

# **PENERAPAN JAMSTACK ARCHITECTURE PADA SISTEM INFORMASI PELAYANAN PUBLIK BRIDA NTB**

Tugas Akhir  
Untuk memenuhi sebagai persyaratan  
Mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh:

**AGUNG DWITATWA ACHMADI  
F1D017003**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK  
UNIVERSITAS MATARAM  
NOVEMBER 2023**

## TUGAS AKHIR

### PENERAPAN JAMSTACK ARCHITECTURE PADA SISTEM INFORMASI PELAYANAN PUBLIK BRIDA NTB

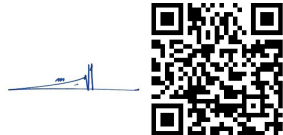
Oleh :

**AGUNG DWITATWA ACHMADI**

F1D017003

Telah diperiksa oleh Tim Pembimbing :

1. Pembimbing Utama



Tanggal: 09/11/2023

Moh. Ali Albar, ST., M.Eng.  
NIP. 198311252015041002

2. Pembimbing Pendamping



Tanggal: 13/11/2023

Noor Alamsyah, ST., MT.  
NIP. 199005052020121017

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Teknik  
Universitas Mataram



Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.  
NIP. 197311302000031001

## TUGAS AKHIR

### PENERAPAN JAMSTACK ARCHITECTURE PADA SISTEM INFORMASI PELAYANAN PUBLIK BRIDA NTB



Oleh :

**AGUNG DWITATWA ACHMADI**  
F1D017003

Telah diujikan di depan penguji  
Pada tanggal 19 Mei 2023

Susunan Tim Penguji :

1. Penguji 1



Tanggal: 13/11/2023

Dwi Ratnasari, S.Kom., M.T.  
NIP. 198509012019032010


2. Penguji 2



Tanggal: 08/11/2023

Nadiyah Agitha, S.Kom., M.MT  
NIP. 198608132018032001

3. Penguji 3



Tanggal: 14/11/2023

Royana Afwani, ST., MT.  
NIP. 198507072014042001

Mataram, 15 November 2023  
Dekan Fakultas Teknik  
Universitas Mataram



Muhamad Syamsu Iqbal, S.T., M.T., Ph.D.  
NIP. 19731130200003100

## **HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Saya menyatakan bahwa tugas akhir ini dengan judul “PENERAPAN JAMSTACK ARCHITECTURE PADA SISTEM INFORMASI PELAYANAN PUBLIK BRIDA NTB” sepenuhnya adalah karya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan saya tidak melakukan penjiplakan atau pengutipan dengan cara yang tidak sesuai dengan etika keilmuan yang berlaku. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Mataram, 14 November 2023

Yang membuat pernyataan

**Agung Dwitatwa Achmadi**

## **PRAKATA**

*Assalamualaikum Wr. Wb.*

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan karunia-NYA sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Sistem Informasi Point of Sales Pada Badan Usaha Milik Desa Mambalan Berbasis Website Menggunakan Metode Agile Modelling” tepat pada waktunya. Pada kesempatan ini pula, penulis menghaturkan terimakasih yang sebesar-besarnya kepada seluruh pihak yang telah mendukung agar terselesaikanya Tugas Akhir ini.

Penulis tentu menyadari Tugas Akhir ini masih jauh dari kata sempurna dan masih terdapat kesalahan dan kekurangan di dalamnya. Untuk itu, diharapkan kritik serta saran dari pembaca untuk Tugas Akhir ini, agar Tugas Akhir ini dapat menjadi lebih baik lagi. Penulis juga mengucapkan banyak-banyak terimakasih kepada semua pihak yang telah membantu penulis sehingga Tugas Akhir ini dapat selesai.

Demikian yang bisa disampaikan, sekali lagi terimakasih atas semua pihak yang telah membantu di dalam penyelesaian Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

*Wassalamualaikum Wr. Wb.*

Mataram, 14 November 2023

## UCAPAN TERIMAKASIH

Penulis menyadari bahwa terselesaikannya Tugas Akhir ini tentunya bukan hanya dari usaha penulis saja. Tugas Akhir ini bisa selesai tepat waktu tentunya berkat dukungan dari semua pihak yang terlibat juga. Oleh karena itu, pada kesempatan ini penulis menghaturkan terimakasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, yang telah memberikan berkat dan kesempatan untuk terus bersyukur bisa berada pada tahapan ini.
2. Kedua orangtua dan keluarga yang selama ini telah memberikan doa dan dukungannya dalam menjalani perkuliahan dan pengerjaan Tugas Akhir ini sehingga dapat terselesaikan.
3. Bapak Moh. Ali Albar, S.T, M.Eng. selaku dosen pembimbing utama yang telah memberikan bimbingan, motivasi, semangat dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
4. Bapak Noor Alamsyah, ST., MT. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
5. Dosen penguji, atas pemberian kritik dan saran yang bersifat membangun serta diperlukan dalam penyelesaian Tugas Akhir ini.
6. Teman-teman terdekat penulis selama menjalani perkuliahan di Program Studi Teknik Informatika selaku *support system* penulis dan telah membantu dalam penyelesaian tugas akhir ini.
7. Badan Riset dan Inovasi Daerah Provinsi NTB selaku *client* pada Tugas Akhir ini.
8. Responden ketika pengumpulan kebutuhan dan penelitian yang telah meluangkan waktunya sehingga saran dan masukannya berguna untuk pengembangan sistem.
9. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan do'a dan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik.

## ABSTRAK

Digitalisasi pelayanan publik di Indonesia telah dilakukan oleh lembaga pemerintahan, termasuk Organisasi Perangkat Daerah (OPD) seperti Badan Riset dan Inovasi Daerah (BRIDA) NTB. BRIDA NTB bekerja sama dengan Badan Kesatuan Bangsa dan Politik Dalam Negeri (BAKESBANGPOLDAGRI) untuk memberikan pelayanan izin penelitian. Namun, aplikasi website yang digunakan saat ini memiliki beberapa kekurangan dalam hal keamanan, performa, responsifitas, dan kemudahan pengembangan. Solusi yang diusulkan adalah menerapkan arsitektur JAMStack yang memberikan keunggulan dalam keamanan, peningkatan performa, efisiensi biaya, dan skalabilitas pengembangan. Selain itu, terdapat keterbatasan dalam rincian data penelitian yang menghambat proses validasi. Untuk mengatasi permasalahan ini, diperlukan perancangan dan pembangunan sistem baru dengan menerapkan arsitektur JAMStack dan penyesuaian sesuai kebutuhan. Metode *fountain* dipilih sebagai kerangka acuan dalam pembangunan sistem ini karena memberikan fleksibilitas yang sulit dicapai oleh metode *waterfall*. Untuk melihat persentase kesuksesan penerapan arsitektur ini, peneliti melakukan pengujian menggunakan metode *blackbox testing* dan *user acceptance testing* (UAT). Dari hasil pengujian menggunakan metode *blackbox testing* didapati bahwa semua fitur yang terdapat pada sistem dapat berjalan dengan baik. Lalu hasil pengujian menggunakan UAT dari total 20 responden didapatkan hasil sebesar 86 % pada sisi *admin* dan 84,51 % pada sisi pengguna dari mahasiswa dan masyarakat umum.

**Kata kunci** – Arsitektur JAMStack, Client Side Rendering (CSR), Static Site Generation (SSG), Javascript, Nextjs.

## DAFTAR ISI

|                                    |      |
|------------------------------------|------|
| ABSTRAK .....                      | v    |
| DAFTAR ISI.....                    | viii |
| DAFTAR GAMBAR .....                | x    |
| DAFTAR TABEL.....                  | xii  |
| BAB I PENDAHULUAN .....            | 1    |
| 1.1 Latar Belakang.....            | 1    |
| 1.2 Rumusan Masalah .....          | 3    |
| 1.3 Batasan Masalah.....           | 3    |
| 1.4 Tujuan Penelitian.....         | 3    |
| 1.5 Manfaat Penelitian .....       | 4    |
| 1.6 Sistematika Penulisan .....    | 4    |
| BAB II TINJAUAN PUSTAKA.....       | 5    |
| 2.1 Penelitian Terkait .....       | 5    |
| 2.2 Teori Penunjang.....           | 8    |
| 2.2.1 JAMStack Architecture .....  | 8    |
| 2.2.2 Client Side Rendering.....   | 9    |
| 2.2.3 Static Site Generation ..... | 9    |
| 2.2.4 Headless CMS.....            | 9    |
| 2.2.5 NextJS.....                  | 10   |
| 2.2.6 REST API.....                | 10   |
| 2.2.7 Javascript .....             | 11   |
| 2.2.8 Blackbox Testing .....       | 11   |
| 2.2.9 Metode Fountain.....         | 11   |
| BAB III METODOLOGI PENELITIAN..... | 13   |
| 3.1 Alat dan Bahan .....           | 13   |
| 3.1.1 Alat .....                   | 13   |
| 3.1.2 Bahan .....                  | 13   |
| 3.2 Metode Pengembangan .....      | 13   |
| 3.2.1 Analysis.....                | 14   |
| 3.2.2 Requirements.....            | 16   |
| 3.2.3 Design.....                  | 17   |



|  |           |
|--|-----------|
| 3.2.4 Coding .....                       | 36        |
| 3.2.5 Testing .....                      | 37        |
| 3.2.6 Maintenance.....                   | 37        |
| <b>BAB IV HASIL DAN PEMBAHASAN .....</b> | <b>38</b> |
| 4.1 Implementasi Arsitektur .....        | 38        |
| 4.2 Requirements .....                   | 39        |
| 4.2.1 NextJS.....                        | 39        |
| 4.2.2 Strapi.....                        | 41        |
| 4.3 Implementasi Coding Frontend.....    | 42        |
| 4.3.1 Sitemap Sistem .....               | 43        |
| 4.3.2 Pembuatan Komponen.....            | 44        |
| 4.3.3 Hasil Implementasi Frontend .....  | 45        |
| 4.4 Implementasi Backend .....           | 53        |
| 4.5 Testing .....                        | 58        |
| 4.5.1 Pengujian Mandiri.....             | 58        |
| 4.5.2 Blackbox Testing .....             | 59        |
| 4.5.3 User Acceptance Testing (UAT)..... | 66        |
| 4.6 Maintenance .....                    | 69        |
| <b>BAB V KESIMPULAN DAN SARAN.....</b>   | <b>70</b> |
| 5.1 Kesimpulan .....                     | 70        |
| 5.2 Saran .....                          | 70        |
| <b>DAFTAR PUSTAKA.....</b>               | <b>72</b> |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Diagram Fishbone berdasarkan tinjauan ublic .....                    | 7  |
| Gambar 2.2 Perbandingan arsitektur tradisional dengan arsitektur jamstack ..... | 8  |
| Gambar 2.3 Perbedaan Headless, Monolithic dan Decoupled Architecture.....       | 10 |
| Gambar 2.4 Tahapan metode fountain. ....  | 12 |
| Gambar 3.1 Metode penelitian.....   | 14 |
| Gambar 3.2 Hasil analisa sistem sebelumnya.....                                 | 16 |
| Gambar 3.4 Gambar arsitektur Jamstack .....                                     | 17 |
| Gambar 3.5 Wireframe design laman register. ....                                | 20 |
| Gambar 3.6 Wireframe design laman login. ....                                   | 20 |
| Gambar 3.7 Wireframe design dari laman landing. ....                            | 21 |
| Gambar 3.8 Wireframe design dari laman penelitian terdaftar.....                | 21 |
| Gambar 3.9 Wireframe design dari laman dashboard.....                           | 22 |
| Gambar 3.10 Wireframe design laman alur pengisian.....                          | 22 |
| Gambar 3.11 Wireframe design laman identitas peneliti.....                      | 23 |
| Gambar 3.12 Wireframe design laman data penelitian.....                         | 23 |
| Gambar 3.13 Wireframe design laman pengajuan penelitian. ....                   | 24 |
| Gambar 3.14 Wireframe design laman dashboard admin.....                         | 24 |
| Gambar 3.15 Diagram konteks <i>data flow diagram</i> (DFD). ....                | 25 |
| Gambar 3.16 <i>data flow diagram</i> (DFD) level 1.....                         | 26 |
| Gambar 3.17 DFD level 2 proses pengajuan surat rekomendasi penelitian. ....     | 26 |
| Gambar 3.18 DFD level 2 proses pengajuan surat izin penelitian. ....            | 27 |
| Gambar 3.19 DFD level 2 proses perilisan surat izin penelitian.....             | 27 |
| Gambar 3.20 <i>Flowchart</i> login. ....  | 28 |
| Gambar 3.21 <i>Flowchart</i> registrasi akun.....                               | 29 |
| Gambar 3.22 <i>Flowchart</i> melihat data penelitian terdaftar. ....            | 30 |
| Gambar 3.23 <i>Flowchart</i> melihat alur pengisian.....                        | 30 |
| Gambar 3.24 <i>Flowchart</i> pengajuan surat rekomendasi penelitian.....        | 31 |
| Gambar 3.25 <i>Flowchart</i> pengajuan surat izin penelitian. ....              | 32 |
| Gambar 3.26 <i>Flowchart</i> verifikasi pengajuan oleh BAKESBANG POLDAGRI. 33   |    |

|   |    |
|---|----|
| Gambar 3.27 <i>Flowchart</i> verifikasi pengajuan surat izin penelitian. .... | 34 |
| Gambar 3.28 <i>Flowchart</i> fitur cetak surat izin penelitian. ....          | 35 |
| Gambar 3.29 Gambar <i>Entity Relationship Diagram (ERD)</i> .....             | 36 |
| Gambar 4. 1 Sitemap sistem informasi pelayanan public BRIDA NTB.....          | 43 |
| Gambar 4. 2 Implementasi <i>file based routing</i> .....                      | 43 |
| Gambar 4. 3 Implementasi <i>file based routing</i> .....                      | 43 |
| Gambar 4. 4 Contoh pembuatan komponen.....                                    | 44 |
| Gambar 4. 5 <i>Source code</i> pembuatan komponen.....                        | 44 |
| Gambar 4. 6 Pembuatan komponen khusus untuk masing masing laman. ....         | 45 |
| Gambar 4. 7 Pembuatan komponen global .....                                   | 45 |
| Gambar 4. 8 Tampilan laman login.....   | 46 |
| Gambar 4. 9 Tampilan laman registrasi. ....                                   | 46 |
| Gambar 4. 10 Tampilan landing page. ....                                      | 47 |
| Gambar 4. 11 Tampilan laman penelitian terdaftar. ....                        | 48 |
| Gambar 4. 12 Tampilan laman detail penelitian terdaftar.....                  | 48 |
| Gambar 4. 13 Tampilan laman dashboard peneliti. ....                          | 49 |
| Gambar 4. 14 Tampilan laman petunjuk pengisian sistem .....                   | 49 |
| Gambar 4. 15 Tampilan laman identitas peneliti. ....                          | 50 |
| Gambar 4. 16 Tampilan laman data penelitian. ....                             | 51 |
| Gambar 4. 17 Tampilan laman pengajuan penelitian.....                         | 51 |
| Gambar 4. 18 Laman dashboard admin BAKESBANGPOLDAGRI.....                     | 52 |
| Gambar 4. 19 laman dashboard admin BRIDA NTB. ....                            | 52 |
| Gambar 4. 23 Skema data penelitian <i>collection</i> .....                    | 53 |
| Gambar 4. 24 Skema identitas peneliti <i>collection</i> .....                 | 54 |
| Gambar 4. 25 Skema pengajuan <i>collection</i> .....                          | 55 |
| Gambar 4. 26 Skema surat izin <i>collection</i> . ....                        | 56 |
| Gambar 4. 27 Skema surat rekomendasi <i>collection</i> . ....                 | 56 |
| Gambar 4. 28 Skema <i>user collection</i> . ....                              | 57 |

## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 3. 1 Perbandingan antara sistem lama dengan sistem baru.....                   | 19 |
| Tabel 4. 1 Pengujian fitur login.....  | 59 |
| Tabel 4. 2 Pengujian fitur register.....   | 60 |
| Tabel 4. 3 Pengujian fitur melihat penelitian terdaftar .....                        | 61 |
| Tabel 4. 4 Pengujian fitur melihat petunjuk pengisian .....                          | 61 |
| Tabel 4. 5 Pengujian fitur pengajuan surat rekomendasi penelitian .....              | 62 |
| Tabel 4. 6 Pengujian fitur pengajuan surat izin penelitian.....                      | 62 |
| Tabel 4. 7 Pengujian fitur verifikasi surat rekomendasi penelitian.....              | 63 |
| Tabel 4. 8 Pengujian verifikasi surat izin penelitian .....                          | 64 |
| Tabel 4. 9 Pengujian fitur cetak surat izin penelitian.....                          | 65 |
| Tabel 4. 10 Tingkat penilaian responden. ....  | 66 |
| Tabel 4. 11 Klasifikasi hasil pengujian .....  | 67 |
| Tabel 4. 12 Hasil penilaian responden dari mahasiswa dan masyarakat umum ...         | 67 |
| Tabel 4. 13 Hasil penilaian responden dari admin badan. ....                         | 67 |
| Tabel 4. 14 Bobot penilaian dari pernyataan mahasiswa dan masyarakat.....            | 68 |
| Tabel 4. 15 Bobot penilaian dari pernyataan admin .....                              | 68 |
| Tabel 4. 16 Hasil perhitungan dari responden mahasiswa dan masyarakat umum.<br>..... | 68 |
| Tabel 4. 17 Hasil perhitungan dari responden admin badan.....                        | 69 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di Indonesia, proses digitalisasi sudah mulai banyak dilakukan oleh lembaga – lembaga pemerintahan, mulai dari pemerintah pusat hingga Pemerintah Daerah. Hal ini dilakukan sesuai dengan instruksi Presiden No 3 Tahun 2003 tentang Kebijakan dan Strategi Nasional Pengembangan E-Government [1]. Banyak dari lembaga – lembaga pemerintahan mulai mengubah pelayanan birokrasi yang sebelumnya dilaksanakan secara manual (*offline*) menjadi digital (*online*). Migrasi pelayanan menuju digitalisasi ini tentunya membutuhkan implementasi yang tepat, guna menunjang kegiatan birokrasi sehingga memberikan pelayanan terbaik kepada masyarakat. Kemudahan dalam mengakses informasi dan pelayanan tentunya menjadi hal penting yang harus diperhatikan, berikut juga kecepatan akses yang menjadi fitur penting dalam digitalisasi sehingga dapat mengikuti perkembangan teknologi yang serba cepat saat ini [2].

Kualitas pelayanan dari masing – masing lembaga pemerintahan dituntut untuk cepat dan tepat kepada masyarakat, hal ini sesuai dengan Peraturan Pemerintah Nomor 96 Tahun 2012 tentang Pelaksanaan Undang – Undang Nomor 25 Tahun 2009 tentang Pelayanan Publik, dimana setiap aparatur pemerintahan dituntut untuk memberikan pelayanan terbaik kepada masyarakat [3]. Untuk meningkatkan kualitas pelayanan tersebut, banyak dari lembaga pemerintahan mulai menerapkan teknologi informasi dalam bentuk aplikasi untuk memberikan pelayanan terbaik kepada masyarakat. Hal ini juga dilakukan oleh Organisasi Perangkat Daerah (OPD) yang ada di daerah Nusa Tenggara Barat [4].

Badan Riset dan Inovasi Daerah (BRIDA) NTB merupakan OPD baru yang dibentuk pada bulan Januari Tahun 2022 berdasarkan peraturan daerah No. 14 Tahun 2021. Sebagai OPD baru tentunya banyak hal yang harus dipersiapkan oleh BRIDA NTB untuk memberikan pelayanan terbaik kepada masyarakat. Salah satu pelayanan publik yang diberikan oleh BRIDA NTB adalah pelayanan izin penelitian. Dalam memberikan pelayanan izin penelitian ini, BRIDA NTB

melakukan kerja sama dengan Badan Kesatuan Bangsa dan Politik Dalam Negeri (BAKESBANGPOLDAGRI) untuk melakukan validasi terhadap pengajuan izin penelitian sebelum nantinya divalidasi kembali dan diterbitkan surat izin penelitian oleh BRIDA NTB.

Pelayanan izin penelitian ini sebelumnya telah dibuat dalam bentuk aplikasi *website* dengan menerapkan arsitektur MVC Tradisional. Konsep MVC ini membagi aplikasi menjadi 3 bagian yang terpisah secara logika tetapi tetap saling berhubungan [5]. Dalam penerapannya, aplikasi pelayanan izin penelitian ini masih memiliki kekurangan baik itu dari sisi pengguna, *admin*, maupun dari sisi pengembang Aplikasi. Kekurangan tersebut mulai dari tingkat keamanan, performa aplikasi, responsifitas hingga kemudahan bagi pengembang dalam mengembangkan aplikasi sesuai dengan kebutuhan masyarakat. Solusi yang dapat dilakukan untuk menangani permasalahan ini adalah dengan menerapkan Arsitektur JAMStack.

JAMStack merupakan salah satu arsitektur pengembangan aplikasi *website* modern yang telah berkembang pesat saat ini semenjak digunakan pertama kali pada tahun 2015. Kata JAMstack merupakan singkatan dari *Javascript*, *API*, dan *Markup* [6]. Dalam JAMStack pembuatan aplikasi terbagi menjadi bagian *Frontend* dan *Backend* yang saling terhubung melalui *API*. JAMStack memberikan kelebihan dari segi kemandirian, peningkatan performa sistem, hemat biaya, hingga pengembangan skala dari aplikasi yang dapat dilakukan dengan lebih mudah [6].

Selain itu, terdapat beberapa permasalahan lain yang ditemukan pada saat pelaksanaan pelayanan menggunakan aplikasi. Yaitu pada aplikasi ini tidak memberikan data yang lebih rinci mengenai penelitian yang akan dilakukan oleh masyarakat, hal ini karena terbatasnya kolom isian yang harus diisi oleh masyarakat saat melakukan pengajuan izin penelitian. Keterbatasan rincian data ini menyebabkan BRIDA NTB harus melakukan validasi secara manual sehingga menyebabkan penundaan penerbitan surat izin penelitian, yang dimana hal ini tentunya bertentangan dengan peraturan pemerintah No. 96 Tahun 2012 yang menuntut seluruh aparat pemerintah untuk memberikan pelayanan terbaik kepada masyarakat.

Oleh karena itu, karena berbagai permasalahan yang dialami tersebut dibutuhkan perancangan dan pembangunan sistem yang baru dengan menerapkan arsitektur JAMStack dan melakukan penyesuaian sesuai dengan kebutuhan sistem yang baru sehingga permasalahan yang ada dapat teratasi. Dalam perancangan sistem ini, tentunya dibutuhkan sebuah metode yang akan menjadi kerangka acuan dan pedoman selama masa pembangunan sistem. Metode yang digunakan adalah metode *fountain*. Metode *fountain* dipilih karena merupakan metode perbaikan dari metode *waterfall*, dimana metode *fountain* memberikan fleksibilitas yang sulit dilakukan menggunakan metode *waterfall*.

## **1.2 Rumusan Masalah**

Dari paparan latar belakang sebelumnya, rumusan masalah yang dapat diambil adalah bagaimana merancang dan membangun Sistem Informasi Pelayanan Publik BRIDA NTB dengan menerapkan arsitektur JAMStack.

## **1.3 Batasan Masalah**

Untuk membatasi cakupan masalah dalam penelitian ini, diberikan batasan masalah yang menjadi acuan agar penelitian yang dilakukan tidak menyimpang dan lebih terarah. Batasan masalah tersebut yaitu :

1. Sistem dibuat menggunakan bahasa pemrograman Javascript.
2. Teknologi *Frontend* yang digunakan dalam pembuatan sistem adalah Nextjs.
3. Teknologi *Backend* yang digunakan dalam pembuatan sistem adalah Strapi.
4. Pembuatan sistem menggunakan arsitektur JAMStack.
5. Studi kasus dalam perancangan sistem ini adalah Badan Riset dan Inovasi Daerah NTB.
6. Pelayanan publik yang dipilih adalah layanan penelitian.

## **1.4 Tujuan Penelitian**

Penelitian ini bertujuan untuk merancang dan membangun Sistem Informasi Pelayanan Publik menggunakan arsitektur JAMStack berdasarkan studi kasus yang terjadi di Badan Riset dan Inovasi Daerah NTB (BRIDA NTB).

## **1.5 Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah :

1. Menghasilkan Sistem Informasi Pelayanan Publik sebagai bentuk peningkatan kualitas pelayanan publik BRIDA NTB.
2. Memberikan kemudahan bagi masyarakat NTB yang mengajukan penelitian melalui BRIDA NTB.
3. Menjadi salah satu inovasi pelayanan publik di lingkungan Pemerintahan Daerah Provinsi Nusa Tenggara Barat.

## **1.6 Sistematika Penulisan**

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut :

### **1. Bab I Pendahuluan**

Bab ini berisi latar belakang, dari penelitian yang dilakukan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematikan penulisan.

### **2. Bab II Tinjauan Pustaka**

Bab ini membahas berbagai tinjauan publik yang ditinjau penulis, baik berupa penelitian sebelumnya maupun teori – teori dari berbagai sumber terpercaya yang mendukung dan berkaitan dengan penelitian tugas akhir ini.

### **3. Bab III Metodologi Penelitian**

Bab ini berisi langkah-langkah penelitian yang akan dilakukan, skenario sistem, struktur perangkat lunak yang dijelaskan dengan bantuan diagram, dan skenario pengujian.

### **4. Bab IV Hasil dan Pembahasan**

Bab ini membahas hasil pengujian berdasarkan skenario pengujian yang telah dilakukan.

### **5. Bab V Penutup**

Bab ini membahas kesimpulan dari penelitian yang telah dilakukan serta halaman untuk kritik dan saran terhadap penelitian ini.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Dalam pelaksanaannya, pembuatan Sistem Informasi Pelayan Publik dalam penelitian ini merujuk pada beberapa penelitian yang pernah dilakukan sebelumnya. Penelitian – penelitian tersebut digunakan sebagai referensi dalam perancangan dan pengembangan aplikasi yang dibuat.

Penelitian pertama dilakukan oleh [7] yang merancang sistem informasi pelayanan public untuk proses perizinan peredaran hasil hutan KPH wilayah Kota Kupang. Dimana dalam pembuatan sistem ini peneliti menggunakan arsitektur MVC tradisional dengan menggunakan bahasa pemrograman PHP (Hypertext Preprocessor) dan XAMPP sebagai *local server* serta bootsratp 4 sebagai UI Framework yang digunakan untuk membangun tampilan dari sistem informasi yang dibuat. Meskipun menggunakan arsitektur pengembangan aplikasi yang masih tradisional, sistem informasi yang dibuat dapat memenuhi tujuan dari pembuatan sistem informasi tersebut, yaitu memberikan kemudahan kepada para petugas dan pemohon sehingga dapat meningkatkan efektifitas dari segi biaya dan waktu.

Penelitian kedua oleh [8] yang membangun sistem informasi pelayanan publik di Kelurahan Banaran. Sistem informasi yang dibuat berfokus untuk memberikan kemudahan bagi masyarakat dalam pelayanan yang bersifat publik, memberikan kemudahan bagi pengelola dalam rekapitulasi data dan interaksi pelayanan dengan masyarakat. Peneliti membangun sistem informasi ini menggunakan framework Laravel. Dimana *Framework* Laravel menggunakan konsep MVC dalam membangun aplikasi. Peneliti juga menggunakan Bootstrap yang menjadi *UI Framework* untuk membangun tampilan antarmuka sistem, dan menggunakan *XAMPP* sebagai *local server*. Dari hasil penelitian yang dilakukan, dapat dilihat bahwa sistem yang dibuat dapat memenuhi kebutuhan layanan masyarakat yang ada di Kelurahan Banaran dengan nilai rata- rata yang didapatkan sebesar 74.9.

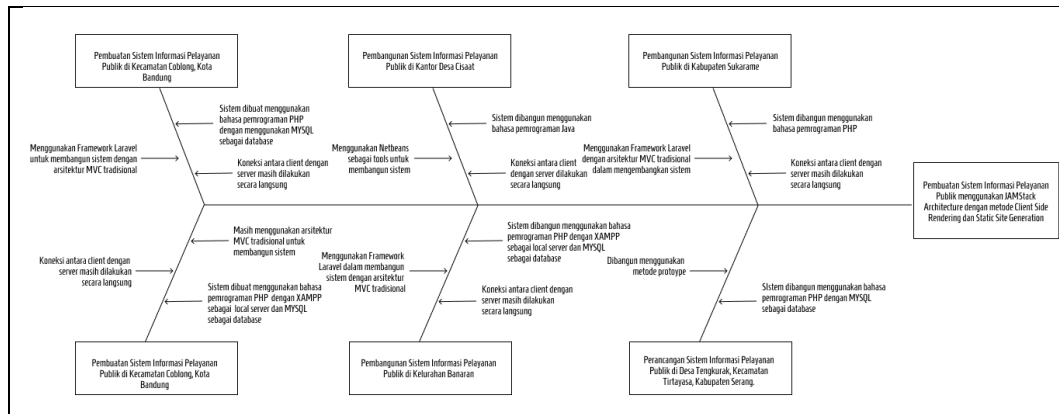
Penelitian ketiga dilakukan oleh [9] yang merancang sistem informasi pelayanan publik di Desa Tengkurak, Kecamatan Tirtayasa, Kabupaten Serang. Peneliti merancang sistem informasi menggunakan bahasa pemrograman PHP dan menggunakan database MYSQL. Dari hasil perancangan sistem yang dilakukan dapat dilihat bahwa sistem yang dibuat dapat memberikan kemudahan bagi masyarakat Desa Tengkurak. Masyarakat dapat mengakses seluruh informasi terkait Desa Tengkurak, masyarakat juga dapat melakukan pengajuan surat yang langsung di proses oleh Petugas Desa.

