

TUGAS AKHIR
RANCANG BANGUN APLIKASI BERBASIS *WEBSITE*
PADA MAKNA *WEDDING ORGANIZER*

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



OLEH:
RAHMAT RAIHAN BAIHAQI
NIM. 3202016038

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

2023

HALAMAN PENGESAHAN

RANCANG BANGUN APLIKASI BERBASIS *WEBSITE* PADA MAKNA *WEDDING ORGANIZER*

Oleh:

RAHMAT RAIHAN BAIHAQI

3202016038

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

Disahkan oleh :

**Ketua Jurusan
Teknik Elektro**

Hasan, S.T., M.T.

NIP. 197108201999031003

**Koordinator Program Studi D-III
Teknik Informatika**

Mariana Syamsudin, S.T., M.T., Ph.D.

NIP. 197503142006042001

Mengetahui,

Direktur Politeknik Negeri Pontianak

Dr. H. Widodo PS, S.T., M.T.

NIP. 197504242000031001

HALAMAN PERNYATAAN
RANCANG BANGUN APLIKASI BERBASIS WEBSITE
PADA MAKNA WEDDING ORGANIZER

OLEH :

RAHMAT RAIHAN BAIHAQI

3202016038

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

Disahkan oleh :

Pembimbing :

Tri Bowo Atmojo, S.T., M.Cs.

NIP. 198407172019031010

**Telah dipertahankan di depan penguji pada tanggal 28 Agustus 2023 dan
dinyatakan memenuhi syarat sebagai Laporan Tugas Akhir.**

Penguji 1

Penguji 2

Muhammad Hasbi, S.T., M.T.

NIP. 197601112014041001

Tommi Suryanto, S.Kom., M.Kom.

NIP. 199010202019031013

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

Nama : Rahmat Raihan Baihaqi
NIM : 3202016038
Jurusan / Program Studi : Teknik Elektro/Teknik Informatika
Judul : Rancang Bangun Aplikasi Berbasis *Website* Pada Makna *Wedding Organizer*.

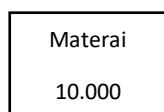
Menyatakan dengan sebenar-benarnya bahwa penulisan Tugas Akhir ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah maupun kegiatan yang tercantum sebagai bagian dari Tugas Akhir ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

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, 28 Agustus 2023

Yang membuat pernyataan,



Rahmat Raihan Baihaqi

3202016038

RIWAYAT HIDUP



Biodata Mahasiswa:

Nama : Rahmat Raihan Baihaqi
NIM : 3202016038
Tempat/Tanggal Lahir : Pontianak / 22 Agustus 2002
Agama : Islam
Alamat : Jalan Prof M Yamin Gg.Kapuas Melawi No.33
Handphone : 085246911453
Email : atzgamingvlog01@gmail.com

ABSTRAK

Makna *Wedding Organizer* merupakan usaha yang sudah berdiri sejak tahun 2020 dan menyediakan jasa *Wedding Organizer* yang berfokus pada memberikan pengalaman pernikahan yang unik dan berkesan bagi setiap pasangan. Dengan berkembangnya kebutuhan akan aksesibilitas dan kemudahan dalam perencanaan pernikahan, pembuatan *website* Makna *Wedding Organizer* menjadi langkah strategis dalam memperluas jangkauan layanan dan memperkuat identitas merek di ranah digital.

Aplikasi website ini digunakan untuk penyedia jasa *WO* di Pontianak yang diharapkan dapat membantu konsumen untuk mengetahui informasi layanan yang tersedia, serta memudahkan pihak *WO* dalam manajemen *booking* dari konsumen. Dimana nantinya *website* ini dapat menampilkan informasi yang diperlukan oleh pengguna yang membutuhkan informasi terkait dengan layanan *WO*. Selain itu aplikasi ini memudahkan pengguna dalam melakukan komunikasi melalui *live chat* dan juga melakukan *booking* yang terdapat pada fitur *website*. *Website* ini dibangun menggunakan bahasa pemrograman PHP dengan Framework CodeIgniter dan *database* MySQL pada *server side*-nya, serta menggunakan CSS dengan Framework Bootstrap untuk tampilan *website*.

Kata Kunci: *Website, Laravel, MySql, PHP, Bootstrap, Wedding Organizer*.

ABSTRACT

Makna Wedding Organizer is a business that has been established since 2020 and provides Wedding Organizer services that focus on providing a unique and memorable wedding experience for every couple. With the growing need for accessibility and convenience in wedding planning, the creation of the Makna Wedding Organizer website is a strategic step in expanding the range of services and strengthening brand identity in the digital realm.

This website application is used for WO service providers in Pontianak which is expected to help consumers find out information on available services, as well as make it easier for WO parties to manage bookings from consumers. Where later this website can display the information needed by users who need information related to WO services. In addition, this application makes it easier for users to communicate via live chat and also make bookings contained in the website features. This website was built using the PHP programming language with the CodeIgniter Framework and MySQL database on the server side, and uses CSS with the Bootstrap Framework to display the website.

Keyword: Website, Laravel, MySql, PHP, Bootstrap, Wedding Organizer

PRAKATA

Puji dan syukur penulis panjatkan kehadirat Allah *Subhanahu Wa Ta'la*, yang telah memberikan rahmat dan hidayah-Nya. Dengan selesainya Tugas Akhir ini tentunya tidak terlepas dari bimbingan dan pengarahan oleh beberapa pihak yang telah memberikan masukan-masukan kepada penulis. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua dan keluarga yang selalu memberikan nasihat dan dukungan dalam proses pengerjaan Projek Tugas Akhir hingga penyusunan laporan Tugas Akhir.
2. Bapak Dr. H. Widodo PS, S.T., M.T., selaku Direktur Politeknik Negeri Pontianak.
3. Bapak Hasan, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Pontianak.
4. Ibu Mariana Syamsudin, S.T., M.T., Ph.D., selaku Ketua Program Studi D-3 Teknik Informatika Politeknik Negeri Pontianak.
5. Bapak Fitri Wibowo, S.S.T., M.T., selaku Koordinator Tugas Akhir Politeknik Negeri Pontianak.
6. Bapak Tri Wibowo Atmojo, S.T., M.Cs., selaku dosen pembimbing yang telah memberikan masukan serta arahan mengenai Tugas Akhir.
7. Bapak Muhammad Hasbi, S.T., M.T., selaku penguji 1.
8. Bapak Tommi Suryanto, S.Kom., M.Kom., selaku penguji 2.
9. Ibu Neny Firdyanti, S.T., M.T., Selaku pembimbing akademik.
10. Seluruh dosen, teknisi, dan staf Jurusan Teknik Elektro khususnya pada Program Studi Teknik Informatika Politeknik Negeri Pontianak.
11. Teman – teman khususnya tim Makna *Wedding Organizer*, Koko *Squad*, dan Desti Muthiah yang telah memberi saran dan *support* dalam pengerjaan Tugas Akhir.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran dari pembaca dapat menjadikan motivasi penulis sehingga dapat lebih baik.

Pontianak, 28 Agustus 2023

Rahmat Raihan Baihaqi

NIM. 3202016038

DAFTAR ISI

| | |
|--|------|
| HALAMAN PENGESAHAN..... | i |
| HALAMAN PERNYATAAN..... | ii |
| HALAMAN PERNYATAAN ORISINALITAS | iii |
| RIWAYAT HIDUP..... | iv |
| ABSTRAK..... | v |
| <i>ABSTRACT</i> | vi |
| PRAKATA..... | vii |
| DAFTAR ISI..... | viii |
| DAFTAR GAMBAR..... | xi |
| DAFTAR TABEL..... | xii |
| DAFTAR LAMPIRAN | xiii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah..... | 2 |
| 1.4 Tujuan Penelitian..... | 2 |
| 1.5 Manfaat Penelitian..... | 2 |
| 1.6 Metodologi | 3 |
| 1.6.1 Metode Pengumpulan Data | 3 |
| 1.6.2. Metode Pengembangan Aplikasi..... | 4 |
| 1.7. Sistematika Penulisan..... | 5 |
| BAB II DASAR TEORI..... | 7 |
| 2.1. Tinjauan Pustaka | 7 |
| 2.2. Dasar Teori | 9 |
| 2.2.1. Wedding Organizer..... | 9 |
| 2.2.2. Aplikasi | 9 |
| 2.2.3. Website | 10 |
| 2.2.4. HTML | 10 |
| 2.2.5. PHP..... | 10 |
| 2.2.6. MySql..... | 11 |
| 2.2.7. XAMPP | 11 |

| | |
|---|-----------|
| 2.2.8. Framework Laravel | 11 |
| BAB III PERANCANGAN SISTEM..... | 13 |
| 3.2.1. Kebutuhan Perangkat Keras | 13 |
| 3.2.2. Kebutuhan Perangkat Lunak | 13 |
| 3.3 Perancangan Sistem..... | 13 |
| 3.3.1. Communication | 13 |
| 3.3.2. Use Case Diagram..... | 14 |
| 3.3.3. Definisi Aktor..... | 15 |
| 3.3.4. Definisi <i>Use Case</i> | 16 |
| 3.3.5. Skenario <i>Use Case</i> | 19 |
| 3.4. Perancangan Basis Data | 27 |
| 3.4.1. Relasi Tabel | 27 |
| 3.4.2. Struktur Tabel..... | 27 |
| 3.5. Perancangan Antar Muka | 32 |
| 3.5.1. Halaman Login | 32 |
| 3.5.2. Halaman registrasi | 32 |
| 3.5.3. Halaman Informasi Paket..... | 33 |
| 3.5.4. Halaman Jadwal | 33 |
| 3.5.5. Halaman konfirmasi <i>booking</i> | 34 |
| 3.5.6. Halaman <i>booking</i> | 34 |
| 3.1. Halaman Client review | 35 |
| BAB IV HASIL DAN PEMBAHASAN..... | 36 |
| 4.1. Implementasi Halaman Antar Muka | 36 |
| 4.1.1. Tampilan Halaman Beranda | 36 |
| 4.1.2. <i>Source Code</i> Tampilan Halaman Beranda..... | 38 |
| 4.1.3. Tampilan Halaman Paket <i>Wedding</i> | 39 |
| 4.1.4. <i>Source Code</i> Tampilan Halaman Paket <i>Wedding</i> | 40 |
| 4.1.5. Tampilan Halaman <i>About</i> | 40 |
| 4.1.6. <i>Source Code</i> Halaman <i>About</i> | 41 |
| 4.1.7. Tampilan Halaman <i>Client Review</i> | 42 |
| 4.1.8. <i>Source Code</i> Halaman Client Review | 43 |
| 4.1.9. Tampilan Halaman Jadwal | 44 |
| 4.1.10. <i>Source Code</i> Tampilan Halaman Jadwal | 45 |
| 4.1.11. Tampilan Halaman Detail Paket..... | 46 |

| | |
|--|----|
| 4.1.12. <i>Source Code</i> Tampilan Halaman Detail Paket | 48 |
| 4.1.13. Tampilan Halaman Keranjang..... | 48 |
| 4.1.14. Source Code Halaman Keranjang | 49 |
| 4.1.15. Tampilan Halaman Pembayaran..... | 50 |
| 4.1.16. <i>Source Code</i> Halaman Pembayaran | 51 |
| 4.1.17. Halaman <i>Login</i> | 53 |
| 4.1.18. Source Code Tampilan Halaman Login | 53 |
| 4.1.19. Tampilan Halaman Registrasi | 54 |
| 4.1.20. <i>Source Code</i> Halaman Registrasi | 55 |
| 4.1.21. Tampilan Halaman <i>Login Admin</i> | 56 |
| 4.1.22. Source Code Halaman Login Admin | 57 |
| 4.1.23. Tampilan Halaman <i>Dashboard Admin</i> | 58 |
| 4.1.24. Source Code Halaman Dashboard Admin..... | 59 |
| 4.1.25. Tampilan Halaman Kelola <i>Web Profile</i> | 60 |
| 4.1.26. Source Code Halaman Kelola Web Profile | 60 |
| 4.1.27. Tampilan Halaman Kelola Paket <i>Wedding</i> | 61 |
| 4.1.28. <i>Source Code</i> Halaman Kelola Paket <i>Wedding</i> | 64 |
| 4.1.29. Tampilan Halaman Kelola Pemesanan | 66 |
| 4.1.30. <i>Source Code</i> Halaman Kelola Pemesanan | 68 |
| 4.1.31. <i>Source Code</i> Halaman Kelola Laporan Pemesanan | 72 |
| 4.1.32. Tampilan Halaman Kelola Jadwal..... | 74 |
| 4.1.33. <i>Source Code</i> Halaman Kelola Jadwal | 75 |
| 4.1.34. Tampilan Halaman Kelola Manajemen Akun | 77 |
| 4.1.35. <i>Source Code</i> Halaman Kelola Manajemen Akun..... | 80 |
| 4.2. Pengujian Sistem Contruction Of Prototype | 84 |
| BAB V KESIMPULAN DAN SARAN..... | 87 |
| 5.1. Kesimpulan..... | 87 |
| 5.2. Saran..... | 87 |
| DAFTAR PUSTAKA | 88 |
| LAMPIRAN..... | 90 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1. 1. Metode Prototype | 4 |
| Gambar 3.1. Use case diagram admin | 14 |
| Gambar 3. 2. Use case diagram untuk pemesanan | 15 |
| Gambar 3. 3. Relasi Tabel | 27 |
| Gambar 3. 4. Tampilan rancangan halaman login | 32 |
| Gambar 3. 5. Tampilan rancangan halaman registrasi | 33 |
| Gambar 3. 6. Tampilan rancangan halaman informasi paket | 33 |
| Gambar 3. 7. Tampilan rancangan halaman jadwal | 34 |
| Gambar 3. 8. Tampilan rancangan halaman konfirmasi booking | 34 |
| Gambar 3. 9. Tampilan rancangan halaman booking | 35 |
| Gambar 3. 10. Tampilan rancangan halaman client review | 35 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1.Tinjauan Pustaka..... | 9 |
| Tabel 3. 1. Definisi Aktor | 15 |
| Tabel 3.2. Definisi use case diagram admin | 16 |
| Tabel 3. 3. Definisi use case diagram pemesanan | 17 |
| Tabel 3. 4. SK 01 Tampilan Beranda..... | 19 |
| Tabel 3. 5. SK 02 Halaman Paket Wedding | 19 |
| Tabel 3. 6. SK 03 Halaman About/tentang | 20 |
| Tabel 3. 7. SK 04 Halaman Client Review | 20 |
| Tabel 3. 8. SK 05 Halaman Keranjang | 21 |
| Tabel 3. 9. SK 06 Halaman Login Pemesanan | 21 |
| Tabel 3. 10. SK 07 Halaman Jadwal | 22 |
| Tabel 3.11. SK 08 Halaman Pembayaran | 22 |
| Tabel 3. 12. SK 09 Halaman Login Admin | 23 |
| Tabel 3. 13. SK 010 Halaman Dashboard admin | 23 |
| Tabel 3. 14. SK 011 Halaman kelola web profile..... | 24 |
| Tabel 3. 15. 012 Halaman kelola paket wedding | 24 |
| Tabel 3. 16. SK 013 Halaman kelola pemesanan | 25 |
| Tabel 3. 17. SK 014 Halaman kelola laporan pemesanan | 25 |
| Tabel 3. 18. SK 015 Halaman kelola jadwal | 26 |
| Tabel 3. 19. SK 016 Halaman kelola akun | 26 |
| Tabel 3. 20. Struktur tabel users | 28 |
| Tabel 3. 21. Struktur tabel admin | 28 |
| Tabel 3. 22. Struktur tabel paket wedding | 29 |
| Tabel 3. 23. Struktur tabel events | 29 |
| Tabel 3. 24. Struktur tabel pemesanan | 30 |
| Tabel 3. 25. Struktur tabel keranjang | 30 |
| Tabel 3.26. Struktur tabel paket wedding foto | 31 |
| Tabel 3. 27. Struktur tabel web | 31 |
| Tabel 4. 1. Pengujian halaman admin | 85 |
| Tabel 4. 2. Pengujian Halaman user/pemesanan | 85 |

