

TUGAS AKHIR

RANCANG BANGUN *WEBSITE* PADA PONDOK PESANTREN TAHFIDZ AL QURAN EL ‘AMANI PONTIANAK

Diajukan Sebagai Persyaratan untuk Menyelesaikan Program Pendidikan
Diploma III Pada Program Studi Teknik Informatika
Jurusan Teknik Elektro Politeknik Negeri Pontianak



OLEH:

BELLA SUKMA ANDRIANI

3201916054

**PROGRAM STUDI D-III TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI PONTIANAK**

2022

HALAMAN PENGESAHAN

RANCANG BANGUN WEB *WEBSITE* PADA PONDOK PESANTREN TAHFIDZ AL QURAN EL ‘AMANI PONTIANAK

Oleh:

**Bella Sukma Andriani
3201916054**

Tugas akhir ini telah diterima dan disahkan sebagai salah satu syarat untuk menyelesaikan Program Pendidikan Diploma III pada Program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Pontianak.

Disahkan oleh :

**Ketua Jurusan
Teknik Elektro**

**Ketua Program Studi
Teknik Informatika**

**Hasan, S.T., M.T.
NIP. 197108201999031003**

**Suheri, S.T., M.Cs.
NIP. 198307172008121005**

**Mengetahui,
Direktur Politeknik Negeri Pontianak**

**Dr. Ir. H. Muhammad Toasin Asha, M.Si.
NIP. 196112251990111001**

HALAMAN PERNYATAAN
RANCANG BANGUN WEB PORTAL ONE STOP SERVICE
APLIKASI LAYANAN PUBLIK DI LINGKUNGAN PEMERINTAHAN
PROVINSI KALIMANTAN BARAT

Oleh:

Bella Sukma Andriani
3201916054

Dosen Pembimbing :

Muhammad Hasbi, S.T.,M.T
NIP. 197601112014041001

Telah dipertahankan di depan penguji pada tanggal 29 Agustus 2022 dan
Dinyatakan memenuhi syarat sebagai Laporan Tugas Akhir

Dosen Penguji :

Penguji I

Penguji II

Budianingsih, S.T.,M.T.
NIP. 198011022012122003

Lindung Siswanto, S.Kom., M.Eng.
NIP. 198406112019031012

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

Nama : Bella Sukma Andriani
NIM : 3201916054
Jurusan / Program Studi : Teknik Elektro / D-III Teknik Informatika
Judul Proposal : Rancang Bangun *Website* Pada Pondok Pesantren
Tahfidz Al ‘Quran El ‘Amani Pontianak

Menyatakan dengan sebenar-benarnya bahwa penulisan Tugas Akhir ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah Tugas Akhir maupun kegiatan yang tercantum sebagai bagian dari Tugas Akhir ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di Politeknik Negeri Pontianak.

Demikian pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Pontianak, 21 Maret 2022

Yang membuat pernyataan,

Materai
10000

BELLA SUKMA ANDRIANI
NIM. 3201916054

RIWAYAT HIDUP



Biodata Mahasiswa :

Nama Mahasiswa	: Bella Sukma Andriani
NIM	: 3201916054
Tempat / Tanggal Lahir	: Sui. Asam / 12 September 2001
Jenis Kelamin	: Perempuan
Agama	: Islam
Alamat	: Jalan Dr. Wahidin S. Gg. Sejati 2
No. Telepon Rumah / Handphone	: 6289524333618
Email	: bellasukmandriani@gmail.com

ABSTRAK

Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak lembaga pendidikan dan pembelajaran al quran untuk mempersiapkan generasi Islam Qur’aniy, terletak di Jln Dr. Wahidin Sudirohusodo Komplek Batara Indah 1, no 31 RT 001/RW 025 Pontianak Kota, Kota Pontianak. Saat ini Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak masih dalam tahap perkembangan, oleh karena itu diperlukannya sebuah *website* untuk operasional dalam menyebarluaskan informasi terkait Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dan pengelolaan donasi. Selama ini, bantuan donasi yang didapatkan dari lingkungan masyarakat sekitar dan dilakukan secara langsung. Namun, cara tersebut kurang efektif sehingga bantuan donasi yang didapatkan kurang maksimal karena terbatasnya penyampaian informasi yang merata mengenai bantuan.

Metode pengembangan yang digunakan dalam pembuatan *website* ini adalah metode *Waterfall*. Adapun tahapan dalam metode *waterfall* ialah analisis, desain, pembuatan kode program dan pengujian program. Sementara itu, perangkat lunak yang digunakan dalam membangun sistem ini yaitu *Framework Laravel*, serta database menggunakan MariaDB.

Website yang telah dibangun adalah *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak berfungsi untuk mempermudah pengguna dalam mendapatkan informasi yang ada dan mempermudah donatur dalam melakukan donasi.

Kata-kata kunci : Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak, *Website*, *Framework Laravel*

ABSTRACT

Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak an educational institution and learning al quran to prepare the generation of Islamic Quraniy, located on Jln Dr. Wahidin Sudirohusodo Batara Indah Complex 1, no. 31 RT 001/RW 025 Pontianak City, Pontianak City. Currently the Tahfidz Al Quran El 'Amani Islamic Boarding School Pontianak is still in the development stage, therefore a website is needed for operations in disseminating information related to the Tahfidz Al Quran El 'Amani Islamic Boarding School in Pontianak and managing donations. So far, donations have been obtained from the surrounding community and carried out directly. However, this method is less effective so that the donation assistance obtained is not optimal due to the limited distribution of evenly distributed information regarding assistance.

The development method used in making this website is the Waterfall method. The stages in the waterfall method are analysis, design, program code generation and program testing. Meanwhile, the software used in building this system is the Laravel Framework, and the database uses MariaDB.

The website that has been built is the website of the Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak which functions to make it easier for users to get existing information and make it easier for donors to make donations.

Keywords: : Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak, Website, Framework Laravel

PRAKATA

Alhamdulillah puji dan syukur atas kehadiran Allah *Subhanahu wa Ta'ala* yang telah melimpahkan rahmat dan karunia-Nya. Sehingga pada akhirnya penulis berkesempatan dalam menyelesaikan tugas akhir ini dengan baik. Pada kesempatan ini penulis menyampaikan ucapan terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah berperan sehingga dapat terselesaikannya tugas akhir ini, antara lain :

1. Kedua orang tua dan keluarga yang senantiasa selalu memberikan nasihat, doa dan dukungannya selama ini;
2. Bapak Dr.Ir. H. Muhammad. Toasin Asha, M.Si., selaku Direktur Politeknik Negeri Pontianak;
3. Bapak Hasan, S.T., M.T., selaku Ketua Jurusan Teknik Elektro;
4. Bapak Suheri, S.T., M.Cs., selaku Ketua Program Studi D3 Teknik Informatika;
5. Bapak Muhammad Diponegoro, S.Kom., M.Cs. selaku Koordinator Tugas Akhir Program Studi Teknik Informatika;
6. Bapak Muhammad Hasbi, S.T., M.T., selaku dosen pembimbing yang telah banyak membantu, memberikan bimbingan, pengarahan, kritik dan masukan;
7. Ibu Budianingsih, S.T., M.T. selaku dosen penguji 1 dan Bapak Lindung Siswanto, S.Kom., M.Eng. selaku dosen penguji 2 yang telah memberikan kritik dan saran selama proses penyusunan Tugas Akhir ini;
8. Seluruh staff pengajar dan administrasi Politeknik Negeri Pontianak, khususnya di Jurusan Teknik Elektro Program Studi Teknik Informatika;
9. Pihak Pondok Pesantren Tahfidz Quran El 'Amani Pontianak yang sangat membantu memberikan data kepada penulis untuk menunjang Tugas Akhir;
10. Teman-teman Program Studi Teknik Informatika angkatan 2019 khususnya kelas 6B Teknik Informatika yang telah banyak membantu dan memberikan support selama ini dari awal hingga akhir.

Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini tentu masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca yang membangun sangat penulis harapkan untuk menjadi pelajaran di kemudian hari.

Penulis berharap semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi kita semua. Terima kasih.

Pontianak, 30 Agustus 2022

Bella Sukma Andriani

DAFTAR ISI

HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN.....	iii
HALAMAN PERNYATAAN ORISINALITAS.....	iv
ABSTRAK	vi
ABSTRACT	vii
PRAKATA	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian.....	4
1.6.1. Metode Pengumpulan Data.....	4
1.6.1.1. Metode Wawancara	4
1.6.1.2. Metode Studi Literatur	4
1.6.1.3. Metode Observasi	5
1.6.2. Metode Pengembangan Sistem.....	5
1.6.2.1. Metode <i>Waterfall</i>	5
1.7. Sistem Penulisan	7
BAB II DASAR TEORI.....	8
2.1. Tinjauan Pustaka.....	8
2.2. Dasar Teori.....	9
2.2.1. <i>Website</i>	9
2.2.2. HTML (<i>Hypertext Markup Language</i>).....	9
2.2.3. XAMPP	9
2.2.4. MySQL.....	9
2.2.5. PHP (<i>Hypertext Preprocessor</i>).....	10
2.2.6. <i>Bootstrap</i>	10
2.2.7. Java Script	10
2.2.8. <i>Framework</i> Laravel	11

BAB III PERANCANGAN SISTEM	13
3.1. Gambaran Umum.....	13
3.2. Analisis Kebutuhan.....	13
3.2.1. Analisis Kebutuhan Data	13
3.2.2. Kebutuhan Perangkat Keras.....	13
3.2.3. Kebutuhan Perangkat Lunak	14
3.3. Perancangan Sistem	14
3.3.1. <i>Use Case</i>	14
3.3.2. Skenario <i>Use Case</i>	17
3.4. Perancangan Basis Data	29
3.4.1. Relasi Tabel.....	29
3.4.2. Struktur Tabel	30
3.5. Perancangan Antar Muka.....	34
3.5.1. <i>Mockup</i> Tampilan Halaman Utama	34
3.5.2. <i>Mockup</i> Halaman Profil	34
3.5.3. <i>Mockup</i> Halaman Guru dan Santri.....	35
3.5.5. <i>Mockup</i> Halaman Donasi	36
3.5.5. <i>Mockup</i> Halaman Kontak Kami	37
3.5.6. <i>Mockup</i> Halaman <i>Login</i>	38
3.5.7. <i>Mockup</i> Halaman <i>Dashboard</i> Admin.....	38
3.5.8. <i>Mockup</i> Halaman <i>Dashboard</i> Data Guru dan Santri.....	38
3.5.9. <i>Mockup</i> Halaman <i>Dashboard</i> Program.....	39
3.5.10. <i>Mockup</i> Halaman <i>Dashboard</i> Kelola Donasi	40
3.5.11. <i>Mockup</i> Halaman <i>Dashboard</i> Pesan Kritik dan Saran	42
3.5.12. <i>Mockup</i> Halaman <i>Dashboard</i> Profile Admin	43
BAB IV HASIL DAN PEMBAHASAN.....	44
1.1. Implementasi <i>Database</i>	44
1.2. Implementasi Halaman <i>Login</i>	45
1.3. Implementasi Halaman <i>Dashboard</i> Admin	46
1.4. Implementasi Halaman Menu Guru dan Santri.....	48
1.5. Implementasi Halaman Menu Program	53
1.6. Implementasi Halaman Menu Laporan Keuangan.....	58
1.8. Implementasi Halaman <i>User</i>	65
1.9. Implementasi Halaman Profil	66
1.10. Implementasi Tampilan Halaman Guru dan Santri.....	67

1.11. Implementasi Halaman Program.....	68
1.12. Implementasi Halaman Donasi	69
1.13. Implementasi Halaman Kontak Kami	70
1.14. Pengujian Sistem.....	71
BAB V PENUTUP	73
5.1. Kesimpulan	73
5.2. Saran	73
DAFTAR PUSTAKA	74

DAFTAR GAMBAR

Gambar 1.1. Metode <i>Waterfall</i>	5
Gambar 3.1. <i>Use Case Diagram</i>	14
Gambar 3.2. Rancangan <i>Database</i>	30
Gambar 3.3. <i>Mockup</i> Halaman Utama.....	34
Gambar 3.4. <i>Mockup</i> Halaman Profil	35
Gambar 3.5. <i>Mockup</i> Halaman Guru dan Santri	35
Gambar 3.6. <i>Mockup</i> Halaman Program.....	36
Gambar 3.7. <i>Mockup</i> Halaman <i>Login</i>	38
Gambar 3.9. <i>Mockup</i> Halaman Kontak Kami	48
Gambar 3.10. <i>Mockup</i> Halaman <i>Dashboard</i> Admin	38
Gambar 3.12. <i>Mockup</i> Halaman <i>Dashboard</i> Galeri.....	39
Gambar 3.16. <i>Mockup</i> Halaman <i>Dashboard</i> Guru dan Santri	39
Gambar 3.17. <i>Mockup</i> Halaman <i>Dashboard</i> Program.....	40
Gambar 3.18. <i>Mockup</i> Halaman <i>Dashboard</i> Kategori.....	41
Gambar 3.19. <i>Mockup</i> Halaman <i>Dashboard</i> Laporan Keuangan	41
Gambar 3.21. <i>Mockup</i> Halaman <i>Dashboard</i> Donasi	42
Gambar 4.1. Implementasi <i>Database</i>	47
Gambar 4.2. Implementasi Halaman <i>Login</i>	45
Gambar 4.3. <i>Source Code</i> Menampilkan Halaman <i>Login</i>	46
Gambar 4.4. Implementasi Halaman <i>Dashboard</i> Admin.....	47
Gambar 4.5. <i>Source Code</i> Menampilkan Halaman <i>Dashboard</i> Admin.....	47
Gambar 4.25. Implementasi Halaman <i>Dashboard</i> Guru dan Santri	48
Gambar 4.26. <i>Source Code</i> Menampilkan Halaman Guru dan Santri	48
Gambar 4.27. Implementasi Halaman <i>Dashboard</i> Guru dan Santri	49
Gambar 4.28. Implementasi Halaman <i>Dashboard</i> Guru dan Santri Edit	51
Gambar 4.30. Implementasi Halaman <i>Dashboard</i> Hapus	53
Gambar 4.31. Implementasi Halaman <i>Dashboard</i> Program.....	54
Gambar 4.32. <i>Source Code</i> Menampilkan Halaman <i>Dashboard</i> Program.....	54
Gambar 4.33. Implementasi Halaman <i>Dashboard</i> Program.....	55
Gambar 4.34. Implementasi Halaman <i>Dashboard</i> Program.....	56
Gambar 4.35. <i>Source Code</i> Menampilkan <i>Form</i> Data Program.....	57

Gambar 4.36. Implementasi Halaman <i>Dashboard</i> Program Hapus.....	58
Gambar 4.45. Implementasi Halaman <i>Dashboard</i> Donasi	60
Gambar 4.46. Implementasi Halaman <i>Dashboard</i> Donasi Edit.....	61
Gambar 4.48. Implementasi Halaman <i>Dashboard</i> Donasi Hapus	63
Gambar 49. Implementasi Halaman <i>Dashboard</i> Laporan Keuangan	58
Gambar 4.50. <i>Source Code</i> Halaman <i>Dashboard</i> Laporan Keuangan	59
Gambar 4.51. Implementasi Halaman <i>Dashboard</i> Laporan Keuangan	63
Gambar 4.52. <i>Source Code</i> Menampilkan <i>Form</i> Profil	64
Gambar 4.53. Implementasi Halaman <i>Home</i>	65
Gambar 4.54. Implementasi Halaman Profil.....	66
Gambar 4.55. Implementasi Halaman Guru Dan Santri	67
Gambar 4.56. Implementasi Halaman Program	68
Gambar 4.57. Implementasi Halaman Registrasi.....	69
Gambar 4.58. Implementasi Halaman Kontak Kami	70

DAFTAR TABEL

Tabel 3.1. Deskripsi Aktor	15
Tabel 3.2. Deskripsi <i>Use Case</i>	15
Tabel 3.3. Skenario <i>Use Case</i> “Login”	17
Tabel 3.5. Skenario <i>Use Case</i> “Mengelola data galeri”	20
Tabel 3.9. Skenario <i>Use Case</i> “Mengolah Data Guru dan Santri”	18
Tabel 3.10. Skenario <i>Use Case</i> “Mengolah Tambah Data Guru”	18
Tabel 3.11. Skenario <i>Use Case</i> “Edit Data Guru”	19
Tabel 3.12. Skenario <i>Use Case</i> “Menghapus Data Guru dan Santri ”	20
Tabel 3.13. Skenario <i>Use Case</i> “Tambah Data Program”	21
Tabel 3.14. Skenario <i>Use Case</i> “Edit data program”	21
Tabel 3.15. Skenario <i>Use Case</i> “Hapus data program”	22
Tabel 3.16. Skenario <i>Use Case</i> “Mengelola donasi”	23
Tabel 3.17. Skenario “Skenario Melihat Laporan Penggunaan Dana ”	24
Tabel 3.18. Skenario Melihat Tampilan <i>Home</i>	26
Tabel 3.19. Skenario Melihat Tampilan Profil.....	26
Tabel 3.20 Skenario Melihat Tampilan Program.....	27
Tabel 3.21. Skenario Melihat Tampilan Donasi	28
Tabel 3.22. Skenario Melihat Tampilan Program.....	28
Tabel 3.23. Skenario Melihat Tampilan Kontak Kami	29
Tabel 3.24. Tabel <i>Users</i>	31
Tabel 3.25. Tabel Visi Misi.....	31
Tabel 3.27. Tabel Struktur Lembaga.....	32
Tabel 3.28. Tabel Program.....	33
Tabel 3.29. Tabel Program.....	33
Tabel 4.1. Pengujian <i>Users</i>	71
Tabel 4.2. Pengujian <i>Form Admin</i>	72

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan ilmu dan teknologi yang semakin pesat khususnya teknologi informasi dan telekomunikasi yang memberikan peranan penting dalam kehidupan manusia. Salah satunya adalah teknologi website yang kini telah banyak digunakan dalam membantu mempermudah penyampaian sebuah informasi. Mulai dari perusahaan-perusahaan, sekolah-sekolah, bisnis, perguruan tinggi, lembaga atau organisasi dan sebagainya yang telah banyak memanfaatkan teknologi *website* dalam kegiatan penjualan, promosi, pembelajaran yang memberikan banyak manfaat dan bisa dikatakan telah menjadi suatu kebutuhan penting dalam mempermudah penyampaian suatu informasi dengan cara efektif yang dapat diakses kapan saja, darimana saja, dan oleh siapa saja, melalui media internet. Pentingnya *website* untuk mengimbangi perkembangan zaman yang semakin pesat dan sangat berdampak, khususnya pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak membutuhkan sebuah *website* sebagai penyedia informasi.

Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak merupakan suatu lembaga yang bersifat keagamaan yang berfokus pada hafalan Al Quran sampai saat ini masih dalam tahap pembangunan dan perkembangan dalam beroperasi. Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak ini berada di dalam Komplek Batara Indah 1 di Jalan Dr Wahidin Sudirohusodo. Pondok Pesantren Tahfidz Quran El ‘Amani telah beroperasi sejak tahun 2012. Pertama kali beroperasi, jumlah santri waktu itu masih sangat sedikit. Pada operasionalnya, Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dalam mencari calon santri untuk menghafal di Pondok tersebut dengan keterbatasan media dalam penyampaian informasi ke masyarakat luas Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak hanya mengandalkan anak-anak dari kerabat dan media mulut ke mulut. Sehingga masyarakat di luar daerah sulit mencari informasi tentang Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Maka dalam tahap perkembangannya, untuk mengatasi permasalahan tersebut maka kebutuhan akan suatu *website* pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak

sangatlah penting dan tepat untuk menjalani operasional pemasaran pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak sebagai bentuk mengikuti perkembangan zaman agar bisa menjangkau publik yang ingin menghafal Al Qur’an di Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak.

Selain itu dalam tahap pembangunan Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak menerima sumbangan dari pihak publik seperti masyarakat umum, lembaga sosial, perusahaan, atau instansi pemerintahan yang seringkali disebut sebagai donatur untuk operasional dalam pembangunan infrastruktur. Selama ini, Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak dalam hal pengelolaan donasinya, tidak mencatat donasi masuk yang diberikan oleh donatur dan penggunaan dana donasi yang keluar serta tidak adanya bukti laporan dalam penggunaan dana donasi, hal tersebut mengakibatkan tidak jelasnya pihak Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak dalam mempergunakan/mengelola donasi yang masuk, kurangnya pertanggungjawaban kepada pihak donatur, tidak mengetahui kontribusi dana donasi yang masuk dan yang dikeluarkan, dan tidak dapat mengetahui aset dimiliki sehingga hal tersebut akan berpengaruh pada kinerja yang dilakukan pihak Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak.

Dalam pencatatan donasi untuk memberikan kemudahan dan meningkatkan transparansi dan akuntabilitas dana donasi dan demi membangun kepercayaan dari publik terutama dari pihak publik yang berkeinginan untuk menjadi salah satu donatur pada Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak, maka adanya sebuah pencatatan pengelolaan donasi yang masuk dan keluar yang dihasilkan secara benar sangat diperlukan. Dengan menggunakan media komputer yang sangat berperan banyak dalam hal mempermudah kegiatan yang dilakukan seperti mencatat, membuat dan menghasilkan suatu informasi dalam waktu singkat sehingga dapat menghemat waktu.

Berdasarkan latar belakang diatas, dan dalam rangka untuk membantu perkembangan dan proses kinerja maka membuat *website* pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dianggap perlu maka diusulkan tugas akhir dengan judul “Rancang Bangun *Website* pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak”.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, rumusan masalah pada penelitian ini adalah bagaimana merancang sebuah *website* pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak?

1.3. Batasan Masalah

Dalam penyusunan tugas akhir ini, terdapat batas ruang lingkup masalah sebagai berikut :

- 1) *Website* yang akan dibangun akan diterapkan pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak.
- 2) *Website* yang akan dibangun nantinya menggunakan Laravel versi 8 sebagai *framework* bahasa pemrograman PHP dan *Bootstrap* sebagai *framework* CSS.
- 3) Pengguna *website* ini terdiri dari admin dan donatur/user.
- 4) Menu item terdiri dari home, profil, data guru dan santri, program, donasi dan kontak kami.
- 5) Pada menu donasi, admin dapat melihat banyaknya donasi yang masuk dan mencatat donasi yang digunakan/keluar sebagai laporan penggunaan dana serta melihat saldo, sedangkan donatur/user dapat melakukan donasi dengan memasukkan data yang diminta dan mengirimkan bukti transfer.

1.4. Tujuan Penelitian

Dengan dilakukannya penelitian untuk keperluan tugas akhir diharapkan dapat menghasilkan *website* pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dapat memudahkan penyampaian informasi dengan jangkauan yang lebih luas mengenai Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak kepada masyarakat tidak hanya di Pontianak saja tetapi hingga ke luar provinsi dan menghasilkan sistem yang dapat membantu admin dalam melakukan proses pengelolaan banyaknya donasi yang masuk dan mencatat donasi yang digunakan/keluar sebagai laporan penggunaan dana yang mendukung proses kinerja pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak.