Penelitian keempat dilakukan oleh [10] yang menggunakan *Framework* Laravel dalam membuat sistem informasi pelayanan publik di Kecamatan Coblong Kota Bandung. Pada pembuatan sistem ini peneliti menggunakan bahasa pemrograman PHP dengan menggunakan MYSQL sebagai jenis *database* yang digunakan. Dari hasil penelitian yang dilakukan, dapat dilihat bahwa sistem yang dibangun dapat memberikan manfaat kepada masyarakat sehingga pelayanan masyarakat yang diberikan menjadi lebih efektif dan efisien. Akan tetapi masih terdapat hambatan seperti belum adanya sistem komputerisasi pada bidang pelayanan yang menimbulkan hambatan saat pelayanan masyarakat.

Penelitian kelima dilakukan oleh [11] yang membangun sistem informasi pelayanan publik menggunakan bahasa pemrograman Java dengan menggunakan *tools* yang disediakan pada aplikasi *Netbeans*. Sistem informasi yang dibuat oleh peneliti bertujuan untuk meningkatkan kualitas pelayanan publik dari Kantor Desa Cisaat. Dari hasil penelitian yang dilakukan, sistem yang dibuat dapat berjalan dengan baik. Sistem ini dapat memberikan kemudahan bagi perangkat desa dalam melakukan rekapitulasi data desa. Sistem ini juga dapat meningkatkan keefektifan pengarsipan data, sehingga meminimalisir terjadinya penyalahgunaan data.

Penelitian keenam dilakukan oleh [12] dengan menerapkan metode *extreme programming* dalam membangun sistem informasi manajemen pelayanan publik di Kabupaten Sukarame. Untuk membangun sistem ini, peneliti menggunakan bahasa pemrograman PHP dengan *Framework* Laravel yang menggunakan arsitektur MVC tradisional. Dari hasil penelitian yang dilakukan, sistem yang dibuat dapat

berjalan dengan sangat baik, hal ini dapat dilihat dari hasil *blackbox* testing dengan akurasi 100%.



Gambar 2. 1 Diagram Fishbone berdasarkan tinjauan 7ubic7

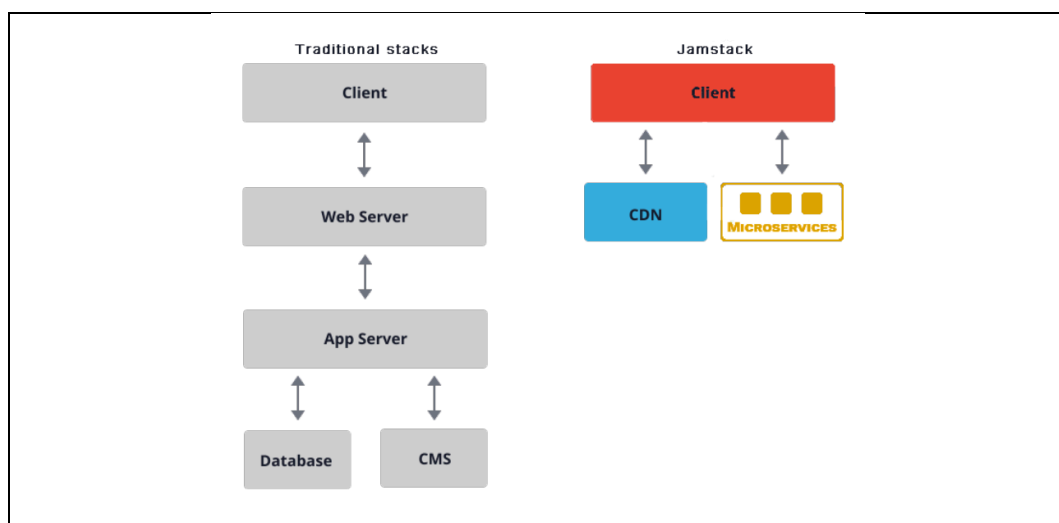
Diagram yang tersaji pada Gambar 2.1 merupakan diagram *Fishbone* dari sistem yang pernah dibuat dalam penelitian sebelumnya, dimana teknologi – teknologi yang digunakan masih menggunakan arsitektur MVC tradisional dengan menggunakan bahasa pemrograman PHP, yang dimana dengan menggunakan arsitektur MVC dalam pengembangan sistem masih memiliki kekurangan jika dibandingkan dengan arsitektur pengembangan sistem modern. Peneliti sebelumnya juga rata – rata menggunakan metode *Server Side Rendering* pada saat *client* melakukan interaksi dengan *server*, dimana dengan metode ini semua *logic* pada sistem akan dijalankan pada *server* terlebih dahulu sebelum mengirim respon ke *client*. Jika *logic* yang ada pada *server* semakin kompleks, maka *server* akan semakin lama melakukan komputasi yang akan menyebabkan penurunan performa kecepatan dari sistem tersebut. Untuk itu, peneliti bermaksud untuk membuat Sistem Informasi Pelayanan Publik menggunakan arsitektur pengembangan sistem modern, yaitu arsitektur JAMStack dengan metode *Client Side Rendering* dan *Static Site Generation* agar tidak memberatkan *server* dalam melakukan komputasi, sehingga *server* dapat mengirim respon ke *client* dengan lebih cepat. Hal ini tentunya akan meningkatkan performa dari sistem informasi yang dibuat.

## 2.2 Teori Penunjang

Dalam penelitian ini dibutuhkan pemahaman terkait kata – kata kunci seperti *JAMStack Architecture*, *Client Side Rendering*, *Static Site Generation*, Pelayanan Publik, dan *Headless CMS*. Terdapat teori – teori pendukung lainnya yang juga dibutuhkan dalam pengembangan sistem. Berikut merupakan teori penunjang dalam penelitian ini :

### 2.2.1 *JAMStack Architecture*

*JAMStack Architecture* merupakan salah satu arsitektur pengembangan aplikasi *web* modern yang terbentuk diantara tahun 2015 -2016. *JAMStack* terdiri dari 3 pilar utama yaitu *Javascript*, *API*, dan *Markup*. *JAMStack* hadir sebagai solusi dari permasalahan dalam aplikasi *web* dimana ketika terjadi peningkatan pengguna yang menyebabkan *server* tidak bisa mengatasi *request* yang terlalu banyak. *JAMstack* menggabungkan antara *dynamic content* dengan *static web-pages*, dimana *Javascript* yang akan berperan untuk mengatur konten dinamis yang kemudian di-embed kedalam *static HTML pages* lalu kemudian disimpan pada *Content Delivery Network (CDN)*. Dengan mekanisme *API* yang menghubungkan antara *Frontend* dan *Backend* membuat kemungkinan menggunakan *library* pihak ketiga menjadi lebih mudah [6].



Gambar 2. 2 Perbandingan arsitektur tradisional dengan arsitektur jamstack [13].

### **2.2.2 Client Side Rendering**

*Client Side Rendering* pada dasarnya merupakan sebuah metode dimana pada sisi *client* akan melakukan *fetching* data yang diinginkan kepada *server* sebelum ditampilkan kepada pengguna. Pada pengembangan *web* modern, pada sisi *client* disebut sebagai *Frontend* dan pada sisi *server* disebut sebagai *Backend*.

Baik *Frontend* dan *Backend* dibangun secara terpisah. Dimana pada bagian *Backend* dibangun menggunakan salah satu bahasa pemrograman kemudian mengirim data yang dikelola melalui *API*. Lalu dari sisi *Frontend* melakukan *Fetching* data untuk di-*rendering* (ditampilkan) kepada pengguna [14]. Pada pengembangan *web* modern, *Rendering* data rata – rata menggunakan bahasa pemrograman *Javascript* dengan *Framework* seperti *Angular*, *Vuejs*, maupun library seperti *Reactjs* [14].

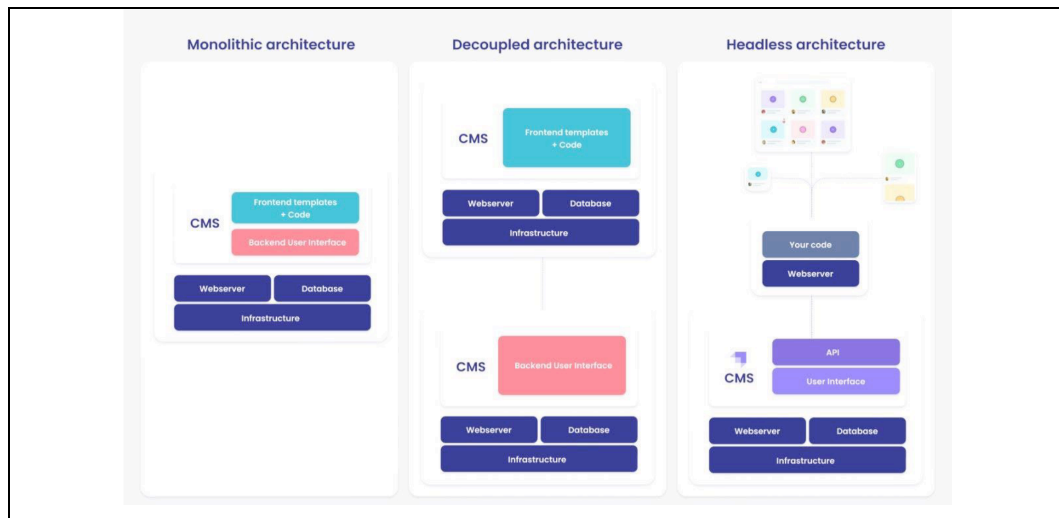
### **2.2.3 Static Site Generation**

*Static Site Generation* merupakan sebuah metode yang akan membuat halaman *website* yang dibangun menjadi *file-file* statis *HTML*, *CSS* dan *Javascript* untuk dikonsumsi oleh pengguna. *Static Site Generation* akan memungkinkan proses *loading* halaman menjadi lebih cepat. Hal ini dikarenakan halaman pada *website* akan dibuat menjadi halaman statis yang sudah tertanam data – data yang diinginkan didalamnya, berbeda dengan halaman dinamis yang memerlukan pembacaan *script* dan *database* yang membuat proses penampilan halaman *website* memerlukan waktu yang lebih lama [15].

### **2.2.4 Headless CMS**

*Headless Content Management System (Headless CMS)* merupakan salah satu bentuk dari *Backend Content Management System* dimana pada bagian *body content* (data) terpisah dengan bagian *display content* (tampilan). Data yang terdapat pada *Headless CMS* nantinya akan dikirim melalui *APIs request*. Hal ini memungkinkan data tersebut data dikonsumsi oleh banyak *device* [16].

Selain *Headless Architecture* yang digunakan oleh *Headless CMS*, terdapat *architecture* lain seperti *monolithic architecture* dan *decoupled architecture*. Baik *Headless Architecture*, *monolithic architecture*, maupun *decoupled architecture* memiliki karakteristik yang menjadi pembeda dari masing – masing *architecture*.



Gambar 2.3 Perbedaan Headless, Monolithic dan Decoupled Architecture.

Pada Gambar 2.2 dapat dilihat perbedaan antara masing – masing arsitektur yang digunakan dari teknologi CMS. Dalam *Monolithic Architecture* baik bagian *Frontend*, *Backend* dan *database* dijadikan menjadi satu, sedangkan pada *Decoupled Architecture* bagian *Fronend* maupun *Backend* terpisah, lalu pada *Headless Architecture* hanya berfokus pada perancangan pada *Backend* saja [16].

*Headless CMS* merupakan teknologi yang sangat fleksibel karena hanya berfokus untuk mengatur *Backend* sedangkan pada bagian *Frontend* diberikan kebebasan untuk memilih teknologi yang diinginkan [16].

### 2.2.5 NextJS

*Next JS* merupakan *React Framework* yang memungkinkan untuk membuat *static sites pages*. *Next JS* ini dibangun dan dikembangkan oleh perusahaan yang bernama Vercel. *Next JS* merupakan *Framework* yang mudah dipelajari oleh semua developer yang sudah familiar dengan *HTML*, *CSS Javascript* dan *React*. Salah satu fitur unggulan yang dapat digunakan adalah *Static Site Generation* yang memungkinkan *Next JS* untuk membuat *static pages* dan *API Routes* yang mempermudah *developer* dalam mengatur *routing* dari aplikasi yang dibuat. *Next JS* juga mempunyai fitur *Image Optimization* yang dapat mengoptimisasikan *loading* gambar pada saat dimuat dalam aplikasi [16].

### 2.2.6 REST API

*REST API* merupakan layanan *web service* dimana proses pertukaran data dilakukan melalui protokol *HTTP* sedangkan data yang dipertukarkan

dikonversikan ke dalam format *Javascript Object Notation* (JSON). Interaksi antara *client* dan *server* dilakukan dengan mengirimkan data melalui *Uniform Request Identifier* (URI) dimana *server* akan memberikan *response* atas *request* yang dikirimkan oleh *client*. Adapun *HTTP method* yang digunakan pada proses pertukaran data ini antara lain adalah POST, PUT, GET dan DELETE [17].

### **2.2.7 Javascript**

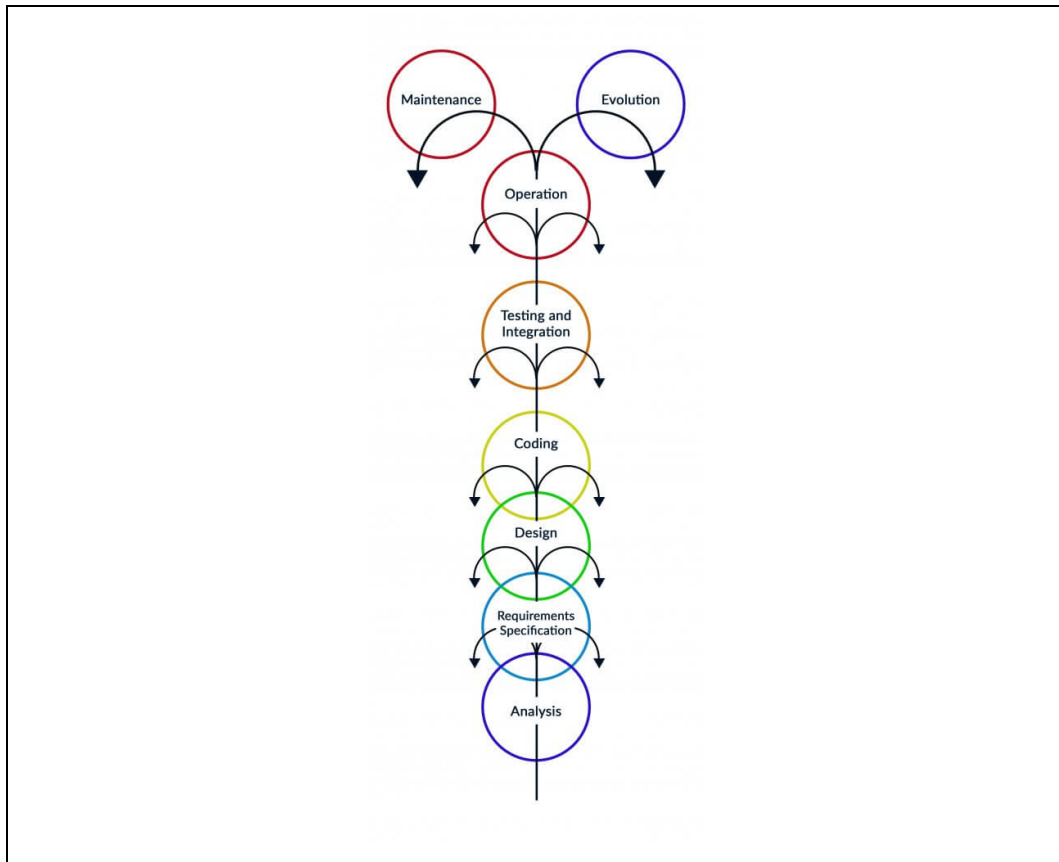
*Javascript* merupakan bahasa pemrograman pada sisi *client*. *Javascript* merupakan bahasa pemrograman yang mendekati bahasa manusia atau bisa dikatakan bahasa tingkat tinggi, hal ini yang membuat *Javascript* disebut sebagai bahasa tingkat tinggi yang membuatnya mudah dipelajari. *Javascript* dibuat untuk memungkinkan website menjadi lebih interaktif, seperti menampilkan dan menghilangkan objek – objek tertentu [18].

### **2.2.8 Blackbox Testing**

Metode *Blackbox Testing* adalah proses pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, dimana *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Terdapat dua teknik yang digunakan dalam metode *Blackbox Testing* yaitu *Equivalence Partitioning* dan *Boundary Value Analysis* [19].

### **2.2.9 Metode Fountain**

Metode fountain merupakan perbaikan logis dari metode *waterfall*. Metode *fountain* memungkinkan untuk melewati tahapan – tahapan yang lain kecuali tahapan *design* sehingga dalam pembangunan sistem dapat dilakukan dengan lebih cepat jika dibandingkan dengan metode *waterfall* yang harus melewati tahapan sesuai urutan. [20]



Gambar 2. 4 Tahapan metode fountain.

Gambar 2.4 merupakan gambar dari tahapan – tahapan yang akan dilewati oleh pengembang jika menggunakan metode fountain dalam melakukan pengembangan sistem informasi. Tahapan dalam metode fountain terdiri atas tahapan *Analysis*, *Requirements*, *Design*, *Coding*, *Testing and Integration*, dan *Maintenance*. [20]



## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Alat dan Bahan**

Dalam pembangunan sistem informasi pelayanan publik dengan menerapkan JAMStack arsitektur, peneliti menggunakan alat dan bahan sebagai berikut :

##### **3.1.1 Alat**

Peralatan yang digunakan untuk membangun sistem adalah :

1. *Hardware* laptop Macbook Air M1 dengan RAM 8GB dan SSD 256GB.
2. *Operating System* MacOS Monterey *version* 12.4.
3. *Software* Visual Studio Code untuk menulis *script code*.
4. Bahasa pemrograman Javascript.
5. *Framework* NextJS sebagai teknologi *frontend*.
6. Nodejs sebagai *runtime* Javascript untuk sisi *frontend* dan *backend*
7. Strapi sebagai *backend*.
8. Nginx sebagai *Web Server*.
9. Mysql sebagai *database* sistem.

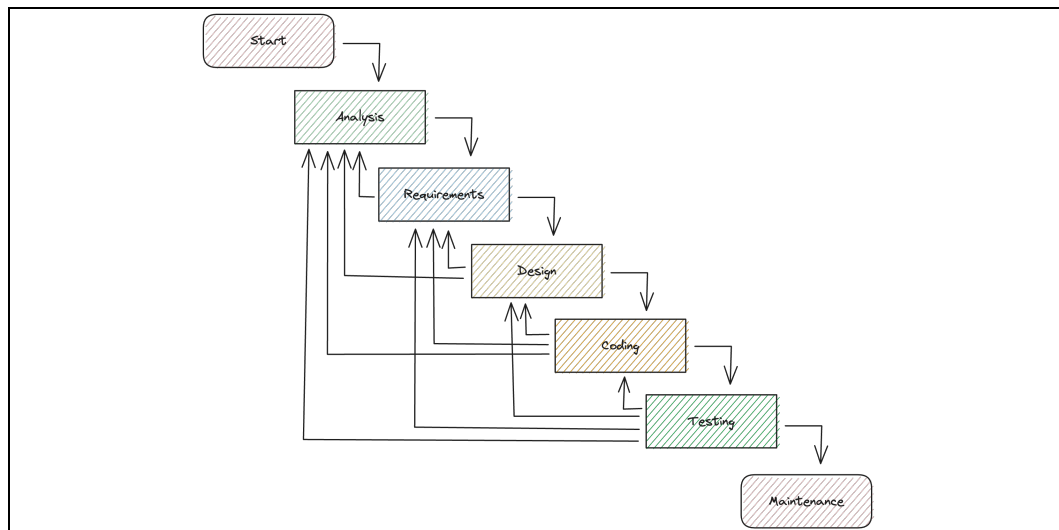
##### **3.1.2 Bahan**

Bahan yang digunakan dalam pembuatan sistem adalah :

1. Literatur yang didapatkan dari jurnal.
2. Sistem informasi Pelayanan Publik yang sudah dibuat sebelumnya.
3. Penelitian-penelitian yang telah dilakukan sebelumnya.
4. Data-data yang diberikan oleh BRIDA NTB.

#### **3.2 Metode Pengembangan**

Dalam membangun sebuah sistem, diperlukan sebuah metode yang akan menjadi acuan dalam pengembangan. Untuk itu dalam membangun sistem informasi pelayanan publik ini metode yang digunakan adalah metode *fountain*. Tahapan pengembangan sistem dengan menggunakan metode *fountain* memiliki kemiripan dengan metode *waterfall*, hal ini karena metode *fountain* merupakan metode perbaikan dari metode *waterfall*.



Gambar 3.1 Metode penelitian.

Pada Gambar 3.1 ditampilkan tahapan pengembangan sistem dengan menggunakan metode fountain. Tahapan – tahapan tersebut antara lain :

### 3.2.1 *Analysis*

Pada tahapan ini dilakukan analisa pada sistem pelayanan publik sebelumnya untuk menjadi acuan pengembangan sistem yang baru sehingga permasalahan yang terdapat pada sistem pelayanan publik sebelumnya dapat diatasi pada sistem yang baru. Pada tahapan ini peneliti menggunakan metode wawancara dan pengujian sistem secara langsung.

#### 3.2.1.1 Metode Wawancara

Wawancara yang dilakukan berfokus kepada fungsionalitas dari sistem yang sudah ada. Langkah – langkah yang dilakukan oleh peneliti dalam melakukan wawancara adalah :

##### 1. Identifikasi Responden

Langkah ini dilakukan oleh peneliti untuk mengidentifikasi responden yang akan diwawancara seperti tim pengembang sistem dan bidang yang melaksanakan pelayanan penelitian di BRIDA NTB.

##### 2. Perumusan Pertanyaan

Pada langkah ini, peneliti menyiapkan pertanyaan – pertanyaan yang relevan mengenai sistem layanan penelitian yang akan dikembangkan. Hal ini dilakukan untuk memastikan kelancaran dan nilai informasi dalam proses wawancara yang dilakukan.

### 3. Pelaksanaan Wawancara

Wawancara dilakukan dengan responden sesuai dengan panduan yang telah dibuat. Selama wawancara, peneliti mencatat secara sistematis jawaban yang didapatkan dari responden.

### 4. Analisis Hasil Wawancara

Setelah menyelesaikan wawancara, peneliti akan melakukan analisis data sehingga hasil analisis tersebut dari wawancara yang telah dilakukan dapat menjadi referensi yang digunakan selama proses penelitian.

#### **3.2.1.2 Pengujian Sistem Secara Langsung**

Setelah melakukan wawancara, peneliti selanjutnya melakukan pengujian sistem secara langsung. Pengujian sistem berfokus pada proses aliran data, keamanan, dan performa dari sistem saat digunakan oleh admin maupun peneliti. Hasil pengujian ini nantinya akan menjadi acuan bagi pengembang dalam menentukan bagian mana saja yang harus dikembangkan dan diterapkan pada sistem yang baru.

#### **3.2.1.3 Hasil Analisa**

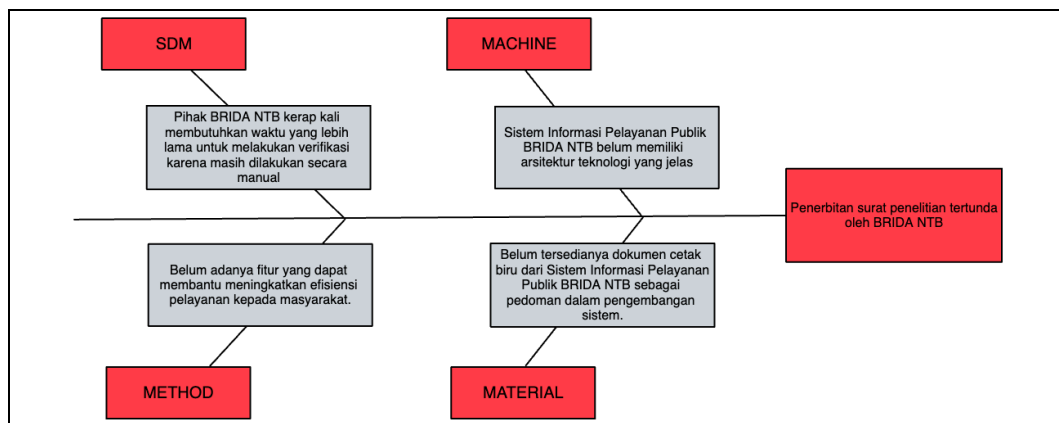
Dari hasil analisa yang dilakukan, ditemukan beberapa permasalahan pada fungsionalitas sistem yaitu:

1. *SSL Certificate* yang tidak terpasang dengan baik. Permasalahan ini ditemukan oleh peneliti saat membuka sistem dan beberapa kali di-*block* oleh browser dengan alasan website tersebut tidak aman karena belum terinstall *SSL Certificate*.
2. Disfungsi aplikasi saat pengguna mengirim data yang mengakibatkan data pengguna tidak berhasil terkirim ke sistem. Hal ini masih terkait dengan *SSL Certificate* yang tidak terpasang dengan baik sehingga menyebabkan *request* yang dikirim pengguna di-*block* oleh server.
3. Tidak terdapatnya fitur yang dapat mengefisiensi pelayanan, sehingga pelayanan tertentu masih dilakukan secara manual. Hal ini menyebabkan terjadi keterlambatan pada saat penerbitan surat izin penelitian.
4. Kolom isian data yang harus diisi oleh pengguna kurang deskriptif, sehingga dari BRIDA NTB sulit melakukan tabulasi data.

Lalu permasalahan yang ditemukan pada infrastruktur sistem antara lain :

1. Peneliti menemukan implementasi keamanan yang kurang baik yaitu tidak menggunakan protocol HTTPS.
2. Peneliti mengalami kendala saat mengidentifikasi aliran data dari sistem sebelumnya karena tidak adanya *source code* dari sistem sebelumnya.
3. Peneliti mengalami kesulitan dalam melakukan pengembangan sistem karena tidak ada akses ke *source code* dari sistem sebelumnya.

Hasil analisa yang dilakukan oleh peneliti dapat digambarkan kedalam *diagram fishbone* dibawah ini.



Gambar 3.2 Hasil analisa sistem sebelumnya.

Berdasarkan diagram *fishbone* Pada Gambar 3.2, peneliti menemukan akar permasalahan yang menjadi penyebab perlunya dilakukan pengembangan sistem yang baru dengan menerapkan arsitektur JAMStack dan mengaplikasikan solusi dari berbagai permasalahan tersebut dalam aplikasi.

### 3.2.2 Requirements

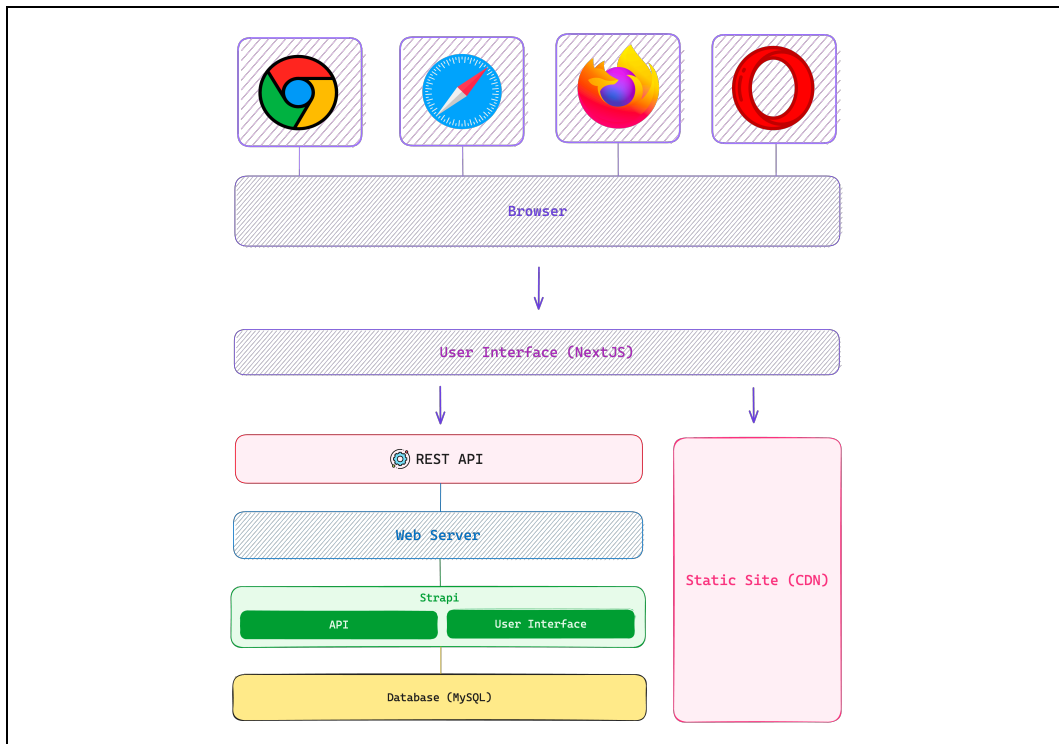
Setelah melakukan analisa sistem yang akan dibuat, selanjutnya pada tahapan ini dilakukan identifikasi hal – hal yang diperlukan untuk membangun sistem. Dalam pembangunan sistem yang baru dengan menerapkan arsitektur JAMStack, peneliti memerlukan teknologi yang dapat mendukung pengembangan sehingga sistem dapat dibuat secara maksimal. Karena pada arsitektur JAMStack *Frontend* dan *Backend* terpisah, maka pada sisi *Frontend* peneliti menggunakan NextJs sebagai *framework*, sedangkan pada sisi *Backend* peneliti menggunakan Strapi.

*Framework* NextJS dipilih karena beberapa fitur yang dibutuhkan untuk mempercepat proses komputasi dalam sistem. Pada NextJS pengembang bisa menentukan dalam menggunakan *Server Side Rendering*, *Client Side Rendering* maupun *Satic Site Generation*. Dengan fitur ini, pengembang dapat meningkatkan tingkat efisiensi *resource* yang dibutuhkan pada sistem sehingga dapat meningkatkan performa dari sistem yang akan dibuat. Kemudian pada sisi *BackEnd*, Strapi dipilih untuk mempermudah pengembang dalam mempersiapkan *endpoint* API yang dibutuhkan untuk proses komputasi sistem. Dengan fitur yang diberikan oleh Strapi, pengembang bisa lebih berfokus untuk penyelesaian *business flow* dari aplikasi karena kebutuhan dari sisi *BackEnd* sudah diselesaikan oleh fitur didalam Strapi secara otomatis.