DAFTAR LAMPIRAN

| | |
|---|----|
| Lampiran 1. Foto Bersama Tim Makna Wedding Organizer untuk Melakukan Observasi dan Wawancara..... | 90 |
| Lampiran 2. Contoh dari Manajemen Data Pemesanan Menggunakan Aplikasi Note dari Smarthphone | 90 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Website atau situs *web* adalah halaman informasi yang disampaikan oleh pemilik situs (admin) kepada pengunjung (*user*). *Web* adalah sebuah sistem yang saling terkait dalam sebuah dokumen yang berformat *hypertext* yang berisi beragam informasi, baik tulisan, gambar, suara, video dan informasi multimedia lainnya dan dapat diakses melalui sebuah perangkat yang disebut *web browser*. Untuk menterjemahkan dokumen dalam bentuk *hypertext* ke dalam bentuk dokumen yang bisa dipahami, maka *web browser* melalui *web client* akan membaca halaman *web* yang tersimpan di sebuah *web server* melalui protokol yang biasa disebut HTTP (*Hypertext Transfer Protocol*) [1].

Makna *wedding organizer* adalah sebuah usaha yang bergerak dibidang jasa *wedding organizer* yang telah berdiri sejak tahun 2021 yang belokasi di jalan Putri Candramidi, Gg. Sudihati, kota Pontianak yang bekerja sama dengan *vendor* dekorasi, dokumentasi, penata rias, *entertainment*, *master of ceremony*. *Wedding Organizer* (WO) adalah suatu jasa yang berfungsi secara pribadi membantu calon pengantin dan keluarga dalam perencanaan dan supervisi pelaksanaan rangkaian acara pesta pernikahan sesuai dengan jadwal dan *budget* yang telah ditetapkan [2].

Sistem pemesanan pada Makna *wedding organizer* saat ini, yaitu dengan cara menghubungi *wedding planner* melalui telepon, *whatsapp*, ataupun datang secara langsung. Kemudian pemesan menentukan paket mana yang akan dipilih, setelah menentukan paket pemesan diminta untuk memenuhi syarat pemesanan dan membuat perjanjian pemesanan. Dilihat dari sistem penyewaan saat ini yang masih dilakukan secara manual dan belum adanya sistem aplikasi penyewaan secara *online* mengakibatkan pihak dari Makna *wedding organizer* kewalahan dalam memanajemen proses pemesanan, kesulitan dalam mencatat jadwal, kesulitan dalam mencari data laporan pemesanan dan data diri pemesan yang di data didalam aplikasi *note smartphone*.

Berdasarkan penjelasan yang sudah dipaparkan sebelumnya, maka penulis mengangkat judul “Rancang Bangun Aplikasi *Wedding Organizer* Berbasis

Website”. Diharapkan Dengan adanya *website* ini dapat mengatasi permasalahan yang terjadi pada Makna *wedding organizer* maupun dari pihak pemesan. Selain itu juga proses pengelolaan data diharapkan dapat dilakukan dengan lebih efisien dibandingkan dengan proses yang ada sebelumnya.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang maka perumusan masalah yang ingin diangkat adalah bagaimana merancang dan membangun *website wedding organizer* pada Makna *wedding organizer*.

1.3 Batasan Masalah

Batasan masalah dalam membuat aplikasi ini adalah sebagai berikut :

1. Aplikasi dibangun berbasis *website*.
2. Admin dapat mengelola data informasi, seperti mengubah informasi paket, informasi jadwal, informasi data *booking*, dan mengelola data pemesan, menambahkan data *client review*.
3. Aplikasi dibangun dengan menggunakan Bahasa pemrograman PHP, CSS, Javascript, dan HTML.
4. *Framework* yang digunakan untuk membangun aplikasi adalah Laravel versi 9 *dan Bootstrap* versi 5.
5. Perangkat lunak *web server* dan basis data yang digunakan adalah XAMPP versi 3.3.0.

1.4 Tujuan Penelitian

Berdasarkan permasalahan yang telah dirumuskan di atas, tujuan pembuatan tugas akhir ini adalah menghasilkan sebuah *website* Makna *wedding organizer* yang berisi tentang informasi dan layanan yang dapat menjadi sarana dalam penyampaian informasi kepada calon pengantin.

1.5 Manfaat Penelitian

Adapun manfaat-manfaat dari hasil penelitian yang penulis lakukan adalah:

1. Bagi mahasiswa

Dapat melatih mahasiswa dalam menerapkan pengetahuan serta ilmu-ilmu mengenai pembangunan dan perancangan aplikasi atau sistem

informasi yang telah diperoleh selama menjalani perkuliahan serta dapat memahami bahasa pemrograman dan konsep-konsep basis data.

2. Bagi Makna *wedding organizer*

Menyederhanakan dalam penyajian informasi pemesanan dan pengelolaan data pemesan agar efisien.

3. Bagi penyewa

Membantu pemesan dalam memesan dan mendapatkan informasi mengenai paket *wedding organizer* yang tersedia pada Makna *wedding organizer*.

1.6 Metodologi

Metodologi penelitian bertujuan untuk mendapatkan gambaran yang jelas tentang hal-hal yang diperlukan dalam pembangunan aplikasi. Adapun metode yang digunakan dalam penyusunan tugas akhir adalah sebagai berikut :

1.6.1 Metode Pengumpulan Data

Teknik pengumpulan data yang dilakukan penulis dalam penyusunan proposal tugas akhir ini, yaitu sebagai berikut [3]:

1. Metode Observasi

Observasi merupakan pengamatan yang dilakukan oleh penulis secara langsung pada Makna *wedding organizer* yang menjadi fokus penelitiannya dalam rangka memperoleh data penelitian.

2. Metode Wawancara

Wawancara merupakan suatu teknik pengumpulan data yang dilakukan dengan cara berkomunikasi langsung kepada pihak Makna *wedding organizer*.

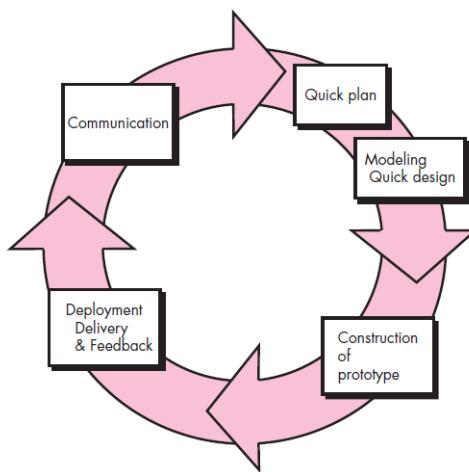
3. Metode Literatur

Literatur merupakan pengumpulan data yang diperlukan untuk dijadikan tinjauan atau acuan dalam proses pembuatan aplikasi *web*. Data yang dikumpulkan bersumber dari buku, jurnal, artikel, dan

sumber-sumber lain dari *browsing* internet serta proposal Tugas Akhir senior terdahulu.

1.6.2. Metode Pengembangan Aplikasi

Metode pengembangan website Makna *Wedding Organizer* adalah menggunakan metode *prototype*. Metode *prototype* melibatkan pengguna sehingga dapat mengidentifikasi kebutuhan pengguna secara langsung dan memastikan bahwa perangkat lunak yang dibangun memenuhi tujuan dari pengguna. Tahapan metode *prototype* disajikan pada Gambar 1.



Gambar 1. 1. Metode Prototype

Tahapan dari metode *prototype* terdiri dari 5 tahapan, yaitu [4]:

1. *Communication*

Communication adalah tahap komunikasi dan pengumpuan data awal, yaitu analisis terhadap kebutuhan pengguna. Melakukan identifikasi masalah terhadap objek penelitian dan mengidentifikasikan proses pemesanan yang lama.

2. *Quick Plan*

Setelah melakukan komunikasi, tahap selanjutnya adalah melakukan desain berupa *Unified Modeling Language* (UML), yang dijabarkan melalui *use case diagram*.

‘

3. *Modeling Quick Design*

Modeling Quick Design dilakukan dengan pembuatan desain *user interface* (UI) secara *storyboard / mockup* untuk dijadikan bahan dalam pembuatan aplikasi.

4. *Construction of Prototype*

Pembentukan *prototype*, yaitu pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan. membangun prototipe yang berfokus pada penyajian. Misalkan membuat input dan output hasil system. Sementara hanya prototype saja dulu selanjutnya akan ada tindak lanjut yang harus di kerjakan.

5. *Deployment Delivery & Feedback*

Setelah hasil rancangan awal atau *prototype* dari aplikasi pemesanan dibuat, selanjutkan melakukan analisis terhadap aplikasi untuk mendapatkan *feedback* dari pengguna tentang fitur dan tampilan dari *prototype* aplikasi. Hal ini dilakukan agar aplikasi yang dibuat sudah sesuai dengan kebutuhan pengguna sehingga tujuan dari pembuatan aplikasi terpenuhi.

1.7. Sistematika Penulisan

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

BAB 1 PENDAHULUAN

Pada BAB 1 Pendahuluan meliputi penjelasan terkait latar belakang, rumusan masalah, batasan masalah, tujuan Tugas Akhir, Manfaat Tugas Akhir, Metodologi dan Sistematika Tugas Akhir.

BAB 2 DASAR TEORI

Pada BAB 2 Dasar teori terdapat pembahasan mengenai landasan teori dan teori dasar yang terkait dengan permasalahan yang dibahas dalam Tugas Akhir ini.

BAB 3 PERANCANGAN SISTEM

Pada BAB 3 Perancangan sistem terdapat penjelasan terkait gambaran sistem lama, gambaran sistem baru. Serta meliputi penjelasan tentang perancangan proses, basis data dan rancangan antar muka pada *website Makna Wedding Organizer*.

BAB 4 HASIL DAN PEMBAHASAN

Pada BAB 4 Hasil dan Pembahasan meliputi tentang penjelasan mengenai implementasi antar muka pengguna data pada *website Makna Wedding Organizer*.

BAB 5 PENUTUP

Penutup berisi penjelasan tentang kesimpulan dan saran bagi pembaca yang dapat digunakan sebagai bahan masukkan dari Tugas Akhir yang telah dibuat.