1.5. Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian Tugas Akhir ini adalah:

1) Bagi Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak

Meningkatkan kredibilitas Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak, memaksimalkan jangkauan penyebaran informasi mengenai Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak, memudahkan pengguna umum/donatur mengenali profil dari Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak, memudahkan mempublikasikan berbagai program/kegiatan yang dilakukan, memudahkan khalayak umum untuk mengakses *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak, dan mempermudah admin dalam mengelola donasi.

2) Bagi *Users* (Donatur/*user*)

Mempermudah dalam mendapatkan informasi mengenai Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dan mengikuti perkembangan kegiatan pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dan melakukan donasi.

1.6. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penyusunan proposal pengajuan Tugas Akhir ini adalah sebagai berikut :

1.6.1. Metode Pengumpulan Data

Teknik mengumpulkan data yang dilakukan dalam penyusunan pengajuan Tugas Akhir ini adalah sebagai berikut :

1.6.1.1. Metode Wawancara

Metode ini dilakukan dengan cara mengumpulkan data secara langsung dengan cara tanya jawab kepada pihak-pihak terkait yang ada di Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak.

1.6.1.2. Metode Studi Literatur

Metode ini merupakan suatu teknik pengumpulan data-data dari penelitian terlebih dahulu, pembelajaran dari berbagai karya tulis dan dokumen seperti buku, jurnal maupun informasi di berbagai situs yang terdapat di internet dengan melakukan kegiatan yang disebut *browsing* yang berkaitan dengan *website* Pondok

Pesantren Tahfidz Al Quran El ‘Amani Pontianak agar dapat dijadikan landasan dalam perancangan membangun *website* informasi Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak.

1.6.1.3. Metode Observasi

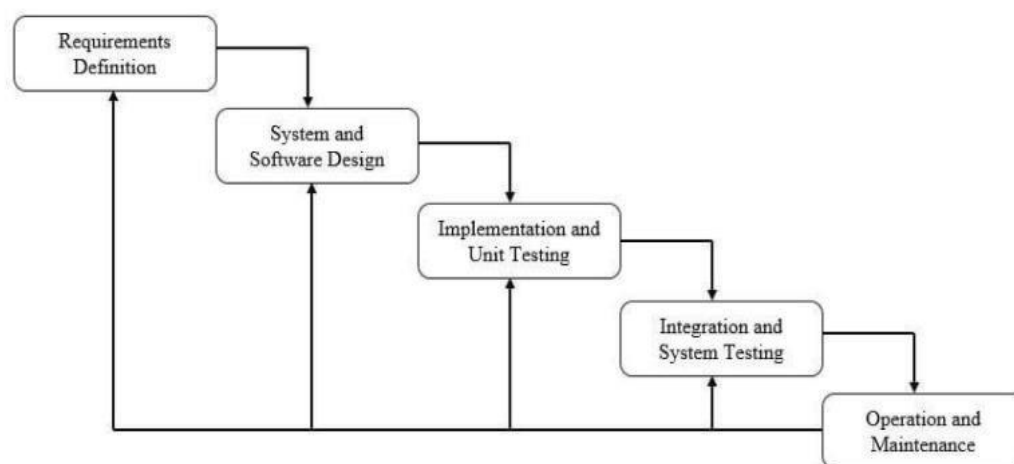
Metode observasi merupakan metode pengumpulan data yang dilakukan dengan cara melakukan pengamatan secara langsung di Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak sebagai objek penelitian. Pengamatan ini dilakukan untuk mengamati atau membaca suatu peristiwa atau sekumpulan fakta pada objek penelitian.

1.6.2. Metode Pengembangan Sistem

1.6.2.1. Metode Waterfall

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu di sebut *waterfall* (Air Terjun).

Ian Sommerville menjelaskan bahwa ada lima tahapan pada Metode Waterfall, yakni *Requirements Analysis and Definition*, *Sytem and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operationa and Maintenance*.



Gambar 1.1. Metode *Waterfall*

Penjelasan Metode Tahapan-Tahapan Metode Waterfall sebagai berikut :

1) *Requirement Analysis* (Definisi Kebutuhan)

Tahapan ini pengembang mencari informasi kebutuhan pengguna terhadap sebuah *website* sistem informasi pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Pengumpulan data informasi ini diperoleh dengan cara observasi, studi literatur dan wawancara. Informasi yang diperoleh berupa data atau informasi mengenai spesifikasi kebutuhan pada *website* yang akan dikembangkan.

2) *System and Software Design* (Desain Sistem dan Perangkat Lunak)

Informasi mengenai kebutuhan dari tahap *Requirement Analysis*, selanjutnya pada tahap ini diimplementasikan pada desain pengembangan. Perancangan sistem dengan menggunakan perangkat pemodelan sistem seperti *use case* dan rancangan stuktur menu pengunjung. Tujuannya untuk mengetahui gambaran sistem yang akan dibangun pada *website*.

3) *Implementation and Unit Testing* (Penulisan Kode Program dan Testing Unit)

Tahap *implementation and unit testing* merupakan tahap pemrograman dengan menggunakan bahasa pemrograman seperti HTML, PHP, JavaScript dan MySQL. Pada tahapan ini juga dilakukan pengujian dan pemeriksaan terhadap sistem yang telah dibuat untuk menemukan kesalahan sehingga dapat diperbaiki.

4) *Integration and System Testing* (Pengujian Sistem)

Setelah dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem oleh pihak Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak secara keseluruhan untuk mengidentifikasi kemungkinan menemukan adanya kekurangan, kesalahan dan kegagalan sistem.

5) *Operation and Maintenance* (Penerapan Program dan Pemeliharaan)

Pada tahap terakhir dalam Metode Waterfall, *website* yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan, tetapi tahap pemeliharaan sistem tidak digunakan karena hanya sampai pada tahap pengujian saja [1].

Penulis memilih metode *waterfall* karena metode ini memiliki alur yang jelas, dilakukan secara berurutan dan sangat sistematis dan terperinci sehingga sistem pengerjaan akan terjadwal dengan baik dan mudah dikontrol. Setiap proses tidak dapat dikerjakan bersamaan sehingga harus menyelesaikan tahapan satu kemudian melanjutkan ke tahapan berikutnya sehingga dapat meminimalisir kesalahan yang akan dilakukan.

1.7. Sistem Penulisan

Adapun sistem penulisan Tugas Akhir ini secara garis besarnya terbagi menjadi 5 (lima) bab, sebagai berikut :

BAB 1 : PENDAHULUAN

Pada bab 1 berisi penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dilakukan, manfaat dari penelitian, metodologi penelitian yang dilakukan pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dan sistematika penulisan Tugas Akhir.

BAB 2 : DASAR TEORI

Pada bab ini akan diuraikan mengenai beberapa hal yang berhubungan dan terkait dengan referensi-referensi, kajian terlebih dahulu dan dasar teori yang berhubungan dengan permasalahan yang dibahas serta perangkat yang akan digunakan pada saat pembuatan tugas akhir ini.

BAB 3 : RANCANGAN SISTEM

Pada bab ini membahas mengenai gambaran umum *website* pada Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak, analisis kebutuhan, *Use Case*, perancangan basis data dan perancangan antarmuka.

BAB 4 : HASIL DAN PEMBAHASAN

Pada bab 4 membahas mengenai hasil dari sistem yang telah dibuat yaitu *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak beserta tampilan antarmuka dan pengujian sistem.

BAB 5 : PENUTUP

Pada bab ini akan diuraikan tentang kesimpulan dari pembahasan permasalahan serta saran-saran untuk membangun *website* yang lebih baik untuk kedepannya.

BAB II

DASAR TEORI

2.1. Tinjauan Pustaka

Dari beberapa referensi jurnal yang telah dikumpulkan oleh penulis, dapat dijadikan perbandingan dalam penelitian yang pertama oleh Ibnu Dwi Lesmono, Fahlepi Roma Doni, dan Sumarna berjudul “Rancang Bangun Sistem Informasi Pendaftaran Santri Baru Berbasis Web pada Pondok Pesantren Al-Fatah, aplikasi ini merupakan sistem informasi berbasis web yang dirancang khusus untuk menampilkan berbagai macam informasi seputar pendaftaran online pada Pondok Pesantren Al-Falah. Penelitian ini memiliki persamaan dengan sistem yang akan dibuat oleh penulis yaitu merancang sistem informasi berbasis website yang adanya fitur untuk melakukan pendaftaran secara *online*, metode yang digunakan untuk mengumpulkan data adalah observasi, studi literatur dan wawancara. Hak akses diberikan untuk admin. Sistem informasi pendaftaran santri baru berbasis web ini merupakan salah satu alternatif untuk memberikan kemudahan bagi calon santri yang ingin mendaftar serta membantu pihak pesantren dalam pengolahan data informasi calon santri dengan mudah [2].

Pada penelitian kedua oleh Weni Kurniasih berjudul Rancang Bangun Website Pondok Pesantren Darul Quran memiliki persamaan dan perbedaan dengan sistem yang akan dibuat oleh penulis. Persamaannya adalah sama-sama merupakan sistem informasi berbasis website yang membahas tentang pendaftaran santri baru dan perbedaannya pada fitur untuk memberikan donasi dengan adanya bukti transfer, notifikasi batas pembayaran, notifikasi pada email tentang pembatalan donasi jika tidak melakukan tranfer dalam waktu 1 x 24 jam [3].

Ketiga, penelitian yang dilakukan oleh Sistem Informasi Akademik Pondok Pesantren Mamba’Ul Hisan Isyhar Berbasis Web. Penelitian ini merancang sebuah sistem akademik dalam pencatatan informasi terkait pembayaran dan penilaian santri. Tujuannya dapat menghasilkan rancangan sistem portal pesantren di Indonesia berbasis web, yang dapat membantu sekaligus memudahkan siswa dan masyarakat umum dalam menemukan kriteria pesantren yang diinginkan [4].

Adapun dari referensi yang telah didapatkan, ada beberapa fitur yang dapat digunakan pada *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak.

Menu-menu yang ada pada *website* Pondok Pesantren Tahfidz Al Quran adalah Home, Profil, Struktur Lembaga, Data Guru dan Santri, Program, Kontak Kami dan Donasi. Pada menu donasi, sistem yang akan dibuat yaitu adanya pencatatan dana donasi masuk oleh donatur dan yang dikeluarkan oleh admin serta adanya pencatatan adanya laporan penggunaan dana donasi.

2.2. Dasar Teori

2.2.1. Website

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman [5].

2.2.2. HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman web browser. Tag-tag HTML selalu diawali dengan <x> dan diakhiri dengan </x> dimana x tag HTML itu seperti b,i,u, dll [6].

2.2.3. XAMPP

XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (*localhost*) yang terdiri beberapa program antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl [7].

2.2.4. MySQL

MySQL adalah salah satu jenis *database* server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang *database* sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* sehingga

mudah untuk digunakan. MySQL juga bersifat *open source* dan *free* pada berbagai *platform* kecuali pada *windows* yang bersifat *shareware*. MySQL didistribusikan dengan lisensi *open source* GPL (*General Public License*) mulai versi 3.23, pada bulan Juni 2020 [8].

2.2.5. PHP (*Hypertext Preprocessor*)

Hypertext Preprocessor (PHP) adalah suatu bahasa pemrograman yang digunakan untuk membuat web dinamis, walau bisa juga digunakan untuk membuat program lain. Tentunya bahasa pemrograman PHP berbeda dengan HTML, pada PHP Script/kode yang dibuat tidak dapat ditampilkan pada halaman/muka website begitu saja, tapi harus diproses terlebih dahulu oleh web server lalu ditampilkan dalam bentuk halaman *website* di *web browser*, Script PHP juga dapat disisipkan pada HTML dan Script PHP selalu diawali dengan `<php` dan diakhiri dengan `>`.

Manajemen database yang biasanya digunakan untuk pemrograman PHP misalnya seperti MySQL, tapi ada juga yang menggunakan Oracle, Microsoft Acces dan lain lain. PHP disebut juga sebagai bahasa pemrograman *script server side*, karena PHP di proses pada komputer server [9].

2.2.6. Bootstrap

Bootstrap adalah sebuah *library framework* CSS yang dibuat khusus untuk bagian pengembangan *front-end website*. *Bootstrap* juga merupakan salah satu *framework* HTML, CSS dan Javascript yang paling populer di kalangan *web developer* yang digunakan untuk mengembangkan sebuah *website* yang *responsive*. Sehingga halaman *website* nantinya dapat menyesuaikan sesuai dengan ukuran monitor *device* (*dekstop*, *table*, *ponsel*) yang digunakan pengguna di saat mengakses *website* dari *browser* [10].

2.2.7. Java Script

Java script merupakan salah satu bahasa pemrograman tingkat tinggi dan dinamis dalam membuat *website*. *Java script* populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Google Chrome, Internet Explorer

(IE), *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *java script* dapat disisipkan dalam halaman web menggunakan tag *SCRIPT* [11].

2.2.8. *Framework Laravel*

Laravel adalah *framework* Bahasa pemrograman *Hypertext Preprocessor* (PHP) yang ditujukan untuk memudahkan para *developer* dalam melakukan pengembangan aplikasi berbasis *web* dengan menerapkan konsep MVC (*Model, View, Controller*). MVC (*Model View Controller*) adalah sebuah pendekatan *software* yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi seperti manipulasi data, *controller* dan *user interface*. Ada 5 konsep arsitektur pada *framework* Laravel yang mempunyai masing-masing fungsi di antaranya :

1) *Route*

Berfungsi sebagai pemberi akses pada setiap *request* sesuai alur yang telah ditentukan. Di dalam routes memiliki 4 instruksi standar, yakni:

- a) *Get* berfungsi untuk memanggil request.
- b) *Put* berfungsi untuk mengubah data sesuai request
- c) *Post* berfungsi untuk menambahkan data sesuai *request*
- d) *Delete* berfungsi untuk menghapus data sesuai *request*

2) *Controller*

Controller adalah bagian yang menjadi penghubung antara *model* dan *view*. *Controller* memiliki perintah-perintah yang berfungsi untuk memproses bagaimana data ditampilkan dari *Model* ke *View* atau sebaliknya. Struktur *controller* pada penulisan kode program di Laravel meliputi :

- a) *Index* berfungsi untuk menampilkan data keseluruhan
- b) *Create* berfungsi untuk memanggil *form* yang berisikan kolom masukan.
- c) *Store* berfungsi untuk menyimpan data ke dalam tabel.
- d) *Show* berfungsi untuk menampilkan data sesuai Id.
- e) *Edit* berfungsi memanggil data sesuai Id yang berisikan *form* masukan untuk proses *update*.
- f) *Update* berfungsi untuk memperbaharui data pada tabel.
- g) *Destroy* Berfungsi untuk menghapus data sesuai Id.

3) *Model*

Merupakan sekumpulan data yang memiliki fungsi-fungsi untuk mengelola suatu tabel pada sebuah *database*. Struktur pemodelan data pada Laravel yakni memiliki properti yang terdiri dari *table*, *primary key* dan *fillable*. Ketiga fungsi tersebut harus di *protected*. Pada bagian tabel harus diisi dengan nama tabel yang sesuai pada *database*, di bagian *primary key* harus diisi sesuai *primary key* pada tabel tersebut dan pada bagian *fillable* diisi dengan bagian-bagian yang mencakup dalam tabel tersebut.

4) *View*

Merupakan file yang berisi kode HTML (*Hypertext Markup Language*) yang berfungsi untuk menampilkan suatu data ke dalam *browser*. Format *view* pada Laravel harus menggunakan istilah *blade*, contohnya seperti: *view.blade.php*.

5) *Migrations*

Merupakan proses perancangan suatu tabel, dalam hal ini *migrations* berfungsi sebagai *blueprint database* atau dapat diistilahkan sebagai penyedia sistem kontrol untuk skema *database*.

BAB III

PERANCANGAN SISTEM

3.1. Gambaran Umum

Website Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak adalah sebuah aplikasi berbasis *website* yang dibuat untuk mempermudah pengurus dalam mengolah dan melaporkan donasi yang terkumpul sehingga donasi yang masuk dan keluar jelas dan terstruktur serta memudahkan masyarakat dalam mencari informasi mengenai Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Pada *website* ini juga terdapat fitur donasi sehingga mudah bagi para pengunjung untuk melakukan donasi.

3.2. Analisis Kebutuhan

Analisa kebutuhan merupakan tahapan yang bertujuan untuk melakukan proses pengumpulan data dan informasi yang digunakan untuk mendukung pembuatan sistem, menganalisa kebutuhan yang diperlukan untuk membuat *website*. Analisa kebutuhan dapat berupa perangkat keras dan perangkat lunak yang digunakan serta bertujuan untuk mendukung semua keperluan yang akan digunakan pada *website* yang akan dibuat sebagai berikut.

3.2.1. Analisis Kebutuhan Data

Data yang dapat dikelola *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak adalah sebagai berikut:

- a. Data galeri, data visi-misi, sejarah dan struktur lembaga
- b. Data guru dan santri, data program/kegiatan

3.2.2. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam membangun *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak yang penulis gunakan yaitu terdiri dari :

1. Laptop Asus
2. *Processor Intel Core i3*
3. RAM 4 GB
4. SSD 256 GB

3.2.3. Kebutuhan Perangkat Lunak

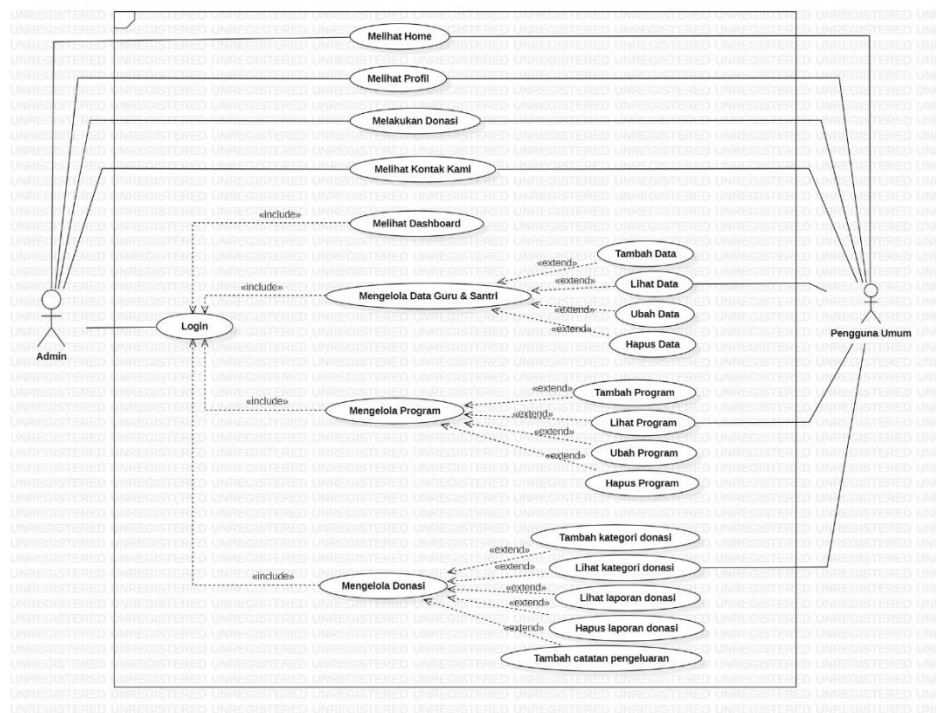
Kebutuhan perangkat lunak dalam membangun *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak yang penulis gunakan yaitu terdiri dari :

- Visual Studio Code*,
- Microsoft edge*
- Sistem Operasi *Microsoft Windows 10 64bit*,
- XAMPP*
- Framework Laravel versi 8*
- phpMyAdmin*
- MySQL server*

3.3. Perancangan Sistem

3.3.1. Use Case

Use case diagram adalah alat dalam pemodelan perangkat lunak yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem yang dirancang. Hasil representasi dari skema ini dibuat secara sederhana dan bertujuan untuk memudahkan *user* dalam membaca informasi yang diberikan. Berikut gambaran *use case* untuk sebuah sistem *website* Pondok Pesantren El Amani Pontianak pada gambar 3.1.



Gambar 3.1. Use Case Diagram

Gambaran fungsionalitas dari sebuah sistem yang menjalankan tentang pekerjaan yang dilakukan aktor. Untuk masuk kedalam sistem admin perlu memasukkan email dan password. Sedangkan user/pengguna umum/donatur bisa melihat informasi pada *website* Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak dan melakukan donasi tanpa harus melakukan login terlebih dahulu.

a. Deskripsi Aktor

Berdasarkan Gambar 3.1. *Use Case diagram* memiliki dua aktor yaitu admin dan pengguna umum/*user*. Admin memiliki akses *full controll* pada sistem ini, sedangkan pengguna umum/*user* hanya dapat mengakses tampilan depan dan melakukan konfirmasi transfer dengan mengisi data yang diminta.

Tabel 3.1. Deskripsi Aktor

No.	Nama Aktor	Deskripsi
1.	Admin	Admin memiliki akses <i>full controll</i> , mengelola tampilan home (galeri), visi-misi, sejarah, struktur lembaga, program, data guru dan santri, donasi dan laporan penggunaan dana donasi.
2.	Donatur/user	Donatur/User dapat mengunjungi <i>website</i> untuk melihat/mengakses tampilan <i>home</i> (galeri), visi-misi, sejarah, struktur lembaga, program, data guru dan santri, kontak kami serta melakukan donasi.

b. Deskripsi *Use Case*

Adapun deskripsi *Use Case* menggambarkan deskripsi dan fungsi dari setiap *use case*. Berikut ini deskripsi dapat dilihat pada Tabel 3.2.

Tabel 3.2. Deskripsi *Use Case*

No.	Nama <i>Use Case</i>	Deskripsi
1.	<i>Login</i>	Fungsi melakukan <i>login</i> pada saat admin masuk ke <i>website</i> bagian <i>dashboard</i> .

