

**PENERAPAN *SINGLE PAGE APPLICATION* DALAM  
PENGEMBANGAN ANTARMUKA APLIKASI GALANG DANA  
BERBASIS WEB MENGGUNAKAN KERANGKA KERJA  
LARAVEL BREEZE**

**SKRIPSI**

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi  
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)  
dalam Teknologi Informasi



Diajukan Oleh:

**ALIM MUGHANIL KARIM**

NIM: 1908096035

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG  
2022**

## PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama : Alim Mughanil Karim

NIM : 1908096035

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

**Penerapan *Single Page Application* Dalam Pengembangan  
Antarmuka Aplikasi Galang Dana Berbasis Web Menggunakan  
Kerangka Kerja Laravel Breeze**

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 22 Desember 2022

Pembuat Pernyataan,



Alim Mughanil Karim

NIM: 1908096035



KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI WALISONGO  
FAKULTAS SAINS DAN TEKNOLOGI  
Jl. Prof. Dr. Hamka Ngaliyan Semarang  
Telp.024-7601295 Fax.7615387

### PENGESAHAN

Naskah skripsi berikut ini:

Judul : Penerapan *Single Page Application* Dalam Pengembangan Antarmuka Aplikasi Galang Dana Berbasis Web Menggunakan Kerangka Kerja Laravel Breeze

Penulis : **Alim Mughanil Karim**




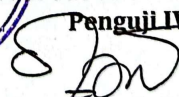

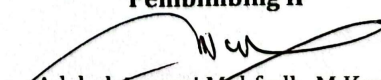
NIM : 1908096035

Jurusan : Teknologi Informasi

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Teknologi Informasi.

Semarang, 29 Desember 2022

**DEWAN PENGUJI**

<p><b>Penguji I</b></p>  <p>Nur Cahyo H.W., S.T., M.Kom. NIP. 197312222006041001</p>	<p><b>Penguji II</b></p>  <p>Dr. Masy Ari Ulinuha, M.T. NIP. 198108122011011007</p>
<p><b>Penguji III</b></p>  <p>Khotibul Umam, M.Kom. NIP. 197908272011011007</p>	<p><b>Penguji IV</b></p>  <p>Hery Mustofa, M.Kom. NIP. 198703172019031007</p>
<p><b>Pembimbing I</b></p>  <p>Dr. Masy Ari Ulinuha, M.T. NIP. 198108122011011007</p>	<p><b>Pembimbing II</b></p>  <p>Adzhal Arwani Mahfudh, M.Kom. NIP. 199107032019031006</p>

## NOTA PEMBIMBING I

Semarang, 22 Desember 2022

Yth. Ketua Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
UIN Walisongo Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:  
Judul : Penerapan *Single Page Application* Dalam Pengembangan Antarmuka Aplikasi Galang Dana Berbasis Web Menggunakan Kerangka Kerja Laravel Breeze

Nama : Alim Mughanil Karim

NIM : 1908096035

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqsyah.

Wassalamu'alaikum. wr. wb.

Pembimbing I,



Dr. Masy Ari Ulinuha, M.T.  
NIP. 198108122011011007

## NOTA PEMBIMBING II

Semarang, 22 Desember 2022

Yth. Ketua Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
UIN Walisongo Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : Penerapan *Single Page Application* Dalam Pengembangan Antarmuka Aplikasi Galang Dana Berbasis Web Menggunakan Kerangka Kerja Laravel Breeze

Nama : Alim Mughanil Karim

NIM : 1908096035

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqsyah.

Wassalamu'alaikum. wr. wb.

Pembimbing II,



Adzhal Arwani Mahfudh, M.Kom.  
NIP. 199107032019031006

## ABSTRAK

Pesatnya perkembangan teknologi memberikan sebuah gagasan terkait pemanfaatannya dalam proses penggalangan dana. Gagasan tersebut dikemukakan oleh Yayasan Amal Bakti Nusantara Tegal yang ingin membuat sebuah *platform digital* untuk penggalangan dana dalam bentuk aplikasi web dengan merek dagang Berbagi Mulia. Aplikasi web tersebut nantinya diharapkan dapat menjembatani berbagai organisasi sosial non-profit dengan mengkampanyekan program yang digagasnya untuk mendapatkan pendanaan dari para donatur.

Kebutuhan terkait performa yang baik dalam penyajian data dengan antarmuka web yang modern menjadi sebuah keharusan. Untuk itu, penggunaan *single page application* sebagai metode penyajian halaman web menjadi salah satu solusinya. Dalam pengembangannya, teknologi yang dipakai adalah Laravel Breeze dikarenakan Laravel sebagai *server-side* dan ReactJS sebagai *client-side* dapat berkomunikasi secara langsung melalui perantara InertiaJS, sehingga tidak membutuhkan REST API. Sedangkan, metode pengembangan sistem yang dipakai adalah metode *prototype* dengan beberapa tahapan, yaitu *communication, quick plan, modeling quick design, dan deployment delivery and feedback*.

Pengujian performa menggunakan PageSpeed Insight dan GT Metrix dengan hasil pada PageSpeed Insight mendapatkan nilai rata-rata 94,8 di perangkat *desktop* dan 73,8 di perangkat *mobile*. Sedangkan, GT Metrix didapatkan nilai rata-rata 75,8%. Penerapan *single page application* juga menyelesaikan ketidakefektifan *multi page application* dalam menyajikan halaman web. Hal tersebut diketahui melalui proses analisis kedua metode tersebut dengan membandingkannya saat proses transisi halaman web.

**Kata kunci:** *Single Page Application*, Laravel Breeze, Aplikasi Web, Penggalangan Dana

## KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya atas selesainya pembuatan laporan skripsi dengan judul “Penerapan *Single Page Application* Dalam Pengembangan Antarmuka Aplikasi Galang Dana Berbasis Web Menggunakan Kerangka Kerja Laravel Breeze”. Laporan ini disusun sebagai salah satu syarat menyelesaikan program pendidikan Strata-1 Program Studi Teknologi Informasi di Universitas Islam Negeri Walisongo Semarang.

Penyusunan laporan ini tidak terlepas dari bantuan beberapa pihak, oleh karena itu penulis hendak mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Imam Taufiq, M.Ag selaku Rektor UIN Walisongo Semarang,
2. Bapak Dr. H. Ismail, M.Ag selaku Dekan Fakultas Sains dan Teknologi,
3. Bapak Nur Cahyo Hendro Wibowo S.T., M.Kom. selaku Ketua Program Studi Teknologi Informasi,
4. Bapak Dr. Masy Ari Ulinuha M.T. selaku Sekretaris Program Studi Teknologi Informasi dan sekaligus sebagai Dosen Pembimbing I yang telah memberi arahan terkait penyelesaian laporan skripsi ini,

5. Bapak Adzhal Arwani Mahfudh M.Kom. selaku Dosen Pembimbing II yang telah memberi arahan terkait penyelesaian laporan skripsi ini,
6. Kepada Bapak dan Ibu Dosen Program Studi Teknologi Informasi yang telah membimbing penulis selama mengikuti perkuliahan,
7. Kepada pihak Yayasan Amal Bakti Nusantara Tegal yang telah memberikan kesempatan kepada penulis untuk berkontribusi dalam pembuatan *platform digital* penggalangan dana hingga terselesaikannya laporan skripsi ini.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang tidak dapat disebutkan satu persatu dan terlibat dalam penyusunan laporan skripsi sehingga dapat diselesaikan dengan baik. Semoga penyusunan laporan skripsi ini dapat memberikan manfaat kepada semua pihak, terutama di bidang pendidikan.

Semarang, 22 Desember 2022

Penulis,



Alim Mughanil Karim  
NIM: 1908096035



## DAFTAR ISI

PERNYATAAN KEASLIAN .....	<b>Error! Bookmark not defined.</b>
PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
NOTA PEMBIMBING I .....	ii
NOTA PEMBIMBING II .....	iii
ABSTRAK .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR TABEL .....	x
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xvi
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
A. Latar Belakang Masalah .....	1
B. Rumusan Masalah .....	7
C. Batasan Masalah .....	7
D. Tujuan Penelitian .....	8
E. Manfaat Penelitian .....	8
<b>BAB II LANDASAN PUSTAKA .....</b>	<b>10</b>
A. Kajian Pustaka .....	10
1. <i>Single Page Application</i> .....	10
2. Laravel Breeze .....	11
3. ReactJS .....	12
4. Tailwind CSS dan DaisyUI .....	14
5. Xampp dan MySQL .....	15
6. <i>Core Web Vital</i> .....	16
B. Kajian Penelitian yang Relevan .....	18
<b>BAB III METODE PENELITIAN .....</b>	<b>22</b>
A. Metode Pengumpulan Data .....	22
1. Studi Lapangan .....	22
2. Studi Pustaka .....	23
B. Metode Pengembangan Sistem .....	23
1. <i>Communication</i> .....	24
2. <i>Quick Plan</i> .....	24
3. <i>Modeling Quick Design</i> .....	25
4. <i>Construction of Prototype</i> .....	28

5.	<i>Deployment Delivery and Feedback</i> .....	30
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN</b> .....		<b>32</b>
A.	<i>Communication</i> .....	32
B.	<i>Quick Plan</i> .....	33
C.	<i>Modelling Quick Design</i> .....	35
1.	<i>Unified Modelling Language (UML)</i> .....	36
2.	<i>Entity Relationship Diagram (ERD)</i> .....	49
3.	<i>Wireframe</i> .....	51
D.	<i>Construction of Prototype</i> .....	56
1.	Proses Pengkodean.....	56
2.	Hasil Tampilan Aplikasi Web.....	64
E.	<i>Deployment Delivery and Feedback</i> .....	77
1.	Proses <i>Deployment</i> .....	77
2.	Hasil Analisis <i>Feedback</i> .....	79
3.	Hasil Pengujian <i>Black Box Testing</i> .....	82
4.	Hasil Pengujian <i>User Acceptance Test</i> .....	95
5.	Hasil Analisis Performa.....	101
F.	Analisis Penerapan <i>Single Page Application</i> .....	106
1.	Analisis Struktur Kode.....	107
2.	Analisis Hasil Komparasi.....	108
<b>BAB V KESIMPULAN DAN SARAN</b> .....		<b>117</b>
A.	Kesimpulan .....	117
B.	Saran .....	119
DAFTAR PUSTAKA.....		120

## DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian terdahulu .....	18
Tabel 4.1 Hasil analisis <i>feedback</i> dan perbaikannya .....	79
Tabel 4.2 Hasil pengujian pada fitur pembatasan halaman berdasarkan <i>role</i> pengguna.....	82
Tabel 4.3 Hasil pengujian pada fitur registrasi akun <i>User</i> dan akun Admin Lembaga.....	84
Tabel 4.4 Hasil pengujian pada fitur pengelolaan program galang dana.....	86
Tabel 4.5 Hasil pengujian pada fitur pengelolaan laporan berita .....	87
Tabel 4.6 Hasil pengujian pada fitur donasi baru.....	88
Tabel 4.7 Hasil pengujian pada fitur <i>payment gateway</i> verifikasi manual.....	88
Tabel 4.8 Hasil pengujian pada fitur <i>payment gateway</i> verifikasi otomatis .....	89
Tabel 4.9 Hasil pengujian pada fitur promosi program galang dana.....	92
Tabel 4.10 Hasil pengujian pada fitur penarikan dana .....	93
Tabel 4.11 Hasil pengujian pada fitur integrasi dengan Facebook Ads melalui kode Pixel .....	94
Tabel 4.12 Hasil pengujian <i>user acceptance test</i> (UAT) oleh <i>Super Admin</i> .....	96
Tabel 4.13 Hasil pengujian <i>user acceptance test</i> (UAT) oleh Admin Lembaga.....	97
Tabel 4.14 Lambang fitur dalam proses analisis UAT pada <i>role User</i> .....	99
Tabel 4.15 Pembobotan klasifikasi kualitas pada pengujian <i>user acceptance test</i> .....	99
Tabel 4.16 Hasil pengujian <i>user acceptance test</i> (UAT) oleh <i>User</i> sebagai donatur .....	100
Tabel 4.17 Hasil perhitungan performa web menggunakan Page Speed Insight pada perangkat <i>dekstop</i> .....	102

Tabel 4.18 Hasil perhitungan performa web menggunakan Page Speed Insight pada perangkat <i>mobile</i> .....	103
Tabel 4.19 Hasil perhitungan performa web menggunakan GT Metrix.....	104

## DAFTAR GAMBAR

Gambar 3.1 Tahapan pengembangan sistem model <i>prototype</i> .....	24
Gambar 4.1 Diagram alur kerja aplikasi galang dana berbasis web .....	35
Gambar 4.2 Hasil perancangan <i>use case diagram</i> .....	37
Gambar 4.3 Aktivitas <i>User</i> dalam proses donasi .....	38
Gambar 4.4 Aktivitas <i>User</i> dalam mempromosikan program galang dana .....	39
Gambar 4.5 Aktivitas Admin Lembaga dalam proses pendaftaran akun baru .....	40
Gambar 4.6 Aktivitas Admin Lembaga dalam mengelola program galang dana .....	41
Gambar 4.7 Aktivitas Admin Lembaga dalam mengelola laporan berita galang dana .....	42
Gambar 4.8 Aktivitas Admin Lembaga dalam proses penarikan dana .....	43
Gambar 4.9 Aktivitas Admin Lembaga dalam mengintegrasikan program galang dana dengan Facebook Ads .....	44
Gambar 4.10 Hasil perancangan <i>class diagram</i> pada implementasi <i>model</i> .....	45
Gambar 4.11 Hasil perancangan <i>class diagram</i> pada implementasi <i>controller</i> .....	46
Gambar 4.12 <i>Sequence diagram</i> dari proses donasi .....	47
Gambar 4.13 <i>Sequence diagram</i> dari proses promosi program oleh <i>User</i> .....	47
Gambar 4.14 <i>Sequence diagram</i> dari proses pendaftaran Admin Lembaga baru .....	48
Gambar 4.15 <i>Sequence diagram</i> dari proses pengelolaan program galang dana oleh Admin Lembaga .....	48
Gambar 4.16 <i>Sequence diagram</i> dari proses pengelolaan laporan berita galang dana oleh Admin Lembaga ....	48
Gambar 4.17 <i>Sequence diagram</i> dari proses penarikan dana oleh Admin Lembaga .....	49

Gambar 4.18 <i>Sequence diagram</i> dari proses integrasi Facebook Ads dengan program galang dana .....	49
Gambar 4.19 Hasil perancangan <i>entity relationship diagram</i> (ERD).....	50
Gambar 4.20 Rancangan tampilan halaman <i>homepage</i> .....	51
Gambar 4.21 Rancangan halaman detail program.....	51
Gambar 4.22 Rancangan halaman donasi baru .....	52
Gambar 4.23 Rancangan halaman pembayaran.....	52
Gambar 4.24 Rancangan status pembayaran.....	53
Gambar 4.25 Rancangan halaman <i>login</i> .....	53
Gambar 4.26 Rancangan halaman registrasi akun <i>User</i> .....	54
Gambar 4.27 Rancangan halaman registrasi akun Admin Lembaga.....	54
Gambar 4.28 Rancangan halaman dari akun <i>User</i> .....	55
Gambar 4.29 Rancangan halaman dari akun Admin Lembaga.....	55
Gambar 4.30 Rancangan halaman dari akun <i>Super Admin</i> .....	56
Gambar 4.31 Pola desain MVC pada Laravel Breeze.....	57
Gambar 4.32 Sampel hasil penulisan kode <i>route URL</i> .....	58
Gambar 4.33 Sampel hasil penulisan kode <i>controller</i> pada fitur pengelolaan program oleh Admin Lembaga.....	59
Gambar 4.34 Sampel hasil penulisan kode pada <i>model Campaign</i> .....	61
Gambar 4.35 Kode ReactJS untuk koneksi melalui InertiaJS ke <i>tag HTML</i> dengan <i>id</i> bernama <i>app</i> .....	63
Gambar 4.36 Kode dokumen HTML beserta <i>tag div</i> dengan <i>id</i> bernama <i>app</i> .....	64
Gambar 4.37 Hasil tampilan halaman <i>homepage</i> .....	64
Gambar 4.38 Hasil tampilan halaman detail program.....	65
Gambar 4.39 Hasil tampilan halaman donasi baru.....	65
Gambar 4.40 Hasil tampilan halaman pembayaran verifikasi manual.....	66
Gambar 4.41 Hasil tampilan halaman pembayaran verifikasi otomatis .....	66
Gambar 4.42 Hasil tampilan halaman status pembayaran.....	67
Gambar 4.43 Hasil tampilan halaman <i>dashboard</i> akun <i>User</i> .	67

Gambar 4.44 Hasil tampilan halaman pembuatan promosi program baru.....	68
Gambar 4.45 Hasil tampilan halaman profil akun <i>User</i> .....	68
Gambar 4.46 Hasil tampilan halaman <i>dashboard</i> akun Admin Lembaga .....	69
Gambar 4.47 Hasil tampilan halaman pengelolaan program.....	69
Gambar 4.48 Hasil tampilan halaman pembuatan program baru.....	70
Gambar 4.49 Hasil tampilan halaman pengelolaan laporan berita dan integrasi Facebook Ads dengan kode Pixel.....	70
Gambar 4.50 Hasil tampilan halaman data donasi.....	71
Gambar 4.51 Hasil tampilan halaman penarikan dana baru .....	71
Gambar 4.52 Hasil tampilan halaman profil Admin Lembaga .....	72
Gambar 4.53 Hasil tampilan halaman <i>dashboard</i> akun <i>Super Admin</i> .....	72
Gambar 4.54 Hasil tampilan halaman pengelolaan akun Admin Lembaga .....	73
Gambar 4.55 Hasil tampilan halaman pengelolaan akun <i>User</i> .....	73
Gambar 4.56 Hasil tampilan halaman pengelolaan program galang dana .....	74
Gambar 4.57 Hasil tampilan halaman pengelolaan kategori program galang dana.....	74
Gambar 4.58 Hasil tampilan halaman pengelolaan penarikan dana .....	75
Gambar 4.59 Hasil tampilan halaman permintaan pembayaran baru.....	75
Gambar 4.60 Hasil tampilan halaman riwayat pembayaran.....	76
Gambar 4.61 Hasil tampilan halaman pengaturan metode pembayaran yang di dukung .....	76

Gambar 4.62 Hasil tampilan halaman profil akun <i>Super Admin</i> .....	77
Gambar 4.63 Sampel pengujian menggunakan Page Speed Insight pada perangkat <i>desktop</i> .....	102
Gambar 4.64 Sampel pengujian menggunakan Page Speed Insight pada perangkat <i>mobile</i> .....	103
Gambar 4.65 Sampel hasil pengujian menggunakan GT Metrix .....	104
Gambar 4.66 Perbandingan kode program antara SPA dan MPA di <i>framework</i> Laravel .....	107
Gambar 4.67 Respons tipe konten dari <i>server</i> saat <i>multi page application</i> pertama kali dibuka.....	109
Gambar 4.68 Respons tipe konten dari <i>server</i> saat <i>single page application</i> pertama kali dibuka.....	109
Gambar 4.69 Struktur kode <i>HTML</i> saat <i>multi page application</i> pertama kali dibuka .....	110
Gambar 4.70 Struktur kode <i>HTML</i> saat <i>single page application</i> pertama kali dibuka .....	110
Gambar 4.71 Respons tipe konten dari <i>server</i> saat transisi halaman pada <i>multi page application</i> .....	111
Gambar 4.72 Respons tipe konten dari <i>server</i> saat transisi halaman pada <i>single page application</i> .....	112
Gambar 4.73 <i>Preview</i> dari respons tipe konten saat transisi halaman pada <i>single page application</i> .....	112
Gambar 4.74 Struktur kode hasil manipulasi <i>DOM</i> dalam proses transisi halaman di <i>single page application</i> .....	112



## DAFTAR LAMPIRAN

Lampiran 1 Komunikasi awal terkait kebutuhan sistem melalui percakapan di pesan WhatsApp .....	123
Lampiran 2 Pembentukan kantor sekretariat bersama sebagai bentuk keseriusan membangun <i>platform digital</i> Berbagi Mulia .....	124
Lampiran 3 Percakapan pesan Whatsapp terkait pengisian kuesioner untuk kebutuhan fitur aplikasi web .....	125
Lampiran 4 Hasil kuesioner analisis kebutuhan.....	126
Lampiran 5 Percakapan pesan Whatsapp terkait survey pengalaman pengguna sebagai bagian dari proses UAT <i>Super Admin</i> .....	131
Lampiran 6 Percakapan pesan Whatsapp terkait survey pengalaman pengguna sebagai bagian dari proses UAT Admin Lembaga .....	132
Lampiran 7 Percakapan pesan Whatsapp terkait survey pengalaman pengguna sebagai bagian dari proses UAT <i>User</i> sebagai donatur.....	133
Lampiran 8 Hasil kuesioner pengujian UAT <i>Super Admin</i> ...	134
Lampiran 9 Hasil kuesioner pengujian UAT Admin Lembaga .....	137
Lampiran 10 Hasil kuesioner pengujian UAT <i>User</i> sebagai donatur.....	140

# BAB I

## PENDAHULUAN

### A. Latar Belakang Masalah

Pesatnya perkembangan teknologi saat ini, memiliki dampak yang signifikan dalam kehidupan masyarakat. Hal tersebut berdampak pada dibutuhkannya inovasi dari berbagai sektor kehidupan dalam mengadopsi teknologi agar tetap mengikuti perkembangan zaman, salah satu diantaranya adalah penerimaan donasi dalam program galang dana oleh organisasi sosial atau lembaga. Penggalangan dana adalah kegiatan penghimpunan dana dengan cara menjual gagasan kepada para donatur yang kemudian bisa dimanfaatkan untuk membiayai terlaksananya gagasan tersebut (Pratiwi, 2010).

Penggalangan dana yang dilakukan oleh lembaga sosial berbasis keagamaan merupakan bentuk penghimpunan dana yang menjadi bagian dari perintah agama, yaitu zakat, infak, sedekah, dan wakaf (Nurul Bariyah, 2016). Perintah tersebut tertuang dalam QS. al-Taubah: 103 dengan bacaan sebagai berikut:

حُذِّ مِنْ أَمْوَالِهِمْ صَدَقَةٌ تُطَهِّرُهُمْ وَتُزَكِّيهِمْ بِهَا وَصَلِّ عَلَيْهِمْ إِنَّ  
صَلَوَاتِكَ سَكَنٌ لَهُمْ وَاللَّهُ سَمِيعٌ عَلِيمٌ

Artinya: “Ambillah zakat dari harta mereka (guna) menyucikan dan membersihkan mereka, dan doakanlah mereka karena sesungguhnya doamu adalah ketenteraman bagi mereka. Allah Maha Mendengar lagi Maha Mengetahui” (QS. al-Taubah: 103).

Proses penggalangan dana atau *fundraising* memiliki peran yang sangat penting bagi keberlangsungan hidup dari sebuah lembaga atau organisasi sosial. Hal ini dikarenakan organisasi sosial merupakan salah satu organisasi non-profit yang tidak mendapatkan keuntungan dari usaha yang dijalankannya. Organisasi pelayanan sosial yang telah lama berdiri tentunya menyusun dan menggunakan strategi *fundraising* agar dapat mencapai tujuan serta mempertahankan keberlangsungan organisasinya (Rachmasari et al., 2016). Oleh sebab itu, organisasi sosial harus selalu berinovasi dalam melakukan penggalang dana agar dapat sesuai dengan perkembangan zaman.

Yayasan Amal Bakti Nusantara Tegal merupakan salah satu organisasi nirlaba yang bergerak di bidang pendidikan, sosial, dan keagamaan. Yayasan ini berlokasi di Kabupaten Tegal dan telah memiliki status badan

hukum sejak tahun 2016. Dalam menjalankan organisasinya, Yayasan Amal Bakti Nusantara Tegal mendapatkan pendanaan dari orang tua siswa dan dinas pendidikan melalui kegiatan belajar mengajar di PAUD Bunaya yang dikelolanya. Penggalangan dana juga kerap kali dilakukan dengan tujuan untuk menyantuni kaum duafa dengan wali murid sebagai donaturnya.

Pesatnya perkembangan teknologi memberikan sebuah gagasan terkait pemanfaatannya dalam proses penggalangan dana. Gagasan tersebut dikemukakan oleh Yayasan Amal Bakti Nusantara Tegal yang ingin membuat sebuah *platform digital* untuk penggalangan dana dalam bentuk aplikasi web dengan merek dagang Berbagi Mulia. Aplikasi web tersebut nantinya diharapkan dapat menjembatani berbagai organisasi sosial non-profit dengan mengkampanyekan program yang digagasnya untuk mendapatkan pendanaan dari para donatur yang ingin mendonasikan sebagian hartanya melalui aplikasi web yang telah dibuat.

Aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh *browser* web (Susanti et al.,

2017). Pengguna dapat menjalankan aplikasi web melalui *web browser* menggunakan perangkat yang terhubung dengan jaringan internet. Dikarenakan menggunakan jaringan internet, maka informasi yang ditampilkanpun dapat disebarluaskan dengan mudah, baik di skala nasional atau bahkan internasional. Bagi para penggalang dana, hal tersebut diharapkan dapat membawa donatur baru dengan memberikan kemudahan berdonasi melalui aplikasi berbasis web. Oleh karena itu, diperlukan sebuah sistem informasi yang mumpuni dalam menangani semua proses penggalangan dana, mulai dari pengelolaan lembaga selaku pemilik program, pengelolaan informasi program yang ditampilkan, pengelolaan proses donasi hingga pengelolaan pembayaran yang dilakukan oleh donatur.

Untuk menjawab kebutuhan tersebut, dalam penelitian ini akan dilakukan pengembangan aplikasi galang dana berbasis web. Aplikasi web ini nantinya diharapkan memiliki kinerja yang baik dalam pemrosesan data dan dilengkapi dengan antarmuka web yang *modern*. Untuk itu, penelitian ini akan berfokus pada penerapan *single sage application* sebagai metode

penyajian halaman web dengan menggunakan beberapa teknologi yang diintegrasikan oleh Laravel Breeze.

*Single page application* (SPA) merupakan teknologi yang bekerja di dalam *browser* yang tidak membutuhkan *reload page* saat digunakan (Santoso, 2021). Penggunaan *Single page application* ini memiliki kinerja dalam penyajian halaman web yang lebih cepat dari pada *multi page application* (MPA). Hal tersebut dikarenakan *multi page application* menggunakan cara tradisional dalam menyajikan sebuah halaman web (Santoso, 2021). *Multi page application* yang tersusun atas banyak halaman menyebabkan dibutuhkan proses pemuatan ulang setiap kali ada interaksi pengguna pada sebuah sistem aplikasi web. Berbeda dengan *single page application* yang tersusun atas satu halaman saja dan bersifat dinamis. Hal ini menyebabkan sebagian besar interaksi pengguna dapat ditangani dari sisi *client* dan hanya sedikit yang mencapai sisi *server*, sehingga proses penyajian halaman dapat dijalankan lebih cepat tanpa pemuatan ulang dari halaman yang sedang ditampilkan. *Single page application* menggunakan *library* ReactJS dapat memaksimalkan *web browser* agar lebih cepat

dalam menampilkan konten atau isi dari keseluruhan halaman secara sempurna (Santoso, 2021).

Penggunaan *library* ReactJS sebagai teknologi yang membangun *single page application* di dalam ekosistem Laravel Breeze juga merupakan sesuatu yang baru dalam dunia pemrograman. Laravel Breeze sendiri merupakan sekumpulan teknologi yang menjadi salah satu ekosistem dalam *framework* Laravel. Dalam pengembangannya, *Single page application* biasanya membutuhkan REST API sebagai penghubung antara *server-side* dengan *client-side*. Namun, Laravel Breeze menggunakan metode yang berbeda, yaitu dengan menggunakan InertiaJS yang menjadi perantara bagi Laravel sebagai *server-side* dan ReactJS sebagai *client-side* yang membuat keduanya dapat berkomunikasi secara langsung, sehingga tidak membutuhkan REST API. Hal tersebut mengakibatkan proses pengembangan aplikasi web dapat diselesaikan lebih cepat dengan keamanan sistem yang lebih optimal. Terlebih lagi, *framework* Laravel dibangun menggunakan bahasa pemrograman PHP yang telah didukung oleh banyak layanan *cloud hosting* dengan harga yang relatif murah. Hal tersebut menjadi solusi bagi *platform digital*

yang baru dirintis seperti Berbagi Mulia milik Yayasan Amal Bakti Nusantara Tegal.

## **B. Rumusan Masalah**

Berdasarkan latar belakang masalah di atas, diperoleh rumusan masalah yang dapat diangkat pada penelitian ini, yaitu bagaimana penerapan *single page application* (SPA) sebagai metode penyajian halaman dalam proses pengembangan antarmuka aplikasi galang dana berbasis web menggunakan teknologi yang diintegrasikan oleh Laravel Breeze.