BAB II

DASAR TEORI

2.1. Tinjauan Pustaka

| NO | Nama | Judul | Peneliti sebelumnya | | Peneliti | |
|----|------------------|---|--|--|--|---|
| | | | Fitur/Menu | Software | Fitur/Menu | Software |
| 1 | Irkantini (2011) | Perancangan <i>E-Commerce</i> di Wulan <i>Wedding Organizer</i> | <ul style="list-style-type: none"> • Input data secara online • Halaman dashboard • Halaman login • Halaman register • Halaman menu • Halaman transaksi • Pencarian menu • Halaman admin | PHP, HTML, MySql, Framework Codeigniter | <ul style="list-style-type: none"> • Input data secara online • Halaman dashboard • Halaman login • Halaman register • Halaman menu • Halaman transaksi • Pencarian menu • Halaman admin | PHP,HTML, MySql, Framework Laravel |

| | | | | | | |
|---|---|--|---|--|--|--|
| 2 | Efmi Maiyana, Mira Susanti, Fitri Rahayu (2019) | Penyewaan Online <i>Wedding Organizer</i> Berbasis Web | <ul style="list-style-type: none"> • Halaman Login • Halaman Register • Halaman Dashboard • Halaman Menu • Halaman Detail Menu • Halaman Admin | PHP, MySQL, HTML | <ul style="list-style-type: none"> • Halaman Login • Halaman Register • Halaman Dashboard Halaman Menu • Halaman Detail Menu • Halaman Admin | PHP, MySQL, HTML, <i>Framework Laravel</i> |
| 3 | Dea Fitri Ananda (2021) | Rancang Bangun Aplikasi Penyedia Jasa <i>Wedding Organizer</i> Anugrah Kreasi Pontianak Berbasis Website | <ul style="list-style-type: none"> • Pembuatan surat perjanjian • <i>live chat</i> • <i>Sort</i> • Halaman Login • Halaman Register • Halaman <i>Home</i> • Halaman Kategori • Halaman Kalender • Halaman Galeri | PHP, MySQL, HTML, <i>Framework Codeigniter</i> , | Pembuatan surat perjanjian, sortr | PHP, MySQL, HTML, <i>Framework Laravel</i> . |

| | | | | | |
|--|--|-------------------------------|--|--|--|
| | | • Halaman <i>Check-Out</i> | | | |
|--|--|-------------------------------|--|--|--|

Tabel 2.1.Tinjauan Pustaka

2.2. Dasar Teori

2.2.1. Wedding Organizer

Wedding Organizer adalah suatu jasa khusus yang secara pribadi membantu calon pengantin dan keluarga dalam perencanaan dan supervise pelaksanaan rangkaian acara pesta pernikahan sesuai dengan jadwal yang telah ditetapkan, antara lain [5]:

1. Memberikan informasi mengenai berbagai macam hal yang behubungan dengan acara pernikahan.
2. Membantu merumuskan konsep pernikahan.
3. Fasilitas, negosiasi dan koordinasi dengan pihak gedung/hotel dan supplier/vendor seperti : catering , dekorasi, fotografer, perias, grup music.
4. Membantu memilih jenis makanan, kuantitas dan food tasting.
5. Menghubungi setiap rekanan untuk follow up yang diperlukan.
6. Mengikuti rapat koordinasi dengan keluarga, juga memimpin rapat teknis dengan para rekanan.
7. Mengkoordinasikan acara dan seluruh rekanan pada hari H, memonitoring proses dan detail acara serta membantu mengingatkan calon pengantin atau anggot akeluaraga yang berkepentingan mengenai hal -hal yang berkaitan dengan kelancaran penyelenggaraan acara.

2.2.2. Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian

aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpade sesuai dengan kemampuan yang dimilikinya. Aplikasi merupakan suatu perangkat komputer yang siap bagi *user* [1].

2.2.3. Website

Web atau situs adalah halaman informasi yang disampaikan oleh pemilik situs (admin) kepada pengunjung (*user*). Disetiap web, halaman disebut front page atau home page. Web adalah sebuah sistem yang saling terkait dalam sebuah dokumen yang berformat hypertext yang berisi beragam informasi, baik tulisan, gambar, suara, video dan informasi multimedia lainnya dan dapat diakses melalui sebuah perangkat yang disebut web browser. Untuk menterjemahkan dokumen dalam bentuk hypertext ke dalam bentuk dokumen yang bisa dipahami, maka web browser melalui web client akan membaca halaman web yang tersimpan di sebuah web server melalui protokol yang biasa disebut http (*Hypertext Transfer Protocol*) [1].

2.2.4. HTML

HTML (*HyperText Markup Language*) merupakan bahasa standard yang digunakan untuk membuat dokumen web. Yang dapat dilakukan dengan HTML adalah [1]:

1. Mengontrol tampilan dari halaman web dan *content*-nya.
2. Membuat online form yang bisa digunakan untuk menangani pendaftaran dan transaksi secara online.

Menambahkan objek-objek seperti gambar, audio, video, animasi dan juga java applet dalam dokumen HTML.

2.2.5. PHP

PHP (*Hypertext Preprocessor*) dikenal sebagai sebuah bahasa scripting yang menyatu dengan tag-tag HTML yang diekseskuasi di server dan membuat halaman web yang dinamis. Bahasa pemrograman PHP dapat dipakai di hampir semua web server yang ada di pasaran seperti Apache dan Microsoft IIS. PHP dapat dijalankan pada berbagai sistem operasi seperti *Linux*, *Windows*, dan *FreeBSD*. PHP bersifat open source dan gratis [6]. Pada pengembangan aplikasi pengarsipan dokumen paspor menggunakan PHP versi 7.

2.2.6. MySql

MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL). MySQL adalah DBMS (*Databae Management Sistem*) yang *open source* dengan dua bentuk lisensi yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas [7]. Beberapa alasan menggunakan MySQL untuk membuat basis data *database* yaitu :

1. MySQL menyediakan sistem basis data berkecepatan tinggi yang sempurna untuk proyek-proyek kecil hingga menengah.
2. Dapat menangani *database* dengan skla besar yaitu dengan jumlah *record* lebih dari 50 juta.
3. *Database* MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.
4. MySQL dapat digunakan untuk mengembangkan aplikasi berbasis dekstop maupun berbasis web.
5. MySQL dapat berjalan stabil di berbagai sistem operasi seperti windows, Linux, Unix.

2.2.7. XAMPP

XAMPP adalah sebuah aplikasi open source terkait pengelolaan *server* yang dikembangkan oleh Apache Friends. XAMPP bersifat *open source* dan dapat dijalankan diberbagai sistem operasi. XAMPP sendiri terdiri dari Apache, MariaDB, PHP dan Perl [8].

2.2.8. Framework Laravel

Laravel adalah sebuah *framework web* berbasis PHP yang *open source* dan tidak berbayar diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC (*Model-View-Controller*). *Model* pada Laravel merupakan representasi struktur data yang memiliki fungsi-fungsi yang dapat digunakan untuk mngelola basis data. *View* pada Laravel adalah bagian yang mengatur antarmuka aplikasi agar pengguna dapat berinteraksi dengan aplikasi yang dibuat. *Controller* pada Laravel merupakan jembatan antara *Model* dan *View* dengan mengirimkan permintaan dari *View* dan menerima respons dari *Model* [7]. Versi Laravel yang digunakan adalah Laravel versi 9. *Framework* Laravel memiliki keunggulan yang menjadikannya lebih baik

dari pada *framework* lainnya, yaitu *performance* lebih cepat, *reload* data lebih stabil, memiliki keamanan data, menggunakan fitur canggih seperti *blade* menggunakan konsep HMVC (*Hierarchical Model View Controller*), tersedianya *library-library* yang sudah siap untuk digunakan dan adanya fitur pengelolaan *migrations* untuk pembuatan skema *table* pada *database* [9].

BAB III

PERANCANGAN SISTEM

3.1 Gambaran Umum

Website penyedia Jasa Wedding Organizer pada Makna Wedding Organizer adalah sebuah sistem informasi berbasis internet yang dibuat untuk mempermudah konsumen atau masyarakat dalam mencari informasi mengenai penyedia jasa *wedding organizer* pada *website* ini.

3.2 Analisis Kebutuhan

Analisis kebutuhan cukup penting dalam proses penyelesaian tugas akhir ini karena untuk mengetahui semua kebutuhan apa saja yang diperlukan dalam pembuatan *website* penyedia jasa *wedding organizer* pada Makna *Wedding Organizer*. Adapun analisis kebutuhan dari *website* ini terdiri dari:

3.2.1. Kebutuhan Perangkat Keras

Perangkat keras dibutuhkan agar *website* yang dibangun dapat berjalan dengan lancar. Adapun perangkat tersebut adalah laptop Lenovo yang menggunakan prosessor Intel Core i7 12700H dan RAM sebesar 16GB.

3.2.2. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan adalah sistem operasi *Windows 11*, *Visual Studio Code*, *XAMPP*, dan *Browser Brave*.

3.3 Perancangan Sistem

Percancangan sistem *website* Makna *Wedding Organizer* pada tugas akhir ini adalah sebagai berikut :

3.3.1. Communication

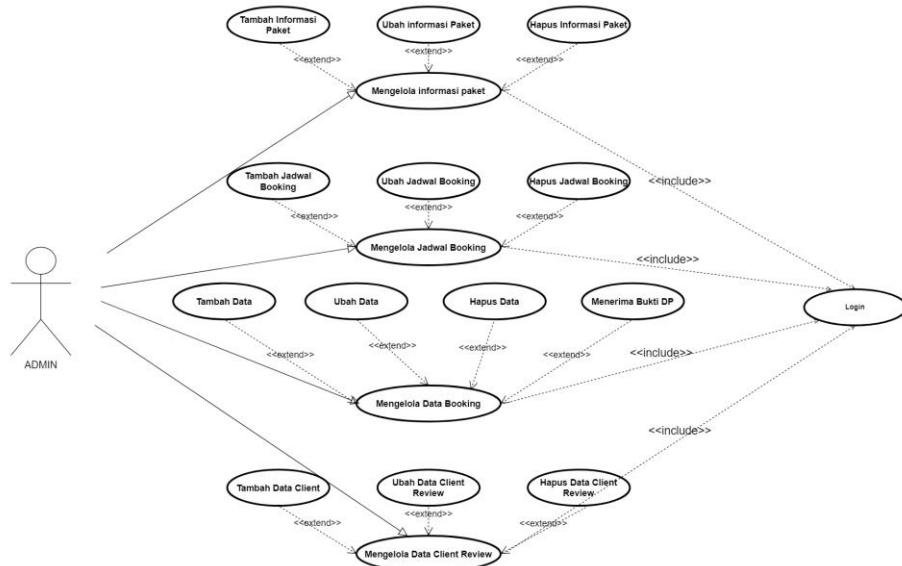
Communication sangat penting untuk penelitian ini ditahap ini penulis melakukan observasi dan wawancara secara langsung dengan pihak dari Makna *wedding organizer*, penulis menanyakan tentang kekurangan selama ini didalam memanajemen data dan menanyakan fitur apa saja yang harus ada didalam *website* , dengan observasi dan wawancara yang sudah dilakukan maka menghasilkan beberapa kesimpulan , yaitu :

1. *Website* harus menampilkan beberapa menu seperti *home*, informasi paket, jadwal, *booking*, dan *client review*.
2. *Website* terlihat menarik dan interaktif.

3.3.2. Use Case Diagram

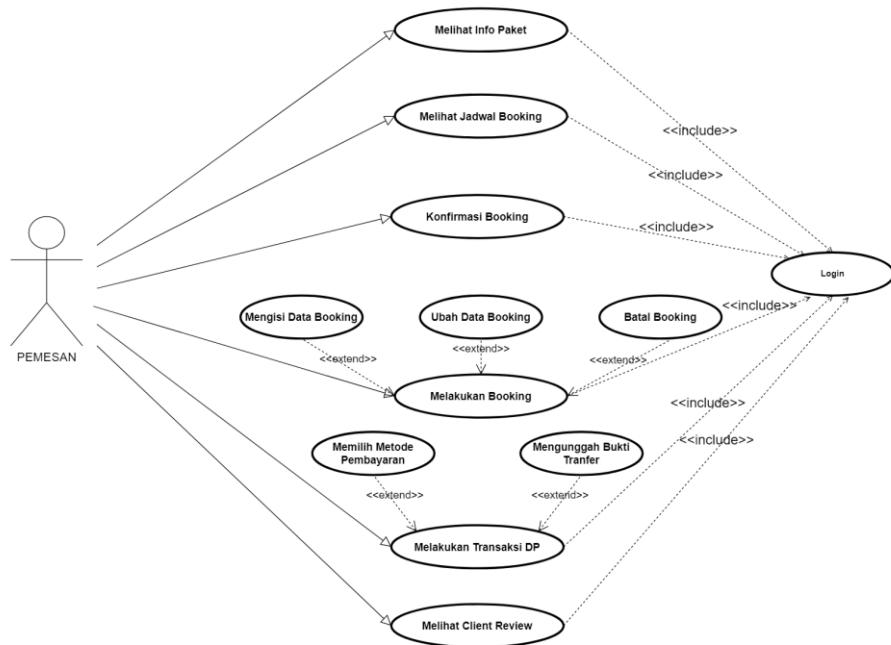
Use case diagram menggambarkan interaksi antara *actor* dan komponen yang ada pada sistem yang akan dibangun. Pembuatan *use case* juga penerapan dari metodologi *prototype* yang digunakan dalam pengembangan aplikasi ini, yaitu metodologi *Quick Plan* yang dimana di metode ini penulis membuat desain berupa *Unified Modelling Language* (UML) yang disini penulis menggunakan *use case* sebagai UML nya.

Berikut *use case diagram* untuk admin disajikan pada gambar 3.1



Gambar 3.1. Use case diagram admin

Berikut adalah *use case diagram* untuk pemesan disajikan pada gambar 3.2



Gambar 3. 2. Use case diagram untuk pemesanan

3.3.3. Definisi Aktor

Pada *use case* diagram diatas dapat diketahui bahwa sistem ini memiliki 2 aktor yaitu admin dan pemesan. Admin yaitu user yang memiliki hak akses penuh untuk mengelola data-data pada *website* dengan melakukan login terlebih dahulu. Pemesan yaitu dapat memilih atau booking layanan jasa *wedding organizer* serta melihat mengenai apa itu Makna *wedding organizer*; *client review*, paket wedding tanpa perlu melakukan login terlebih dahulu. Untuk melakukan pemesanan dan melihat jadwal pemesan harus melakukan login terlebih dahulu.