2.	Mengelola Data Guru dan Santri	Fungsi bagi admin untuk mengelola (menambah/edit/hapus) data guru dan santri pondok pesantren tahfidz al quran el 'amani pontianak
3.	Mengelola data program	Fungsi bagi admin untuk mengelola (menambah/edit/hapus) data program pondok pesantren tahfidz quran el 'amani pontianak
4.	Mengelola data donasi	Fungsi bagi admin untuk menambah kategori donasi, melihat donasi yang masuk, menambah catatan donasi yang keluar dan melihat saldo.
5.	Melihat Tampilan Home	Pengguna umum dan admin dapat melihat tampilan home dimana terdapat informasi dan tampilan gambar
6.	Melihat Tampilan Profil	Dapat melihat tampilan profil dimana terdapat informasi mengenai visi misi, sejarah dan struktur lembaga
9.	Melakukan donasi	Fungsi bagi pengguna umum/donatur untuk melakukan donasi (dengan mengisi data dan bukti transfer tanpa harus melakukan login terlebih dahulu)
10.	Melihat data guru dan santri	Pengguna umum/donatur dan admin dapat melihat informasi umum mengenai guru dan santri pondok pesantren tahfidz quran el 'amani pontianak
11.	Melihat program	Pengguna umum/donatur dan admin dapat melihat informasi program program yang

		ada dipondok pesantren tahfidz al quran el ‘amani pontianak
12.	Melihat kontak kami	Pengguna umum/donatur dan admin dapat melihat kontak dan sosial media untuk menggali informasi atau komunikasi admin secara personal dan memberikan kritik saran maupun doa di kolom tersedia

3.3.2. Skenario Use Case

Skenario *Use Case* bertujuan untuk mendeskripsikan diagram *Use Case*. *Use Case* dibuat dalam bentuk tabel yang berguna untuk memahami alur setiap proses *Use Case* yang digunakan. Adapun skenario *Use Case* pada *website* ini dijelaskan pada tabel dibawah ini :

Tabel 3.3. Skenario *Use Case* “Login”

Nomor Skenario	SK-01
Nama <i>Use Case</i>	<i>Login</i>
Ringkasan	Admin login ke <i>Website</i> Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak
Aktor	Admin
Kondisi Awal	Layar <i>login</i> ditampilkan dan <i>textfield email</i> dan <i>password</i> kosong
Deskripsi	<ol style="list-style-type: none"> 1. Admin memasukkan email pada <i>textfield email</i> 2. Admin dan donatur memasukkan password pada <i>textfield password</i> 3. Admin dan donatur menekan tombol login <p>Sistem akan mengecek <i>email</i> dan <i>password</i></p>
Alternatif	-
Kondisi Akhir	Layar login ditutup dan muncul layar menu <i>dashboard</i> sesuai dengan hak akses admin sebagai pengguna

Berikut ini merupakan *use case* mengelola data guru dan santri yang merupakan alur kerja admin yang dapat mengelola data guru dan santri.

Tabel 3.4. Skenario *Use Case* “Mengolah Data Guru dan Santri”

Nomor Skenario	SK-02
Nama <i>Use Case</i>	Mengelola data guru dan santri
Ringkasan	Admin dapat melihat, menambah, mengedit dan menghapus data guru dan santri
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data guru dan santri
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu yang diinginkan (tambah/edit/hapus) data guru dan santri 2. Sistem akan memproses perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Data guru dan santri berubah sesuai dengan data yang diinput

Berikut ini merupakan *use case* tambah data guru dan santri yang merupakan alur kerja admin yang dapat menambah data guru.

Tabel 3.5. Skenario *Use Case* “Mengolah Tambah Data Guru dan Santri”

Nomor Skenario	SK-03
Nama <i>Use Case</i>	Tambah data guru dan santri
Ringkasan	Admin dapat menambah data guru dan santri
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data guru dan santri

Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu tambah data guru dan santri 2. Admin mengisi data guru dan santri 3. Admin menekan tombol simpan 4. Sistem akan mengeluarkan pesan bahwa ada penambahan
Alternatif	-
Kondisi Akhir	Data guru dan santri berhasil ditambahkan

Berikut ini merupakan *use case* edit data guru yang merupakan alur kerja admin yang dapat mengedit data guru.

Tabel 3.6. Skenario *Use Case* “Edit Data Guru”

Nomor Skenario	SK-04
Nama <i>Use Case</i>	Edit data guru dan santri
Ringkasan	Admin dapat mengedit data guru dan santri
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data guru dan santri
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih data guru dan santri yang akan diubah dengan memilih edit 2. Admin melakukan perubahan 3. Admin menekan tombol simpan 4. Sistem akan mengeluarkan pesan bahwa pengeditan data berhasil
Alternatif	-
Kondisi Akhir	Data guru dan santri berhasil diedit

Berikut ini merupakan *use case* hapus data guru yang merupakan alur kerja admin yang dapat menghapus data guru dan santri.

Tabel 3.7. Skenario *Use Case* “Menghapus Data Guru dan Santri ”

Nomor Skenario	SK-05
Nama <i>Use Case</i>	Hapus data guru dan santri
Ringkasan	Admin dapat menghapus data guru dan santri
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data guru dan santri
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih data guru dan santri yang akan dihapus 2. Admin menghapus data guru dan santri 3. Admin menekan tombol hapus 4. Sistem akan mengeluarkan pesan bahwa hapus data berhasil
Alternatif	-
Kondisi Akhir	Data guru dan santri berhasil dihapus

Berikut ini merupakan *use case* mengelola data program yang merupakan alur kerja admin yang dapat mengelola data program.

Tabel 3.8. Skenario *Use Case* “Mengelola Data Program”

Nomor Skenario	SK-06
Nama <i>Use Case</i>	Mengelola data program
Ringkasan	Admin dapat menambahkan, mengedit dan menghapus data program
Aktor	Admin
Kondisi Awal	Admin telah masuk ke dalam aplikasi dan aplikasi sedang menampilkan textfield judul dan gambar kosong
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu yang diinginkan (tambah/edit/hapus) data program

	2. Sistem akan memproses perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Data program berubah sesuai dengan yang diinput

Berikut ini merupakan *use case* tambah data program yang merupakan alur kerja admin yang dapat menambah data program.

Tabel 3.9. Skenario *Use Case* “Tambah Data Program”

Nomor Skenario	SK-07
Nama <i>Use Case</i>	Tambah data program
Ringkasan	Admin dapat menambah data program
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data program
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu tambah data program 2. Admin mengisi data program 3. Admin menekan tombol simpan 4. Sistem akan mengeluarkan pesan bahwa penambahan data berhasil
Alternatif	-
Kondisi Akhir	Data program berhasil ditambahkan

Berikut ini merupakan *use case* edit data program yang merupakan alur kerja admin yang dapat mengedit data program.

Tabel 3.10. Skenario *Use Case* “Edit data program”

Nomor Skenario	SK-08
Nama <i>Use Case</i>	Edit data program
Ringkasan	Admin dapat mengedit data program

Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data program
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih data yang akan diubah dengan memilih edit 2. Admin mengedit data program 3. Admin menekan button simpan 4. Sistem akan mengeluarkan bahwa pengeditan data berhasil
Alternatif	-
Kondisi Akhir	Data program berhasil diedit

Berikut ini merupakan *use case* menghapus data program yang merupakan alur kerja admin yang dapat menghapus data program.

Tabel 3.11. Skenario *Use Case* “Hapus data program”

Nomor Skenario	SK-09
Nama <i>Use Case</i>	Menghapus data program
Ringkasan	Admin dapat menghapus data program
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang data program
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih data program yang akan dihapus 2. Admin menghapus data program 3. Admin menekan tombol hapus 4. Sistem akan mengeluarkan pesan bahwa hapus data berhasil
Alternatif	-
Kondisi Akhir	Data program berhasil dihapus

Berikut ini merupakan *use case* mengelola data donasi yang merupakan alur kerja admin yang dapat mengelola data donasi.

Tabel 3.12. Skenario *Use Case* “Mengelola donasi”

Nomor Skenario	SK-10
Nama <i>Use Case</i>	Mengelola dana donasi
Ringkasan	Admin dapat menambah, mengedit, menghapus data
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang dana donatur
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu kelola yang diinginkan (tambah/edit/hapus) data donatur 2. Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Data berubah sesuai dengan pengelolaan yang dilakukan

Berikut ini merupakan *use case* mengelola data donasi dengan menambah kategori donasi yang merupakan alur kerja admin yang dapat mengelola data donasi.

Tabel 3.13. Skenario *Use Case* “Tambah kategori donasi”

Nomor Skenario	SK-11
Nama <i>Use Case</i>	Tambah kategori donasi
Ringkasan	Admin dapat menambah kategori donasi
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang kategori donasi yang ada

Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu tambah kategori donasi 2. Admin mengisi kategori baru 3. Admin menekan tombol simpan 4. Sistem akan mengeluarkan pesan bahwa ada penambahan
Alternatif	-
Kondisi Akhir	Data berubah sesuai dengan pengelolaan yang dilakukan

Berikut ini merupakan *use case* melihat laporan donasi yang merupakan alur kerja admin yang dapat melihat laporan donasi.

Tabel 3.14. Skenario “Skenario Melihat Laporan Donasi”

Nomor Skenario	SK-12
Nama <i>Use Case</i>	Melihat laporan donasi
Ringkasan	Admin dapat melihat laporan donasi
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang laporan donasi
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu melihat data laporan donasi 2. Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Data laporan donasi ditampilkan

Berikut ini merupakan *use case* hapus laporan donasi yang merupakan alur kerja admin yang dapat menghapus laporan donasi.

Tabel 3.15. Skenario “Skenario Menghapus Laporan Donasi”

Nomor Skenario	SK-13
Nama <i>Use Case</i>	Menghapus laporan donasi
Ringkasan	Admin dapat menghapus laporan donasi
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang laporan donasi
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih data program yang akan dihapus 2. Admin menghapus data program 3. Admin menekan tombol hapus 4. Sistem akan mengeluarkan pesan bahwa hapus data berhasil
Alternatif	-
Kondisi Akhir	Data laporan donasi ditampilkan

Berikut ini merupakan *use case* mengelola data donasi dengan menambah catatan pengeluaran yang merupakan alur kerja admin yang dapat mengelola data donasi.

Tabel 3.4. Skenario *Use Case* “Tambah catatan pengeluaran”

Nomor Skenario	SK-14
Nama <i>Use Case</i>	Tambah catatan pengeluaran
Ringkasan	Admin dapat menambah catatan pengeluaran
Aktor	Admin
Kondisi Awal	Admin telah masuk ke aplikasi dan aplikasi sedang menampilkan informasi tentang catatan pengeluaran
Deskripsi	<ol style="list-style-type: none"> 1. Admin memilih menu tambah catatan pengeluaran 2. Admin mengisi catatan pengeluaran baru

	3. Admin menekan tombol simpan 4. Sistem akan mengeluarkan pesan bahwa ada penambahan
Alternatif	-
Kondisi Akhir	Data berubah sesuai dengan pengelolaan yang dilakukan

Berikut ini merupakan *use case* melihat tampilan home yang merupakan alur kerja user yang dapat melihat tampilan halaman *home*.

Tabel 3.17. “Skenario Melihat Tampilan *Home*”

Nomor Skenario	SK-15
Nama <i>Use Case</i>	Melihat tampilan <i>home</i>
Ringkasan	<i>User</i> dapat melihat tampilan <i>home</i>
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> telah masuk ke <i>website</i> dan dapat mengakses tampilan <i>home</i> tanpa harus melakukan registrasi dan <i>login</i>
Deskripsi	Admin memilih menu melihat tampilan <i>home</i> . Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Tampilan halaman <i>home</i> ditampilkan

Berikut ini merupakan *use case* melihat tampilan profil yang merupakan alur kerja *user* yang dapat melihat tampilan halaman profil.

Tabel 3.18. “Skenario Melihat Tampilan Profil”

Nomor Skenario	SK-16
Nama <i>Use Case</i>	Melihat tampilan profil
Ringkasan	<i>User</i> dapat melihat tampilan profil
Aktor	<i>User</i>

Kondisi Awal	<i>User</i> telah masuk ke <i>website</i> dan dapat mengakses tampilan profil tanpa harus melakukan registrasi dan login
Deskripsi	Admin memilih menu melihat tampilan profil. Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Tampilan halaman profil ditampilkan

Berikut ini merupakan *use case* melihat tampilan data guru dan santri yang merupakan alur kerja user yang dapat melihat tampilan halaman data guru dan santri.

Tabel 3.19 “Skenario Melihat Tampilan Data Guru dan Santri”

Nomor Skenario	SK-17
Nama <i>Use Case</i>	Melihat tampilan data guru dan santri
Ringkasan	<i>User</i> dapat melihat tampilan data guru dan santri
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> telah masuk ke <i>website</i> dan dapat mengakses tampilan data guru dan santri tanpa harus melakukan registrasi dan login
Deskripsi	Admin memilih menu melihat tampilan data guru dan santri. Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Tampilan halaman data guru dan santri ditampilkan

Berikut ini merupakan *use case* melihat tampilan program yang merupakan alur kerja user yang dapat melihat tampilan halaman program.

Tabel 3.20. “Skenario Melihat Tampilan Program”

Nomor Skenario	SK-18
Nama <i>Use Case</i>	Melihat tampilan program
Ringkasan	<i>User</i> dapat melihat tampilan program
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> telah masuk ke <i>website</i> dan dapat mengakses tampilan program tanpa harus melakukan registrasi dan login
Deskripsi	Admin memilih menu melihat tampilan data guru dan santri. Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Tampilan halaman program ditampilkan

Berikut ini merupakan *use case* melihat tampilan donasi yang merupakan alur kerja user yang dapat melihat tampilan halaman donasi.

Tabel 3.5. “Skenario Melihat Tampilan Donasi”

Nomor Skenario	SK-19
Nama <i>Use Case</i>	Melihat tampilan donasi berisi <i>form</i> donatur
Ringkasan	<i>User</i> dapat melihat tampilan <i>form</i> donasi
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> telah masuk ke <i>website</i> dan dapat mengakses tampilan donasi tanpa harus melakukan registrasi dan login
Deskripsi	Admin memilih menu melihat tampilan donasi. Sistem akan mengerjakan perintah sesuai menu yang dipilih
Alternatif	-
Kondisi Akhir	Tampilan halaman donasi ditampilkan

Berikut ini merupakan *use case* melihat tampilan kontak kami yang merupakan alur kerja *user* yang dapat melihat tampilan halaman kontak kami.

Tabel 3.22. “Skenario Melihat Tampilan Kontak Kami”

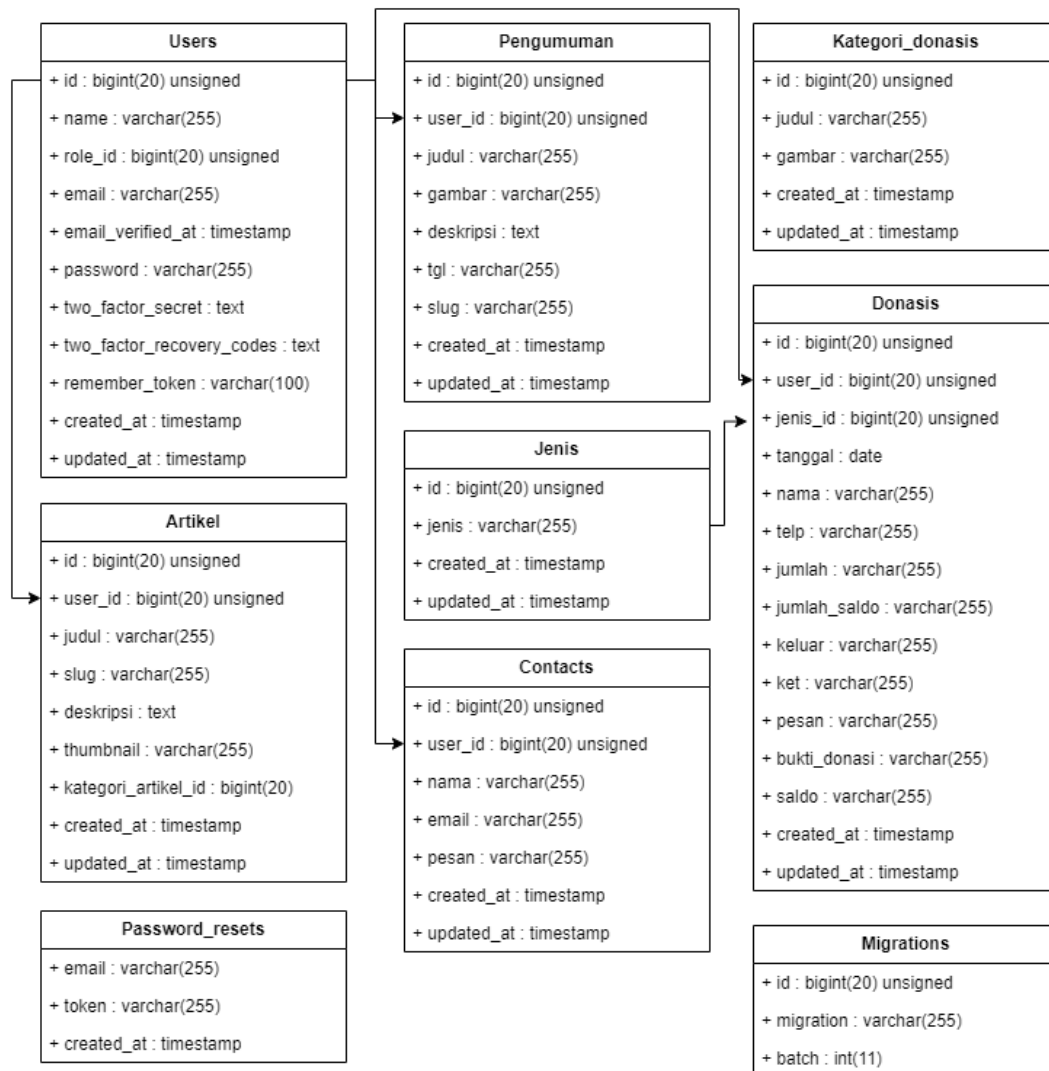
Nomor Skenario	SK-20
Nama <i>Use Case</i>	Melihat tampilan kontak kami
Ringkasan	<i>User</i> dapat melihat tampilan kontak kami
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> telah masuk ke <i>website</i> dan dapat mengakses tampilan donasi tanpa harus melakukan registrasi dan login
Deskripsi	Admin memilih menu melihat tampilan program. Sistem akan mengerjakan perintah sesuai menu yang dipilih.
Alternatif	-
Kondisi Akhir	Tampilan halaman kontak kami ditampilkan

3.4. Perancangan Basis Data

Rancangan data untuk membangun website Pondok Pesantren Tahfidz Al Quran El ‘Amani ini menggunakan data-data yang tersimpan didalam tabel-tabel pada *database*.

3.4.1. Relasi Tabel

Tabel adalah objek dasar yang merupakan jantung dari database relasional. Tujuannya adalah menyimpan informasi. Tabel yang satu dengan yang lainnya bisa dihubungkan. Adapun relasi tabel dari website Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak terlihat pada gambar 3.2.

Gambar 3.2. Rancangan *Database*

Relasi tabel yang terlihat pada gambar 3.2. merupakan hubungan relasi antara satu tabel dengan tabel yang lain yang ada dalam sebuah *database*. Relasi tabel melibatkan kolom *primary key* sebuah tabel dengan sebuah kolom *foreign key* dalam tabel lainnya. Kedua kolom inilah yang digunakan untuk membangun sebuah relasi.

3.4.2. Struktur Tabel

Adapun struktur tabel-tabel yang digunakan pada *database* dari *website* ini adalah sebagai berikut :

a. Tabel *Users*

Tabel *Users* digunakan untuk menyimpan data admin dan donatur. Adapun struktur tabel *Users* dapat dilihat pada tabel 3.23. berikut :

Tabel 3.23. “Tabel *Users*”

Nama Field	Tipe Data	Panjang Data	Key
<i>Id</i>	<i>Bigint</i>	20	<i>Primary Key</i>
<i>Name</i>	<i>Varchar</i>	255	
<i>Role_id</i>	<i>Bigint</i>	20	
<i>Email</i>	<i>Varchar</i>	255	<i>Foreign Key</i>
<i>Email_verified_at</i>	<i>timestamp</i>		
<i>Password</i>	<i>Varchar</i>	255	
<i>Two_factor_secret</i>	<i>Text</i>		
<i>Two_factor_recovery_codes</i>	<i>Text</i>	255	
<i>Remember_token</i>	<i>Varchar</i>	100	
<i>Created_at</i>	<i>Timestamp</i>		
<i>Update_at</i>	<i>Timestamp</i>		

b. Tabel Pengumuman

Tabel Pengumuman digunakan untuk menyimpan data guru dan santri dari Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Adapun struktur tabel Pengumuman dapat dilihat pada tabel 3.24. berikut :

Tabel 3.24. “Tabel Pengumuman”

Nama Field	Tipe Data	Panjang Data	Primary Key
<i>Id</i>	<i>Bigint</i>	20	<i>Primary Key</i>
<i>User_id</i>	<i>Bigint</i>	20	<i>Foreign Key</i>
<i>Judul</i>	<i>Varchar</i>	255	
<i>Gambar</i>	<i>Varchar</i>	255	
<i>Deskripsi</i>	<i>Text</i>		
<i>Tgl</i>	<i>Varchar</i>	255	
<i>Slug</i>	<i>Varchar</i>	255	
<i>Created_at</i>			
<i>Update_at</i>	<i>Timestamp</i>		

c. Tabel Jenis

Tabel Jenis digunakan untuk membedakan apakah data yang masuk/diinput termasuk pemasukan atau pengeluaran. Adapun struktur tabel jenis dapat dilihat pada tabel 3.25. berikut :

Tabel 3.25. “Tabel Jenis”

Nama Field	Tipe Data	Panjang Data	Primary Key
Id	<i>Bigint</i>	20	<i>Primary Key</i>
Jenis	<i>Varchar</i>	225	<i>Foreign Key</i>
<i>Create_at</i>	<i>Timestamp</i>		
<i>Update_at</i>	<i>Timestamp</i>		

d. Tabel Artikel

Tabel artikel digunakan untuk menyimpan data program. Adapun struktur tabel program dapat dilihat pada tabel 3.26. berikut :

Tabel 3.6. “Tabel Artikel”

Nama Field	Tipe Data	Panjang Data	Primary Key
Id	<i>Bigint</i>	20	<i>Primary Key</i>
User_id	<i>Bigint</i>	20	<i>Foreign Key</i>
Judul	<i>Varchar</i>	255	
Slug	<i>Varchar</i>	255	
Deskripsi	<i>Text</i>		
Thumbnail	<i>Varchar</i>	255	
Kategori_artikel_id	<i>Bigint</i>	20	
<i>Create_at</i>	<i>Timestamp</i>		
<i>Update_at</i>	<i>Timestamp</i>		

e. Tabel Contact

Tabel *Contact* digunakan untuk menyimpan data pengunjung yang memberikan kritik dan sarannya untuk Pondok Pesantren Tahfidz Al Quran El ‘Amani. Adapun struktur tabel *contact* dapat dilihat pada tabel 3.27.

