**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. 19680717 199303 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 “**………………………………**” 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 Bapak ……………., sebagai pembimbing utama, Bapak Aditya Pradana, S.T., M.Eng, 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:

………………., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

………………., selaku Kepala Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

………………., 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, 2 Pebruari 2021

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 ISI

[ABSTRAK v](#_Toc203050563)

[*ABSTRACT* vi](#_Toc203050564)

[DAFTAR ISI vii](#_Toc203050565)

[DAFTAR TABEL x](#_Toc203050566)

[DAFTAR GAMBAR xi](#_Toc203050567)

[DAFTAR LAMPIRAN xiii](#_Toc203050568)

[BAB I 1](#_Toc203050569)

[1.1. Latar Belakang 1](#_Toc203050570)

[1.2. Identifikasi Masalah 2](#_Toc203050571)

[1.3. Batasan Masalah 3](#_Toc203050572)

[1.4. Maksud dan Tujuan Penelitian 4](#_Toc203050573)

[1.5. Manfaat Penelitian 4](#_Toc203050574)

[1.6. Metodologi Penelitian 5](#_Toc203050575)

[1.7. Sistematika Penulisan 5](#_Toc203050576)

[BAB II 7](#_Toc203050577)

[2.1. Metodologi Penelitian 7](#_Toc203050578)

[2.2. Analisis Sentimen 8](#_Toc203050579)

[2.3. Text Preprocessing 9](#_Toc203050580)

[2.4. ReactJS 10](#_Toc203050581)

[2.5. Express 10](#_Toc203050582)

[2.6. Javascript 11](#_Toc203050583)

[2.7. Bootstrap 12](#_Toc203050584)

[**1.** **MySQL** 12](#_Toc203050585)

[2.8. Unified Modeling Language (UML) 12](#_Toc203050586)

[2.8.1 Use Case Diagram 12](#_Toc203050587)

[**2.9.2** ***Activity Diagram*** 14](#_Toc203050588)

[**2.9.3** ***Deployment Diagram*** 14](#_Toc203050589)

[**2.10** ***Entity Relationship Diagram (ERD)*** 14](#_Toc203050590)

[2.9. Usability Testing 14](#_Toc203050591)

[**2.11** ***Heuristic Evaluation*** 14](#_Toc203050592)

[**2.12** **Skala Likert** 14](#_Toc203050593)

[**2.13** **Heroku** 15](#_Toc203050594)

[BAB III BAB III 17](#_Toc203050595)

[3.1. Data Penelitian 17](#_Toc203050596)

[**3.1.1** **Kebutuhan Pengguna** 17](#_Toc203050597)

[**3.1.2** **Kebutuhan Data** 17](#_Toc203050598)

[**3.1.3** **Kebutuhan Perangkat Lunak** 17](#_Toc203050599)

[**3.1.4** **Kebutuhan Perangkat Keras** 17](#_Toc203050600)

[**3.1.5** **Kebutuhan Sistem** 18](#_Toc203050601)

[**3.1.6** ***User Story*** 19](#_Toc203050602)

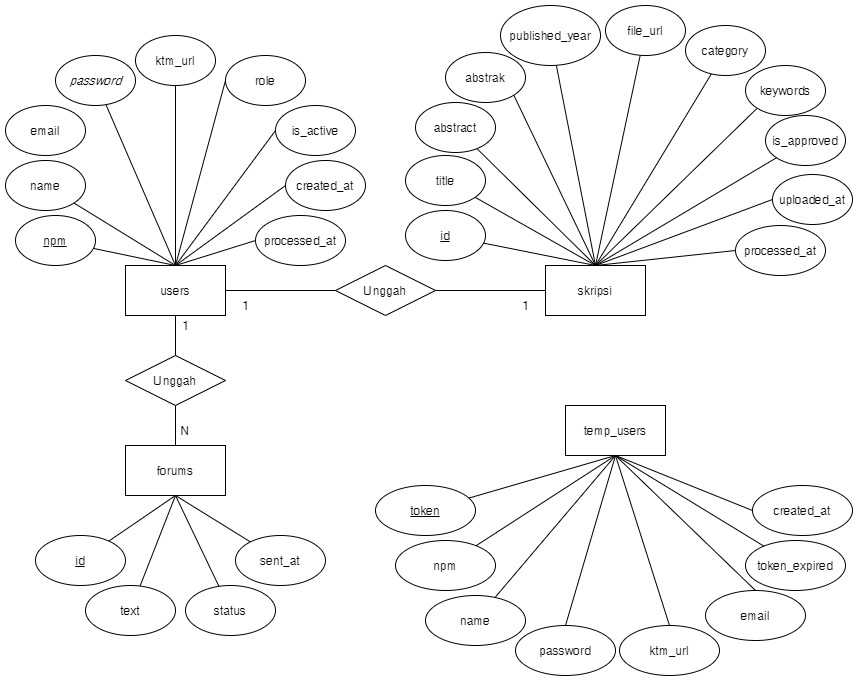
[**3.2** **Fase Perencanaan** 19](#_Toc203050603)

[**3.3** **Fase Iterasi** 20](#_Toc203050604)

[**3.3.1** **Analisis Sistem** 20](#_Toc203050605)

[**3.3.2** **Analisis Arsitektur Menggunakan *Deployment Diagram*** 23](#_Toc203050606)

[**3.3.3** **Desain Basis Data** 24](#_Toc203050607)

[BAB IV  25](#_Toc203050608)

[**3.3.4** **Desain Antarmuka** 27](#_Toc203050609)

[**3.3.5** ***Testing*** 28](#_Toc203050610)

[**3.4** **Fase Produksi** 28](#_Toc203050611)

[**3.5** **Fase Pemeliharaan dan Fase Akhir** 29](#_Toc203050612)

[BAB IV 30](#_Toc203050613)

[**4.1** **Impelementasi Program** 30](#_Toc203050614)