Tabel 3. 1. Definisi Aktor

| No. | Nama | Deskripsi |
|-----|-------|--|
| 1 | Admin | Kelola data layanan jasa, <i>web profile</i> , paket wedding, mengelola data pemesanan, mengelola data laporan pemesanan, mengatur jadwal, dan memanajemen akun yang terdaftar di web. |

| | | |
|---|---------|--|
| 2 | Pemesan | Konsumen harus melakukan registrasi terlebih dahulu untuk dapat melakukan pemesanan paket wedding dan melihat jadwal yang tersedia. Pemesan dapat melihat paket yang tersedia , beranda, tentang, dan <i>client review</i> |
|---|---------|--|

3.3.4. Definisi *Use Case*

Definisi *use case* bertujuan untuk mendeskripsikan atau menjelaskan dari *use case* diagram. *Use case* diagram juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan juga mempresentasikan sebuah interaksi aktor dengan sistem yang ada. Dengan demikian, *use case* dapat dipresentasikan dengan urutan yang sederhana. Berikut adalah penjelasannya :

Tabel 3.2. Definisi use case diagram admin

| No. | Hak Akses | Keterangan |
|-----|---------------------------|---|
| 1 | Login Admin | Admin harus melakukan login terlebih dahulu untuk melakukan verifikasi <i>username</i> dan <i>password</i> agar admin dapat melakukan pengelolaan data |
| 2 | Mengelola informasi paket | Setelah berhasil login admin dapat mengelola informasi paket yang terdiri dari tambah informasi paket, ubah informasi paket, hapus informasi paket |
| 3 | Mengelola jadwal booking | Admin dapat mengelola jadwal booking sesuai pemesan yang sudah memenuhi persyaratan, didalam pengelolaan jadwal booking admin juga dapat tambah jadwal booking, |

| | | |
|---|----------------------------------|--|
| | | ubah jadwal booking, dan hapus jadwal booking |
| 4 | Mengelola data pemesana | Admin dapat mengelola data booking dari pemesan yang sudah memenuhi syarat, admin juga dapat melakukan tambah data booking, ubah data booking, hapus data booking, dan menerima bukti transferan DP dari pemesan |
| 5 | Mengelola data client review | Admin dapat juga mengelola data client review untuk ditampilkan ke <i>website</i> , admin dapat menambah data <i>review</i> , ubah data <i>review</i> , hapus data <i>review</i> . |
| 6 | Mengelola web profile | Admin dapat mengelola data web profile seperti sosial media, logo, alamat, deskripsi |
| 7 | Mengelola data laporan pemesanan | Admin dapat mengelola data laporan pemesanan seperti nama pemesan, nama paket, jumlah paket, tempat acara, tanggal acara, dan total harga. |
| 8 | Mengelola data akun | Admin dapat mengelola akun yang terdaftar di website, admin dapat mengelola data akun admin maupun pemesan. |

Tabel 3. 3. Definisi *use case* diagram pemesanan

| No. | Hak akses | Keterangan |
|-----|-----------|------------|
| | | |

| | | |
|---|------------------------------|--|
| 1 | Login user/pemesan | Verifikasi <i>username</i> dan <i>password</i> untuk pemesan agar dapat melakukan pemesanan dan melihat jadwal. |
| 2 | Melihat informasi paket | Setelah melakukan login pemesan dapat melihat informasi paket yang sudah admin sediakan |
| 3 | Melihat jadwal booking | Pemesan dapat melihat jadwal untuk mencocokkan jadwal hari agar tidak bertabrakan sama pemesan sebelumnya |
| 4 | Konfirmasi booking | Setelah pemesan mendapatkan jadwal yang sesuai pemesan dapat melakukan konfirmasi booking |
| 5 | Melakukan booking | Setelah melakukan konfirmasi pemesan dapat melakukan booking, pemesan dapat mengisi data booking, ubah data booking, batal booking |
| 6 | Melakukan transaksi DP | Setelah mengisi data booking pemesan dapat melakukan transaksi DP yang menjadi syarat pemesanan, pemesan dapat memilih metode pembayaran, dan mengunggah bukti transfer yang sudah disiapkan di form |
| 7 | Melihat <i>client review</i> | Pemesan dapat juga melihat <i>client review</i> sebelumnya yang sudah menggunakan jasa dari Makna <i>wedding organizer</i> , untuk menambah rasa percaya menggunakan jasa dari |

| | | |
|---|-------------------|---|
| | | Makna <i>wedding organizer</i> : |
| 8 | Melihat Keranjang | Pemesan dapat melihat keranjang yang berisi ringkasan pesan dari paket yang sudah di pilih dan masuk ke keranjang |

3.3.5. Skenario *Use Case*

Skenario *Use Case* bertujuan untuk mendeskripsikan atau menjelaskan *Use Case* diagram. *Use Case* dideskripsikan secara tekstual dalam bentuk *use case* scenario untuk menjelaskan interaksi yang terjadi antara aktor dengan sistem. Berikut adalah penjelasannya :

Tabel 3. 4. SK 01 Tampilan Beranda

| | |
|--------------|---|
| No. Skenario | SK 01 |
| No/Nama UC | UC-01/Beranda |
| Ringkasan | Pemesan dapat melihat halaman beranda |
| Aktor | Pemesan |
| Kondisi | Tampil halaman beranda |
| Deskripsi | User dapat melihat halaman beranda, dan galeri foto |
| Alternatif | - |

Tabel 3. 5. SK 02 Halaman Paket Wedding

| | |
|--------------|---------------------|
| No. Skenario | SK 02 |
| No/Nama UC | UC-02/Paket Wedding |

| | |
|------------|---|
| Ringkasan | Pemesan dapat melihat halaman paket wedding, memasukkan ke keranjang dan detail paket |
| Aktor | Pemesan |
| Kondisi | Tampil halaman paket wedding, Tampil detail paket |
| Deskripsi | User dapat melihat halaman paket wedding, dan detail paket |
| Alternatif | - |

Tabel 3. 6. SK 03 Halaman About/tentang

| | |
|--------------|---|
| No. Skenario | SK 01 |
| No/Nama UC | UC-03/Tentang |
| Ringkasan | Pemesan dapat melihat halaman tentang |
| Aktor | Pemesan |
| Kondisi | Tampil halaman tentang |
| Deskripsi | User dapat melihat halaman tentang, deskripsi mengenai Makna <i>wedding organizer</i> |
| Alternatif | - |

Tabel 3. 7. SK 04 Halaman Client Review

| | |
|--------------|-----------------------------|
| No. Skenario | SK 04 |
| No/Nama UC | UC-04/ <i>Client Review</i> |

| | |
|------------|--|
| Ringkasan | Pemesan dapat melihat halaman <i>client review</i> |
| Aktor | Pemesan |
| Kondisi | Tampil halaman <i>client review</i> |
| Deskripsi | User dapat melihat halaman <i>client review</i> yang berisi video. |
| Alternatif | - |

Tabel 3. 8. SK 05 Halaman Keranjang

| | |
|--------------|---|
| No. Skenario | SK 05 |
| No/Nama UC | UC-05/Keranjang |
| Ringkasan | Pemesan dapat melihat halaman keranjang dan melihat ringkasan pesanan |
| Aktor | Pemesan |
| Kondisi | Tampil halaman keranjang |
| Deskripsi | User dapat melihat halaman keranjang dan melihat ringkasan pesanan. |
| Alternatif | - |

Tabel 3. 9. SK 06 Halaman Login Pemesanan

| | |
|--------------|-------------|
| No. Skenario | SK 06 |
| No/Nama UC | UC-06/Login |

| | |
|------------|--|
| Ringkasan | Pemesan dapat melakukan login |
| Aktor | Pemesan |
| Kondisi | Tampil halaman login |
| Deskripsi | User dapat melakukan login dengan <i>username</i> dan <i>password</i> untuk melakukan pemesanan dan melihat jadwal |
| Alternatif | - |

Tabel 3. 10. SK 07 Halaman Jadwal

| | |
|--------------|--|
| No. Skenario | SK 07 |
| No/Nama UC | UC-07/Jadwal |
| Ringkasan | Pemesan dapat melihat jadwal |
| Aktor | Pemesan |
| Kondisi | Tampil halaman jadwal |
| Deskripsi | User dapat melihat jadwal untuk memastikan tanggal yang ingin dipilih tidak ada pemesan lain |
| Alternatif | - |

Tabel 3.11. SK 08 Halaman Pembayaran

| | |
|--------------|------------------------------------|
| No. Skenario | SK 08 |
| No/Nama UC | UC-08/Pembayaran |
| Ringkasan | Pemesan dapat melakukan pembayaran |

| | |
|------------|--|
| Aktor | Pemesan |
| Kondisi | Tampil halaman pembayaran |
| Deskripsi | User dapat melakukan pembayaran, memilih tempat acara, dan tanggal acara |
| Alternatif | - |

Tabel 3. 12. SK 09 Halaman Login Admin

| | |
|--------------|---|
| No. Skenario | SK 09 |
| No/Nama UC | UC-09/login admin |
| Ringkasan | Admin dapat melakukan login |
| Aktor | Admin |
| Kondisi | Tampil halaman dashboard |
| Deskripsi | Admin dapat melakukan login agar dapat mengelola data website |
| Alternatif | - |

Tabel 3. 13. SK 010 Halaman Dashboard admin

| | |
|--------------|-------------------------------|
| No. Skenario | SK 010 |
| No/Nama UC | UC-010/Dashboard |
| Ringkasan | Admin dapat melihat dashboard |
| Aktor | Admin |
| Kondisi | Tampil halaman dashboard |

| | |
|------------|--|
| Deskripsi | Admin dapat melihat halaman dashboard yang berisi ringkasan di web |
| Alternatif | - |

Tabel 3. 14. SK 011 Halaman kelola web profile

| | |
|--------------|---|
| No. Skenario | SK 0111 |
| No/Nama UC | UC-011/web profile |
| Ringkasan | Admin dapat mengelola web profile |
| Aktor | Admin |
| Kondisi | Tampil halaman kelola web profile |
| Deskripsi | Admin dapat melihat halaman web profile dan mengelola web profile |
| Alternatif | - |

Tabel 3. 15. 012 Halaman kelola paket wedding

| | |
|--------------|-------------------------------------|
| No. Skenario | SK 012 |
| No/Nama UC | UC-012/kelola paket wedding |
| Ringkasan | Admin dapat mengelola paket wedding |
| Aktor | Admin |
| Kondisi | Tampil halaman kelola paket wedding |

| | |
|------------|---|
| Deskripsi | Admin dapat melihat halaman paket wedding dan mengelola paket wedding |
| Alternatif | - |

Tabel 3. 16. SK 013 Halaman kelola pemesanan

| | |
|--------------|--|
| No. Skenario | SK 013 |
| No/Nama UC | UC-012/kelola pemesanan |
| Ringkasan | Admin dapat mengelola pemesanan |
| Aktor | Admin |
| Kondisi | Tampil halaman kelola pemesanan |
| Deskripsi | Admin dapat melihat halaman kelola pemesanan |
| Alternatif | - |

Tabel 3. 17. SK 014 Halaman kelola laporan pemesanan

| | |
|--------------|--|
| No. Skenario | SK 014 |
| No/Nama UC | UC-014/kelola laporan pemesanan |
| Ringkasan | Admin dapat mengelola pemesanan |
| Aktor | Admin |
| Kondisi | Tampil halaman kelola laporan pemesanan |
| Deskripsi | Admin dapat melihat halaman kelola laporan pemesanan |
| Alternatif | - |

Tabel 3. 18. SK 015 Halaman kelola jadwal

| | |
|--------------|--|
| No. Skenario | SK 015 |
| No/Nama UC | UC-015/kelola jadwal |
| Ringkasan | Admin dapat mengelola jadwal |
| Aktor | Admin |
| Kondisi | Tampil halaman kelola jadwal |
| Deskripsi | Admin dapat melihat halaman kelola jadwal dan mengelola jadwal |
| Alternatif | - |

Tabel 3. 19. SK 016 Halaman kelola akun

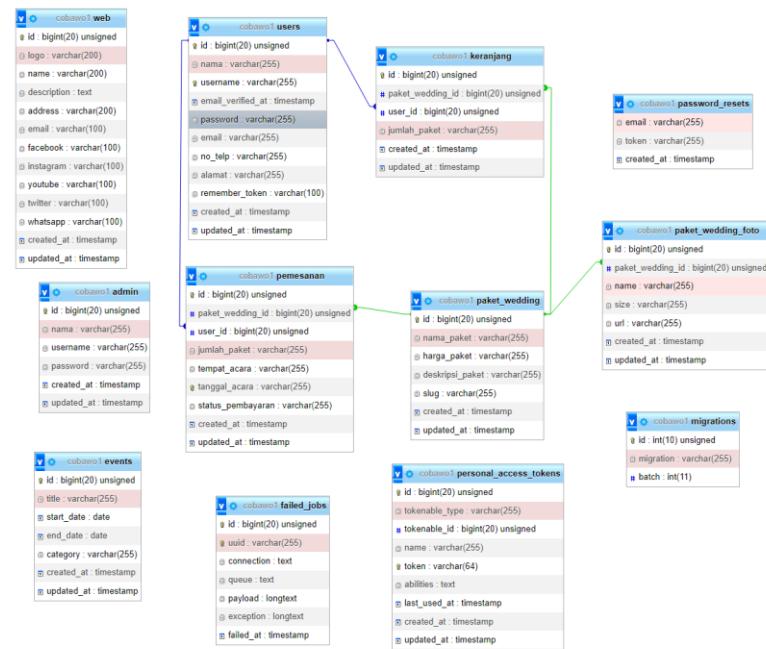
| | |
|--------------|--|
| No. Skenario | SK 016 |
| No/Nama UC | UC-016/kelola akun |
| Ringkasan | Admin dapat mengelola akun |
| Aktor | Admin |
| Kondisi | Tampil halaman kelola akun |
| Deskripsi | Admin dapat melihat halaman kelola akun dan mengelola akun |
| Alternatif | - |

3.4. Perancangan Basis Data

Rancangan basis data untuk membangun aplikasi ini menggunakan data-data yang tersimpan di dalam tabel-tabel pada *database*.