### 3.2.3 Design

Pada tahap ini dilakukan perancangan desain sistem berdasarkan hasil yang didapatkan pada tahap *Analysis* dan *Requirements*. Sesuai dengan konsep arsitektur JAMStack, proses perancangan sistem berfokus pada dua sisi, yaitu pada sisi *FrontEnd* dan pada sisi *BackEnd*.

#### 3.2.3.1 Design Arsitektur



Gambar 3.3 Gambar arsitektur Jamstack

Pada Gambar 3.3, dapat dilihat gambaran arsitektur dari sistem yang akan dibangun. Sistem ini akan menerapkan arsitektur JAMStack dengan menggunakan Next.js di sisi *frontend* dan Strapi di sisi *backend*. Untuk menghubungkan sisi frontend dengan sisi backend, peneliti menggunakan REST API sebagai metode komunikasi antara keduanya.

Penerapan arsitektur JAMStack dapat memberikan keuntungan bagi peneliti dalam menerapkan solusi dari permasalahan yang ditemukan. Arsitektur JAMStack memberikan keunggulan yang tidak ditemukan dalam arsitektur lain, seperti :

1. Kinerja yang Tinggi

Arsitektur JAMStack biasanya menghasilkan situs web dengan kinerja yang sangat cepat karena laman yang dibuat menggunakan teknologi SSG akan di-*cache* dan didistribusikan melalui Content Delivery Network (CDN).

2. Keamanan yang Ditingkatkan

Karena sebagian besar logika aplikasi dieksekusi di sisi klien, serangan seperti SQL injection dan Cross-Site Scripting (XSS) cenderung lebih sulit dilakukan.

3. Kemudahan Pengembangan

Pengembang dapat menggunakan berbagai teknologi front-end untuk membangun tampilan, seperti React, Vue, atau Angular, yang memungkinkan pengembangan yang lebih fleksibel dan modern.

4. Skalabilitas

Karena sebagian besar beban server dihilangkan, situs yang dibangun dengan JAMStack cenderung lebih mudah diskalakan secara horizontal.

5. Pemeliharaan yang Mudah

Pemeliharaan situs menjadi lebih sederhana karena sebagian besar konten dikelola sebagai file statis atau disimpan di sistem manajemen konten (CMS).

Keunggulan yang didapatkan dari penerapan arsitektur JAMStack dapat menjadi solusi untuk mengatasi permasalahan yang saat ini dihadapi oleh sistem yang lama. Masalah-masalah seperti performa aplikasi, keamanan, dan skalabilitas sistem dapat lebih mudah diatasi dengan menerapkan arsitektur JAMStack.

Berikut merupakan perbandingan dari sistem lama dengan sistem baru yang menggunakan arsitektur JAMStack dengan menerapkan solusi untuk menyelesaikan permasalahan yang didapatkan dari sisi fungsionalitas sistem

Table 3. 1 Perbandingan antara sistem lama dengan sistem baru.

| Permasalahan | Sistem Lama   | Sistem Baru   |
|--------------|---|---|
| Keamanan     | SSL tidak terinstall dengan baik sehingga sistem di- <i>block</i> oleh <i>browser</i>                         | <ul style="list-style-type: none"> <li>- Menggunakan SSL resmi milik Provinsi NTB.</li> <li>- Menerapkan SSG untuk membuat file statis sehingga mengurangi komputasi server.</li> </ul> |
| Skalabilitas | <i>Source code</i> tidak tersedia sehingga kemungkinan peningkatan skalabilitas aplikasi tidak bisa dilakukan | Skalabilitas cenderung lebih mudah dilakukan karena sisi <i>frontend</i> terpisah dengan sisi <i>backend</i> .  |
| Performa     | Performa tidak stabil karena sistem sepenuhnya bergantung kepada kualitas komputasi server                    | Performa lebih cepat dan stabil karena sistem tidak sepenuhnya bergantung kepada kualitas komputasi server.   |

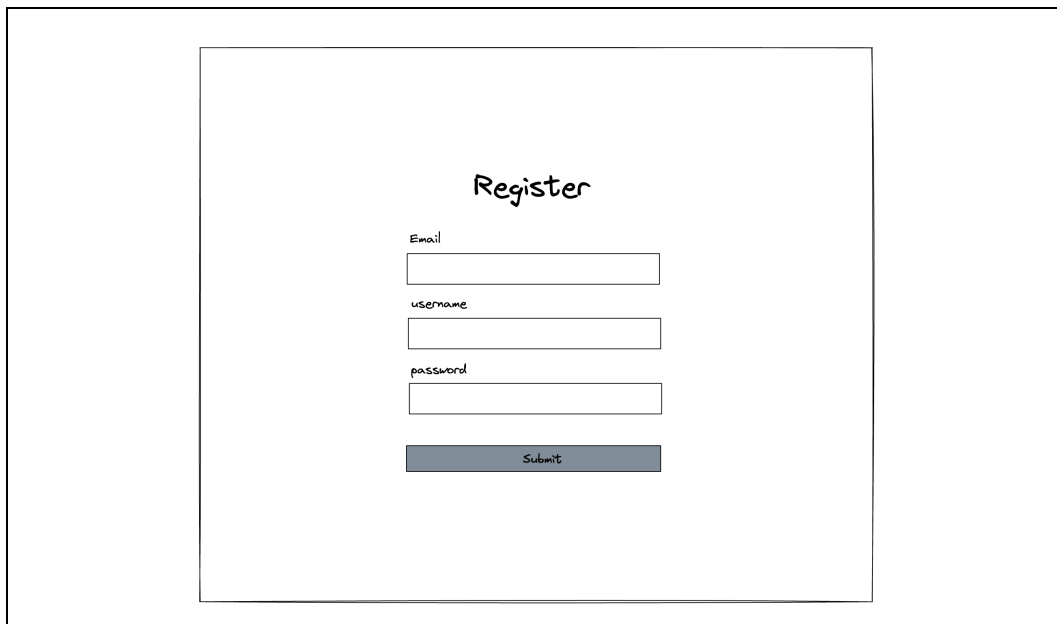
### 3.2.3.2 *Design FrontEnd*

Perancangan desain sistem pada sisi *FrontEnd* dilakukan untuk memberikan kenyamanan bagi pengguna saat menggunakan sistem. Pada tahap ini peneliti merancang tampilan dari fitur – fitur yang dapat diakses oleh setiap pengguna pada setiap level yang disajikan dalam bentuk *wireframe design*.

#### 3.2.3.2.1 *Wireframe Design*

*Wireframe design* merupakan sebuah kerangka atau *blueprint* dari tampilan aplikasi yang akan dibuat. *Wireframe* biasanya dibuat sebelum pembuatan produk/aplikasi untuk memberikan gambaran seperti apa produk/aplikasi yang akan dibuat. Berikut merupakan *wireframe design* dari fitur – fitur sistem yang akan dibuat :

## 1. Laman *register*

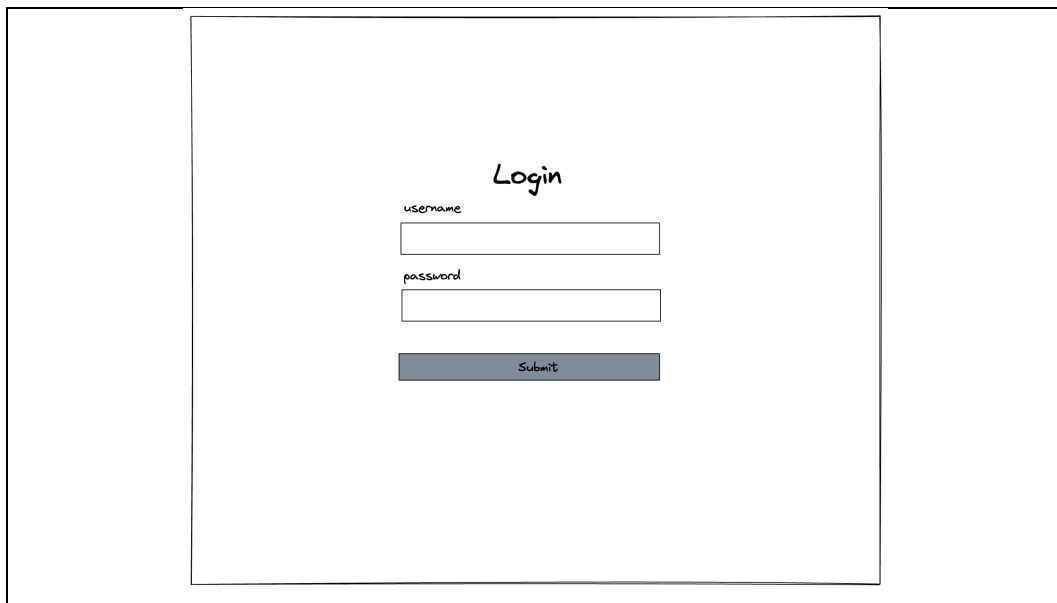


The wireframe shows a registration form titled "Register". It contains three input fields labeled "Email", "username", and "password", and a "Submit" button.

Gambar 3.4 Wireframe design laman register.

Gambar 3.4 menunjukkan rancangan tampilan dari laman *register*. Pada laman ini aktor *user* harus mengisi *email*, *username*, dan *password* untuk melakukan registrasi sebelum menggunakan sistem.

## 2. Laman *login*



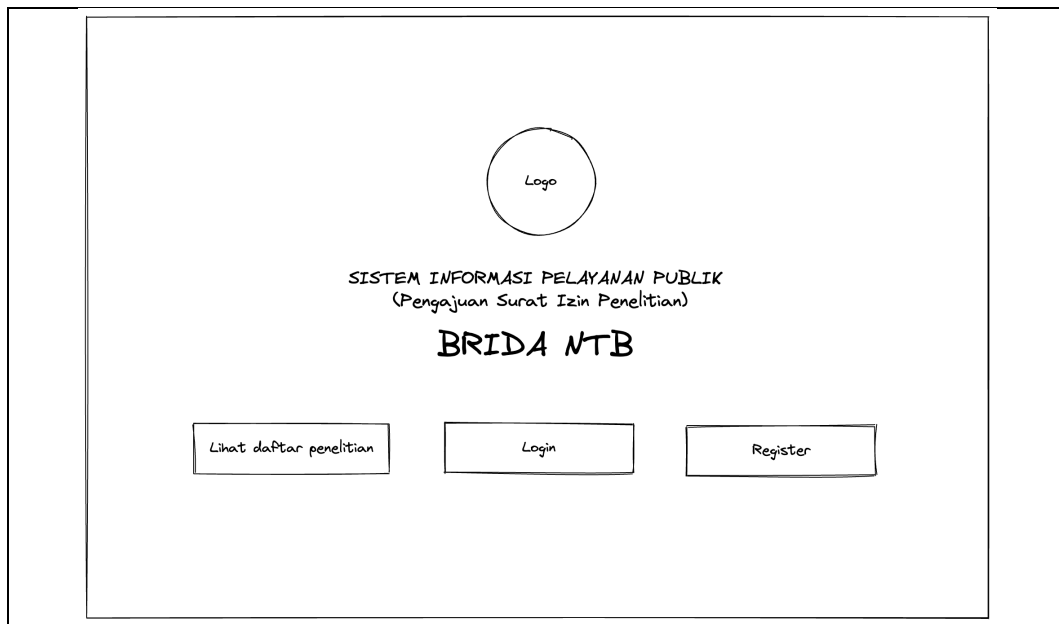
The wireframe shows a login form titled "Login". It contains two input fields labeled "username" and "password", and a "Submit" button.

Gambar 3.5 Wireframe design laman login.

Gambar 3.5 menunjukkan rancangan tampilan dari laman *login* pada sistem. Pada laman ini pengguna harus memasukkan *username* dan *password* yang sudah dibuat sebelumnya pada laman *register*.



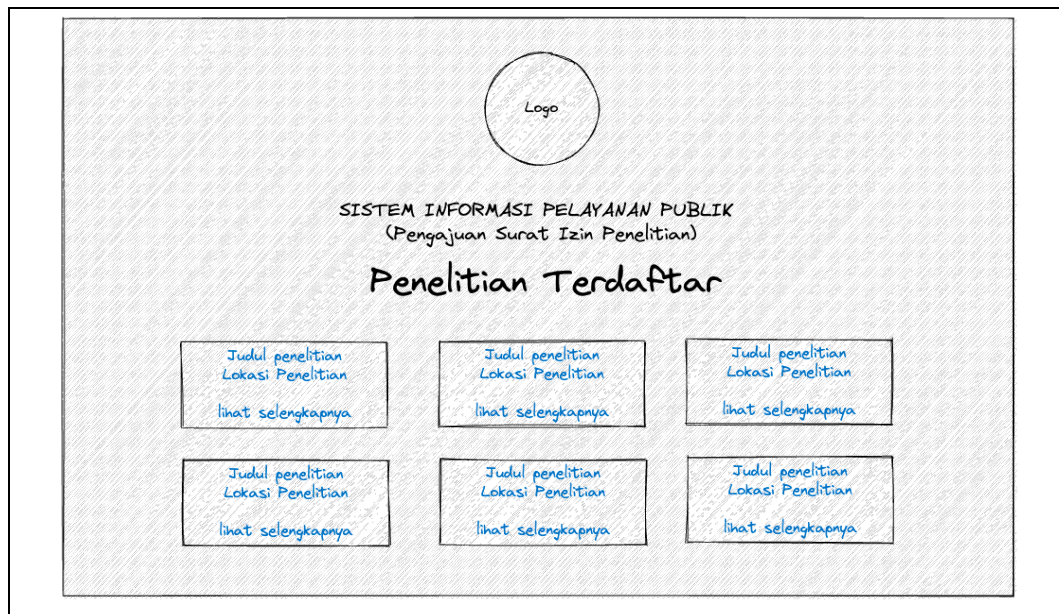
### 3. Laman *Landing*



Gambar 3.6 Wireframe design dari laman landing.

Gambar 3.6 merupakan laman pertama yang akan dilihat oleh pengguna saat membuka sistem. Pada laman ini pengguna akan diminta memilih salah satu dari menu daftar penelitian, *login* dan *register*.

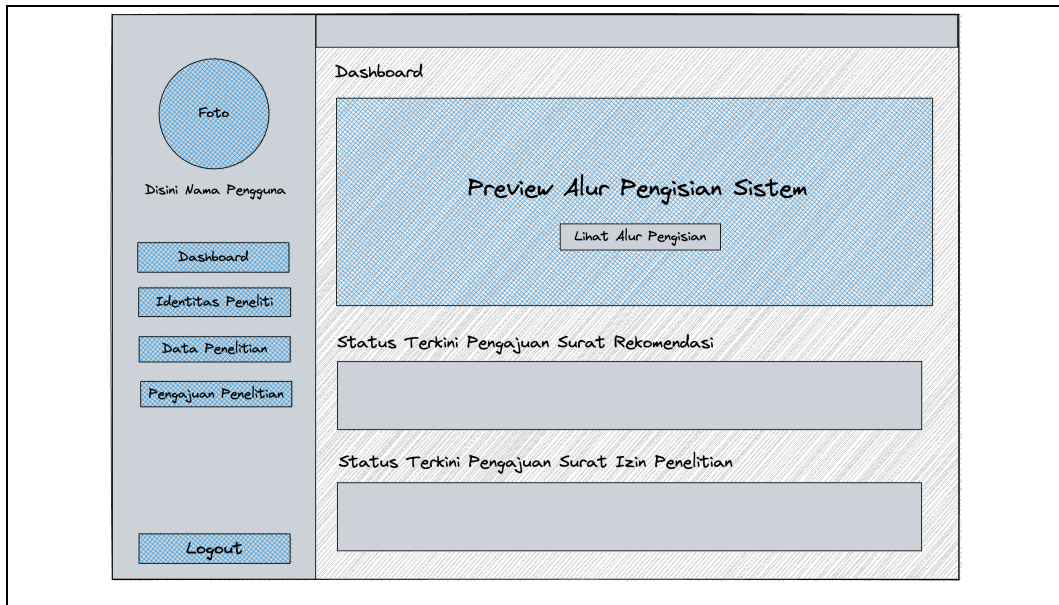
### 4. Laman penelitian terdaftar



Gambar 3.7 Wireframe design dari laman penelitian terdaftar.

Gambar 3.7 merupakan rancangan tampilan dari laman penelitian terdaftar. Laman ini nantinya akan berisi daftar penelitian yang sudah mendapatkan surat izin penelitian dari BRIDA NTB.

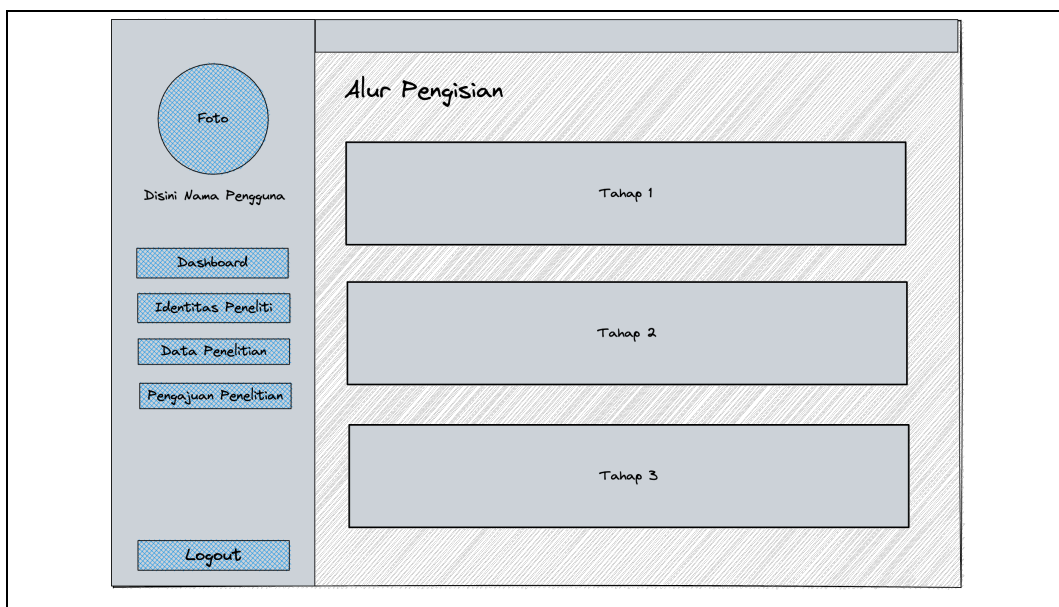
### 5. Laman *dashboard*



Gambar 3.8 Wireframe design dari laman dashboard.

Gambar 3.8 menunjukkan rancangan tampilan dari laman *dashboard*. Laman ini merupakan laman pertama yang akan dilihat oleh aktor *user* setelah melakukan *login* pada sistem.

### 6. Laman Petunjuk pengisian sistem



Gambar 3.9 Wireframe design laman alur pengisian.

Gambar 3.9 menunjukkan rancangan tampilan dari laman alur pengisian sistem. Pada laman ini aktor *user* dapat melihat tahapan pengisian beserta kolom isian apa saja yang perlu diisi.

## 7. Laman pengisian identitas peneliti

The wireframe shows a user interface for 'Identitas Peneliti'. On the left is a sidebar with a circular 'Foto' placeholder, the text 'Disini Nama Pengguna', and navigation buttons for 'Dashboard', 'Identitas Peneliti', 'Data Penelitian', 'Pengajuan Penelitian', and 'Logout'. The main content area is titled 'Identitas Peneliti' and contains a large text box for a description: 'Disini deskripsi tentang laman identitas peneliti'. Below this are nine input fields arranged in a 3x3 grid, each with a 'label isian' label above it. A blue 'Simpan' button is located at the bottom of the form.

Gambar 3.10 Wireframe design laman identitas peneliti.

Gambar 3.10 merupakan rancangan tampilan dari laman identitas peneliti. Pada laman ini aktor *user* harus mengisi kolom isian mengenai data diri dari *user* tersebut.

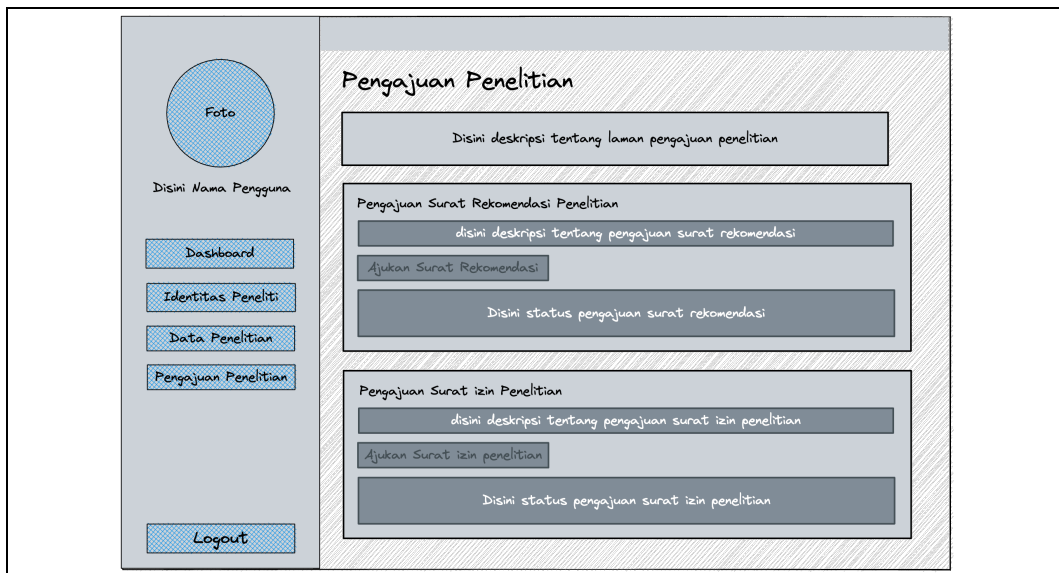
## 8. Laman pengisian data penelitian

The wireframe shows a user interface for 'Data Penelitian'. It features the same sidebar as Gambar 3.10, with the 'Data Penelitian' button highlighted. The main content area is titled 'Data Penelitian' and contains a large text box for a description: 'Disini deskripsi tentang laman Data Penelitian'. Below this are nine input fields arranged in a 3x3 grid, each with a 'label isian' label above it. A blue 'Simpan' button is located at the bottom of the form.

Gambar 3.11 Wireframe design laman data penelitian.

Gambar 3.11 merupakan rancangan tampilan dari laman data penelitian. Pada laman ini aktor *user* harus mengisi kolom isian mengenai penelitian yang akan dilakukan.

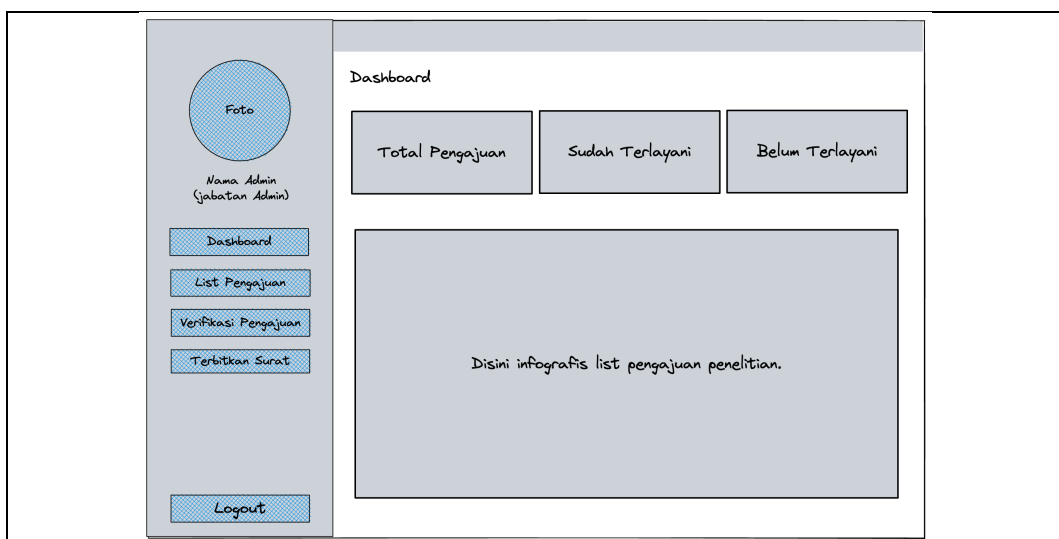
### 9. Laman pengajuan penelitian



Gambar 3.12 Wireframe design laman pengajuan penelitian.

Gambar 3.12 merupakan rancangan tampilan dari laman pengajuan penelitian. Pada laman ini aktor *user* harus mengikut setiap langkah pengajuan, mulai dari pengajuan surat rekomendasi penelitian hingga pengajuan surat izin penelitian.

### 10. Laman *dashboard* admin



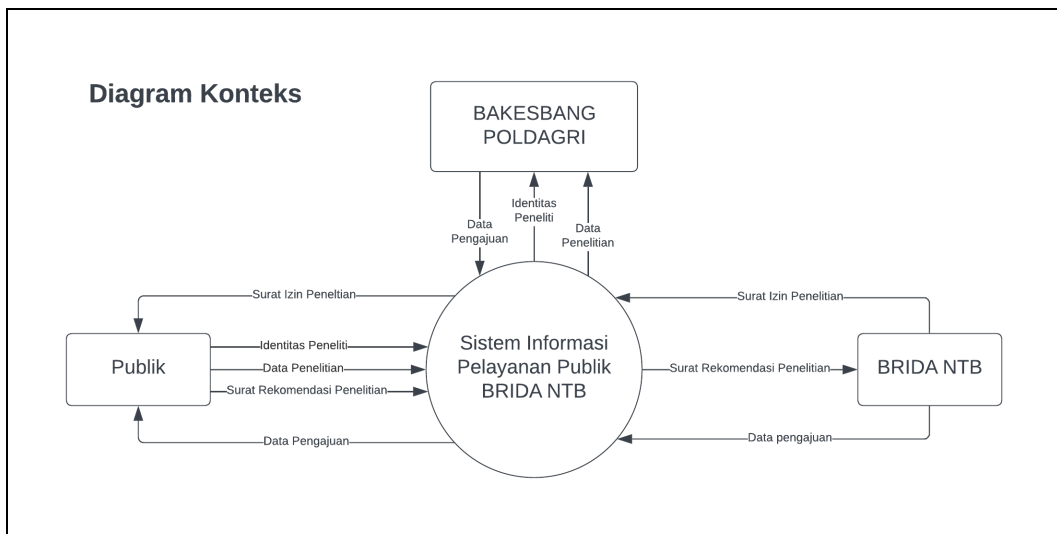
Gambar 3.13 Wireframe design laman dashboard admin.

Gambar 3.13 menunjukkan rancangan tampilan dari laman *dashboard* admin. Dimana pada laman ini aktor admin dapat melihat daftar pengajuan dari *user*. Admin juga dapat melakukan verifikasi, melihat data yang sudah di verifikasi, hingga menerbitkan surat.

### 3.2.3.3 Design BackEnd

Perancangan design sistem pada sisi *BackEnd* berfokus kepada *design* arsitektur yang akan digunakan untuk mempermudah proses yang terjadi pada sisi *FrontEnd*. Pada tahapan ini, setiap fitur dan alur pemrosesan data yang akan terjadi pada sisi *BackEnd* digambarkan menggunakan beberapa diagram. Alur data dan aktor yang akan menggunakan sistem digambarkan menggunakan *Data Flow Diagram* (DFD), sedangkan penjelasan alur penggunaan fitur – fitur utama yang dapat diakses pada sistem digambarkan *Flowchart Diagram*, dan relasi antar entitas pada database digambarkan menggunakan *Entity Relationship Diagram* (ERD).

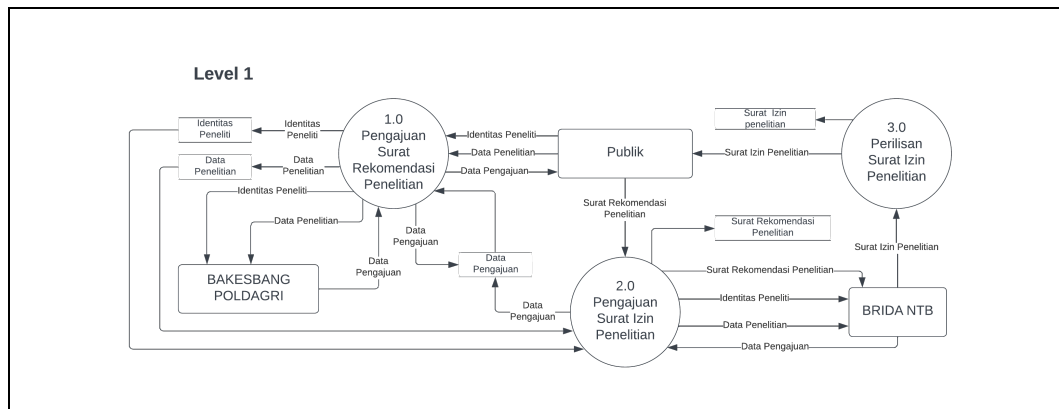
#### 3.2.3.3.1 Data Flow Diagram (DFD)



Gambar 3.14 Diagram konteks *data flow diagram* (DFD).

Gambar 3.14 merupakan diagram konteks atau gambaran umum mengenai alur data dari sistem yang akan dibuat, dimana pada sistem terdapat tiga entitas yang akan bertukar data, yaitu entitas Publik, entitas BAKESBANG POLDAGRI, dan entitas BRIDA NTB.

