. ……. ……….. …….. . ……….. …….. ……….. .
2. ……….. ………….. . ……….. ……….. ……… ………… …………. ……. ……….. …….. . ……….. …….. ……….. .
3. ……….. ………….. . ……….. ……….. ……… ………… …………. ……. ……….. …….. . ……….. …….. ……….. .
4. **Saran**

Berdasarkan penelitian yang dilakukan, penulis memberikan saran yang dapat

diimplementasikan pada penelitian selanjutnya, yaitu sebagai berikut:

1. 1. ……….. ………….. . ……….. ……….. ……… ………… …………. ……. ……….. …….. . ……….. …….. ……….. .
2. 2. ……….. ………….. . ……….. ……….. ……… ………… …………. ……. ……….. …….. . ……….. …….. ……….. .
3. 3. ……….. ………….. . ……….. ……….. ……… ………… …………. ……. ……….. …….. . ……….. …….. ……….. .

# DAFTAR PUSTAKA

Sukamto, R.A. dan Shalahuddin, M. (2018) *Rekayasa Perangkat Lunak*. Bandung: Informatika.

Ater, T. (2017) *Building Progressive Web Apps: Bringing the Power of Native to the Browser*. Sebastopol: O’Reilly Media, Inc.

Beck, K. and Andreas, C. (2004) *Extreme Programming Explained: Embrace Change*. Boston: Addison Wesley.

Beck, K. and Fowler, M. (2000) *Planning Extreme Programming*. Addison Wesley.

Ferguson, R. (2019) *Beginning JavaScript. The Ultimate Guide to Modern JavaScript Development*. New Jersey: Apress.

Freeman, A. (2019) *Pro React 16*. London: Apress.

Gs.statcounter.com (2020) *Desktop vs Mobile vs Tablet Market Share Worldwide StatCounter Global Stats*. Available at: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet (Accessed: 1 June 2020).

Hahn, E. M. (2016) *Express In Action. Writing, building, and testing Node.js applications*. New York: Manning Publications Co.

Handiwidjojo, W. dan Ernawati, L. (2016) ‘Pengukuran Tingkat Ketergunaan (Usability) Sistem Informasi Keuangan Studi Kasus: Duta Wacana Internal Transaction (Duwit)’, *JUISI*, 2(1).

Heroku.com (2020) *What is Heroku?* Available at: https://www.heroku.com/about (Accessed: 1 June 2020).

Heroku (2020) *Heroku Dynos*. Available at: https://www.heroku.com/dynos (Accessed: 1 June 2020).

Hume, D. A. (2017) *Progressive Web Apps*. New York: Manning Publications.

Jaya, T. S. (2018) ‘Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)’, *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 3(2).

Karpagam, D. V. *et al.* (2017) ‘Performance Enhancement of Webpage Using Progressive Web App Features’, *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 4(3).

Krishna, T. S. R. *et al.* (2011) ‘Survey on Extreme Programming in Software Engineering’, in *International Journal of Computer Trends and Technology*.

Maryuliana. dkk. (2016) ‘Sistem Informasi Angket Pengukuran Skala Kebutuhan Materi Pembelajaran Tambahan Sebagai Pendukung Pengambilan Keputusan Di Sekolah Menengah Atas Menggunakan Skala Likert’, *Jurnal Transistor Elektro dan Informatika (TRANSISTOR EI)*, 1(2).

Mehta, C. *et al.* (2018) *MySQL 8 Administrator’s Guide*. Birmingham: Packt.

Moran, K. (2019) *Usability Testing 101*. Available at: https://www.nngroup.com/articles/usability-testing-101/ (Accessed: 1 June 2020).

Nadia, R., dkk. (2018) ‘Rancang Bangun Aplikasi CallTenant dengan Penyimpanan Basis Data untuk Form Dinamis Menggunakan Framework Laravel’, *JURNAL TEKIK*, 7(1).

Nielsen, J. (1993) *Usability Engineering*. Morgan Kaufmann.

Nielsen, J. (1994) *10 Heuristics for User Interface Design: Article by Jakob Nielsen*. Available at: https://www.nngroup.com/articles/ten-usability-heuristics/ (Accessed: 2 June 2020).