## **C. Batasan Masalah**

Berdasarkan rumusan masalah di atas, maka dapat di peroleh batasan masalah yang dibahas pada penelitian ini antara lain sebagai berikut:

1. Proses penerapan *single page application* (SPA) ini akan menggunakan beberapa teknologi yang telah diintegrasikan oleh Laravel Breeze, seperti Laravel yang berperan sebagai *server-side*, ReactJS sebagai *client-side*, dan InertiaJS sebagai penghubung antar keduanya,
2. Antarmuka aplikasi web ini akan dibangun dengan pengkodean menggunakan Tailwind CSS yang terintegrasi di dalam *library* ReactJS,



3. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *software development lifecycle* (SDLC) model *prototype*,
4. Pengujian sistem yang digunakan pada penelitian ini adalah metode *black box testing* dan *user acceptance test*.

#### **D. Tujuan Penelitian**

Tujuan yang ingin dicapai pada penelitian ini adalah menerapkan *single page application* (SPA) sebagai metode penyajian halaman dalam proses pengembangan antarmuka aplikasi galang dana berbasis web melalui penggunaan *library* ReactJS dan beberapa teknologi lain yang telah diintegrasikan oleh Laravel Breeze.

#### **E. Manfaat Penelitian**

1. Manfaat Akademis

Penelitian ini diharapkan dapat menambah koleksi pustaka akademik dengan topik pembahasan terkait penerapan *single page application* sebagai metode penyajian halaman dalam pengembangan antarmuka aplikasi berbasis web menggunakan teknologi yang telah diintegrasikan oleh Laravel Breeze.

2. Manfaat Praktis

Penelitian ini diharapkan dapat memberikan manfaat secara umum bagi lembaga atau organisasi sosial sebagai sarana untuk mendapatkan pendanaan melalui kegiatan penggalangan dana menggunakan aplikasi web yang telah dibuat dan secara khusus bagi Yayasan Amal Bakti Nusantara Tegal selaku penggagas merek dagang Berbagi Mulia dapat menggunakan aplikasi web yang telah dibuat sebagai *platform* digital untuk penggalangan dana, baik untuk yayasannya sendiri maupun untuk organisasi sosial atau lembaga lain yang mendaftar sebagai penggalang dana.

## BAB II

### LANDASAN PUSTAKA

#### A. Kajian Pustaka

##### 1. *Single Page Application*

*Single page application* atau biasa disebut dengan SPA merupakan teknologi yang bekerja di dalam *browser* yang tidak membutuhkan *reload page* saat digunakan (Santoso, 2021). Dalam penggunaan SPA, aplikasi web hanya akan menampilkan satu halaman saja yang bersifat dinamis.

Saat pertama kali dibuka, aplikasi web akan melakukan inisialiasi dengan melakukan *rendering* semua *resource* yang dibutuhkan, seperti modul *HTML*, *CSS*, dan *Javascript*. Kemudian *resouce* tersebut akan dimuat berdasarkan interaksi pada aplikasi web. Hal tersebut membuat proses kerja aplikasi web menjadi lebih cepat karena setiap interaksi pengguna dapat ditangani dari sisi *client*, sehingga tidak perlu memuat ulang halaman pada setiap perubahannya. Sedangkan, sisi *server* hanya akan menangani kebutuhan pengambilan data yang ditampilkan dalam bentuk *Javascript Object Notation (JSON)*.

## 2. Laravel Breeze

Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*model view controller*). Laravel adalah *web development* berbasis MVC yang ditulis dalam bahasa PHP dan dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal serta biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi melalui sintaks yang ekspresif, jelas dan menghemat waktu (Hermanto et al., 2019).

Kelebihan laravel di antara *framework* PHP lainnya adalah banyaknya ekosistem yang mengintegrasikan teknologi pemrograman yang memudahkan para penggunanya. Salah satu di antaranya adalah Laravel Breeze. Laravel Breeze merupakan *package* Laravel yang digunakan sebagai *starter kit* untuk memulai dalam pengembangan aplikasi web beserta antarmukanya yang dilengkapi dengan fungsi otentifikasi dasar seperti *login*, *register*, *reset password*, hingga verifikasi *email*.

Secara bawaan, teknologi yang digunakan dalam pengembangan antarmuka di Laraval Breeze adalah

Blade Template yang berbasis *multi page application*. Namun, Laravel Breeze juga menawarkan pilihan teknologi antarmuka lain berbasis Javascript seperti ReactJS dan VueJS yang biasa digunakan dalam pembuatan aplikasi web *single page application*. Selain itu, Laravel Breeze juga telah dilengkapi Tailwind CSS sebagai kerangka kerja yang memudahkan dalam membuat antarmuka aplikasi web. Untuk menangani kebutuhan pertukaran data antara *server-side* dan *client-side*, Laravel Breeze menggunakan InertiaJS sebagai pengganti dari REST API.

### 3. ReactJS

ReactJS atau dapat juga disebut dengan React merupakan *library* Javascript berbasis komponen yang mengikuti paradigma pemrograman deklaratif. Tampilan deklaratif tersebut dapat digunakan untuk membuat antarmuka interaktif yang kompleks dan disajikan sebagai komponen presentasi dari sebuah web (Mukhiya & Hoang Hung, 2018). ReactJS biasanya digunakan untuk membangun antarmuka aplikasi web berbasis *single page application* (SPA). ReactJS menerapkan konsep *component-based*,

artinya setiap antarmuka yang dibangun, tersusun dari beberapa komponen yang disatukan hingga membentuk sebuah halaman web yang utuh. Dengan menerapkan konsep *component-base* ini, memungkinkan pengguna membuat komponen yang dapat digunakan berulang kali pada halaman yang berbeda.

Fitur lain yang diunggulkan oleh ReactJS adalah Virtual *DOM*. Virtual *DOM* (*document object model*) adalah sebuah konsep dalam pemrograman yang merepresentasikan secara virtual dari antarmuka pengguna yang disimpan di dalam memori dan disinkronkan dengan *DOM* asli oleh *library* ReactJS seperti ReactDOM. *DOM* sendiri merupakan *interface* yang digunakan oleh JavaScript dalam mengubah tampilan sebuah situs web secara dinamis. Manfaat penerapan virtual *DOM* oleh ReactJS adalah setiap kali terjadi perubahan pada *DOM*, virtual *DOM* akan berubah seketika karena terletak di memori. Karena itu, pembaruan *DOM* tidak dilakukan secara rutin dan kecepatan kinerja aplikasi tidak terganggu oleh *DOM*. (Ham, 2019)

#### 4. Tailwind CSS dan DaisyUI

Berdasarkan situs resminya, [tailwindcss.com](https://tailwindcss.com), Tailwind CSS didefinisikan sebagai *framework* CSS yang bersifat *utility-first* untuk membangun desain antarmuka dengan cepat yang dikemas dalam bentuk *class HTML*. Dengan mengusung konsep *utility-first*, Tailwind CSS dapat memudahkan pengguna dalam memodifikasi antarmuka aplikasi web dengan menyusun *class-class utility* milik Tailwind CSS untuk membentuk komponen antarmuka sesuai selera.

Dalam beberapa kasus, konsep *utility-first* tidak selamanya cocok dengan penggunaannya terutama bagi yang baru beralih dari *framework* CSS lain berbasis komponen seperti Bootstrap. Untuk mengatasi hal tersebut, penggunaan Tailwind CSS dapat ditambahkan *library* khusus yang menyediakan komponen antarmuka seperti DaisyUI. DaisyUI adalah sebuah pustaka komponen untuk Tailwind CSS dan dapat digunakan secara gratis karena sifatnya *open-source*. DaisyUI menawarkan *class* Tailwind CSS yang lebih sederhana dengan adanya kemiripan penamaan *class* komponen seperti milik Bootstrap. Hal tersebut dapat memudahkan

pengguna yang baru beralih dari Bootstrap ke Tailwind CSS tanpa mengurangi keunggulan dari Tailwind CSS itu sendiri yang memberi kebebasan dalam mendesain antarmuka aplikasi web.

#### 5. Xampp dan MySQL

Xampp adalah perangkat lunak *open-source* yang berisi kompilasi beberapa program serta dapat dijalankan oleh banyak sistem operasi. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl (Riyadli et al., 2020). Nama Xampp merupakan singkatan dari X (*cross-platform*), Apache, MySQL, PHP dan Perl.

Pada penelitian kali ini, Xampp digunakan sebagai MySQL Server sebagai basis data relasional yang menangani penyimpanan data dari sebuah sistem informasi. MySQL menggunakan bahasa SQL atau *Structured Query Language* yang merupakan bahasa standar dalam mengelola *relational database management system* (RDBMS). Dikarenakan Laravel sendiri telah memiliki *web server* dalam bentuk perintah Artisan, maka Apache milik Xampp hanya



berfungsi sebagai *web server* untuk PHPMyAdmin. PHPMyAdmin adalah alat yang dapat digunakan dengan mudah untuk manajemen *database* MySQL secara visual, sehingga kita tidak perlu lagi harus menulis perintah SQL setiap akan melakukan perintah operasi *database* (Damayanti, 2019). PHPMyAdmin secara otomatis telah terinstall bersama paket yang dibawa oleh Xampp.

#### 6. *Core Web Vital*

*Core web vital* adalah serangkaian faktor utama yang ditetapkan oleh Google untuk menilai pengalaman pengguna dari sebuah situs webv (Walton, 2022).Beberapa metrik pengukuran *core web vital* yang digunakan pada penelitian ini, antara lain sebagai berikut:

##### a. *First Contentful Paint* (FCP)

FCP digunakan untuk menghitung performa pemuatan halaman berkaitan dengan konten dari *Document Object Model (DOM)* yang pertama kali muncul dalam sebuah halaman web. Ambang batas nilai FCP yang harus dicapai agar sebuah halaman web dikatakan memiliki performa yang baik adalah kurang dari 1800 ms.

b. *Total Blocking Time (TBT)*

TBT digunakan untuk menghitung total waktu pemuatan halaman yang diblokir dan membuatnya tidak bisa merespon interaksi pengguna. Sejak tahun 2020, TBT menggantikan metrik *First Input Delay (FID)* sebagai salah satu metrik utama dalam *core web vital*. Ambang batas nilai TBT yang harus dicapai agar sebuah halaman web dikatakan memiliki performa yang baik adalah kurang dari 200 ms.

c. *Largest Contentful Paint (LCP)*

LCP digunakan untuk menghitung performa pemuatan halaman yang berkaitan dengan blok gambar atau teks yang ukurannya paling besar. Ambang batas nilai LCP yang harus dicapai agar sebuah halaman web dikatakan memiliki performa yang baik adalah kurang dari 2500 ms.

d. *Cumulative Layout Shift (CLS)*

CLS digunakan untuk mengukur stabilitas visual seperti pergeseran *layout* (tata letak) yang tidak terduga di halaman web. Ambang batas nilai CLS yang harus dicapai agar sebuah halaman

web dikatakan memiliki performa yang baik adalah kurang dari 0,1.

## B. Kajian Penelitian yang Relevan

Di bawah ini merupakan beberapa penelitian sebelumnya yang relevan berdasarkan pada kesamaan topik atau pemanfaatan teknologi yang menjadi referensi dari penelitian ini. Penjelasan singkat terkait perbandingan setiap penelitian terdahulu disajikan dalam tabel berikut ini:

Tabel 2.1 Perbandingan penelitian terdahulu

1	<b>Judul</b>	Implementasi <i>Framework</i> Laravel Untuk Membangun Sistem Administrasi Pembayaran Air Pada PPAB Perum Dolog Dengan Metode Waterfall
	<b>Penulis</b>	(Husada, 2019)
	<b>Teknologi</b>	<i>Framework</i> Laravel dan MySQL
	<b>Kelebihan</b>	Menghasilkan sistem administrasi pembayaran air yang berisi informasi catatan pembayaran air pelanggan, data pelanggan, dan informasi tagihan air
	<b>Kekurangan</b>	Masih menggunakan <i>multi page application</i> sebagai metode penyajian halaman web
2	<b>Judul</b>	Sistem Informasi Layanan Pemesanan Pada Restoran Cindelaras Kota Medan Dengan Menggunakan <i>Framework</i> Laravel
	<b>Penulis</b>	(Sinaga, 2020)
	<b>Teknologi</b>	<i>Framework</i> Laravel dan MySQL
	<b>Kelebihan</b>	Menghasilkan sistem informasi reservasi tempat dan menu yang ada pada

		restoran dengan transfer bank sebagai metode pembayarannya
	<b>Kekurangan</b>	Masih menggunakan <i>multi page application</i> sebagai metode penyajian halaman web
3	<b>Judul</b>	Pengembangan <i>E-Learning</i> Menggunakan Laravel dan React
	<b>Penulis</b>	(Septiawan, 2019)
	<b>Teknologi</b>	Laravel, ReactJS, dan PostgreSQL
	<b>Kelebihan</b>	<ol style="list-style-type: none"> <li>1. Menghasilkan aplikasi <i>e-learning</i> yang digunakan sebagai sarana pendistribusian materi pelajaran dari pengajar ke siswa,</li> <li>2. Adanya fitur kuis untuk mengukur pemahaman siswa terhadap materi yang diberikan</li> </ol>
	<b>Kekurangan</b>	Telah menggunakan <i>single page application</i> sebagai metode penyajian halaman web, namun masih membutuhkan REST API sebagai penghubung antara <i>server-side</i> dengan <i>client-side</i>
4	<b>Judul</b>	Rancang Bangun Sistem Informasi Penyewaan Alat-Alat Outdoor Berbasis Web
	<b>Penulis</b>	(Hidayati, 2020)
	<b>Teknologi</b>	Laravel, VueJS dan MySQL
	<b>Kelebihan</b>	Menghasilkan sistem informasi berbasis website <i>single page application</i> yang digunakan untuk pemesanan alat secara <i>online</i> menggunakan dan dilengkapi laporan transaksi secara akurat untuk keperluan pengelolaan data
	<b>Kekurangan</b>	<ol style="list-style-type: none"> <li>1. Telah menggunakan <i>Single Page Application</i> sebagai metode penyajian halaman web, namun masih membutuhkan REST API sebagai penghubung antara <i>server-side</i></li> </ol>

		dengan <i>client-side</i> 2. Belum mengimplementasikan <i>payment gateway</i> sebagai metode pembayarannya
5	<b>Judul</b>	Pembuatan Aplikasi Web Untuk Manajemen Proyek <i>Software</i> Dalam Tim Berbasis <i>Single Page Application</i> Menggunakan <i>Library</i> ReactJS
	<b>Penulis</b>	(Hargianto, 2018)
	<b>Teknologi</b>	NodeJS, ReactJS dan MySQL
	<b>Kelebihan</b>	Menghasilkan aplikasi web berbasis <i>single page application</i> yang digunakan untuk memudahkan dalam memanajemen suatu proyek perangkat lunak
	<b>Kekurangan</b>	Telah menggunakan <i>single page application</i> sebagai metode penyajian halaman web, namun masih membutuhkan REST API sebagai penghubung antara <i>server-side</i> dengan <i>client-side</i>

Dari beberapa ulasan penelitian diatas, dapat diketahui belum ada yang membahas terkait penerapan *single page application* menggunakan *library* ReactJS dan teknologi lain dalam ekosistem Laravel Breeze. Berikut ini merupakan beberapa yang menjadi perbedaan dengan penelitian oleh penulis kali ini:

1. Penerapan *single page application* (SPA) sebagai metode penyajian halaman web tanpa REST API di dalam ekosistem Laravel Breeze melalui penggunaan InertiaJS sebagai penghubung antara *server-side* dan

- client-side*. Penerapan SPA ini diharapkan dapat memberikan kinerja yang baik dalam pemrosesan data karena tidak dibutuhkannya pemuatan halaman pada setiap interaksi yang dilakukan oleh pengguna,
2. Minimalisasi biaya operasional dari sistem yang berjalan. Aplikasi web dengan *framework* Laravel ini, dibangun menggunakan bahasa pemrograman PHP yang telah didukung oleh banyak layanan *cloud hosting* dengan harga yang relatif murah,
  3. Aplikasi web ini akan memiliki fitur yang lengkap dalam menunjang proses penggalangan dana oleh berbagai organisasi sosial yang terdaftar, seperti pengelolaan program galang dana, pengelolaan laporan berita galang dana, dan fitur penarikan dana dari donasi yang terkumpul,
  4. Aplikasi web ini juga diharapkan dapat memudahkan para donatur yang ingin berdonasi dengan disediakan fitur metode pembayaran verifikasi otomatis menggunakan Midtrans sebagai *payment gateway* dan transfer bank sebagai metode pembayaran dengan verifikasi manual.

## **BAB III**

### **METODE PENELITIAN**

#### **A. Metode Pengumpulan Data**

##### **1. Studi Lapangan**

Studi lapangan dilakukan untuk pengumpulan data yang dibutuhkan demi terselenggaranya penelitian ini. Beberapa hal yang dilakukan pada tahap ini antara lain sebagai berikut:

##### **a. Kuesioner**

Kuesioner merupakan alat pengumpulan data primer dengan metode survei untuk memperoleh opini responden. Kuesioner biasanya digunakan untuk memperoleh informasi pribadi misalnya sikap, opini, harapan dan keinginan responden (Pujihastuti, 2010). Pengisian kuesioner ini dilakukan oleh pihak Yayasan Amal Bakti Nusantara Tegal. Dari hasil pengisian kuesioner tersebut, penulis akan mendapatkan informasi yang menjadi kebutuhan sistem dari aplikasi web yang akan dibuat.

##### **b. Observasi**

Penulis juga akan melakukan observasi atau pengamatan secara langsung terkait permasalahan yang ada di Yayasan Amal Bakti

Nusantara Tegal dan dapat dijadikan sebagai acuan pentingnya dilakukannya penelitian ini.

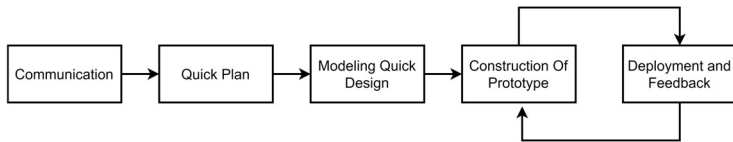
## 2. Studi Pustaka

Untuk mendapatkan referensi terhadap teori-teori yang diperlukan, penulis melakukan studi pustaka dengan mencari literatur yang berhubungan dengan penelitian ini dan mempelajarinya dengan seksama. Literatur tersebut dapat berupa jurnal penelitian, skripsi, *e-book*, dan artikel yang beredar di internet.

### **B. Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *software development lifecycle* (SDLC) model *prototype*. SDLC adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut (Wahid, 2020). Sedangkan, model *prototype* merupakan model pengembangan perangkat lunak yang dapat diterapkan saat kebutuhan pengguna hanya dapat didefinisikan secara umum dan tidak terperinci terkait fungsi dan fitur yang akan dibuat (Pressman, 2010). Beberapa tahapan dalam model ini dapat di lihat pada diagram berikut ini:





Gambar 3.1 Tahapan pengembangan sistem model *prototype*

Pada penelitian ini, akan menerapkan tahapan model *prototype* di atas yang dijelaskan pada uraian berikut ini:

### 1. *Communication*

Pada tahap ini, dilakukan komunikasi awal dengan mendefinisikan secara umum terkait kebutuhan sistem apa yang akan dibuat. Proses komunikasi dilakukan melalui percakapan di aplikasi WhatsApp yang mengumpulkan informasi berupa sistem seperti apa yang akan dibuat nantinya.

### 2. *Quick Plan*

Pada tahap ini, dilakukan perencanaan awal mengenai kebutuhan sistem yang akan dibuat dengan lebih terperinci. Proses perencanaan awal ini dilakukan dengan mendefinisikan apa saja fitur aplikasi web yang akan dibuat dan diperjelas dengan menggambarkan cara kerjanya dalam bentuk sebuah diagram. Untuk validasi kebutuhan sistem tersebut, dilakukan pengisian kuesioner oleh pihak Yayasan Amal Bakti Nusantara Tegal selaku pemilik *platform*

*digital* Berbagi Mulia dan permintaan pengisian kuesioner tersebut disampaikan melalui aplikasi WhatsApp.

### 3. *Modeling Quick Design*

Pada tahap ini, dilakukan perancangan sistem berdasarkan perencanaan awal yang telah disepakati. Tujuannya adalah agar dapat memberikan gambaran terhadap alur kerja aplikasi web yang lebih lengkap. Beberapa hal yang dihasilkan pada tahap ini antara lain sebagai berikut:

#### 1. *Unified Modelling Language (UML)*

*Unified modelling language (UML)* adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Haviluddin, 2011). UML biasanya digunakan sebagai pemodelan secara visual dalam perancangan perangkat lunak yang menggunakan paradigma pemrograman berorientasi objek. UML memiliki tiga kategori utama yang menjelaskan arsitektur sistem dan saling terintegrasi, yaitu *structural diagram*, *behaviour diagram* dan *interaction diagram* (Haviluddin, 2011). Terdapat

empat diagram yang dibuat pada tahap ini dan mewakili setidaknya satu kategori utama pemodelan UML dengan hasil sebagai berikut:

a. *Use case diagram*

*Use case diagram* ini termasuk ke dalam kategori *behavior diagram* yang digunakan untuk mendeskripsikan hubungan antara sebuah sistem dengan penggunaanya atau bahkan dengan sistem lain. *Use case diagram* terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, ataupun sistem lain yang berinteraksi dengan sistem yang akan dibuat (Kurniawan & Syarifuddin, 2020).

b. *Activity Diagram*

*Activity diagram* adalah diagram menggambarkan aliran aktivitas dalam sistem yang sedang dirancang (Kurniawan & Syarifuddin, 2020). *Activity diagram* ini termasuk ke dalam kategori *behavior diagram* yang digunakan untuk memodelkan setiap proses pada sebuah sistem perangkat lunak

dan merupakan pengembangan dari *use case diagram*.

c. *Class Diagram*

*Class diagram* termasuk ke dalam kategori *structural diagram* yang mendeskripsikan struktur statis dari kelas dalam sebuah sistem dengan menggambarkan atribut, operasi dan hubungan antara kelas (Haviluddin, 2011). *Class diagram* membantu dalam visualisasi kelas dari sebuah sistem perangkat lunak yang akan dibuat.

d. *Sequence Diagram*

*Sequence diagram* termasuk ke dalam kategori *interaction diagram* yang menggambarkan setiap tahapan dalam sistem terkait interaksi objek berdasarkan urutan waktu (Haviluddin, 2011). *Sequence diagram* digunakan untuk menjelaskan tahapan yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.

2. *Entity Relationship Diagram (ERD)*

*Entity relationship diagram* atau ERD adalah sebuah diagram yang menggambarkan relasi

setiap entitas dalam perancangan basis data. Sebuah ERD mendeskripsikan data yang akan disimpan ke dalam sistem maupun batasannya (Latukolan et al., 2019).

### 3. *Wireframe*

*Wireframe* merupakan sebuah kerangka yang mengatur tata letak setiap komponen pada antarmuka website atau aplikasi. Komponen tersebut berkaitan dengan teks, gambar, tombol, dan lain sebagainya. *Wireframe* masuk ke dalam golongan *low fidelity* yang artinya desain dengan tingkat persepsi rendah dengan tujuan untuk menunjukkan tata letak (Julianto, 2020). Pembuatan *wireframe* ini merupakan salah satu bentuk perancangan antarmuka dari aplikasi web yang akan dibuat.

### 4. *Construction of Prototype*

Pada tahap ini, dilakukan pembuatan *prototype* melalui aktivitas pengkodean dengan menerapkan perancangan sistem yang telah dibuat sebelumnya. Pengkodean tersebut dikerjakan menggunakan beberapa teknologi yang telah diintegrasikan oleh Laravel Breeze, di antaranya: Laravel sebagai *server*

*side*, ReactJS sebagai *client side*, dan InertiaJS sebagai penghubung keduanya.

Laravel Breeze secara bawaan menggunakan Tailwind CSS sebagai *framework* CSS yang digunakan mengatur tata letak antarmuka suatu aplikasi web. Dikarenakan Tailwind CSS merupakan *framework* CSS berbasis *utility*, maka diperlukan *library* tambahan yang menyediakan komponen CSS seperti DaisyUI. DaisyUI menawarkan *class* CSS berbasis komponen yang lebih sederhana dan memiliki kemiripan nama *class* CSS milik Bootstrap. Hal tersebut memudahkan penulis, dalam mengimplementasikan rancangan antarmuka yang telah dibuat sebelumnya.

Untuk menunjang aktivitas pengkodean ini, beberapa alat tambahan seperti Visual Studio Code dan Xampp. Dikarenakan Laravel telah memiliki *web server*-nya sendiri dalam bentuk perintah artisan, maka Xampp hanya digunakan sebagai server MySQL sebagai DBMS yang mendukung menggunakan fitur *object relation mapping* (ORM) milik Laravel. Sedangkan Visual Studio Code merupakan teks *editor* yang digunakan untuk menangani semua aktivitas pengkodean dari pengembangan suatu aplikasi web.

## 5. *Deployment Delivery and Feedback*

Pada tahap ini, dilakukan proses *deployment* dari aplikasi web yang telah dibuat ke dalam layanan *cloud hosting*, kemudian dilakukan analisis *feedback* oleh pihak Yayasan Amal Bakti Nusantara Tegal terkait aplikasi web yang sudah dibuat. Pada tahap ini juga dilakukan pengujian sistem sebelum dapat digunakan oleh para pengguna nantinya. Pengujian ini menggunakan metode *black box testing* yang mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak tersebut. Sedangkan, teknik *black box testing* yang digunakan adalah *equivalence partitioning*. Teknik *equivalence partitioning* merupakan teknik pengujian yang menggunakan masukan pada setiap menu di dalam sistem informasi dengan dikelompokkan berdasarkan fungsinya (Wijaya & Astuti, 2021).

Untuk kebutuhan validasi terhadap fitur aplikasi web yang telah dibuat, pada tahap ini juga dilakukan pengujian berbasis *user acceptance test* (UAT) dengan bentuk pengisian kuesioner berdasarkan setiap *role* pengguna aplikasi web ini, seperti pihak Yayasan

Amal Bakti Nusantara Tegal sebagai *Super Admin* atau pemilik dari *platform digital* Berbagi Mulia, Admin Lembaga sebagai lembaga penggalang dana dan *User* sebagai donatur dari proses penggalangan dana. *User acceptance testing* (UAT) merupakan pengujian akhir dari pengembangan sebuah produk untuk mendapatkan validasi terhadap sistem yang dibangun apakah sudah sesuai dengan kebutuhan pengguna (Munthe et al., 2016). Pengujian UAT dilakukan untuk mendapatkan tanggapan dari setiap pengguna terhadap bekerja tidaknya fitur aplikasi web yang telah dibuat. Pada tahap ini juga dilakukan pengujian performa aplikasi web menggunakan beberapa tools seperti PageSpeed Insight dan GT Metrix. Pengujian performa dilakukan untuk mengetahui kinerja aplikasi web yang telah dibuat melalui perhitungan dengan beberapa metrik yang telah distandarisasi sebagai *core web vital*.



## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### A. *Communication*

Proses komunikasi awal dilakukan pengumpulan data melalui percakapan di dalam grup WhatsApp yang beranggotakan Mas Rokhim sebagai Pembina Yayasan Amal Bakti Nusantara Tegal sekaligus Direktur Berbagi Mulia, Mas Umam sebagai *Digital Marketing* Berbagi Mulia dan penulis sendiri sebagai *Web Developer* Berbagi Mulia.

Berdasarkan bukti yang terlampir pada Lampiran 1, didapatkan informasi terkait definisi secara umum terkait kebutuhan sistem dari aplikasi web yang akan dibuat. Yayasan Amal Bakti Nusantara Tegal ingin membuat sebuah *platform digital* untuk penggalangan dana yang dapat menjembatani berbagai organisasi sosial non-profit dengan mengkampanyekan program yang digagasnya untuk mendapatkan pendanaan dari para donatur. Dalam percakapan tersebut, tertulis juga besaran dana operasional yang dapat digunakan oleh para penggalang dana sebesar 12,5% dari total donasi yang terkumpul.

Penulis juga melakukan observasi atau pengamatan langsung terhadap permasalahan yang dialami oleh Yayasan Amal Bakti Nusantara Tegal. Berdasarkan bukti yang terlampir pada Lampiran 2, Yayasan Amal Bakti

Nusantara Tegal memiliki keseriusan dalam membangun *platform digital* penggalangan dana yang tercermin dengan pembentukan kantor sekretariat bersama untuk kepengurusan Berbagi Mulia.