3.4.1. Relasi Tabel

Tabel adalah objek dasar yang merupakan inti dari *database* relasional. Tujuannya adalah menyimpan informasi. Tabel yang satu dengan yang lainnya bisa dihubungkan. Adapun relasi tabel dari *website* terdapat pada gambar 3.3



Gambar 3. 3. Relasi Tabel

3.4.2. Struktur Tabel

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

a. Tabel Users

Tabel users digunakan untuk menyimpan data pengguna *website* untuk *user* yang ingin melakukan pemesanan jasa di *website wedding organizer* ini.

Tabel 3. 20. Struktur tabel users

| Nama Field | Tipe Data | Keterangan |
|----------------|---------------|---|
| Id | Bigint (20) | Primary key, untuk menampung id user / poemesan |
| Nama | Varchar (255) | |
| Username | Varchar (255) | |
| Password | Varchar (255) | |
| Email | Varchar (255) | |
| No_telp | Varchar (255) | |
| Alamat | Varchar (255) | |
| Remember_token | Varchar (100) | |

b. Tabel Admin

Tabel admin digunakan untuk menyimpan data login admin yang berguna untuk verifikasi ke halaman kelola admin.

Tabel 3. 21. Struktur tabel admin

| Nama Field | Tipe Data | Keterangan |
|------------|---------------|---------------------------------------|
| Id | Bigint (20) | Primary key, untuk menampung id admin |
| Nama | Varchar (255) | |
| Username | Varchar (255) | |
| Password | Varchar (255) | |

c. Tabel paket wedding

Tabel paket wedding digunakan untuk menyimpan infomasi mengenai paket wedding.

Tabel 3. 22. Struktur tabel paket wedding

| Nama Field | Tipe Data | Keterangan |
|-----------------|---------------|---------------------------------------|
| Id | Bigint (20) | Primary key, untuk menampung id paket |
| Nama_paket | Varchar (255) | |
| Harga_paket | Varchar (255) | |
| Deskripsi_paket | Varchar (255) | |
| slug | Varchar (255) | |

d. Tabel events

Tabel events digunakan untuk menyimpan data jadwal acara

Tabel 3. 23. Struktur tabel events

| Nama Field | Tipe Data | Keterangan |
|------------|---------------|-------------|
| Id | Bigint (20) | Primary key |
| Title | Varchar (255) | |
| Start_date | Varchar (255) | |
| End_date | Varchar (255) | |
| Category | Varchar (255) | |

e. Tabel Pemesanan

Tabel pemesanan digunakan untuk menyimpan data mengenai informasi pemesanan.

Tabel 3. 24. Struktur tabel pemesanan

| Nama Field | Tipe Data | Keterangan |
|-------------------|---------------|-------------|
| Id | Bigint (20) | Primary key |
| Paket_wedding_id | Bigint (20) | Foreign key |
| User_id | Bigint (20) | Foreign key |
| Jumlah_paket | Varchar (255) | |
| Tempat_acara | Varchar (255) | |
| Tanggal_acara | Varchar (255) | Foreign key |
| Status_pembayaran | Varchar (255) | |

f. Tabel keranjang

Tabel keranjang digunakan untuk menyimpan data keranjang.

Tabel 3. 25. Struktur tabel keranjang

| Nama Field | Tipe Data | Keterangan |
|------------------|---------------|-------------|
| Id | Bigint (20) | Primary key |
| Paket_wedding_id | Bigint (20) | Foreign key |
| User_id | Bigint (20) | Foreign key |
| Jumlah_paket | Varchar (255) | |

g. Tabel paket wedding foto

Tabel paket wedding foto digunakan untuk menyimpan foto dari paket wedding.

Tabel 3.26. Struktur tabel paket wedding foto

| Nama Field | Tipe Data | Keterangan |
|------------------|---------------|-------------|
| Id | Bigint (20) | Primary key |
| Paket_wedding_id | Bigint (20) | Foreign key |
| Name | Varchar (255) | |
| Size | Varchar (255) | |
| Url | Varchar (255) | |

h. Tabel web

Tabel web digunakan untuk menyimpan data informasi mengenai web yang diisi oleh admin

Tabel 3. 27. Struktur tabel web

| Nama Field | Tipe Data | Keterangan |
|-------------|---------------|-------------|
| Id | Bigint (20) | Primary key |
| logo | Varchar (255) | Foreign key |
| Name | Varchar (255) | |
| Description | Text | |
| Address | Varchar (255) | |
| Email | Varchar (255) | |
| Facebook | Varchar (255) | |
| Instagram | Varchar (255) | |

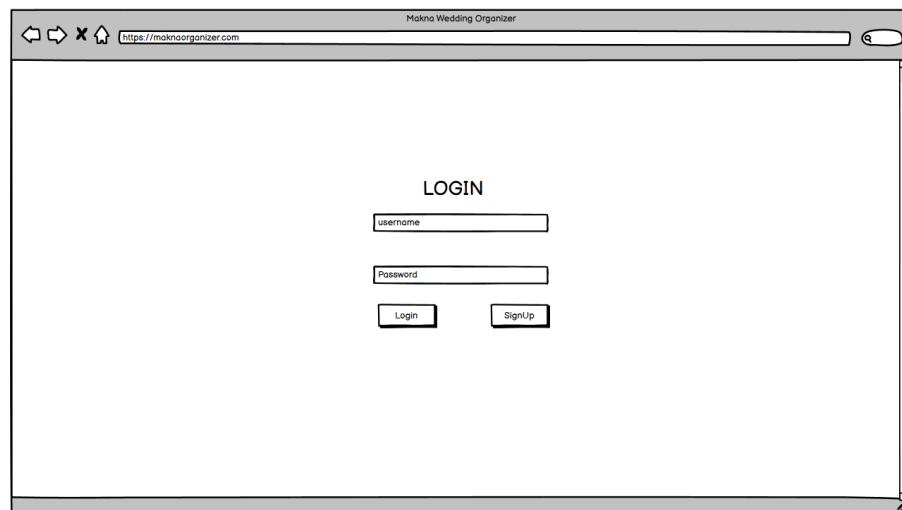
| | | |
|----------|---------------|--|
| Whatsapp | Varchar (255) | |
|----------|---------------|--|

3.5. Perancangan Antar Muka

Didalam rancangan sistem ini penulis menerapkan salah satu dari metodologi pengembangan aplikasi *prototype* yaitu *Modelling Quick Design*, didalam metode ini penulis membuat rancangan sistem antarmuka yang akan diterapkan di dalam *website* yang sesuai dengan hasil dari observasi dan wawancara yang telah penulis laksanakan. Penulis membuat rancangan antarmuka berupa *mockup* atau rancangan dasar yang bisa memberikan gambaran kepada pihak tim Makna *wedding organizer* bagaimana antarmuka nanti yang akan dibuat seperti menu menu yang akan dibuat, fitur, warna, *layout*, *typogragphy*, dan lainnya.

3.5.1. Halaman Login

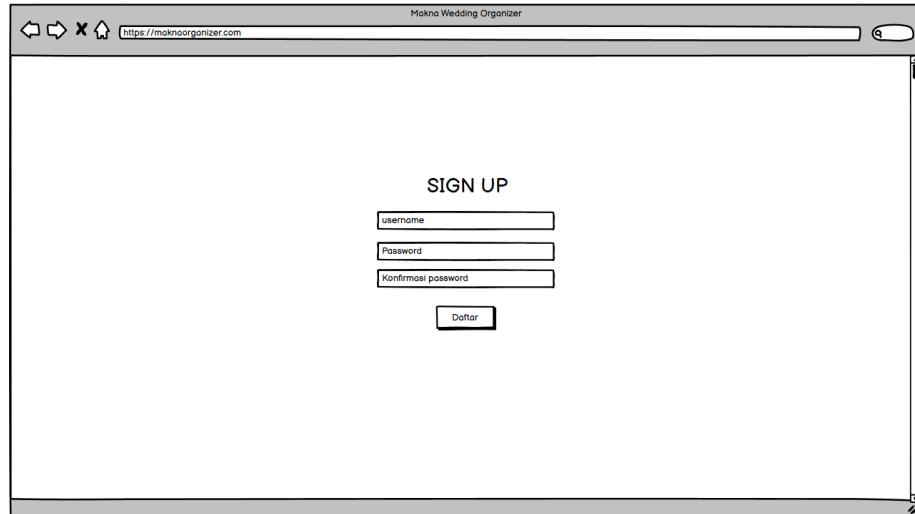
Halaman login adalah suatu proses untuk masuk kedalam sistem yang berisi *username* dan *password*, admin dan pengguna dapat memasukkan *username* dan *password* di kolom *username* dan *password*.



Gambar 3. 4. Tampilan rancangan halaman login

3.5.2. Halaman registrasi

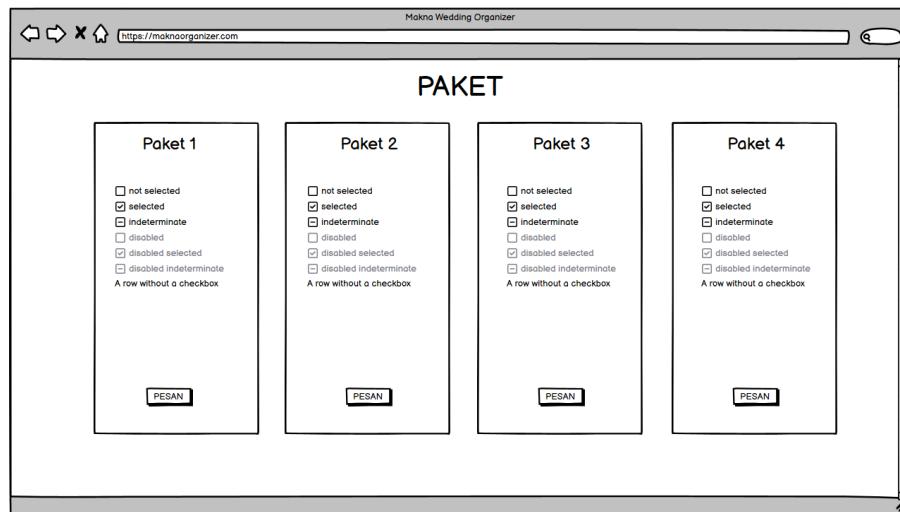
Di halaman ini jika pemesan belum mempunyai akun, pemesan dapat melakukan pendaftaran akun di halaman ini.



Gambar 3. 5. Tampilan rancangan halaman registrasi

3.5.3. Halaman Informasi Paket

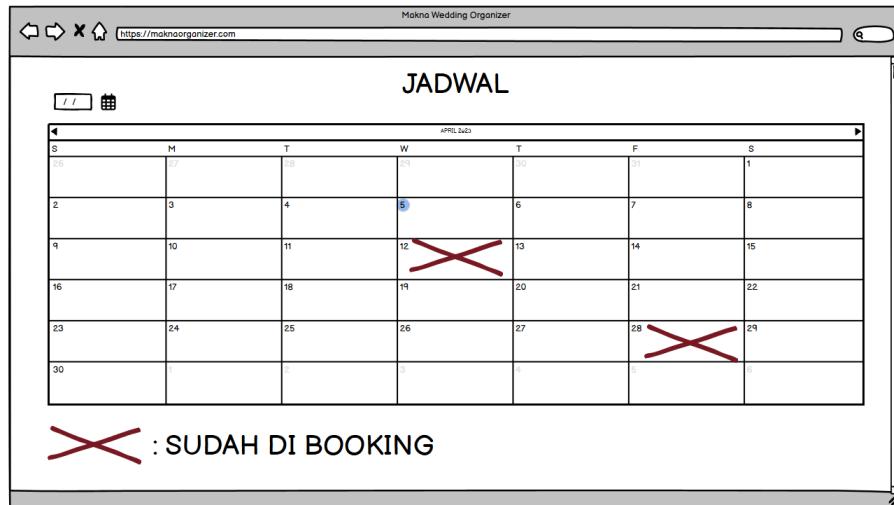
Di halaman informasi paket pemesan dapat melihat beberapa paket yang bisa pemesan pilih sesuai keinginan.