[**4.1.1** **Halaman Utama** 30](#_Toc203050615)

[**4.1.2** **Halaman Register** 33](#_Toc203050616)

[**4.1.3** **Menu *Login*** 35](#_Toc203050617)

[**4.1.4** **Halaman Unggah Skripsi** 35](#_Toc203050618)

[**4.1.5** **Halaman Menu Admin** 35](#_Toc203050619)

[**4.1.6** **Halaman Verifikasi Akun** 35](#_Toc203050620)

[**4.1.7** **Halaman Tinjau Skripsi** 35](#_Toc203050621)

[**4.1.8** **Halaman Detail Skripsi** 35](#_Toc203050622)

[**4.1.9** **Fitur Pencarian dan Penyaringan** 35](#_Toc203050623)

[**4.1.10** **Halaman Profil danFitur *Edit Password*** 35](#_Toc203050624)

[**4.1.11** **Halaman Status Skripsi** 35](#_Toc203050625)

[……. ……….. …………. ……… …………… . ………… …….. …… ……….. 35](#_Toc203050626)

[……. ……….. …………. ……… …………… . ………… …….. …… ……….. 36](#_Toc203050627)

[**4.1.12** **Implementasi PWA** 36](#_Toc203050628)

[……. ……….. …………. ……… …………… . ………… …….. …… ……….. 36](#_Toc203050629)

[……. ……….. …………. ……… …………… . ………… …….. …… ……….. 36](#_Toc203050630)

[**4.2** **Fase Produksi** 36](#_Toc203050631)

[……. ……….. …………. ……… …………… . ………… …….. …… ……….. ……. ……….. …………. ……… …………… . ………… …….. …… ……….. 36](#_Toc203050632)

[**4.2.1** **Rilisan Kecil** 36](#_Toc203050633)

[……. ……….. …………. ……… …………… . ………… …….. …… 36](#_Toc203050634)

[**4.2.2** **Pengujian** 36](#_Toc203050635)

[**4.3** **Fase Pemeliharaan dan Fase Akhir** 36](#_Toc203050636)

[**4.3.1** ***Feedback*** 36](#_Toc203050637)

[**4.3.2** **Implementasi *Feedback*** 36](#_Toc203050638)

[BAB V 37](#_Toc203050639)

[**5.1** **Kesimpulan** 37](#_Toc203050640)

[**5.2** **Saran** 37](#_Toc203050641)

[DAFTAR PUSTAKA 38](#_Toc203050642)

[BAB VI LAMPIRAN 40](#_Toc203050643)

[RIWAYAT HIDUP 55](#_Toc203050644)

DAFTAR TABEL

[Gambar 2.3 Tabel 2.1 Simbol-Simbol pada *Use Case Diagram* 13](#_Toc203047517)

[Gambar 2.4 Tabel 2.5 Titik Respons 15](#_Toc203047518)

[Gambar 3.1 Tabel 3.1 Kebutuhan Sistem Pengguna: Mahasiswa 17](#_Toc203047519)

[Gambar 3.7 Tabel 3.5 Atribut Tabel skripsi 25](#_Toc203047520)

[Gambar 3.8 Tabel 4.1 *Black Box Testing* pada Fitur Registrasi 32](#_Toc203047521)

DAFTAR GAMBAR

[Gambar 2.3 Alur *Request* pada Express (Hahn, 2016:7) 11](#_Toc203048171)

[Gambar 3.2 Gambar 3.2 *Use Case Diagram* 21](#_Toc203048172)

[*Gambar 3.3* Gambar 3.3 *Activity Diagram* 23](#_Toc203048173)

[*Gambar 3.4* Gambar 3.4 *Deployment Diagram* 24](#_Toc203048174)

[*Gambar 3.5* Gambar 3.5 *Entity Relationship Diagram* 25](#_Toc203048175)

[Gambar 3.6 Gambar 3.6 Model Data Fisik 26](#_Toc203048176)

[Gambar 4.1 Halaman Utama pada *Desktop* (a) dan *Mobile* (b) 31](#_Toc203048177)

# DAFTAR LAMPIRAN

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

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

# 

**PENDAHULUAN**

## Latar Belakang

Tindakan kekerasan penjajahan Isreal terhadap Palestina semakin meningkat secara signifikan sejak 7 Oktober 2023, penjajahan tersebut telah berlangsung selama 77 tahun sejak tanggal 14 Mei 1948. Penjajahan ini dimulai dengan perpecahan ideologi antara pemukim israel dan palestina, hingga saat ini telah berkembang menjadi permasalahan global dan geopolitik international. bahkan dalam ranah ekonomi dan sumber daya manusia.

Serangan isreal pada tanggal 7 Oktober 2023 disebut sebagai Operasi Banjir Al-Aqsa. Serangan tersebut merupakan serangan terbesar dalam Sejarah peperangan antara Israel dan Palestina. Operasi Banjir Al-Aqsa bermula dengan pengeboman Isreal ke daerah pemukim palestina, lalu kelompok pejuang pembebasan Palestina (hamas) membalas pengeboman tersebut pada tanggal 6 oktober dan isreal mengumumkan genosida dihari selanjutnya dengan meluncurkan 2000 bom dalam satu hari dengan narasi ‘membela diri’.

Narasi tersebut menyetir banyak media, Dimana banyak berita menyebut pejuang palestina sebagai teroris dan isreal hanya membela diri setelah melakukan pengeboman. Saat ini media adalah kamera dunia. sehingga isreal menggunakan kesempatan tersebut untuk menyebarkan narasi palsu yang juga disebut *hasbara project* Dimana Isreal menyetir media dalam skala besar yang dapat berpengaruh dalam menyetir opini publik.

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, diharapkan penelitian ini dapat memberikan pemahaman tentang pandangan dan sikap pengguna sosial media terhadap isu Palestina khususnya sejak projek Operasi Banjir Al-Aqsa melalui platform media sosial. Hasil penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam perkembangan sentimen pengguna media sosial terhadap isu genosida Palestina.