Repanovici, A. (2009) ‘Marketing Research about Attitudes, Difficulties and Interest of Academic Community about Institutional Repository’, *Proceedings of the 3rd International Conference in Management, Marketing and Finances*, MMF’09, pp. 88–95.

Ropianto, M. (2016) ‘Pemahaman Penggunaan Unified Modelling Language’, *JT-IBSI*, 01(01).

Santoni, M. (2018) *Progressive Web Apps browser support & compatibility*. Available at: https://www.goodbarber.com/blog/progressive-web-apps-browser-support-compatibility-a883/ (Accessed: 20 January 2019).

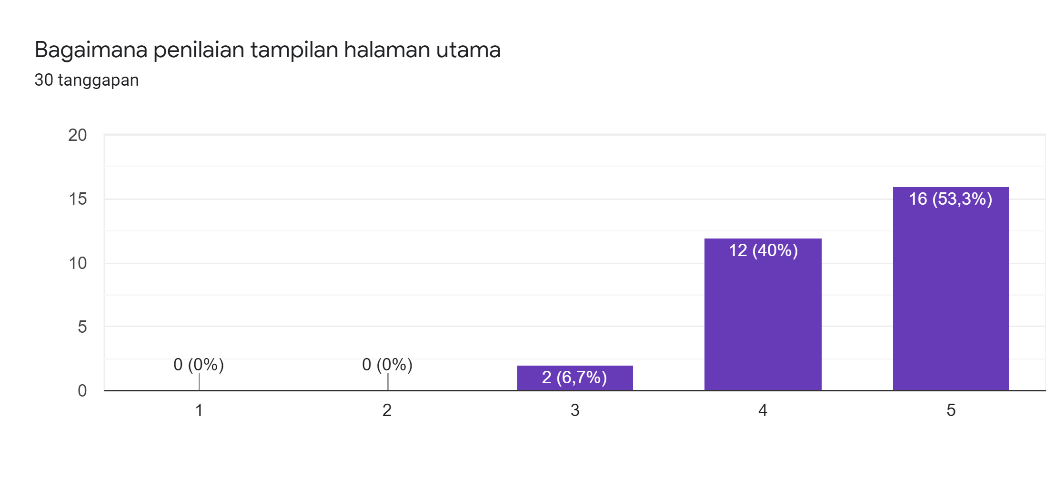
Singh, H. and Bhatt, M. (2016) *Learning Web Development with React and Bootstrap*. Birmingham: Packt Publishing Ltd.

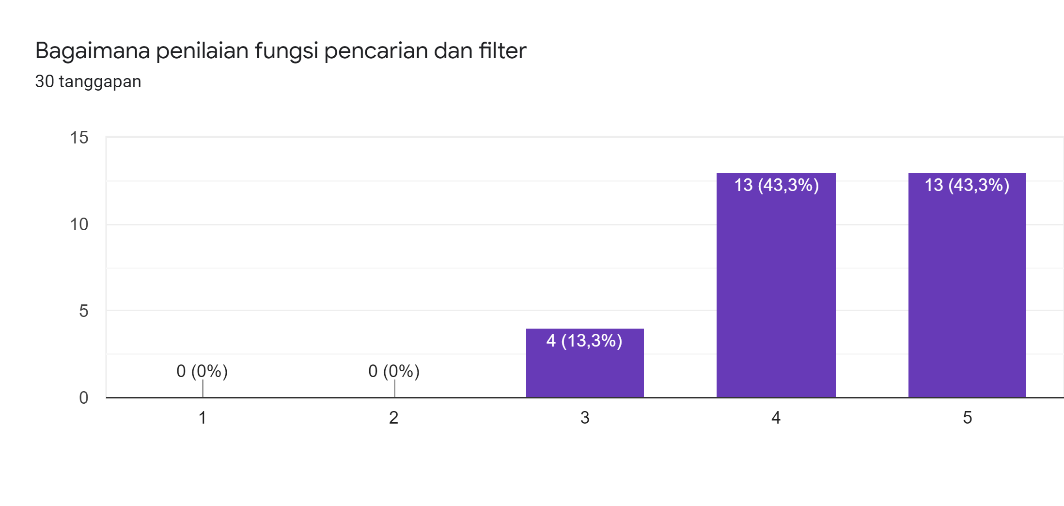
Spillner, A., *et al.* (2014) *Software Testing Foundations*. Santa Barbara: Rocky Nook Inc.