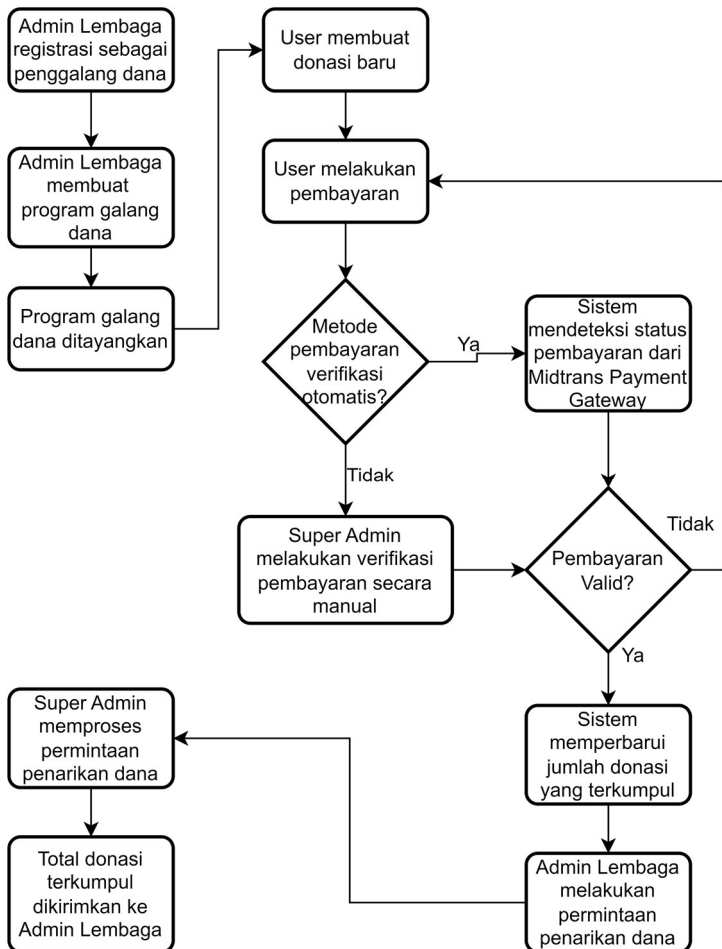
## **B. Quick Plan**

Proses perencanaan awal dilakukan melalui pendefinisian yang lebih terperinci terkait kebutuhan sistem dari aplikasi web yang akan dibuat. Proses ini menggunakan metode kuesioner oleh pihak Yayasan Amal Bakti Nusantara Tegal yang disampaikan melalui pesan WhatsApp (terlampir pada Lampiran 3) sebagai validasi terhadap fitur aplikasi web yang diajukan oleh penulis. Berdasarkan bukti yang terlampir pada Lampiran 4, Beberapa fitur aplikasi web yang telah disetujui adalah sebagai berikut:

1. Fitur pembagian *role* pengguna terdaftar terkait pengelolaan halaman sesuai hak aksesnya,
2. Fitur registrasi bagi penggalang dana dan donatur,
3. Fitur pengelolaan program galang dana yang ditampilkan,
4. Fitur pengelolaan laporan berita dari program galang dana,

5. Fitur pengelolaan donasi hingga proses pembayaran oleh donatur,
6. Fitur promosi program galang dana oleh akun donatur yang terdaftar,
7. Fitur penarikan dana yang dilakukan oleh lembaga pemilik program galang dana,
8. Fitur terkait transfer bank sebagai metode pembayaran yang diverifikasi secara manual,
9. Fitur terkait *bank virtual account* sebagai metode pembayaran yang diverifikasi secara otomatis menggunakan *Midtrans Payment Gateway*,
10. Fitur terkait integrasi dengan layanan *marketing tools*, seperti Facebook Ads.

Beberapa fitur di atas menghasilkan alur kerja sistem aplikasi web yang digambarkan pada diagram berikut ini:



Gambar 4.1 Diagram alur kerja aplikasi galang dana berbasis web

### C. *Modelling Quick Design*

Pada tahap ini, dilakukan perancangan sistem berdasarkan perencanaan awal yang telah disepakati.

Perancangan sistem ini menggunakan beberapa pemodelan seperti UML, ERD, dan *Wireframe* dengan hasil sebagai berikut:

1. *Unified Modelling Language* (UML)

Terdapat empat diagram yang dibuat pada tahap ini dan mewakili setidaknya satu kategori utama dari pemodelan UML. Keempat diagram tersebut antara lain sebagai berikut:

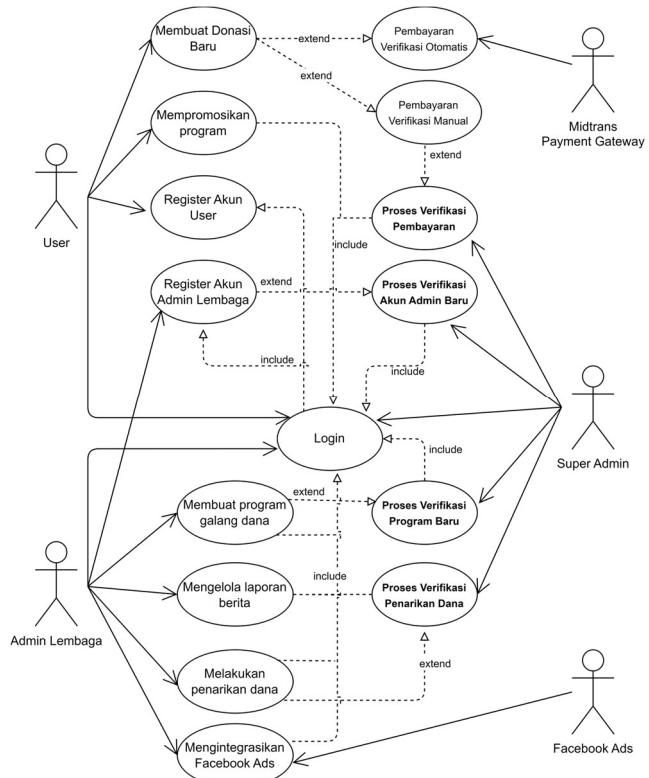
a. *Use case diagram*

*Use case diagram* ini termasuk ke dalam kategori *behavior diagram* yang digunakan untuk mendeskripsikan hubungan antara sebuah sistem dengan penggunanya atau bahkan dengan sistem lain. Oleh karena itu, berdasarkan Gambar 4.1 yang merupakan proses alur kerja aplikasi web menjelaskan bahwa ada beberapa aktor yang terlibat dalam sistem aplikasi web ini, antara lain sebagai berikut:

- 1) *User* sebagai donatur,
- 2) Admin Lembaga sebagai penggalang dana,
- 3) *Super Admin* selaku Yayasan Amal Bakti Nusantara Tegal sebagai pemilik *platform digital* Berbagi Mulia,

- 4) Midtrans *Payment Gateway* sebagai sistem lain untuk metode pembayaran verifikasi otomatis,  
 5) Facebook Ads sebagai sistem lain untuk kebutuhan *digital marketing tools*.

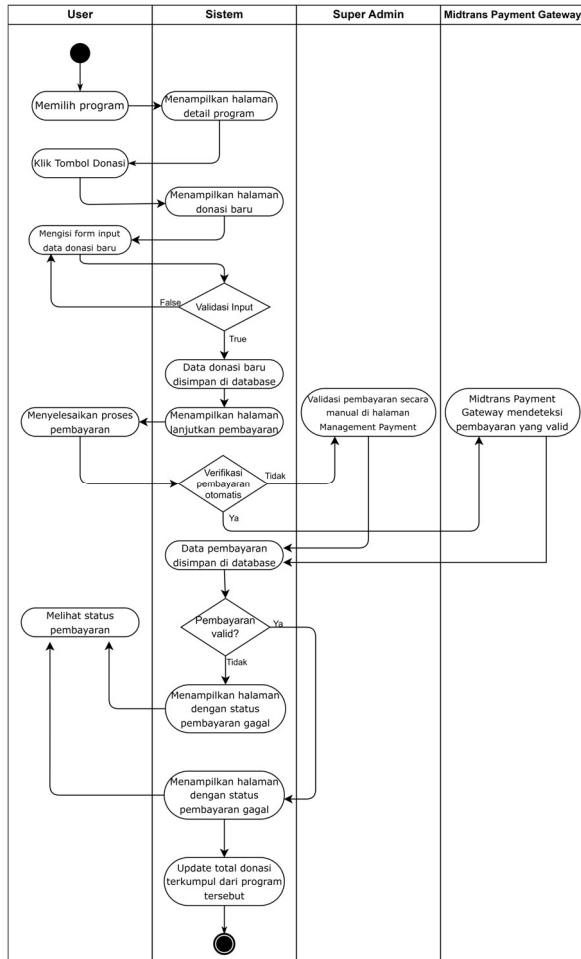
Beberapa *use case* yang dapat digambarkan pada setiap aktor diatas adalah sebagai berikut:



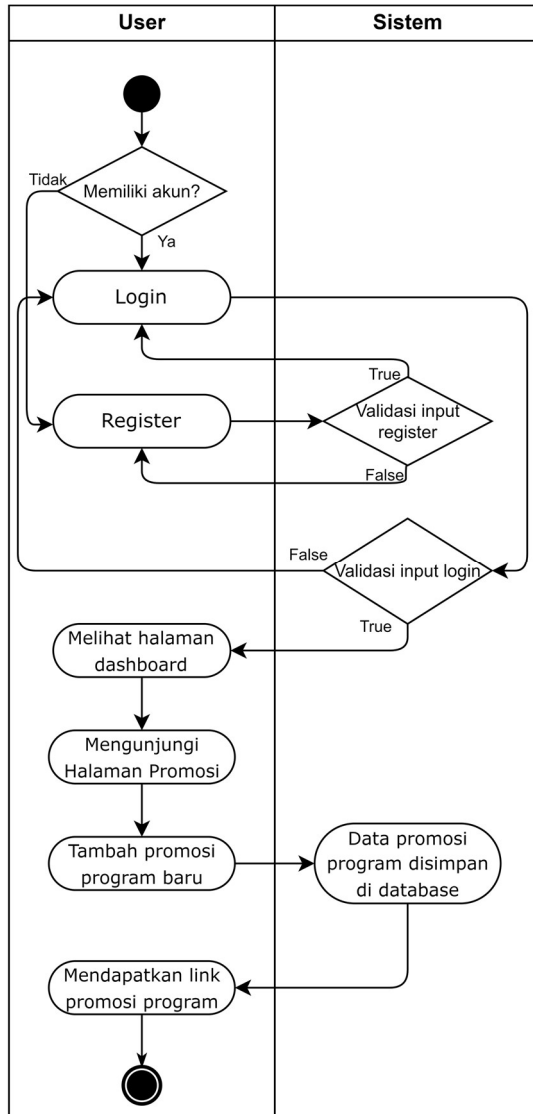
Gambar 4.2 Hasil perancangan *use case diagram*

b. *Activity Diagram*

*Activity diagram* digunakan untuk memodelkan setiap proses pada sebuah sistem dengan hasil sebagai berikut:

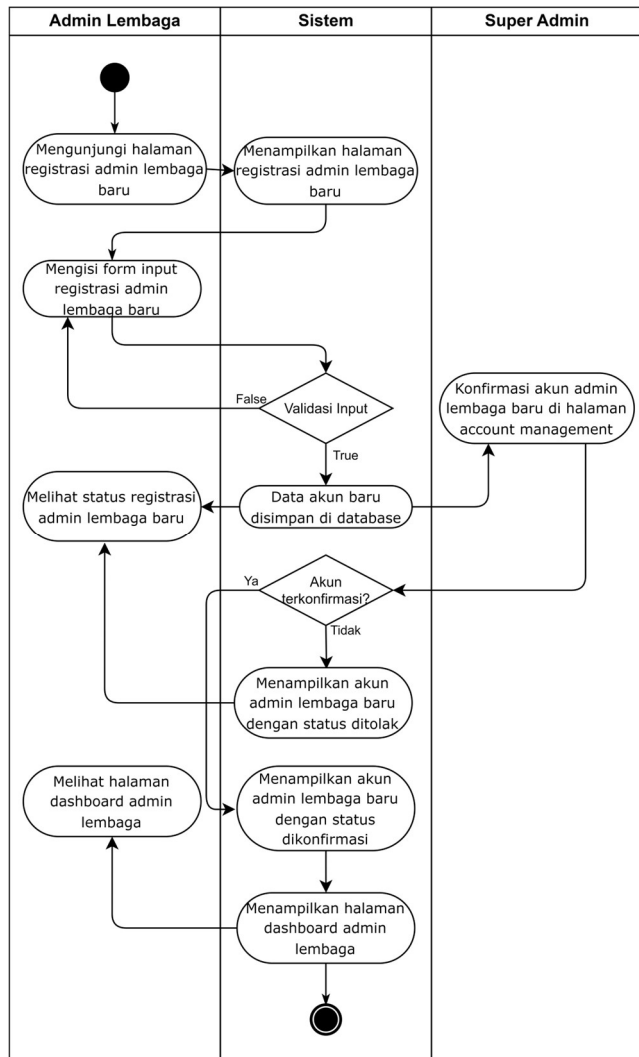


Gambar 4.3 Aktivitas *User* dalam proses donasi

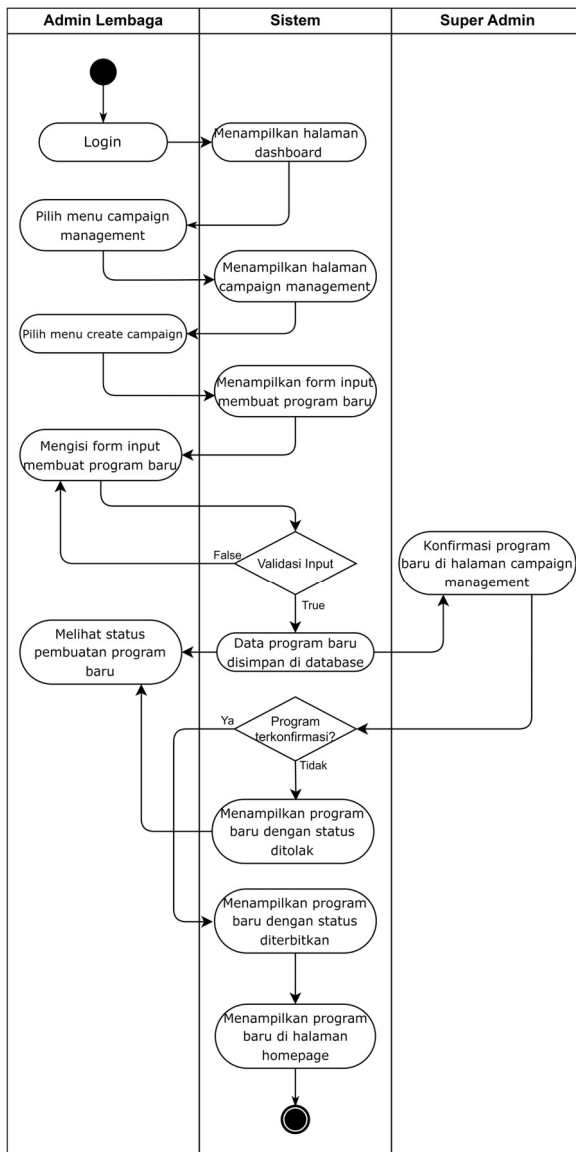


Gambar 4.4 Aktivitas *User* dalam mempromosikan program galang dana

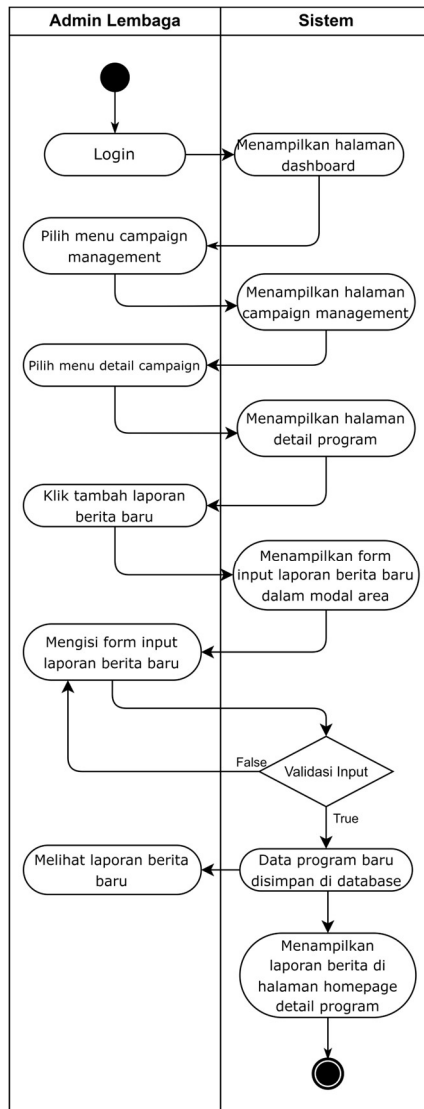




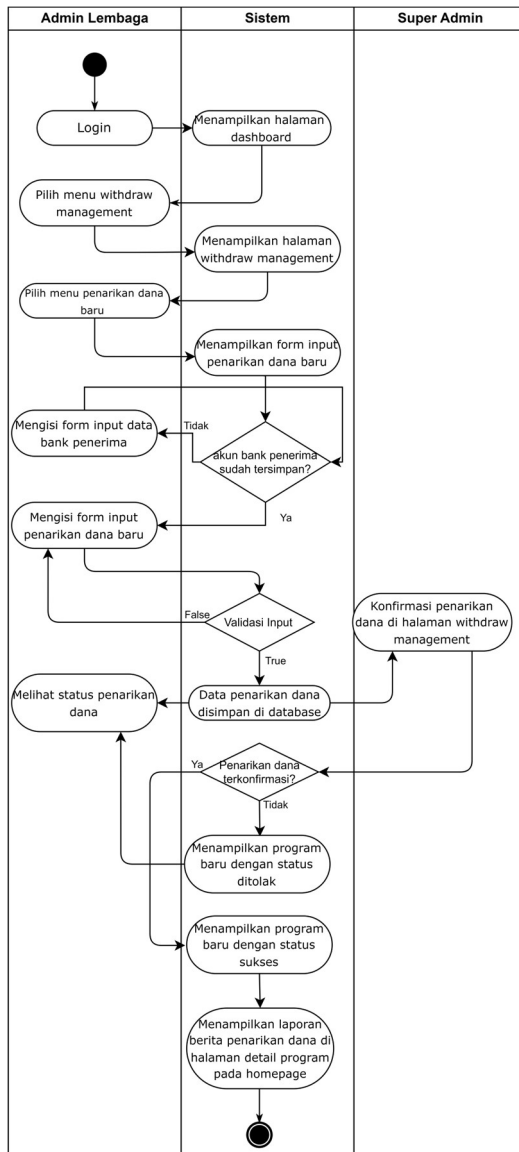
Gambar 4.5 Aktivitas Admin Lembaga dalam proses pendaftaran akun baru



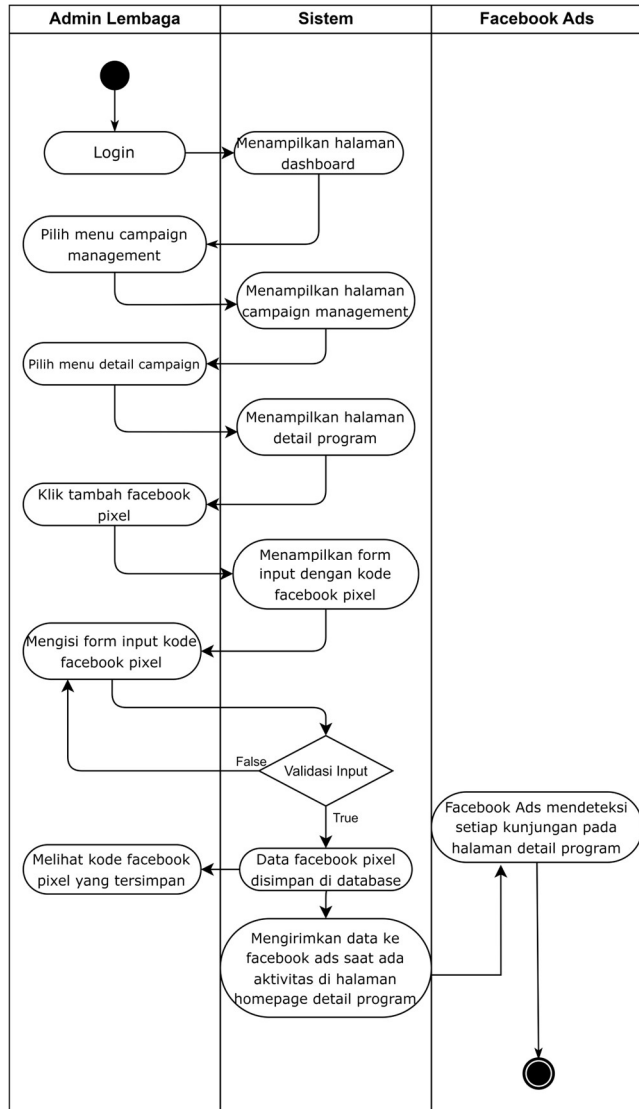
Gambar 4.6 Aktivitas Admin Lembaga dalam mengelola program galang dana



Gambar 4.7 Aktivitas Admin Lembaga dalam mengelola laporan berita galang dana



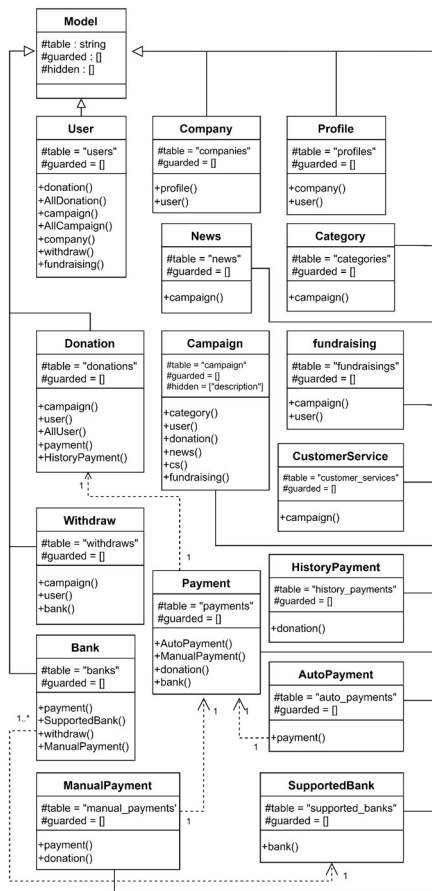
Gambar 4.8 Aktivitas Admin Lembaga dalam proses penarikan dana



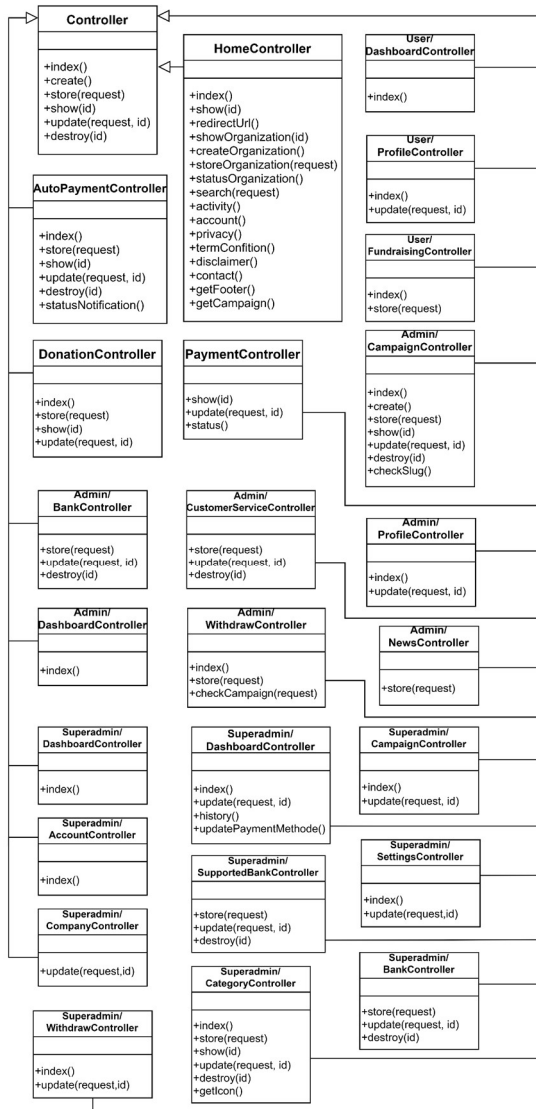
Gambar 4.9 Aktivitas Admin Lembaga dalam mengintegrasikan program galang dana dengan Facebook Ads

c. *Class Diagram*

*Class Diagram* termasuk ke dalam kategori *structural diagram* yang membantu dalam visualisasi kelas dari sebuah sistem perangkat lunak dengan hasil sebagai berikut:



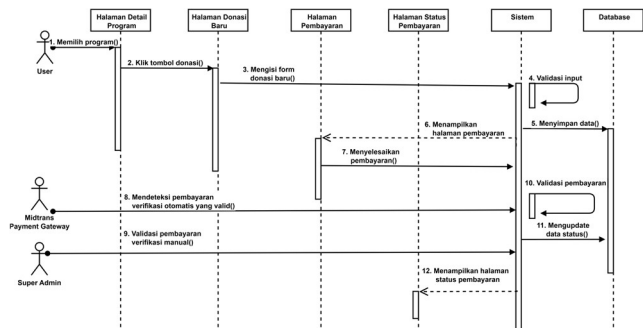
Gambar 4.10 Hasil perancangan *class diagram* pada implementasi *model*



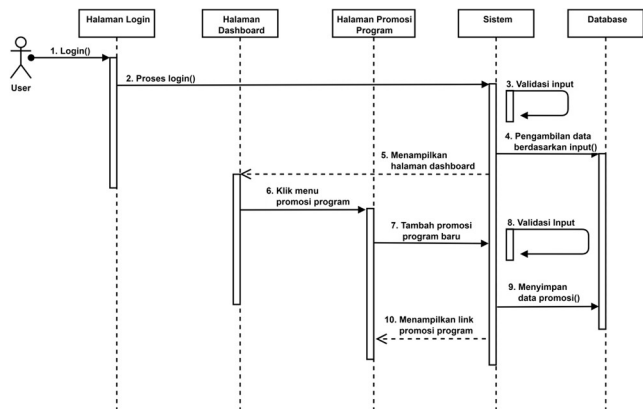
Gambar 4.11 Hasil perancangan *class diagram* pada implementasi *controller*

d. *Sequence Diagram*

*Sequence diagram* termasuk ke dalam kategori *interaction diagram* yang menggambarkan beberapa tahapan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Berikut ini merupakan hasil dari perancangan *sequence diagram* yang telah dibuat:

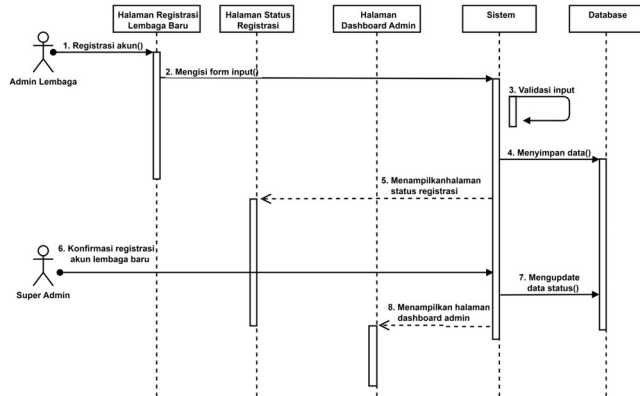


Gambar 4.12 *Sequence diagram* dari proses donasi

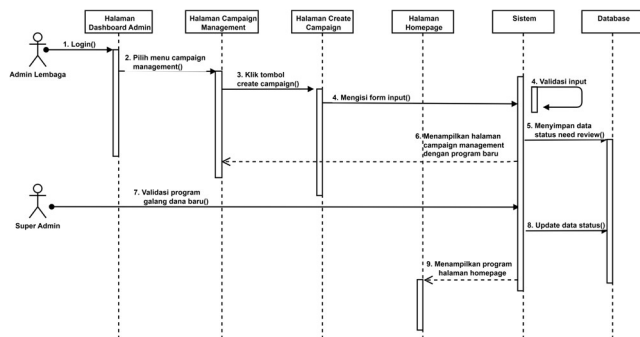


Gambar 4.13 *Sequence diagram* dari proses promosi program oleh User

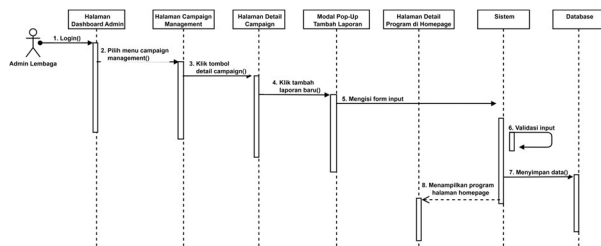




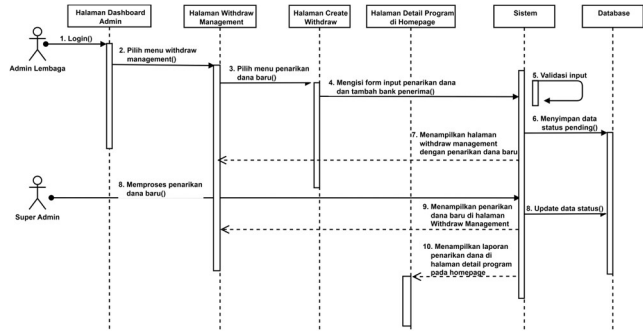
Gambar 4.14 *Sequence diagram* dari proses pendaftaran Admin Lembaga baru