## Identifikasi Masalah

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

Kebutuhan apa saja yang diperlukan pada tiap XP dalam pengembangan web repositori?

Bagaimana implementasi PWA pada web repositori untuk meningkatkan kualitas web?

Apa pengaruh XP pada proses pengembangan web dan pengaruh implementasi PWA pada kualitas web?

## Batasan Masalah

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

1. Web aplikasi dibangun dengan metode pengembangan perangkat lunak XP*.*
2. Web aplikasi yang dibangun mengimplementasikanPWA fitur *add to homescreen* pada web sehingga dapat digunakan seperti *native app* dan fungsi tampilan saat *offline*.
3. Web aplikasi diakses menggunakan *browser* Chrome atau Mozilla dengan perangkat *desktop* maupun *smartphone* Android agar fungsi PWA yang diimplementasikan dapat berjalan.
4. Target pengguna adalah mahasiswa Teknik Informatika Unpad dan Admin yaitu pegawai tata usaha Teknik Informatika.
5. Materi digital yang dapat diakses pada web repositori adalah skripsi mahasiswa Teknik Informatika Unpad.

## Maksud dan Tujuan Penelitian

Maksud dari penelitian ini adalah melihat hasil dari perancangan repositori skripsi Teknik Informatika Unpad yang mengimplementasikan PWA dengan metode pengembangan perangkat lunak XP.

Tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah:

1. Dapat menghasilkan repositori institusi yang memudahkan mahasiswa Teknik Informatika Unpad untuk mengakses skripsi baik melalui *desktop* maupun *smartphone*.
2. Dapat mengimplementasikan metode XP pada pengembangan web repositori skripsi Teknik Informatika Unpad
3. Dapat mengimplementasikan PWA pada web repositori sehingga web aplikasi yang bertingkah seperti *native application* pada perangkat yang digunakan dan tetap menampilkan informasi disaat *offline.*

## Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Memudahkan mahasiswa Teknik Informatika Unpad untuk mengakses skripsi melalui *desktop* maupun *smartphone*.
2. Menambah pengetahuan dan pembelajaran mengenai metode XP pada pengembangan web repositori.
3. Menghasilkan web yang bertingkah seperti *native application* pada perangkat yang digunakan dan memiliki tampilan disaat *offline* dengan mengimplementasikan PWA*.*

## Metodologi Penelitian

Jenis penelitian yang akan digunakan ialah *Research and Development*, yaitu penelitian dengan menerapkan langkah-langkah yang ada untuk menghasilkan sebuah produk perangkat lunak. Tahapan yang akan dilalui adalah sebagai berikut:

1. Studi literatur. Pada tahap ini penulis mencari referensi dan informasi dari buku, jurnal, maupun artikel yang mendukung penelitian.
2. Penerapan metode XP*.* Pada tahap ini penulis membangun aplikasi sesuai dengan tahapan pada metode XP
   1. Eksplorasi, yaitu menentukan kebutuhan dan membuat *user story.*
   2. Perencanaan, yaitu menetapkan prioritas pengerjaan fitur.
   3. Iterasi untuk dirilis, yaitu melakukan analisis, desain, dan *testing.*
   4. Produksi, yaitu melakukan rilisan kecil untuk mendapatkan *feedback.*
   5. Pemeliharaan, yaitu melakukan pembaharuan dan rilisan.
3. Penulisan laporan. Pada tahap ini penulis menuliskan hasil penelitian ke dalam laporan skripsi.

## 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 dijelaskan seluruh landasan teori yang berhubungan dengan penelitian, yaitu tentang metode pengembangan perangkat lunak yang digunakan, penjelasan teoritis mengenai bahasa pemrograman dan *framework* yang digunakan dalam proses pengimplementasian aplikasi, serta teori lainnya guna memahami permasalahan yang dibahas.

**BAB III ANALISIS DAN PERANCANGAN**

Pada bab ini dijelaskan tentang metode pengembangan aplikasi yang digunakan meliputi analisis kebutuhan sistem, perancangan aplikasi, diagram pemodelan sistem, model perancangan data dan rancangan antarmuka pengguna.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini dijelaskan tentang implementasi aplikasi yang telah dibangun, tampilan aplikasi, pengujian aplikasi, serta hasil dari penggunaan metode XPdan PWA pada web.

**BAB V KESIMPULAN DAN SARAN**

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

# 

**TINJAUAN PUSTAKA**

## Metodologi Penelitian

*Cross Industry Standard Process for Data Mining* (CRISP-DM) adalah model proses standar terbuka yang menggambarkan pendekatan umum yang digunakan oleh pakar data mining (Wirth dan Hipp, 2000). Adapun tahapan dari proses ini dijabarkan 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.

## 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).

Penelitian yang dilakukan oleh (Song *et al*., 2018) dalam papernya yang berjudul *Text Sentiment Analysis Based on Convolutional Neural Network and Bidirectional LSTM Model* menyatakan bahwa metode *supervised learning* menghasilkan performa evaluasi yang signifikan dibandingkan dengan *lexicon based.* Hal ini menunjukkna bahwa penggunaan *machine learning* dapat menghasilkan hasil yang lebih optimal.

## 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

## ReactJS

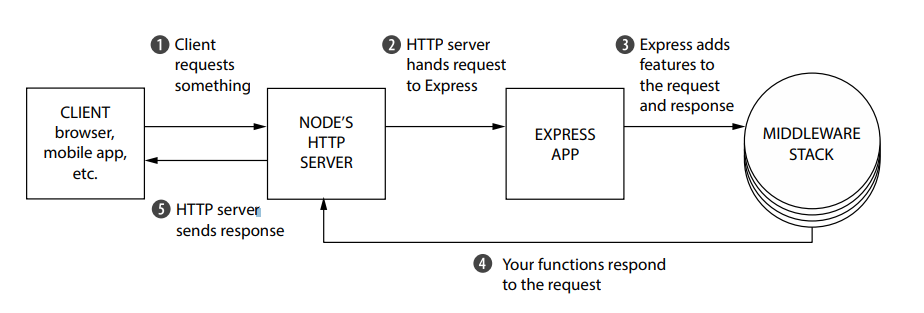
ReactJS adalah kerangka kerja *(framework) open-source* yang fleksibel dan kuat untuk mengembangkan aplikasi sisi klien. ReactJS membutuhkan isyarat dari pengembangan sisi *server* dan menerapkannya pada elemen HTML, hal itu menciptakan pondasi yang memudahkan *developer* membangun aplikasi web (Freeman, 2019:31).