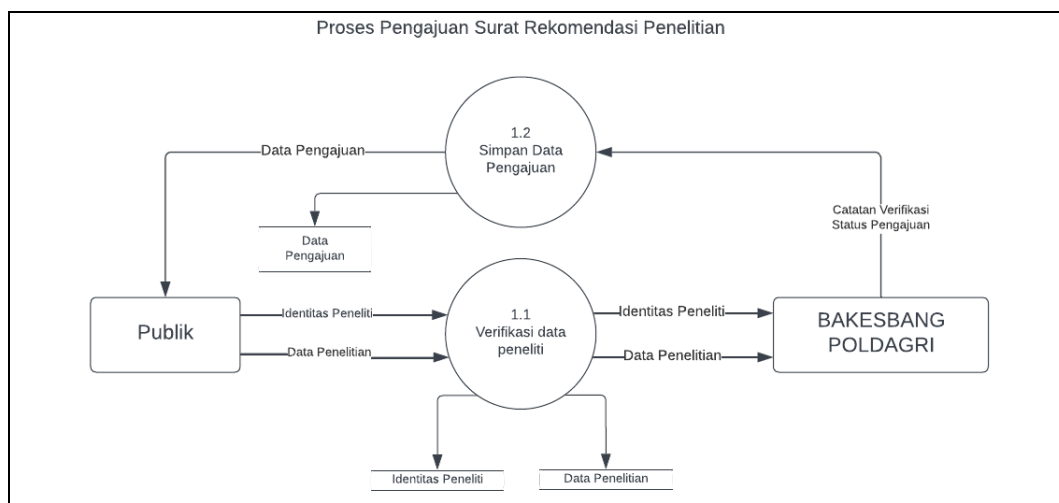
## 1. Diagram Level 1



Gambar 3.15 data flow diagram (DFD) level 1.

Gambar 3.15 merupakan gambar level 1 dari *data flow diagram* (DFD) sistem yang akan dibuat. Pada DFD level 1 ini terdapat tiga proses utama yaitu proses Pengajuan Surat Rekomendasi Penelitian, proses Pengajuan Surat Izin Penelitian dan proses Perilisan Surat Izin Penelitian.

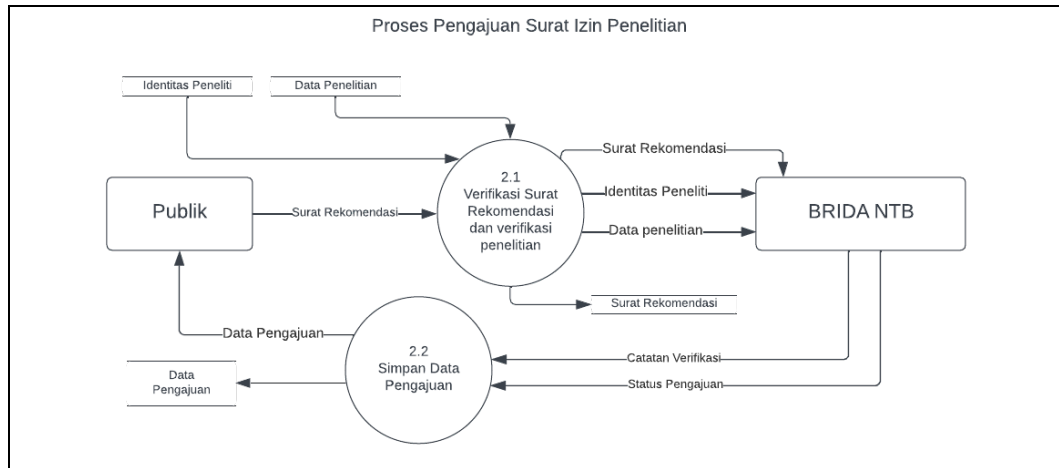
## 2. Diagram Level 2



Gambar 3.16 DFD level 2 proses pengajuan surat rekomendasi penelitian.

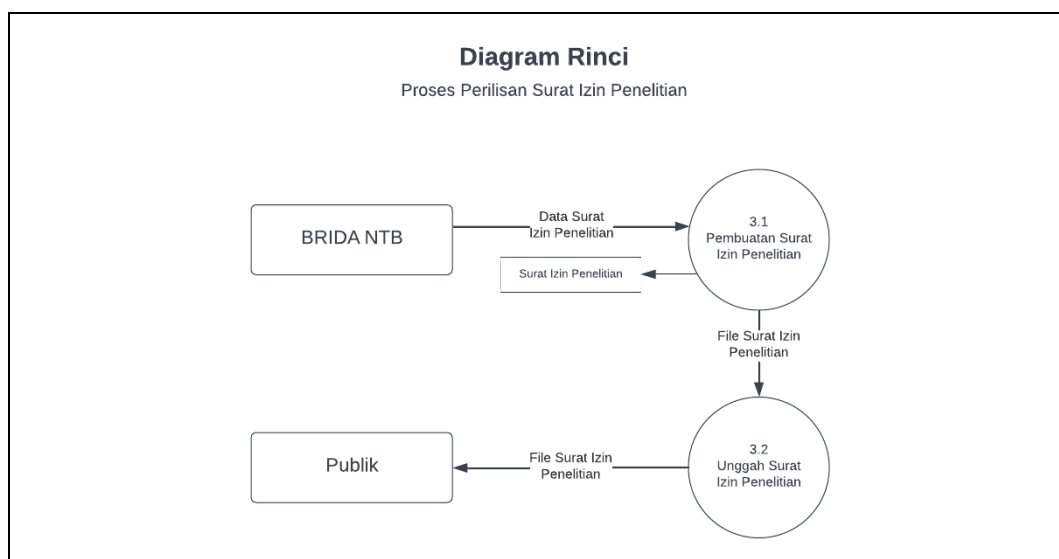
Pada Gambar 3.16 merupakan gambar level 2 dari proses pengajuan surat rekomendasi penelitian. Pada proses ini entitas Publik akan menginput data identitas peneliti dan data penelitian kedalam sistem yang selanjutnya akan diverifikasi oleh entitas BAKESBANG POLDAGRI. Hasil catatan verifikasi yang telah dilakukan nantinya akan dikirim kembali beserta dengan status pengajuan yang diisi oleh entitas BAKESBANG POLDAGRI kedalam sistem untuk disimpan

sebagai data pengajuan. Data pengajuan ini berfungsi sebagai pusat informasi mengenai proses pengajuan yang dilakukan oleh entitas Publik.



Gambar 3.17 DFD level 2 proses pengajuan surat izin penelitian.

Pada Gambar 3.17 dapat dilihat proses pengajuan surat izin penelitian yang digambarkan menggunakan diagram rinci. Pada proses tersebut sistem akan mengambil data dari entitas Publik seperti data identitas peneliti dan data penelitian yang sudah diinput pada proses pertama. Sistem juga akan menerima surat rekomendasi penelitian dari Publik untuk dilakukan verifikasi data oleh entitas BRIDA NTB. Seluruh hasil verifikasi berupa catatan verifikasi dan status pengajuan nantinya akan disimpan ke dalam sistem sebagai data pengajuan. Data pengajuan ini berfungsi sebagai pusat informasi mengenai proses pengajuan yang dilakukan oleh entitas Publik.



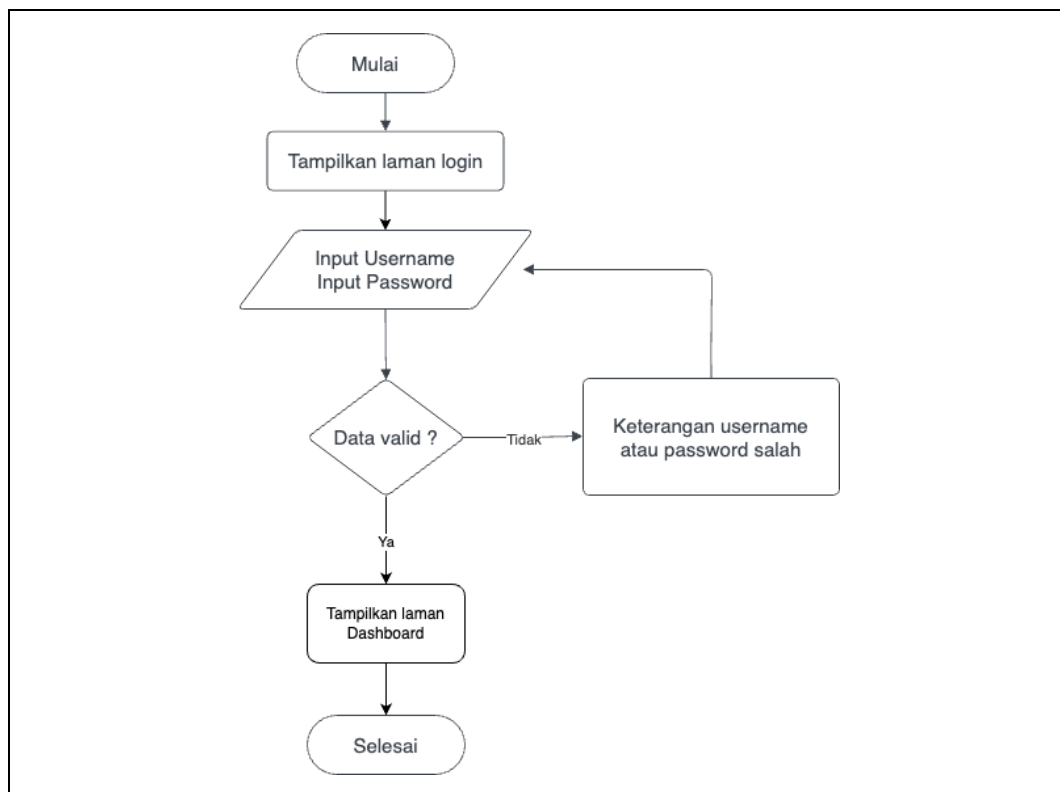
Gambar 3.18 DFD level 2 proses perilsan surat izin penelitian.

Pada Gambar 3.18 dapat dilihat proses dari perilisan surat izin penelitian yang digambarkan menggunakan diagram rinci. Pada proses tersebut entitas BRIDA NTB akan menginput data surat izin penelitian kedalam sistem. Setelah disimpan, data surat izin penelitian tersebut kemudian dibuat menjadi file dalam format PDF yang diunggah kembali oleh BRIDA NTB ke dalam sistem untuk di unduh nantinya oleh entitas Publik.

### 3.2.3.3.2 *Flowchart Diagram*

Pada sistem yang dibuat terdapat beberapa fitur yang digunakan untuk menyelesaikan permasalahan pada sistem. Fitur – fitur ini memiliki alur proses yang digambarkan menggunakan *flowchart* diagram. Fitur – fitur tersebut adalah :

1. Fitur login

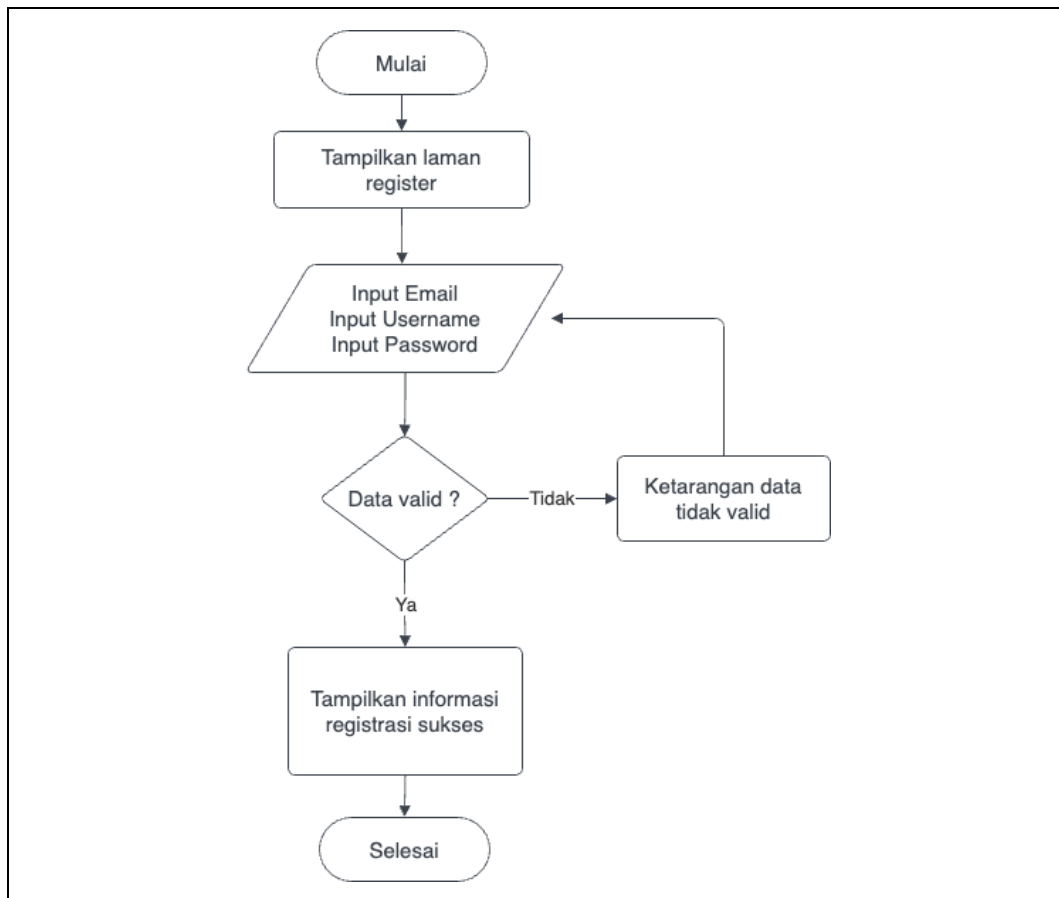


Gambar 3.19 *Flowchart* login.

Pada Gambar 3.19 *Flowchart* login. merupakan alur dari fitur login pada sistem, dimana pengguna akan diminta untuk memasukkan data *username* dan *password* yang kemudian divalidasi oleh sistem, jika *username* dan *password* tidak valid sistem akan menampilkan keterangan *username* atau *password* salah sedangkan jika valid, sistem akan menampilkan dashboard dari sistem.



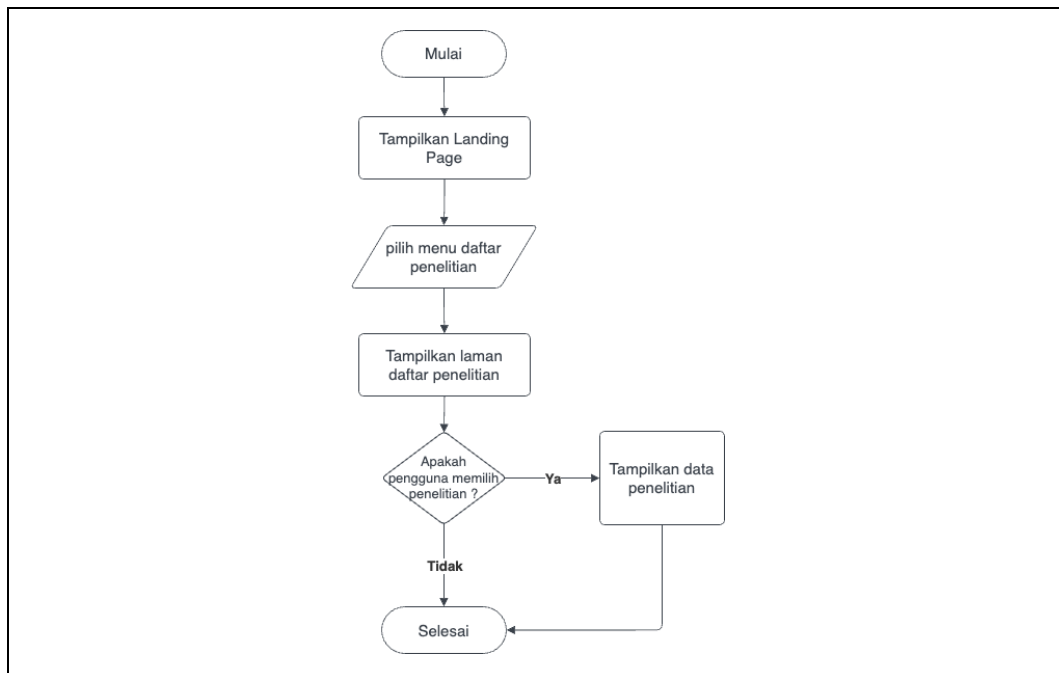
## 2. Fitur Registrasi



Gambar 3.20 *Flowchart* registrasi akun.

Pada Gambar 3.20 *Flowchart* registrasi akun, dapat dilihat alur dari proses registrasi akun. Sistem akan meminta pengguna untuk memasukkan alamat *email*, *username* dan *password* yang akan digunakan nantinya pada saat login. Data yang diinputkan tersebut nantinya akan divalidasi oleh sistem. Setiap isian data memiliki aturan validasi yang berbeda. Hal ini dilakukan untuk memastikan data yang diinput oleh pengguna merupakan data yang valid. Jika data yang diinputkan tidak valid, sistem akan menampilkan keterangan data tidak valid dan pengguna akan diminta kembali memasukkan data sesuai dengan aturan validasi yang sudah ditentukan oleh sistem. Jika data yang diinputkan sudah valid, sistem akan menampilkan informasi registrasi sukses.

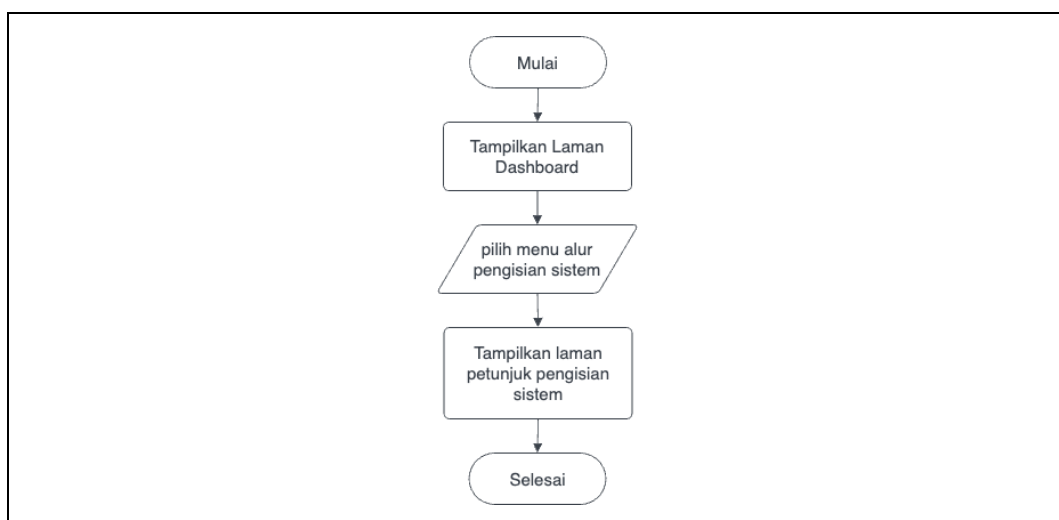
### 3. Fitur Melihat Penelitian Terdaftar



Gambar 3.21 *Flowchart* melihat data penelitian terdaftar.

Pada Gambar 3.21 *Flowchart* melihat data penelitian terdaftar. dapat dilihat *flowchart* dari fitur melihat daftar penelitian yang sudah terdaftar pada sistem. Fitur ini bisa diakses oleh pengguna secara langsung tanpa harus melakukan autentikasi terlebih dahulu. Hal ini dikarenakan fitur ini bertujuan untuk memberikan informasi mengenai penelitian - penelitian yang sudah dilakukan dan sudah terdaftar pada sistem kepada publik.

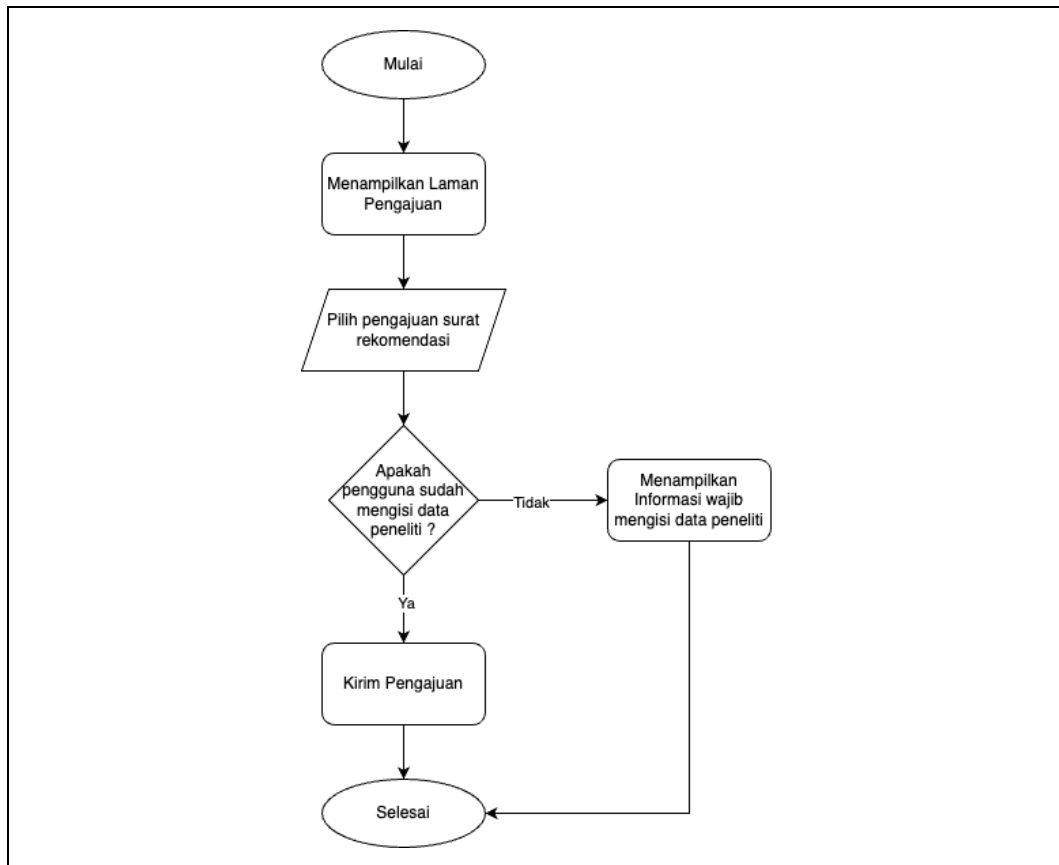
### 4. Fitur Melihat Petunjuk Pengisian



Gambar 3.22 *Flowchart* melihat alur pengisian.

Pada Gambar 3.22 dapat dilihat alur mengenai pengaksesan fitur melihat alur pengisian sistem. Fitur ini dapat diakses langsung oleh pengguna melalui menu pada *dashboard*. Fitur ini bertujuan untuk memberikan informasi mengenai petunjuk pengisian data pada sistem.

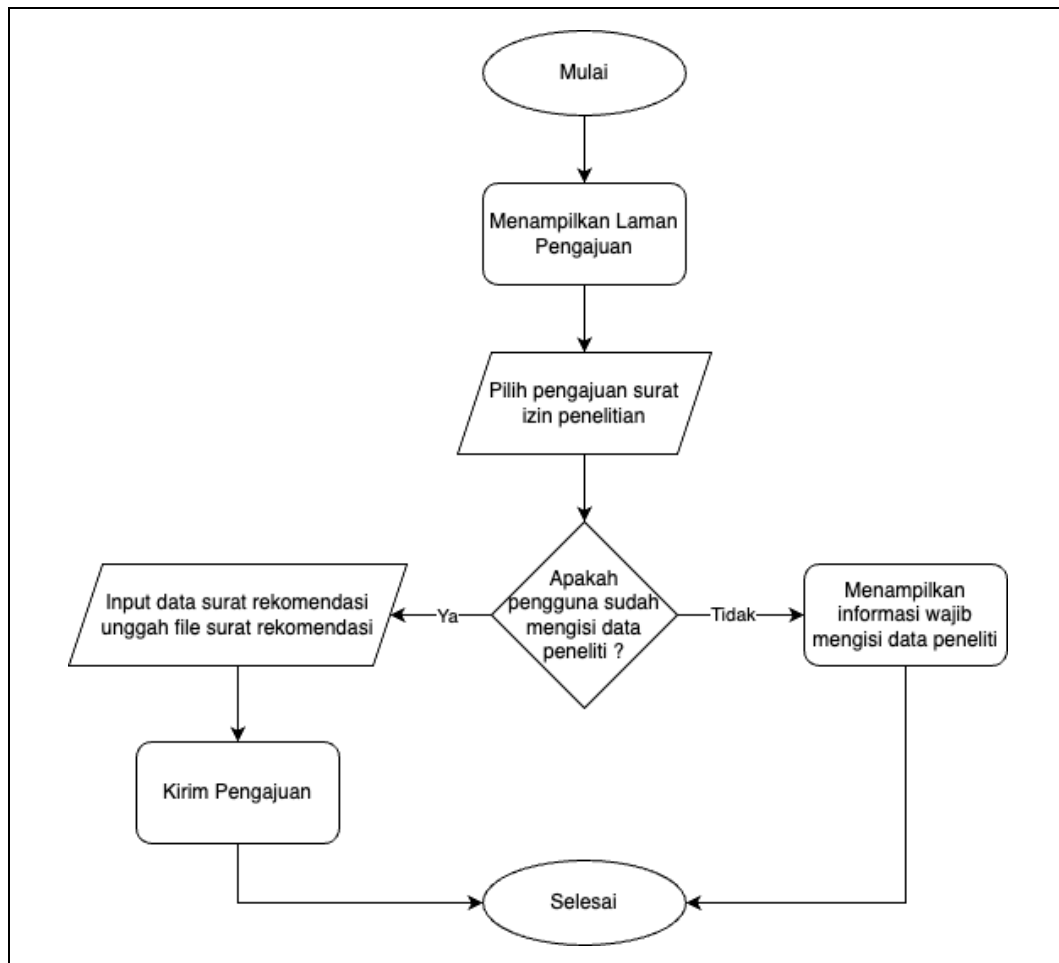
## 5. Fitur Pengajuan Surat Rekomendasi Penelitian



Gambar 3.23 *Flowchart* pengajuan surat rekomendasi penelitian.

Pada Gambar 3.23 dapat dilihat *flowchart* dari fitur pengajuan surat rekomendasi penelitian. Fitur ini dapat diakses langsung oleh pengguna dengan memilih menu pengajuan pada laman *dashboard*. Sebelum mengirim pengajuan surat rekomendasi, sistem akan melakukan pengecekan data pengguna terlebih dahulu. Jika pengguna belum menginput data peneliti seperti identitas peneliti, instansi peneliti dan data penelitian, maka sistem akan meminta pengguna untuk mengisi data peneliti terlebih dahulu. Jika sudah, sistem akan mengirim pengajuan surat rekomendasi tersebut kepada BAKESBANG POLDAGRI.

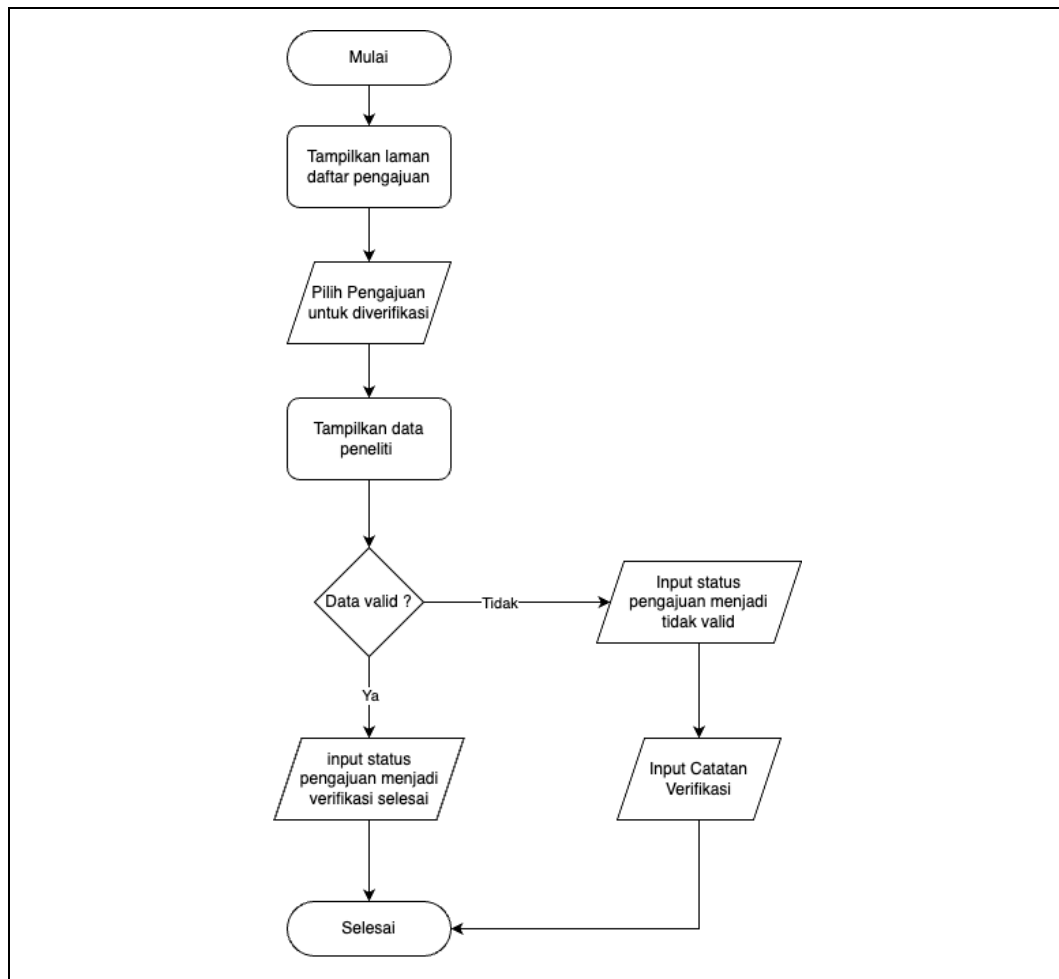
## 6. Fitur Pengajuan Surat Izin Penelitian



Gambar 3.24 *Flowchart* pengajuan surat izin penelitian.