Tabel 3.7. “Tabel *Contact*”

Nama Field	Tipe Data	Panjang Data	Primary Key
<i>Id</i>	<i>Bigint</i>	20	<i>Primary Key</i>
<i>User_id</i>	<i>Varchar</i>	20	<i>Foreign Key</i>
<i>Nama</i>	<i>Varchar</i>	255	
<i>Email</i>	<i>Varchar</i>	255	
<i>Pesan</i>	<i>Varchar</i>		
<i>Create_at</i>	<i>Timestamp</i>		
<i>Update_at</i>	<i>Timestamp</i>		

f. Tabel Donasis

Tabel Donasis digunakan untuk menyimpan data keuangan termasuk pemasukan dan pengeluaran dan jumlah saldo pada Pondok Pesantren Tahfidz Al Quran El ‘Amani. Adapun struktur tabel donasis dapat dilihat pada tabel 3.19

Tabel 3. 8. Tabel Donasis

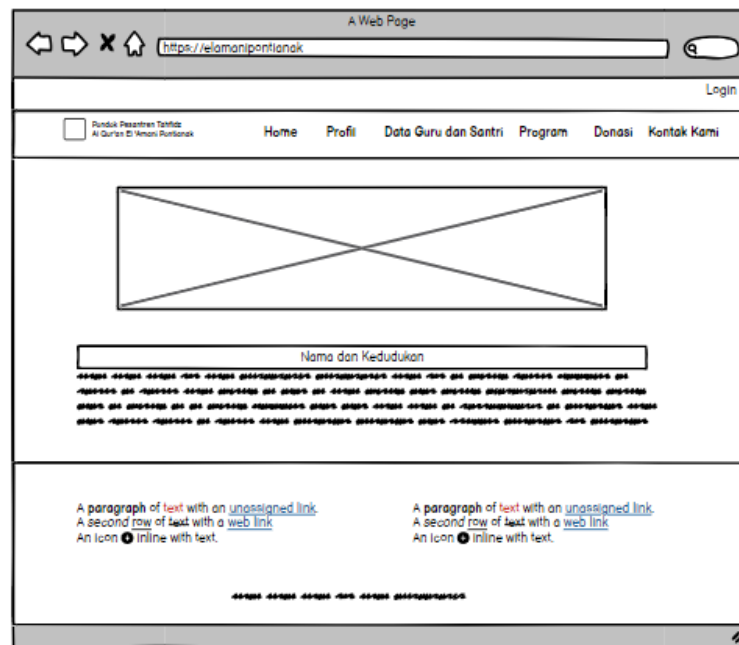
Nama Field	Tipe Data	Panjang Data	Primary Key
<i>Id</i>	<i>Bigint</i>	20	Primary Key
<i>User_id</i>	<i>Bigint</i>	20	Foreign Key
<i>Jenis_id</i>	<i>Bigint</i>	20	
<i>Tanggal</i>	<i>Date</i>		
<i>Nama</i>	<i>Varchar</i>	255	
<i>Telp</i>	<i>Varchar</i>	255	
<i>Jumlah</i>	<i>Varchar</i>	255	
<i>Jumlah_saldo</i>	<i>Varchar</i>	255	
<i>Keluar</i>	<i>Varchar</i>	255	
<i>Ket</i>	<i>Varchar</i>	255	
<i>Pesan</i>	<i>Varchar</i>	255	
<i>Bukti_donasi</i>	<i>Varchar</i>	255	
<i>Saldo</i>	<i>Varchar</i>	255	
<i>Create_at</i>	<i>Timestamp</i>		
<i>Update_at</i>	<i>Timestamp</i>		

3.5. Perancangan Antar Muka

Adapun rancangan antar muka yang akan dibuat pada *Website* Pondok Pesantren Tahfidz Al Quran El ‘Amani adalah sebagai berikut :

3.5.1. *Mockup Tampilan Halaman Utama*

Halaman utama merupakan halaman yang muncul pertama kali ketika pengguna umum mengunjungi *website* pada bagian *front end*. Berikut rancangan tampilan dari halaman utama pada gambar 3.3.

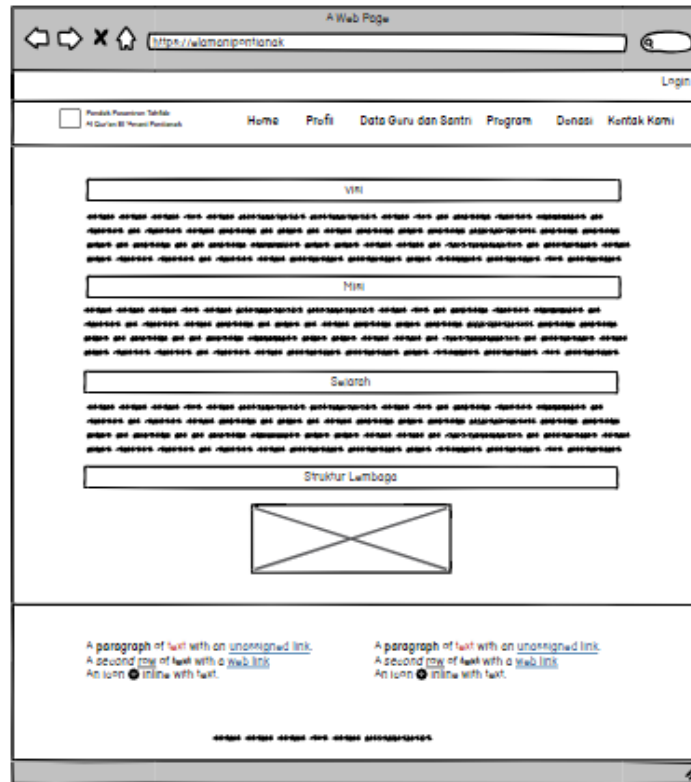


Gambar 3.3. *Mockup* Halaman Utama

Rancangan pada halaman ini terdapat informasi mengenai foto dan nama kedudukan Pondok Pesantren Tahfidz Al Quran El ‘Amani secara umum.

3.5.2. *Mockup Halaman Profil*

Halaman profil merupakan halaman yang kedua saat pengguna umum mengunjungi alamat *website* pada bagian *front end*. Berikut rancangan tampilan dari halaman profil pada gambar 3.4.

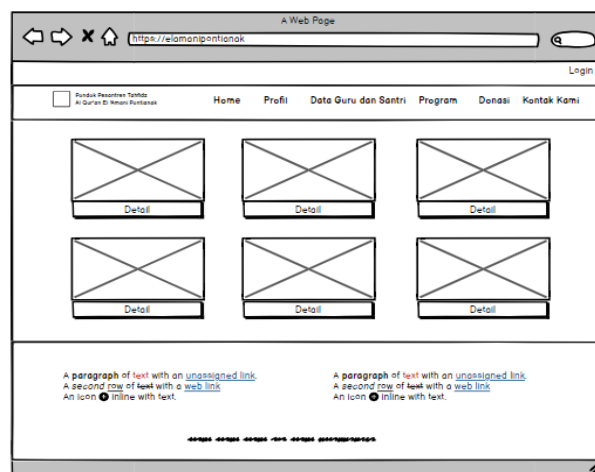


Gambar 3.4. *Mockup* Halaman Profil

Halaman profil merupakan halaman yang menampilkan visi misi, sejarah dan stuktur lembaga Pondok Tahfidz Quran El ‘Amani Pontianak.

3.5.3. *Mockup Halaman Guru dan Santri*

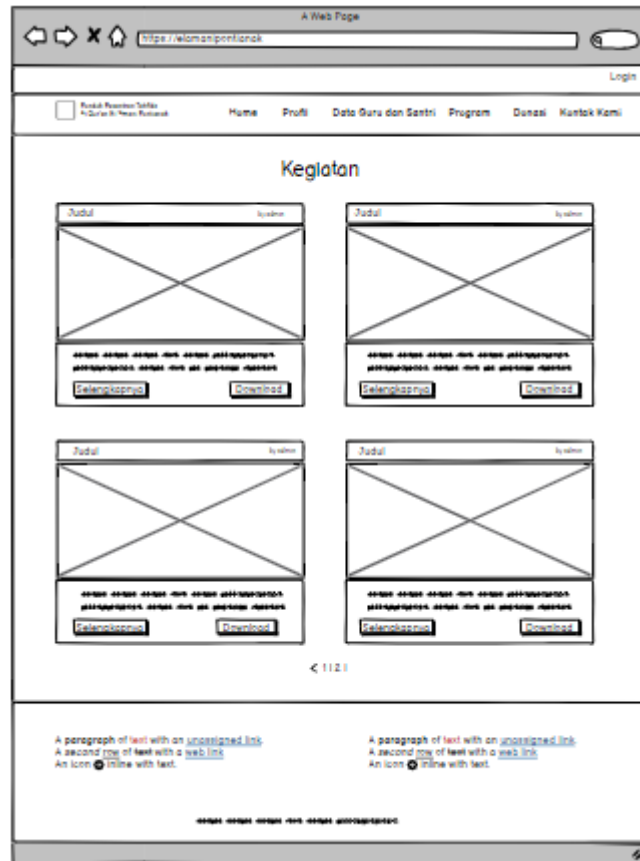
Halaman Guru merupakan halaman yang menampilkan Informasi mengenai data data umum guru guru dan santri Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak. Berikut rancangannya pada gambar 3.5.



Gambar 3.5. *Mockup* Halaman Guru dan Santri

3.5.4. Mockup Halaman Program

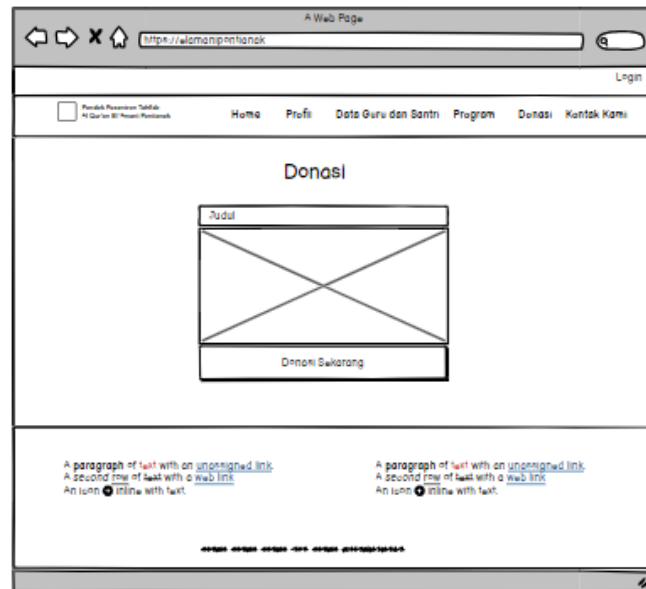
Halaman program merupakan halaman yang menampilkan program yang ada di Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak. Berikut rancangan tampilan dari halaman Program pada gambar 3.6.



Gambar 3.6. Mockup Halaman Program

3.5.5. Mockup Halaman Donasi

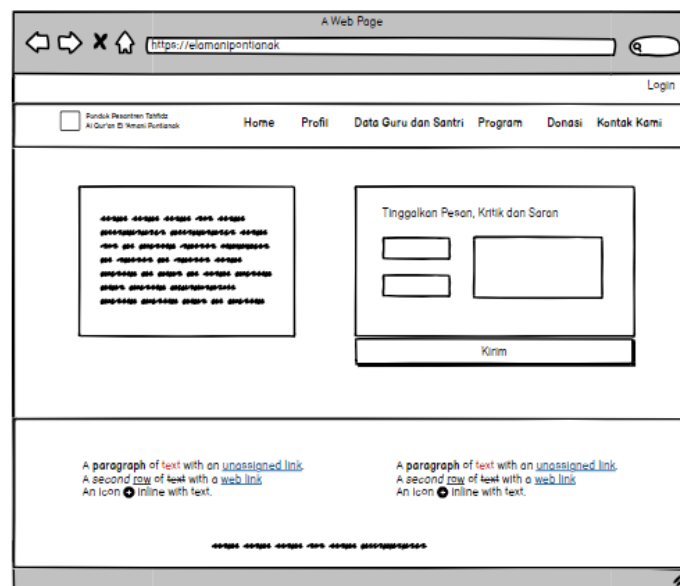
Halaman donasi merupakan tampilan yang menawarkan pengunjung atau tampilan yang dituju oleh calon donatur untuk melakukan/memberikan donasi ke Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak. Tampilan donasi dapat dilihat pada gambar 3.7.



Gambar 3.7. *Mockup* Halaman Donasi

3.5.5. *Mockup* Halaman Kontak Kami

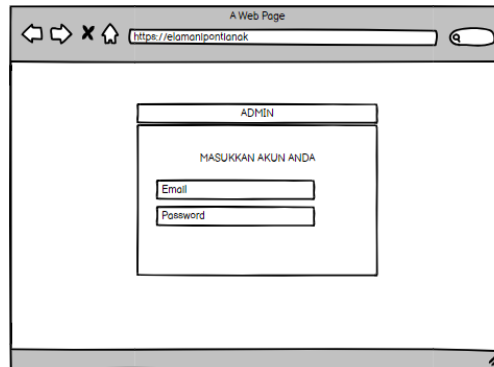
Halaman kontak kami menampilkan informasi lokasi dan nomor pihak pengurus yang berwenang pada Pondok Pesantren Tahfidz Al Quran El Amani Pontianak untuk pengunjung dan donatur yang punya kepentingan untuk menghubungi maupun datang langsung. Pada halaman ini terdapat fitur untuk pengunjung meninggalkan pesan bisa berupa kritik saran maupun doa dan harapan yang akan masuk ke *dashboard* admin sebagai bentuk evaluasi tim pengurus. Adapun tampilan halaman dapat dilihat pada Gambar 3.8.



Gambar 3.8. *Mockup* Halaman Kontak Kami

3.5.6. Mockup Halaman Login

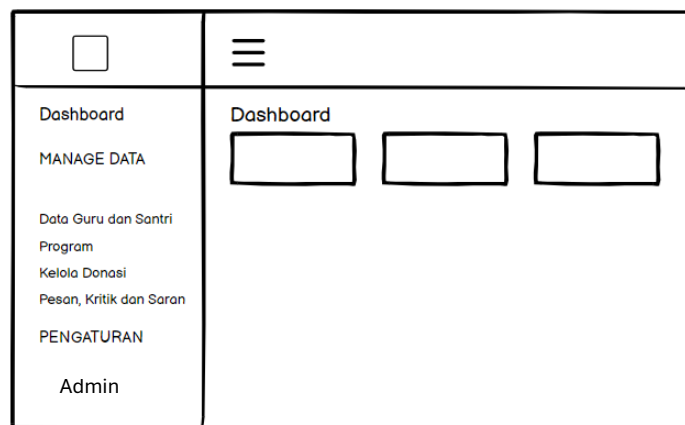
Halaman *Login* merupakan tampilan yang digunakan admin untuk melakukan *login* ke *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Tampilan *login* dapat dilihat pada gambar 3.9.



Gambar 3.9. Mockup Halaman Login

3.5.7. Mockup Halaman Dashboard Admin

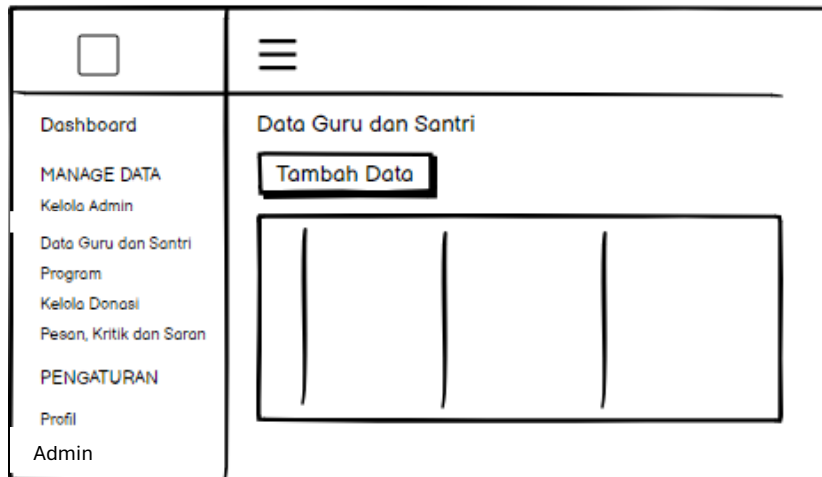
Halaman *dashboard* admin adalah halaman yang pertama kali ditampilkan ketika pengguna dengan hak akses sebagai admin berhasil *login*. Adapun tampilan halaman *dashboard* admin dapat dilihat pada gambar 3.10.



Gambar 3.7. Mockup Halaman Dashboard Admin

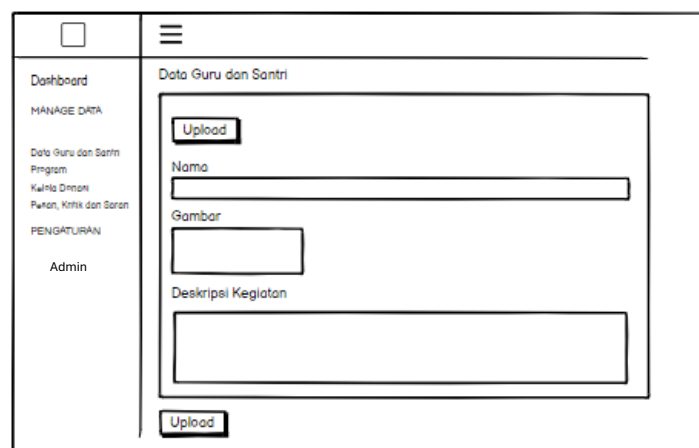
3.5.8. Mockup Halaman Dashboard Data Guru dan Santri

Halaman *dashboard* guru dan santri adalah menu yang dikelola oleh admin. Pada menu ini admin dapat menambah, mengedit dan menghapus data guru dan santri pada *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Berikut rancangan halaman *dashboard* guru dan santri pada gambar 3.11



Gambar 3.11. *Mockup Halaman Dashboard Data Guru dan Santri*

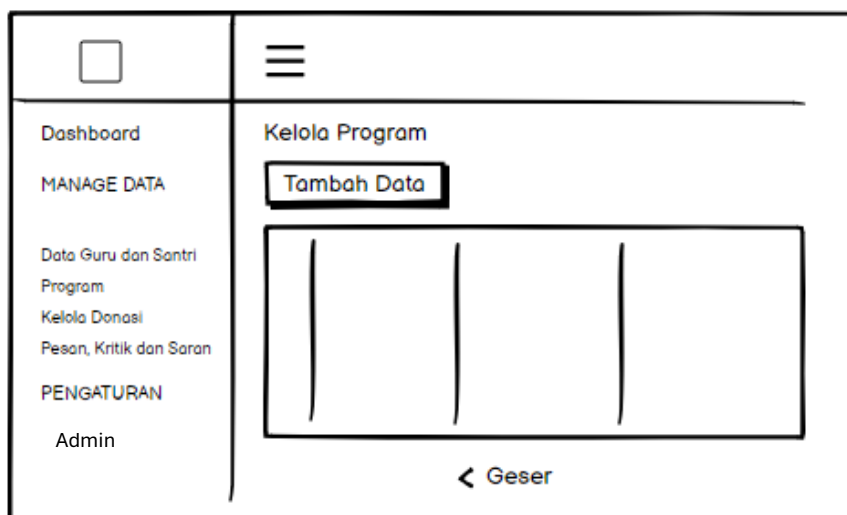
Jika *button* “Tambah data” ataupun *button* “Edit” diklik maka akan muncul halaman berikut, dimana berfungsi untuk menambah data guru dan santri atau mengubah data guru dan santri yang salah (akan muncul data lama yang akan diubah). Adapun tampilan halaman *dashboard* Tambah/Edit Guru dan Santri dapat dilihat pada gambar 3.12.



Gambar 3.12. *Mockup Halaman Dashboard Tambah/Edit Guru dan Santri*

3.5.9. *Mockup Halaman Dashboard Program*

Halaman *dashboard* program adalah menu yang dikelola oleh admin. Pada menu ini admin dapat menambah, mengedit dan menghapus data program pada *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Berikut rancangan halaman *dashboard* program pada gambar 3.13.



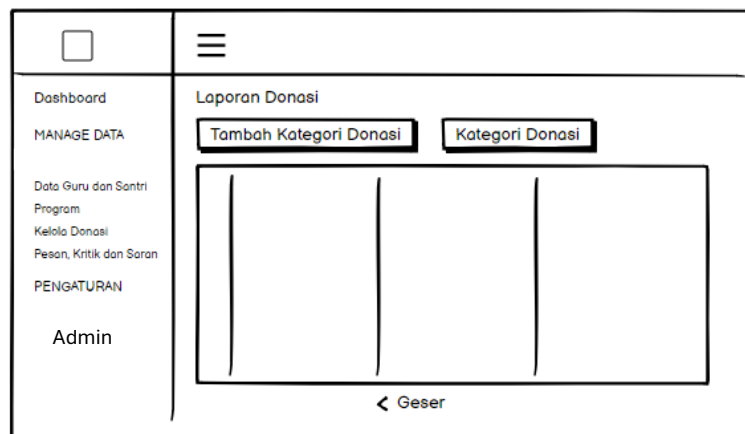
Gambar 3.13. *Mockup Halaman Dashboard Program*

Jika *button* “Tambah Data” ataupun *button* “Edit” diklik maka akan muncul halaman berikut, dimana berfungsi untuk menambah data program atau mengubah data program yang salah (akan muncul data lama yang akan diubah). Berikut rancangan halaman *dashboard* program pada gambar 3.14.

Gambar 3.14. *Mockup Halaman Dashboard Tambah/Edit Program*

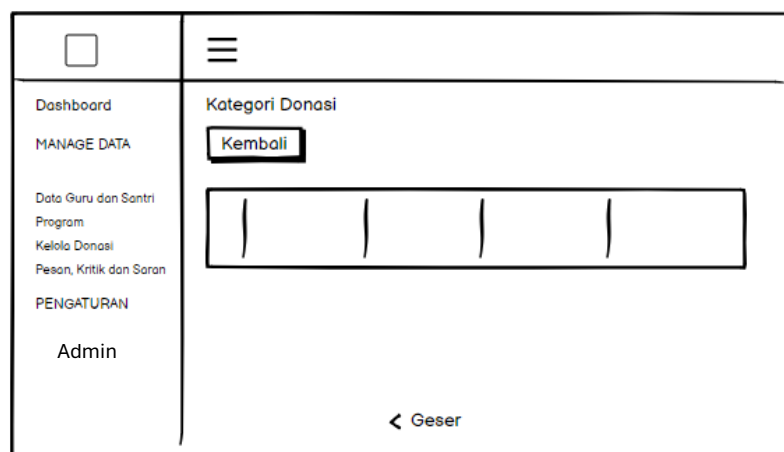
3.5.10. *Mockup Halaman Dashboard Kelola Donasi*

Halaman menu kelola donasi merupakan salah satu menu dari admin untuk mengelola data keuangan. Pada menu ini admin dan donatur hanya dapat melihat laporan keuangan yang masuk dan yang keluar yang telah di input oleh donatur dan admin sesuai kategori. Berikut rancangan halaman *dashboard* laporan keuangan pada gambar 3.15.



Gambar 3.15. *Mockup Halaman Dashboard Laporan Keuangan*

Halaman *dashboard* kategori adalah menu yang dikelola oleh admin. Pada menu ini admin dapat menambah, mengedit dan menghapus data kategori yang ada pada kelola donasi *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Berikut rancangan halaman *dashboard* kategori pada gambar 3.16.



Gambar 3.16. *Mockup Halaman Dashboard Kategori*

Jika *button* “Tambah Kategori Donasi” diklik maka akan muncul halaman berikut, dimana berfungsi untuk menambah kategori.