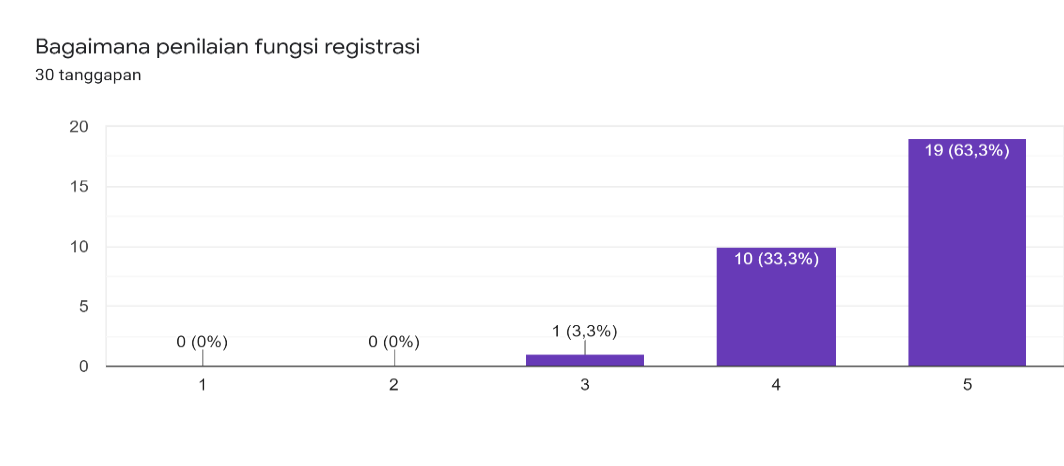
Sutedjo, M. (2014) ‘Pengelolaan Repositori Perguruan Tinggi dan Pengembangan Repositori Karya Seni’, *Makalah Seminar Nasional ”Digital Local Content: Strategi Membangun Repository Karya Seni”, GKU FSR ISI Yogyakarta*.

# LAMPIRAN

**Lampiran 1 Ringkasan Hasil *Usability Testing***

****

****

****

**Lampiran 2** **Kode Web Aplikasi**

Register

|  |
| --- |
| import React, { Component } from 'react'  import bg2 from '../icons/bg2.webp'  import axios from 'axios'  import { scrollToTop } from '../helpers/autoScroll'  import { ProgressBar, Modal} from 'react-bootstrap'  import { Redirect } from 'react-router-dom'  import { connect } from 'react-redux'  export class Register extends Component {  initialState = {  showLoading:false,  name:'',  npm: '',  email:'',  pass:'',  rePass: '',  message: '',  status:'',  displayForm1:'block',  displayForm2:'none',  displayForm3:'none',  file:null,  progress:34  }  state=this.initialState  next = (e) =>{  e.preventDefault()  this.setState({  showLoading:true,  })  let {name, npm, pass, email}= this.state  let data={  name:name,  email:email,  npm:npm,  password:pass  }  axios({  method: 'POST',  url: '/check-form',  data: data  }).then(res => {  this.setState({  message: '',  displayForm1: 'none',  displayForm2: 'block',  progress: this.state.progress + 33,  showLoading:false  })  scrollToTop()  }).catch(err => {  this.setState({  showLoading:false  })  if (err.response) {  this.setState({  message: err.response.data.message,  status: err.response.data.status,  })  }  else{  this.setState({  message: 'Network error, Cek Koneksi Anda',  status: 500,  })  }  })  } |