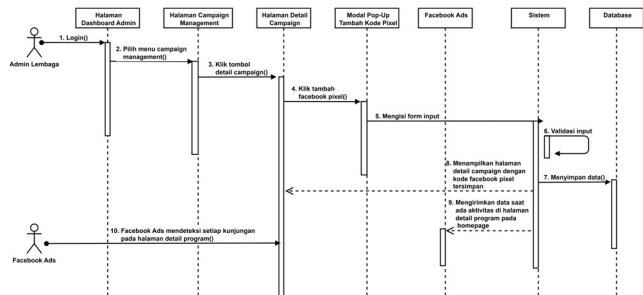
Gambar 4.15 *Sequence diagram* dari proses pengelolaan program galang dana oleh Admin Lembaga



Gambar 4.16 *Sequence diagram* dari proses pengelolaan laporan berita galang dana oleh Admin Lembaga



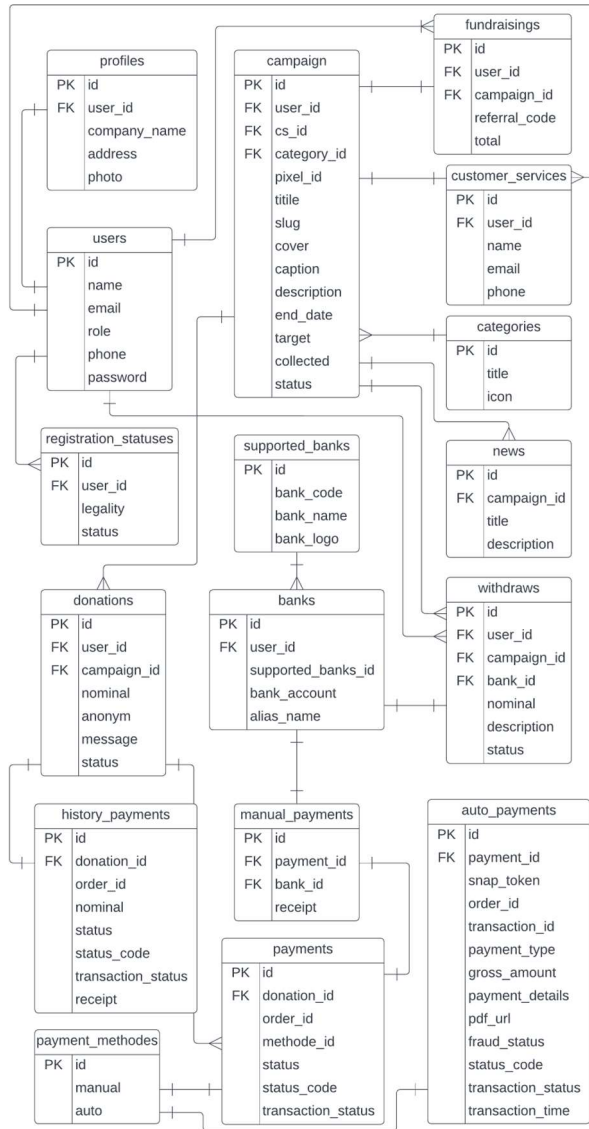
Gambar 4.17 Sequence diagram dari proses penarikan dana oleh Admin Lembaga



Gambar 4.18 Sequence diagram dari proses integrasi Facebook Ads dengan program galang dana

## 2. Entity Relationship Diagram (ERD)

Entity relationship diagram atau ERD digunakan untuk menggambarkan relasi setiap entitas dalam perancangan basis data dengan hasil sebagai berikut:

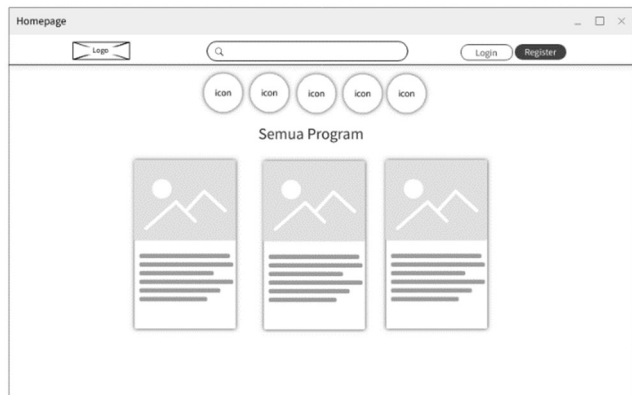


Gambar 4.19 Hasil perancangan *entity relationship diagram* (ERD)

### 3. *Wireframe*

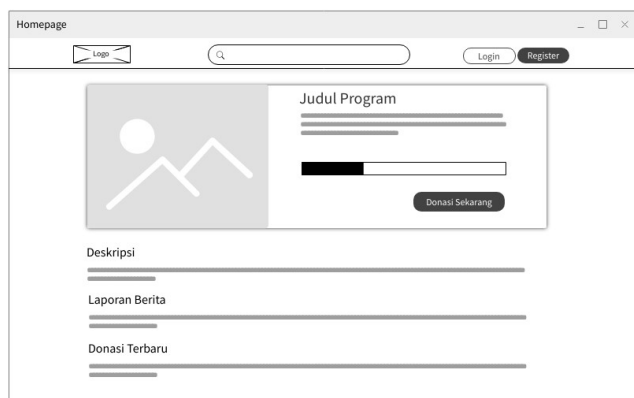
Pembuatan *wireframe* ini merupakan salah satu bentuk perancangan antarmuka dengan hasil sebagai berikut:

#### 1) Halaman *Homepage*



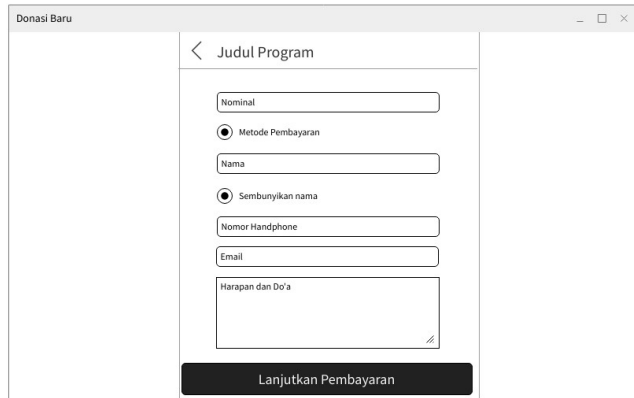
Gambar 4.20 Rancangan tampilan halaman *homepage*

#### 2) Halaman Detail Program



Gambar 4.21 Rancangan halaman detail program

### 3) Halaman Donasi Baru



Donasi Baru

< Judul Program

Nominal

Metode Pembayaran

Nama

Sembunyikan nama

Nomor Handphone

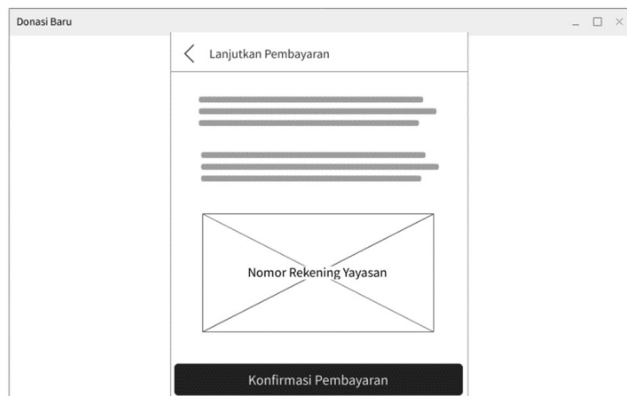
Email

Harapan dan Do'a

Lanjutkan Pembayaran

Gambar 4.22 Rancangan halaman donasi baru

### 4) Halaman Pembayaran



Donasi Baru

< Lanjutkan Pembayaran

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

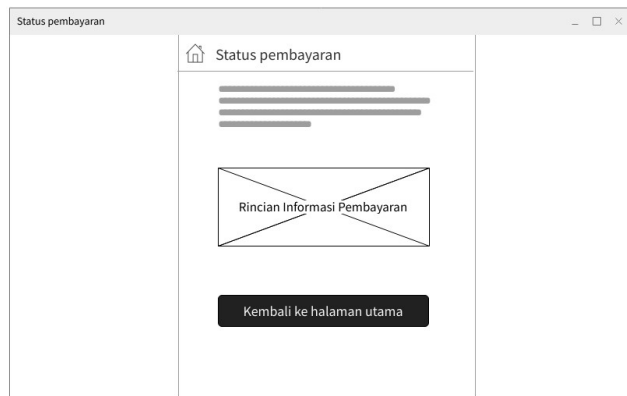
\_\_\_\_\_

Nomor Rekening Yayasan

Konfirmasi Pembayaran

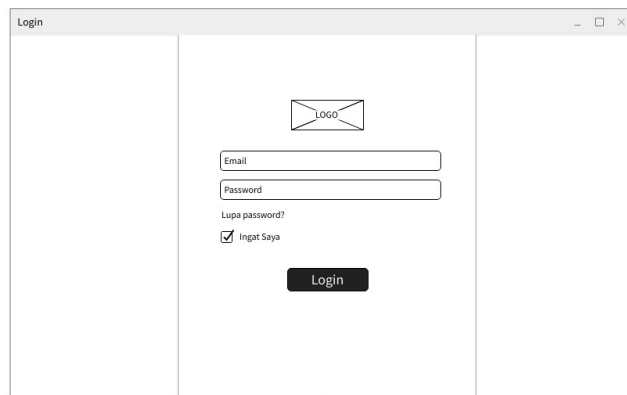
Gambar 4.23 Rancangan halaman pembayaran

## 5) Halaman Status Pembayaran



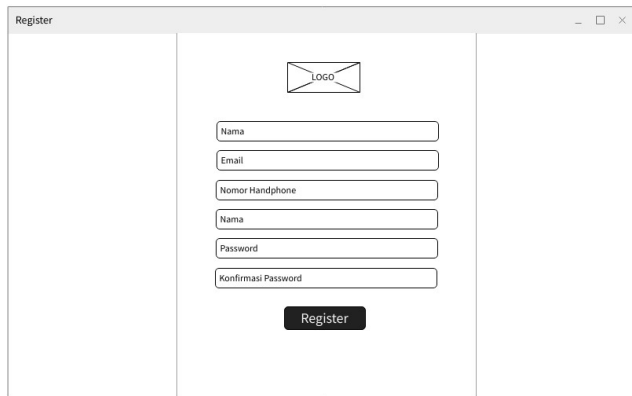
Gambar 4.24 Rancangan status pembayaran

## 6) Halaman *Login*



Gambar 4. 25 Rancangan halaman *login*

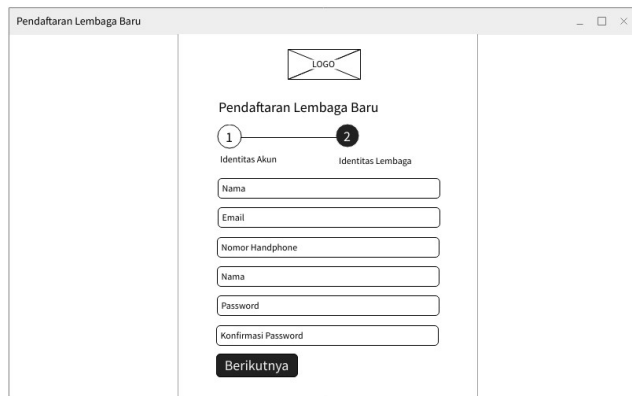
## 7) Halaman Registrasi Akun *User*



The screenshot shows a web browser window titled "Register". At the top center is a placeholder for a logo. Below it is a registration form with the following fields: "Nama", "Email", "Nomor Handphone", "Nama", "Password", and "Konfirmasi Password". A dark "Register" button is positioned at the bottom of the form.

Gambar 4.26 Rancangan halaman registrasi akun *User*

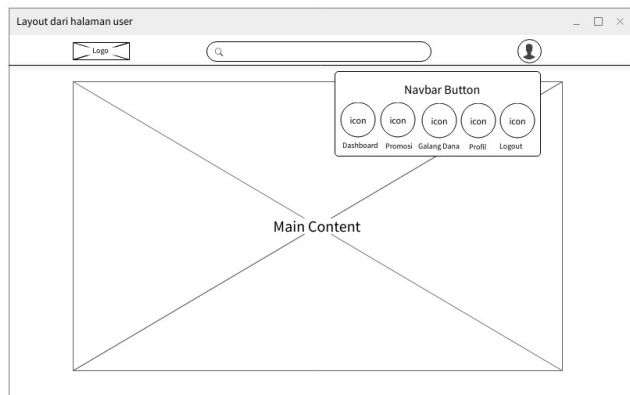
## 8) Halaman Registrasi Akun Admin Lembaga



The screenshot shows a web browser window titled "Pendaftaran Lembaga Baru". At the top center is a placeholder for a logo. Below it is a registration form with a two-step process: "1 Identitas Akun" and "2 Identitas Lembaga". The form includes the following fields: "Nama", "Email", "Nomor Handphone", "Nama", "Password", and "Konfirmasi Password". A dark "Berikutnya" button is positioned at the bottom of the form.

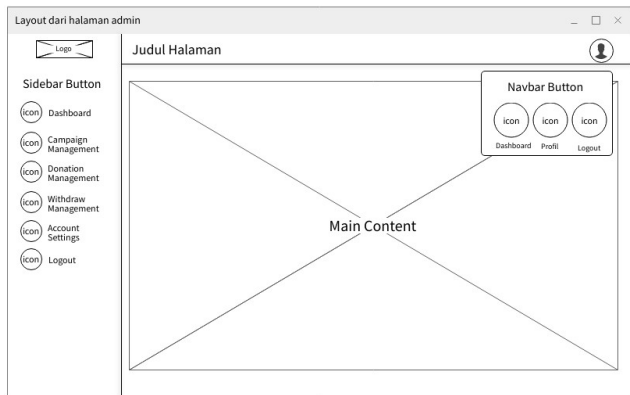
Gambar 4. 27 Rancangan halaman registrasi akun Admin Lembaga

## 9) Halaman Akun *User*



Gambar 4.28 Rancangan halaman dari akun *User*

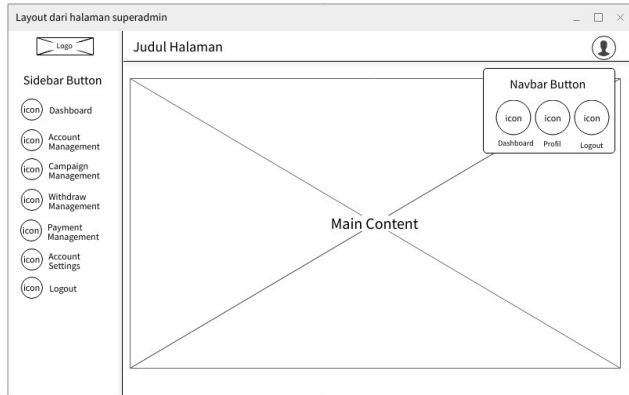
## 10) Halaman Akun Admin Lembaga



Gambar 4.29 Rancangan halaman dari akun Admin Lembaga



## 11) Halaman Akun *Super Admin*



Gambar 4.30 Rancangan halaman dari akun *Super Admin*

### D. Construction of Prototype

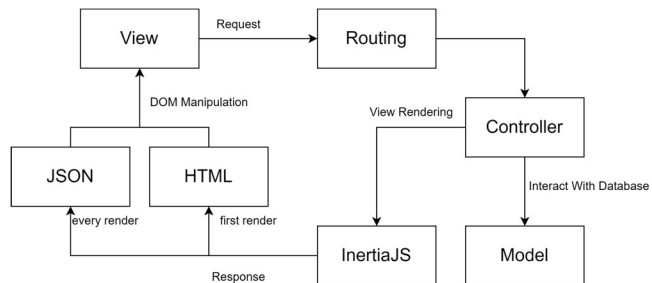
#### 1. Proses Pengkodean

Proses pengkodean dilakukan dengan mengikuti aturan yang dimiliki *framework* Laravel. *Framework* Laravel menggunakan pola desain MVC (*model view controller*) untuk memisahkan beberapa layer yang menyusun sebuah aplikasi web, seperti *model* sebagai *database layer*, *view* sebagai *presentation layer*, dan *controller* sebagai *business logic layer*.

Laravel Breeze memiliki banyak pilihan teknologi yang dapat dipakai sebagai *presentation layer*. Hal tersebutlah yang membedakan Laravel Breeze dengan ekosistem *framework* Laravel lain atau bahkan

*framework* bahasa pemrograman PHP lain, seperti *CodeIgniter*.

Pada penelitian ini, akan digunakan teknologi ReactJS sebagai *presentation layer* dengan InertiaJS sebagai penghubung antara Laravel sebagai *server side* dan ReactJS *client side*. Oleh karena itu, proses pengkodean pada Laravel Breeze memiliki sedikit perbedaan di bagian *presentation layer*. Berikut ini gambaran pola desain MVC yang diterapkan oleh Laravel Breeze:



Gambar 4.31 Pola desain MVC pada Laravel Breeze

Dari gambar diatas, dapat diperoleh proses pengkodean pada penelitian ini dilakukan berdasarkan tahapan sebagai berikut:

a. Pembuatan *Route URL*

Pembuatan *route URL* dilakukan dengan memanfaatkan fitur *resource controller* milik *framework* Laravel yang memberikan

kesederhanaan pada penulisan kode dengan hanya satu baris kode saja dapat yang menghasilkan tujuh perintah *route URL*. *Route URL* yang dihasilkannya tersebut terdiri atas fitur dasar terkait CRUD (*create read update delete*) dari sebuah aplikasi web yang dapat dimodifikasi sesuai dengan kebutuhan dengan mengarahkannya pada setiap *method* bawaan *framework* Laravel di *class controller*. Berikut ini merupakan sampel hasil dari pembuatan *route URL* yang telah dibuat:



```

berbagimula - web.php
1 Route::resource('autopayment', AutoPaymentController::class);
2 Route::resource('donation', DonationController::class)->except('index','destroy','update');
3 Route::resource('payment', PaymentController::class)->except('index','destroy');
4 Route::middleware(['auth','User'])->name('user.-')->group(function () {
5     Route::resource('user', userDashboard::class)->only('index');
6     Route::resource('user/fundraising', userFundraising::class);
7     Route::resource('user/profile', userProfile::class);
8 });
9 Route::middleware(['auth','Admin','Approved'])->name('admin.-')->group(function () {
10    Route::resource('admin', adminDashboard::class)->only('index');
11    Route::resource('admin/bank', adminBank::class)->only('store','update','destroy');
12    Route::resource('admin/campaign', adminCampaign::class)->only('index','create','store','update','destroy');
13    Route::resource('admin/customer-service', adminCustomerService::class)->only('store','update','destroy');
14    Route::resource('admin/news', adminNewsInfo::class)->only('store');
15    Route::resource('admin/profile', adminProfile::class)->only('index','update');
16    Route::resource('admin/withdraw', adminWithdraw::class)->only('index','store');
17 });
18 Route::middleware(['auth','SuperAdmin'])->name('superadmin.-')->group(function () {
19    Route::resource('superadmin', superadminDashboard::class)->only('index');
20    Route::resource('superadmin/category', superAdminCategory::class)->except('show');
21    Route::resource('superadmin/supported-bank', superAdminSuppBank::class)->only('store','update','destroy');
22    Route::resource('superadmin/bank', superAdminBank::class)->except('show');
23    Route::resource('superadmin/account', superAdminAccount::class)->only('index');
24    Route::resource('superadmin/account', superAdminAccount::class)->only('index','update');
25    Route::resource('superadmin/company', superAdminCompany::class)->only('update');
26    Route::resource('superadmin/payment', superAdminPayment::class)->only('index','update','destroy');
27    Route::resource('superadmin/campaign', superAdminCampaign::class)->only('index','update','show');
28    Route::resource('superadmin/withdraw', superAdminWithdraw::class)->only('index','update','destroy');
29    Route::resource('superadmin/settings', superAdminSettings::class)->only('index','update','destroy');
30 });

```

Gambar 4.32 Sampel hasil penulisan kode *route URL*

## b. Pembuatan *Controller*

Pembuatan *controller* ini memiliki keterkaitan dengan *route URL* yang telah dibuat. *Route URL* yang menggunakan fitur *resource controller*, secara otomatis akan mengarahkan *URL* tersebut menuju *method* bawaan milik *framework* Laravel pada setiap *class controller* yang telah didefinisikan. Berikut ini merupakan sampel dari hasil pembuatan *controller* yang telah dibuat:

```

1 class CampaignController extends Controller{
2     public function index() {
3         $campaign = Campaign::where('user_id',Auth::user()->id)->get();
4         $data = ['title'=>'Campaign Management', 'campaign'=>$campaign];
5         return Inertia::render('Admin/Campaign/Index', $data);
6     }
7     public function store(Request $request){
8         $validatedData = $request->validate([
9             'title' => 'required|max:255',
10            'target' => 'required|numeric',
11            'end_date' => 'required',
12            'description' => 'required',
13        ]);
14        $store = Campaign::create($validatedData);
15        if ($store){
16            return redirect()->back()->with('success','Buat Campaign Baru Berhasil');
17        } else {
18            return redirect()->back()->with('error','Buat Campaign Baru Gagal');
19        }
20    }
21    public function update(Request $request, $id){
22        $campaign = Campaign::where('id',$id)->first();
23        $validatedData['status'] = 1;
24        $update = $campaign->update($validatedData);
25        if ($update){
26            return redirect()->back()->with('success','Edit Campaign Berhasil');
27        } else {
28            return redirect()->back()->with('error','Edit Campaign Gagal');
29        }
30    }
31 }

```


Gambar 4.33 Sampel hasil penulisan kode *controller* pada fitur pengelolaan program oleh Admin Lembaga

c. Pembuatan *Model*

Pembuatan *model* ini dilakukan untuk merepresentasikan tabel di *database* agar dapat mempermudah interaksinya saat menggunakan fitur Eloquent ORM milik *framework* Laravel. Fitur tersebut berbasis ORM atau *Object Relational Mapping* yang memungkinkan proses manipulasi data dari *database* dengan paradigma berorientasi objek.

Saat menggunakan ORM, pengembang tidak perlu lagi menuliskan perintah bahasa pemrograman SQL dalam proses manipulasi data. Salah satu contohnya adalah pada Gambar 4.3, terdapat *method* bernama *update* berisi variabel *campaign* sebagai objek yang merepresentasikan *class model Campaign*. Saat ingin melakukan *update* data pada *database*, pengembang hanya perlu memanggil *method* ORM bernama *update* pada objek *campaign* dengan parameter yang berisi data baru. Penggunaan fitur Eloquent ORM ini sangat membantu para pengembang dalam membuat aplikasi berbasis web karena kemudahan yang ditawarkannya. Berikut ini

merupakan sampel dari hasil pembuatan *model* yang telah dibuat:

The image shows a screenshot of a code editor window titled 'berbagimulia - Campaign.php'. The code defines a PHP class named 'Campaign' that extends a 'Model' class. The class contains several protected and public properties and methods. The protected properties are '\$guarded' (an empty array), '\$table' (set to 'campaign'), and '\$hidden' (set to an array containing 'description'). The public methods include 'category()', 'donation()', 'news()', 'user()', 'cs()', 'withdraw()', and 'fundraising()', each returning a query object using methods like 'belongsToMany' or 'hasOne' from the Laravel ORM. The code is numbered from 1 to 26.

```
1 class Campaign extends Model{
2     protected $guarded = [];
3     protected $table= 'campaign';
4     protected $hidden = ['description'];
5     public function category(){
6         return $this->belongsTo(Category::class);
7     }
8     public function donation(){
9         return $this->hasMany(Donation::class,'campaign_id','id');
10    }
11    public function news(){
12        return $this->hasMany(News::class,'campaign_id','id');
13    }
14    public function user(){
15        return $this->belongsTo(User::class, 'user_id', 'id');
16    }
17    public function cs(){
18        return $this->hasOne(CustomerService::class, 'id', 'cs_id');
19    }
20    public function withdraw(){
21        return $this->hasOne(Withdraw::class,'campaign_id', 'id');
22    }
23    public function fundraising(){
24        return $this->hasMany(Fundraising::class,'campaign_id','id');
25    }
26 }
```

Gambar 4.34 Sampel hasil penulisan kode pada *model Campaign*

#### d. Pembuatan *View*

Pembuatan *view* dilakukan dengan menggunakan *library* ReactJS sebagai *client side* dari aplikasi web. Seperti yang terlihat pada

Gambar 4.1, *output* dari *controller* dapat berupa *view rendering* menggunakan InertiaJS.

Ketika aplikasi web pertama kali di muat, *server* akan mengembalikan dokumen *HTML* secara lengkap dengan aset yang dibutuhkan seperti *CSS* dan *Javascript*. Respons *HTML* ini berisikan *tag div* pada elemen *body HTML* dengan *id* tertentu yang digunakan sebagai titik pemasangan dalam proses manipulasi *document object model (DOM)* untuk menciptakan *single page application*. Untuk keperluan pengolahan data, *server* juga akan mengembalikan sekumpulan data dalam *format JSON* yang akan di *render* setiap kali ada interaksi dengan sistem aplikasi web. Isi dari sekumpulan data tersebut antara lain sebagai berikut:

- a. *Component*, berisikan nama komponen dari halaman *Javascript* yang di *render*,
- b. *Props*, berisikan data yang dikembalikan oleh *controller*,
- c. *URL*, berisikan alamat *URL* halaman,
- d. *Versi*, berisikan versi aset saat ini.

Berikut ini merupakan sampel dari hasil pembuatan view menggunakan *library* ReactJS yang telah dibuat:

```
berbagimulia - app.jsx
1 import "../css/app.css";
2 import React from "react";
3 import { render } from "react-dom";
4 import { createInertiaApp } from "@inertiajs/inertia-react";
5 import { InertiaProgress } from "@inertiajs/progress";
6 import { resolvePageComponent } from "laravel-vite-plugin/inertia-helpers";
7
8 const appName =
9   window.document.getElementsByTagName("title")[0]?.innerText ||
10   "Berbagi Mulia";
11
12 createInertiaApp({
13   id: "app",
14   title: (title) => `${title} - ${appName}`,
15   resolve: (name) =>
16     resolvePageComponent(
17       `./Pages/${name}.jsx`,
18       import.meta.glob("./Pages/**/*.jsx")
19     ),
20   setup({ el, App, props }) {
21     return render(<App {...props} />, el);
22   },
23 });
24
25 InertiaProgress.init({ color: "#312E81" });
```

Gambar 4.35 Kode ReactJS untuk koneksi melalui InertiaJS ke *tag HTML* dengan *id* bernama *app*



```

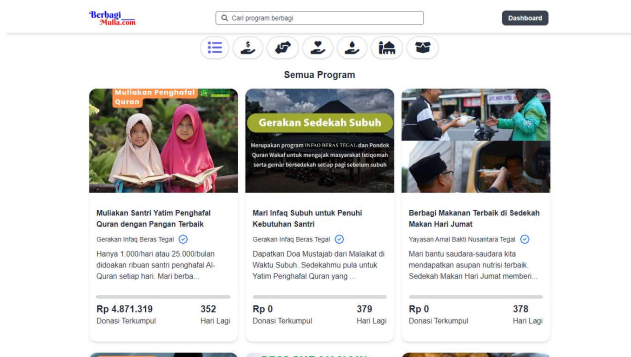
berbagimulia - app.blade.php
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
7     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
8     <title inertia>{{ config('app.name', 'Berbagi Mulia') }}</title>
9     <link rel="preload" as="image" href="/img/logo.webp">
10    @viteReactRefresh
11    @vite('resources/js/app.jsx')
12    @inertiaHead
13 </head>
14
15 <body class="font-sans antialiased">
16     <div id="app" data-page="{{ json_encode($page) }}"></div>
17 </body>
18
19 </html>
20

```

Gambar 4.36 Kode dokumen *HTML* beserta *tag div* dengan *id* bernama *app*

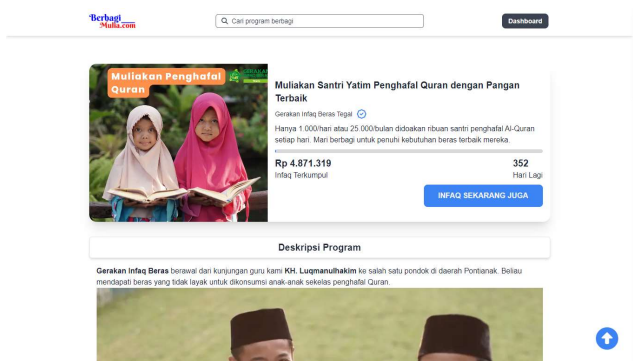
## 2. Hasil Tampilan Aplikasi Web

### a. Halaman *Homepage*



Gambar 4.37 Hasil tampilan halaman *homepage*

## b. Halaman Detail Program



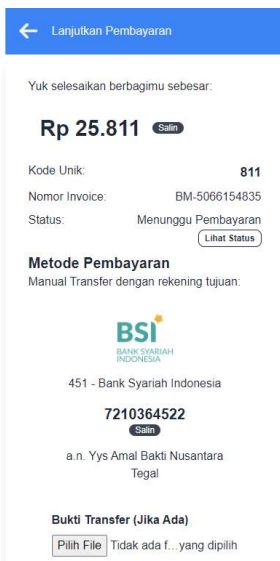
Gambar 4.38 Hasil tampilan halaman detail program

## c. Halaman Donasi Baru

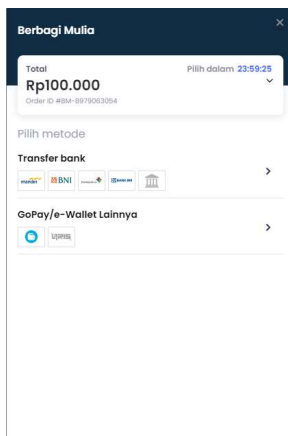
The screenshot shows a mobile donation form. At the top, there is a blue header with a back arrow and the text 'Mulai Santri Yatim Penghafal Quran dengan Pangan Terbaik'. Below the header, the text reads: 'Yuk, isi data donasimu sebelum lanjut pembayaran' and 'Jika ada pertanyaan lain, bisa hubungi kami lewat tombol whatsapp di bawah, ya.' Below this is a WhatsApp icon and the number '089512342344 (CS Mamas)'. The 'Pilihan Nominal' section has radio buttons for 'Rp 1.000.000', 'Rp 500.000', 'Rp 250.000', 'Rp 100.000', 'Rp 50.000', and 'Lainnya'. The 'Metode Pembayaran' section has radio buttons for 'Bank Virtual Account dan E-Wallet' (selected) and 'BSI 7210364522 Yys Amal Bakti Nusantara Tegah'. Below this is a 'Nama' field, a 'Sembunyikan Nama (Hamba Allah)' toggle, a 'Nomor Whatsapp' field, and a blue button that says 'LANJUT PEMBAYARAN'.