Gambar 3.17. *Mockup Halaman Dashboard Donasi*

3.5.11. *Mockup Halaman Dashboard Pesan Kritik dan Saran*

Halaman *dashboard* pesan, kritik dan saran adalah menu yang dikelola oleh admin. Pada menu ini admin menampung pesan kritik dan saran yang diberikan/ditinggalkan oleh pengunjung/donatur sebagai bentuk untuk evaluasi tim pengurus atas kritik, kesan dan pesan pengunjung untuk Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Berikut rancangan halaman *dashboard* program pada gambar 3.18.

Gambar 3.18. *Mockup Halaman Dashboard Pesan Kritik dan Saran*

3.5.12. Mockup Halaman *Dashboard Profile Admin*

Halaman *dashboard profile* adalah menu yang dikelola oleh admin. Pada menu ini admin dapat mengubah nama admin sebagai hak akses pada *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak. Berikut rancangan halaman *dashboard* profil pada gambar 3.19.



Gambar 3.19. Mockup Halaman *Dashboard* Profil

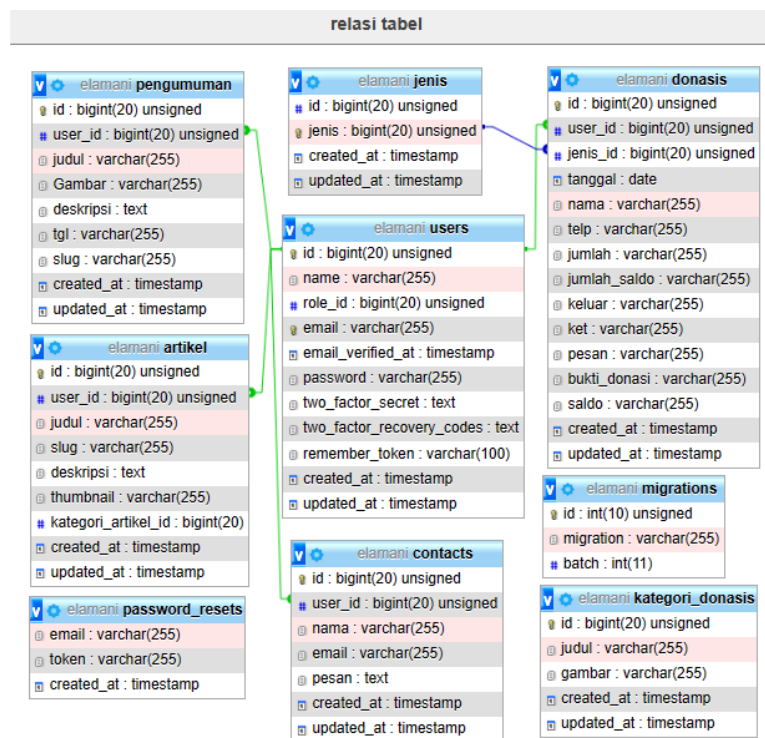
BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan membahas hasil dari perancangan membangun *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak yang telah dilakukan pada bab sebelumnya, pembahasan yang akan dibahas pada bab ini adalah mengenai implementasi setiap halaman adalah sebagai berikut :

1.1. Implementasi *Database*

Implementasi basis data merupakan suatu tahapan dalam proses perancangan basis data. Tahap ini merupakan implementasi dari hasil pemodelan logical dan fisik. Bahasa perintah yang digunakan, baik itu untuk definisi data ataupun penyimpanan data harus sesuai dengan DBMS yang dipilih. Implementasi penyusunan basis data dimulai dari pembuatan berkas-berkas data (tabel-tabel *entity*) kosong yang akan digunakan untuk menyimpan data dalam basis data. Kemudian dilanjutkan dengan pemasukan data untuk tiap instan *entity*.

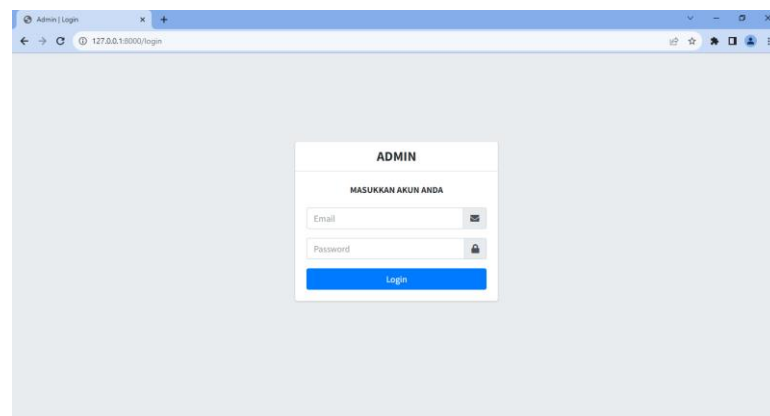


Gambar 4.1. Implementasi *Database*

Relasi tabel yang terlihat pada gambar 4.1. merupakan hubungan relasi antara satu tabel dengan tabel yang lain yang ada dalam sebuah *database*. Relasi tabel melibatkan kolom *primary key* sebuah tabel dengan sebuah kolom *foreign key* dalam tabel lainnya. Kedua kolom inilah yang digunakan untuk membangun sebuah relasi antar tabel.

1.2. Implementasi Halaman Login

Halaman *login* pada *website* berhasil diimplementasikan sesuai dengan rancangan yang telah dibuat sebelumnya, berikut merupakan tampilan hasil implementasi halaman *login* yang dapat dilihat pada Gambar 4.2.



Gambar 4.2. *Implementasi* Halaman Login

Halaman *login* merupakan tampilan yang muncul saat admin dan donatur akan mengakses halaman *dashboard*, berfungsi untuk memverifikasi akun yang terdaftar agar bisa masuk ke halaman *dashboard* sesuai dengan hak akses setiap pengguna.

Berikut potongan kode program yang digunakan untuk membuat halaman login dapat dilihat pada gambar 4.3.

```

<div class="login-box mb-5">
  <div class="card card-outline">
    <div class="card-header text-center">
      <a href="#" class="h4"><b>ADMIN</b></a>
    </div>
    <div class="card-body">
      <p class="login-box-msg">MASUKKAN AKUN ANDA</p>

      <form action="{{ route('login') }}" method="post">
        @csrf
        <div class="input-group mb-3">
          <input value="{{ @old('email') }}" name="email" type="email"
            class="form-control @error('email') is-invalid @enderror" placeholder="Email">
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-envelope"></span>
            </div>
          </div>
          @error('email')
            <div class="invalid-feedback">
              {{ $message }}
            </div>
          @enderror
        </div>
        <div class="input-group mb-3">
          <input value="{{ @old('password') }}" name="password" type="password"
            class="form-control @error('password') is-invalid @enderror" placeholder="Password">
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-lock"></span>
            </div>
          </div>
          @error('password')
            ...
          @enderror
        </div>
        <div class="row">
          <!-- /.col -->
          <div class="col">
            {{-- <a href="admin.users.index" class="btn btn-primary btn-block">Login</a> --}}
            <button type="submit" class="btn btn-primary btn-block">Login</button>
          </div>
          <!-- /.col -->
        </div>
      </form>
    </div>
  </div>
</div>

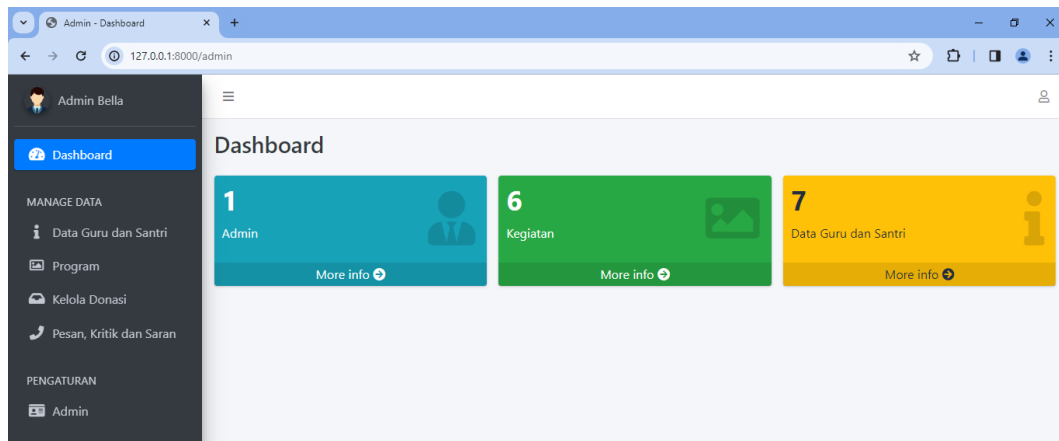
```

Gambar 4.3. Source Code Menampilkan Halaman Login

Saat *administrator* memasukkan data akun di kolom input dan menekan tombol *login*, maka kode program di atas akan berjalan dan melakukan pengecekan data akun yang telah terdaftar, jika proses *login* berhasil pengguna akan diarahkan ke halaman *dashboard*, namun jika data tidak sesuai maka fungsi *loginError* akan memberikan nilai balik bahwa proses login gagal dikarenakan data yang dimasukkan di kolom input tidak sesuai dengan data yang terdaftar.

1.3. Implementasi Halaman *Dashboard* Admin

Halaman *dashboard* pada *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak berhasil diimplementasikan sesuai dengan rancangan yang telah dibuat sebelumnya, berikut merupakan tampilan hasil implementasi halaman *dashboard* yang dapat dilihat pada Gambar 4.4.



Gambar 4.4. Implementasi Halaman *Dashboard* Admin

Halaman *dashboard* admin adalah halaman yang pertama kali ditampilkan ketika pengguna dengan hak akses sebagai admin berhasil *login*. Pada halaman *dashboard* admin menampilkan *scoreboard* yaitu jumlah foto yang ditampilkan pada halaman utama, banyaknya kegiatan, jumlah guru dan santri dan jumlah program.

```
@extends('layouts.backend.app', [
    'title' => 'Dashboard',
    'contentTitle' => 'Dashboard',
])
@section('content')
    <!-- Small boxes (Stat box) -->
    @can('admin1')
        <div class="row justify-content-center ">
            <div class="col-lg-4 col-7">
                <!-- small box -->
                <div class="small-box bg-info">
                    <div class="inner">
                        <h3>@count('users')</h3>
                        <p>Admin</p>
                    </div>
                    <div class="icon">
                        <i class="fas fa-user-tie"></i>
                    </div>
                    <a href="{{ route('admin.users.index') }}" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
                </div>
            </div>
            <div class="col-lg-4 col-7">
                <!-- small box -->
                <div class="small-box bg-success">
                    <div class="inner">
                        <h3>@count('artikel')</h3>
                    </div>
                </div>
            </div>
        </div>
    @endcan
</section>
```

Gambar 4.1. Source Code Menampilkan Halaman *Dashboard* Admin

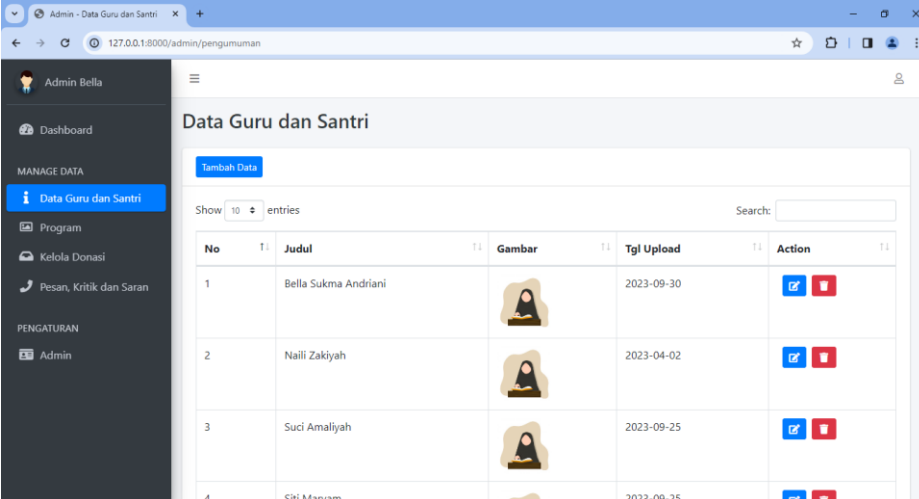
Halaman *dashboard* berfungsi untuk menampilkan jumlah total banyaknya data yang ada. Semua daftar menu yang tampil merupakan hak akses *administrator*. Kode program ini berfungsi menampilkan tombol “More info” yang jika di klik akan mengarah ke data terkait.













1.4. Implementasi Halaman Menu Guru dan Santri

Halaman menu kelola data guru dan santri merupakan salah satu menu dari admin untuk mengelola data guru dan santri yang akan tampil pada *website*. Pada menu kelola data guru dan santri ini admin dapat menambah, mengubah dan menghapus data guru dan santri. Dibawah ini merupakan penjelasan dari fitur menu kelola data guru dan santri.

1.4.1. Lihat Data Guru dan Santri

Pada halaman lihat data guru dan santri ditampilkan daftar guru dan santri yang telah terdaftar pada tabel guru dan santri. Adapun tampilan halaman lihat data guru dan santri dapat dilihat pada Gambar 4.6.



No	Judul	Gambar	Tgl Upload	Action
1	Bella Sukma Andriani		2023-09-30	 
2	Nalli Zakiyah		2023-04-02	 
3	Suci Amaliyah		2023-09-25	 
4	Siti Marvam		2023-09-25	 

Gambar 4.6. Implementasi Halaman *Dashboard* Guru dan Santri

Berikut potongan kode program untuk menampilkan *dashboard* halaman data guru dan santri

```
<div class="card">
  <div class="card-header">
    <a href="{{ route('admin.pengumuman.create') }}" class="btn btn-primary btn-sm">Tambah Data</a>
  </div>
  <div class="card-body table-responsive">
    <table id="dataTables" class="table table-bordered table-hover">
      <thead>
        <tr>
          <th>No</th>
          <th>Judul</th>
          <th>Gambar</th>
          <th>Tgl Upload</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        @php
          $no = 1;
        @endphp
        @foreach ($pengumuman as $pn)
          <tr>
            <td>{{ $no++ }}</td>
            <td>{{ $pn->judul }}</td>
            <td>{{ asset('storage/' . $pn->gambar) }}</td>
            <td>{{ $pn->tgl_upload }}</td>
            <td>
              <a href="{{ route('admin.pengumuman.update', $pn->id) }}" class="btn btn-primary btn-sm">Edit</a>
              <a href="{{ route('admin.pengumuman.delete', $pn->id) }}" class="btn btn-danger btn-sm">Delete</a>
            </td>
          </tr>
        @endforeach
      </tbody>
    </table>
  </div>
</div>
```

Gambar 4.7. *Source Code* Menampilkan Halaman *Dashboard* Guru dan Santri

Pada Gambar 4.7. merupakan tampilan halaman *dashboard* guru dan santri menampilkan semua data guru dan santri untuk ditampilkan pada *website* dan gambar ke halaman *front end* dalam *framework* Laravel diterapkan menggunakan konsep MVC (*model*, *view* dan *controller*).

1.4.2. Tambah Data Guru dan Santri

Halaman tambah data guru dan santri berfungsi untuk menambahkan data guru dan santri. Adapun tampilan halaman tambah data guru dan santri dapat dilihat pada gambar 4.8.

Gambar 4.8. Implementasi Halaman *Dashboard* Guru dan Santri Tambah

Pada halaman tambah data guru dan santri terdapat *form* yang berisi kolom nama, gambar dan deskripsi. Pada bagian bawah *form* terdapat tombol *upload* yang berfungsi menyimpan data dan menampilkan data ke halaman depan yang akan masuk ke *database* tabel galeri dan tombol batal untuk menutup *form* tambah data guru dan santri.

Berikut potongan kode program untuk menampilkan halaman *dashboard* tambah guru dan santri pada gambar 4.9.

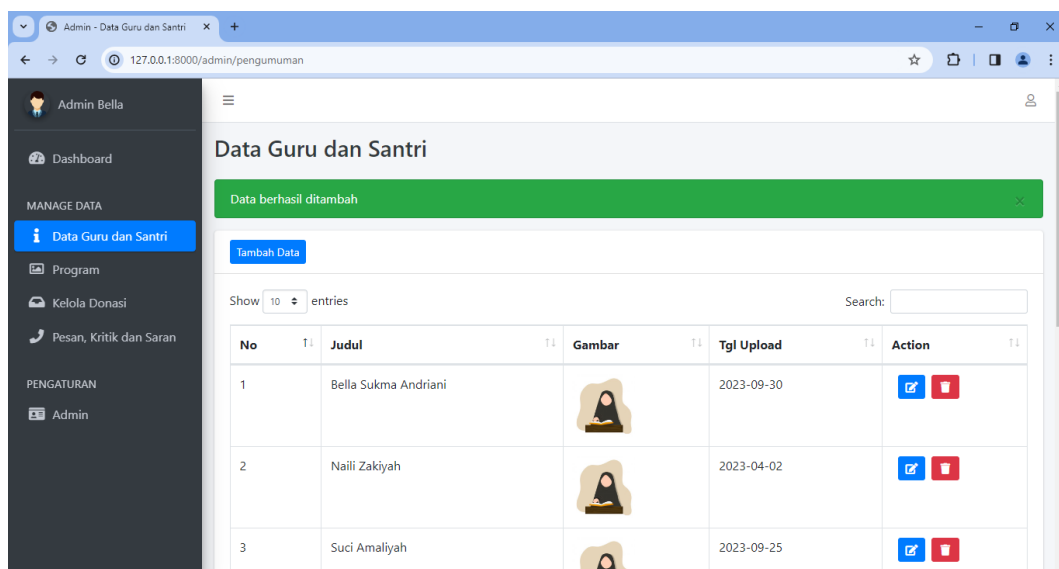

```

<div class="card">
  <div class="card-header">
    <a href="{{ route('admin.pengumuman.index') }}" class="btn btn-success btn-sm">Kembali</a>
  </div>
  <div class="card-body">
    <form method="POST" enctype="multipart/form-data" action="{{ route('admin.pengumuman.store') }}">
      @csrf
      <div class="form-group">
        <label for="judul">Nama</label>
        <input required="" type="text" name="judul" placeholder="" class="form-control title">
      </div>
      <div class="col-lg-6">
        <div class="form-group">
          <label>Gambar</label>
          <input type="file" name="Gambar" class="dropify form-control" data-height="250" data-allowed-file-extensions="png jpg gif jpeg svg webp jfif" required>
        </div>
      </div>
      <div class="form-group">
        <label for="deskripsi">Deskripsi</label>
        <textarea required="" name="deskripsi" id="deskripsi" class="text-dark form-control summernote"></textarea>
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-primary btn-sm">UPLOAD</button>
      </div>
    </div>
  </div>
</form>
</div>

```

Gambar 4.9. Source Code Menampilkan Halaman Tambah Data Guru dan Santri

Halaman tambah data aplikasi berfungsi sebagai kolom untuk *administrator* menambahkan data yang akan ditampilkan ke halaman *front end* dan disimpan ke *database*. Untuk menyimpan data ke database diperlukan fungsi pada bagian *controller*, nilai dari *form* diambil dan diproses di bagian fungsi *store* pada *controller*, berikut hasil implementasi jika data berhasil ditambah pada gambar 4.10.



Gambar 4.10. Implementasi Halaman *Dashboard* Tambah Guru dan Santri

Pada Gambar 4.10. merupakan tampilan halaman *dashboard* tambah data guru dan santri menampilkan semua data yang berhasil diinput atau ditambah oleh admin sebagai hak akses. Jika ingin menambah lebih banyak data guru dan santri admin

bisa mengklik tombol tambah data, lakukan proses perulangan setiap jumlah data yang ada.

1.4.3. Edit Data Guru dan Santri

Halaman edit data guru dan santri berfungsi mengubah data guru dan santri yang ada di *database*. Adapun tampilan halaman edit data galeri dapat dilihat pada gambar 4.11.

Gambar 4.11. Implementasi Halaman *Dashboard* Edit Guru dan Santri

Pada halaman edit data guru dan santri terdapat *form* yang berisi kolom nama, gambar dan deskripsi. Pada bagian bawah *form* terdapat tombol *update* yang berfungsi menyimpan data dan menampilkan data ke halaman depan yang akan masuk ke database tabel galeri dan tombol batal untuk menutup form edit data guru dan santri.

Berikut potongan kode program untuk menampilkan halaman *dashboard* edit guru dan santri pada gambar 4.12.

```

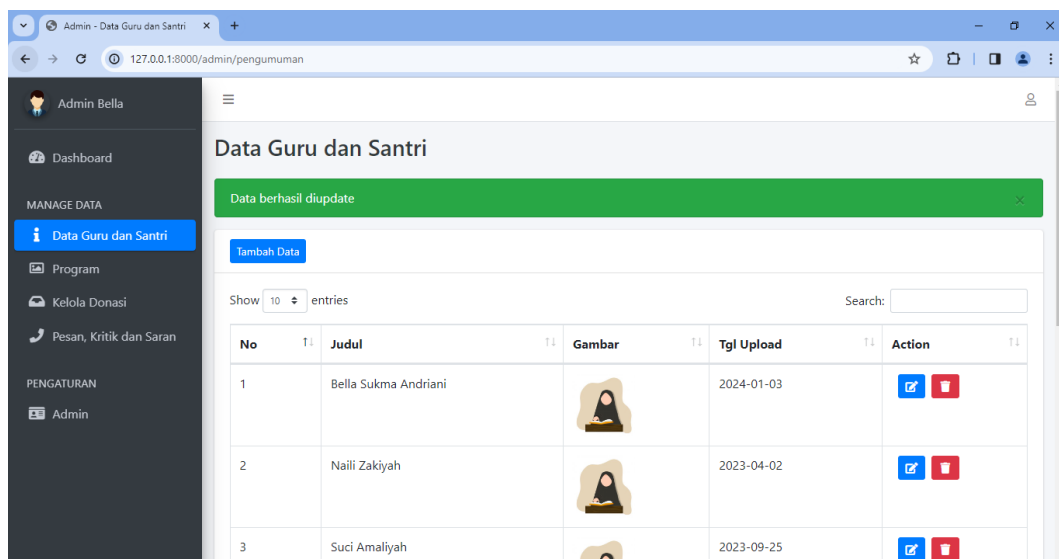
<div class="card">
  <div class="card-header">
    <a href="{{ route('admin.pengumuman.index') }}" class="btn btn-success btn-sm">Kembali</a>
  </div>
  <div class="card-body">
    <form method="POST" enctype="multipart/form-data"
      action="{{ route('admin.pengumuman.update', $pengumuman->id) }}">
      @csrf
      @method('PATCH')
      <div class="form-group">
        <label for="judul">Nama</label>
        <input value="{{ $pengumuman->judul }}" required="" type="text" name="judul" placeholder=""
          class="form-control title">
      </div>
      <div class="form-group">
        <label for="deskripsi">Gambar</label>
        <input type="file" name="Gambar" class="dropify form-control" data-height="250"
          data-allowed-file-extensions="png jpg gif jpeg svg webp jfif">
      </div>
      <div class="form-group">
        <label for="deskripsi">Deskripsi</label>
        <textarea value="" required="" name="deskripsi" id="deskripsi" class="text-dark form-control summernote">
          {!! $pengumuman->deskripsi !!}</textarea>
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-primary btn-sm">UPDATE</button>
      </div>
    </form>
  </div>
</div>

```

Gambar 4.12. *Source Code* Menampilkan Halaman Edit Data Guru dan Santri

Halaman *edit* data aplikasi berfungsi sebagai kolom untuk memperbarui data yang sebelumnya telah dimasukkan oleh *administrator*, perubahan yang dilakukan akan disimpan ke *database*.