*Login*

|  |
| --- |
| import React, { PureComponent } from 'react'  import UserMenu from '../components/UserMenu'  import AdminMenu from '../components/AdminMenu'  import { Link, Redirect } from 'react-router-dom'  import { setToken, delToken} from '../reducers/authReducer'  import { connect } from 'react-redux'  import { Modal } from 'react-bootstrap'  import { FaRegCheckCircle } from 'react-icons/fa'  import '../styles/nav.css'  import axios from 'axios'  import { scrollToTop } from '../helpers/autoScroll'  import MediaQuery from 'react-responsive'  export class Nav extends PureComponent {  state = {  npm: '',  pass: '',  message: '',  status: null,  showLogin: false,  showLoading:false,  justLoggedIn:false, //ketika pertama kali login, agar refresh tidak redirect  }  handleInput = (e) =>{  this.setState({  status:'',  [e.target.id] : e.target.value  })  }  submitLogin = e => {  e.preventDefault()  this.setState({  showLoading:true  })  axios({  method: 'post',  url: '/login',  data: {  npm: this.state.npm,  password: this.state.pass  }  }).then(res => {  let loginInfo = res.data.data  console.log(loginInfo)  if (loginInfo.isLogged){ //response didapat  this.props.login(loginInfo) //set state global  this.setState({ //show modal  showLoading:false,  status:res.data.status,  justLoggedIn:true,  showLogin:true,  })  scrollToTop()  setTimeout(() =>  this.setState({ //hide modal  showLogin:false  }), 1000)  }  else{  this.setState({  showLoading:false,  status:500  })  }  }).catch((err) => {  if(err.response){  this.setState({  message:err.response.data.message,  status:err.response.data.status,  showLoading:false,  })  } else{  this.setState({  status: 500,  showLoading:false,  })  }  })  }  logout = () =>{  this.props.logout()  }  handleClose = () => {  this.setState({  showLogin:false  })  } |

Unggah

|  |
| --- |
| import React, { Component } from 'react'  import { connect } from 'react-redux'  import { Redirect, Link } from 'react-router-dom'  import axios from 'axios'  import { Spinner, Modal } from 'react-bootstrap'  export class Upload extends Component {  initialState={  skripsi:{},  isLoaded:false,  offline: false,  showLoading:false,  title:'',  titleAlert:'initial',  year:'',  yearAlert:'initial',  abstract:'',  abstractAlert:'initial',  abstrak:'',  abstrakAlert:'initial',  category:'',  keywords:'',  file:null,  message: '',  status:'',  }  state=this.initialState  submit = (e) =>{  e.preventDefault()  this.setState({  showLoading:true  })  let {title, year, abstrak, abstract, category, keywords} = this.state  let {file} = this.state  const formData = new FormData()  formData.append('file', file)  formData.append('title', title)  formData.append('year', year)  formData.append('abstrak', abstrak)  formData.append('abstract', abstract)  formData.append('category', category)  formData.append('keywords', keywords)  axios({  method: 'POST',  url: `/user/upload/`,  data: formData,  headers:{  'Content-Type':'multipart/form-data',  'Authorization': this.props.token  }  }).then((res) =>{  this.refs.uploadForm.reset()  this.setState({  message:res.data.message,  status:res.data.status,  showLoading:false  })  }).catch((err) => {  this.setState({  showLoading:false  })  if( err.response){  this.setState({  message:err.response.data.message,  status:err.response.data.status,  })  }  })  }  handleInput = (e) =>{  if(e.target.id==='title'){  e.target.value=e.target.value.replace(/\n/g, ' ')  }  this.setState({  [e.target.id] : e.target.value,  })  if (e.target.id==='title'){  this.setState({  titleAlert: e.target.value,  })  }  else if (e.target.id==='year'){  this.setState({  yearAlert: e.target.value,  })  }  else if (e.target.id==='abstract'){  this.setState({  abstractAlert: e.target.value,  })  }  else if (e.target.id==='abstrak'){  this.setState({  abstrakAlert: e.target.value,  })  }  } |