Gambar 4.39 Hasil tampilan halaman donasi baru

#### d. Halaman Pembayaran

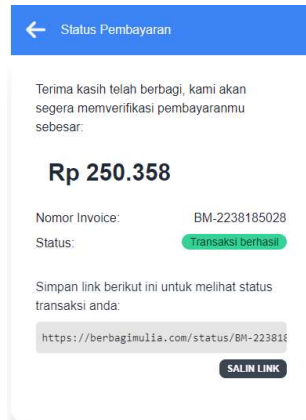


Gambar 4.40 Hasil tampilan halaman pembayaran verifikasi manual



Gambar 4.41 Hasil tampilan halaman pembayaran verifikasi otomatis

### e. Halaman Status Pembayaran

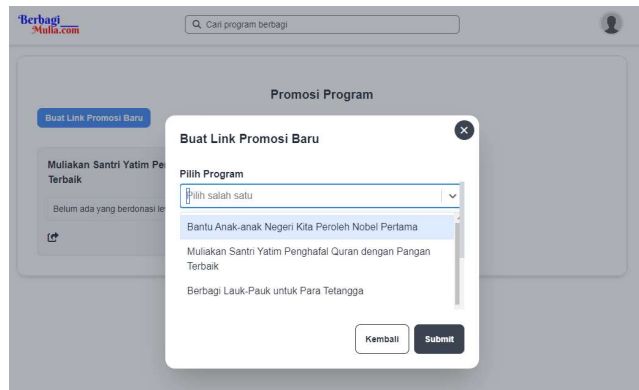


Gambar 4.42 Hasil tampilan halaman status pembayaran

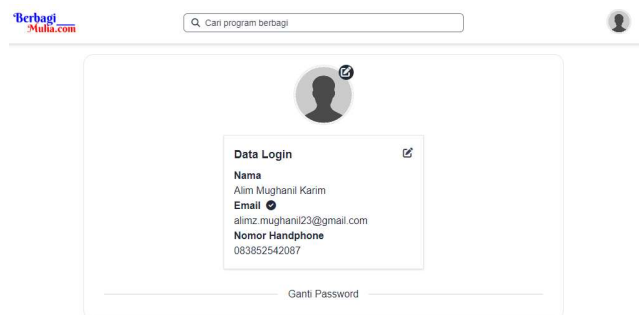
### f. Halaman Akun *User*



Gambar 4.43 Hasil tampilan halaman *dashboard* akun *User*

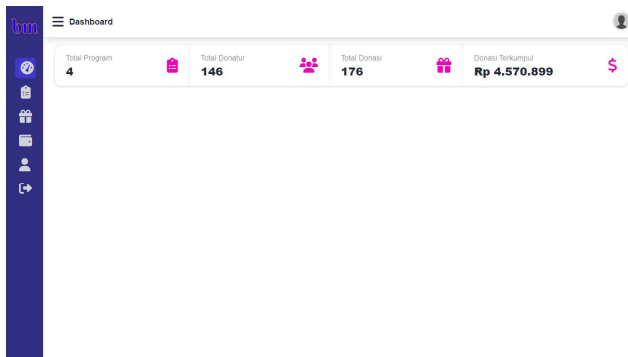
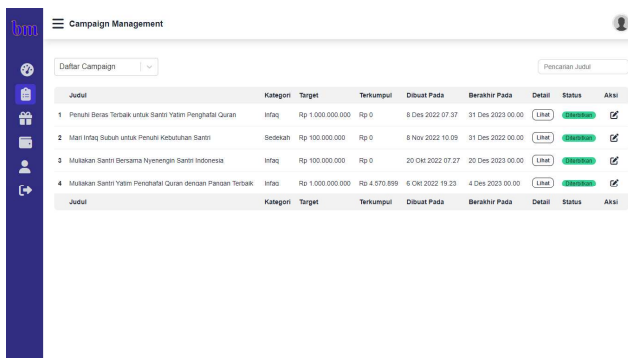


Gambar 4.44 Hasil tampilan halaman pembuatan promosi program baru



Gambar 4.45 Hasil tampilan halaman profil akun *User*

## g. Halaman Akun Admin Lembaga

Gambar 4.46 Hasil tampilan halaman *dashboard* akun Admin Lembaga

Gambar 4.47 Hasil tampilan halaman pengelolaan program

**Campaign Management**

Buat Campaign Baru

Judul

Slug  
<http://berdagimika.com/program/>

\*Slug bisa dimodifikasi dan hanya dapat dibuat sekali

Kategori: Customer Service  
 Pilih Salah Satu: Oportional

Target: Rp. Berakhir Pada: 10.000.000

Foto Cover  
 [Pilih File] Tidak ada file yang dipilih

Deskripsi

Gambar 4.48 Hasil tampilan halaman pembuatan program baru

**Campaign Management**

Daftar Campaign

Mullakan Santri Yatim Penghafal Quran dengan Pangan Terbaik

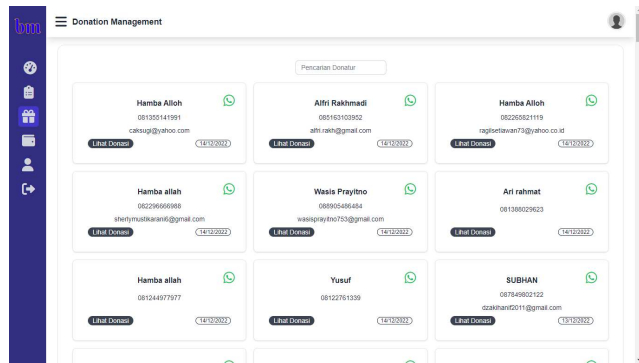
Facebook Pixel  
 Pixel ID 471994844696473

Laporan Berita  
 [Buat Laporan] [Lihat Semua]

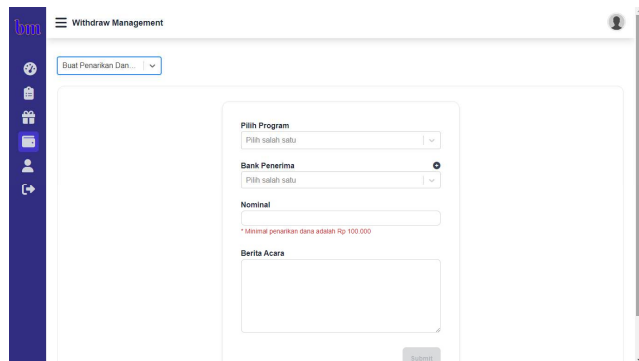
Teks Publikasi  
 [Keterangan Singkat] [Deskripsi]

Total Donasi	176
Donasi Berhasil	81
Dana Berkumpul	Rp 4.570.899
Dana Target	Rp 1.000.000.000
Berakhir Pada	4 Des 2023 00:00
Status	🟢

Gambar 4.49 Hasil tampilan halaman pengelolaan laporan berita dan integrasi Facebook Ads dengan kode Pixel

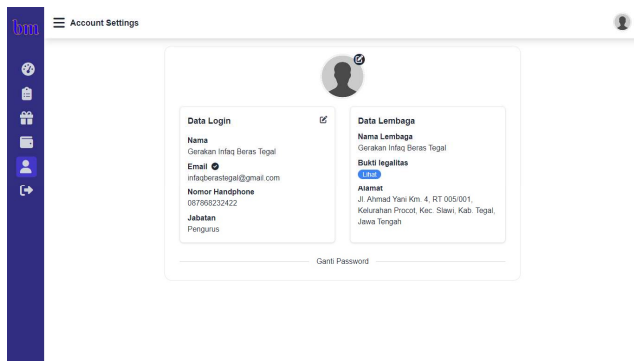


Gambar 4.50 Hasil tampilan halaman data donasi



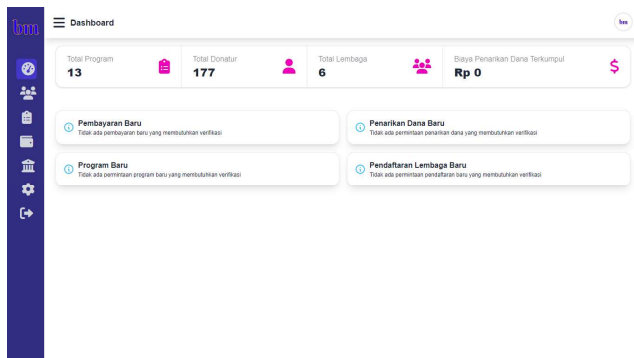
Gambar 4.51 Hasil tampilan halaman penarikan dana baru





Gambar 4.52 Hasil tampilan halaman profil Admin Lembaga

#### h. Halaman Akun *Super Admin*



Gambar 4.53 Hasil tampilan halaman *dashboard* akun *Super Admin*

The screenshot shows the 'Account Management' interface for 'Lembaga'. It features a sidebar with navigation icons and a main content area with a search filter and a table of organization records.

Nama Lembaga	Legalitas	Data Pengurus	Dibuat Pada	Status	Aksi
1 Berbagi Mula	<a href="#">Lihat</a>	<a href="#">Lihat</a>	6 Okt 2022 09:55	<span style="color: green;">✔</span>	No Action
2 Gerakan Infaq Beras Tegat	<a href="#">Lihat</a>	<a href="#">Lihat</a>	6 Okt 2022 13:27	<span style="color: green;">✔</span>	No Action
3 KHO	<a href="#">Lihat</a>	<a href="#">Lihat</a>	6 Okt 2022 09:55	<span style="color: green;">✔</span>	No Action
4 Yayasan Amat Bakti Nusantara Tegat	<a href="#">Lihat</a>	<a href="#">Lihat</a>	6 Okt 2022 09:55	<span style="color: green;">✔</span>	No Action
5 Komunitas Hobi Sedekah	<a href="#">Lihat</a>	<a href="#">Lihat</a>	24 Okt 2022 13:42	<span style="color: green;">✔</span>	No Action
6 Gerakan Infaq Beras Semarang	<a href="#">Lihat</a>	<a href="#">Lihat</a>	7 Des 2022 07:54	<span style="color: green;">✔</span>	No Action
7 Masjid Huseinie	<a href="#">Lihat</a>	<a href="#">Lihat</a>	8 Des 2022 11:17	<span style="color: green;">✔</span>	No Action

Gambar 4.54 Hasil tampilan halaman pengelolaan akun Admin Lembaga

The screenshot shows the 'Account Management' interface for 'User'. It features a sidebar with navigation icons and a main content area with a search filter and a table of user records.

Nama	No. Handphone	Email	Grade	Dibuat Pada	Profil	Aksi
1 Alim Mughani Karm	08385242087	alimz.mughani123@gmail.com	<a href="#">Edit User</a>	6 Okt 2022 13:23	<a href="#">Lihat</a>	<a href="#">👤</a>
2 Marbot Masjid Hijrah	09951022806	hobosedekahindonesia@gmail.com	<a href="#">Edit User</a>	6 Okt 2022 18:44	<a href="#">Lihat</a>	<a href="#">👤</a>
3 Sultan Ahmet	08522222222	zaid.102.85@gmail.com	<a href="#">Edit User</a>	10 Okt 2022 18:37	<a href="#">Lihat</a>	<a href="#">👤</a>
4 Berbagi aja	08116644522	qori@gmail.com	<a href="#">Edit User</a>	10 Okt 2022 18:41	<a href="#">Lihat</a>	<a href="#">👤</a>
5 Jaki Umam	05222222222	forviana@gmail.com	<a href="#">Edit User</a>	16 Okt 2022 06:59	<a href="#">Lihat</a>	<a href="#">👤</a>
6 Cuyin	08322878113	Nurokhtoyim@gmail.com	<a href="#">Edit User</a>	19 Okt 2022 07:45	<a href="#">Lihat</a>	<a href="#">👤</a>
7 Alim Mughani	087796584465	alim.mughani123@gmail.com	<a href="#">Edit User</a>	19 Okt 2022 13:27	<a href="#">Lihat</a>	<a href="#">👤</a>
8 Sabaa	081212684011	tazakaa0@yahoo.com	<a href="#">Edit User</a>	27 Okt 2022 04:43	<a href="#">Lihat</a>	<a href="#">👤</a>
9 Jaki Umam	081111122222	muhammad@gmail.com	<a href="#">Edit User</a>	27 Okt 2022 19:59	<a href="#">Lihat</a>	<a href="#">👤</a>
10 Marbot Masjid	08212345678	mayksho@gmail.com	<a href="#">Edit User</a>	27 Okt 2022 20:08	<a href="#">Lihat</a>	<a href="#">👤</a>
11 Mbsder	08536599645		<a href="#">Edit User</a>	4 Des 2022 10:58	<a href="#">Lihat</a>	<a href="#">👤</a>

Gambar 4.55 Hasil tampilan halaman pengelolaan akun User

**Campaign Management**

Daftar Campaign

Judul	Kategori	Terkumpul	Target	Berakhir	Detail	Status	Aksi
1. Bantu Anak-anak Negeri Kita Peroleh Nobel Perdamaian	Vakaf	Rp 0	Rp 100.000.000	31 Des 2025 00.00	<a href="#">LINE</a>	Completed	No Action
2. Mulakan Saati Yatim Penghafal Quran dengan Pangan Terbaik	Infraq	Rp 4.570.899	Rp 1.000.000.000	4 Des 2023 00.00	<a href="#">LINE</a>	Completed	No Action
3. Berbagi Lauk-Pauk untuk Pina Terlanta	Pangan	Rp 0	Rp 30.000.000	31 Des 2024 23.59	<a href="#">LINE</a>	Completed	No Action
4. Pemuli Beraq Terbaik untuk Saati Yatim Penghafal Quran	Infraq	Rp 0	Rp 1.000.000.000	31 Des 2023 00.00	<a href="#">LINE</a>	Completed	No Action
5. Berbagi Muahaf'Al-Quran untuk Pevanben dan Madrasah	Vakaf	Rp 0	Rp 100.000.000	23 Des 2023 00.00	<a href="#">LINE</a>	Completed	No Action
6. Mulakan Saati Bersama Nyereingin Saati Indonesia	Infraq	Rp 0	Rp 100.000.000	20 Des 2023 00.00	<a href="#">LINE</a>	Completed	No Action
7. Berbagi Makanan Terbaik di Sodekah Makan Hari Jumad	Pangan	Rp 0	Rp 10.000.000	30 Des 2022 00.00	<a href="#">LINE</a>	Completed	No Action
8. Pengadaan Ambulance Gratis untuk Warga Desa	Infraq	Rp 0	Rp 20.000.000	31 Des 2022 00.00	<a href="#">LINE</a>	Completed	No Action
9. Bangun Pusat Peradaban Masjid KH0 Tegaklandong	Vakaf	Rp 0	Rp 5.000.000.000	20 Des 2025 00.00	<a href="#">LINE</a>	Completed	No Action
10. Yik Bangun Kembali Masjid Jakarta Islam Center yang Terbakar	Vakaf	Rp 0	Rp 100.000.000	30 Des 2022 00.00	<a href="#">LINE</a>	Completed	No Action
11. Mari Infraq Subuh untuk Pemulu Kebutuhan Saati	Sedekah	Rp 0	Rp 100.000.000	31 Des 2022 00.00	<a href="#">LINE</a>	Completed	No Action
12. Gerak Cepat Gempa Cianjur!!	Bencana Alam	Rp 0	Rp 100.000.000	31 Des 2022 00.00	<a href="#">LINE</a>	Completed	No Action
13. Pemuli Beraq Terbaik Saati Penghafal Quran Area Semarang dik	Infraq	Rp 0	Rp 1.000.000.000	7 Des 2023 00.00	<a href="#">LINE</a>	Completed	No Action

Gambar 4.56 Hasil tampilan halaman pengelolaan program galang dana

**Campaign Management**

Daftar Kategori

Buat Kategori Baru

Sedekah 1 Campaign	Bencana Alam 1 Campaign	Komersialisasi 0 Campaign	Infraq 4 Campaign	Ifta' 4 Campaign
Pangan 2 Campaign				

Gambar 4.57 Hasil tampilan halaman pengelolaan kategori program galang dana

**Withdraw Management**

Daftar Penarikan Dana

Program	Pengalangan Dana	Nominal	Fee	Dibuat	Detail Bank	Status	Aksi
1. Mulakan Sami Yatim Penghafal Quran dengan Pangan Terbaik	Intbaq Beras Tegak	Rp 750.000	Rp 40.000	17 Des 2022 22:14	<a href="#">LIHAT</a>	<span style="color: green;">Success</span>	<span>✓</span>
2. Mulakan Sami Yatim Penghafal Quran dengan Pangan Terbaik	Intbaq Beras Tegak	Rp 95.000	Rp 5.000	13 Des 2022 23:45	<a href="#">LIHAT</a>	<span style="color: green;">Success</span>	No Action
3. Mulakan Sami Yatim Penghafal Quran dengan Pangan Terbaik	Intbaq Beras Tegak	Rp 95.000	Rp 5.000	13 Des 2022 23:58	<a href="#">LIHAT</a>	<span style="color: red;">Failed</span>	No Action

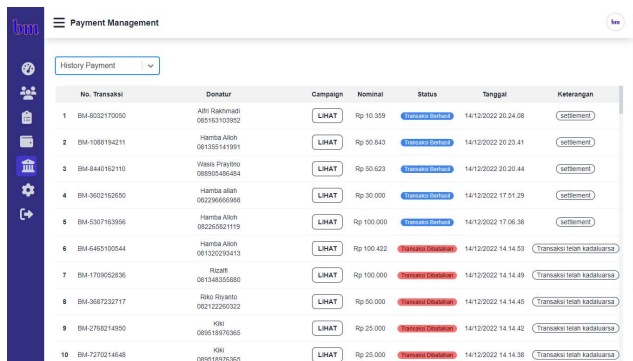
Gambar 4.58 Hasil tampilan halaman pengelolaan penarikan dana

**Payment Management**

Payment Request

No. Transaksi	Donatur	Campaign	Nominal	Rincian Pembayaran	Waktu	Status	Aksi
1. BMA-06122020834	Sofiani / rum387 0802090346337	<a href="#">LIHAT</a>	Rp 50.000	Automatic Payment	13/12/22 20:34	<span style="color: green;">Monevnya Berhasil</span>	No Action
2. BMA-3514190009	Hamba Allah 081277564703	<a href="#">LIHAT</a>	Rp 25.000	Automatic Payment	13/12/22 19:09	<span style="color: green;">Monevnya Berhasil</span>	No Action
3. BMA-1142184015	Evi 085732465592	<a href="#">LIHAT</a>	Rp 25.000	Automatic Payment	13/12/22 18:10	<span style="color: green;">Monevnya Berhasil</span>	No Action
4. BMA-7818174937	Titi Suharti 082224977819	<a href="#">LIHAT</a>	Rp 100.000	Automatic Payment	13/12/22 17:37	<span style="color: green;">Monevnya Berhasil</span>	No Action
5. BMA-6483100544	Hamba Allah 081320293413	<a href="#">LIHAT</a>	Rp 100.422	Belum ada data	13/12/22 10:44	<span style="color: red;">Monevnya Berhasil</span>	<span>✗</span>
6. BMA-1709052836	RizalR 081348305880	<a href="#">LIHAT</a>	Rp 100.000	Automatic Payment	13/12/22 05:36	<span style="color: red;">Monevnya Berhasil</span>	<span>✗</span>
7. BMA-3687232717	Riko Riyanto 081320905932	<a href="#">LIHAT</a>	Rp 50.000	Automatic Payment	12/12/22 23:17	<span style="color: red;">Monevnya Berhasil</span>	<span>✗</span>
8. BMA-2768214959	Kiki 089518976365	<a href="#">LIHAT</a>	Rp 25.000	Automatic Payment	12/12/22 21:50	<span style="color: red;">Monevnya Berhasil</span>	<span>✗</span>
9. BMA-7270214648	Kiki 089518976365	<a href="#">LIHAT</a>	Rp 25.000	Automatic Payment	12/12/22 21:48	<span style="color: red;">Monevnya Berhasil</span>	<span>✗</span>

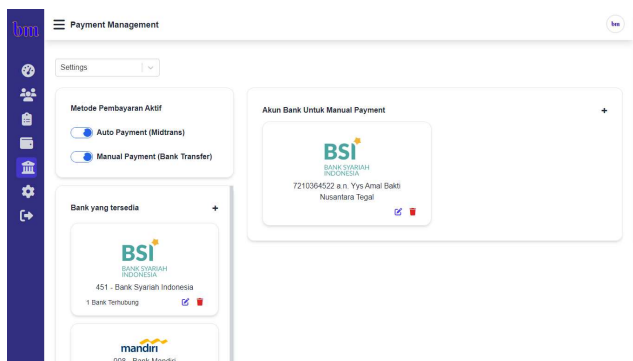
Gambar 4.59 Hasil tampilan halaman permintaan pembayaran baru



The screenshot displays the 'Payment Management' interface with a 'History Payment' dropdown menu. Below the menu is a table listing 10 transactions. Each row includes a transaction ID, donor name and ID, campaign name, nominal amount, status, date, and a 'settlement' button. The status of the transactions varies between 'LUNAS' (Paid) and 'Transaksi Dibatalkan' (Transaction Cancelled).

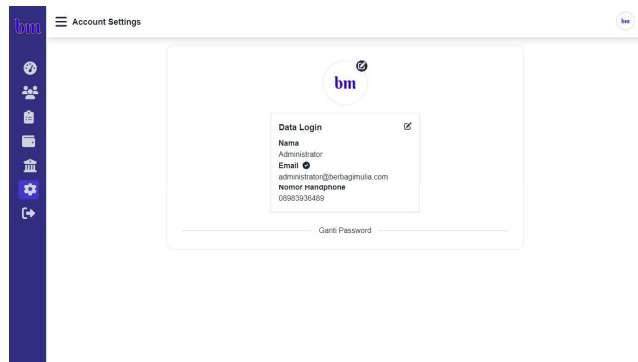
No. Transaksi	Donatur	Campaign	Nominal	Status	Tanggal	Keterangan
1	BM-803217050 Ain Diahmadi 080163103932	LHAT	Rp 10.359	Transaksi Dibatalkan	14/12/2022 20:24:08	settlement
2	BM-1088194211 Hamba Alah 081355141991	LHAT	Rp 50.843	Transaksi Dibatalkan	14/12/2022 20:23:41	settlement
3	BM-8440162110 Vibha Playfito 080909466484	LHAT	Rp 50.623	Transaksi Dibatalkan	14/12/2022 20:20:44	settlement
4	BM-3652162500 Hamba Alah 082296666988	LHAT	Rp 30.000	Transaksi Dibatalkan	14/12/2022 17:01:29	settlement
5	BM-5307163956 Hamba Alah 082295821119	LHAT	Rp 100.000	Transaksi Dibatalkan	14/12/2022 17:06:38	settlement
6	BM-6485100544 Hamba Alah 081320293413	LHAT	Rp 100.422	Transaksi Dibatalkan	14/12/2022 14:14:53	Transaksi telah kadaluarsa
7	BM-1709628236 Riafiti 081348555880	LHAT	Rp 100.000	Transaksi Dibatalkan	14/12/2022 14:14:49	Transaksi telah kadaluarsa
8	BM-3687232717 Riko Riyanto 082122260322	LHAT	Rp 50.000	Transaksi Dibatalkan	14/12/2022 14:14:45	Transaksi telah kadaluarsa
9	BM-2765214900 Kiki 085916976365	LHAT	Rp 25.000	Transaksi Dibatalkan	14/12/2022 14:14:42	Transaksi telah kadaluarsa
10	BM-7270114548 Kiki 085916976365	LHAT	Rp 25.000	Transaksi Dibatalkan	14/12/2022 14:14:38	Transaksi telah kadaluarsa

Gambar 4.60 Hasil tampilan halaman riwayat pembayaran



The screenshot displays the 'Payment Management' interface with a 'Settings' dropdown menu. The 'Metode Pembayaran Aktif' section shows two options: 'Auto Payment (Midtrans)' and 'Manual Payment (Bank Transfer)'. The 'Bank yang tersedia' section lists two banks: 'BSI - Bank Syariah Indonesia' and 'mandiri - Bank Mandiri'. The 'Akun Bank Untuk Manual Payment' section shows a selected account for 'BSI - BANK SYARIAH INDONESIA' with account number '7210364522 a.n. Yys Amal Bakti Nusantara Tegai'.

Gambar 4.61 Hasil tampilan halaman pengaturan metode pembayaran yang di dukung



Gambar 4.62 Hasil tampilan halaman profil akun *Super Admin*

## E. *Deployment Delivery and Feedback*

### 1. *Proses Deployment*

Proses *deployment* dilakukan untuk mengunggah aplikasi web yang telah dibuat ke dalam jaringan internet melalui layanan *cloud hosting*. Pemilihan *framework* Laravel Breeze sebagai teknologi aplikasi web memiliki peran penting dalam proses ini. Hal tersebut dikarenakan Laravel Breeze merupakan salah satu ekosistem dalam *framework* Laravel yang tersusun di atas bahasa pemrograman PHP.

Bahasa pemrograman PHP telah didukung oleh banyak sekali *provider* yang menyediakan layanan *cloud hosting*. Hal tersebut memberikan banyak pilihan terutama dalam aspek kesesuaian harga.