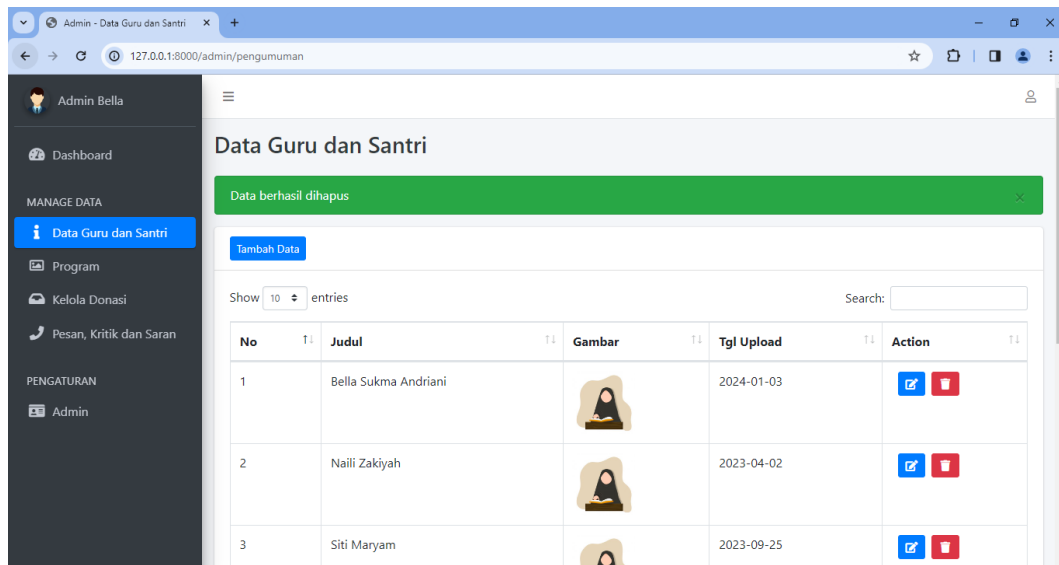
Dalam proses *edit* data dilakukan pada fungsi *update* pada *controller*, kode program mengedit data memiliki kesamaan dengan kode program untuk menyimpan data, dengan perbedaan pada bagian data yang diambil disertai dengan id sebagai tanda pengenal, berikut kode program proses *edit* data. Berikut hasil implementasi jika data berhasil ditambah pada gambar 4.13.



Gambar 4.13. Implementasi Halaman *Dashboard* Edit Guru dan Santri

1.4.4. Hapus Data Guru dan Santri

Pada tabel data guru dan santri terdapat beberapa kolom, pada kolom sebelah kanan terdapat 2 pilihan yaitu edit dan hapus, ketika admin memilih tombol hapus maka akan muncul notifikasi seperti pada gambar dibawah, apabila admin menekan tombol hapus maka data slider tersebut terhapus sedangkan jika admin menekan tombol batal maka data tidak akan dihapus seperti pada gambar 4.14.



Gambar 4.14. Implementasi Halaman *Dashboard* Hapus

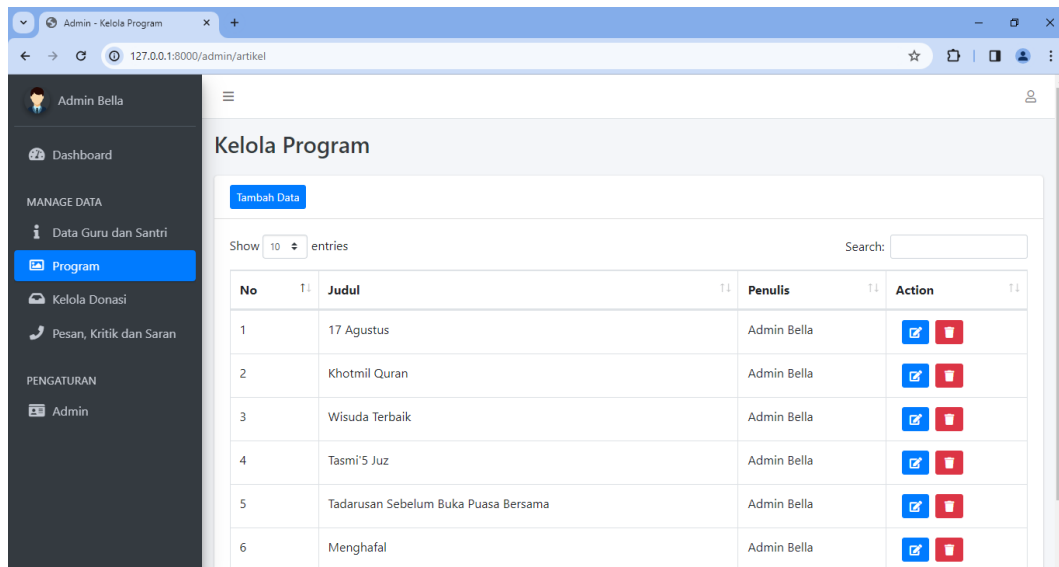
Fitur hapus data guru dan santri berfungsi untuk menghapus data data-data yang dipilih. Saat *administrator* menekan tombol *delete* maka *pop up* konfirmasi akan keluar sebagai persetujuan, apakah yakin atau tidak untuk menghapus data yang dipilih.

1.5. Implementasi Halaman Menu Program

Halaman menu kelola data program merupakan salah satu menu dari admin untuk mengelola data program yang tampil pada *website*. Pada menu kelola data program admin dapat mengedit dan menghapus data program. Dibawah ini merupakan penjelasan dari fitur menu kelola program.

1.5.1. Lihat Data Program

Pada halaman lihat data program ditampilkan daftar program yang telah terdaftar pada tabel program. Adapun tampilan halaman lihat data program dapat dilihat pada gambar 4.15.



Gambar 4.15. Implementasi Halaman *Dashboard* Program

Pada tabel data program terdapat beberapa kolom, pada kolom sebelah kanan terdapat 2 pilihan yaitu edit dan hapus, ketika admin memilih tombol hapus maka akan muncul notifikasi, apabila admin menekan tombol hapus maka data program tersebut terhapus sedangkan jika admin menekan tombol batal maka data tidak akan dihapus.

Berikut potongan kode program untuk menampilkan halaman *dashboard* program pada gambar 4.16.

```
<div class="card">
  <div class="card-header">
    <a href="{{ route('admin.artikel.create') }}" class="btn btn-primary btn-sm">Tambah Data</a>
  </div>
  <div class="card-body table-responsive">
    <table id="dataTable" class="table table-bordered table-hover">
      <thead>
        <tr>
          <th>No</th>
          <th>Judul</th>
          <th>Penulis</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        @php
          $no = 1;
        @endphp

        @foreach ($artikel as $art)
          <tr>
            <td>{{ $no++ }}</td>
            <td>{{ $art->judul }}</td>
            <td>{{ $art->user->name }}</td>
```

Gambar 4.16. Source Code Menampilkan Halaman *Dashboard* Program

Pada Gambar 4.6. merupakan tampilan halaman *dashboard* program menampilkan semua data program untuk ditampilkan pada *website* dan gambar ke halaman *front end* dalam *framework* Laravel diterapkan menggunakan konsep MVC (*model*, *view* dan *controller*).

1.5.2. Tambah Data Program

Halaman tambah data program berfungsi untuk menambahkan data program pada tabel program. Adapun tampilan halaman tambah data program dapat dilihat pada gambar 4.17.

Gambar 4.17. Implementasi Halaman *Dashboard* Program Tambah

Pada halaman tambah data program terdapat *form* yang berisi judul kegiatan, masukkan gambar dan deskripsi. Pada bagian bawah *form* terdapat tombol upload yang berfungsi menyimpan data yang akan masuk ke *database* tabel program dan tombol batal untuk menutup *form* tambah data program.

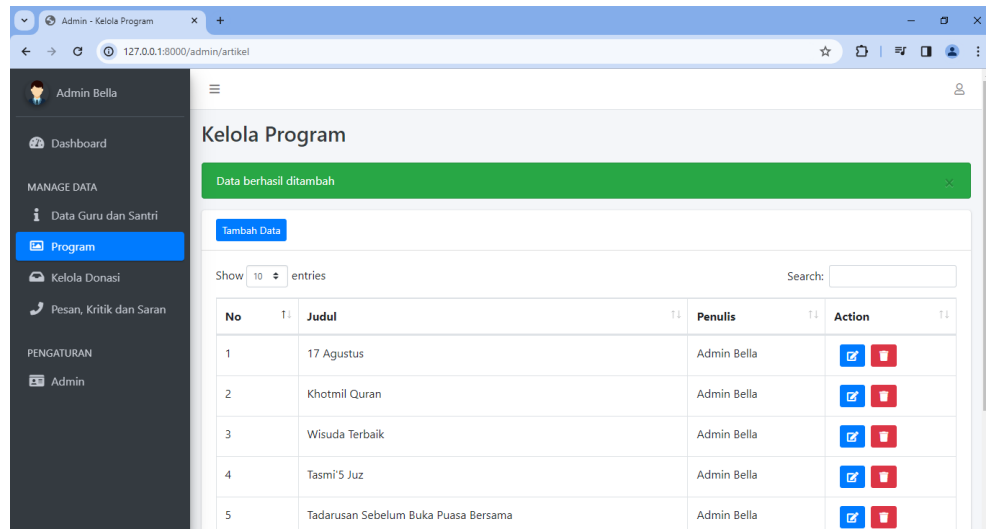
Berikut potongan kode program untuk menampilkan halaman *dashboard* tambah data guru dan santri pada gambar 4.18.

```
<div class="card">
  <div class="card-header">
    <h4 class="card-title">Masukkan Kegiatan Pada Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak/h4>
  </div>
  <div class="card-body">
    <form method="POST" enctype="multipart/form-data" action="{% route('admin.artikel.store') %}">
      <csrf
      <div class="form-group">
        <label for="judul">Judul Kegiatan</label>
        <input required="" type="text" name="judul" placeholder="" class="form-control" required>
      </div>
      <div class="row">
        <div class="col-lg-6">
          <div class="form-group">
            <label>Masukkan Gambar</label>
            <input type="file" name="file" class="dragify form-control" data-height="100" data-allowed-file-extensions="png jpg gif jpeg svg webp jpeg" required>
          </div>
        </div>
        <div class="col-lg-6">
          <div class="form-group">
            <label>Kategori</label>
            <select required="" class="form-control" name="kategori_artikel_id">
              <option selected="" disabled=""> PILIH KATEGORI </option>
              <foreach ( $kategori as $kategori )>
                <option value="{ $kategori->id }">{ $kategori->nama_kategori }</option>
              </foreach>
            </select>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Gambar 4.18. Source Code Halaman *Dashboard* Tambah Data Program

Halaman tambah data program berfungsi sebagai kolom untuk *administrator* menambahkan data yang akan ditampilkan ke halaman *front end* dan disimpan ke

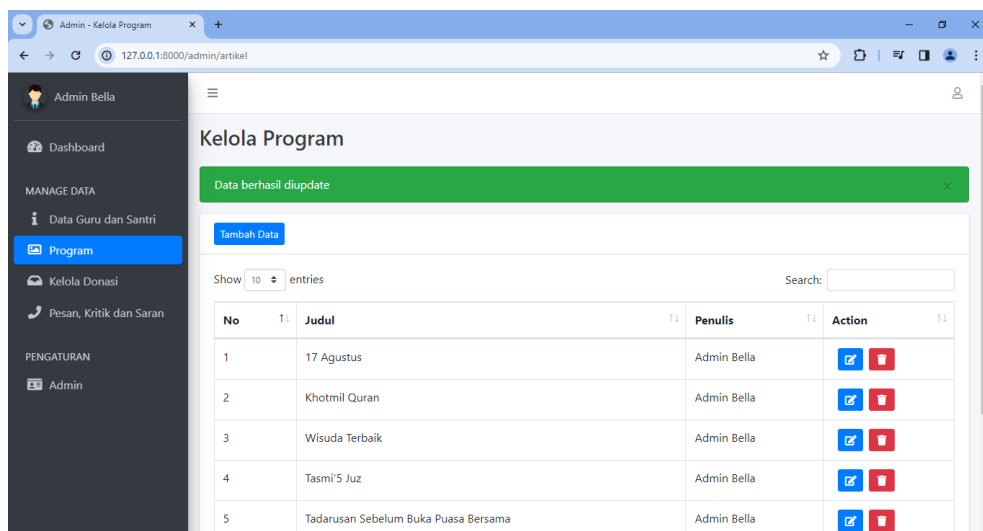
database. Untuk menyimpan data ke *database* diperlukan fungsi pada bagian *controller*, nilai dari *form* diambil dan diproses di bagian fungsi *store* pada *controller*, berikut hasil implementasi jika data berhasil ditambah pada gambar 4.19.



Gambar 4.19. Implementasi Halaman *Dashboard* Edit Program

1.5.3. Edit Data Program

Halaman edit data program berfungsi mengubah data program yang ada di *database*. Adapun tampilan halaman edit data program dapat dilihat pada gambar 4.20.



Gambar 4.20. Implementasi Halaman *Dashboard* Program Edit

Halaman *edit* data aplikasi berfungsi sebagai kolom untuk memperbarui data yang sebelumnya telah dimasukkan oleh *administrator*, perubahan yang dilakukan akan disimpan ke *database*.

Berikut potongan kode program untuk menampilkan halaman *dashboard* edit data guru dan santri pada gambar 4.21.

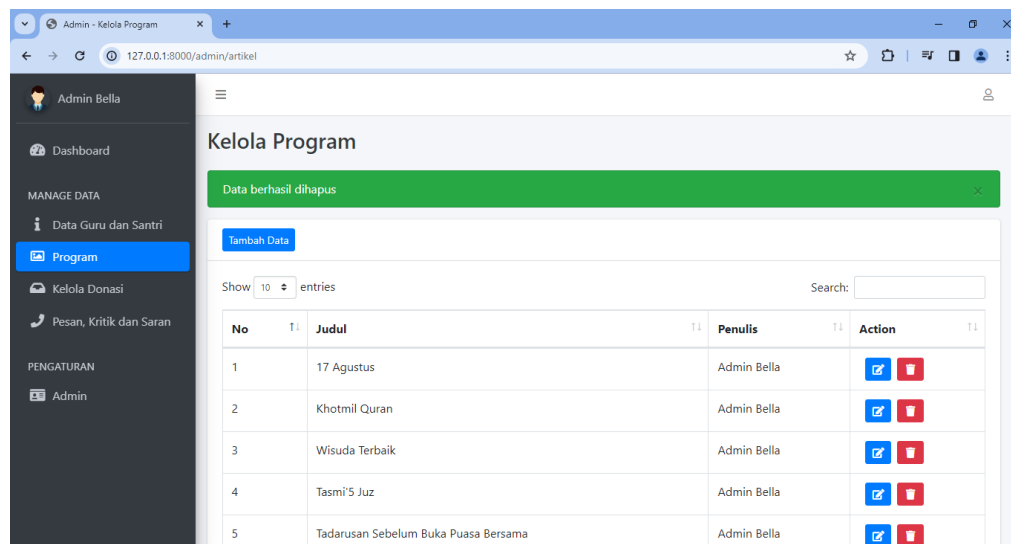
```
<div class="card">
  <div class="card-header">
    <h4 class="card-title">Edit Kegiatan Pada Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak</h4>
  </div>
  <div class="card-body">
    <form method="POST" enctype="multipart/form-data"
      action="{{ route('admin.artikel.update', $artikel->id) }}"
      {{ method_field('PUT') }}
      @csrf
    >
      <div class="form-group">
        <label for="judul">Judul Kegiatan</label>
        <input value="{{ $artikel->judul }}" required="" type="text" name="judul" placeholder=""
          class="form-control">
      </div>
      <div class="row">
        <div class="col-lg-6">
          <div class="form-group">
            <label>Masukkan Gambar</label>
            <input type="file" name="file" class="dropify form-control" data-height="190"
              data-allowed-file-extensions="png jpg gif jpeg svg webp jfif" required>
          </div>
        </div>
        <div class="col-lg-6">
          <div class="form-group">
            <label>Kategori</label>
            <select required="" class="form-control" name="kategori_artikel_id">
              <option selected="" disabled="">- PILIH KATEGORI -</option>
              @foreach ($kategoriArtikel as $kategori)
                <option value="{{ $kategori->id }}">{{ $kategori->nama_kategori }}</option>
              @endforeach
            </select>
          </div>
        </div>
      </div>
      <div id="form-group">
        <label for="deskripsi">Deskripsi Kegiatan</label>
        <textarea required="" name="deskripsi" id="deskripsi" class="text-dark form-control summernote">{{ $artikel->deskripsi }}
      </div>
    </div>
    <div class="card-footer">
      <button type="submit" class="btn btn-primary">UPDATE</button>
    </div>
  </form>
</div>
```

Gambar 4.21. *Source Code* Menampilkan *Form* Tambah dan Edit Data Program

Dalam proses *edit* data dilakukan pada fungsi *update* pada *controller*, kode program mengedit data memiliki kesamaan dengan kode program untuk menyimpan data, dengan perbedaan pada bagian data yang diambil disertai dengan id sebagai tanda pengenalan, berikut kode program proses *edit* data.

1.5.4. Hapus Data Program

Fitur hapus data program berfungsi untuk menghapus data program yang dipilih. Adapun tampilan hapus data program dapat dilihat pada gambar 4.22.

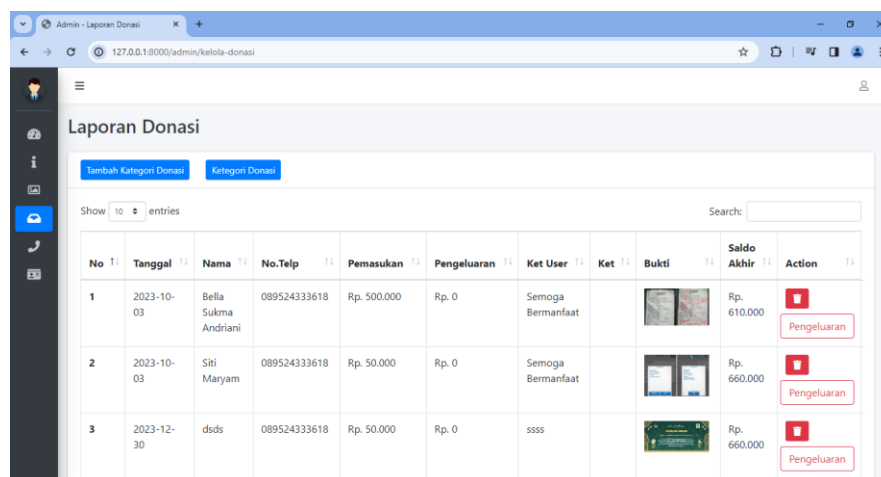


Gambar 4.22. Implementasi Halaman *Dashboard* Program Hapus

Pada tabel data program terdapat beberapa kolom, pada kolom sebelah kanan terdapat 2 pilihan yaitu edit dan hapus, ketika admin memilih tombol hapus maka akan muncul notifikasi seperti pada gambar diatas, apabila admin menekan tombol hapus maka data program tersebut terhapus sedangkan jika admin menekan tombol batal maka data tidak akan dihapus.

1.6. Implementasi Halaman Menu Laporan Keuangan

Halaman menu laporan keuangan merupakan salah satu menu penting dari admin untuk mengelola data keuangan dari *website*. Pada menu laporan data keuangan admin dapat melihat, menambahkan, mengedit dan menghapus data keuangan. Adapun tampilan menu keuangan dapat dilihat pada gambar 4.23.



Gambar 4.23. Implementasi Halaman *Dashboard* Laporan Keuangan

Pada halaman keuangan ditampilkan data keuangan dan opsi yang terdiri dari edit dan hapus. Pada bagian atas tabel terdapat tombol tambah data yang berfungsi untuk menambahkan data keuangan.

Berikut potongan kode program untuk menampilkan halaman *dashboard* laporan keuangan pada gambar 4.24.

```
<body>
@foreach ($daftardonasi as $donasi)
{{-- @if($donasi->jenis_id != '2') --}}
<tr>
<th scope="row">{{ $loop->iteration }}</th>
<td>{{ $donasi->tanggal }}</td>
<td>{{ $donasi->nama }}</td>
<td>{{ $donasi->telp }}</td>
{{-- <td>{{ $donasi->jumlah }}</td> --}}
<td>Rp. {{ number_format($donasi->jumlah, 0, ',', '.') }}</td>
<td>Rp. {{ number_format($donasi->keluar, 0, ',', '.') }}</td>
<td>{{ $donasi->pesan }}</td>
<td>{{ $donasi->ket }}</td>
<td></td>
<td>Rp. {{ number_format($donasi->saldo, 0, ',', '.') }}</td>

<td class="row">
<form method="POST" onsubmit="return confirm('Apakah Anda Yakin');"
action="{{ route('has-donasi.destroy', $donasi->id) }}"
@csrf
@method('DELETE')
<button type="submit" class="btn btn-danger btn-sm ml-2 mr-2"><i
class="fas fa-trash fa-fw"></i></button>
</form>
<a href="/admin/kelola-donasi/pengeluaran/create/{{ $donasi->id }}"
class="btn btn-outline-danger mt-2">Pengeluaran</a>
</td>
</tr>
{{-- @endif --}}
@endforeach
</body>
```

Gambar 4.24. *Source Code* Halaman *Dashboard* Laporan Keuangan

Pada saat menampilkan data donasi menggunakan number_format disisi *controller* pada saat data ingin ditampilkan di *view*. Di laravel dapat menambahkan *directive custom* dimana di dalam *directive custom* ini bertujuan untuk melakukan format mata uang. Kode diatas berfungsi untuk membuat format rupiah dan *directive custom* di blade dengan nama *currency*. Denga menambahkan *directive custom currency* maka tidak perlu melakukan format number di sisi *controller* pada setiap *controller*.

1.6.1. Tambah Kategori Donasi

Halaman tambah kategori donasi berfungsi untuk melakukan donasi berserta bukti transfer. Adapun tampilan halaman tambah donasi dapat dilihat pada gambar 4.24.

Gambar 4.24. Implementasi Halaman *Dashboard* Donasi Tambah

Pada halaman tambah donasi terdapat *form* yang berisi kolom judul dan gambar. Pada bagian bawah *form* terdapat tombol upload yang berfungsi menyimpan data yang akan tercatat oleh sistem dan tombol batal untuk menutup *form*.

Berikut potongan kode program untuk menampilkan halaman *dashboard* tambah kategori donasi pada gambar 4.25.