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ……..

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

## Express

Express adalah kerangka kerja relatif kecil yang berada di atas fungsionalitas *server* web Node.js untuk menyederhanakan API dan menambahkan fitur baru yang bermanfaat. Express memudahkan pengaturan fungsionalitas aplikasi dengan *middleware* (fungsi-fungsi penanganan permintaan yang lebih kecil) dan perutean, menambah utilitasuntuk objek HTTP Node.js, memfasilitasi *rendering* tampilan HTML dinamis, mendefinisikan standar yang mudah diimplementasikan (Hahn, 2016:6).



Alih-alih membuat satu fungsi penangan permintaan monolitik yang besar, Express meminta *developer* untuk menulis banyak fungsi yang lebih kecil (banyak di antaranya bisa merupakan fungsi pihak ketiga). Kemudian fungsi dieksekusi untuk setiap permintaan. Fungsi-fungsi penanganan permintaan yang lebih kecil ini disebut fungsi middleware. Express meminimaliskan pengkodean pada Node.js dengan menambahkan fitur-fitur. Hal ini membuat kerja *develope*r menjadi lebih mudah dan sederhana. Express meningkatkan NodeJS seperti halnya Bootstrap untuk HTML / CSS.

## Javascript

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ……..

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

## Bootstrap

…………… ……………. …………..…… …….

* 1. **MySQL**

…………… ……………. …………..…… …….

## Unified Modeling Language (UML)

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

### Use Case Diagram

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ……..

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

Tabel ‎II.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 |

* + 1. ***Activity Diagram***

…………… ……………. …………..…… …….

* + 1. ***Deployment Diagram***

…………… ……………. …………..…… …….

* 1. ***Entity Relationship Diagram (ERD)***

…………… ……………. …………..…… …….

## Usability Testing

…………… ……………. …………..…… …….

* 1. ***Heuristic Evaluation***

…………… ……………. …………..…… …….

* 1. **Skala Likert**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ……..

Tabel ‎II.2 Titik Respons

| **No** | **Pilihan Titik Respons** | **Nilai** |
| --- | --- | --- |
| 1 | Sangat Buruk | 1 poin |
| 2 | Buruk | 2 poin |
| 3 | Cukup | 3 poin |
| 4 | Bagus | 4 poin |
| 5 | Sangat Bagus | 5 poin |

Maka, hasil penilaian dari kuesioner tersebut dapat dihitung dengan menentukan interval dan kriteria nilai terlebih dahulu dengan menggunakan persamaan 2.1.

(2.1)

Berikutnya, untuk menentukan total skor dari setiap butir pertanyaan dapat menggunakan persamaan 2.2.

(2.2)

Karena nilai tertinggi dari pilihan titik respons adalah 5, maka dari itu interpretasi nilai hasil dapat dihitung dengan menggunakan persamaan 2.3.

(2.3)

* 1. **Heroku**

…………… ……………. …………..…… …….

**ANALISIS DAN PERANCANGAN**

Sebagaimana yang telah dijelaskan pada bab sebelumnya, tealh disajikan tinjauan Pustaka yang berkaitan dengan penelitian ino. 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 twwet dengan fokus utama pada…………… ……………. …………..…… …….

## Tahapan Penelitian

### Pengumpulan Data

### Data Balancing

### Pelabelan Data

### Data Preprocessing

### Data Training, Validation dan Predictiing

### Lapisan Word Embedding (Word2vec)

### Lapisan Long Short-Term Memory

### Proses Pembuatan Model LSTM

## Perancangan Aplikasi

### Use Case Diagram

### Activity Diagram

### Sequence Diagram

### Deployment Diagram

### Perancangan Antarmuka

## Analisis Kebutuhan

### Kebutuhan Perangkat Lunak

### Kebutuhan Perangkat Keras

Tabel 3.1 Kebutuhan Sistem Pengguna: Mahasiswa

| **Pengguna : Mahasiswa** | |
| --- | --- |
| **Fitur** | **Deskripsi** |
| Registrasi | Pengguna mendaftarkan diri dengan mengisi nama, NPM, *email*, *password,* dan foto KTM. |
| *Login* | Pengguna *login* dengan mengisi NPM dan *password* |
| Cari Skripsi | Pengguna dapat mencari skripsi berdasarkan judul, nama penulis, kata kunci terkait, dan menyaring skripsi berdasarkan tahun atau kategori skripsi |
| Melihat Skripsi | Pengguna yang telah *login* dapat melihat *file* skripsi |
| Unggah Skripsi | Pengguna dapat mengunggah skripsi |
| Edit *Password* | Pengguna dapat mengubah kata sandi pada halaman profil |
| Status Skripsi | Pengguna dapat mengecek status skripsi apakah telah disetujui admin atau tidak |
| Tampilan *offline* | Web tetap memiliki tampilan pada saat koneksi *offline* |
| *Add to homescreen* | Web dapat ditambahkan ke *homescreen* dan digunakan seperti *native app* |
| Lupa *Password* | Pengguna mendapatkan kata sandi baru yang dikirimkan melalui *email* |
| Kontak admin | Pengguna dapat mengirim pesan ke admin |

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

…………… ……………. …………..…… …….

* + 1. ***User Story***

…………… ……………. …………..…… …….