Detail Skripsi

|  |
| --- |
| import React, { Component } from 'react'  import { Spinner } from 'react-bootstrap'  import axios from 'axios'  import { connect } from 'react-redux'  import { FaChevronLeft, FaChevronRight } from 'react-icons/fa'  import {Document, pdfjs, Page} from 'react-pdf'  pdfjs.GlobalWorkerOptions.workerSrc = `//cdnjs.cloudflare.com/ajax/libs/pdf.js/${pdfjs.version}/pdf.worker.js`  export class SkripsiDetail extends Component {  state={  skripsi:[],  isLoaded:false,  offline:false,  pageNumber:1,  numPages:null  }  getData =()=>{  let id = this.props.match.params.id  if(this.props.token){  axios({  method: 'get',  url: `/skripsi/detail/`,  params:{  id : id  },  headers: {  Authorization: this.props.token  }  }).then(res=>{  this.setState({  skripsi: res.data[0],  isLoaded: true  })  }).catch(err=>{  if(err.response){  console.log(err.response)  }  })  }  else{  axios({  method: 'get',  url: `/skripsi/info/`,  params:{  id : id  }  }).then(res=>{  this.setState({  skripsi: res.data[0],  isLoaded: true  })  }).catch(err=>{  if(err.response){  console.log(err.response)  }  })  }  }  onDocumentLoadSuccess = ({ numPages }) => {  this.setState({ numPages });  }  next = () => {  let {pageNumber, numPages} = this.state  if( pageNumber<=numPages){  this.setState({pageNumber:this.state.pageNumber+1})  }  }  before = () => {  let {pageNumber} = this.state  if( pageNumber>1){  this.setState({pageNumber:this.state.pageNumber-1})  }  }  componentDidMount(){  if (navigator.onLine){  this.getData()  this.setState({  offline:false  })  }  else{  this.setState({  offline:true,  })  }  } |

Profil dan Edit *Password*

|  |
| --- |
| import React, { Component } from 'react'  import { Spinner, Modal} from 'react-bootstrap'  import { connect } from 'react-redux'  import axios from 'axios'  import moment from 'moment'  export class ProfileInfo extends Component {  state={  user:{},  isLoaded:false,  showLoading:false,  offline:false,  newPass:'',  oldPass:'',  confirmPass:'',  message:'',  status:null  }  getProfile= ()=>{  axios({  method: 'get',  url: `/user/profile/`,  headers: {  Authorization: this.props.token  }  }).then(res=>{  this.setState({  user: res.data,  isLoaded: true  })  }).catch(err=>{  console.log(err.response)  })  }  componentDidMount(){  if (navigator.onLine){  this.getProfile()  this.setState({  offline:false  })  }  else{  this.setState({  offline:true,  })  }  } |

Status Skripsi

|  |
| --- |
| import React, { Component } from 'react'  import { connect } from 'react-redux'  import axios from 'axios'  import {Spinner} from 'react-bootstrap'  import { FaCheck, FaFilePdf, FaTimes } from 'react-icons/fa'  import { Link } from 'react-router-dom'  import moment from 'moment'  export class SkripsiStatus extends Component {  state={  skripsi:{},  isLoaded:false,  Offline:false  }  getSkripsi=()=>{  axios({  method: 'get',  url: `/user/skripsi/`,  headers: {  Authorization: this.props.token  }  }).then(res=>{  this.setState({  skripsi: res.data,  isLoaded: true  })  }).catch(err=>{  if(err.response){  console.log(err.response)  }  })  }  componentDidMount(){  if (navigator.onLine){  this.getSkripsi()  this.setState({  offline:false  })  }  else{  this.setState({  offline:true,  })  }  } |

Kontak Admin

|  |
| --- |
| import React, { Component } from 'react'  import { Redirect } from 'react-router-dom'  import { Spinner } from 'react-bootstrap'  import { connect } from 'react-redux'  import axios from 'axios'  import Forum from '../components/Forum'  class UserForum extends Component {  state={  text:'',  dataLoaded:false,  isLoading:false,  chats:[],  offline:false,  message:''  }  handleText = e =>{  this.setState({  [e.target.id] : e.target.value.trim()  })  }  submit = e => {  let {text} =this.state  e.preventDefault()  if(text){  this.setState({  isLoading:true  })  axios({  method: 'post',  url: '/user/insert-text',  headers: {  Authorization:this.props.token  },  data: {  text: text,  }  }).then(res => {  this.setState({  isLoading:false,  text:''  })  this.getForum()  this.refs.messages.reset()  }).catch(err=>{  console.log(err.response)  if(err.response){  this.setState({  message:err.response.data.message  })  }  this.setState({  isLoading:false,  })  setTimeout(() =>  this.setState({  message:''  }), 5000)  })  }  }  getForum= ()=>{  axios({  method: 'get',  url: '/user/forum',  headers: {  Authorization:this.props.token  }  }).then(res=>{  this.setState({  dataLoaded:true,  chats:res.data  })  }).catch((err) => {  if(err.response){  console.log(err.response.statusText)  }  this.setState({  dataLoaded:true  })  })  }  componentDidMount(){  if (navigator.onLine){  this.getForum()  this.setState({  offline:false  })  }  else{  this.setState({  offline:true,  })  }  } |