Gambar 3. 6. Tampilan rancangan halaman informasi paket

3.5.4. Halaman Jadwal

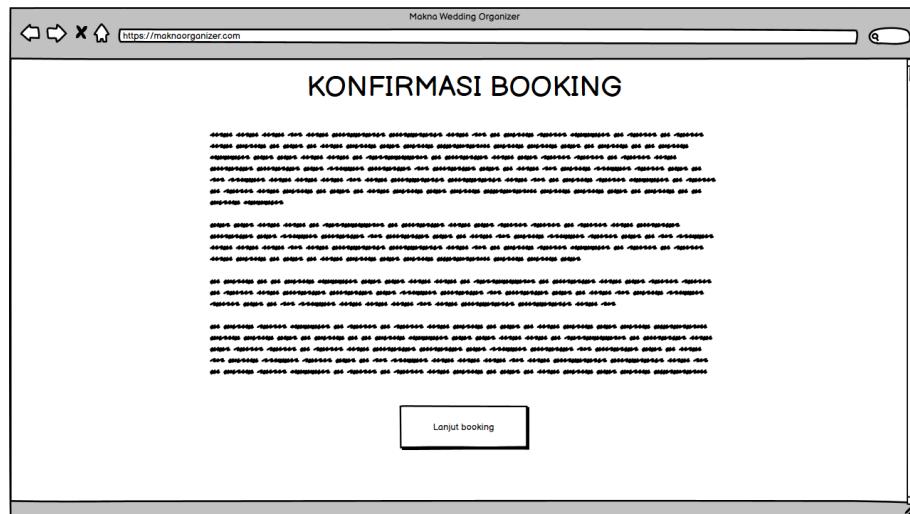
Pada halaman ini pemesan dapat melihat jadwal yang berguna untuk menyesuaikan jadwal acara agar tidak bertabrakan dengan pemesan sebelumnya yang sudah *booking*.



Gambar 3. 7. Tampilan rancangan halaman jadwal

3.5.5. Halaman konfirmasi *booking*

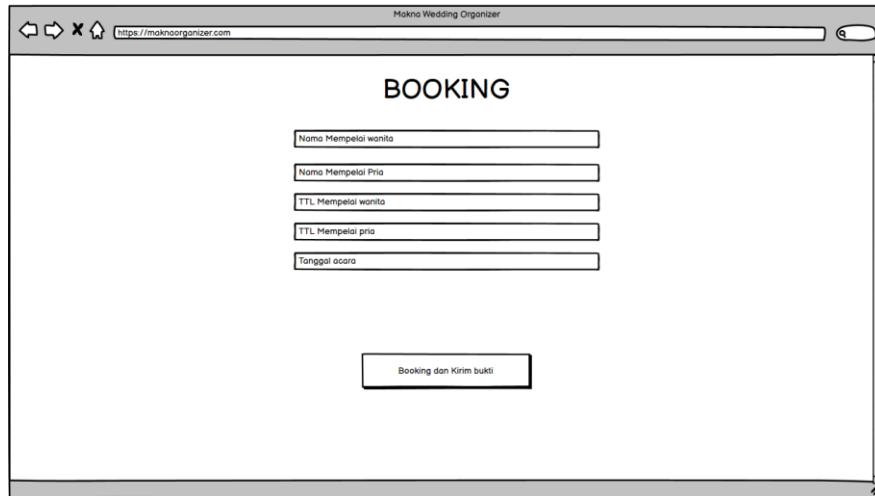
Pada halaman ini pemesan akan ditampilkan beberapa informasi terkait konfirmasi *booking* seperti nama, tanggal, agar pemesan memastikan data yang dimasukkan sudah benar.



Gambar 3. 8. Tampilan rancangan halaman konfirmasi booking

3.5.6. Halaman *booking*

Pada halaman ini pemesan diperintahkan untuk mengisi *form* yang dapat pemesan isi seperti biodata diri dan tanggal acara.

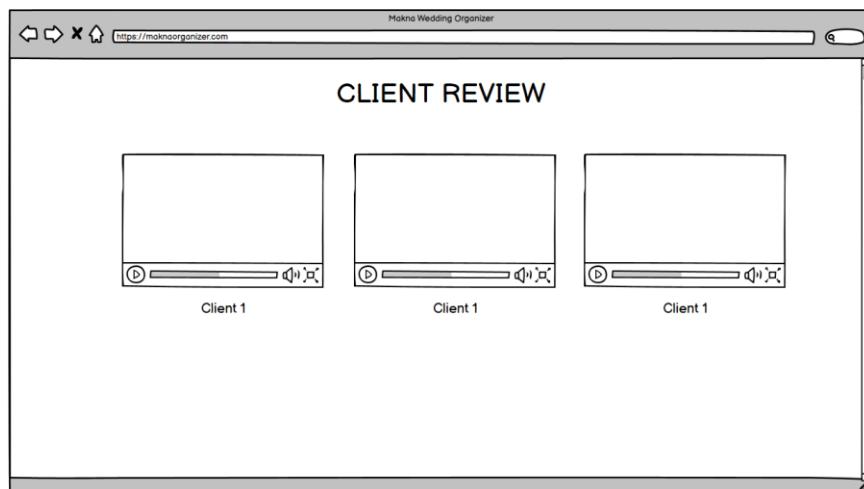


Rancangan halaman booking yang menampilkan formulir dengan inputan untuk Nama Mempelai wanita, Nama Mempelai Pria, TTL Mempelai wanita, TTL Mempelai pria, dan Tanggal acara. Di bawahnya terdapat tombol "Booking dan Kirim bukti".

Gambar 3. 9. Tampilan rancangan halaman booking

3.1. Halaman Client review

Pada halaman ini pemesan dapat melihat beberapa *review* dari *client* sebelumnya yang sudah menggunakan jasa dari Makna *wedding organizer*.



Gambar 3. 10. Tampilan rancangan halaman client review

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Halaman Antar Muka

4.1.1. Tampilan Halaman Beranda

Halaman beranda merupakan halaman utama pada saat pemesan mengakses *website* Makna *Wedding organizer*. Pada halaman beranda, menampilkan halaman berupa kata slogan dari Makna *Wedding Organizer*; Terdapat tombol Produk kami yang berfungsi untuk melihat paket-paket yang tersedia, serta terdapat beberapa galeri foto dari *client* Makna *Wedding Organizer* yang sudah menggunakan jasa *wedding organizer*. Terdapat juga bagian *footer* dari *web* yang berisi sosial media dari Makna *Wedding Organizer*, Lokasi dan informasi, serta pintasan galeri yang mengarahkan pemesan pada sosial media *Instagram* dari Makna *Wedding Organizer*. Tampilan dari halaman beranda dapat dilihat pada gambar 4.1



Beranda Paket Wedding About Client Review

Mari Rencanakan Pernikahan Yang Sempurna Bersama Makna Wedding Organizer

Persiapkan Pernikahan dengan Beragam
Kemudahan & Penawaran Eksklusif

[Produk Kami](#)



Ikuti Kita di Instagram

Galleri yang indah bukti pencapaian dan kualitas dari jasa
yang kami sediakan.





Sosial Media

f
i
e-mail
s

Lokasi Dan Informasi

- 📍 Jl. Putri Candramidi
Gg Sudihati
- 🕒 9:00 am to 5:00 pm
Senin - Sabtu
- 📞 +6281258609161

Galleri








Gambar 4.1. Tampilan Halaman Beranda

4.1.2. Source Code Tampilan Halaman Beranda

Di kode ini penulis menggunakan *section* serta tag *div* yang berguna untuk membaca *class-class* dari *CSS*. Potongan *source code* halaman beranda dapat dilihat pada gambar 4.2

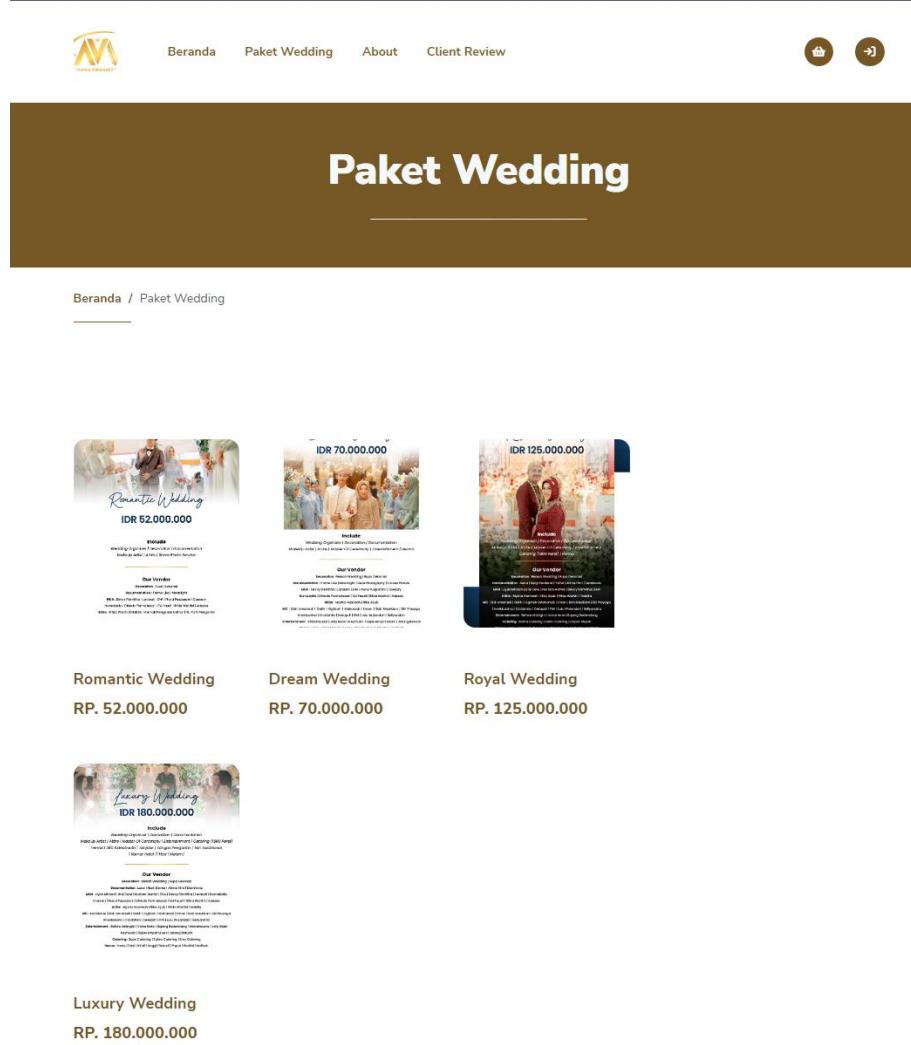
```
@section('content')
    <!-- Hero section start -->
    <section class="hero">
        <div class="container d-xl-flex align-items-start">
            <div class="hero_about col-xl-6">
                <div class="hero_header">
                    <h2 class="hero_header-title" style="color: #765827;">Mari Rencanakan Pernikahan Yang Sempurna Bersama
                    Makna Wedding Organizer</h2>
                    <p class="hero_header-text" style="color: black;">
                        Persiapkan Pernikahan dengan Beragam Kemudahan & Penawaran Ekslusif
                    </p>
                    <a class="hero_header-btn btn" href="{{ route('paket-wedding') }}">Produk Kami</a>
                </div>
            </div>
            <div class="hero_promo col-xl-6">
                <div>
                    <picture>
                        <source data-srcset="{{ asset('home5.png') }}" srcset="{{ asset('home5.png') }}"
                            type="image/webp" />
                        
                    </picture>
                </div>
            </div>
        </div>
    </section>
    <!-- Hero section end -->
    <!-- Featured products section start -->

    <!-- Instagram section start -->
    <section class="instagram" style="margin-top: 100px;">
        <div class="container">
            <div class="instagram_header">
                <h2 class="instagram_header-title" style="color: #765827;">Ikuti Kita di Instagram</h2>
                <p class="instagram_header-text" style="color: black;">
                    Galleri yang indah bukti pencapaian dan kualitas dari jasa yang kami sediakan.
                </p>
            </div>
        </div>
        <div class="instagram_slider swiper">
            <div class="swiper-wrapper">
                <div class="instagram_slider-slide swiper-slide">
                    <a class="link" href="https://instagram.com" target="_blank" rel="noopener noreferrer">
                        <picture>
                            <source data-srcset="{{ asset('footer-1.jpg') }}" srcset="{{ asset('footer-1.jpg') }}"
                                type="image/webp" />
                            
                        </picture>
                        <span class="overlay d-flex justify-content-center align-items-center">
                            <i class="icon-instagram"></i>
                        </span>
                    </a>
                </div>
                <div class="instagram_slider-slide swiper-slide">
                    <a class="link" href="https://instagram.com" target="_blank" rel="noopener noreferrer">
                        <picture>
                            <source data-srcset="{{ asset('footer-2.jpg') }}" srcset="{{ asset('footer-2.jpg') }}"
                                type="image/webp" />
                            
                        </picture>
                        <span class="overlay d-flex justify-content-center align-items-center">
                            <i class="icon-instagram"></i>
                        </span>
                    </a>
                </div>
            </div>
        </div>
    </section>
    <!-- Instagram section end -->
```

Gambar 4. 2. Source code halaman beranda

4.1.3. Tampilan Halaman Paket Wedding

Halaman Paket *Wedding* merupakan halaman yang menampilkan beberapa informasi terkait dengan paket *wedding* apa saja yang terdapat pada Makna *Wedding Organizer*. Paket yang ditawarkan oleh Makna *Wedding Organizer* untuk saat ini terdapat 3 paket yaitu *Romantic Wedding*, *Dream Wedding*, *Royal Wedding*, dan *Luxury Wedding*. Di halaman ini juga menampilkan harga dari paket-paket yang tersedia, serta di halaman ini pemesan dapat memasukkan paket yang diinginkan ke dalam keranjang dan melihat detail dari paket. Tampilan dari halaman Paket *Wedding* dapat dilihat pada gambar 4.3