Dalam hal ini, penulis berhasil mendapatkan layanan *cloud hosting* dengan harga Rp 10.000 pertahun, tidak termasuk harga domain web. Harga tersebut penulis dapatkan dari promo yang ditawarkan oleh IDCloudHost pada *event* di bulan September 2022 dari harga normalnya Rp 170.000 pertahun. Harga tersebut terbilang murah untuk ukuran layanan *cloud hosting* dengan spesifikasi sebagai berikut:

- a. *CPU: single core,*
- b. *Storage: 1 GB SSD NVMe,*
- c. *RAM: 512 MB,*
- d. *Bandwidth dan database tak terbatas.*

Berikut ini beberapa langkah yang dilakukan pada proses *deployment*:

- a. Lakukan *building* Javascript menjadi aset statis yang optimal untuk mode *production* dengan perintah *npm run build* di terminal lokal,
- b. Kompres semua *codebase* dari aplikasi web ke dalam ekstensi zip,
- c. Unggah *file zip* tersebut ke dalam *cloud hosting* melalui *dashboard* Cpanel dan ekstrak di dalam direktori *public\_html*,

- d. Siapkan *database* di *cloud hosting* dan konfigurasi identitas *database* di *file env* dari aplikasi web,
  - e. Buat *Route URL* baru untuk memanggil perintah “*artisan migrate --seed*” dalam proses migrasi *database* di *web browser* dan hapus setelah prosesnya selesai,
  - f. Lakukan optimasi aplikasi web dengan mengunjungi halaman “*/optimize*” untuk menjalankan perintah “*artisan optimize*” di *web browser* yang hanya dapat dilakukan oleh *Super Admin*.
2. Hasil Analisis *Feedback*

Analisis *feedback* dilakukan untuk mengetahui beberapa *feedback* yang diberikan oleh pihak Yayasan Amal Bakti Nusantara Tegal terkait fitur aplikasi web yang dibuat. Analisis *feedback* dilakukan setelah proses *deployment* dengan hasil sebagai berikut:

Tabel 4.1 Hasil analisis *feedback* dan perbaikannya

No	<i>Feedback</i>	Perbaikan
1	Ganti nama tombol donasi baru sesuai dengan kategori programnya	Nama tombol donasi telah diganti sesuai dengan kategori program utama, yaitu infak, wakaf, dan sedekah. Selain ketiga kategori tersebut, kata donasi tetap digunakan sebagai nama tombol donasi baru



No	Feedback	Perbaikan
2	Tambahkan <i>contact person</i> dari <i>customer service</i> pada halaman donasi baru	Nomor WhatsApp <i>customer service</i> ditampilkan pada halaman donasi baru yang dapat diganti sesuai dengan <i>customer service</i> dari program galang dana
3	Ganti warna utama aplikasi web menjadi lebih muda	Warna utama aplikasi web berhasil diubah dengan mengganti kode warna pada <i>class primary</i> menjadi biru muda dengan kode warna "#3b82f6"
4	Data yang ditampilkan pada halaman detail program harus dibatasi terlebih dahulu	Ditambahkannya tombol lihat selengkapnya untuk mem-batasi konten yang tampil saat pertama kali halaman detail program dimuat
5	Fitur promosi program menyebabkan fungsi <i>error</i> pada halaman donasi baru	Ditemukan kesalahan sistem dalam mendeteksi kode promosi. Solusinya adalah menggunakan <i>method</i> milik Javascript untuk mendeteksi data yang dikirimkan dalam sebuah <i>link URL</i> bernama <i>URLSearchParams</i>
6	Tambahkan nama yayasan di bagian <i>footer</i> aplikasi web	Nama yayasan berhasil ditambahkan pada bagian <i>footer</i> aplikasi web
7	Ganti tipe <i>form</i> dalam memilih metode pembayaran pada halaman donasi baru	<i>Form select</i> pada pemilihan metode pembayaran diganti menggunakan <i>form</i> dengan tipe <i>radio button</i> dan ditambahkan sedikit <i>styling</i> dengan CSS pada bagian <i>form label</i> -nya
8	<i>Form email</i> pada halaman donasi	Sistem berhasil mendeteksi pengisian <i>form email</i> menjadi

No	<i>Feedback</i>	Perbaikan
	baru diganti menjadi bersifat opsional	opsional, yang artinya sistem tetap akan memproses donasi baru walaupun <i>form email</i> tidak diisi
9	Sistem tidak dapat menangani teks yang berisi emoji	Mengganti konfigurasi tipe karakter <i>database</i> MySQL pada <i>framework</i> Laravel, seperti <i>charset</i> dengan nilai " <i>utf8mb4</i> ", dan <i>collation</i> dengan nilai " <i>utf8mb4_unicode_ci</i> ",
10	Pemuatan halaman masih terlalu lama ketika aplikasi web pertama kali dibuka	Implementasi <i>code-splitting</i> menggunakan fungsi <i>React Lazy</i> agar modul Javascript yang dimuat hanya yang benar-benar ditampilkan saat aplikasi web dibuka pertama kali dan integrasikan aplikasi web dengan Cloudflare sebagai <i>Content Delivery Network (CDN)</i> untuk menyimpan <i>cache</i> aset statis
11	Pemuatan halaman yang berisi gambar terlalu lama	Implementasi <i>lazy loading</i> pada setiap gambar menggunakan <i>library react-lazy-load-image-component</i>
12	Tambahkan fungsi <i>draft</i> agar dapat menyimpan program baru yang belum siap untuk dipublikasi	Fungsi <i>draft</i> dalam proses pembuatan program baru berhasil ditambahkan dengan mengubah status publikasinya menjadi " <i>draft</i> "
13	Fungsi kustomisasi <i>slug</i> program tidak berjalan	Menghapus atribut <i>read-only</i> di <i>tag input slug</i> pada <i>form</i> pembuatan program baru
14	Adanya kesulitan mengidentifikasi pembayaran	Ditambahkan kode unik pembayaran pada jumlah nominal donasi untuk

No	Feedback	Perbaikan
	dengan verifikasi manual	memudahkan identifikasi pembayaran yang sah

### 3. Hasil Pengujian *Black Box Testing*

Pengujian *black box* dilakukan untuk mengetahui berfungsi tidaknya suatu fitur dari aplikasi web yang telah dibuat. Pengujian ini dilakukan oleh pengembang setelah dilakukannya proses *deployment*. Berikut ini merupakan hasil pengujian *black box* pada setiap fitur yang telah dibuat:

#### a. Pengujian fitur pembatasan halaman

Tabel 4.2 Hasil pengujian pada fitur pembatasan halaman berdasarkan *role* pengguna

1	Deskripsi Pengujian	<i>Login</i> sebagai akun <i>User</i> dan akses <i>link url /user</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman <i>dashboard</i> akun <i>User</i>
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman <i>dashboard</i> akun <i>User</i>
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	<i>Login</i> sebagai <i>User</i> dan akses <i>link url /superadmin</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman peringatan dengan kode 403
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman peringatan kode 403 dengan keterangan “Silahkan <i>Login</i> sebagai <i>Super Admin</i> ”
	Kesimpulan	Sesuai
3	Deskripsi Pengujian	<i>Login</i> sebagai <i>User</i> dan akses <i>link url /admin</i>

	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman peringatan dengan kode 403
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman peringatan kode 403 dengan keterangan “Silahkan <i>Login</i> sebagai Admin Lembaga”
	Kesimpulan	Sesuai
4	Deskripsi Pengujian	<i>Login</i> sebagai Admin Lembaga dan akses <i>link url /admin</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman <i>dashboard</i> akun Admin Lembaga
	Hasil Pengujian	<i>Login</i> berhasil dan halaman <i>dashboard</i> akun Admin Lembaga
	Kesimpulan	Sesuai
5	Deskripsi Pengujian	<i>Login</i> sebagai Admin Lembaga dan akses <i>link url /superadmin</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman peringatan dengan kode 403
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman peringatan kode 403 dengan keterangan “Silahkan <i>Login</i> sebagai <i>Super Admin</i> ”
	Kesimpulan	Sesuai
6	Deskripsi Pengujian	<i>Login</i> sebagai Admin Lembaga dan akses <i>link url /user</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman peringatan dengan kode 403
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman peringatan kode 403 dengan keterangan “Silahkan <i>Login</i> sebagai Pengguna Biasa”
	Kesimpulan	Sesuai
7	Deskripsi Pengujian	<i>Login</i> sebagai <i>Super Admin</i> dan akses <i>link url /superadmin</i>
	Hasil Yang Diharapkan	(Husada, 2019) Tampil halaman <i>dashboard Super Admin</i>
	Hasil	<i>Login</i> berhasil dan tampil

	Pengujian	halaman <i>dashboard Super Admin</i>
	Kesimpulan	Sesuai
8	Deskripsi Pengujian	<i>Login</i> sebagai <i>Super Admin</i> dan akses <i>link url /admin</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman peringatan dengan kode 403
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman peringatan kode 403 dengan keterangan “Silahkan <i>Login</i> sebagai Admin Lembaga”
	Kesimpulan	Sesuai
9	Deskripsi Pengujian	<i>Login</i> sebagai <i>Super Admin</i> dan akses <i>link url /user</i>
	Hasil Yang Diharapkan	(Husada, 2019)Tampil halaman peringatan dengan kode 403
	Hasil Pengujian	<i>Login</i> berhasil dan tampil halaman peringatan kode 403 dengan keterangan “Silahkan <i>Login</i> sebagai pengguna biasa”
	Kesimpulan	Sesuai

b. Pengujian fitur registrasi akun

Tabel 4.3 Hasil pengujian pada fitur registrasi akun User dan akun Admin Lembaga

1	Deskripsi Pengujian	Mengosongkan <i>form</i> registrasi
	Hasil Yang Diharapkan	(Husada, 2019)Registrasi tidak dapat diproses
	Hasil Pengujian	Tombol registrasi tidak bisa di klik dan muncul peringatan “mohon isi bidang ini”
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	Mengisikan semua <i>form</i> registrasi dengan benar
	Hasil Yang Diharapkan	(Husada, 2019)Sistem memproses registrasi akun

	Hasil Pengujian	Sistem berhasil menambahkan data ke <i>database</i> dan muncul halaman <i>login</i> dengan peringatan “registrasi berhasil”
	Kesimpulan	Sesuai
3	Deskripsi Pengujian	Mengisi <i>form email</i> dengan dengan <i>email</i> yang sudah terdaftar
	Hasil Yang Diharapkan	(Husada, 2019)Registrasi tidak dapat diproses
	Hasil Pengujian	Registrasi tidak diproses oleh sistem dan muncul peringatan “alamat <i>email</i> ini sudah digunakan yang lain”
	Kesimpulan	Sesuai
4	Deskripsi Pengujian	Mengisi semua <i>form</i> registrasi akun Admin Lembaga dengan benar
	Hasil Yang Diharapkan	(Husada, 2019)Sistem memproses registrasi akun Admin Lembaga
	Hasil Pengujian	Sistem berhasil menambahkan data ke <i>database</i> dengan status “menunggu persetujuan” dan muncul halaman <i>login</i> dengan peringatan “registrasi berhasil”
	Kesimpulan	Sesuai
5	Deskripsi Pengujian	Akun Admin lembaga baru melakukan proses <i>login</i>
	Hasil Yang Diharapkan	(Husada, 2019)Sistem membatasi akses halaman bagi akun lembaga baru
	Hasil Pengujian	Sistem menampilkan halaman status pendaftaran hingga status pendaftaran diperbarui oleh <i>Super Admin</i> menjadi disetujui
	Kesimpulan	Sesuai

6	Deskripsi Pengujian	<i>Super Admin</i> mengonfirmasi registrasi akun Admin Lembaga
	Hasil Yang Diharapkan	(Husada, 2019)Sistem memperbarui status registrasi lembaga
	Hasil Pengujian	Status registrasi Admin Lembaga menjadi berhasil diperbarui menjadi “pendaftaran disetujui” dan dapat mengakses halaman <i>dashboard</i> Admin Lembaga
	Kesimpulan	Sesuai
7	Deskripsi Pengujian	<i>Super Admin</i> menolak registrasi akun Admin Lembaga
	Hasil Yang Diharapkan	(Husada, 2019)Sistem memperbarui status registrasi akun Admin Lembaga
	Hasil Pengujian	Status registrasi lembaga menjadi berhasil diperbarui menjadi “pendaftaran ditolak” dan tidak dapat mengakses halaman <i>dashboard</i> Admin Lembaga
	Kesimpulan	Sesuai

c. Pengujian fitur pengelolaan program galang dana

Tabel 4.4 Hasil pengujian pada fitur pengelolaan program galang dana

1	Deskripsi Pengujian	Mengosongkan <i>form</i> pembuatan program baru
	Hasil Yang Diharapkan	Pembuatan program baru tidak dapat diproses(Husada, 2019)
	Hasil Pengujian	Muncul <i>error</i> pada setiap <i>form</i> dengan keterangan “bidang ini wajib diisi”
	Kesimpulan	Sesuai
2	Deskripsi	Mengisi semua <i>form</i> pembuatan

	Pengujian	program baru dengan benar
	Hasil Yang Diharapkan	Sistem memproses pembuatan program baru(Husada, 2019)
	Hasil Pengujian	Program baru berhasil ditambahkan di <i>database</i> dengan status “Menunggu konfirmasi”
	Kesimpulan	Sesuai
3	Deskripsi Pengujian	<i>Super Admin</i> mengonfirmasi program baru
	Hasil Yang Diharapkan	Sistem memperbarui status penayangan program (Husada, 2019)
	Hasil Pengujian	Status program baru berhasil diperbarui dan ditayangkan pada halaman <i>homepage</i>
	Kesimpulan	Sesuai

d. Pengujian fitur pengelolaan laporan berita

Tabel 4.5 Hasil pengujian pada fitur pengelolaan laporan berita

1	Deskripsi Pengujian	Mengosongkan <i>form</i> pembuatan laporan berita baru
	Hasil Yang Diharapkan	Pembuatan program baru tidak dapat diproses (Husada, 2019)
	Hasil Pengujian	Muncul <i>error</i> pada setiap <i>form</i> dengan keterangan “bidang ini wajib diisi”
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	Mengisi semua <i>form</i> pembuatan laporan berita baru dengan benar
	Hasil Yang Diharapkan	Sistem memproses pembuatan laporan berita baru (Husada, 2019)
	Hasil Pengujian	Laporan berita baru berhasil disimpan dan di-tayangkan pada



		halaman detail program
	Kesimpulan	Sesuai

e. Pengujian fitur donasi baru

Tabel 4.6 Hasil pengujian pada fitur donasi baru

1	Deskripsi Pengujian	Mengosongkan <i>form</i> pembuatan donasi baru
	Hasil Yang Diharapkan	Pembuatan donasi baru tidak dapat diproses (Husada, 2019)
	Hasil Pengujian	Muncul <i>error</i> pada <i>form</i> nominal, nama, nomor <i>handphone</i> dan metode pembayaran dengan keterangan "bidang ini wajib diisi"
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	Mengisi semua <i>form</i> pembuatan donasi baru dengan benar
	Hasil Yang Diharapkan	Sistem memproses pembuatan donasi baru (Husada, 2019)
	Hasil Pengujian	Donasi baru berhasil ditambahkan di <i>database</i> dengan status "Menunggu pembayaran" dan muncul halaman pembayaran
	Kesimpulan	Sesuai

f. Pengujian fitur *payment gateway* verifikasi manual

Tabel 4.7 Hasil pengujian pada fitur *payment gateway* verifikasi manual

1	Deskripsi Pengujian	<i>User</i> melakukan konfirmasi pembayaran
	Hasil Yang Diharapkan	Sistem dapat menangani gambar bukti pembayaran (Husada, 2019)
	Hasil Pengujian	Sistem berhasil menyimpan data gambar dan memperbarui status

		pembayaran menjadi “Menunggu konfirmasi” serta dialihkan ke halaman status pembayaran
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	<i>Super Admin</i> mengonfirmasi pembayaran secara manual
	Hasil Yang Diharapkan	Sistem dapat memproses konfirmasi pembayaran beserta dampak yang ditimbulkannya (Husada, 2019)
	Hasil Pengujian	Status pembayaran berhasil diperbarui menjadi “berhasil” dan berdampak pada bertambahnya jumlah donasi yang terkumpul di suatu program hingga data donasi ditampilkan pada halaman detail program
	Kesimpulan	Sesuai
3	Deskripsi Pengujian	<i>Super Admin</i> menolak pembayaran secara manual
	Hasil Yang Diharapkan	Sistem dapat memproses penolakan pembayaran (Husada, 2019)
	Hasil Pengujian	Status pembayaran berhasil diperbarui menjadi “dibatalkan” yang dapat dilihat di halaman status pembayaran beserta alasan penolakannya
	Kesimpulan	Sesuai

g. Pengujian fitur *payment gateway* verifikasi otomatis

Tabel 4.8 Hasil pengujian pada fitur *payment gateway* verifikasi otomatis

1	Deskripsi	<i>User</i> memilih metode
---	-----------	----------------------------

	Pengujian	pembayaran otomatis yang disediakan seperti <i>Bank Virtual Account</i> dan Gopay
	Hasil Yang Diharapkan	Sistem dapat menangani permintaan metode pembayaran yang dipilih (Husada, 2019)
	Hasil Pengujian	Sistem berhasil menampilkan data pembayaran yang berasal dari <i>Midtrans Payment Gateway</i>
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	<i>User</i> membayar sesuai dengan metode pembayaran otomatis yang dipilih dan tetap membuka halaman pembayaran
	Hasil Yang Diharapkan	Sistem dapat mendeteksi pembayaran yang dilakukan oleh <i>User</i> (Husada, 2019)
	Hasil Pengujian	Sistem berhasil memperbarui status pembayaran yang dikirimkan oleh <i>Midtrans Payment Gateway</i> menjadi " <i>settlement</i> "
	Kesimpulan	Sesuai
3	Deskripsi Pengujian	<i>User</i> membayar sesuai dengan metode pembayaran otomatis yang dipilih, tetapi halaman pembayaran tidak dibuka kembali
	Hasil Yang Diharapkan	Sistem dapat mendeteksi pembayaran yang dilakukan oleh <i>User</i> (Husada, 2019)
	Hasil Pengujian	Sistem berhasil memperbarui status pembayaran yang dikirimkan oleh <i>Midtrans Payment Gateway</i> menjadi " <i>settlement</i> ", setelah halaman pembayaran atau halaman status pembayaran atau dilakukan

		pengecekan status secara manual oleh <i>Super Admin</i> di halaman <i>payment management</i>
	Kesimpulan	Sesuai
4	Deskripsi Pengujian	<i>User</i> memilih metode pembayaran otomatis tetapi tidak langsung membayar
	Hasil Yang Diharapkan	Sistem dapat menyimpan data pembayaran otomatis yang dipilih (Husada, 2019)
	Hasil Pengujian	Sistem berhasil menyimpan data pembayaran yang dikirimkan oleh <i>Midrans Payment Gateway</i> menjadi " <i>pending</i> "
	Kesimpulan	Sesuai
5	Deskripsi Pengujian	<i>User</i> memilih metode pembayaran otomatis tetapi tidak membayar hingga batas waktu pembayaran berakhir
	Hasil Yang Diharapkan	Sistem dapat memperbarui status pembayaran otomatis yang dipilih (Husada, 2019)
	Hasil Pengujian	Sistem berhasil memperbarui status pembayaran yang dikirimkan oleh <i>Midrans Payment Gateway</i> menjadi " <i>expire</i> "
	Kesimpulan	Sesuai
6	Deskripsi Pengujian	<i>User</i> tidak memilih metode pembayaran otomatis hingga batas waktu pembayaran berakhir
	Hasil Yang Diharapkan	Sistem dapat memperbarui status pembayaran yang dipilih (Husada, 2019)
	Hasil Pengujian	Sistem berhasil memperbarui status pembayaran menjadi

		"transaksi telah kadaluarsa" yang dilakukan secara manual oleh <i>Super Admin</i> di halaman <i>payment management</i>
	Kesimpulan	Sesuai

h. Pengujian fitur promosi program galang dana

Tabel 4.9 Hasil pengujian pada fitur promosi program galang dana

1	Deskripsi Pengujian	<i>User</i> memilih program yang ingin dipromosikan
	Hasil Yang Diharapkan	Sistem dapat menyimpan data promosi program yang dipilih (Husada, 2019)
	Hasil Pengujian	Sistem menyimpan data promosi program dan menampilkan tautan yang dapat dibagikan kepada <i>User</i> lain sebagai donatur
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	<i>User</i> lain selaku donatur melakukan donasi melalui tautan promosi program yang dibagikan
	Hasil Yang Diharapkan	Sistem dapat mendeteksi donasi melalui kode promosi yang dibagikan (Husada, 2019)
	Hasil Pengujian	Sistem dapat mendeteksi donasi yang berasal dari tautan promosi dan memperbarui total donasi terkumpul dari kode promosi yang dibagikan ketika ada donasi baru dengan status pembayaran " <i>settlement</i> "
	Kesimpulan	Sesuai

## i. Pengujian fitur penarikan dana

Tabel 4.10 Hasil pengujian pada fitur penarikan dana

1	Deskripsi Pengujian	Admin Lembaga mengisi penarikan dana beserta informasi bank penerima setelah ada donasi yang sudah terkumpul
	Hasil Yang Diharapkan	Sistem dapat menyimpan data penarikan dana baru dan mengurangi jumlah donasi terkumpul (Husada, 2019)
	Hasil Pengujian	Sistem berhasil menyimpan data penarikan dana baru ke <i>database</i> dengan status "sedang diproses" dan jumlah donasi terkumpul telah berkurang
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	<i>Super Admin</i> melakukan konfirmasi penarikan dana
	Hasil Yang Diharapkan	Sistem dapat memperbarui status penarikan dana dan berita laporan penarikan dana akan muncul di halaman detail program pada <i>homepage</i> (Husada, 2019)
	Hasil Pengujian	Status penarikan dana berhasil diperbarui menjadi " <i>success</i> " dan berita laporan penarikan dana muncul di halaman detail program pada <i>Homepage</i>
	Kesimpulan	Sesuai
3	Deskripsi Pengujian	<i>Super Admin</i> melakukan penolakan penarikan dana
	Hasil Yang Diharapkan	Sistem dapat memperbarui status penarikan dana dan pengurangan jumlah donasi terkumpul dibatalkan (Husada,

		2019)
	Hasil Pengujian	Status penarikan dana berhasil diperbarui menjadi “ <i>Rejected</i> ” dan jumlah donasi terkumpul dikembalikan seperti sebelumnya
	Kesimpulan	Sesuai

j. Pengujian fitur integrasi dengan Facebook Ads

Tabel 4.11 Hasil pengujian pada fitur integrasi dengan Facebook Ads melalui kode Pixel

1	Deskripsi Pengujian	Kode Facebook Pixel ditambahkan oleh Admin pada setiap program yang berjalan
	Hasil Yang Diharapkan	Sistem dapat menangani integrasi program galang dana dengan Facebook Ads (Husada, 2019)
	Hasil Pengujian	Kode Facebook Pixel berhasil disimpan di <i>database</i> dan sistem berhasil mengirimkan data aktivitas kunjungan halaman detail program di <i>homepage</i> kepada Facebook Ads
	Kesimpulan	Sesuai
2	Deskripsi Pengujian	Mengosongkan <i>form</i> tambah kode Facebook Pixel
	Hasil Yang Diharapkan	Sistem tidak dapat menyimpan data kode Facebook Pixel (Husada, 2019)
	Hasil Pengujian	Tombol <i>submit</i> tidak dapat diklik dan tidak ada interaksi penambahan data pada sistem
	Kesimpulan	Sesuai

Total pengujian yang dilakukan berjumlah 39 dengan persentase kesesuaian antara hasil yang diharapkan dan hasil pengujian sebanyak 100%. Sehingga, kesimpulan yang diambil dari pengujian *black box* adalah semua fitur telah berjalan sesuai dengan harapan pengembang.

#### 4. Hasil Pengujian *User Acceptance Test*

Pengujian *user acceptance test* (UAT) dilakukan sebagai validasi terhadap kualitas fitur aplikasi web yang telah dibuat melalui survei pengalaman pengguna dalam bentuk pengisian kuesioner berdasarkan *role* setiap pengguna yang terdapat pada aplikasi web ini, yaitu *Super Admin* selaku pihak Yayasan Amal Bakti Nusantara Tegal sebagai pemilik dari *platform digital* Berbagi Mulia, Admin Lembaga sebagai lembaga penggalang dana dan *User* sebagai donatur dari proses penggalangan dana. Permintaan kepada para responden untuk mengikuti survey pengalaman pengguna ini disampaikan melalui percakapan di aplikasi WhatsApp dengan bukti yang terlampir pada Lampiran 5, Lampiran 6, dan Lampiran 7.



Berdasarkan hasil kuesioner yang terlampir pada Lampiran 8, total responden yang berpartisipasi hanyalah satu orang saja, yaitu pihak Yayasan Amal Bakti Nusantara Tegal selaku pemilik platform digital Berbagi Mulia dengan hasil perhitungan pada pengujian *user acceptance test* (UAT) untuk setiap fitur dari *role Super Admin* yang telah dibuat adalah sebagai berikut:

Tabel 4.12 Hasil pengujian *user acceptance test* (UAT) oleh *Super Admin*

Fitur	Kualitas	Persentase	Total Responden
<i>Login</i> sebagai <i>Super Admin</i> selaku pemilik platform digital Berbagi Mulia	Sangat Baik	100%	1
Verifikasi program galang dana baru	Sangat Baik	100%	1
Verifikasi Akun Admin selaku Lembaga Galang Dana Baru	Sangat Baik	100%	1
Verifikasi pembayaran baru secara manual	Sangat Baik	100%	1
Verifikasi penarikan dana baru	Sangat Baik	100%	1

Total pengujian yang dilakukan berdasarkan fitur dari *role Super Admin* yang dirancang dalam *use case*

*diagram* berjumlah lima fitur. Hasil pengujian User Acceptance Test pada *role Super Admin* memberikan kesimpulan bahwa seluruh fitur yang tersedia untuk *role Super Admin* pada aplikasi web aplikasi web yang telah dibuat memiliki kualitas yang sangat baik.

Berdasarkan hasil kuesioner yang terlampir pada lampiran 9, total responden yang berpartisipasi sebanyak dua orang yang mewakili tiga lembaga galang dana yang berbeda, yaitu DKM Nurul Barokah, Gerakan Infaq Beras Tegal, dan KHO. Hasil perhitungan pada pengujian *user acceptance test* (UAT) untuk setiap fitur dari *role Admin Lembaga* yang telah dibuat adalah sebagai berikut:

Tabel 4.13 Hasil pengujian *user acceptance test* (UAT) oleh Admin Lembaga

Fitur	Kualitas	Persentase	Total Responden
Registrasi akun Admin Lembaga sebagai lembaga penggalang dana	Sangat Baik	100%	3
Login untuk Akun Admin Lembaga sebagai Lembaga Penggalang Dana	Sangat Baik	100%	3
Pengelolaan Laporan Berita Program Galang Dana	Sangat Baik	100%	3
Mengintegrasikan	Sangat	100%	3

Fitur	Kualitas	Persentase	Total Responden
Facebook Ads dengan Program Galang Dana	Baik		
Pengarikan Dana Terkumpul dari Program Galang Dana	Sangat Baik	66.7%	2
	Belum Mencoba	33,3%	1

Total pengujian yang dilakukan berdasarkan fitur dari role Admin Lembaga yang dirancang dalam *use case diagram* berjumlah enam fitur. Hasil pengujian *user acceptance test* pada role Admin Lembaga memberikan kesimpulan bahwa terdapat lima fitur yang sudah diuji oleh ketiga responden dan diberikan nilai dengan kualitas sangat baik. Sedangkan, sisa satu fitur lainnya hanya diuji oleh satu responden dengan nilai kualitas sangat baik, dua responden lainnya menjawab belum mencoba fitur penarikan dana dari total donasi yang terkumpul.

Berdasarkan hasil kuesioner yang terlampir pada Lampiran 10, total responden yang berpartisipasi sebanyak 13 responden. Total pengujian yang dilakukan berdasarkan fitur dari *role User* yang dirancang dalam *use case diagram* berjumlah enam

fitur. Untuk memudahkan proses analisis, setiap fitur tersebut dilambangkan dengan simbol berikut ini:

Tabel 4.14 Lambang fitur dalam proses analisis UAT pada *role User*

Lambang	Fitur
A	Pembuatan Donasi Baru
B	Pembayaran dengan Transfer Bank
C	Pembayaran dengan Bank Virtual Account atau Gopay
D	Registrasi Akun User atau Donatur
E	Login untuk Akun User atau Donatur
F	Promosi Program Galang Dana oleh akun User atau Donatur

Dikarenakan pada lampiran 9 memiliki hasil kuesioner dengan variasi data yang berbeda, maka diperlukan proses pembobotan data dengan klasifikasi berdasarkan kualitas sebagai berikut:

Tabel 4.15 Pembobotan klasifikasi kualitas pada pengujian user acceptance test

Lambang	Kualitas	Bobot
SB	Sangat Baik	4
B	Baik	3
KB	Kurang Baik	2
TB	Tidak Baik	1
BM	Belum Mencoba	0

Berdasarkan tabel pembobotan diatas, maka diperoleh hasil perhitungan pada pengujian *user acceptance test* (UAT) untuk setiap fitur dari *role User* sebagai donatur adalah sebagai berikut:

Tabel 4.16 Hasil pengujian *user acceptance test* (UAT) oleh *User* sebagai donatur

Fitur	Kualitas (y <sub>1</sub> )	Bobot (a <sub>1</sub> )	Total Responden (b)	Penilaian			
				a <sub>1</sub> x b (c)	Total (d)	(a <sub>2</sub> )	(y <sub>2</sub> )
A	SB	4	8	32	47	3,6	B
	B	3	5	15			
B	SB	4	6	24	33	2,5	KB
	B	3	3	9			
	BM	0	4	0			
C	SB	4	8	32	32	2,4	KB
	BM	0	5	0			
D	SB	4	5	20	41	3,1	B
	B	3	7	21			
	BM	0	1	0			
E	SB	4	5	20	41	3,1	B
	B	3	7	21			
	BM	0	1	0			
F	SB	4	6	24	42	3,2	B
	B	3	6	18			
	BM	0	1	0			

Rumus yang digunakan dalam perhitungan (a<sub>2</sub>) di Tabel 4.16 diatas adalah pembagian antara nilai (c) dengan total responden sejumlah 13 orang yang kemudian menghasilkan nilai pembobotan akhir (a<sub>2</sub>) dan dapat diidentifikasi klasifikasi nilai pembobotan tersebut berdasarkan Tabel 4.15. Hasil pengujian *user acceptance test* pada *role User* sebagai donatur memberikan kesimpulan bahwa terdapat empat fitur yang sudah diuji oleh semua responden dan mendapatkan nilai dengan kualitas yang baik.