Pada Gambar 3.24 dapat dilihat *flowchart* dari fitur pengajuan surat izin penelitian. Pada fitur ini, sistem akan melakukan pengecekan terhadap data pengguna. Jika pengguna belum mengisi data peneliti, sistem akan menampilkan informasi untuk wajib mengisi data peneliti. Jika sudah, sistem akan meminta pengguna untuk mengisi data surat rekomendasi penelitian dan menunggah file surat rekomendasi penelitian yang sudah didapatkan dari BAKESBANG POLDAGRI tersebut kedalam sistem. Setelah itu, sistem akan mengirim pengajuan tersebut untuk ditinjau nantinya oleh BRIDA NTB.

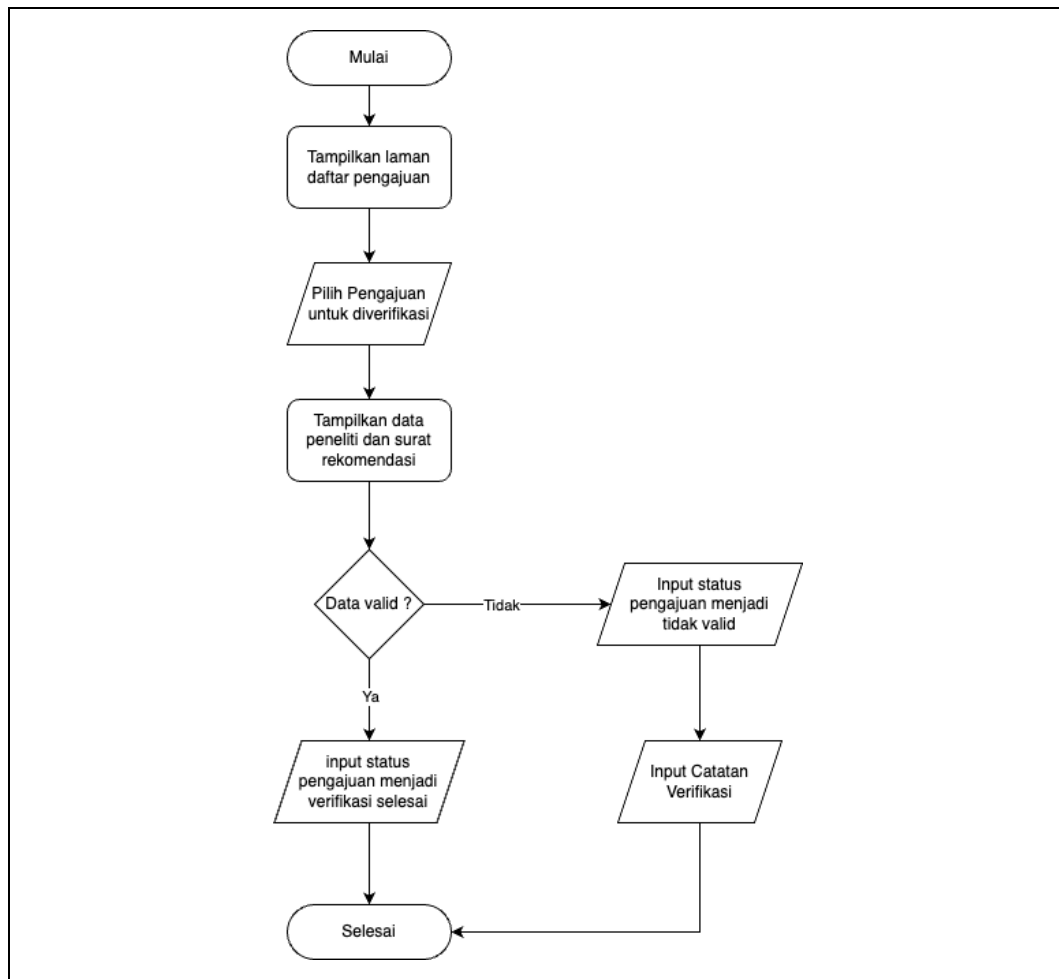
## 7. Fitur Verifikasi Pengajuan Surat Rekomendasi Penelitian



Gambar 3.25 *Flowchart* verifikasi pengajuan oleh BAKESBANG POLDAGRI.

Pada Gambar 3.25 dapat dilihat *flowchart* verifikasi pengajuan surat rekomendasi oleh BAKESBANG POLDAGRI. Pada fitur ini, pengguna dapat melakukan validasi terhadap pengajuan surat rekomendasi yang sudah tersimpan pada sistem. Untuk mengakses fitur ini, pengguna dapat memilih menu daftar pengajuan pada *dashboard*. Saat melakukan verifikasi, pengguna dapat melihat seluruh data peneliti yang melakukan pengajuan surat rekomendasi. Data peneliti tersebut yaitu identitas peneliti, instansi peneliti dan data penelitian. Jika data peneliti tersebut tidak valid, pengguna dapat memilih status tidak valid dan memasukkan alasan ketidakvalidan tersebut pada catatan verifikasi. Sebaliknya jika data peneliti tersebut valid, pengguna dapat memperbarui status pengajuan menjadi verifikasi selesai.

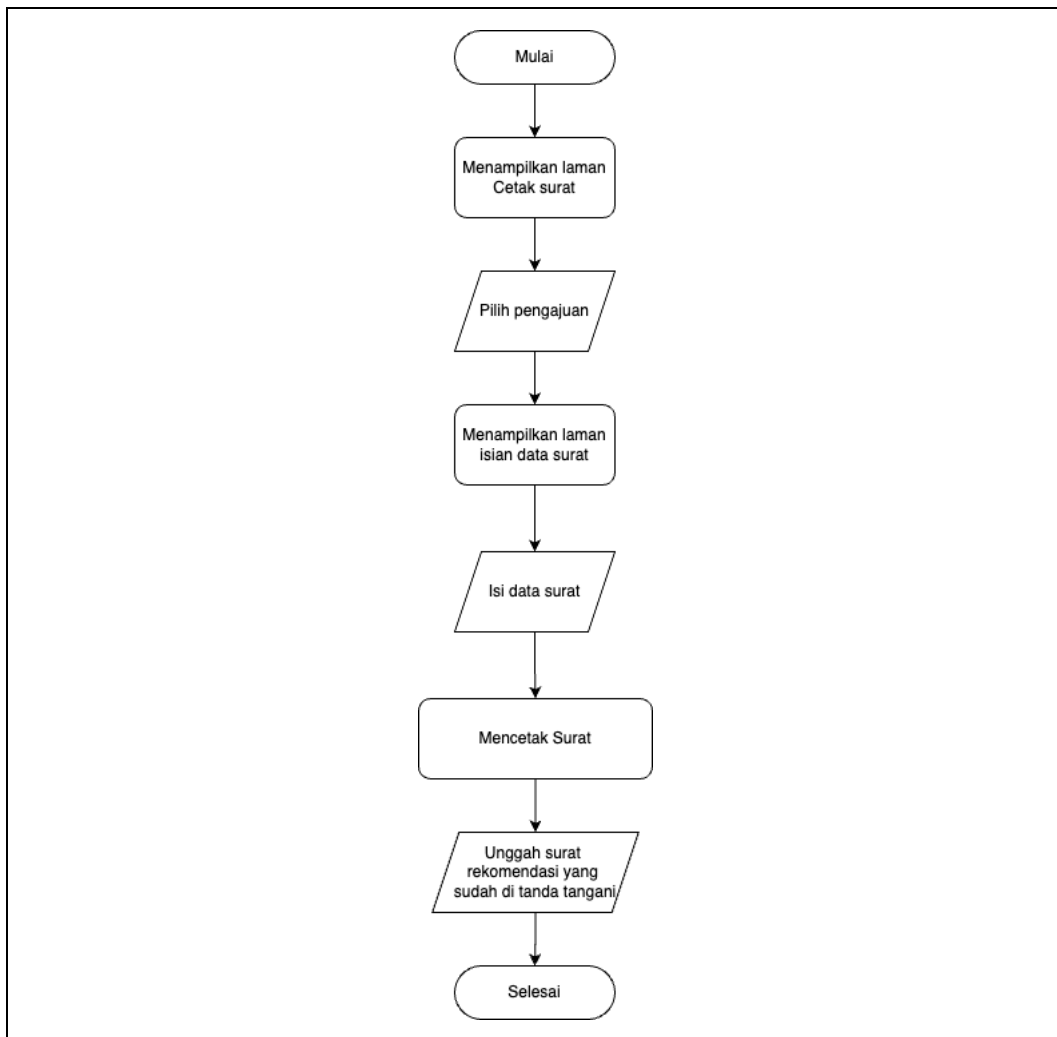
## 8. Fitur Verifikasi Pengajuan Surat Izin Penelitian



Gambar 3.26 *Flowchart* verifikasi pengajuan surat izin penelitian.

Pada Gambar 3.26 dapat dilihat *flowchart* verifikasi pengajuan surat rekomendasi oleh BRIDA NTB. Pada fitur ini, pengguna dapat melakukan validasi terhadap pengajuan surat izin penelitian yang sudah tersimpan pada sistem. Untuk mengakses fitur ini, pengguna dapat memilih menu daftar pengajuan pada *dashboard*. Saat melakukan verifikasi, pengguna dapat melihat seluruh data peneliti yang melakukan pengajuan surat izin penelitian beserta dengan surat rekomendasi yang didapatkan dari BAKESBANG POLDAGRI. Jika data peneliti tersebut tidak valid, pengguna dapat memilih status tidak valid dan memasukkan alasan ketidakvalidan tersebut pada catatan verifikasi. Sebaliknya jika data peneliti tersebut valid, pengguna dapat memperbarui status pengajuan menjadi verifikasi selesai.

## 9. Fitur Cetak Surat Izin Penelitian.

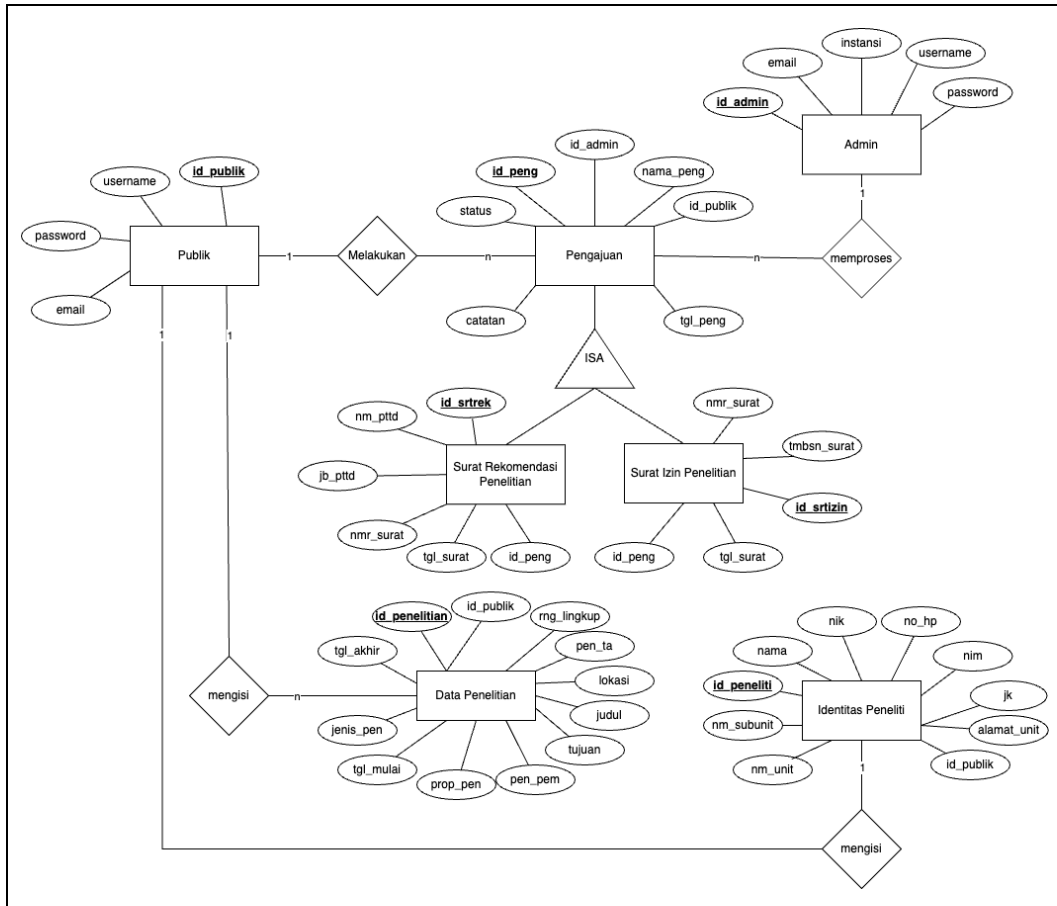


Gambar 3.27 *Flowchart* fitur cetak surat izin penelitian.

Pada Gambar 3.27 dapat dilihat proses pencetakan surat izin penelitian yang dilakukan oleh BRIDA NTB. Pengguna dapat mengakses fitur ini dengan memilih menu cetak surat pada *dashboard*. Pada laman cetak surat berisi pengajuan surat izin penelitian yang telah melewati proses verifikasi sebelumnya. Pengguna harus memilih pengajuan penelitian yang ingin diberikan surat izin penelitian. Setelah itu pengguna akan diminta untuk mengisi data surat izin penelitian yang akan dicetak.

### 3.2.3.3.3 Entity Relationship Diagram (ERD)

Gambaran relasi dari setiap entitas yang terdapat pada aplikasi digambarkan menggunakan *Entity Relationship Diagram (ERD)* dibawah ini :



Gambar 3.28 Gambar Entity Relationship Diagram (ERD)

### 3.2.4 Coding

Tahapan *coding* merupakan tahapan pembuatan aplikasi dengan menggunakan bahasa pemrograman untuk menulis *script code* sesuai dengan alur pengerjaan aplikasi/sistem. Pada tahapan ini peneliti menggunakan bahasa pemrograman *javascript* dengan menggunakan library ReactJS untuk sisi FrontEnd. Lalu untuk sisi BackEnd peneliti menggunakan bahasa pemrograman *javascript* dengan memanfaatkan *Headless Architecture* yang dimiliki oleh Strapi. Dengan fitur ini peneliti bisa lebih fokus untuk menyelesaikan alur sistem pada sisi *FrontEnd*, karena pada sisi *BackEnd* sudah diatasi oleh CMS dari Strapi.



### **3.2.5 Testing**

Pada tahapan ini, peneliti melakukan *testing* aplikasi untuk memastikan aplikasi berjalan sesuai dengan yang diinginkan. Peneliti lebih berfokus untuk melakukan *testing* dari segi fungsionalitas sistem daripada tampilan sistem. Pada tahap ini peneliti menggunakan metode *blackbox testing*, dan *user acceptance testing (UAT)* untuk melakukan testing aplikasi. Metode *blackbox testing* digunakan untuk melakukan *testing* secara garis besar tanpa melihat *script code* yang digunakan. Sedangkan untuk *user acceptance testing* digunakan untuk memastikan bahwa sistem yang dikembangkan telah memenuhi harapan pengguna.

Dalam *blackbox testing*, peneliti menyediakan beberapa pertanyaan yang nantinya akan diisi oleh responden berdasarkan hasil pengujian saat menggunakan sistem. Masing – masing pertanyaan nantinya akan terdapat 5 level sesuai dengan bobot di masing – masing level. Hal ini untuk menentukan apakah sistem sudah berjalan sesuai dengan yang diinginkan atau tidak.

Untuk *user acceptance testing (UAT)*, peneliti menggunakan metode *questioner* yang akan diberikan kepada mahasiswa dan masyarakat umum untuk memberikan penilaian sistem dari sisi peneliti yang akan mengajukan izin penelitian, lalu *questioner* juga diberikan kepada admin yang ada dari BRIDA NTB untuk memberikan penilaian sistem dari sisi admin OPD baik untuk BRIDA NTB maupun BAKESBANGPOLDAGRI.

### **3.2.6 Maintenance**

Pada tahap ini, peneliti melakukan *maintenance* atau perawatan terhadap sistem yang dibuat. Peneliti akan melakukan perbaikan terhadap kesalahan minor maupun major yang dapat mempengaruhi performa dari sistem. Pada tahapan ini juga dilakukan pengembangan pada setiap fitur maupun fungsi yang berjalan pada sistem. Hal ini biasanya dilakukan untuk menjaga performa dari sistem yang dibuat.

## BAB IV

### HASIL DAN PEMBAHASAN

Setelah melewati tahap perancangan dan penentuan metodologi, peneliti selanjutnya melakukan implementasi sistem dengan menggunakan metode penelitian yang telah dijabarkan pada bab sebelumnya. Pada tahapan implementasi ini akan dilakukan implementasi arsitektur JAMStack dimana pada sisi *frontend* menggunakan *framework* NextJS dan pada sisi *backend* menggunakan *Headless* CMS Strapi.

#### 4.1 Implementasi Arsitektur

Untuk membangun sistem yang menerapkan arsitektur JAMStack terdapat beberapa hal yang perlu diperhatikan. Hal – hal tersebut antara lain :

1. Menggunakan teknologi SSG (Static Site Generation)

Arsitektur JAMStack biasanya menghasilkan situs web dengan kinerja yang sangat cepat karena laman yang dibuat menggunakan teknologi SSG akan di-*cache* dan didistribusikan melalui Content Delivery Network (CDN). Dengan menerapkan SSG, terdapat beberapa keuntungan yang didapatkan, seperti kinerja aplikasi yang lebih baik, keamanan yang ditingkatkan, dan sumber daya yang digunakan lebih efisien.

2. Menggunakan teknologi *frontend* untuk membangun tampilan

Arsitektur JAMStack biasanya menggunakan berbagai teknologi *frontend* untuk membangun tampilan, seperti React, Vue, atau Angular, yang memungkinkan pengembangan yang lebih fleksibel dan modern.

3. Menggunakan teknologi *backend* yang mendukung

Bagian *backend* dari arsitektur JAMStack biasanya digunakan untuk menyediakan data melalui API, seperti REST API atau GraphQL, yang diperlukan untuk ditampilkan oleh *frontend*.

Oleh karena itu untuk membangun sistem menggunakan arsitektur JAMStack, peneliti perlu mempersiapkan teknologi pendukung yang dapat memenuhi kebutuhan arsitektur JAMStack mulai dari sisi *frontend* maupun pada sisi *backend*.

## 4.2 Requirements

Untuk memenuhi kebutuhan arsitektur JAMStack, peneliti memilih menggunakan framework NextJS pada sisi *frontend* dan Strapi pada sisi *backend*. baik NextJS maupun Strapi sudah menyediakan *tools* bawaan yang diperlukan pada saat proses *development*.

### 4.2.1 NextJS

Peneliti menggunakan NextJS sebagai framework untuk membangun *user interface* menggunakan React API. Dari beberapa fitur bawaan yang dimiliki oleh NextJS, terdapat beberapa fitur yang dibutuhkan oleh peneliti, antara lain :

#### 4.2.1.1 Static Site Generation (SSG)

*Static site generation* (SSG) merupakan salah satu fitur pada NextJS yang memungkinkan peneliti untuk membuat file statis dari halaman yang sudah ditentukan. Dengan fitur ini, peneliti akan mendapatkan file statis yang memiliki data yang dibutuhkan untuk ditampilkan, termasuk *script* yang dibutuhkan untuk membuat laman tersebut menjadi lebih reaktif. Sehingga file statis yang didapatkan dapat di-*cache* dan didistribusikan secara global melalui *Content Delivery Network* (CDN).

```
1 export async function getStaticPaths() {
2   const response = await fetch("http://localhost:1337/api/pengajuans").then(
3     (res) => res.json()
4   );
5
6   const theData = response.data as PengajuanType[];
7   const paths = theData.map((item) => {
8     return {
9       params: { id: item.id.toString() },
10    };
11  });
12
13  return {
14    paths,
15    fallback: false,
16  };
17 }
```

Gambar 4. 1 Contoh penerapan *static site generation* (SSG).

Fitur ini digunakan oleh peneliti untuk mengolah halaman yang menampilkan data penelitian yang telah terdaftar dalam sistem. Setiap penelitian yang sudah terdaftar dibuat menjadi file statis sesuai dengan data yang ada

menggunakan teknik SSG. Dengan demikian, ketika pengguna mengakses salah satu data penelitian, server akan langsung mengirimkan file statis yang telah dibuat dengan menggunakan teknik SSG, sehingga server tidak perlu melakukan komputasi tambahan untuk menampilkan data penelitian tersebut.

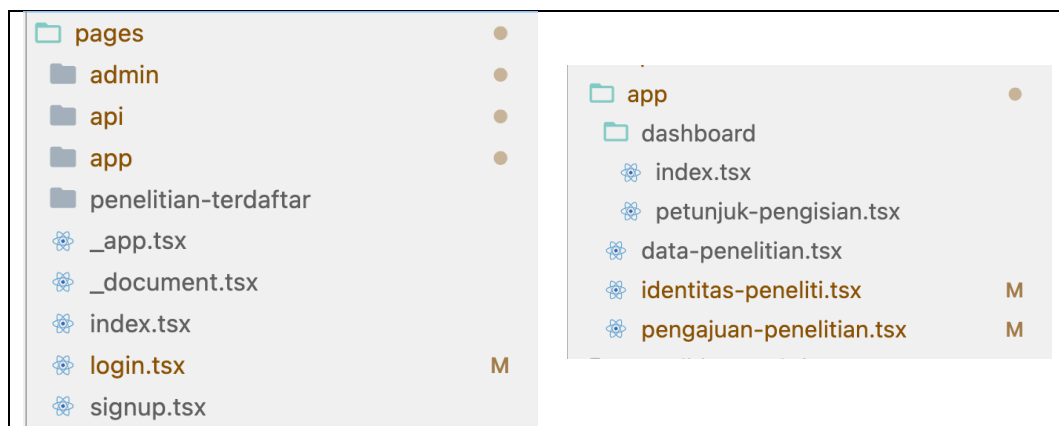
#### 4.2.1.2 Client Side Rendering (CSR)

*Client side rendering* (CSR) merupakan salah satu fitur atau teknik yang dapat diterapkan pada NextJS selaku *framework* yang dibangun menggunakan React API. Dengan menggunakan CSR, segala bentuk reaktifitas halaman pada sistem dapat dilakukan pada sisi *client* tanpa harus melalui proses pada sisi *server*. Sehingga hal ini dapat mengurangi komputasi pada *server*.

Fitur ini digunakan oleh peneliti untuk mengolah halaman yang memerlukan reaktivitas saat berinteraksi dengan pengguna. Dalam sistem yang dibuat, fitur ini digunakan untuk mengatur reaktivitas dari sisi pengguna. Sehingga segala perubahan tampilan dan fungsi lain yang terdapat pada halaman yang diakses oleh pengguna saat menggunakan sistem, diproses pada sisi client tanpa melibatkan komputasi tambahan pada server.

#### 4.2.1.3 File Based Routing

*File based routing* merupakan fitur yang dimiliki oleh NextJS untuk mengatur *pages route* pada sistem. Fitur ini nantinya akan membuat routes secara otomatis sesuai dengan posisi *file* pada folder *pages*. Dengan fitur ini, peneliti dapat membuat *routes* dari setiap laman tanpa melakukan konfigurasi secara manual.

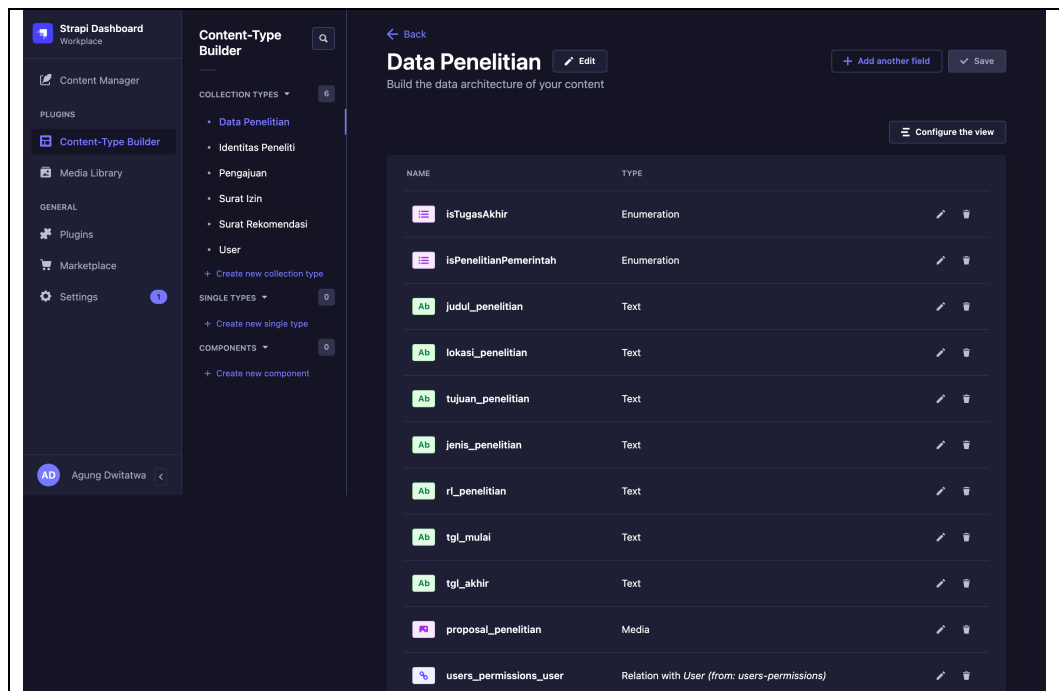


Gambar 4.2 Contoh penerapan *file based routing*.

#### 4.2.1.4 React API

React API merupakan salah satu fitur yang bisa diakses pada NextJS yang memungkinkan peneliti untuk mengguakan API yang sudah disediakan oleh *library* React. Fitur ini digunakan oleh peneliti untuk mengatur *state* dan *life cycle* dari komponen – komponen yang dibuat untuk membangun halaman yang interaktif.

#### 4.2.2 Strapi



Gambar 4. 3 Contoh gambar *dashboard* strapi dan *collection* strapi.

Selanjutnya, peneliti menggunakan strapi untuk membangun aplikasi pada sisi *Backend*. Fitur – fitur yang dimiliki oleh strapi dapat mempermudah peneliti untuk membangun aplikasi pada sisi *Backend* dengan menyediakan *abstraction layer* untuk mengatur database. Fitur fitur yang digunakan oleh peneliti untuk membangun sistem antara lain :

##### 4.2.2.1 Collection

Fitur *collection* yang dimiliki oleh Strapi digunakan peneliti untuk mengatur skema dari kumpulan data (*data collection*) yang dibutuhkan untuk membangun sistem. Dengan fitur ini juga memungkinkan peneliti untuk membuat REST API secara otomatis sesuai dengan skema yang dibuat pada *collection*.

#### 4.2.2.2 REST API

REST API merupakan salah satu teknik komunikasi yang digunakan untuk menghubungkan aplikasi *client* dengan *server*. Dengan REST API, aplikasi *client* dapat mengirim permintaan (*request*) kepada *server* dan menerima respon (*response*) berupa data atau hasil operasi yang diperlukan untuk berinteraksi dengan *server* secara efisien. Pada Strapi, REST API dibuat secara otomatis berdasarkan *collection* yang sudah dibuat sebelumnya. Hal ini memudahkan peneliti saat membangun sistem, dimana peneliti tidak perlu melakukan konfigurasi REST API secara manual.

#### 4.2.2.3 Authentication & Authorization

*Authentication* dan *Authorization* adalah teknik yang digunakan untuk mengatur hak akses pengguna saat mereka mengakses sistem. Dalam Strapi, fitur-fitur ini tersedia secara otomatis, sehingga peneliti dapat dengan mudah memanfaatkannya tanpa perlu melakukan konfigurasi manual.

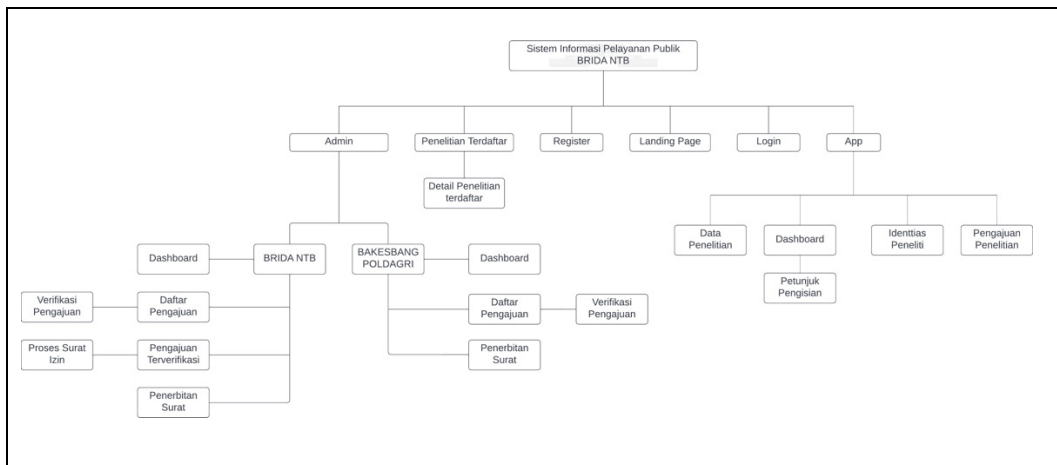
#### 4.2.2.4 Upload API

*Upload API* merupakan salah satu fitur bawaan yang dimiliki oleh Strapi untuk melakukan *file management*. Fitur ini digunakan oleh peneliti untuk memproses file yang dibutuhkan pada sistem. Dengan adanya fitur ini, peneliti dapat melakukan *file management* dengan lebih mudah.