Gambar 4. 3. Tampilan halaman paket wedding

4.1.4. Source Code Tampilan Halaman Paket Wedding

Di Halaman Paket Wedding ini penulis menggunakan *tag* dari HTML berupa *div* yang berfungsi untuk memanggil *class* dari *CSS*, serta pada halaman ini penulis menggunakan perulangan *foreach* untuk melakukan perulangan dan memanggil data dari *database* untuk menampilkan data yang sudah dimasukkan admin. *Source Code* dari halaman paket *wedding* dapat dilihat pada gambar 4.4

```

46     <div class="shop-wrapper section--noph">
47         <div class="container d-flex flex-lg-row flex-wrap flex-column justify-content-between">
48             <div class="shop_products d-flex flex-column">
49                 <ul class="shop_products-list d-sm-flex flex-wrap">
50                     @foreach ($paket_wedding as $item)
51                         @foreach ($item->getFirstFoto as $foto)
52                             <li class="shop_products-list-item col-sm-6 col-md-4 col-lg-6 col-xl-4" data-order="1">
53                                 <div class="wrapper d-flex flex-column">
54                                     <div class="media">
55                                         <div>
56                                         <div class="overlay d-flex flex-column align-items-center justify-content-center">
57                                             <ul class="action d-flex align-items-center justify-content-center">
58                                                 <li class="list-item">
59                                                     <input type="hidden" name="paket_wedding_id"
60                                                       id="paketWeddingId({$item->id})"
61                                                       value="{{ $item->id }}"
62                                                     @if (Auth::guard('web')->user())
63                                                         <input type="hidden" name="user_id"
64                                                       id="userId({$item->id})"
65                                                       value="{{ isset(Auth::guard('web')->user()->id) ? Auth::guard('web')->user()->id : '' }}"
66                                                     @endif
67                                                     @if (Auth::guard('web')->user())
68                                                       @php
69                                                       $check = \App\Models\Pemesanan::where('user_id', Auth::guard('web')->user()->id)->count();
70                                                       @endphp
71                                                       @if ($check >= 1)
72                                                         <a href="{{ route('pemesanan-wa', Auth::guard('web')->user()->id) }}">
73                                                             <button type="button"
74                                                               class="action_link d-flex align-items-center justify-content-center">
75                                                               <i class="icon-basket"></i>
76                                                         </a>
77                                                       @else
78                                                         <button type="button"
79                                                           class="action_link d-flex align-items-center justify-content-center"
80                                                           onclick="keranjangButtonInPaketWeeding({{$item->id}})">
81                                                               <i class="icon-basket"></i>
82                                                         </button>
83                                                       @endif
84                                                       @else
85                                                         <a data-bs-toggle="offcanvas" data-bs-target="#loginCanvas">

```

Gambar 4. 4. Source code halaman paket wedding

4.1.5. Tampilan Halaman *About*

Halaman *About* / tentang merupakan halaman yang menampilkan deskripsi mengenai Makna *Wedding Organizer*; di halaman ini terdapat galeri foto tim dari Makna *Wedding Organizer* serta penjelasan mengenai apa itu Makna *Wedding Organizer*. Tampilan dari halaman *About* dapat dilihat pada gambar 4.5



Beranda / About



Tentang Kami

MAKNA ORGANIZER adalah salah satu wedding organizer yang ada di Pontianak - Kalimantan Barat yang membantu merencanakan dan mewujudkan impian acara pernikahan calon pengantin mulai dari persiapan, budgeting, pertemuan keluarga, membuat janji temu dan menemani saat meeting dengan vendor, merekomendasikan vendor terbaik yang sesuai dengan harapan calon pengantin, menemani saat fitting baju pengantin, hingga mengkoordinir pelaksanaan pada saat acara dilaksanakan



Gambar 4. 5. Tampilan about

4.1.6. Source Code Halaman About

Pada halaman *About* ini penulis menggunakan *tag-tag* HTML seperti *div* yang berfungsi untuk memanggil *class* dari *CSS*. Untuk deskripsi penulis menggunakan *tag p* yang berfungsi untuk paragraph. *Source Code* halaman *About* dapat dilihat pada gambar 4.6

```


## Tentang Kami



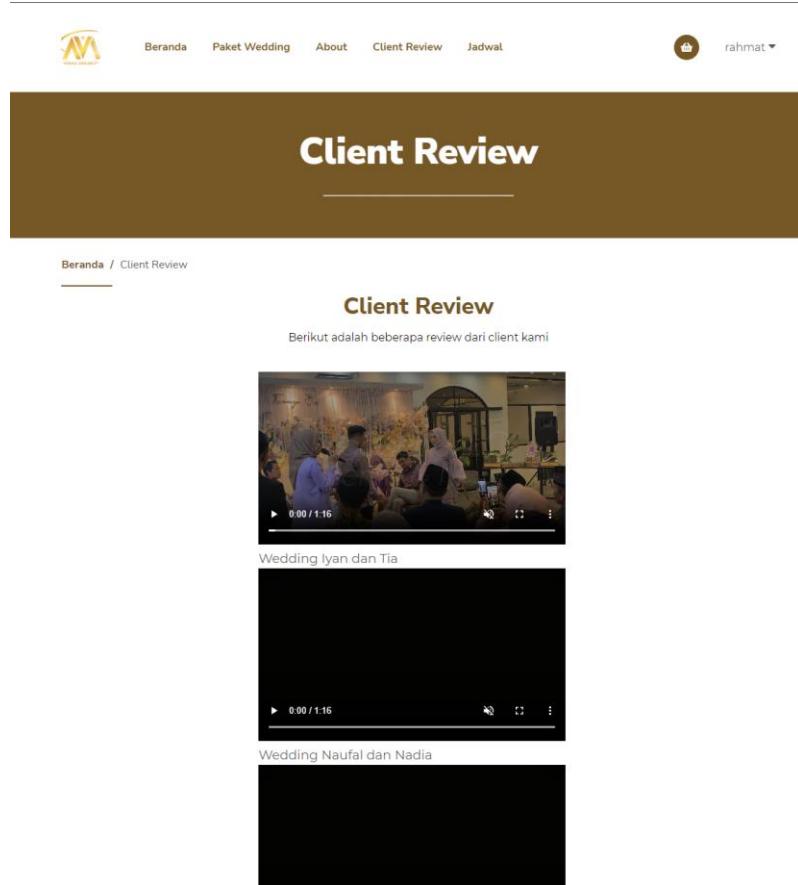
MAKNA ORGANIZER adalah salah satu wedding organizer yang ada di Pontianak - Kalimantan Barat yang membantu merencanakan dan mewujudkan impian acara pernikahan calon pengantin mulai dari persiapan, budgeting, pertemuan keluarga, membuat janji temu dan menemani saat meeting dengan vendor, merekomendasikan vendor terbaik yang sesuai dengan harapan calon pengantin, menemani saat fitting baju pengantin, hingga mengkoordinir pelaksanaan pada saat acara dilaksanakan


```

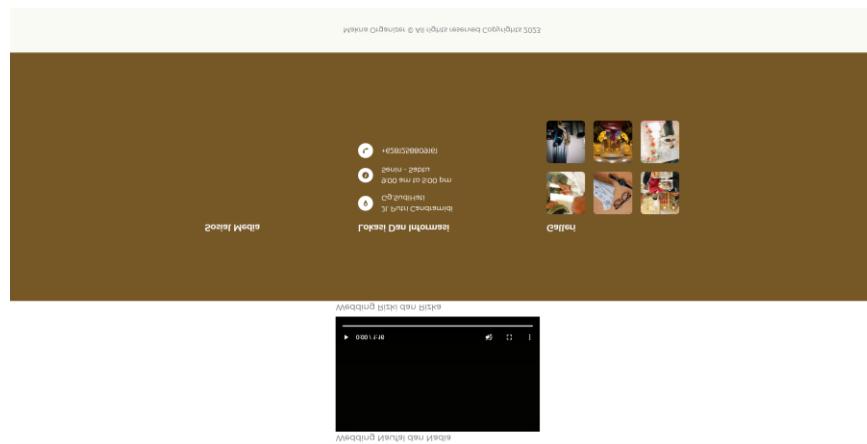
Gambar 4. 6. Source code dari halaman about

4.1.7. Tampilan Halaman *Client Review*

Halaman *Client Review* merupakan halaman yang akan menampilkan informasi berupa *video* mengenai ulasan dari pemesan yang sudah menggunakan jasa *Wedding Organizer* pada Makna *Wedding Organizer* berupa kesan dan pesan untuk Makna *Wedding Organizer*. Tujuan dari halaman ini adalah memberikan rasa percaya pada pemesan untuk memilih Makna *Wedding Organizer* sebagai *Wedding Organizer*. Tampilan Halaman *Client Review* dapat dilihat pada gambar 4.7 dan gambar 4.8



Gambar 4. 7. Tampilan halaman client review



Gambar 4. 8. Tampilan halaman client review -footer

4.1.8. Source Code Halaman Client Review

Pada halaman *Client Review* ini penulis menggunakan *tag* dari HTML berupa *div* yang berfungsi untuk memanggil *class* dari *CSS*, penulis juga menggunakan *tag video* yang berfungsi untuk menampung *file video*. *Source Code* tampilan *Client Review* dapat dilihat pada gambar 4.9

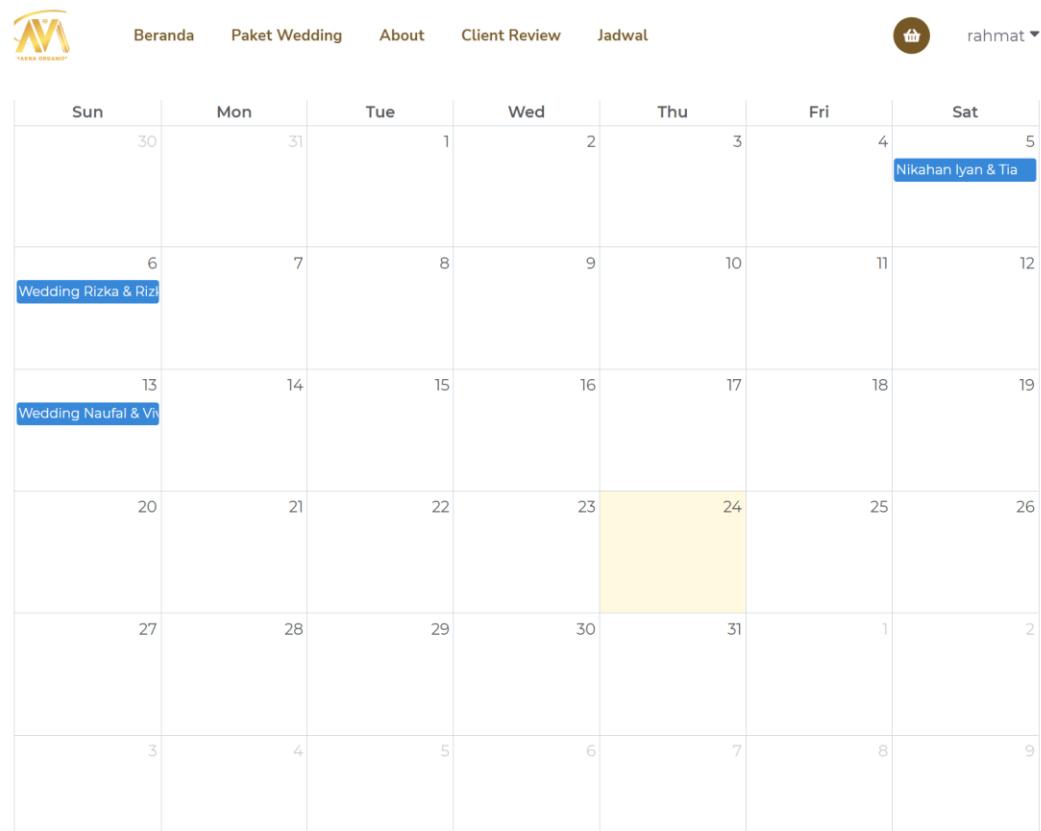
```
<div class="video-container">
  <div class="video-wrapper">
    {{-- <div class="video-container"> --}}
    <video controls muted>
      <source src="{{ asset('LAMARAN.mp4') }}" type="video/mp4">
    </video>
    <p class="description">Wedding Iyan dan Tia</p>
    {{-- </div> --}}
    {{-- <div class="video-container"> --}}
    <video controls muted>
      <source src="{{ asset('LAMARAN.mp4') }}" type="video/mp4">
    </video>
    <p class="description">Wedding Naufal dan Nadia</p>
    {{-- </div> --}}
    {{-- <div class="video-container"> --}}
    <video controls muted>
      <source src="{{ asset('LAMARAN.mp4') }}" type="video/mp4">
    </video>
    <p class="description">Wedding Rizki dan Rizka</p>
    {{-- </div> --}}
  </div>
</div>

</div>
```

Gambar 4.9. Source code halaman client review

4.1.9. Tampilan Halaman Jadwal

Halaman Jadwal merupakan halaman yang akan menampilkan tanggal, bulan dan tahun, yang berfungsi untuk memberikan informasi ke pemesan terkait jadwal atas ketersediaan tanggal, bulan , dan tahun. Halaman ini dapat di akses oleh pemesan setelah berhasil melakukan *login*. Terdapat sebuah tanda berwarna biru yang menandakan bahwa tanggal yang ditandai sudah terdapat pemesan lain di tanggal tersebut, dengan kesimpulan pemesan harus memilih tanggal lain yang tersedia yang tidak ada tanda di kotak tanggal. Tampilan halaman Jadwal dapat dilihat pada gambar 4.10