```
<div class="card">
  <div class="card-header">
    <a href="{{ route('Kelola-Donasi') }}" class="btn btn-success btn-sm">Kembali</a>
  </div>
  <div class="card-body">
    <form action="{{ route('donasi.store') }}" method="POST" enctype="multipart/form-data" >
      @csrf
      <div class="form-group">
        <label for="judul">Judul</label>
        <input required="" type="text" name="judul" placeholder="" class="@error('judul') is-invalid @enderror form-control title">
      </div>
      <div class="col-lg-6">
        <div class="form-group">
          <label>Gambar</label>
          <input type="file" name="gambar" class="@error('gambar') is-invalid @enderror dropify form-control" data-height="250" required>
        </div>
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-primary btn-sm">UPLOAD</button>
      </div>
    </div>
  </form>
</div>
```

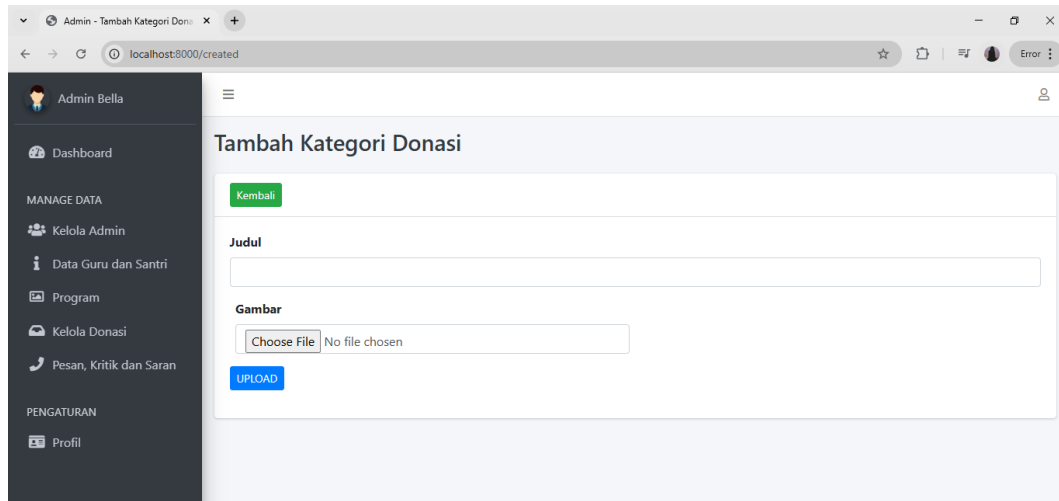
Gambar 4.25. *Source Code* Halaman *Dashboard* Tambah Kategori Donasi

Halaman tambah data kategori donasi berfungsi sebagai kolom untuk *administrator* menambahkan data yang akan ditampilkan ke halaman *front end* dan disimpan ke *database*. Untuk menyimpan data ke database diperlukan fungsi pada

bagian *controller*, nilai dari *form* diambil dan diproses di bagian fungsi *store* pada *controller*.

1.6.2. Edit Kategori Donasi

Halaman edit data donasi berfungsi mengubah kategori. Adapun tampilan halaman edit kategori donasi dapat dilihat pada gambar 4.26.



Gambar 4.26. Implementasi Halaman Edit Kategori Donasi

Halaman *edit* data aplikasi berfungsi sebagai kolom untuk memperbarui data yang sebelumnya telah dimasukkan oleh *administrator*, perubahan yang dilakukan akan disimpan ke *database*. Berikut potongan kode program untuk menampilkan halaman *dashboard* edit kategori donasi pada gambar 4.27.

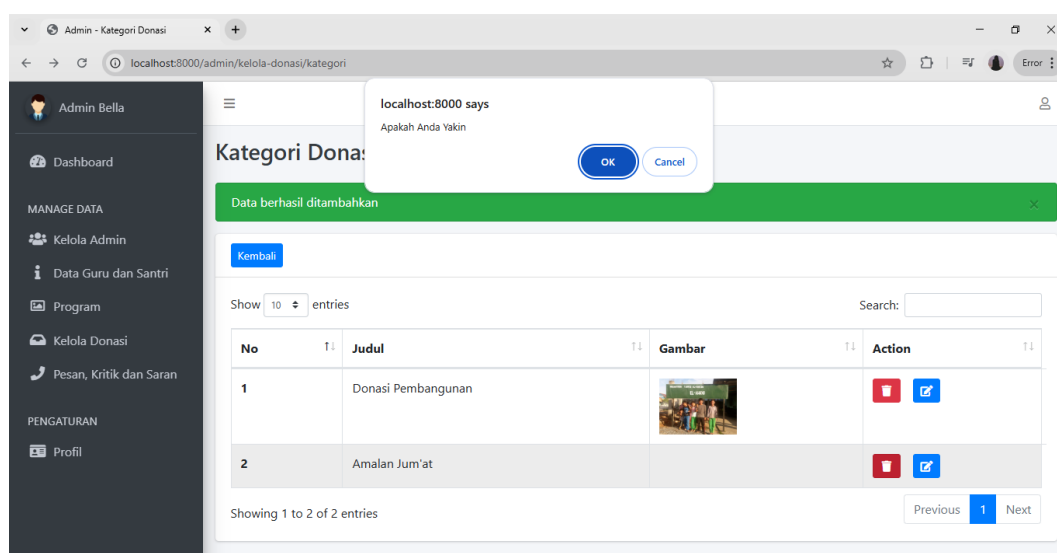
```
<div class="card">
  <div class="card-header">
    <a href="{{ route('donasi.kategori') }}" class="btn btn-success btn-sm">Kembali</a>
  </div>
  <div class="card-body">
    <form method="POST" enctype="multipart/form-data" action="{{ route('donasi.update', $kdbid->id) }}" >
      @csrf
      <div class="form-group">
        <label for="judul">Judul</label>
        <input value="{{ $kdbid->judul }}" required="" type="text" name="judul" placeholder="" class="@error('judul') is-invalid @enderror form-control title">
      </div>
      <div class="col-lg-6">
        <div class="form-group">
          <label>Gambar</label>
          <small class="text-danger">*Klik Jika Ingin Diubah</small>
          <input type="file" name="gambar" class="@error('gambar') is-invalid @enderror dropify form-control" data-height="250" >
        </div>
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-primary btn-sm">UPLOAD</button>
      </div>
    </form>
  </div>
</div>
```

Gambar 4.27. Menampilkan Halaman *Dashboard* Edit Kategori Donasi

Dalam proses *edit* data dilakukan pada fungsi *update* pada *controller*, kode program mengedit data memiliki kesamaan dengan kode program untuk menyimpan data, dengan perbedaan pada bagian data yang diambil disertai dengan id sebagai tanda pengenalan, berikut kode program proses *edit* data.

1.6.3. Hapus Kategori Donasi

Fitur hapus data kategori donasi berfungsi untuk menghapus data kategori donasi yang dipilih. Adapun tampilan hapus data kategori donasi dapat dilihat pada gambar 4.28.

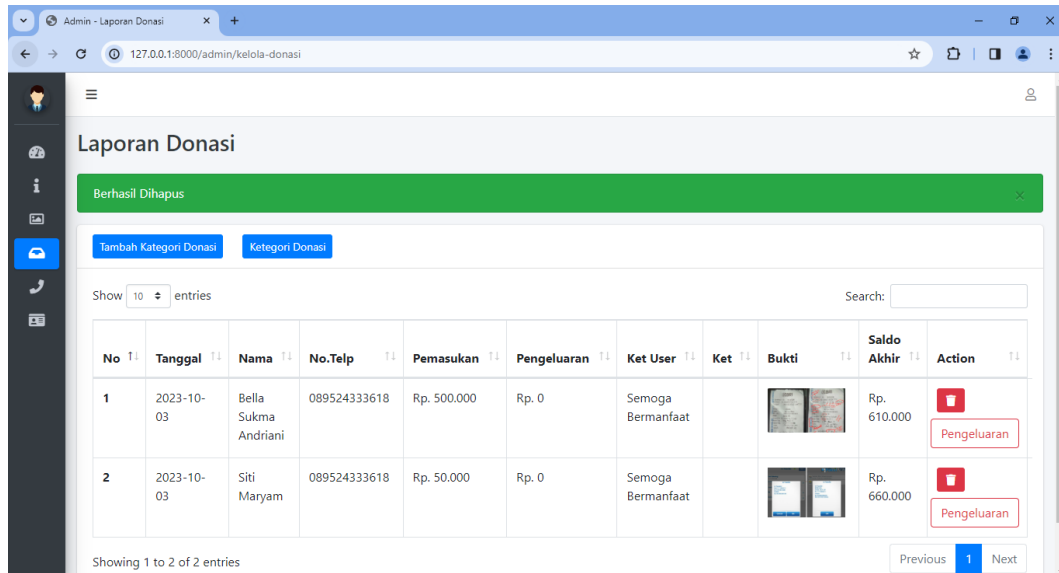


Gambar 4.28. Implementasi Halaman *Dashboard* Program Hapus

Pada tabel data program terdapat beberapa kolom, pada kolom sebelah kanan terdapat 2 pilihan yaitu edit dan hapus, ketika admin memilih tombol hapus maka akan muncul notifikasi seperti pada gambar diatas, apabila admin menekan tombol hapus maka data kategori donasi terpilih terhapus sedangkan jika admin menekan tombol batal maka data tidak akan dihapus.

1.6.4. Hapus Data Donasi

Fitur hapus data donasi berfungsi untuk menghapus data donasi. Adapun tampilan hapus data donasi dapat dilihat pada gambar 4.29.

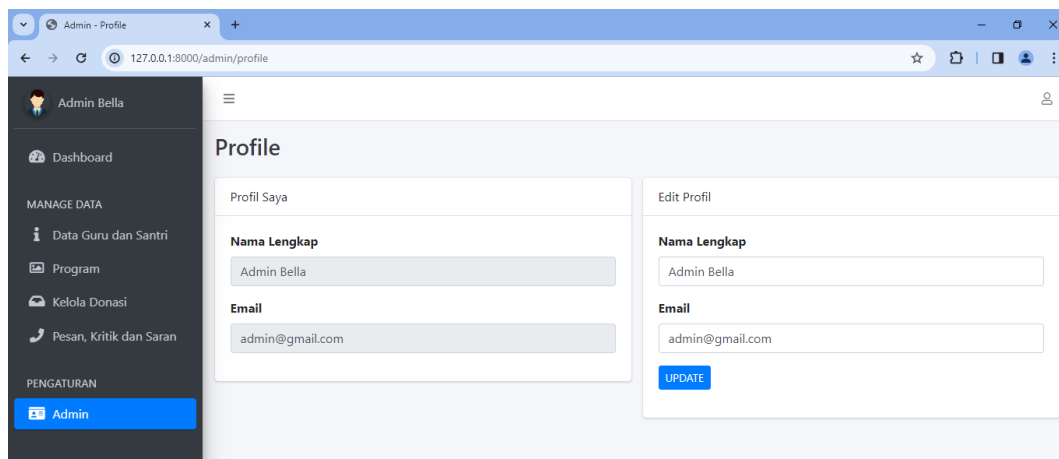


Gambar 4.29. Implementasi Halaman *Dashboard* Donasi Hapus

Pada tabel data donasi terdapat beberapa kolom, pada kolom sebelah kanan terdapat 2 pilihan yaitu edit dan hapus, ketika admin memilih tombol hapus maka akan muncul notifikasi seperti pada gambar diatas, apabila admin menekan tombol hapus maka data program tersebut terhapus sedangkan jika admin menekan tombol batal maka data tidak akan dihapus.

1.7. Implementasi Halaman Admin

Halaman menu profil merupakan fitur untuk dapat melihat dan data pribadi admin sebagai hak akses *dashboard* website Pondok Pesantren Tahfidz Al Qur'an El Amani Pontianak. Adapun halaman menu profil dapat dilihat pada gambar 4.30.



Gambar 4.30. Implementasi Halaman *Dashboard* Admin

Pada halaman data admin pada bagian edit profil terdapat *form* yang berisi kolom nama lengkap dan email, admin dapat mengubah data pribadi pada kolom tersebut. Pada bagian bawah *form* terdapat tombol *update* yang berfungsi untuk menyimpan data admin terbaru.

Berikut potongan kode program untuk menampilkan halaman *dashboard* admin pada gambar 4.31.

```
<div class="row">
  <x-card>
    <x-slot name="col">6</x-slot>
    <x-slot name="header">Profil Saya</x-slot>
    <div class="form-group">
      <label>Nama Lengkap</label>
      <input type="text" disabled="" value="{{ auth()->user()->name }}" class="form-control" name="">
    </div>
    <div class="form-group">
      <label>Email</label>
      <input type="text" disabled="" value="{{ auth()->user()->email }}" class="form-control" name="">
    </div>
  </x-card>