### 4.3 Implementasi Coding Frontend

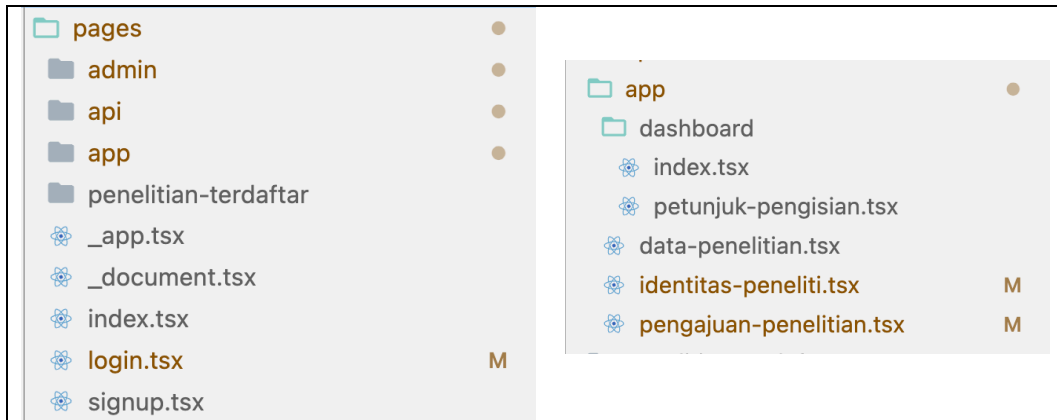
Dalam tahap ini, peneliti melakukan *coding* untuk mengimplementasikan tampilan yang sudah dirancang dalam wireframe yang telah dibuat pada bab sebelumnya. Sesuai dengan kebutuhan dalam membangun sistem berdasarkan arsitektur JAMStack, peneliti memanfaatkan framework Next.js untuk membangun aplikasi pada sisi frontend. Selama tahap ini, peneliti membuat berbagai komponen yang membentuk setiap halaman yang ada dalam sistem. Komponen-komponen ini dibangun menggunakan bahasa pemrograman TypeScript dengan memanfaatkan React API untuk menjadikan komponen-komponen tersebut lebih interaktif. Untuk mempermudah pembuatan komponen yang dibutuhkan, peneliti membangun *sitemap* dari sistem yang dibuat.

### 4.3.1 Sitemap Sistem

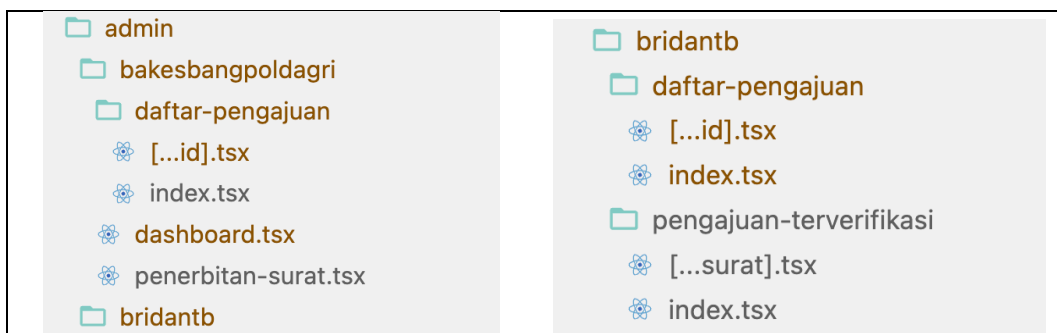


Gambar 4. 4 Sitemap sistem informasi pelayanan public BRIDA NTB

Pada Gambar Gambar 4. 4 dapat dilihat *sitemap* dari sistem informasi pelayanan publik yang akan dibuat. *Sitemap* ini akan menjadi acuan pada saat peneliti membuat *page routes* dari sistem dengan menggunakan fitur *file based routing* yang dimiliki oleh NextJS. Pembuatan laman sesuai dengan *sitemap* dapat dilihat pada Gambar 4. 5 dan Gambar 4. 6 dibawah ini.



Gambar 4. 5 Implementasi *file based routing*.



Gambar 4. 6 Implementasi *file based routing*.

### 4.3.2 Pembuatan Komponen

Setelah membuat laman, selanjutnya peneliti membuat komponen – komponen yang menjadi unit pembangun dari masing masing laman. Komponen yang dibuat dibagi menjadi 2 kategori, yaitu komponen khusus untuk masing – masing laman, dan komponen global dengan fungsi yang lebih spesifik. Dibawah ini merupakan salah satu contoh pembuatan komponen.

```
1 import SaveButtonIcon from "@svg-icons/form-app-icons/SaveButtonIcon";
2
3 type Props = {
4   label: string;
5   className?: string;
6   type?: "submit" | "button" | "reset";
7 };
8
9 export default function FormButton({ label, type, className }: Props) {
10  return (
11    <button
12      type={type}
13      className={`text-sm bg-slate-800 text-gray-300 px-6 py-2 font-medium rounded-md
14        ${className} flex items-center gap-2 hover:bg-slate-700`}
15    >
16      <SaveButtonIcon className="w-3" />
17      {label}
18    </button>
19  );
20 }
21
```

Gambar 4. 7 Contoh pembuatan komponen.

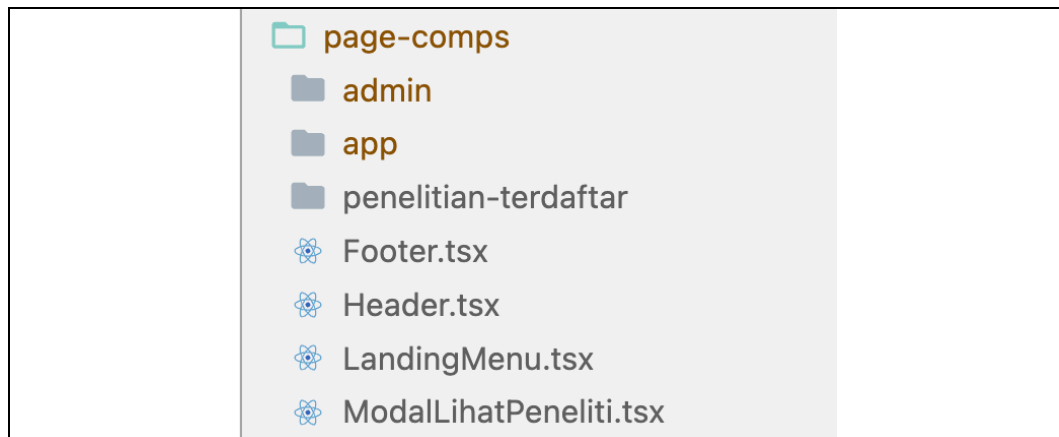
Pada Gambar 4. 7 dapat dilihat contoh pembuatan komponen yang akan menjadi unit pembangun pada setiap laman. Pada pembuatan komponen ini peneliti menggunakan konsep *functional component* dimana fungsi yang dibuat tersebut harus mengembalikan Typescript XML (TSX) yang merupakan tag XML yang sudah dimodifikasi untuk membuat *interface* layaknya HTML pada *website*.

```
1 export default function FormButton({ label, type, className }: Props) {
2  return (
3    <button
4      type={type}
5      className={`text-sm bg-slate-800 text-gray-300 px-6 py-2 font-medium rounded-md
6        ${className} flex items-center gap-2 hover:bg-slate-700`}
7    >
8      <SaveButtonIcon className="w-3" />
9      {label}
10    </button>
11  );
12 }
```

Gambar 4. 8 Source code pembuatan komponen

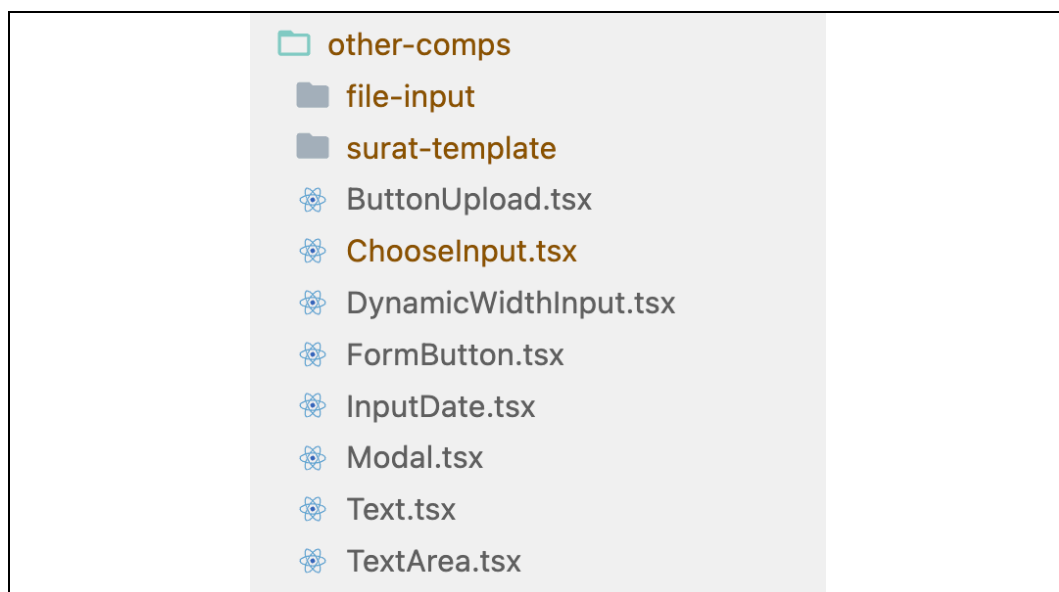
Pada Gambar 4. 8 merupakan contoh *functional component* dengan nama *FormButton* yang mengembalikan elemen TSX berupa *button* yang memiliki *type* dan *className* property serta memiliki *icon* dengan nama *SaveButtonIcon*.





Gambar 4. 9 Pembuatan komponen khusus untuk masing masing laman.

Pada Gambar 4. 9 dapat dilihat komponen – komponen yang digunakan untuk setiap laman pada sistem. Komponen khusus ini memiliki *behavior* masing – masing sesuai dengan fungsi tertentu yang diinginkan pada setiap laman.



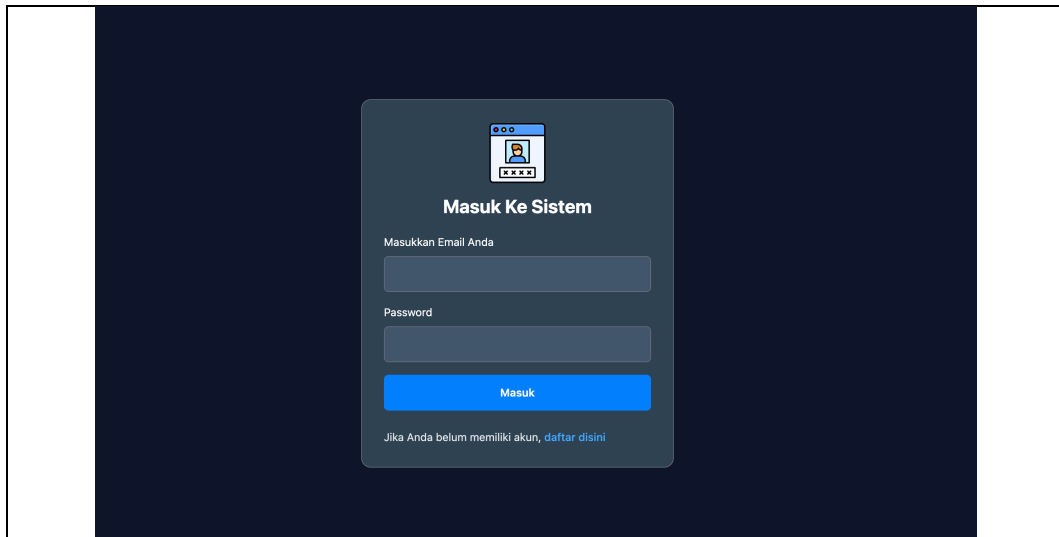
Gambar 4. 10 Pembuatan komponen global

Pada Gambar Gambar 4. 10 dapat dilihat komponen – komponen yang digunakan pada setiap laman pada sistem. Komponen global ini memiliki fungsi yang lebih spesifik daripada komponen khusus. Contohnya komponen Text yang hanya digunakan untuk menerima inputan dari pengguna.

### 4.3.3 Hasil Implementasi Frontend

Pada tahap ini peneliti mulai membangun setiap laman dengan menggunakan komponen – komponen yang dibuat sebelumnya. Berikut merupakan hasil implementasi dari komponen yang sudah dibuat pada setiap laman

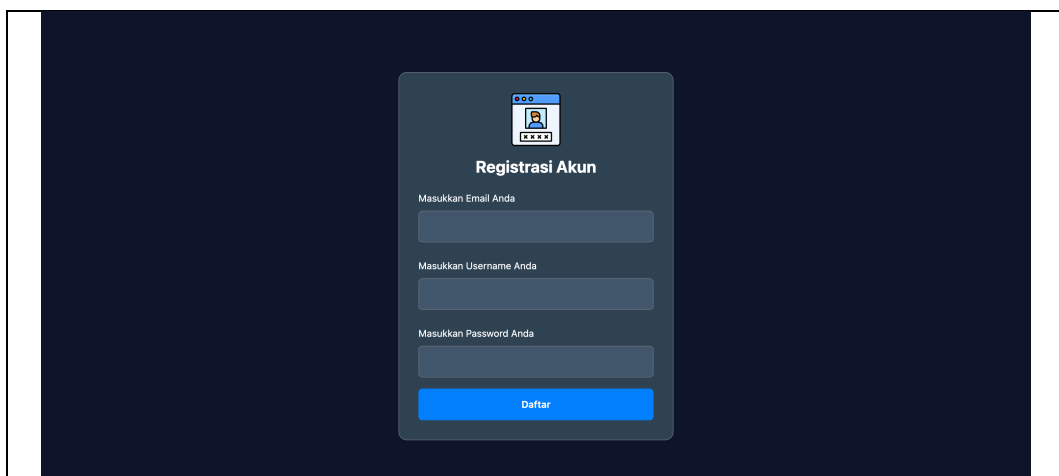
### 4.3.3.1 Login



Gambar 4. 11 Tampilan laman login.

Pada Gambar 4. 11 Tampilan laman login. merupakan desain tampilan dari laman login pada sistem. Pada laman login ini terdapat komponen yang dibuat untuk menerima inputan dari pengguna. Di laman login ini pengguna akan diminta untuk memasukkan *email* dan *password* sesuai dengan akun yang digunakan saat mendaftar. *Email* dan *password* ini nantinya akan dikirim melalui *post request* ke *endpoint* otentikasi yang dibuat menggunakan strapi. *Response* yang didapatkan nantinya akan dikirim kembali komponen untuk diproses sesuai dengan *response status* yang didapatkan.

### 4.3.3.2 Registrasi

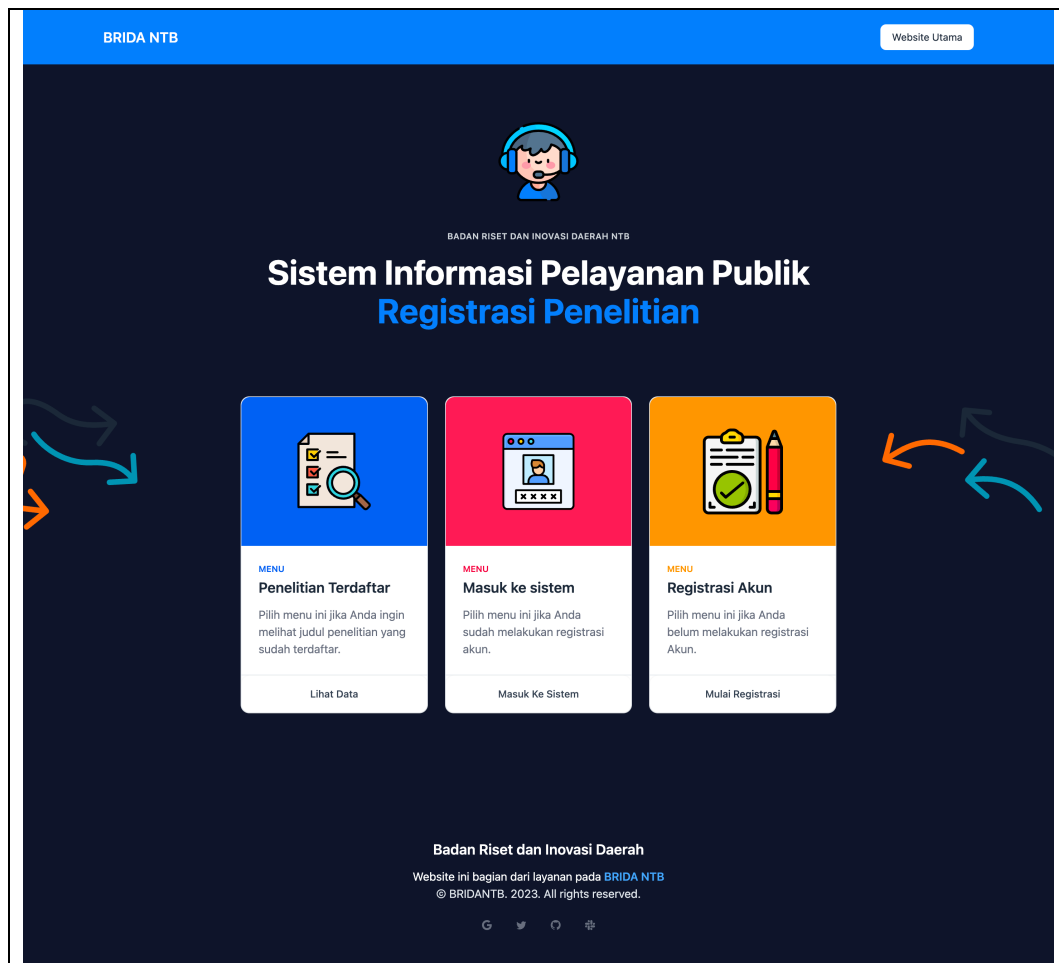


Gambar 4. 12 Tampilan laman registrasi.

Pada Gambar 4. 12 Tampilan laman registrasi. merupakan tampilan dari laman registrasi. Sama seperti laman login, di laman registrasi juga terdapat

komponen yang dibuat untuk menerima inputan dari pengguna. Di laman registrasi pengguna diminta untuk menginput *email*, *username*, dan *password* yang akan menjadi *credentials data* dari akun pengguna. Data yang diinputkan oleh pengguna ini nantinya akan dikirim melalui *post request* ke *endpoint* registrasi yang dibuat menggunakan strapi.

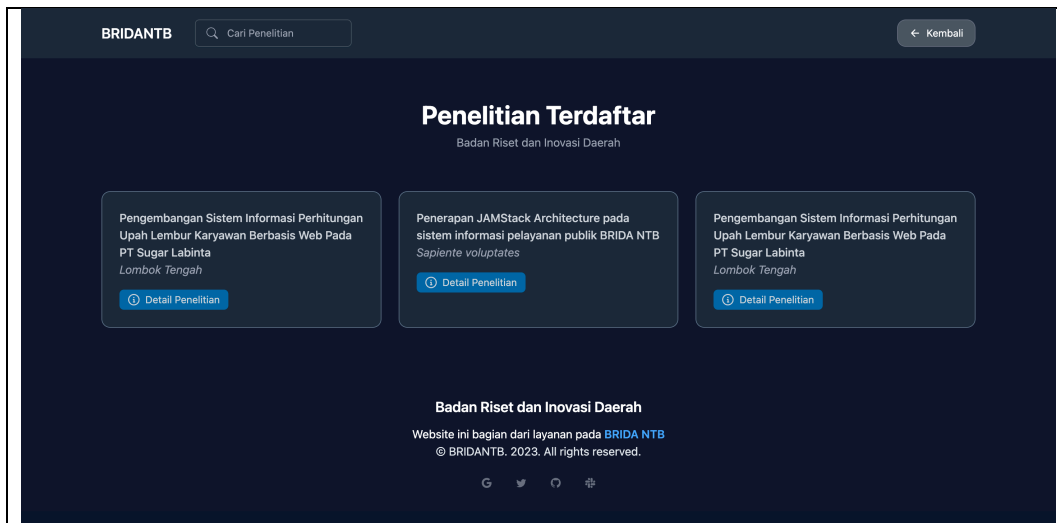
### 4.3.3.3 Landing Page



Gambar 4. 13 Tampilan landing page.

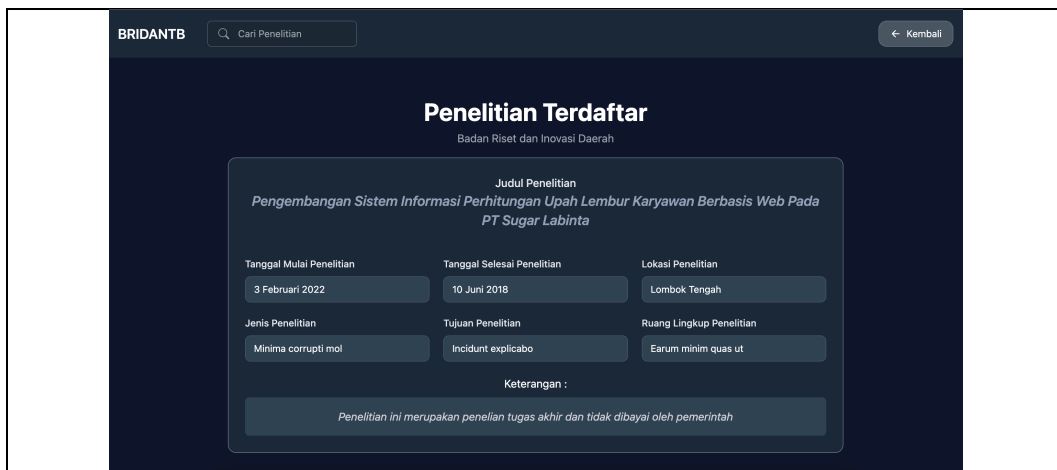
Pada Gambar 4. 13 Tampilan landing page. merupakan tampilan dari *landing page* yang bisa diakses oleh pengguna tanpa melakukan login terlebih dahulu. Pada laman ini, pengguna dapat memilih dari tiga menu yang tersedia, yaitu menu Penelitian Terdaftar, menu Login, dan menu Registrasi. Setiap menu ini akan mengarahkan pengguna ke halaman yang sesuai dengan nama menu tersebut.

#### 4.3.3.4 Laman Penelitian Terdaftar



Gambar 4. 14 Tampilan laman penelitian terdaftar.

Laman penelitian terdaftar yang dapat dilihat pada Gambar 4. 14 Tampilan laman penelitian terdaftar. dibuat untuk menampilkan judul penelitian yang sudah menerima surat izin penelitian dari BRIDA NTB. Pada laman ini pengguna dapat melihat detail dari penelitian terdaftar dengan memilih menu detail penelitian. Nantinya, pengguna akan diarahkan ke laman detail penelitian terdaftar seperti gambar dibawah ini

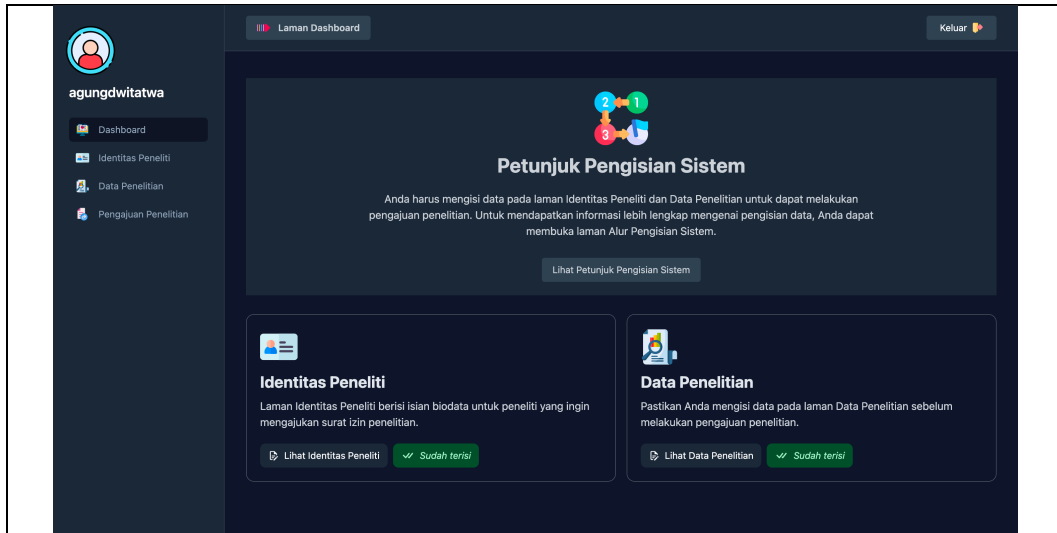


Gambar 4. 15 Tampilan laman detail penelitian terdaftar.

Pada Gambar 4. 15 dapat dilihat tampilan dari detail penelitian yang sudah terdaftar pada sistem. Detail penelitian yang ditampilkan adalah tanggal mulai mulai penelitian, tanggal selesai penelitian, lokasi penelitian, jenis penelitian, tujuan penelitian, ruang lingkup penelitian serta keterangan apakah penelitian

tersebut merupakan tugas akhir atau tidak, dan apakah penelitian tersebut dibiayai menggunakan APBN oleh pemerintah atau tidak.

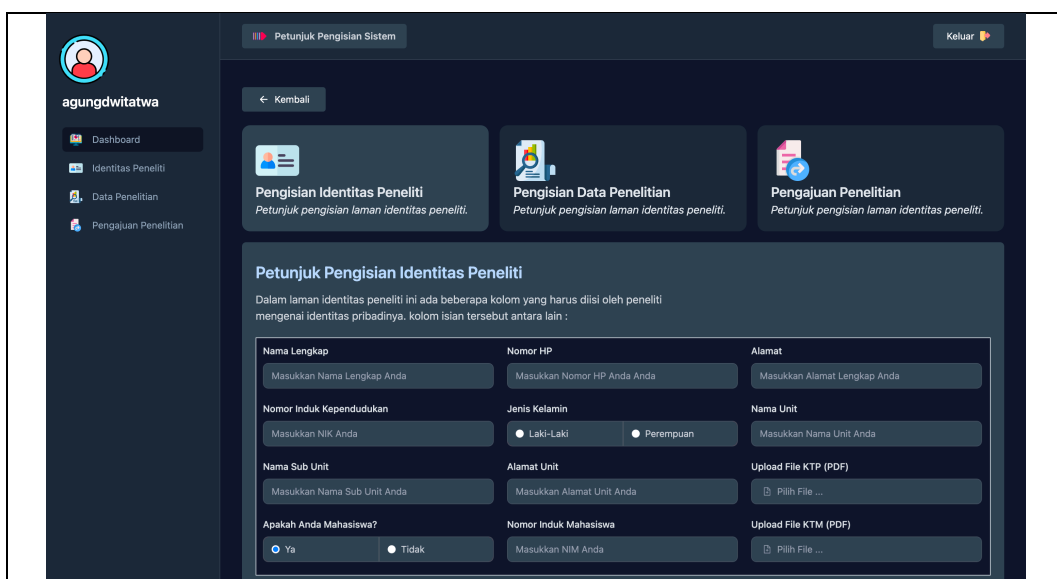
#### 4.3.3.5 Laman Dashboard Peneliti



Gambar 4. 16 Tampilan laman dashboard peneliti.

Gambar 4. 16 Tampilan laman dashboard peneliti. merupakan tampilan dari laman *dashboard* yang akan diakses oleh peneliti. Pada laman ini pengguna akan ditampilkan informasi mengenai pengisian data identitas peneliti dan pengisian data penelitian. Peneliti juga bisa mengakses petunjuk pengisian data dengan cara memilih menu untuk melihat petunjuk pengisian sistem.

#### 4.3.3.6 Laman Petunjuk Pengisian Sistem



Gambar 4. 17 Tampilan laman petunjuk pengisian sistem

Pada Gambar 4. 17 merupakan tampilan dari laman petunjuk pengisian sistem. Pada laman ini pengguna nantinya akan diberikan informasi mengenai pengisian data dari masing – masing laman pada sistem, termasuk juga beberapa hal yang harus diperhatikan oleh peneliti sebelum mengisi data di masing masing laman.

#### 4.3.3.7 Laman Identitas Peneliti

The screenshot shows a web interface for a researcher's identity page. The page is titled "Laman Identitas Peneliti" and contains a form for updating personal and institutional information. The form fields include: Nama Lengkap (Agung Dwitatwa Achmadi), Nomor HP (+6285337273108), Alamat (Jalan Bypass 35 Lombok Barat), Nomor Induk Kependudukan (5202072007990001), Jenis Kelamin (radio buttons for Laki-Laki and Perempuan, with Laki-Laki selected), Nama Unit (Universitas Mataram), Nama Sub Unit (Fakultas Teknik), Alamat Unit (Jalan Pemuda 45, Gomong Mataram), and Upload File KTP (PDF) (with buttons for Lihat File and Hapus File). At the bottom, there is a question "Apakah Anda Mahasiswa?" with radio buttons for Ya (selected) and Tidak, and an Update Data button.

Gambar 4. 18 Tampilan laman identitas peneliti.

Pada Gambar 4. 18 merupakan tampilan dari laman identitas peneliti. Laman identitas ini merupakan tempat di mana peneliti akan diminta untuk mengisi beragam data pribadi yang diperlukan untuk keperluan penelitian. Data pribadi yang harus diisi oleh pengguna meliputi nama lengkap, nomor handphone yang dapat dihubungi, alamat tempat tinggal, nomor induk kependudukan (NIK) sebagai identifikasi resmi, dan juga informasi mengenai jenis kelamin peneliti. Selain itu, laman ini juga meminta pengguna untuk memberikan data terkait instansi atau organisasi tempat pengguna bekerja atau bernaung. Data tersebut terdiri atas nama unit atau departemen di instansi tersebut, selanjutnya, nama sub unit atau divisi jika ada, serta alamat lengkap unit tersebut. Semua kolom isian ini harus diisi secara cermat dan akurat sesuai dengan aturan validasi data yang berlaku.

### 4.3.3.8 Laman Data Penelitian

agungdwitwa

Dashboard

Identitas Peneliti

Data Penelitian

Pengajuan Penelitian

Laman Data Penelitian

Keluar

Laman Data Penelitian

Pada laman ini peneliti wajib mengisi data penelitian yang Akan dilakukan. Pastikan data form dibawah ini terisi dengan data yang valid untuk memperlancar proses validasi data nantinya.

Form Data Penelitian

Petunjuk Pengisian

Apakah penelitian ini Tugas Akhir Anda?

Ya Tidak

Apakah penelitian Anda dibiayai APBD/APBN ?