* 1. **Fase Perencanaan**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

…………… ……………. …………..…… …….

…………… ……………. …………..…… …….

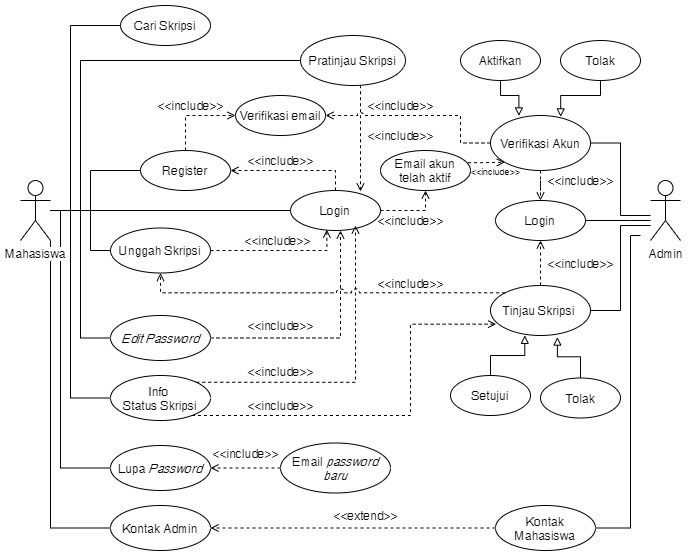
* 1. **Fase Iterasi**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

1. **Analisis Sistem**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

* 1. ***Use Case diagram***

****

Gambar 3.2 *Use Case Diagram*

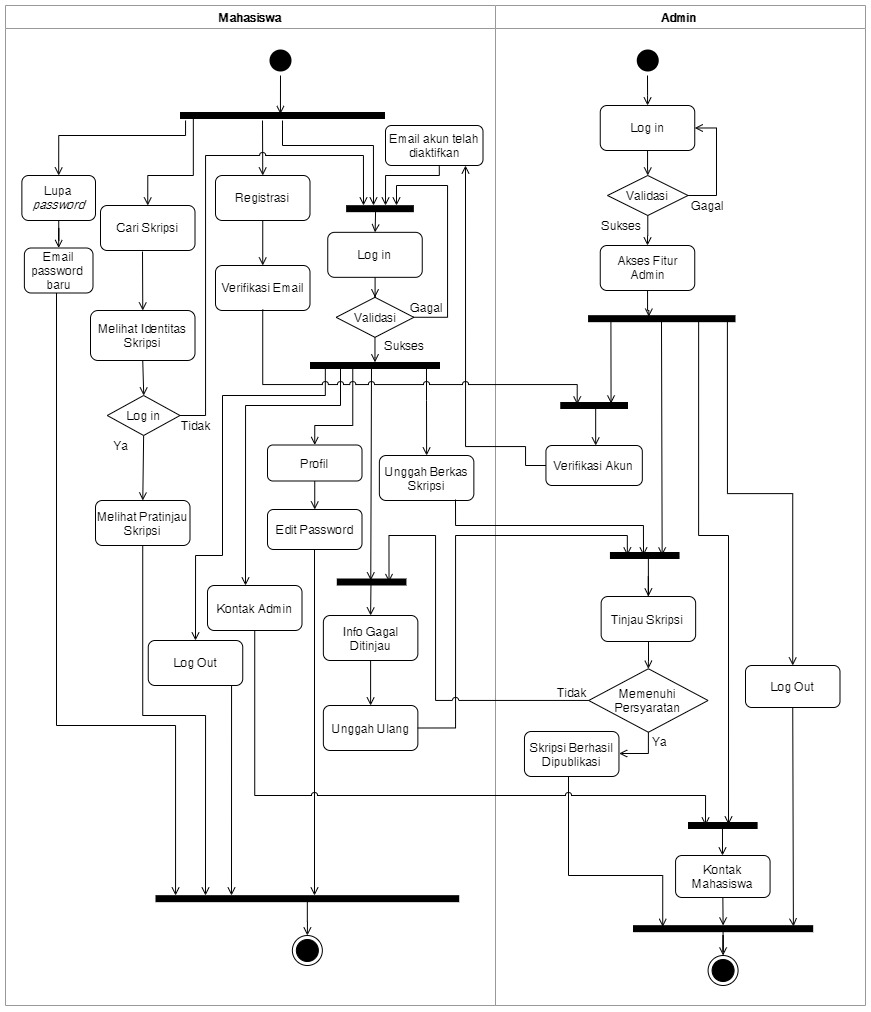
*Use* *case* *diagram* menjelaskan kelakuan dari sistem web repositori skripsi Teknik Informatika Unpad. Pada Gambar 3.2*,* terdapat dua aktor yang merepresentasikan dua jenis pengguna, yaitu mahasiswa dan admin. Mahasiswa dapat melakukan registrasi akun, *login,* pencarian skripsi, melihat skripsi, unggah skripsi, *edit password*, dan melihat info tinjauan skripsi. Pencarian skripsi dapat dilakukan tanpa *login,* sementara fitur-fitur lainnya dapat dilakukan setelah *login*. *Login* dapat dilakukan apabila mahasiswa telah mendaftarkan akun dan akun telah diaktifkan oleh admin. Admin dapat melakukan verifikasi akun dan tinjau skripsi apabila admin telah *login.*

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

* 1. ***Activity diagram***

Diagram aktivitas menggambarkan aliran kerja dari suatu sistem. Pada Gambar 3.3, terdapat dua *swimlane* dengan label mahasiswa dan admin. *Swimlane* memisahkan aktivitas berdasarkan pengguna. Mulai dari status awal, mahasiswa dapat melakukan tiga aktivitas yaitu registrasi, *login*, mencari skripsi. Akun yang didaftarkan akan ditinjau oleh admin terlebih dahulu. Apabila memenuhi syarat maka akun akan diaktifkan. Setelah diaktifkan, mahasiswa dapat *login.*

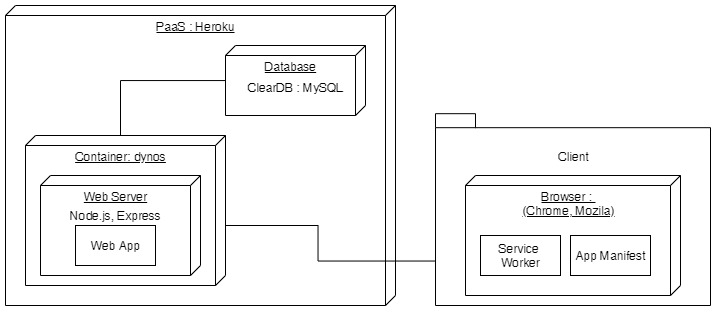
……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….