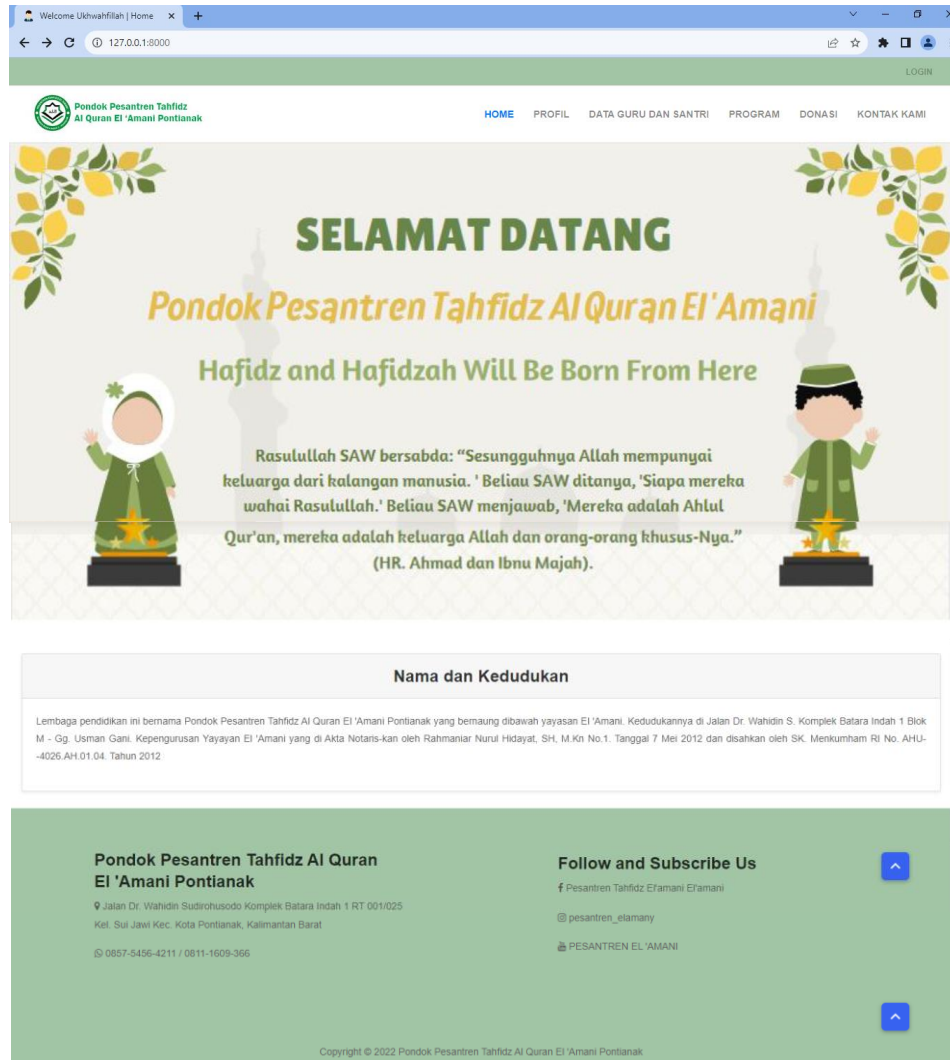
  <x-card>
    <x-slot name="col">6</x-slot>
    <x-slot name="header">Edit Profil</x-slot>
    <form method="POST" action="{{ route('user-profile-information.update') }}">
      @csrf
      @method('put')
      <div class="form-group">
        <label>Nama Lengkap</label>
        <input type="text" value="{{ auth()->user()->name }}" class="form-control @error('name') is-invalid @enderror" name="name">
        @error('name')
        <div class="invalid-feedback">
          {{ $message }}
        </div>
        @enderror
      </div>
      <div class="form-group">
        <label>Email</label>
        <input type="text" value="{{ auth()->user()->email }}" class="form-control @error('email') is-invalid @enderror" name="email">
        @error('email')
        <div class="invalid-feedback">
          {{ $message }}
        </div>
        @enderror
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-primary btn-sm">UPDATE</button>
      </div>
    </form>
  </x-card>
</div>
```

Gambar 4.31. *Source Code* Menampilkan *Form* Menu Profil

Untuk menampilkan data pada tabel dibutuhkan data dari model *Post* dan fungsi yang menampilkan data di bagian *controller*, berikut merupakan kode program untuk menampilkan data pada tabel *Posts*. Data aplikasi disimpan dalam variabel *\$posts*, setelah itu data dikirimkan ke halaman tampilan (*view*). Pada bagian *view* setiap data ditampilkan melalui proses perulangan, selanjutnya data ditampilkan ke dalam komponen HTML dengan cara *\$post->nama_kolom*. Untuk menyimpan data ke *database* diperlukan fungsi pada bagian *controller*, nilai dari *form* diambil dan diproses di bagian fungsi *store* pada *controller*, berikut kode program untuk menyimpan data.

1.8.Implementasi Halaman User

Menu beranda berhasil diimplementasikan sesuai dengan rancangan yang telah dibuat sebelumnya. Halaman beranda merupakan halaman awal yang ditampilkan ketika *website* diakses. Adapun tampilan halaman ini dapat dilihat pada gambar 4.32. di bawah ini.



Gambar 4.32. Implementasi Halaman *Home*

Halaman ini menampilkan informasi berupa foto, nama dan kedudukan secara umum Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak. Pada bagian *footer* terdapat informasi alamat serta sosial media yang bisa diakses untuk mengetahui informasi mengenai Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak dari media lain.

1.9. Implementasi Halaman Profil

Halaman Profil menampilkan informasi mengenai visi misi, sejarah dan stuktur lembaga pada Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak. Adapun tampilan halaman ini dapat dilihat pada Gambar 4.33.

The screenshot shows a web browser displaying the profile page of Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak. The page layout includes a header with navigation links (HOME, PROFIL, DATA GURU DAN SANTRI, PROGRAM, DONASI, KONTAK KAMI) and a main content area with the following sections:

- Visi:** "Menjadi lembaga pendidikan yang mampu mengantarkan peserta didik menjadi sumber daya manusia yang religius, mencintai Allah dan Rasulnya, memiliki kompetensi al-Qur'an dan kebahasaan yang tinggi, profesional serta berjiwa kreatif-proaktif yang diadopsi dengan moral dan etika yang baik".
- Misi:**
 - a. Menjamin Penyelenggara kegiatan pembelajaran tahfidz al-Qur'an standar
 - b. Menjamin Penyelenggara pembelajaran bahasa dari soft skill, khususnya penggunaan bahasa Arab dan bahasa Inggris
 - c. Menjamin dukungan kompetensi penguasaan life skill dengan sasaran agar peserta didik dapat meningkatkan kompetensi akademik, meningkatkan profesionalisme dan mengembangkan jaringan kerjasama, serta
 - d. Meningkatkan kualitas hidup masyarakat melalui pendidikan agama Islam yang berbasis pada al-Qur'an dan nilai-nilai ajaran Islam
- Sejarah Pendirian:**

Kekhidupan dan keistiqomahan (integritas) Drs. H. Noor Faiz dan Hj. Rif'ati dalam mengasuh peminat tahfidz (hapalan) al-Qur'an (yatim, du'afa dan masyarakat umum) telah menggerakkan hati Drs. Hj.Siti Habibah binti H.Umar Thaha untuk ikut serta bersama para donatur dan pendukung terdahulu dalam proyek khayrulkum man ta'allamal Qur'an wa 'allamahu l'yang terbaik diantaranya adalah yang mempelajari al-Qur'an dan mengajarkannya. Untuk itu, "Mbah" Habibah dalam keterbatasan materinya kemudian mewaafkan rumah dan mengajak anak-menantu cucu, kerabat, sahabat dan masyarakat muslim lainnya untuk tidak ketinggalan dalam mewujudkan cita-cita lamaran universal umat, yakni "Menghidupkan al-Qur'an" melalui lembaga pendidikan dan pembelajaran al-qur'an Pesantren Tahfidz el-Amani

Alhamdulillah, dalam waktu yang tidak terlalu lama, santri-santri berdatangan sehingga bukan hanya rumah waqaf el-Amani, namun tumpangan rumah Bunda (ghiet di sebelahnya pun sudah tidak memadai bagi para santri. Kondisi ini mengingatkan para pengelola bahwa ta'lim al-Qur'an yang berorientasi pada profit ukhrowi berbasis jaji lahi merupakan sesuatu yang muskil dan harus dibagi dengan masyarakat luas. Kebutuhan pesantren Tahfidz el 'Amani terhadap tempat belajar dan asrama para santri hanya memungkinkan melalui kerja bersama (tatawву) sambil berdoa semoga Allah memberikan dan ma'laikat mencatat amal kita amin
- Struktur Lembaga:**

SUSUNAN ORGANISASI YAYASAN EL 'AMANI

 - PENDIRI YAYASAN** → 1. Drs. Hj. Siti Habibah, 2. Drs. H. Noor Faiz, 3. Hj. Rif'ati
 - PEMBAWA** → 1. Ketua: Dr. H. Dar Surya Alimasa, MA; 2. Anggota: Dr. H. Wajidi Sayah, M. Ag; H. Ulf Adhikri, Lc., M. Ag; Muhammad Saleh, ST, MT; Roflan Faiz
 - PENGAWAS** → 1. Ketua: H. H. Supriat, Iah; 2. Anggota: H. Rifa'i; Miftahul Huda, S. Pd; Yuzenda Windaatun, ST; H. Esi Syah Hayati, MP
 - PENGURUS** → 1. Ketua: Drs. H. Noor Faiz; 2. Sekretaris: Fati Kusumayanti, S. Soc., M. Si; 3. Bendahara: Drs. H. Siti Habibah
 - Bidang Pendidikan** → 1. Ketua: Drs. H. Moh. Amin Santoso, M. Ag; 2. Anggota: Wismarini, S. Ag, MA; Moh. Faza' Amin, S., MA; Fati Sulmarani, S. Psi., M. Psi
 - Bidang Ekonomi** → 1. Ketua: H. Rudy Effendi, S. Ag; 2. Anggota: Tri Nugro Setiawan, DE; Hidayat, ST; H. Esi Sana, S. Ag; Eka Rudy Adhanyah; Luthi Kurni

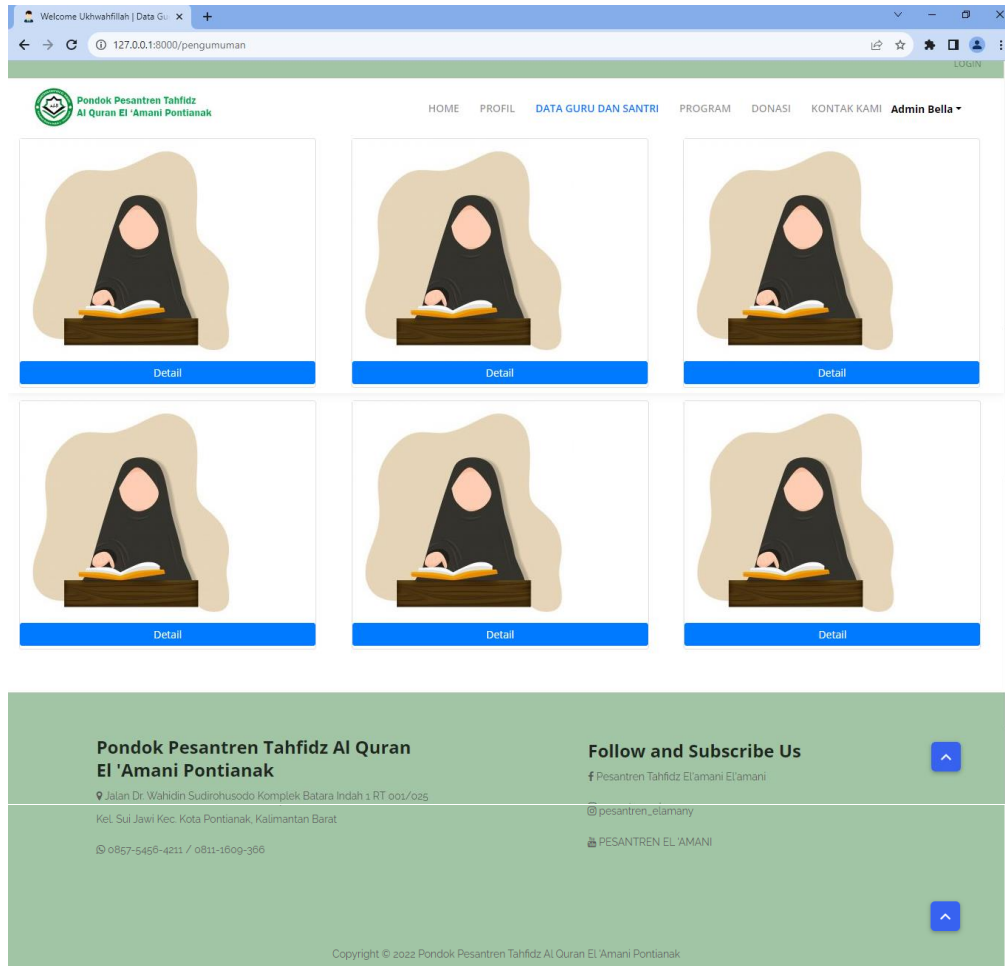
The footer contains contact information for Pondok Pesantren Tahfidz Al Quran El 'Amani Pontianak, including the address (Jalan Dr. Wahidin Sudirhusodo Komplek Batara Indah 1 RT 000/009, Kel. Sul Jawa Koc, Kota Pontianak, Kalimantan Barat), phone numbers (0878-5493-4211 / 0812-1809-386), and social media links (Facebook: Pesantren Tahfidz El 'Amani Elamany, Instagram: pesantren_elamany, Telegram: PESANTREN EL 'AMANI).

Gambar 4.33. Implementasi Halaman Profil

Pada halaman profil merupakan halaman yang berisikan informasi tentang informasi visi dan misi, selain itu halaman profil ini juga berisi struktur lembaga atau susunan organisasi yayasan el 'amani.

1.10. Implementasi Tampilan Halaman Guru dan Santri

Halaman guru dan santri menampilkan informasi mengenai data-data guru dan santri yang terdaftar atau masih berperan aktif dalam kegiatan operasional Pondok. Adapun tampilan halaman ini dapat dilihat pada Gambar 4.34.

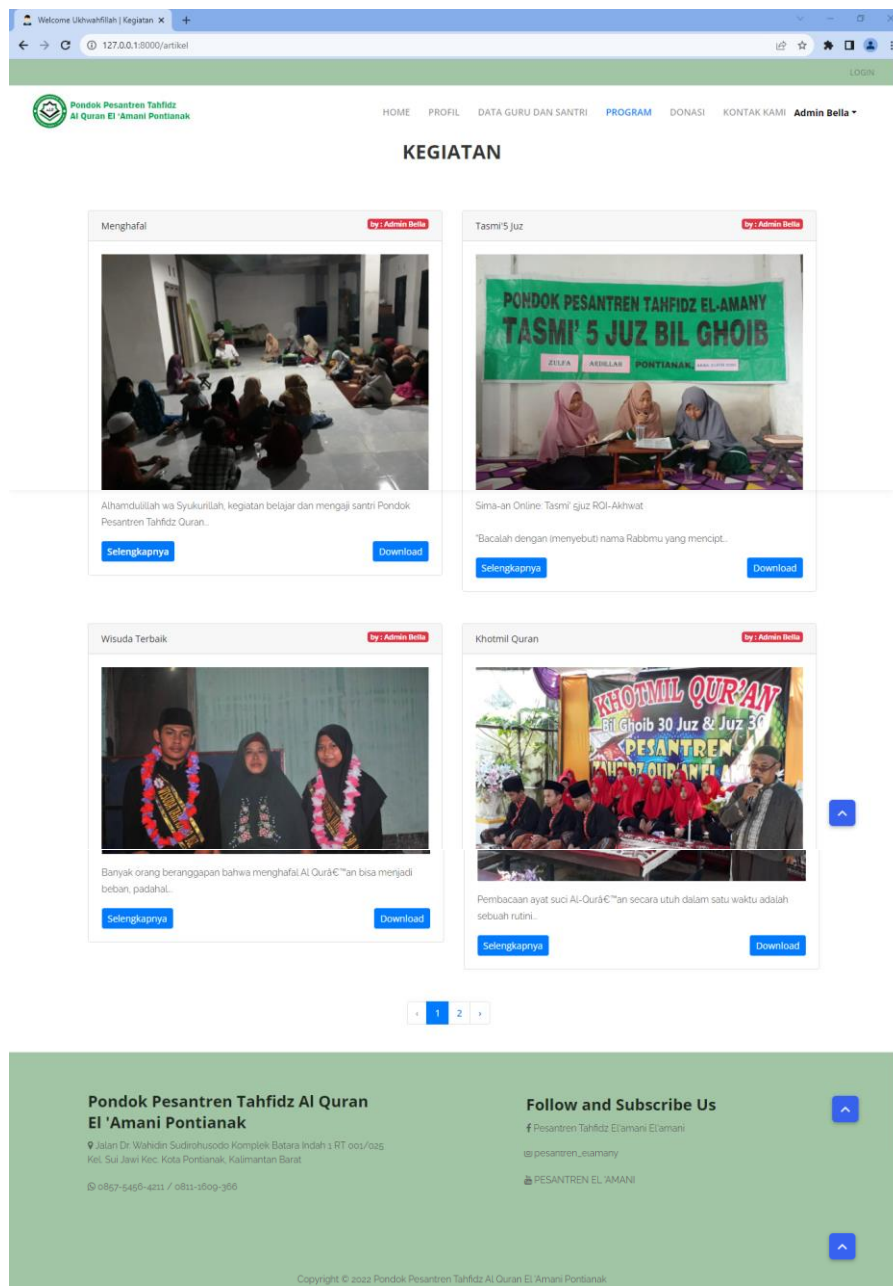


Gambar 4.34. Implementasi Halaman Guru Dan Santri

Halaman guru dan santri berhasil diimplementasikan sesuai dengan rancangan yang telah dibuat sebelumnya. Halaman ini berfungsi untuk menampilkan data data guru dan santri sudah terdaftar, pada halaman ini terdapat tombol detail untuk melihat informasi mengenai data seseorang yang dicari maka informasi yang dipaparkan akan lebih spesifik.

1.11. Implementasi Halaman Program

Halaman program menampilkan informasi mengenai program, kegiatan atau rutinitas operasional apa saja yang ada di Pondok Tahfidz Al Quran El 'Amani Pontianak sehingga pengunjung lebih valid mendapatkan informasi untuk mendaftarkan anak anak mereka. Adapun tampilan halaman ini dapat dilihat pada Gambar 4.35.



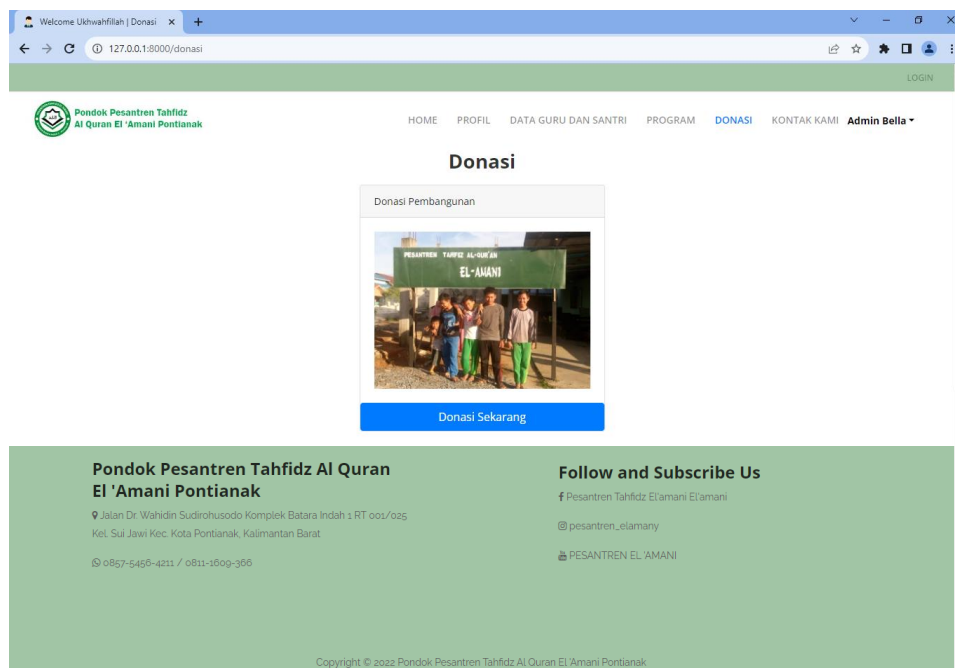
Gambar 4.35. Implementasi Halaman Program

Halaman program berhasil diimplementasikan sesuai dengan rancangan yang telah dibuat sebelumnya. Halaman ini berfungsi untuk menampilkan data program

yang tersedia. Pada halaman ini terdapat tombol selengkapnya untuk melihat informasi mengenai data seseorang yang dicari maka informasi yang dipaparkan akan lebih spesifik.

1.12. Implementasi Halaman Donasi

Halaman donasi menampilkan card yang menawarkan pengunjung untuk berdonasi. Caranya sangat mudah hanya mengisi *form* sesuai data yang diminta dan konfirmasi dengan mengirimkan bukti transfer, maka data kan masuk ke *dashboard* admin. Adapun tampilan halaman ini dapat dilihat pada gambar 4.36.



Gambar 4.36. Implementasi Halaman Donasi

Halaman donasi berhasil diimplementasikan sesuai dengan rancangan yang telah dibuat sebelumnya. Halaman ini berfungsi untuk menampilkan donasi yang sedang diselenggarakan pihak Pondok Tahfidz Al Quran El ‘Amani Pontianak. Pada halaman ini terdapat tombol untuk “Donasi Sekarang” sebagai alat pengguna untuk berkontribusi.

The screenshot shows a web browser window with the URL '127.0.0.1:8000/donasi'. The page title is 'Welcome UkhwahFikrah | Donasi'. The main content is a form titled 'Donasi Pembangunan'. The form fields are: 'Nama Donatur' (Masukkan Nama), 'Inpute Tanggal' (mm/dd/yyyy --:-- --), 'No Telepon' (No Telepon), 'Nominal' (Masukkan Nominal), 'Bukti Donasi' (Choose File, No file chosen), and 'Keterangan' (Keterangan). At the bottom of the form are 'Close' and 'Kirim' buttons. The footer of the page includes 'Pondok Pesantren Tahfidz Al Quran El Amani Pontianak' and a 'Subscribe Us' button.

Gambar 4.37. Implementasi Halaman Donasi

Ketika *button* “Donasi Sekarang” diklik pada halaman donasi maka akan muncul *card* yang terdapat *form* yang berisi kolom nama donatur, tanggal, nomor telepon, nominal, bukti transaksi transferan dan terakhir keterangan. Pada bagian bawah *form* terdapat tombol kirim yang berfungsi menyimpan data yang akan tercatat oleh sistem dan tombol batal untuk menutup *form*.

1.13. Implementasi Halaman Kontak Kami

Halaman kontak kami menampilkan informasi lokasi dan nomor pihak pengurus yang berwenang pada Pondok Pesantren Tahfidz Al Quran El Amani Pontianak untuk pengunjung dan donatur yang punya kepentingan untuk menghubungi maupun datang langsung. Adapun tampilan halaman dapat dilihat pada gambar 4.38.

The screenshot shows a web browser window with the URL '127.0.0.1:8000/contact/create'. The page title is 'Welcome UkhwahFikrah | Kontak'. The main content is divided into two sections: 'Info Kontak' and 'Tinggalkan Pesan, Kritik dan Saran'. The 'Info Kontak' section lists: 'No Telp' (085754564211/08111609366), 'Instagram' (@pesantren_elamany), 'Facebook' (Pesantren Tahfidz El'Amani), 'Youtube' (Pesantren El'Amani), and 'Jl.Dr.Wahidin S. Komplek Batara Indah 1 RT 001/025, Kel.Sungai Jawi, Kec. Kota Pontianak, Kalimantan Barat'. The 'Tinggalkan Pesan, Kritik dan Saran' section has input fields for 'Nama' and 'Email', a 'Pesan' text area, and a 'KIRIM' button. The footer includes 'Pondok Pesantren Tahfidz Al Quran El Amani Pontianak', contact details, and social media links. A copyright notice at the bottom reads 'Copyright © 2022 Pondok Pesantren Tahfidz Al Quran El Amani Pontianak'.

Gambar 4.38. Implementasi Halaman Kontak Kami

Pada halaman ini juga terdapat *form* pesan, kritik dan saran yang berfungsi untuk menampung apapun yang menjadi pendukung berkembangnya Pondok Pesantren Tahfidz Quran El ‘Amani Pontianak, masukkan data yang akan tercatat oleh sistem setelah tombol kirim diklik dan masuk ke *database*.

1.14. Pengujian Sistem

Pengujian sistem dimaksudkan untuk menguji semua fitur – fitur yang dibuat apakah sudah sesuai dengan yang diharapkan. Pengujian sistem yang digunakan dalam *website* ini adalah metode pengujian *black box*. Pengujian *black box* merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Daftar pengujian dapat dilihat pada beberapa tabel dibawah ini.

1) Tabel Pengujian Pengunjung

Pada tabel pengunjung dilakukan pengujian terkait fitur-fitur yang terdapat didalamnya. Pengujian dilakukan ketika user pertama kali masuk kehalaman beranda. Di halaman beranda terdapat menu home, profil, guru dan santri, donasi, program dan kontak kami.

Tabel 4.1. Pengujian *Users*

No	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1	Mengakses halaman utama	Menampilkan halaman home	sesuai
2	Memilih menu profil	Menampilkan halaman profil	sesuai
3	Memilih menu guru dan santri	Menampilkan halaman guru dan santri	sesuai
4	Memilih menu program	Menampilkan halaman program	sesuai
4	Memilih menu donasi	Menampilkan halaman donasi	sesuai
5	Memilih menu kontak kami	Menampilkan halaman kontak kami	sesuai

2) Tabel Pengujian *Form* Admin

Pada tabel ini yang menjalankannya adalah admin. Admin dapat dapat mengakses menu-menu yang ada seperti menu *dashboard*, slider, galeri, pesantren dengan sub-menu profil, visi & misi, sejarah struktur lembaga, guru dan santri, program, serta keuangan dengan sub-menu kategori dan kas. Disetiap menu terdapat fitur tambah, edit dan hapus.

Tabel 4.2. Pengujian *Form* Admin

No	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1	Memilih menu <i>dashboard</i>	Menampilkan halaman <i>dashboard</i>	sesuai
2	Memilih menu kelola admin	Menampilkan halaman kelola admin	sesuai
3	Memilih menu program	Menampilkan halaman berisi data program/kegiatan	sesuai
4	Memilih menu guru dan santri	Menampilkan halaman berisi data guru dan santri	sesuai
5	Memilih menu laporan keuangan	Menampilkan halaman berisi data keuangan	sesuai
6	Memilih menu pesan, kritik dan saran	Menampilkan halaman pesan, kritik dan saran	sesuai
7	Memilih menu profil admin	Menampilkan halaman profil admin	sesuai

BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan yang dapat penulis ambil dari membangun *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak adalah sebagai berikut:

1. Telah berhasil dibangun sebuah *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak dengan sistem penggunaannya yang dibedakan menjadi 2 aktor dari sistem ini terbagi menjadi dua, yaitu administrator dan donatur/user. Pengguna umum dapat melakukan donasi dimana donatur harus mengirimkan bukti transfer.
2. Pada *website* juga terdapat fitur donasi yang bertujuan untuk mempermudah pengguna umum/donatur dalam menyalurkan donasi, selain itu *website* ini juga sangat membantu pengurus (admin) dalam mengelola donasi yang masuk dan keluar.
3. *Website* dapat memiliki laporan data pemasukan dan pengeluaran dimana pemasukan berasal dari donatur/user yang melakukan donasi dan pengeluaran dari admin dan hasil bersih dari pemasukan dikurang pengeluaran.

5.2. Saran

Adapun saran yang dapat membantu dalam pengembangan *website* Pondok Pesantren Tahfidz Al Quran El ‘Amani Pontianak ini adalah sebagai berikut:

1. Perlu ditingkatkan lagi fitur donasi dengan menyediakan fitur transfer secara langsung. Saat ini beragam fitur baru turut dikembangkan demi meningkatkan kualitas layanan seperti adanya QRIS Transfer yang memungkinkan pengguna dapat melakukan transfer baik sebagai pengirim atau penerima dana, serta melakukan penarikan dan penyetoran uang tunai hanya dengan melakukan scan QRIS pada merchant atau ATM.
2. Mengembangkan *website* dengan pesan dalam *website* yang memungkinkan pengguna berkomunikasi satu sama lain atau mendukung dalam *website*.
3. *Website* yang telah dirancang belum tersedia riwayat donasi pengguna, diharapkan untuk pengembangan selanjutnya untuk mempertimbangkan sisi performa dengan membuat *website* menerapkan pelacakan dan riwayat donasi yang memungkinkan pengguna melihat status dan riwayat donasi mereka.

DAFTAR PUSTAKA

- [1] ranahresearch, "Pengertian Metode Waterfall dan Tahap-Tahapnya," Ranah Research, [Online]. Available: Pengertian Metode Waterfall dan Tahap-Tahapnya. [Accessed 2 Maret 2022].
- [2] F. R. D. S. Ibnu Dwi Lesmono, "Rancang Bangun Sistem Informasi Pendaftaran Santri Baru Berbasis Web pada Pondok Pesantren Al Falah," *Konferensi Nasional Ilmu Sosial & Teknologi (KNiST)*, pp. 181-185, 2014.
- [3] W. Kurniasih, Rancang Bangun Website Pondok Pesantren Darul Qur'an Wadda'wah, Pontianak, 2020.
- [4] S. Zaelani, Sistem Informasi Akademik Pondok Pesantren Mamba'ul Hisan Isyhar Berbasis Web, 2017.
- [5] R. Hidayat, in *Cara Praktis Membangun Website Gratis*, Jakarta, PT Elex Media Komputindo, 2010, p. 2.
- [6] S. Anhar, in *Panduan Menguasai PHP dan MySQL Secara Otodidak*, Jakarta Selatan, PT TransMedia, 2010, p. 40.
- [7] R. A. C. P. Dadi Hasanudin, in *Sistem Pendukung Keputusan Tentang Beasiswa Menggunakan Metode Fuzzy MADM dan SAW*, Bandung, Kreatif Industri Nusantara, 2020, p. 33.
- [8] H. P. Y. S. Kusno Harianto, in *Sistem Monitoring Lulusan Perguruan Tinggi Dalam Memasuki Dunia Kerja Menggunakan Tracer Study*, Surabaya, Media Sahabat Cendekia, 2019, p. 14.
- [9] A. Nawadwipa, "Apa itu PHP dan apa kegunaannya," Nawadwipa, 26 Februari 2021. [Online]. Available: <https://www.nawadwipa.co.id/apa-itu-php-dan-apa-kegunaannya>. [Accessed 3 Maret 2022].
- [10] M. B. Muhammad Hendra Sunarya, in *Pemrograman Internet*, Banjarmasin, Deepublish dan Politeknik Banjarmasin, 2020, p. 16.
- [11] S. Tonni Limbong, in *Pemrograman Web Dasar*, Yayasan Kita Menulis, 2021, p. 69.
- [12] T. Yulianto, "Pengenal PHP," in *Pengenal PHP*, Jakarta, 2007, 2007, pp. 1-2.
- [13] H. Yuliansyah, "Perancangan replikasiBasis Data MySQL," *Jurnal Informatika*, vol. VIII, no. 1, pp. 1-11, 2014.
- [14] idCloudHost, Belajar Cepat Framework Codeigniter untuk pemula, idCloudHost, 2020.
- [15] W. Widyantoro, Buku Panduan Bootstrap, Jakarta: academia.edu, 2009.
- [16] G. W. Sasmito, "Penerapan Metode Waterfall pada Desain Sistem Informasi Geografis," *Jurnal Pengembangan IT(JPIT)*, vol. II, no. 3, pp. 1-7, 2017.
- [17] A. Aha, Membangun Aplikasi Katering Makanan dan Minuman Berbasis Website Pada Kota Pontianak Kalimantan Barat, Pontianak: Tugas Akhir Analia Aha , 2018.
- [18] P. Wulandari, "Rancang bangun aplikasi dapur online (D-Line) berbasis website," *Media Komputer* , vol. II, no. 13, pp. 40-50, 2012.

- [19] A. R. A. d. A. P. Rina Firliana1, " Sistem Informasi Pemesanan Katering Berbasis Web," *Jurnal Basis Web*, vol. II, no. 10, pp. 20-22, 2016.
- [20] B. Intermedia, "Apa itu Laravel? Fitur dan Kelebihan Laravel dibanding Framework lainnya," JagoanHosting_, 5 Februari 2020. [Online]. Available: <https://www.jagoanhosting.com/blog/framework-laravel/>. [Accessed 3 Maret 2022].