Ya Tidak

Judul Penelitian

Penerapan JAMStack Architecture pada sist...

Lokasi Penelitian

Sapiente voluptates

Tujuan Penelitian

Qui minima non dolor

Jenis Penelitian

Quibusdam et aut et

Ruang Lingkup Penelitian

Dolores in ullam mol

Masukkan Tanggal Mulai Penelitian

11 Juli 2017

Masukkan Tanggal Akhir Penelitian

11 Juni 2023

Proposal Penelitian

Lihat File Hapus File

Update Data

Gambar 4. 19 Tampilan laman data penelitian.

Pada Gambar 4. 19 merupakan tampilan dari laman data penelitian, dimana pada laman ini peneliti akan diminta untuk mengisi data – data mengenai penelitian yang akan dilakukan. Data – data yang masukkan oleh peneliti harus mengikuti aturan validasi dari masing – masing kolom sebelum menyimpan data.

### 4.3.3.9 Laman Pengajuan Penelitian

testing02

Dashboard

Identitas Peneliti

Data Penelitian

Pengajuan Penelitian

Laman Pengajuan Penelitian

Keluar

Laman Pengajuan

Pada laman ini peneliti melakukan pengajuan untuk proses registrasi penelitian sehingga BRIDA NTB dapat mengeluarkan surat izin penelitian. Mohon untuk memperhatikan informasi yang tertera di form pengajuan.

1 Pengajuan Surat Rekomendasi >> 2 Pengajuan Surat Izin Penelitian >> 3 Download Surat Izin Penelitian

Pengajuan Surat Rekomendasi

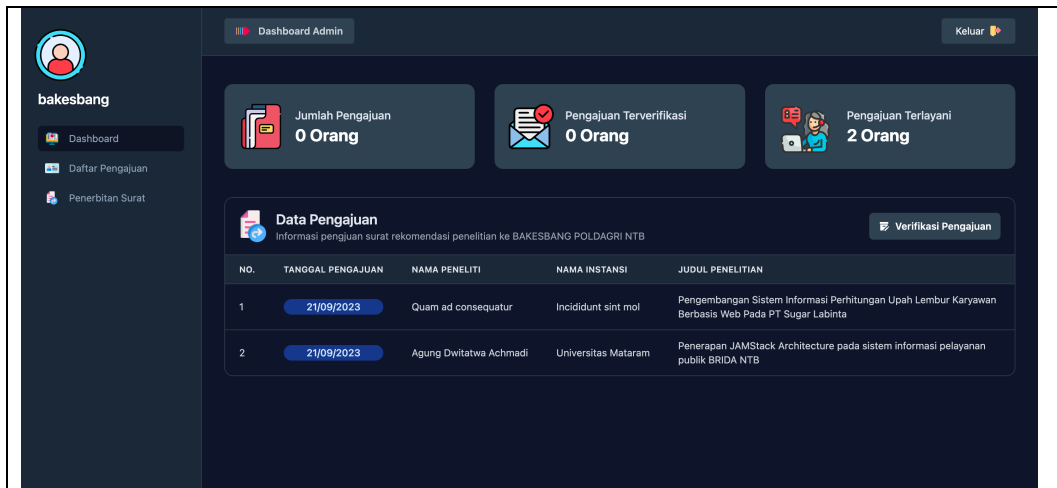
Surat rekomendasi ini diperlukan sebagai persyaratan untuk dapat mengajukan surat izin penelitian kepada BRIDA Provinsi NTB. Pengajuan surat rekomendasi ini dikirim ke BAKESBANGPOLDAGRI Provinsi NTB yang akan mengevaluasi dan memberikan persetujuan awal terhadap rencana penelitian Anda, yang kemudian menjadi dasar bagi Anda untuk memperoleh izin resmi dari BRIDA. Hal ini dilakukan untuk memastikan bahwa penelitian Anda berjalan sesuai dengan peraturan yang berlaku dan menghindari potensi masalah administratif serta hukum yang dapat muncul selama proses penelitian berlangsung di Provinsi NTB.

Ajukan Surat Rekomendasi

Gambar 4. 20 Tampilan laman pengajuan penelitian.

Gambar 4. 20 merupakan tampilan dari laman pengajuan penelitian. Laman ini merupakan tempat peneliti mengajukan surat rekomendasi kepada BAKESBANGPOLDAGRI dan surat izin penelitian kepada BRIDA NTB. Pada laman ini peneliti bisa melihat progres pengajuan layanan yang telah dilakukan.

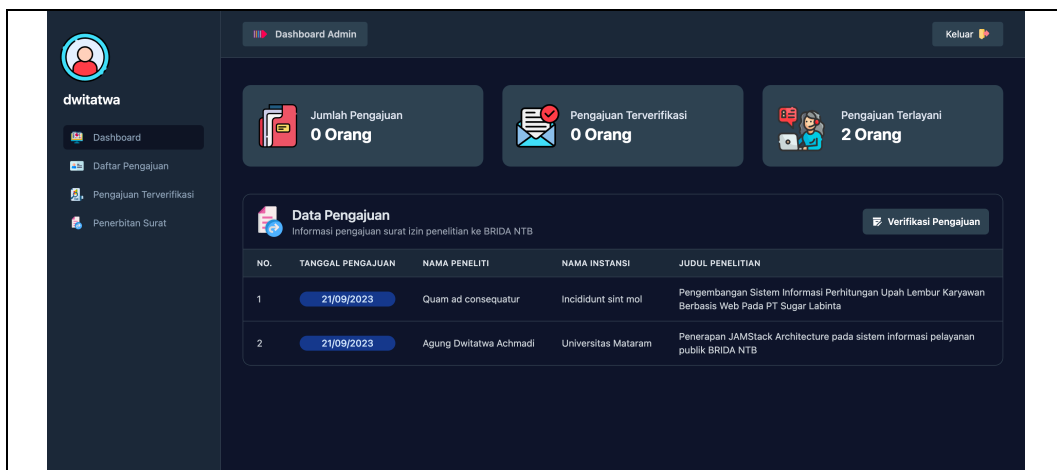
#### 4.3.3.10 Laman Dashboard Admin Bakesbangpoldagri



Gambar 4. 21 Laman dashboard admin BAKESBANGPOLDAGRI.

Gambar 4. 21 merupakan tampilan dari laman dashboard yang akan diakses oleh admin BAKESBANGPOLDAGRI. Pada laman ini terdapat 3 menu yang bisa diakses, yaitu dashboard, daftar pengajuan, dan penerbitan surat.

#### 4.3.3.11 Laman Dashboard Admin BRIDA NTB



Gambar 4. 22 laman dashboard admin BRIDA NTB.

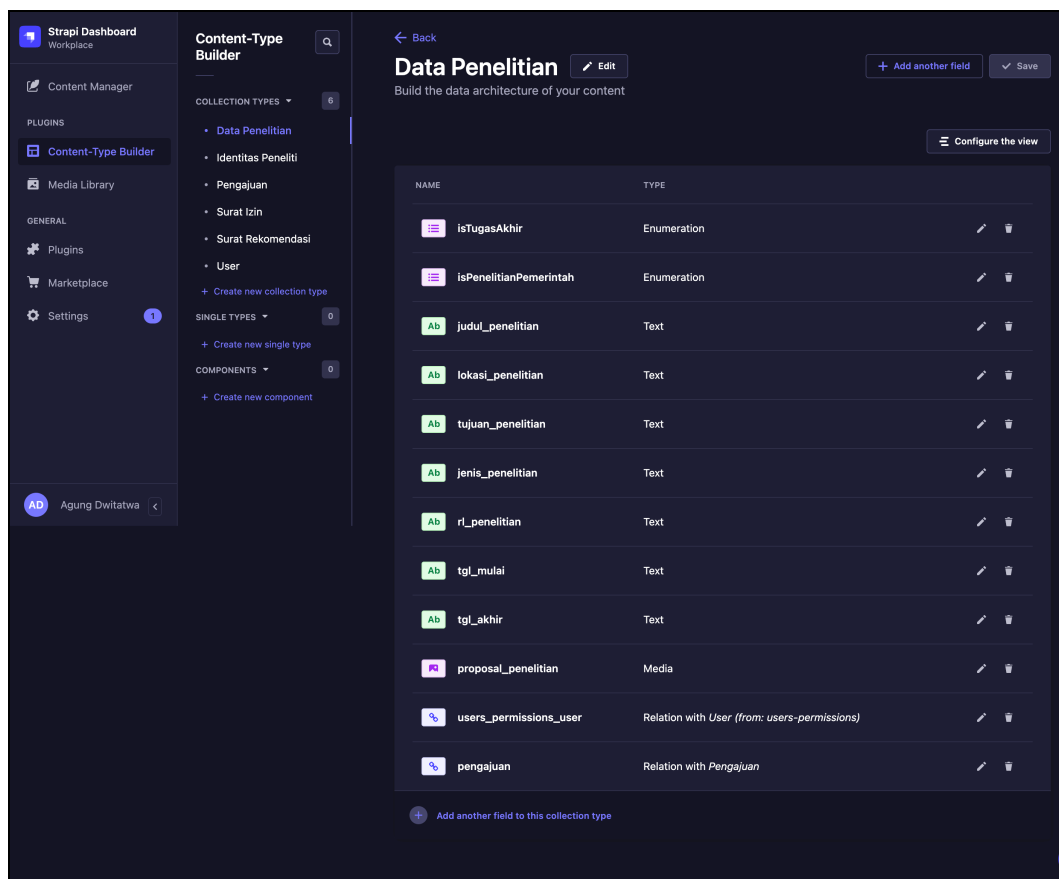
Gambar 4. 22 merupakan tampilan dari laman dashboard admin BRIDA NTB. Pada laman ini terdapat 4 menu yang bisa diakses oleh admin, yaitu menu dashboard, menu daftar pengajuan, menu pengajuan terverifikasi dan penerbitan surat.



## 4.4 Implementasi Backend

Setelah melakukan implementasi pada sisi *frontend*, selanjutnya peneliti melakukan Implementasi pada sisi *backend*. Implementasi pada sisi *backend* dilakukan menggunakan Strapi. Fitur-fitur yang terdapat pada strapi memudahkan peneliti dalam membuat REST API yang dibutuhkan untuk memproses data penelitian. Fitur yang digunakan adalah pembuatan *collection* data yang berfungsi untuk menentukan skema kelompok data yang dibutuhkan pada sistem. Fitur *collection* ini membuat *abstraction layer* antara strapi dengan jenis *database* yang digunakan, sehingga memungkinkan strapi untuk membuat REST API secara otomatis sesuai dengan skema *collection* yang dibuat.

### 4.4.1.1 Pembuatan Data Penelitian Collection

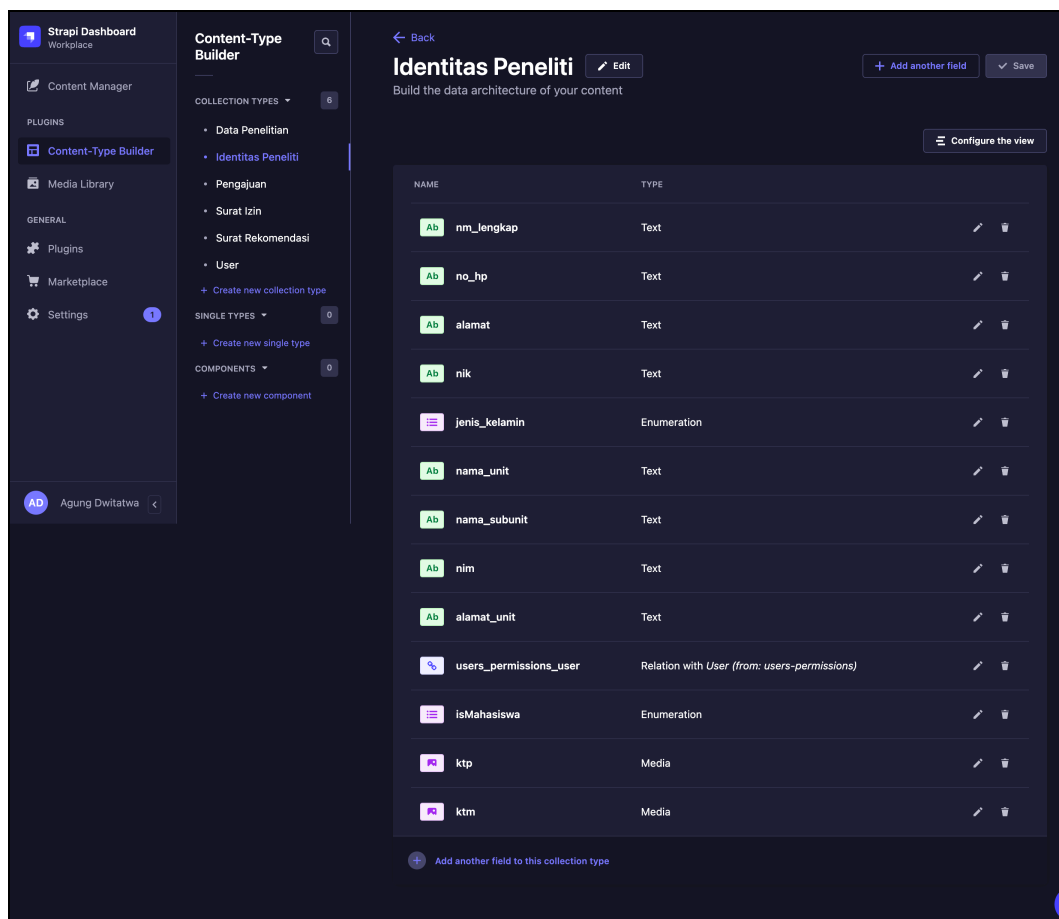


Gambar 4. 23 Skema data penelitian *collection*.

Gambar 4. 23 merupakan skema *collection* untuk data penelitian. Dalam koleksi data penelitian ini, terdapat 7 kolom isian yang semuanya berjenis teks. Kolom-kolom ini mencakup judul penelitian, lokasi penelitian, tujuan penelitian,

jenis penelitian, ruang lingkup penelitian, serta tanggal mulai dan tanggal berakhir penelitian. Selain itu, dalam skema ini juga terdapat 2 kolom isian yang berjenis enumerasi. Yang pertama adalah kolom yang menentukan apakah penelitian ini merupakan tugas akhir atau tidak, dan yang kedua adalah kolom yang digunakan untuk menentukan apakah penelitian ini mendapat pendanaan dari pemerintah atau tidak. Selanjutnya, terdapat satu kolom isian yang berjenis file, yang digunakan untuk menyimpan file proposal penelitian. Selain itu, terdapat juga 2 kolom yang berfungsi untuk menghubungkan data penelitian dengan data pengguna (*user collection*) dan data pengajuan (*pengajuan collection*).

#### 4.4.1.2 Pembuatan Identitas Peneliti Collection

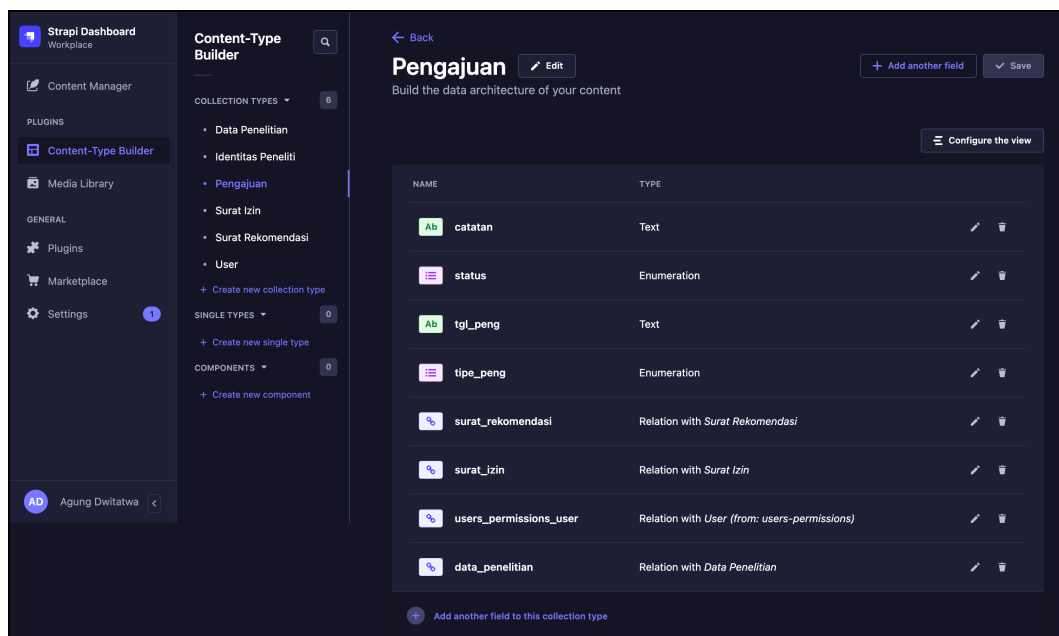


Gambar 4. 24 Skema identitas peneliti *collection*.

Gambar 4. 24 merupakan skema *collection* untuk identitas peneliti. Dalam Identitas Peneliti ini terdapat sejumlah kolom isian berjenis teks yang harus diisi oleh pengguna, seperti nama lengkap, nomor HP, alamat, Nomor Induk

Kependudukan (NIK), nama unit, nama subunit, alamat unit, dan NIM (Nomor Induk Mahasiswa) yang diisi jika pengguna merupakan seorang mahasiswa. Selain itu terdapat dua kolom isian berjenis enumerasi untuk menentukan jenis kelamin dari pengguna dan untuk menentukan apakah pengguna tersebut merupakan mahasiswa atau bukan. Selain informasi teks, terdapat juga dua kolom isian berjenis file yang harus diunggah, yaitu file Kartu Tanda Penduduk (KTP) dan file Kartu Tanda Mahasiswa (KTM) jika pengguna merupakan mahasiswa. Dan yang terakhir, terdapat satu kolom isian khusus yang digunakan untuk menentukan relasi antara Identitas Peneliti *Collection* dengan *User Collection*.

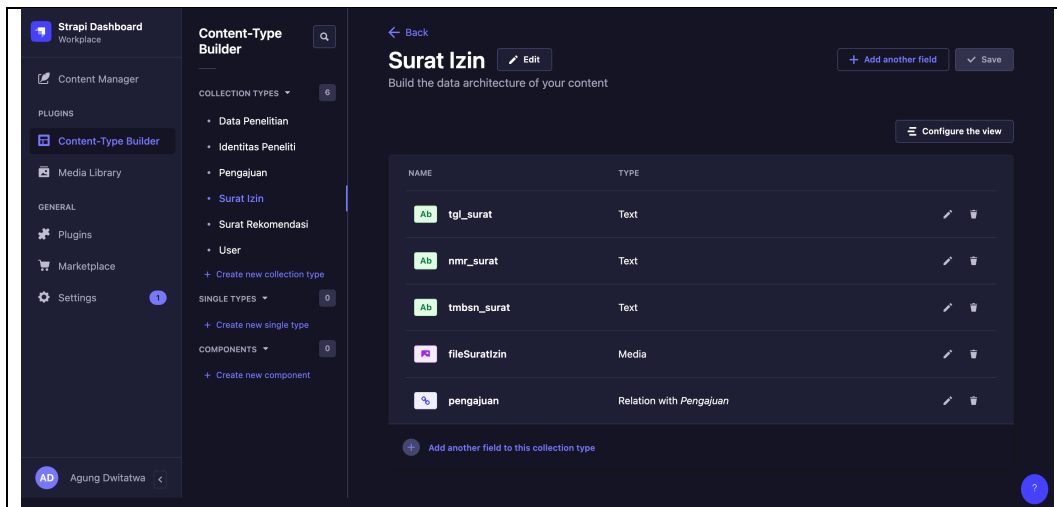
#### 4.4.1.3 Pembuatan Pengajuan Collection



Gambar 4. 25 Skema pengajuan *collection*.

Gambar 4. 25 merupakan skema pengajuan *collection* yang berfungsi sebagai skema penyimpanan data pengajuan pengguna di dalam *database*. Dalam pengajuan *collection*, terdapat 2 kolom isian berjenis text untuk menyimpan catatan dari proses pengajuan dan untuk menyimpan tanggal pengajuan. Selain itu, terdapat 2 kolom isian untuk menentukan status pengajuan dan tipe pengajuan. Lalu, terdapat 4 kolom isian berjenis relasi antara pengajuan *collection* dengan data surat rekomendasi, surat izin, *user*, dan data penelitian *collection*.

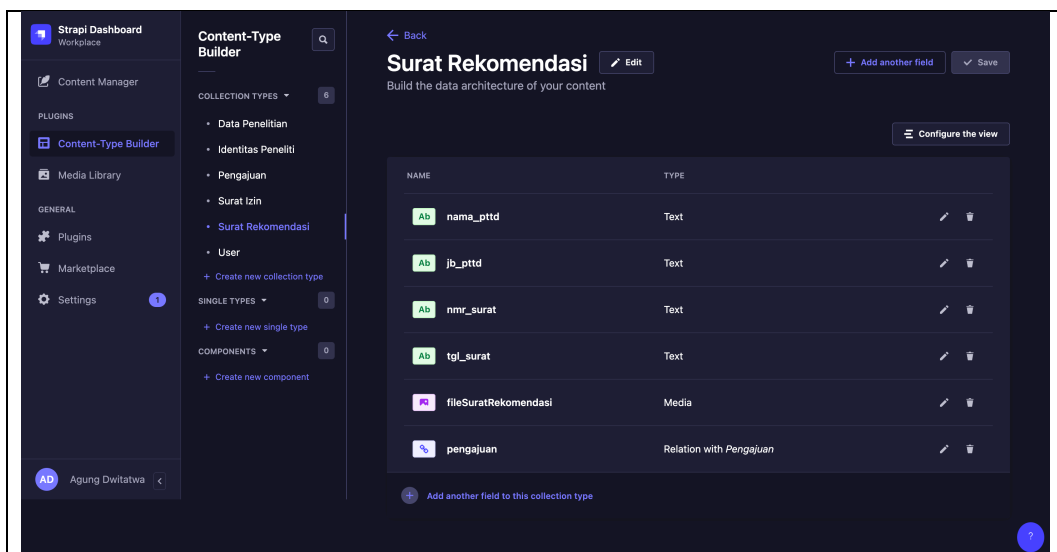
#### 4.4.1.4 Surat Izin Collection



Gambar 4. 26 Skema surat izin collection.

Gambar 4. 26 merupakan skema dari surat izin collection yang berfungsi sebagai skema penyimpanan data surat izin penelitian di dalam database. Dalam collection ini terdapat beberapa kolom isian yang terdiri dari 3 kolom isian berjenis text yaitu kolom isian untuk tanggal surat, nomor surat, dan tembusan surat. Lalu terdapat juga kolom isian berjenis file untuk berkas surat izin penelitian, dan satu kolom isian yang menentukan relasi antara surat izin collection dengan pengajuan collection.

#### 4.4.1.5 Surat Rekomendasi Collection

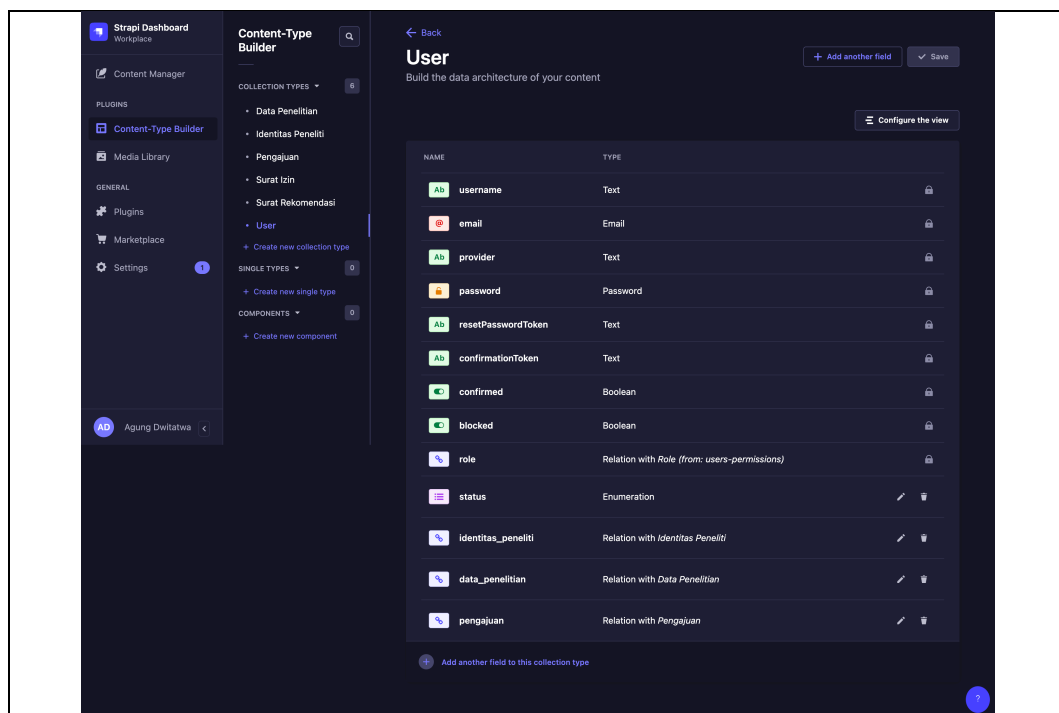


Gambar 4. 27 Skema surat rekomendasi collection.

Gambar 4. 27 merupakan skema dari surat rekomendasi collection yang berfungsi sebagai skema penyimpanan data surat rekomendasi penelitian di dalam

*database*. Dalam *collection* ini terdapat beberapa kolom isian yang terdiri dari 4 kolom isian berjenis text yaitu kolom isian untuk nama penandatanganan, jabatan penandatanganan, nomor surat dan tanggal surat. Lalu terdapat juga kolom isian berjenis *file* untuk berkas surat rekomendasi penelitian, dan satu kolom isian yang menentukan relasi antara surat rekomendasi *collection* dengan pengajuan *collection*.

#### 4.4.1.6 User Collection



Gambar 4. 28 Skema *user collection*.

Gambar 4. 28 merupakan skema dari *user collection* yang berfungsi sebagai skema penyimpanan data pengguna di dalam *database*. Dalam *collection* ini terdapat beberapa kolom isian *default* yang sudah disediakan oleh strapi untuk keperluan *Authentication* dan *Authorization*. Kolom isian *default* tersebut adalah kolom *username*, *email*, *provider*, *password*, *resetPasswordToken*, *confirmationToken*, *confirmed*, dan *role*. Lalu terdapat beberapa kolom isian tambahan yang dibuat untuk keperluan manipulasi data dalam sistem. Kolom isian yang ditambahkan adalah kolom isian *status* untuk menentukan apakah pengguna merupakan peneliti atau admin dari OPD. Lalu terdapat 3 kolom isian yang menentukan relasi antara *user collection* dengan *identitas peneliti*, *data penelitian*, dan *pengajuan collection*.

## 4.5 Testing

Untuk menentukan apakah sistem yang dibuat telah sesuai dengan kebutuhan yang diinginkan oleh pengguna, peneliti menerapkan beberapa metode pengujian antara lain :

### 4.5.1 Pengujian Mandiri

Sebelum melakukan pengujian oleh responden, peneliti terlebih dahulu melakukan pengujian secara mandiri untuk memastikan setiap fungsi pada sistem sudah berjalan dengan baik. Hal ini dilakukan untuk memastikan pada saat pengujian oleh responden nantinya tidak terjadi *crash* atau *error* yang menyebabkan proses pengujian menjadi terganggu.

Pada saat melakukan pengujian mandiri, peneliti menemukan beberapa permasalahan (*bugs*) yang menyebabkan peneliti harus mengubah kembali beberapa hal pada sistem. Perubahan tersebut antara lain :

#### 1. Mengubah komponen *file-input*

Sebelumnya, ketika setiap pengguna mengunggah *file*, komponen *file-input* akan langsung memproses dan mengirim *file* tersebut ke *backend* melalui Upload API. Namun, setelah *file* terunggah, ditemukan masalah di mana *file* yang sudah diunggah tidak terhubung dengan koleksi data yang seharusnya menjadi tempat *file* tersebut berelasi. Oleh karena itu, peneliti memutuskan untuk menggunakan metode yang berbeda, yaitu dengan memanfaatkan form data yang dikirim bersamaan dengan data lain melalui REST API yang sudah dibuat secara otomatis saat membuat *collection*.

#### 2. Membuat *utilities function*

Saat admin BRIDA NTB menyimpan data setelah membuat surat izin penelitian, admin hanya melihat tampilan loading saat proses penyimpanan atau pembaruan data surat izin. Namun, ketika terjadi kesalahan dalam proses pembuatan surat izin, tiba-tiba sistem mengalami *crash* dan admin tidak dapat melanjutkan pengisian data surat izin. Karena masalah ini, peneliti membuat fungsi utilitas yang memiliki peran khusus untuk melacak proses pembuatan surat izin. Fungsi utilitas ini memberikan informasi kepada pengguna apakah surat izin berhasil dibuat atau mengalami *error*.

### 3. Mengimplementasikan *Cookies*

Saat pengguna *login*, data *credentials* yang diperoleh dari pengguna akan disimpan dalam *global state* agar semua halaman dapat mengakses data pengguna dan menentukan apakah pengguna memiliki izin untuk mengakses halaman tersebut. Namun, saat melakukan pengujian mandiri, peneliti menemukan masalah di mana data *credentials* tidak dapat digunakan ketika menerapkan teknik *middleware routes*. Teknik *middleware routes* adalah salah satu metode yang dapat diterapkan dalam Next.js untuk memanipulasi permintaan (*request*) dan tanggapan (*response*) sebelum pengguna mengakses setiap halaman. Teknik ini digunakan untuk mencegah pengguna biasa mengakses halaman admin, dan sebaliknya. Namun, karena data *credentials* pengguna tidak dapat diakses, akhirnya peneliti mengubah metodenya dari yang awalnya menggunakan *global state* menjadi menggunakan *cookies*.

Setelah melaksanakan pengujian secara mandiri dan permasalahan yang ditemukan sudah diperbaiki, selanjutnya peneliti akan melakukan pengujian menggunakan responden yang merepresentasikan setiap level yang terdapat pada sistem.