Gambar 3.3 *Activity Diagram*

1. **Analisis Arsitektur Menggunakan *Deployment Diagram***

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….



Gambar 3.4 *Deployment Diagram*

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

1. **Desain Basis Data**

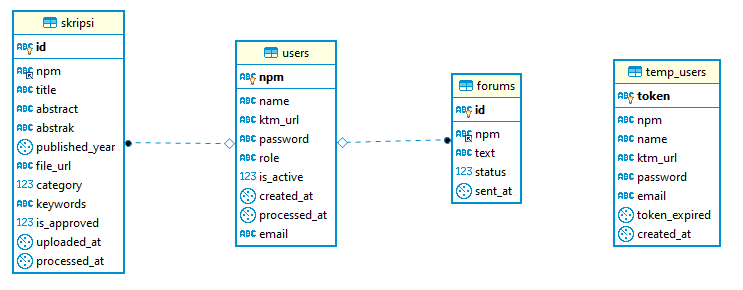
Pada Gambar 3.5 terdapat empat pada entitas dalam ERD yaitu *users, temp\_user,* skripsi, dan *forums*. Tabel *forums* ditambahkan karena hasil *feedback* pada fase pemeliharaan yang menambahkan fitur kontak admin. Tabel *forums* berisi data pesan yang dikirimkan pada fitur kontak admin. Gambar 3.5 adalah ERD yang menjelaskan *database* web aplikasi hasil final rilis.

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

# ERD (2)

Gambar 3.5 *Entity Relationship Diagram*

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….



Gambar 3.6 Model Data Fisik

Tabel 3.5 menjelaskan mengenai tipe data, ukuran, dan keterangan dari tiap atribut pada tabel skripsi yang ada pada *database.* Tabel skripsi berisi data dari skripsi yang diunggah oleh mahasiswa*.* Data tersebut terdiri dari id skripsi, judul, abstrak dalam bahasa inggris dan indonesia, *url file* skripsi, kategori, kata kunci, tahun publikasi, waktu diunggah dan ditinjau oleh admin*,* status unggahan skripsi, serta NPM dari penggunaterkait.

Tabel ‎III.2 Atribut Tabel skripsi

| **No** | **Atribut** | **Tipe** | **Ukuran** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | id | varchar | 5 | *Primary key* sebagai kode identitas skripsi. |
| 2 | npm | varchar | 13 | *Foreign key* dari tabel *users* |
| 3 | title | varchar | 255 | Judul skripsi |
| 4 | abstract | text | - | Abstrak skripsi dalam bahasa inggris |
| 5 | abstrak | text | - | Abstrak skripsi dalam bahasa indonesia |
| 6 | file\_url | varchar | 70 | Url *file* skripsi |
| 7 | category | tinyint | 1 | Kategori skripsi berdasarkan bidang minat. Nilai 1 adalah Sistem Cerdas dan Sistem Grafika (SCSG), 2 adalah Sistem Informasi dan Rekayasa Perangkat Lunak (SIRPL), 3 adalah Minat Jaringan Komputer dan Komunikasi Data (JKKD), dan 4 adalah Ilmu Komputasi dan Metode Numerik (IKMN) |
| 8 | keywords | varchar | 255 | Kata kunci terkait dengan skripsi untuk memudahkan pencarian |
| 9 | published\_year | year | - | Tahun skripsi dipublikasikan |
| 10 | is\_approved | tinyint | 1 | Terdiri dari 0 (ditolak admin), 1 (disetujui oleh admin), dan 2 belum di proses. *Default* awal adalah 2 |
| 11 | uploaded\_at | timestamp | - | Waktu skripsi diunggah |
| 12 | processed\_at | timestamp | - | Waktu skripsi ditinjau oleh admin |

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

1. **Desain Antarmuka**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

……. ……….. …………. ……… …………… . ………… …….. …… …..

……. ……….. …………. ……… …………… . ………… …….. …… …..

……. ……….. …………. ……… …………… . ………… …….. …… …..

1. ***Testing***

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

……. ……….. …………. ……… …………… . ………… …….. ……

……. ……….. …………. ……… …………… . ………… …….. ……

……. ……….. …………. ……… …………… . ………… …….. ……

* 1. **Fase Produksi**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

……. ……….. …………. ……… …………… . ………… …….. ……

……. ……….. …………. ……… …………… . ………… …….. ……

……. ……….. …………. ……… …………… . ………… …….. ……

* 1. **Fase Pemeliharaan dan Fase Akhir**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

……. ……….. …………. ……… …………… . ………… …….. ……

……. ……….. …………. ……… …………… . ………… …….. ……

……. ……….. …………. ……… …………… . ………… …….. ……

# BAB IV

**HASIL DAN PEMBAHASAN**

1. **Impelementasi Program**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… ... …… …….

1. **Halaman Utama**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… . …… …….