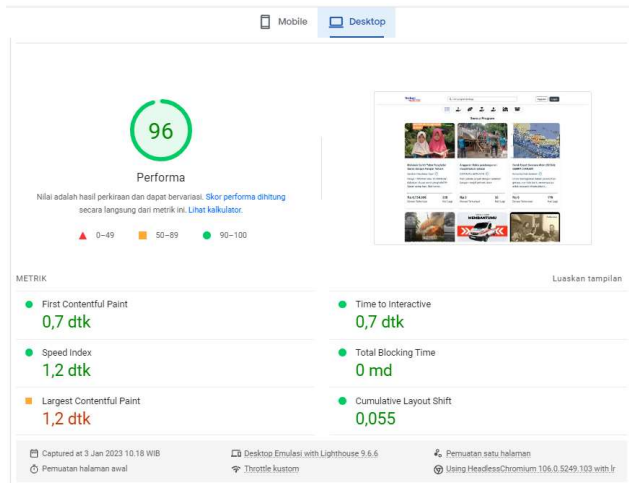
Sedangkan, sisa dua fitur lainnya yang berkaitan dengan metode pembayaran yang pada penerapannya hanya dapat dipilih salah satunya saja antara metode pembayaran manual atau otomatis. Sehingga, ada banyak responden yang belum mencoba kedua fitur tersebut. Oleh karena itulah, pada hasil pengujian di Tabel 4.16 di atas kedua fitur tersebut mendapatkan nilai kurang baik dikarenakan banyak responden yang memilih pilihan jawaban berupa belum mencoba fitur tersebut.

#### 5. Hasil Analisis Performa

Analisis performa dilakukan untuk mengetahui kinerja aplikasi web yang telah dibuat dengan pengukuran menggunakan beberapa *tools* yang biasa digunakan untuk mengukur performa aplikasi web secara *online*, seperti Pagespeed Insight dan GT Metrix. Pengujian ini akan berfokus pada kecepatan pemuatan halaman dengan metode pengukuran yang distandarisasi sebagai *core web vital*.

Proses analisis dilakukan dengan pengujian sebanyak lima kali menggunakan masing-masing *tools* kemudian dihitung nilai rata-rata untuk mengukur konsistensi performa aplikasi web. Berikut ini

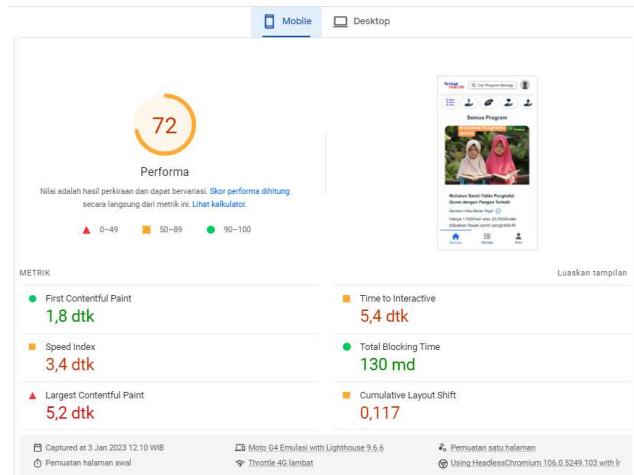
merupakan hasil dari perhitungan performa yang sudah dilakukan:



Gambar 4.63 Sampel pengujian menggunakan Page Speed Insight pada perangkat *desktop*

Tabel 4.17 Hasil perhitungan performa web menggunakan Page Speed Insight pada perangkat *desktop*

No	FCP	LCP	TBT	CLS	Performance Score
1	700 ms	1200 ms	0 ms	0,055	96
2	700 ms	1300 ms	0 ms	0,055	94
3	700 ms	1300 ms	0 ms	0,055	94
4	700 ms	1300 ms	0 ms	0,055	95
5	700 ms	1300 ms	0 ms	0,055	95
<b>Mean</b>	700 ms	1280 ms	0 ms	0,055	94,8

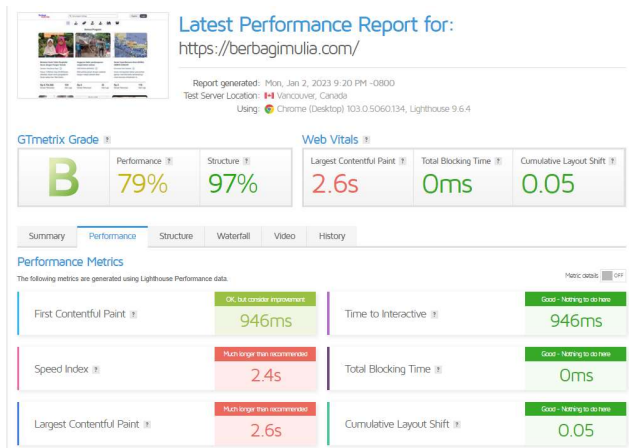


Gambar 4.64 Sampel pengujian menggunakan Page Speed Insight pada perangkat *mobile*

Tabel 4.18 Hasil perhitungan performa web menggunakan Page Speed Insight pada perangkat *mobile*

No	FCP	LCP	TBT	CLS	<i>Performance Score</i>
1	1800 ms	5200 ms	130 ms	0,177	72
2	1800 ms	5300 ms	130 ms	0,177	72
3	1800 ms	5200 ms	130 ms	0,177	74
4	1800 ms	5000 ms	150 ms	0,177	76
5	1700 ms	5300 ms	100 ms	0,177	75
<b><i>Mean</i></b>	1780 ms	5200 ms	128 ms	0,177	73,8





Gambar 4.65 Sampel hasil pengujian menggunakan GT Metrix

Tabel 4.19 Hasil perhitungan performa web menggunakan GT Metrix

No	FCP	LCP	TBT	CLS	Performance Score
1	946 ms	2600 ms	0 ms	0,05	79%
2	883 ms	3200 ms	0 ms	0,05	75%
3	915 ms	3600 ms	0 ms	0,05	72%
4	913 ms	3000 ms	0 ms	0,05	76%
5	949 ms	2800 ms	0 ms	0,05	77%
<b>Mean</b>	921,2 ms	3040 ms	0 ms	0,05	75,8%

Ketiga tabel diatas memberikan perhitungan dengan nilai rata-rata *first contentful paint* (FCP) yang

stabil dibawah 1800 ms. Hal tersebut menjelaskan bahwa aplikasi web terhitung cepat dalam menyajikan halaman web ketika pertama kali dibuka. FCP memiliki peran penting dalam memvisualisasikan respon yang terjadi pada halaman web, sehingga pengunjung tidak mendapatkan tampilan kanvas putih terlalu lama walaupun proses *rendering* halaman masih terjadi.

Selain itu, nilai rata-rata pada metrik *total blocking time* (TBT) juga telah melewati ambang batas nilai performa yang baik dengan nilai di bawah 200 ms. Sedangkan penilaian pada metrik *cumulative layout shift* (CLS) masih memiliki kekurangan pengujian di perangkat *mobile* menggunakan PageSpeed Insight dengan nilai rata-rata sedikit diatas ambang batas yang ditentukan, yaitu 0,177. Hal tersebut berkaitan dengan pemuatan gambar yang berdampak pada pergeseran tata letak (*layout*) saat proses *rendering* halaman web.

Ketiga pengujian tersebut juga memberikan gambaran terkait kekurangan yang masih terdapat pada aplikasi web yang dijelaskan pada penilaian metrik *largest contentful paint* (LCP) yang hanya satu

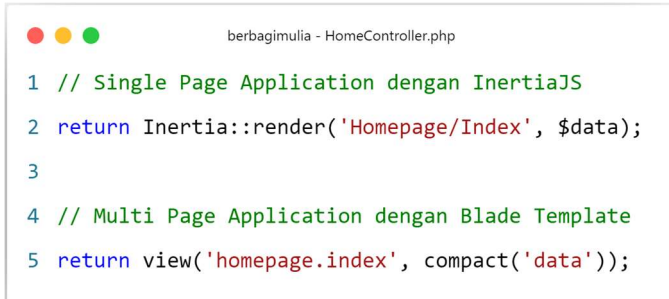
pengujian saja yang lulus ambang batas penilaian performa web yang baik. Dua pengujian lainnya mendapatkan nilai rata-rata LCP lebih dari 2500 ms bahkan yang tertinggi mencapai 5200 ms pada pengujian menggunakan PageSpeed Insight di perangkat *mobile*. Hal tersebut dikarenakan adanya penerapan fitur *lazy loading* pada pemuatan gambar agar proses *rendering* halaman dapat diselesaikan lebih cepat, sehingga konten selain gambar dapat ditampilkan terlebih dahulu untuk mengatasi tampilan kanvas putih saat proses pemuatan halaman berlangsung.

#### **F. Analisis Penerapan *Single Page Application***

Analisis penerapan *single page application* dilakukan untuk mengetahui kelebihanannya sebagai metode penyajian halaman web melalui proses komparasi dengan metode penyajian halaman web lainnya, yaitu *multi page application*. Oleh karena itu, penulis membuat sebuah halaman baru berbasis *multi page application* dalam aplikasi web yang sudah dibuat menggunakan *Blade Template* yang merupakan *templating engine* bawaan *framework* Laravel. Beberapa tahapan yang digunakan pada proses komparasi ini antara lain sebagai berikut:

## 1. Analisis Struktur Kode

Berikut ini merupakan perbandingan kode program pada pembuatan *single page application* dan *multi page application* di *framework* Laravel:



```
berbagimulia - HomeController.php  
1 // Single Page Application dengan InertiaJS  
2 return Inertia::render('Homepage/Index', $data);  
3  
4 // Multi Page Application dengan Blade Template  
5 return view('homepage.index', compact('data'));
```

Gambar 4.66 Perbandingan kode program antara SPA dan MPA di *framework* Laravel

Kedua kode program pada gambar diatas menjelaskan bahwa *single page application* pada *framework* Laravel dibuat menggunakan *method render* milik InertiaJS. Sedangkan, dalam pembuatan *multi page application*, *method* yang digunakan adalah *view* milik *framework* Laravel itu sendiri.

Perbedaan keduanya terletak di parameter pertama pada setiap *method* yang digunakan. Saat menggunakan *method view*, parameter pertama merupakan lokasi tampilan halaman web dalam bentuk struktur halaman *HTML* yang lengkap.

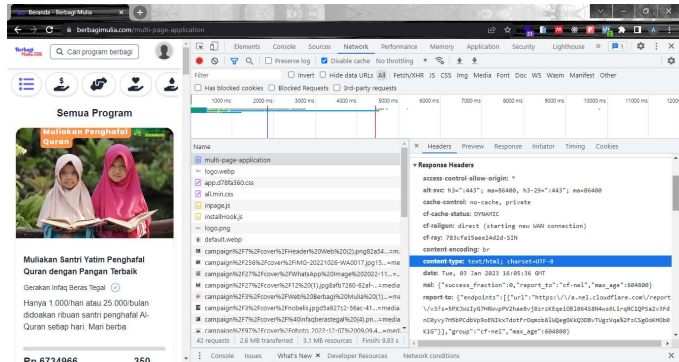
Sedangkan, saat menggunakan *method render* milik InertiaJS, parameter pertama merupakan lokasi komponen dari halaman web yang akan ditampilkan. Komponen tersebut memiliki struktur halaman *HTML* yang tidak lengkap dan dapat digunakan untuk transisi halaman melalui manipulasi *document object model (DOM)* yang menghasilkan *single page application*.

## 2. Analisis Hasil Komparasi

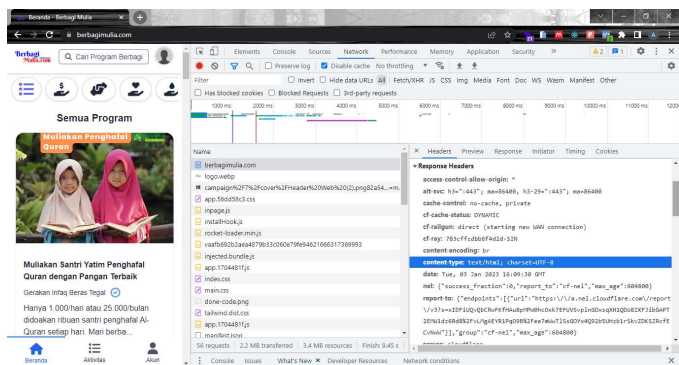
Proses analisis dilakukan menggunakan bantuan Chrome DevTools yang merupakan seperangkat alat pengembangan web di dalam *browser* Google Chrome. Untuk menggunakan fitur Chrome DevTools tersebut, cukup menekan tombol di keyboard dengan kombinasi tombol *Command+Option+C* untuk sistem operasi Mac atau *Control+Shift+C* untuk sistem operasi Windows, Linux, dan ChromeOS pada halaman web yang sedang dibuka.

Berdasarkan hasil analisis yang dilakukan oleh penulis, perbedaan antara *single page application* dengan *multi page application* terletak pada saat proses transisi halaman. Ketika aplikasi web pertama kali dibuka, kedua metode tersebut memiliki

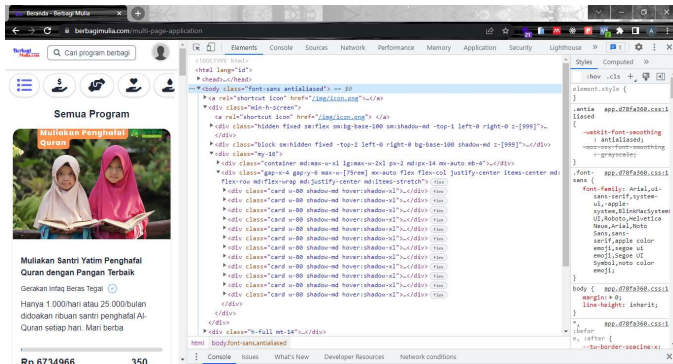
persamaan yaitu melakukan proses *rendering* halaman *HTML* untuk menampilkan antarmuka aplikasi web.



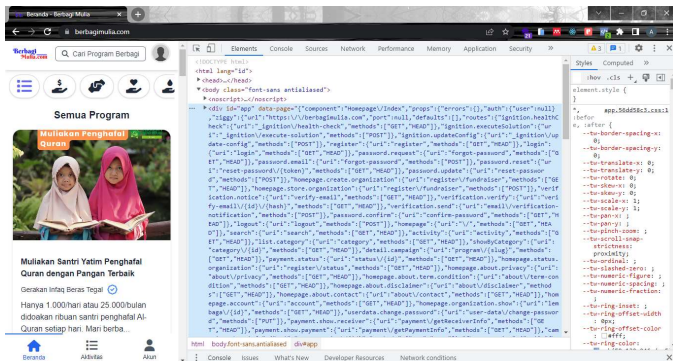
Gambar 4.67 Respons tipe konten dari server saat *multi page application* pertama kali dibuka



Gambar 4.68 Respons tipe konten dari server saat *single page application* pertama kali dibuka



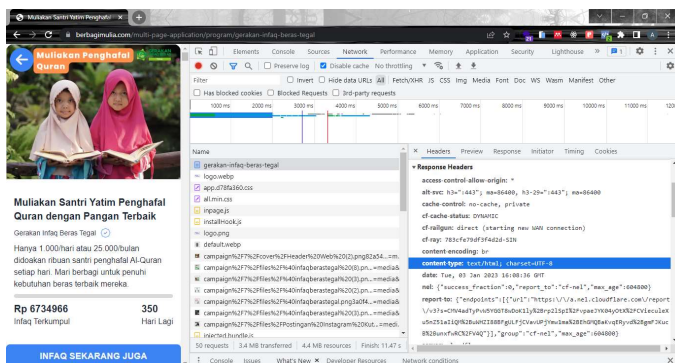
Gambar 4.69 Struktur kode HTML saat multi page application pertama kali dibuka



Gambar 4.70 Struktur kode HTML saat single page application pertama kali dibuka

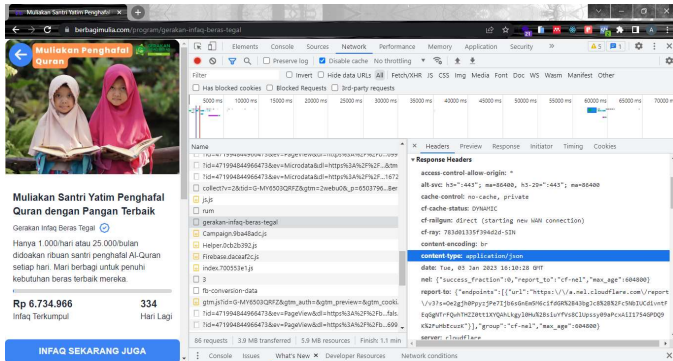
Gambar 4.67 dan Gambar 4.68 diatas menjelaskan bahwa ketika aplikasi web pertama kali dibuka, kedua metode penyajian halaman web tersebut menerima respons dengan tipe konten berupa *text/html* dari server yang dapat di render oleh client atau browser

*browser* dalam menampilkan antarmuka aplikasi web. Sedangkan, Gambar 4.69 dan Gambar 4.70 menjelaskan perbedaan struktur kode *HTML* pada *multi page application* dan *single page application* saat pertama kali dibuka. Perbedaannya terletak pada *single page application* terdapat informasi data semua halaman yang digunakan pada aplikasi web ini dan diberikan oleh *server* saat aplikasi web pertama kali dibuka. Informasi tersebut berlokasi di *tag div* dengan atribut *id* bernama *app* yang menjadi titik pemasangan dalam proses manipulasi *DOM*. Hal tersebut sesuai dengan proses pengkodean aplikasi web ini pada Gambar 4.35 dan Gambar 4.36.



Gambar 4.71 Respons tipe konten dari *server* saat transisi halaman pada *multi page application*



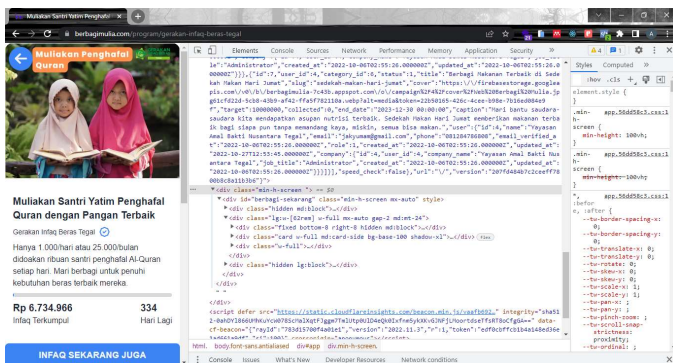


Gambar 4.72 Response tipe konten dari server saat transisi halaman pada *single page application*

```

Headers Preview Response Initiator Timing Cookies
{
  "component": "Homepage/Campaign", ...
  component: "Homepage/Campaign"
  props: {errors: {}, auth: {user: null}, ziggy: {url: "https://berbagi.mia.com/program/gerakan-infaq-beras-tegal"}
  url: "https://berbagi.mia.com/program/gerakan-infaq-beras-tegal"
  version: "207fd48b7c2cefff7800b8c8a11b3b6"
}
    
```

Gambar 4.73 Preview dari respons tipe konten saat transisi halaman pada *single page application*



Gambar 4.74 Struktur kode hasil manipulasi DOM dalam proses transisi halaman di *single page application*

Gambar 4.71 dan Gambar 4.72 diatas menjelaskan proses transisi halaman yang menjadi titik perbedaan antara *multi page application* dengan *single page application*. Saat terjadi transisi halaman, pada Gambar 4.71 menjelaskan sisi *client* dari *multi page application* sepenuhnya menerima struktur kode halaman *HTML* secara lengkap dari sisi *server*. Hal tersebut menyebabkan proses pemuatan ulang dari antarmuka aplikasi web yang sedang ditampilkan karena terjadi pergantian struktur kode halaman *HTML* yang baru. Hal tersebut menyebabkan adanya pemakaian data transfer yang lebih banyak, seperti yang ditampilkan pada Gambar 4.71 sebanyak 3,4 MB.

Berbeda dengan *single page application* yang dijelaskan pada Gambar 4.72, sisi *client* atau dalam hal ini merupakan *web browser* akan secara dinamis melakukan perubahan tampilan halaman web ketika terjadi transisi halaman. Saat proses transisi halaman, *client* akan mengirimkan *http request* ke *server* dalam bentuk konten yang dapat diterima berupa *application/json* melalui *XMLHttpRequest* (XHR). Setelah itu, *server* akan memberikan sebuah *http response* berupa sekumpulan data berbentuk *JSON*.

Didalam data *JSON* tersebut berisikan nama komponen dan alamat *URL* dari halaman yang dituju dan kemudian akan dicocokkan dengan sekumpulan *data-page* yang terdaftar pada halaman *HTML* yang sedang ditampilkan, seperti yang terlihat pada Gambar 4.70 dan Gambar 4.73. Setelah itu, proses perubahan tampilan pun terjadi melalui proses manipulasi *DOM* yang menjadi salah satu kemampuan Javascript.

Javascript *DOM* memungkinkan adanya proses perubahan tampilan aplikasi web secara dinamis tanpa diperlukan proses pemuatan ulang dengan melakukan manipulasi dari halaman *HTML* yang sedang ditampilkan, seperti penambahan elemen baru, perubahan atribut, hingga mengubah *style* elemen yang sudah ada. Hasil akhir dari manipulasi *DOM* ini dapat dilihat pada Gambar 4.74 yang merupakan struktur kode *HTML* yang baru dari antarmuka aplikasi web yang ditampilkan dalam proses transisi halaman di *single page application*.

Dampak dari manipulasi *DOM* ini adalah perubahan tampilan antarmuka aplikasi web tanpa proses pemuatan ulang dari dokumen *HTML* yang

sedang ditampilkan. Hal tersebut memberikan efektifitas dalam penyajian antarmuka aplikasi web dengan mengurangi total data transfer yang terpakai selama penyajian halaman web berlangsung. Seperti yang terlihat pada Gambar 4.68, total data transfer saat pertama kali *single page application* dibuka adalah 2,4 MB dengan *total request* yang terjadi sebanyak 56 *request*. Sedangkan, saat proses transisi halaman yang terjadi pada Gambar 4.73 tertulis total pemakaian data transfer sebanyak 3,9 MB dengan *total request* yang terjadi adalah 86 *request*. Artinya, selisih pemakaian data transfer saat transisi halaman adalah 1,5 MB saja dengan selisih *total request* sebanyak 30 *request* yang terjadi.

Pada *single page application* terjadi penambahan *total request* dan pemakaian data transfer yang mengindikasikan bahwa aplikasi web tidak melakukan pemuatan ulang dari halaman HTML yang sedang ditampilkan. Hal tersebut dibuktikan dengan perbandingan Gambar 4.67 dan Gambar 4.71, terjadi proses pergantian halaman HTML yang berbeda sehingga menghasilkan *total request* dan pemakaian data transfer yang berbeda pula.

Berdasarkan hasil analisis diatas, dapat disimpulkan bahwa *single page application* dapat menyelesaikan ketidakefektifan *multi page application* dalam menyajikan halaman web yang selalu melakukan pemuatan ulang setiap terjadi interaksi oleh pengguna pada antarmuka aplikasi web yang sedang ditampilkan dan berdampak pada total data transfer yang terpakai.

## BAB V

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Berdasarkan hasil penelitian yang dibahas pada bab sebelumnya, dapat diperoleh beberapa kesimpulan yang diambil dari penelitian ini adalah sebagai berikut:

1. Dalam proses analisis penerapan *single page application* menyimpulkan bahwa *single page application* dapat menyelesaikan ketidakefektifan *multi page application* dalam menyajikan halaman web yang selalu melakukan pemuatan ulang setiap terjadi interaksi oleh pengguna pada antarmuka aplikasi web yang sedang ditampilkan. Proses analisis dilakukan dengan membandingkan kedua metode penyajian halaman tersebut saat proses transisi halaman web berlangsung,
2. Dalam proses analisis performa, dilakukan pengujian performa web yang dilakukan sebanyak lima kali melalui perhitungan menggunakan PageSpeed Insight dan GT Metrix. Hasil pengujian menunjukkan pada PageSpeed Insight mendapatkan nilai rata-rata 94,8 di perangkat *desktop* dan 73,8 di perangkat *mobile*. Sedangkan, hasil pengujian menggunakan GT Metrix didapatkan nilai rata-rata 75,8%,

3. Dalam proses pengkodean, penggunaan Laravel Breeze memiliki peran penting dalam mengintegrasikan beberapa teknologi untuk memudahkan proses pengembangan aplikasi web dikarenakan antara Laravel sebagai *server-side* dan ReactJS sebagai *client-side* dapat berkomunikasi langsung melalui perantara InertiaJS, sehingga tidak membutuhkan REST API
4. Dalam proses pengujian berbasis *black box testing* mendapatkan total persentase kesesuaian antara hasil yang diharapkan dan hasil pengujian sebanyak 100%. Sehingga, disimpulkan bahwa aplikasi galang dana berbasis web telah berhasil dibuat dan telah bekerja sesuai dengan harapan pengembang dengan fitur yang lengkap, mulai dari pengelolaan program oleh Admin Lembaga hingga proses pembayaran oleh *User* sebagai donatur melalui metode pembayaran yang disediakan, seperti transfer bank dan Midtrans *payment gateway*.
5. Dalam proses pengujian berbasis *user acceptance test* menyimpulkan bahwa aplikasi galang dana berbasis web telah berhasil dibuat dengan kualitas sangat baik pada fitur yang tersedia di *role Super Admin* dan Admin Lembaga dengan persentase sebanyak 100%

dari total responden yang menjawab. Sedangkan, kualitas pada fitur yang tersedia di *role User* sebagai donatur, setelah dilakukan proses pembobotan didapatkan nilai baik di empat fitur utama dan nilai kurang baik pada kedua fitur metode pembayaran. Hal tersebut disebabkan kedua fitur tersebut pada penerapannya hanya dapat dipilih salah satu antara metode pembayaran manual atau otomatis. Sehingga, saat pengisian kuesioner berlangsung, banyak responden yang memilih pilihan jawaban belum mencoba pada salah satu dari dua fitur metode pembayaran tersebut.

## **B. Saran**

Beberapa saran penulis bagi penelitian selanjutnya adalah sebagai berikut:

1. Lakukan untuk optimasi gambar sehingga dapat meningkatkan penilaian pada metrik *Largest Contentful Paint* (LCP) yang berdampak pada performa aplikasi web,
2. Antarmuka aplikasi ini masih berbasis web yang dibangun menggunakan *library* ReactJS, sehingga jika dibutuhkan pengembangan aplikasi *mobile* dapat dimigrasikan ke dalam *framework* React Native.