#### 4.5.2 Blackbox Testing

Metode pengujian *blackbox* testing ini dipilih untuk melakukan pengujian dari setiap fitur yang terdapat pada sistem. Fitur – fitur yang akan diuji antara lain:

##### 4.5.2.1 Fitur Login

Fitur login merupakan fitur pertama yang akan diuji. Dimana pada fitur ini peneliti akan melakukan pengujian apakah fitur *login* sudah berjalan sesuai dengan yang diinginkan atau tidak.

Tabel 4. 1 Pengujian fitur login.

| Skenario pengujian   | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|--|--|-----------------|------------|
| Kolom isian <i>email</i> dan <i>password</i> diisi dengan data yang tidak sesuai | Proses login tidak berhasil dengan menampilkan pesan <i>error</i> dibagian bawah form login. | Sesuai          | Valid      |

|  |  |        |       |
|--|--|--------|-------|
| Kolom isian <i>email</i> dan <i>password</i> diisi dengan data yang sesuai   | Proses login berhasil dan pengguna akan diarahkan menuju laman <i>dashboard</i>                                  | Sesuai | Valid |
| Kolom isian <i>email</i> dan <i>password</i> diisi dengan data yang tidak sesuai dengan format yang sudah ditentukan | Form <i>login</i> menampilkan pesan <i>validation error</i> dibawah kolom isian <i>email</i> dan <i>password</i> | Sesuai | Valid |

#### 4.5.2.2 Fitur Register

Fitur *register* merupakan fitur yang berfungsi untuk mendaftarkan pengguna baru ke dalam sistem. Skenario pengujian untuk fitur ini dijelaskan secara lebih rinci pada Tabel 4. 2.

Tabel 4. 2 Pengujian fitur register

| Skenario pengujian   | Hasil yang diharapkan   | Hasil pengujian | Kesimpulan |
|--|---|-----------------|------------|
| Kolom isian <i>email</i> , <i>username</i> dan <i>password</i> diisi dengan data yang tidak sesuai                                       | Proses registrasi tidak berhasil dengan menampilkan pesan <i>error</i> dibagian bawah form <i>register</i> .                                      | Sesuai          | Valid      |
| Kolom isian <i>email</i> , <i>username</i> dan <i>password</i> diisi dengan data yang sesuai   | Proses registrasi berhasil dan pengguna akan diarahkan menuju laman <i>dashboard</i>  | Sesuai          | Valid      |
| Kolom isian <i>email</i> , <i>username</i> , dan <i>password</i> diisi dengan data yang tidak sesuai dengan format yang sudah ditentukan | Form menampilkan pesan <i>validation error</i> dibawah kolom isian <i>email</i> , <i>username</i> , dan <i>password</i> saat melakukan registrasi | Sesuai          | Valid      |



#### 4.5.2.3 Fitur Melihat Penelitian Terdaftar

Fitur melihat penelitian terdaftar ini berfungsi untuk melihat daftar penelitian yang sudah mendapatkan surat izin penelitian. Skenario pengujian untuk fitur ini dijelaskan secara lebih rinci pada Tabel 4. 3

Tabel 4. 3 Pengujian fitur melihat penelitian terdaftar

| Skenario pengujian   | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|--|--|-----------------|------------|
| Pengguna mengakses laman penelitian terdaftar tanpa melakukan <i>login</i> . | Sistem menampilkan daftar penelitian yang sudah mendapatkan surat izin penelitian. | Sesuai          | Valid      |
| Pengguna mengakses menu detail penelitian pada salah satu judul penelitian.  | Pengguna dapat melihat detail penelitian yang dipilih.                             | Sesuai          | Valid      |

#### 4.5.2.4 Fitur Melihat Petunjuk Pengisian

Fitur melihat petunjuk pengisian ini berfungsi untuk melihat petunjuk pengisian data pada sistem. Skenario pengujian untuk fitur ini dijelaskan lebih rinci pada Tabel 4. 4

Tabel 4. 4 Pengujian fitur melihat petunjuk pengisian

| Skenario pengujian   | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|--|--|-----------------|------------|
| Pengguna mengakses laman petunjuk pengisian dengan menekan menu pada laman dashboard | Sistem menampilkan melihat petunjuk pengisian untuk setiap isian data pada sistem            | Sesuai          | Valid      |
| Pengguna mengakses laman petunjuk pengisian tanpa melakukan login                    | Pengguna tidak dapat mengakses laman petunjuk pengisian dan diarahkan kembali ke laman login | Sesuai          | Valid      |

#### 4.5.2.5 Fitur Pengajuan Surat Rekomendasi Penelitian

Fitur pengajuan surat rekomendasi penelitian merupakan fitur untuk mendapatkan surat rekomendasi penelitian dari BAKESBANGPOLDAGRI. Skenario pengujian untuk fitur ini dijelaskan lebih rinci pada Tabel 4. 5

Tabel 4. 5 Pengujian fitur pengajuan surat rekomendasi penelitian

| Skenario pengujian  | Hasil yang diharapkan   | Hasil pengujian | Kesimpulan |
|---|---|-----------------|------------|
| Pengguna menekan tombol pengajuan pada laman pengajuan              | Sistem menampilkan status pengisian data identitas peneliti dan data penelitian                     | Sesuai          | Valid      |
| Pengguna tidak mengisi identitas peneliti dan data penelitian       | Pengguna tidak dapat melakukan pengajuan dan sistem menampilkan informasi mengenai syarat pengajuan | Sesuai          | Valid      |
| Pengguna mengakses laman pengajuan penelitian tanpa melakukan login | Pengguna tidak dapat mengakses laman pengajuan penelitian dan diarahkan kembali ke laman login      | Sesuai          | Valid      |

#### 4.5.2.6 Fitur Pengajuan Surat Izin Penelitian

Fitur pengajuan surat izin penelitian merupakan fitur untuk mendapatkan surat izin penelitian dari BRIDA NTB. Skenario pengujian untuk fitur ini dapat dilihat secara lebih rinci pada Tabel 4. 6

Tabel 4. 6 Pengujian fitur pengajuan surat izin penelitian

| Skenario pengujian  | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|---|--|-----------------|------------|
| Pengguna menekan tombol pengajuan pada laman pengajuan dan belum mengisi data surat rekomendasi | Sistem menampilkan form pengisian data surat rekomendasi penelitian dan tidak bisa mengirim pengajuan. | Sesuai          | Valid      |

|  |   |        |       |
|--|---|--------|-------|
| Pengguna tidak mengisi data surat rekomendasi penelitian dan tidak meng- <i>upload</i> file surat rekomendasi penelitian | Pengguna tidak dapat melakukan pengajuan surat izin penelitian dan <i>form</i> akan menampilkan pesan <i>validation error</i> | Sesuai | Valid |
| Pengguna mengakses laman pengajuan penelitian tanpa melakukan login  | Pengguna tidak dapat mengakses laman pengajuan penelitian dan diarahkan kembali ke laman login                                | Sesuai | Valid |

#### 4.5.2.7 Fitur Verifikasi Surat Rekomendasi Penelitian

Fitur verifikasi surat rekomendasi penelitian ini merupakan fitur verifikasi pengajuan surat rekomendasi penelitian dari peneliti yang dilakukan oleh admin BAKESBANGPOLDAGRI. Skenario pengujian untuk fitur ini dapat dilihat secara lebih rinci pada Tabel 4. 7

Tabel 4. 7 Pengujian fitur verifikasi surat rekomendasi penelitian

| Skenario pengujian   | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|--|--|-----------------|------------|
| Admin memilih menu daftar pengajuan                                  | Sistem menampilkan data pengajuan surat rekomendasi  | Sesuai          | Valid      |
| Admin menekan tombol mulai verifikasi pada salah satu data pengajuan | Sistem menampilkan data identitas peneliti dan data penelitian sesuai dengan data pengajuan        | Sesuai          | Valid      |
| Admin mengisi kolom isian catatan verifikasi                         | Catatan tersimpan kedalam <i>database</i> dan dapat dilihat oleh pengguna yang melakukan pengajuan | Sesuai          | Valid      |
| Admin memilih salah satu dari status verifikasi                      | Status pengajuan diperbarui sesuai dengan status yang dipilih                                      | Sesuai          | Valid      |

|  |  |        |       |
|--|--|--------|-------|
| Admin menekan tombol proses surat pada salah satu data pengajuan         | Data pengajuan yang dipilih dihapus dari daftar pengajuan dan dipindahkan ke menu penerbitan surat | Sesuai | Valid |
| Admin memilih menu penerbitan surat                                      | Sistem menampilkan data pengajuan yang sudah terverifikasi   | Sesuai | Valid |
| Admin menekan tombol selesaikan pengajuan pada salah satu data pengajuan | Status pengajuan surat rekomendasi berubah menjadi selesai.  | Sesuai | Valid |

#### 4.5.2.8 Fitur Verifikasi Surat Izin Penelitian

Fitur verifikasi surat izin penelitian ini merupakan fitur verifikasi pengajuan surat izin penelitian dari peneliti yang dilakukan oleh admin BRIDA NTB. Skenario pengujian untuk fitur ini dapat dilihat secara lebih rinci pada Tabel 4. 8

Tabel 4. 8 Pengujian verifikasi surat izin penelitian

| Skenario pengujian   | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|--|--|-----------------|------------|
| Admin memilih menu daftar pengajuan                                  | Sistem menampilkan data pengajuan surat rekomendasi  | Sesuai          | Valid      |
| Admin menekan tombol mulai verifikasi pada salah satu data pengajuan | Sistem menampilkan data identitas peneliti dan data penelitian sesuai dengan data pengajuan        | Sesuai          | Valid      |
| Admin mengisi kolom isian catatan verifikasi                         | Catatan tersimpan kedalam <i>database</i> dan dapat dilihat oleh pengguna yang melakukan pengajuan | Sesuai          | Valid      |
| Admin memilih salah satu dari status verifikasi                      | Status pengajuan diperbarui sesuai dengan status yang dipilih                                      | Sesuai          | Valid      |

|  |   |        |       |
|--|---|--------|-------|
| Admin menekan tombol proses surat izin pada salah satu data pengajuan    | Sistem menampilkan laman untuk membuat data surat izin penelitian | Sesuai | Valid |
| Admin memilih menu penerbitan surat                                      | Sistem menampilkan data pengajuan yang sudah terverifikasi        | Sesuai | Valid |
| Admin menekan tombol selesaikan pengajuan pada salah satu data pengajuan | Status pengajuan surat rekomendasi berubah menjadi selesai.       | Sesuai | Valid |

#### 4.5.2.9 Fitur Cetak Surat Izin Penelitian

Fitur cetak surat izin penelitian merupakan fitur yang digunakan oleh admin BRIDA NTB untuk mencetak surat izin. Skenario pengujian pada fitur ini dapat dilihat pada Tabel 4. 9 dibawah ini

Tabel 4. 9 Pengujian fitur cetak surat izin penelitian

| Skenario pengujian   | Hasil yang diharapkan  | Hasil pengujian | Kesimpulan |
|--|--|-----------------|------------|
| Admin memilih menu pengajuan terverifikasi                             | Sistem menampilkan data pengajuan yang sudah terverifikasi                               | Sesuai          | Valid      |
| Admin menekan tombol proses surat izin pada salah satu data penelitian | Sistem menampilkan laman cetak surat izin penelitian                                     | Sesuai          | Valid      |
| Admin menekan tombol cetak surat                                       | Sistem menampilkan <i>print out window</i> untuk menyimpan surat menjadi <i>file pdf</i> | Sesuai          | Valid      |

### 4.5.3 User Acceptance Testing (UAT)

Berbeda dengan *black box testing*, *user acceptance testing* berfokus pada pengujian langsung yang dilakukan oleh pengguna nantinya. Pengujian ini bertujuan untuk memastikan apakah sistem dapat berfungsi dengan baik berdasarkan perspektif dari pengguna. Pengujian ini melibatkan mahasiswa, masyarakat umum, dan admin dari masing – masing badan. Masing – masing responden nantinya akan diminta untuk mengisi kuisisioner yang berisi beberapa pernyataan mengenai sistem yang dibuat. Responden dari mahasiswa dan masyarakat umum akan diberikan beberapa pernyataan antara lain :

1. Tampilan sistem mudah dipahami dan mudah digunakan pengguna.
2. Fitur pengajuan surat rekomendasi berjalan dengan baik.
3. Fitur pengajuan surat izin penelitian berjalan dengan baik.
4. Fitur *tracking* pengajuan berjalan dengan baik.
5. Berkas pendukung dapat diunggah ke dalam sistem dengan baik.
6. Fitur pendaftaran berjalan dengan baik.
7. Menu penelitian terdaftar dapat diakses dengan baik.

Sedangkan responden dari admin OPD akan diberikan pernyataan antara lain :

1. Tampilan sistem untuk admin mudah dipahami dan mudah digunakan.
2. Data pengajuan ditampilkan dengan baik.
3. Detail dari data pengajuan berisi informasi yang cukup untuk proses verifikasi.
4. Proses verifikasi pengajuan dapat dilakukan dengan mudah.
5. Fitur pemberian catatan saat proses verifikasi dapat berjalan dengan baik.

Berdasarkan pernyataan - pernyataan tersebut, responden diminta untuk memberikan nilai *acceptance* yang terdiri dari 5 tingkat penilaian sesuai dengan yang ditampilkan pada Tabel 4. 10. Hasil penilaian responden tersebut nantinya diklasifikasikan sesuai dengan aturan klasifikasi hasil pengujian pada Tabel 4. 11

Tabel 4. 10 Tingkat penilaian responden.

| Pilihan Penilaian      | Bobot Penilaian |
|------------------------|-----------------|
| A. Sangat Setuju       | 5               |
| B. Setuju              | 4               |
| C. Cukup               | 3               |
| D. Tidak Setuju        | 2               |
| E. Sangat Tidak Setuju | 1               |

Tabel 4. 11 Klasifikasi hasil pengujian

| Klasifikasi Hasil Pengujian | Presentase Hasil Pengujian |
|-----------------------------|----------------------------|
| Sangat Baik                 | Lebih dari 90%             |
| Baik                        | Antara 80% s/d 90%         |
| Cukup Baik                  | Antara 70% - 80%           |
| Kurang Baik                 | Antara 60% - 70%           |
| Tidak Baik                  | Kurang dari 60%            |

#### 4.5.3.1 Hasil Penilaian Responden

Hasil penilaian dari mahasiswa dan masyarakat umum serta hasil penilaian dari admin badan dapat dilihat pada tabel dibawah ini.

Tabel 4. 12 Hasil penilaian responden dari mahasiswa dan masyarakat umum

| Pernyataan   | Jumlah Jawaban |    |   |   |   | Presentase Jawaban |        |        |    |    |
|--------------|----------------|----|---|---|---|--------------------|--------|--------|----|----|
|              | A              | B  | C | D | E | A                  | B      | C      | D  | E  |
| Pernyataan 1 | 2              | 11 | 1 | 0 | 0 | 14.29%             | 78.57% | 7.14%  | 0% | 0% |
| Pernyataan 2 | 2              | 10 | 2 | 0 | 0 | 14.29%             | 71.43% | 14.29% | 0% | 0% |
| Pernyataan 3 | 3              | 10 | 1 | 0 | 0 | 21.43%             | 71.43% | 7.14%  | 0% | 0% |
| Pernyataan 4 | 6              | 6  | 2 | 0 | 0 | 42.86%             | 42.86% | 14.29% | 0% | 0% |
| Pernyataan 5 | 5              | 9  | 0 | 0 | 0 | 35.71%             | 64.29% | 0%     | 0% | 0% |
| Pernyataan 6 | 5              | 8  | 1 | 0 | 0 | 35.71%             | 57.14% | 7.14%  | 0% | 0% |
| Pernyataan 7 | 6              | 8  | 0 | 0 | 0 | 42.86%             | 57.14% | 0.00%  | 0% | 0% |

Tabel 4. 13 Hasil penilaian responden dari admin badan.

| Pernyataan   | Jumlah Jawaban |   |   |   |   | Presentase Jawaban |        |        |    |    |
|--------------|----------------|---|---|---|---|--------------------|--------|--------|----|----|
|              | A              | B | C | D | E | A                  | B      | C      | D  | E  |
| Pernyataan 1 | 2              | 3 | 1 | 0 | 0 | 33.33%             | 50.00% | 16.67% | 0% | 0% |
| Pernyataan 2 | 3              | 3 | 0 | 0 | 0 | 50.00%             | 50.00% | 0.00%  | 0% | 0% |
| Pernyataan 3 | 2              | 2 | 2 | 0 | 0 | 33.33%             | 33.33% | 33.33% | 0% | 0% |
| Pernyataan 4 | 3              | 3 | 0 | 0 | 0 | 50.00%             | 50.00% | 0%     | 0% | 0% |
| Pernyataan 5 | 2              | 4 | 0 | 0 | 0 | 33.33%             | 66.67% | 0%     | 0% | 0% |

Hasil penilaian yang diperoleh tersebut kemudian diolah untuk mendapatkan nilai rata – rata dari setiap pernyataan dengan cara mengalikan setiap poin pernyataan dengan bobot yang sudah ditentukan dan dibagi dengan jumlah responden. Hasil perhitungan untuk masing – masing responden dapat dilihat pada tabel dibawah ini.

Tabel 4. 14 Bobot penilaian dari pernyataan mahasiswa dan masyarakat

| Pernyataan   | Bobot Nilai |       |       |       |       | Jumlah | Rata - Rata |
|--------------|-------------|-------|-------|-------|-------|--------|-------------|
|              | A x 5       | B x 4 | C x 3 | D x 2 | E x 1 |        |             |
| Pernyataan 1 | 10          | 44    | 3     | 0     | 0     | 57     | 4.07        |
| Pernyataan 2 | 10          | 40    | 6     | 0     | 0     | 56     | 4.00        |
| Pernyataan 3 | 15          | 40    | 3     | 0     | 0     | 58     | 4.14        |
| Pernyataan 4 | 30          | 24    | 6     | 0     | 0     | 60     | 4.29        |
| Pernyataan 5 | 25          | 36    | 0     | 0     | 0     | 61     | 4.36        |
| Pernyataan 6 | 25          | 32    | 3     | 0     | 0     | 60     | 4.29        |
| Pernyataan 7 | 30          | 32    | 0     | 0     | 0     | 62     | 4.43        |

Tabel 4. 15 Bobot penilaian dari pernyataan admin

| Pernyataan   | Bobot Nilai |       |       |       |       | Jumlah | Rata - Rata |
|--------------|-------------|-------|-------|-------|-------|--------|-------------|
|              | A x 5       | B x 4 | C x 3 | D x 2 | E x 1 |        |             |
| Pernyataan 1 | 10          | 12    | 3     | 0     | 0     | 25     | 4.17        |
| Pernyataan 2 | 15          | 12    | 0     | 0     | 0     | 27     | 4.50        |
| Pernyataan 3 | 10          | 8     | 6     | 0     | 0     | 24     | 4.00        |
| Pernyataan 4 | 15          | 12    | 0     | 0     | 0     | 27     | 4.50        |
| Pernyataan 5 | 10          | 16    | 0     | 0     | 0     | 26     | 4.33        |

Setelah mendapatkan nilai rata – rata, selanjutnya dilakukan perhitungan persentase untuk mendapatkan nilai kelayakan sistem menggunakan rumus :

$$\text{persentase} = \frac{\text{Nilai rata-rata}}{\text{bobot maximum}} \times 100\%$$

Hasil dari perhitungan dari masing masing responden dapat dilihat pada tabel dibawah ini.

Tabel 4. 16 Hasil perhitungan dari responden mahasiswa dan masyarakat umum.

| Pernyataan   | Perhitungan  | Persentase |
|--------------|--------------|------------|
| Pernyataan 1 | 4.07/5 x 100 | 81.4 %     |
| Pernyataan 2 | 4.00/5 x 100 | 80 %       |
| Pernyataan 3 | 4.14/5 x 100 | 82.8 %     |
| Pernyataan 4 | 4.29/5 x 100 | 85.8 %     |
| Pernyataan 5 | 4.36/5 x 100 | 87.2 %     |
| Pernyataan 6 | 4.29/5 x 100 | 85.8 %     |
| Pernyataan 7 | 4.43/5 x 100 | 88.6 %     |
| Rata – Rata  |              | 84.51 %    |



Tabel 4. 17 Hasil perhitungan dari responden admin badan.

| Penyataan    | Perhitungan         | Persentase |
|--------------|---------------------|------------|
| Pernyataan 1 | $4.17/5 \times 100$ | 83.4 %     |
| Pernyataan 2 | $4.50/5 \times 100$ | 90 %       |
| Pernyataan 3 | $4.00/5 \times 100$ | 80 %       |
| Pernyataan 4 | $4.50/5 \times 100$ | 90 %       |
| Pernyataan 5 | $4.33/5 \times 100$ | 86.6 %     |
| Rata – Rata  |                     | 86 %       |

#### 4.5.3.2 Kesimpulan Pengujian

Dalam Tabel 4.16, penilaian sistem dari perspektif pengguna yang mengajukan izin penelitian, yang diwakili oleh responden dari kalangan mahasiswa dan masyarakat umum, menunjukkan hasil sebesar 84,51%. Hasil tersebut mengindikasikan bahwa penggunaan sistem dari sudut pandang pengguna masyarakat umum dan mahasiswa dapat digolongkan ke dalam kategori "Baik".

Sementara itu, dalam Tabel 4.17, penilaian sistem dari perspektif admin, yang dilakukan oleh admin yang berada di BRIDA NTB, menghasilkan skor sebesar 86%. Hasil tersebut menunjukkan bahwa penggunaan sistem dari sudut pandang admin dapat juga digolongkan ke dalam kategori "Baik".

## 4.6 Maintenance

Tahap pemeliharaan sistem merupakan langkah penting dalam siklus pengembangan perangkat lunak yang menggunakan metode *fountain*. Dalam jangka waktu 30 hari setelah penyelesaian sistem, peneliti berfokus pada aktivitas pemeliharaan yang mencakup perbaikan bug, peningkatan fungsionalitas, pembaruan keamanan, dan evaluasi kinerja.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari uraian dan pembahasan yang telah disajikan, kita dapat merumuskan beberapa simpulan sebagai berikut :

1. Penerapan arsitektur JAMStack dengan menggunakan metode *fountain* sudah berjalan dengan baik sesuai dengan keinginan pengguna. Fitur – fitur tambahan yang terdapat pada sistem seperti *tracking* proses pengajuan mempermudah pengguna dari kalangan mahasiswa maupun masyarakat umum untuk melihat status pengajuan yang sudah dilakukan. Lalu fitur cetak surat dan verifikasi dari sisi admin juga mempermudah admin dari BAKESBANGPOLDAGRI maupun admin dari BRIDA NTB dalam melayani pengajuan surat izin penelitian, sehingga admin dapat memberikan pelayanan terbaik kepada masyarakat.
2. Fitur-fitur yang terdapat dalam sistem berjalan sesuai dengan harapan pengguna. Hal ini terlihat dari hasil pengujian dengan menggunakan metode *Blackbox*, yang menunjukkan bahwa setiap fitur dalam sistem beroperasi dengan baik. Selain itu, hasil pengujian menggunakan metode *User Acceptance Testing* (UAT) juga mencerminkan hasil yang sesuai, baik dari perspektif pengguna (mahasiswa & masyarakat umum) maupun dari perspektif admin. Hasil yang diperoleh dari pengguna (mahasiswa & masyarakat umum) adalah sebesar 84,51%, sementara hasil dari admin adalah sebesar 86%.

#### 5.2 Saran

Adapun saran yang diberikan untuk mengembangkan sistem yang telah dibangun agar berjalan dengan baik dan dapat dikembangkan menjadi lebih baik adalah :

1. Dilakukan *update* secara berkala untuk *package* dan *dependency* pada NextJS agar sistem dapat diperbarui sesuai dengan fitur terbaru dari NextJS.
2. Mengimplementasikan *caching mechanism* pada strapi untuk mempercepat komputasi dari sisi *backend*.

3. Mengimplementasikan *email validation* pada saat pengguna melakukan registrasi, hal ini untuk menghindari pengguna menggunakan email yang tidak valid.
4. Melakukan validasi saat pengguna menginput tanggal mulai penelitian dan tanggal berakhir penelitian pada laman identitas peneliti. Hal ini diperlukan agar pengguna tidak mengisi data tanggal mulai maupun tanggal berakhir dengan data yang tidak valid.

## DAFTAR PUSTAKA

- [1] N. Prisma Yunita and R. D. Aprianto, “KONDISI TERKINI PERKEMBANGAN PELAKSANAAN E-GOVERNMENT DI INDONESIA : ANALISIS WEBSITE,” 2018. [Online]. Available: [www.pandi.id//statistik](http://www.pandi.id//statistik)
- [2] M. I. F. Assegaf, H. Juliani, and N. Sa’adah, “PELAKSANAAN ONLINE SINGLE SUBMISSION (OSS) DALAM RANGKA PERCEPATAN PERIZINAN BERUSAHA DI DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU (DPMPTSP) JAWA TENGAH,” *Diponegoro Law Journal*, vol. 8, no. 2, pp. 1328–1342, Apr. 2019, Accessed: May 25, 2022. [Online]. Available: <https://ejournal3.undip.ac.id/index.php/dlr/article/view/24582>
- [3] M. H. Bisri and B. T. Asmoro, “Etika Pelayanan Publik di Indonesia,” *Journal of Governance Innovation*, vol. 1, no. 1, pp. 59–76, Apr. 2019, doi: 10.36636/jogiv.v1i1.298.
- [4] B. Pengembangan SDM Penelitian Komunikasi dan Informatika Surabaya Jl Raya, “IMPLEMENTASI E-GOVERNMENT PUBLIC RELATIONS SEBAGAI PENINGKATAN PELAYANAN INFORMASI PUBLIK, DILINGKUNGAN PEMERINTAH DAERAH DI JAWA TIMUR DAN NUSA TENGGARA BARAT E-GOVERNMENT PUBLIC RELATIONS IMPLEMENTATION AS IMPROVING PUBLIC INFORMATION SERVICES, IN THE REGIONAL GOVERNMENT ENVIRONMENT IN EAST JAWA AND NUSA TENGGARA BARAT Trisnani,” 2018.
- [5] G. M. Zamroni *et al.*, “JIKA | 184 Implementasi Framework MVC Pada Pemodelan dan Pengembangan Sistem Informasi Masjid Berbasis Web,” 2021.
- [6] D. Markovic, M. Scekcic, A. Bucaioni, and A. Cicchetti, “Could jamstack be the future of web applications architecture?,” in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, New York, NY, USA: ACM, Apr. 2022, pp. 1872–1881. doi: 10.1145/3477314.3506991.
- [7] M. Anike, I. Fattu, J. Samratulangi, O. Barat, K. Lima, and N. Tenggara Timur, “PENGEMBANGAN SISTEM INFORMASI PELAYANAN PUBLIK

- PERIZINAN PEREDARAN HASIL HUTAN,” *Sistem Informasi* |, vol. 8, no. 2, pp. 150–157, 2021.
- [8] M. V. al Hasri and E. Sudarmilah, “Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Website Kelurahan Banaran,” *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 2, pp. 249–260, May 2021, doi: 10.30812/matrik.v20i2.1056.
- [9] A. K. Soemarsono, R. Ho Purabaya, I. Wayan, and W. Pradnyana, *Perancangan Sistem Informasi Desa Berbasis Web untuk Pelayanan Publik pada Desa Tengkurak, Kecamatan Tirtayasa, Kabupaten Serang*. 2021.
- [10] R. A. E. Prayudha and D. R. Rochmawati, “PERANCANGAN SISTEM INFORMASI PELAYANAN PUBLIK MENGGUNAKAN FRAMEWORK LARAVEL & MYSQL DI KECAMATAN COBLONG KOTA BANDUNG,” Kota Bandung, Dec. 2018.
- [11] E. Triana, J. Raya, T. No, K. Gedong, P. Rebo, and J. Timur, “ANALISIS PERANCANGAN SISTEM ADMINISTRASI PELAYANAN PUBLIK KANTOR DESA CISAAT MENGGUNAKAN JAVA NETBEANS,” *Jurnal Riset dan Aplikasi Mahasiswa Informatika*), vol. 02, 2021.
- [12] A. Nurkholis, E. R. Susanto, and S. Wijaya, “Penerapan Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Pelayanan Publik,” 2021.
- [13] D. Markovic, A. Cicchetti, A. Bucaioni, and S. Jamstack, “UNDERSTANDING JAMSTACK AND ITS PERCEPTION IN WEB DEVELOPMENT.”
- [14] M. Beke, “On the Comparison of Software Quality Attributes for Client-side and Server-side Rendering,” Belgium, Jun. 2018.
- [15] A. A. YUSUF, “ANALISIS STATIC SITE GENERATOR PADA WEB RESPONSIF,” 2021.
- [16] N. An, “BUILDING AN E-COMMERCE WEBSITE USING NEXT JS, MANTINE, AND STRAPI,” *Information Technology, CENTRIA UNIVERSITY OF APPLIED SCIENCES*, pp. 14–15, May 2022.
- [17] A. Mujib, M. Yastuti Madrah, A. Riansyah, and N. Muhammad Syaifuddin, “SISTEMASI: Jurnal Sistem Informasi Implementasi RESTful Web Service pada Sistem Informasi Donor ASI Terintegrasi di Indonesia Implementation of RESTful Web Service on Indonesian’s Integrated Breastfeeding Donor Information System.” [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>

- [18] M. DirgaF, “APLIKASI E-LEARNING SISWA SMK BERBASIS WEB,” 2021. [Online]. Available: <https://jurnal.umpar.ac.id/index.php/sylog>
- [19] J. Shadiq, A. Safei, R. Wahyudin Ratu Loly, C. sitasi, L. Rwr, and P. Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing, “INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing,” *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 97–110, 2021.
- [20] F. Kurnia, S. Kom, and N. Putri, “Sistem Informasi Cuti Tahunan Pegawai Dinas Sosial Provinsi Riau,” *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, vol. 6, no. 2, pp. 186–191, 2020.