Lupa *Password*

|  |
| --- |
| import React, { Component } from 'react'  import { Spinner } from 'react-bootstrap'  import { connect } from 'react-redux'  import { Redirect } from 'react-router-dom'  import Bg3 from '../components/Bg3'  import axios from 'axios'  export class Forgot extends Component {  state={  email:'',  npm:'',  checknpm:'',  displaySection1:'block',  displaySection2:'none',  sending:false,  status:'',  message:''  }  handleInput = e => {  this.setState({  [e.target.id] : e.target.value,  })  }  checkNpm = e =>{  this.setState({  checknpm : e.target.value,  })  }  sendEmail = e =>{  e.preventDefault()  this.setState({  sending:true  })  axios({  method: 'put',  url: '/forgot-pass',  data: {  npm: this.state.npm,  email: this.state.email  }  }).then(res => {  this.refs.forgotForm.reset()  this.setState({  sending:false,  status:res.data.status,  message:res.data.message,  displaySection1:'none',  displaySection2:'block'  })  }).catch((err) => {  console.log(err.response)  if(err.response){  this.setState({  status:err.response.data.status,  message:err.response.data.message,  sending:false,  })  }  else{  this.setState({  message: 'Network error, Cek Koneksi Anda',  status: 500,  sending:false,  })  }  })  } |

PWA

|  |
| --- |
| //Lifecycle = register -> install -> activate https://bit.ly/CRA-PWA  //ip localhost ([::1] - 127.0.0.0/8 )  const isLocalhost = Boolean(  window.location.hostname === 'localhost' ||  window.location.hostname === '[::1]' ||  window.location.hostname.match(  /^127(?:\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)){3}$/  )  )  export function register(config) {  //Cek apakah service worker support pada browser  if ('serviceWorker' in navigator) {  const publicUrl = new URL(process.env.PUBLIC\_URL, window.location.href)  //Service worker tidak jalan jika public file diluar url  if (publicUrl.origin !== window.location.origin) {  return  }  //Ketika halaman telah di load  window.addEventListener('load', () => {  const swUrl = `${process.env.PUBLIC\_URL}/sw-custom.js`  if (isLocalhost) {  checkValidServiceWorker(swUrl, config)  navigator.serviceWorker.ready.then(() => {  console.log('Web app served cache-first')  })  } else {  //Register SW  registerValidSW(swUrl, config)  }  })  }  }  function registerValidSW(swUrl, config) {  //------Register SW  navigator.serviceWorker  .register(swUrl)  .then(registration => {  registration.onupdatefound = () => {  const installingWorker = registration.installing  if (installingWorker == null) {  return  }  installingWorker.onstatechange = () => {  //------------install SW  if (installingWorker.state === 'installed') {  //Cek apakah sw aktif  if (navigator.serviceWorker.controller) {  // Update precached content di fetch tapi SW lama masih di jalankan sampai tab di tutup  console.log('Content baru telah di fetch dan siap digunakan saat tab telah ditutup')  if (config && config.onUpdate) {  config.onUpdate(registration)  }  } else {  console.log('Content telah di cached untuk mode offline.')  if (config && config.onSuccess) {  config.onSuccess(registration)  }  }  }  }  }  })  .catch(error => {  console.error('Error saat register service worker:', error)  })  }  function checkValidServiceWorker(swUrl, config) {  // Cek apakah SW sudah ada  fetch(swUrl, {  headers: { 'Service-Worker': 'script' }  })  .then(response => {  const contentType = response.headers.get('content-type')  if (response.status === 404 || (contentType != null && contentType.indexOf('javascript') === -1)) {  // SW tidak di temukan  navigator.serviceWorker.ready.then(registration => {  registration.unregister().then(() => {  window.location.reload()  })  })  } else {  // SW ditemukan. Proses dilanjutkan  registerValidSW(swUrl, config)  }  })  .catch(() => {  console.log('Tidak ada koneksi Internet. Web App dalam offline mode.')  })  }  export function unregister() {  if ('serviceWorker' in navigator) {  navigator.serviceWorker.ready.then(registration => {  registration.unregister()  })  }  } |

# RIWAYAT HIDUP