(a)



(b)

Gambar 4.1 Halaman Utama pada *Desktop* (a) dan *Mobile* (b)

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ………

Script/kode program halaman utama :

|  |
| --- |
| router.get('/list', (req, res) =>{  let sql = `SELECT skripsi.id, skripsi.title, skripsi.published\_year, skripsi.category, skripsi.keywords, users.name FROM skripsi join users on users.npm = skripsi.npm where is\_approved=${1} ORDER BY published\_year desc, skripsi.processed\_at desc`  db.query(sql, (err, result)=>{  if (err) console.log(err)  res.send(result)})  }) |

*Request* dipanggil dalam *componentDidMount* *method*, yang berarti *request* dilakukan setelah inisial *rendering* dilakukan. Berikut adalah kode *request* data skripsi pada *frontend:*

Script/kode program *request* :

|  |
| --- |
| getSkripsi=()=>{ //fungsi request data skripsi menggunakan axios  axios({   method: 'get',    url: '/skripsi/list',   }).then(res=>{  this.setState({     skripsi: res.data,     isLoaded: true,     skripsiFiltered:res.data,     skripsiFilteredTemp:res.data,     skripsiFilteredCat:res.data,     skripsiFilteredYear:res.data,     years: [...new Set(res.data.map((year)=>{        return year.published\_year      }))].sort()  })    localStorage.setItem('list', JSON.stringify(res.data))   }).catch((err) => {    if(err.response) console.log(err.response)   })  }  componentDidMount(){  if (navigator.onLine){ //Saat online, lakukan request data    this.getSkripsi()    this.setState({     offline:false    })  }   else{ //Saat offline ambil data dari local storage    if (localStorage.getItem('list')){     let data = JSON.parse(localStorage.getItem('list'))     this.setState({      skripsi: data,      isLoaded: true,      skripsiFiltered:data,      skripsiFilteredTemp:data,      skripsiFilteredCat:data,      skripsiFilteredYear:data,      years: [...new Set(data.map((year)=>{              return year.published\_year             }))].sort()     })    } }  } |

Data yang didapat dikirimkan sebagai *props* ke *component list* untuk di-*loop* dan ditampilkan ke dalam bentuk *card* pada halaman utama. Pada kode dibawah dapat dilihat bahwa data yang dikirimkan merupakan data hasil *filter* sehingga tiap pencarian maupun penyaringan akan mengubah data yang tampil. Fungsi filter akan dibahas lebih lanjut pada bagian fitur pencarian dan penyaringan.

Script/kode program *filter* :

|  |
| --- |
| const currentPosts = skripsiFiltered.slice(indexOfFirstPost, indexOfLastPost)  <ListCard skripsi={currentPosts} isLoaded={isLoaded}> </ListCard> |

1. **Halaman Registe****r**

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ………

Tabel ‎4.1 *Black Box Testing* pada Fitur Registrasi

| **Skenario** | **Hasil yang diharapkan** | **Hasil** |
| --- | --- | --- |
| Tidak memberi masukan apapun pada *form* data diri | Pengguna tidak dapat melanjutkan ke bagian dua karena tombol lanjuttidak aktif | Berhasil |
| Mengosongkan salah satu *field* | Pengguna tidak dapat melanjutkan ke bagian dua karena tombol lanjuttidak aktif | Berhasil |
| Mengisi NPM atau *email* yang telah terdaftar | Muncul pesan bahwa NPM atau *email* telah terdaftar setelah pengguna mengklik tombol lanjut | Berhasil |
| Mengisi NPM yang salah (bukan angka, !=12 digit) | Muncul pesan bahwa NPM harus 12 digit setelah pengguna selesai menulis NPM dan tombol lanjut tidak aktif | Berhasil |
| Tidak mengisi kolom konfirmasi *password* dengan benar | Muncul pesan bahwa *password* tidak cocok. Pengguna tidak dapat melanjutkan ke bagian dua karena tombol lanjuttidak aktif | Berhasil |
| Mengisi semua *field* data diri dengan benar | Pengguna dapat lanjut ke bagian dua | Berhasil |
| Tidak mengunggah foto KTM | Muncul pesan bahwa *file* tidak boleh kosong | Berhasil |
| Mengunggah *file* yang bukan jpeg atau png | Muncul pesan bahwa *file* harus png, jpg, atau jpeg | Berhasil |
| Menggunggah *file* yang lebih besar dari 5mb | Muncul pesan bahwa *file* terlalu besar | Berhasil |
| Mengunggah *file* foto KTM dengan format jpg/png dengan besar lebih kecil dari 5MB | Muncul pesan bahwa pengguna harus melakukan verifikasi melalui *link* yang dikirimakan ke *email*. *Email* dikirimkan ke pengguna. | Berhasil |
| Mengakses *link* yang dikirimkan ke *email* | Muncul pesan bahwa register berhasil dan pengguna diharap menunggu pengaktifan dari admin. Data pengguna tersimpan di tabel *users* dan dihapus dari *temp\_users* | Berhasil |

……. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ……… …………… . ………… …….. …… ….. ……… …………… . ………… …….. …… ….. ……….. …………. ……… …………… . ………… …….. …… ….. …. …… …. …. …. …. ……. .. …… ………

1. **Menu *Login***

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Unggah Skripsi**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Menu Admin**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Verifikasi Akun**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Tinjau Skripsi**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Detail Skripsi**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Fitur Pencarian dan Penyaringan**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Profil danFitur *Edit Password***

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Halaman Status Skripsi**

……. ……….. …………. ……… …………… . ………… …….. …… ………..

……. ……….. …………. ……… …………… . ………… …….. …… ………..

1. **Implementasi PWA**

……. ……….. …………. ……… …………… . ………… …….. …… ………..

……. ……….. …………. ……… …………… . ………… …….. …… ………..

1. **Fase Produksi**

……. ……….. …………. ……… …………… . ………… …….. …… ……….. ……. ……….. …………. ……… …………… . ………… …….. …… ………..

1. **Rilisan Kecil**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Pengujian**

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Fase Pemeliharaan dan Fase Akhir**

……. ……….. …………. ……… …………… . ………… …….. ……

1. ***Feedback***

……. ……….. …………. ……… …………… . ………… …….. ……

1. **Implementasi *Feedback***

……. ……….. …………. ……… …………… . ………… …….. ……

# 

**KESIMPULAN DAN SARAN**

1. **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. **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).