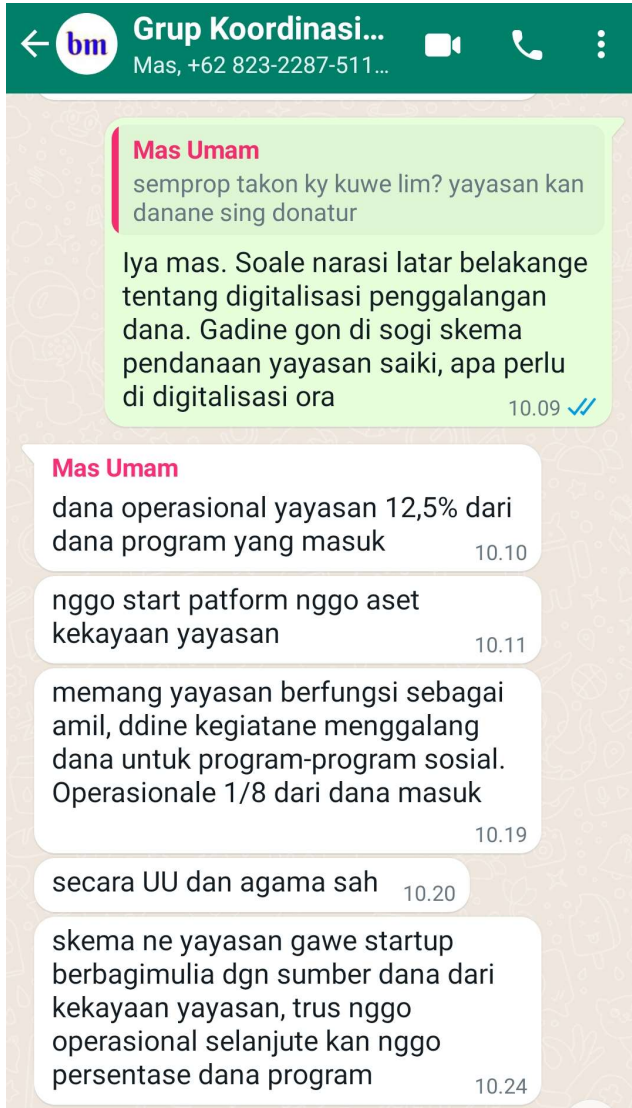
## DAFTAR PUSTAKA

- Damayanti, E. (2019). Sistem Informasi Penjualan Obat Pertanian Berbasis Web pada Toko BUTANI Blora. *Walisongo Journal of Information Technology*, 1(2), 161. <https://doi.org/10.21580/wjit.2019.1.2.4520>
- Ham, H. (2019). *Apa itu React.js?* School of Computer Science Binus University. <https://socs.binus.ac.id/2019/12/30/apa-itu-react-js/>
- Hargianto, T. (2018). *Pembuatan Aplikasi Web Untuk Manajemen Proyek Software Dalam Tim Berbasis Single Page Application Menggunakan Library ReactJS*.
- Haviluddin. (2011). Memahami Penggunaan UML ( Unified Modelling Language ). *Memahami Penggunaan UML (Unified Modelling Language)*, 6(1), 1–15.
- Hermanto, B., Yusman, M., & Nagara, N. (2019). Sistem Informasi Manajemen Keuangan pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel. *Jurnal Komputasi*, 7(1), 17–26. <https://doi.org/10.23960/komputasi.v7i1.2051>
- Hidayati, T. Y. (2020). Rancang Bangun Sistem Informasi Persewaan Alat-alat Outdoor Berbasis Web. In *Jurnal Sains Komputer dan Teknologi Informasi* (Vol. 3, Issue 1).
- Husada, F. R. K. (2019). *Implementasi Framework Laravel Untuk Membangun Sistem Administrasi Pembayaran Air Pada PPAB Perum Dolog Dengan Metode Waterfall* [Universitas Semarang]. <https://eskripsi.usm.ac.id/detail-G21A-405.html>
- Julianto, A. (2020). *Perancangan Ulang Desain Antarmuka Aplikasi Berbasis Web Dengan Menggunakan Metode User Centered Design (Studi Kasus: Petshopgroisir)* [Universitas Komputer Indonesia]. <https://elibrary.unikom.ac.id/id/eprint/3984/>
- Kurniawan, T. B., & Syarifuddin. (2020). Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman Pada Cafeteria No Caffe di Tanjung Balai Karimun

- Menggunakan Bahasa Pemrograman PHP dan MySQL. *Jurnal Tikar*, 1(2), 192–206. [https://ejurnal.universitaskarimun.ac.id/index.php/teknik\\_informatika/article/download/153/121](https://ejurnal.universitaskarimun.ac.id/index.php/teknik_informatika/article/download/153/121)
- Latukolan, M. L. A., Arwan, A., & Ananta, M. T. (2019). Pengembangan Sistem Pemetaan Otomatis Entity Relationship Diagram Ke Dalam Database. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(4), 4058–4065. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5117>
- Mukhiya, S. K., & Hoang Hung, K. (2018). *An Architectural Style for Single Page Scalable Modern Web Application*. 5(4), 6–13. <https://ijrra.net/Vol5issue4/IJRRRA-05-04-02.pdf>
- Munthe, R., Santosa, I., & Ferdiana, R. (2016). Evaluasi Proses Uat (User Acceptance Testing) Dalam Pengembangan Produk Dengan Pendekatan Pengujian Pragmatis. *Universitas Gadjah Mada*.
- Nurul Bariyah, N. O. (2016). Strategi Penghimpunan Dana Sosial Ummat Pada Lembaga-Lembaga Fillantrofi Di Indonesia (Studi Kasus Dompot Peduli Ummat Daarut Tauhid, Dompot Dhuafa Republika, Baznas, Dan Bazis Dki Jakarta). *Jurnal Studi Ekonomi Dan Bisnis Islam*, 1.
- Pratiwi, P. R. (2010). *Strategi penggalangan dana melalui program layanan jemput zakat laziz PP Muhammadiyah* [UIN Syarif Hidayatullah Jakarta]. <http://repository.uinjkt.ac.id/dspace/handle/123456789/21232>
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach, Seventh Edition*. <https://doi.org/10.1145/336512.336521>
- Pujihastuti, I. (2010). Prinsip Penulisan Kuesioner Penelitian. *Jurnal Agribisnis Dan Pengembangan Wilayah*, 2(1), 43–56.
- Rachmasari, Y., Nulhaqim, S. A., & Apsari, N. C. (2016). Penerapan Strategi Fundraising Di Save the Children Indonesia (Fundraising Strategy Implementation in Save

- the Children Indonesia). *Share: Social Work Journal*, 6(1). <https://doi.org/10.24198/share.v6i1.13148>
- Riyadli, H., Arliyana, A., & Saputra, F. E. (2020). Rancang Bangun Sistem Informasi Keuangan Berbasis WEB. *Jurnal Sains Komputer Dan Teknologi Informasi*, 3(1), 98–103. <https://doi.org/10.33084/jsakti.v3i1.1770>
- Santoso, M. F. (2021). Teknik Single Page Application (SPA) Layout Web Dengan Menggunakan React Js Dan Bootstrap. *Jurnal Khatulistiwa Informatika*, 9(2), 107–114. <https://doi.org/10.31294/jki.v9i2.11357>
- Septiawan, H. F. (2019). *Pengembangan E-Learning Menggunakan Laravel Dan React*. Sekolah Tinggi Manajemen Informatika dan Komputer Yogyakarta.
- Sinaga, G. R. U. (2020). *Sistem Informasi Layanan Pemesanan Pada Restoran Cindelas Kota Medan Dengan Menggunakan Framework Laravel* [Universitas Islam Negeri Sumatera Utara]. <http://repository.uinsu.ac.id/12623/>
- Susanti, S., Junianto, E., & Rachman, R. (2017). Implementasi Framework Laravel Pada Aplikasi Pengolah Nilai Akademik Berbasis Web. *Jurnal Informatika (JI) UBSI*, 4(1). <https://doi.org/10.31311/JI.V4I1.1562>
- Wahid, A. A. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK, November*, 1–5.
- Walton, P. (2022). *Web Vital*. <https://web.dev/vitals>
- Wijaya, Y. D., & Astuti, M. W. (2021). Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions. *Jurnal Digital Teknologi Informasi*, 4(1), 22. <https://doi.org/10.32502/digital.v4i1.3163>

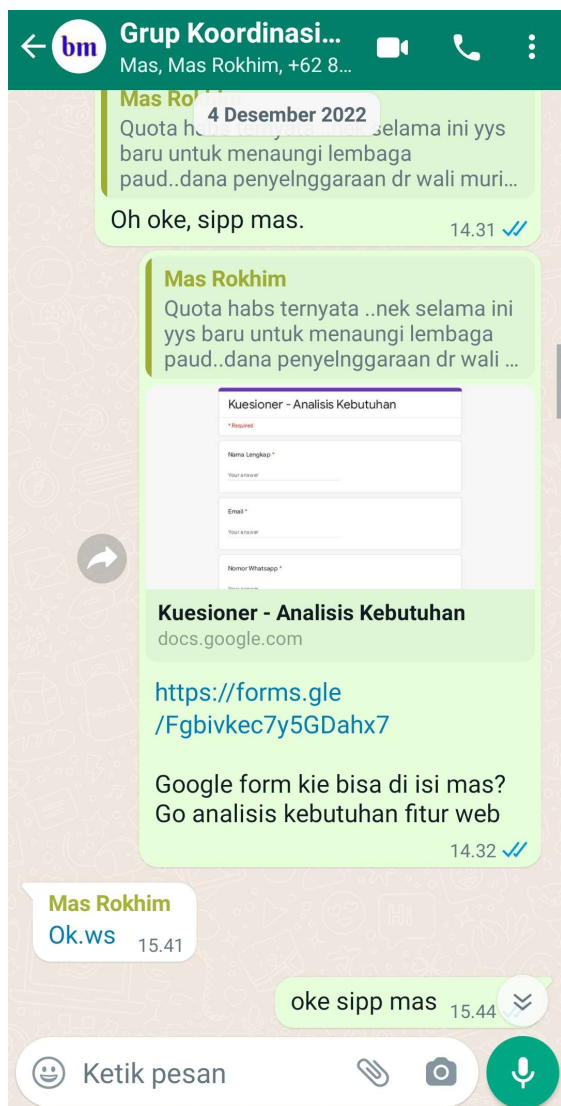
Lampiran 1 Komunikasi awal terkait kebutuhan sistem melalui percakapan di pesan WhatsApp



Lampiran 2 Pembentukan kantor sekretariat bersama sebagai bentuk keseriusan membangun *platform digital* Berbagi Mulia



### Lampiran 3 Percakapan pesan Whatsapp terkait pengisian kuesioner untuk kebutuhan fitur aplikasi web



## Lampiran 4 Hasil kuesioner analisis kebutuhan

### Kuesioner - Analisis Kebutuhan

1 jawaban

[Publikasikan analytics](#)

Nama Lengkap

1 jawaban

Nurokhim

Email

1 jawaban

Nurokhimbekamslawi@gmail.com

Nomor Whatsapp

1 jawaban

082322875113

Jabatan di Yayasan Amal Bakti Nusantara

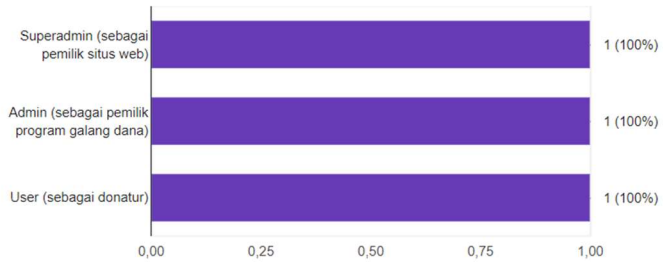
1 jawaban

Pembina

Apa saja role pengguna terdaftar yang dibutuhkan oleh sistem?  
(berkaitan dengan pembatasan akses halaman tertentu)

 Salin

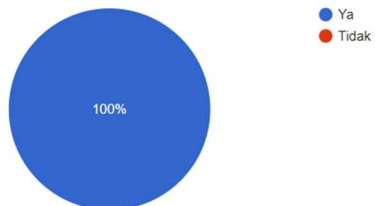
1 jawaban



Apakah diperlukan fitur registrasi akun bagi para penggalang dana dan donatur?

 Salin

1 jawaban



Apakah diperlukan fitur pengelolaan program galang dana yang ditampilkan?

 Salin

1 jawaban

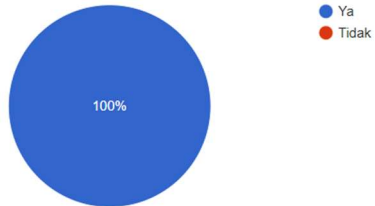




Apakah diperlukan fitur pengelolaan laporan berita dari program galang dana?

 Salin

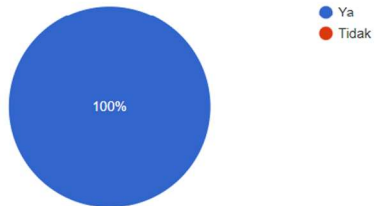
1 jawaban



Apakah diperlukan fitur pengelolaan donasi hingga proses pembayaran oleh donatur?

 Salin

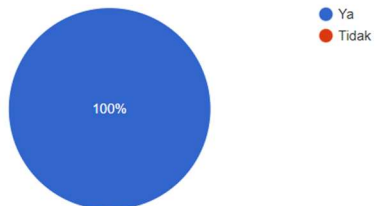
1 jawaban



Apakah diperlukan fitur promosi program galang dana oleh akun donatur yang terdaftar?

 Salin

1 jawaban



Apakah diperlukan fitur penarikan dana yang dilakukan oleh lembaga pemilik program galang dana?

 Salin

1 jawaban



● Ya  
● Tidak

Apakah diperlukan fitur transfer bank sebagai metode pembayaran yang diverifikasi secara manual?

 Salin

1 jawaban



● Ya  
● Tidak

Apakah diperlukan fitur bank virtual account sebagai metode pembayaran yang diverifikasi secara otomatis menggunakan Midtrans Payment Gateway?

 Salin

1 jawaban

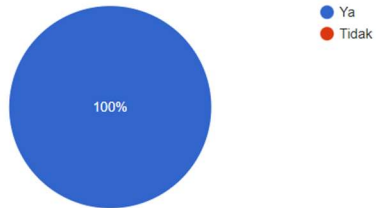


● Ya  
● Tidak

Apakah diperlukan fitur integrasi dengan layanan *marketing tools*, seperti Facebook Ads?

 Salin

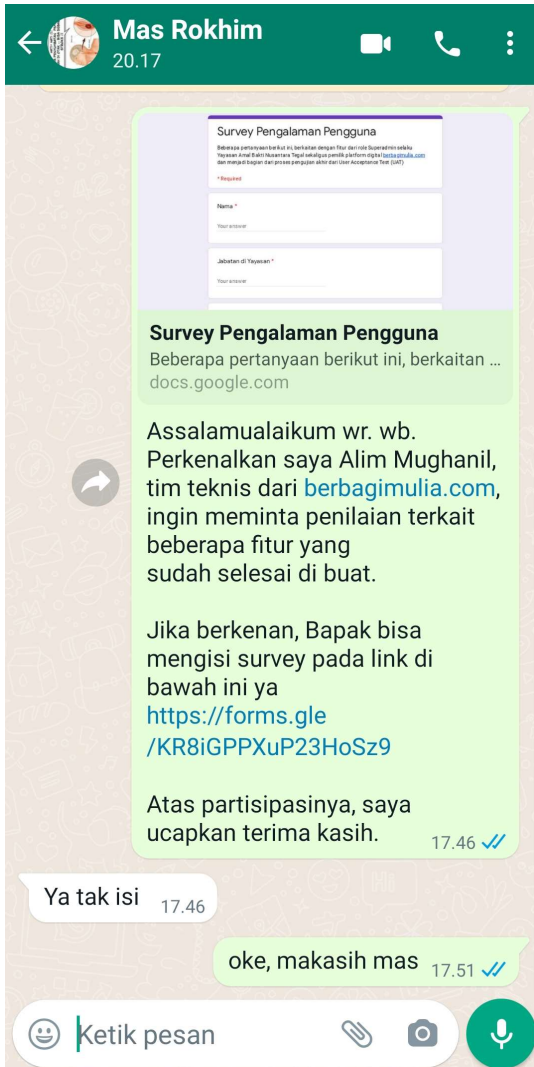
1 jawaban



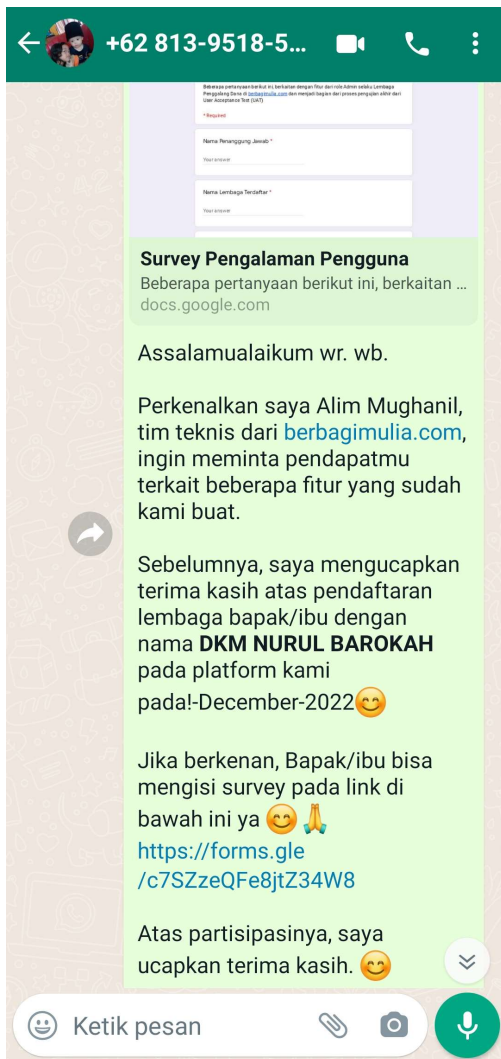
Konten ini tidak dibuat atau didukung oleh Google. [Laporkan Penyalahgunaan](#) - [Persyaratan Layanan](#) - [Kebijakan Privasi](#)

Google Formulir

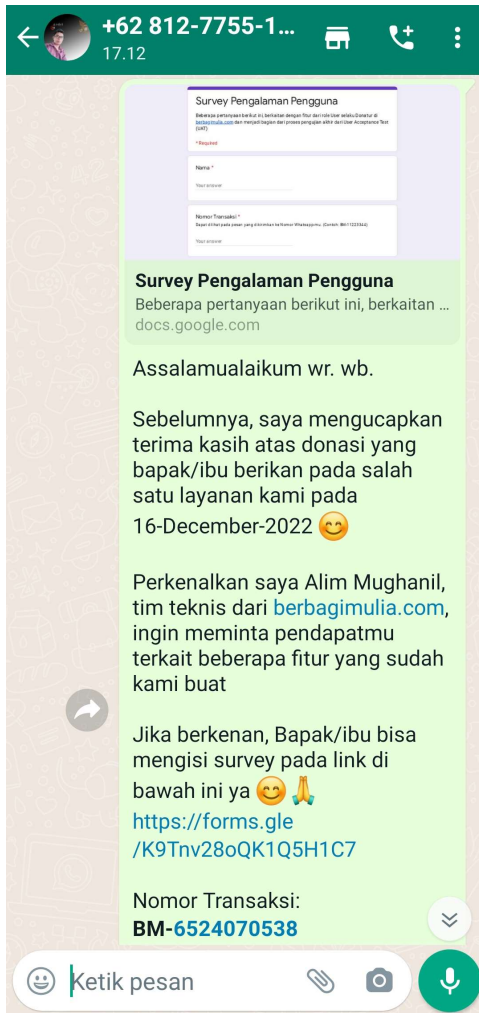
Lampiran 5 Percakapan pesan Whatsapp terkait survey pengalaman pengguna sebagai bagian dari proses UAT *Super Admin*



## Lampiran 6 Percakapan pesan Whatsapp terkait survey pengalaman pengguna sebagai bagian dari proses UAT Admin Lembaga



## Lampiran 7 Percakapan pesan Whatsapp terkait survey pengalaman pengguna sebagai bagian dari proses UAT *User* sebagai donatur



## Lampiran 8 Hasil kuesioner pengujian UAT *Super Admin*

### User Acceptance Test (UAT) - Super Admin

1 jawaban

[Publikasikan analytics](#)

Nama

1 jawaban

Nurokhim

Jabatan di Yayasan


1 jawaban

Pembina

Apakah fitur Login untuk Akun Superadmin Selaku Pemilik Platform Digital Berbagi Mulia telah bekerja dengan baik?

1 jawaban

[Salin](#)



● Sangat Baik  
● Baik  
● Kurang Baik  
● Tidak Baik  
● Belum Mencoba

Kategori	Persentase
Sangat Baik	100%
Baik	0%
Kurang Baik	0%
Tidak Baik	0%
Belum Mencoba	0%

Apakah fitur Verifikasi Program Galang Dana Baru telah bekerja dengan baik?

 Salin

1 jawaban



- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Verifikasi Akun Admin selaku Lembaga Galang Dana Baru telah bekerja dengan baik?

 Salin

1 jawaban



- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Verifikasi Pembayaran Baru Secara Manual telah bekerja dengan baik?

 Salin

1 jawaban



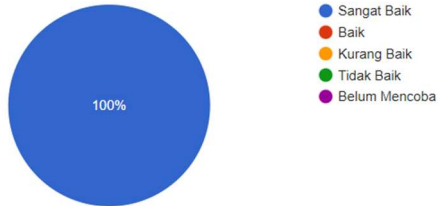
- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba



Apakah fitur Verifikasi Penarikan Dana Baru telah bekerja dengan baik?

 Salin

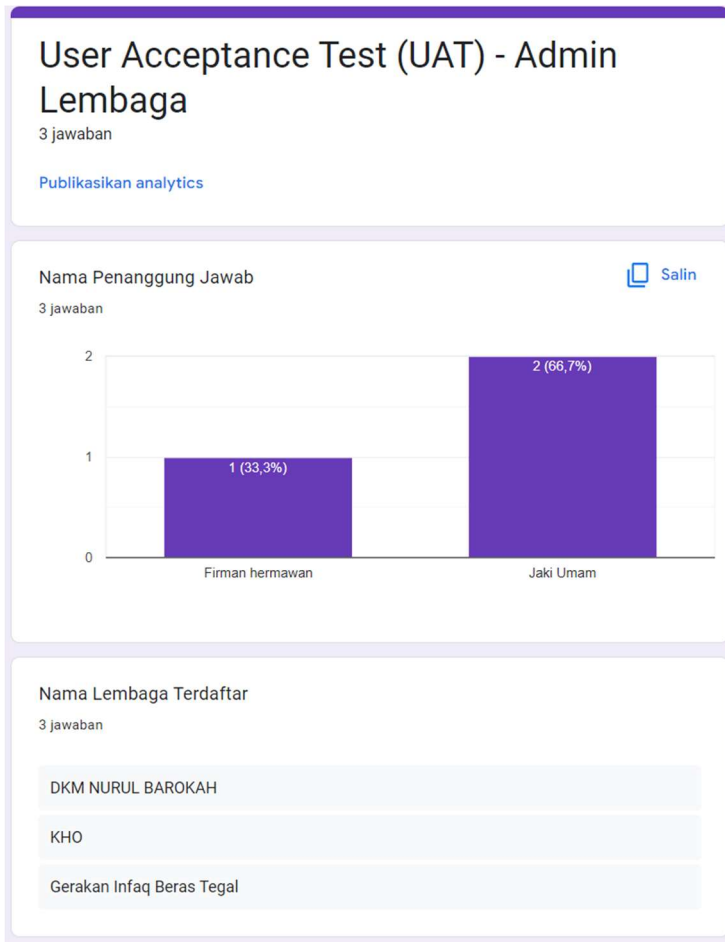
1 jawaban



Konten ini tidak dibuat atau didukung oleh Google. [Laporkan Penyalahgunaan](#) - [Persyaratan Layanan](#) - [Kebijakan Privasi](#)

Google Formulir

## Lampiran 9 Hasil kuesioner pengujian UAT Admin Lembaga



Apakah fitur Registrasi Akun Admin selaku Lembaga Penggalang Dana telah bekerja dengan baik?

 Salin

3 jawaban

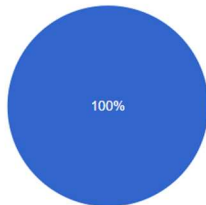


- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Login untuk Akun Admin selaku Lembaga Penggalang Dana telah bekerja dengan baik?

 Salin

3 jawaban



- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Pengelolaan Program Galang Dana telah bekerja dengan baik?

 Salin

3 jawaban



- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Pengelolaan Laporan Berita Program Galang Dana telah bekerja dengan baik?

 Salin

3 jawaban



- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Mengintegrasikan Facebook Ads dengan Program Galang Dana telah bekerja dengan baik?

 Salin

3 jawaban

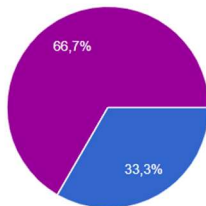


- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

Apakah fitur Penarikan Dana Terkumpul dari Program Galang Dana telah bekerja dengan baik?

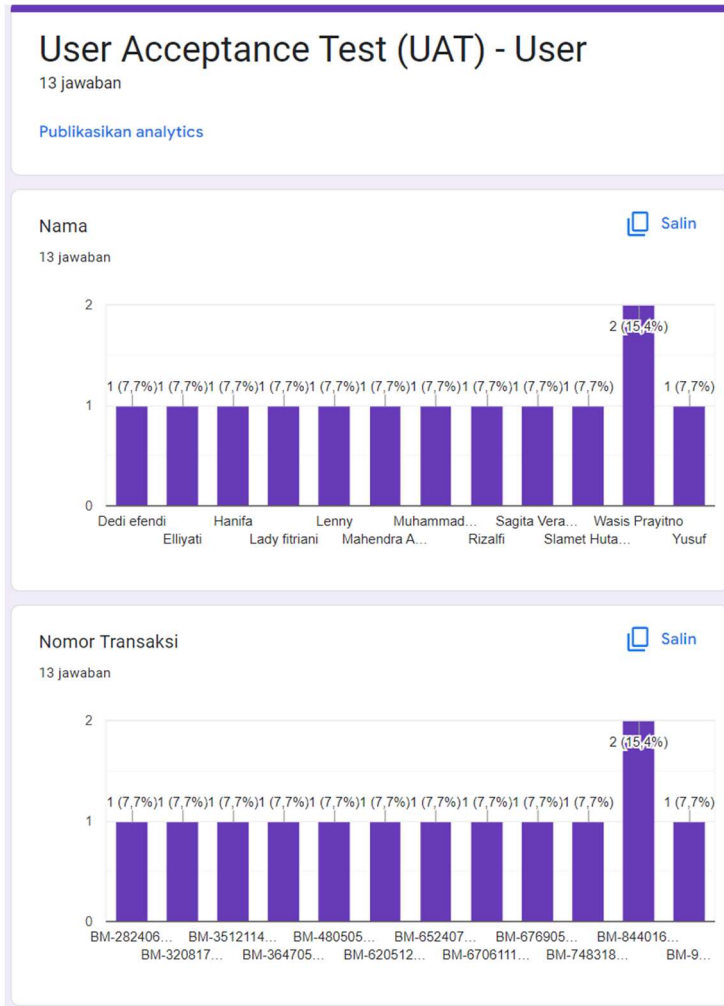
 Salin

3 jawaban



- Sangat Baik
- Baik
- Kurang Baik
- Tidak Baik
- Belum Mencoba

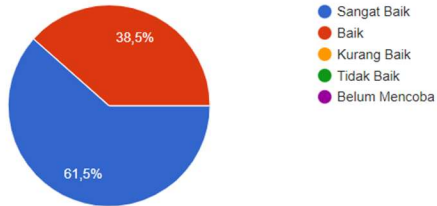
## Lampiran 10 Hasil kuesioner pengujian UAT *User* sebagai donatur



Apakah fitur Pembuatan Donasi Baru telah bekerja dengan baik?

 Salin

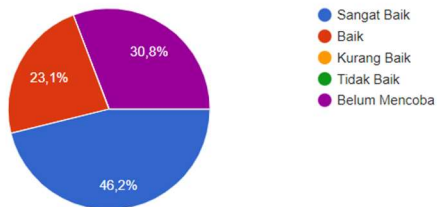
13 jawaban



Apakah fitur Pembayaran dengan Transfer Bank telah bekerja dengan baik?

 Salin

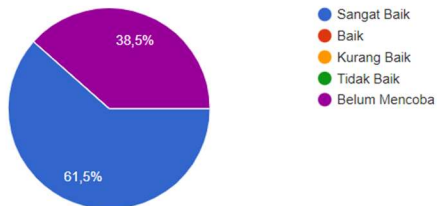
13 jawaban



Apakah fitur Pembayaran dengan Bank Virtual Account atau Gopay telah bekerja dengan baik?

 Salin

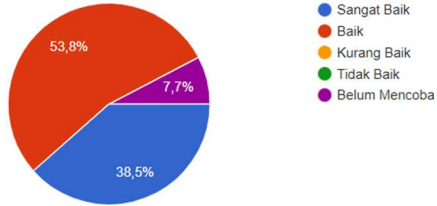
13 jawaban



Apakah fitur Registrasi Akun User atau Donatur telah bekerja dengan baik?

 Salin

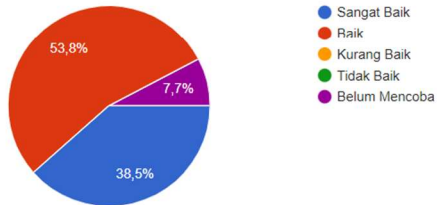
13 jawaban



Apakah fitur Login untuk Akun User atau Donatur telah bekerja dengan baik?

 Salin

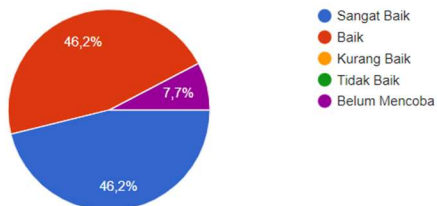
13 jawaban



Apakah fitur Promosi Program Galang Dana oleh akun User atau Donatur telah bekerja dengan baik?

 Salin

13 jawaban



## RIWAYAT HIDUP

### A. Identitas Diri

1. Nama Lengkap : Alim Mughanil Karim
2. Tempat & Tgl. Lahir : Tegal, 23 Mei 2000
3. Alamat Rumah : Jl. Darul Hasanah No. 14,  
Tegalandong, Lebaksiu,  
Tegal, Jawa Tengah
4. HP : 08983936489
5. Email : alim.mughanil23@gmail.com

### B. Riwayat Pendidikan

1. Pendidikan Formal:
  - a. MI NU 01 Tegalandong (2011)
  - b. MTs NU 1 Lebaksiu (2014)
  - c. SMKN 1 Adiwerna (2017)
2. Pendidikan Non-Formal:
  - a. Digital Bootcamp Batch 16 di Camp404 (2022)

Semarang, 22 Desember 2022



Alim Mughanil Karim  
NIM : 1908096035