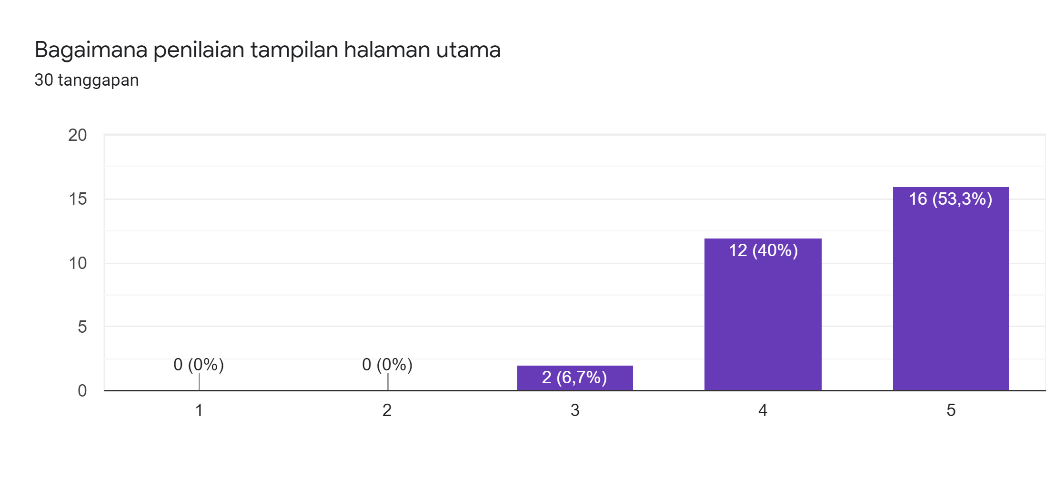
Singh, H. and Bhatt, M. (2016) *Learning Web Development with React and Bootstrap*. Birmingham: Packt Publishing Ltd.

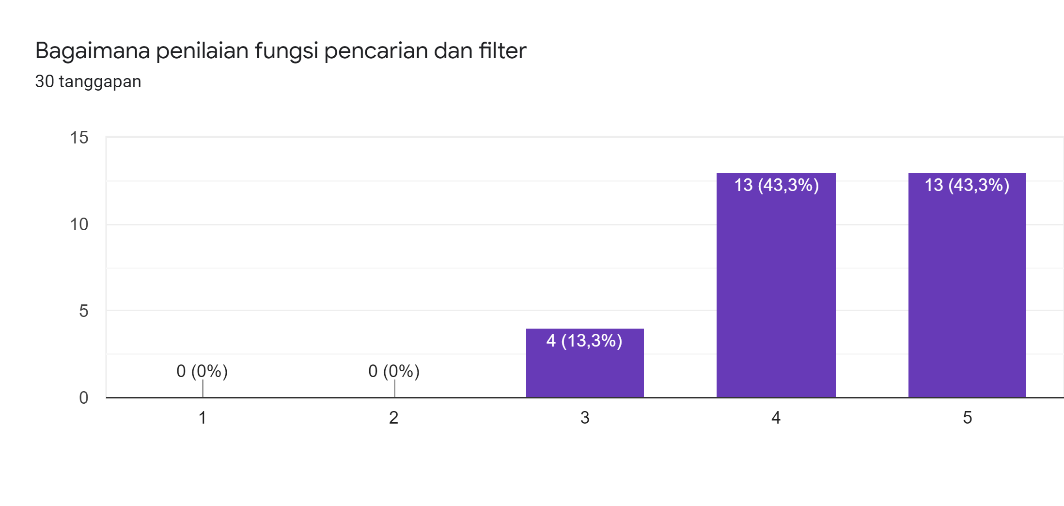
Spillner, A., *et al.* (2014) *Software Testing Foundations*. Santa Barbara: Rocky Nook Inc.

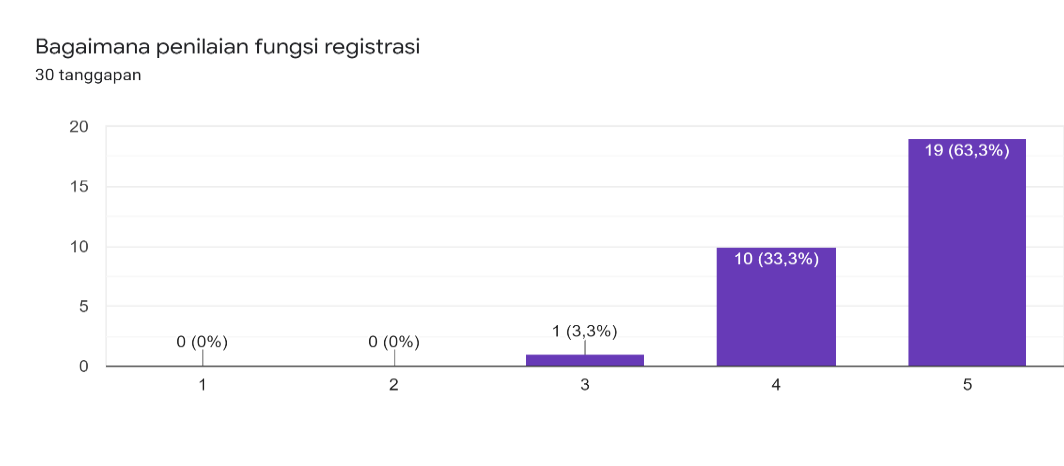
Sutedjo, M. (2014) ‘Pengelolaan Repositori Perguruan Tinggi dan Pengembangan Repositori Karya Seni’, *Makalah Seminar Nasional ”Digital Local Content: Strategi Membangun Repository Karya Seni”, GKU FSR ISI Yogyakarta*.

# 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