Gambar 4.10. Tampilan halaman jadwal

4.1.10. *Source Code* Tampilan Halaman Jadwal

Pada halaman Jadwal penulis membuat kalender menggunakan *javascript* dan *package bootstrap* berupa *datepicker*, serta di kode ini penulis memanggil data dari *database* yang dimana data tersebut sudah di *input* oleh admin. *Source code* dari halaman dapat dilihat pada gambar 4.11

```

document.addEventListener('DOMContentLoaded', function() {
    var calendarEl = document.getElementById('calendar');
    var calendar = new FullCalendar.Calendar(calendarEl, [
        initialView: 'dayGridMonth',
        themeSystem: 'bootstrap5',
        events: `{{ route('events.list') }}`,
        editable: false, // Mengubah opsi editable menjadi false
        dateClick: function (info) {
            // Menghilangkan aksi dateClick
        },
        eventClick: function ({event}) {
            // Menghilangkan aksi eventClick
        },
        eventDrop: function (info) {
            // Menghilangkan aksi eventDrop
        },
        eventResize: function (info) {
            // Menghilangkan aksi eventResize
        }
    ]);
    calendar.render();
});

```

Gambar 4. 11. Source code halaman jadwal

4.1.11. Tampilan Halaman Detail Paket

Halaman Detail Paket merupakan halaman yang akan tampil ketika pemesan menekan tombol yang berlogo mata pada halaman paket *wedding*. Saat halaman ini di akses maka akan menampilkan beberapa informasi mengenai paket yang di pilih seperti nama paket, harga, deskripsi paket, dihalaman ini juga terdapat tombol tambah keranjang yang berfungsi untuk menambahkan paket yang dipilih ke dalam keranjang. Di halaman ini juga terdapat menu paket lainnya yang berfungsi jika pemesan ingin memilih atau melihat detail dari paket lain dapat dengan mudah memilih paket. Tampilan halaman Detail Paket dapat di lihat pada gambar 4.12

The screenshot shows a website for a wedding service. At the top, there is a navigation bar with links to Beranda, Paket Wedding, About, Client Review, Jadwal, and a user account section labeled 'rahmat ▾'. Below the navigation is a large brown header with the text 'Detail Paket' in white. Underneath the header, the breadcrumb navigation shows 'Beranda / Detail Paket'. The main content area features a photograph of a wedding couple and their guests. Overlaid on the photo is the text 'Romantic Wedding' and 'IDR 52.000.000'. Below the photo, there is a section titled 'Include' listing services like Wedding Organizer, Decoration, Documentation, MakeUp Artist, Attire, and Share Photo Service. Another section titled 'Our Vendor' lists various service providers. To the right of the main image, the package name 'Romantic Wedding' is repeated in a larger font, followed by its price 'RP. 52.000.000'. A 'Tambah Ke Keranjang' (Add to Cart) button is located at the bottom right of this section. A small watermark '#SatuhariPenuhMajna' is visible at the bottom of the page.

Paket Lainnya



Dream Wedding

RP. 70.000.000



Romantic Wedding

RP. 52.000.000



Luxury Wedding

RP. 180.000.000



Royal Wedding

RP. 125.000.000

Gambar 4.12. Tampilan halaman detail paket

4.1.12. Source Code Tampilan Halaman Detail Paket

Pada halaman Detail Paket penulis menggunakan beberapa *tag* HTML seperti *section* yang berfungsi untuk pengelompokan konten, *div* yang berfungsi untuk memanggil *class* dari *CSS*, pada halaman detail paket penulis juga menggunakan perulangan *foreach* yang berfungsi untuk memanggil dan menampilkan data dari *database*, data tersebut di dapat dari hasil inputan admin. *Source Code* halaman Detail Paket dapat dilihat pada gambar 4.13

```

<section class="about section--nopb">
    <div class="container">
        <!-- Product main -->
        <div class="about_main d-lg-flex flex-wrap">
            <div class="about_main-slider">
                <div class="about_main-slider--single" data-modal="false">
                    <div class="swiper-wrapper">
                        @foreach ($paket_wedding->get_all_foto as $foto)
                            <div class="swiper-slide">
                                <a href="{{ asset('paket_wedding/' . $foto->url) }}" data-role="gallery">
                                    <picture>
                                        <source data-srcset="{{ asset('paket_wedding/' . $foto->url) }}"
                                                srcset="{{ asset('paket_wedding/' . $foto->url) }}" type="image/webp" />
                                        url) }}" alt="media" />
                                    </picture>
                                </a>
                            </div>
                        @endforeach
                    </div>
                    <div class="swiper-controls d-flex align-items-center justify-content-between">
                        <a class="swiper-button-prev d-inline-flex align-items-center justify-content-center" href="#">
                            <i class="icon-caret_left icon"></i>
                        </a>
                        <a class="swiper-button-next d-inline-flex align-items-center justify-content-center" href="#">
                            <i class="icon-caret_right icon"></i>
                        </a>
                    </div>
                </div>
                <div class="about_main-slider--thumbs">
                    <div class="swiper-wrapper">
                        @foreach ($paket_wedding->get_all_foto as $foto)
                            <div class="swiper-slide">
                                <picture>
                                    <source data-srcsets="{{ asset('paket_wedding/' . $foto->url) }}"
                                            srcset="{{ asset('paket_wedding/' . $foto->url) }}" type="image/webp" />
                                    url) }}" alt="media" />
                                </picture>
                            </div>
                        @endforeach
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Gambar 4.13 *Source Code* Halaman Detail Paket

4.1.13. Tampilan Halaman Keranjang

Halaman keranjang merupakan halaman yang akan menampilkan informasi mengenai ringkasan dari paket yang sudah pemesan pilih dan sudah masuk kedalam keranjang. Pada halaman ini menampilkan paket yang dipilih, nama paket, harga paket, serta pemesan bisa memilih berapa banyak paket yang ingin dipilih dengan cara menekan tombol tambah atau kurang, pemesan juga dapat menghapus pesanan dengan cara menekan tombol hapus, terdapat juga total dari harga yang harus dibayar oleh pemesan. Di halaman keranjang terdapat dua tombol, tombol pertama yaitu tombol untuk melanjutkan pembayaran, yang akan mengarahkan pemesan ke

halaman pembayaran, terdapat juga tombol jadwal yang berfungsi mengarahkan pemesan ke halaman jadwal, terdapat juga warning dengan tulisan merah yang menginformasikan kepada pemesan untuk melihat jadwal terlebih dahulu sebelum melakukan pembayaran. Halaman ini dapat diakses jika pengguna sudah berhasil melakukan *login*. Tampilan halaman keranjang dapat dilihat pada gambar 4.14



Gambar 4.15 Tampilan Halaman Keranjang

4.1.14. Source Code Halaman Keranjang

Pada halaman Keranjang penulis menggunakan beberapa *tag* dari HTML seperti *li* yang berfungsi untuk membuat *list* didalam halaman, *div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag a href* yang berfungsi untuk melakukan *linking* ke target tertentu, *tag picture* yang berfungsi untuk menyimpan data gambar, serta penulis menggunakan perulangan *foreach* untuk menampilkan data di halaman yang data tersebut diambil dari *database model*. *Source code* halaman keranjang dapat dilihat pada gambar 4.16

```

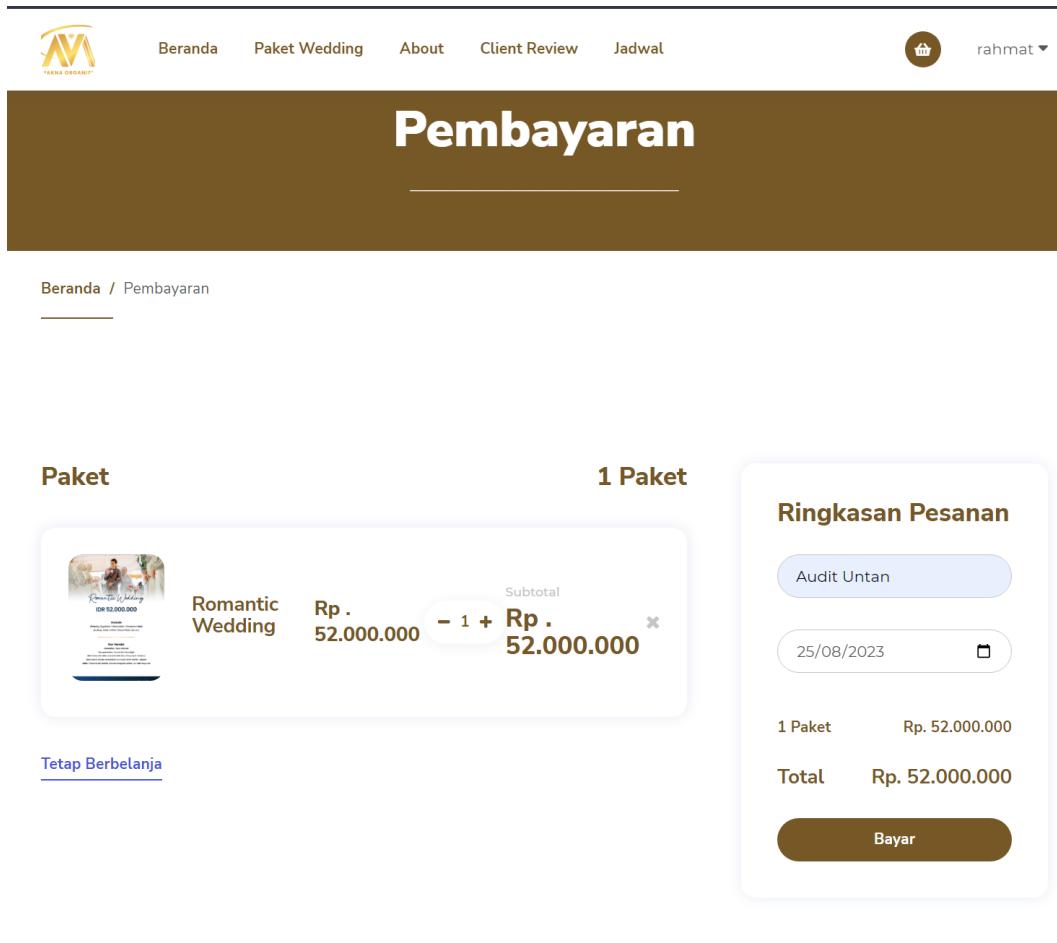
?>
@foreach ($keranjang as $data)
    @foreach ($data->paket_wedding->get_first_foto as $foto)
        <?php
        $jumlah_paket_value += (int) $data->jumlah_paket * (int) $data->paket_wedding->harga_paket;
    ?>
<li class="cartOffcanvas_body-list_item d-sm-flex align-items-center">
    <div class="media">
        <a href="{{ url('paket-wedding', $data->paket_wedding->slug) }}"
            rel="noopener noreferrer" style="width: 100% !important;">
            <picture style="width: 100% !important;">
                <source data-srcset="{{ asset('paket_wedding/' . $foto->url) }}"
                    srcset="{{ asset('paket_wedding/' . $foto->url) }}"
                    type="image/webp" />
                url) }}"
                    style="height: 300px !important; width: 100% !important; object-fit: cover;"
                    alt="media" />
            </picture>
        </a>
    </div>
    <div
        class="main d-flex flex-wrap justify-content-between align-items-end align-items-lg-center">
        <a class="main_title"
            href="{{ url('paket-wedding', $data->paket_wedding->slug) }}"
            rel="noopener noreferrer">
            <span
                class="main_title-product"><{ $data->paket_wedding->nama_paket }></span>
        </a>
        <div class="main_price">
            <span class="price">Rp .
                {{ number_format((float) $data->paket_wedding->harga_paket, 0, ',', '.') }}</span>
        </div>
        <div class="qty d-flex align-items-center justify-content-between">
            <span class="d-flex align-items-center"
                onclick="kurangCanvas('{{ $data->id }}')">
                <i class="icon-minus" style="color: #765827"></i>
            </span>
            <input class="qty_amount" type="number"
                id="quantityValue{{ $data->id }}"
                readonly
                value="{{ $data->jumlah_paket }}"
                min="1" max="99" />
        </div>
    </div>
</li>

```

Gambar 4.16 Source Code Halaman Keranjang

4.1.15. Tampilan Halaman Pembayaran

Halaman Pembayaran merupakan halaman yang akan tampil ketika pemesan menekan tombol lanjutkan pembayaran di halaman keranjang, dan pemesan harus melakukan *login* terlebih dahulu untuk mengakses halaman pembayaran. Pada halaman pembayaran terdapat informasi terkait paket yang sudah pemesan pilih, terdapat juga *form* ringkasan pesan yang bisa pemesan isi yang berupa tempat acara, tanggal acara, di *form* ini terdapat juga total paket yang di pilih dan total harga yang harus di bayar, di halaman pembayaran ini juga terdapat tombol bayar, yang berfungsi untuk menginput data ke dalam *database* dan pemesan akan diarahkan ke halaman pembayaran-wa. Tampilan halaman Pembayaran dapat dilihat pada gambar 4.17



Gambar 4.17 Tampilan Halaman Pembayaran

4.1.16. *Source Code* Halaman Pembayaran

Pada halaman Pembayaran penulis menggunakan beberapa *tag* HTML seperti *tag section* yang berfungsi untuk mengelompokkan konten, *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag h* yang berfungsi untuk membuat judul berupa *heading*, *tag image* dan *img* yang berfungsi untuk menampung data gambar, *tag form* untuk membuat formulir, *tag input* berfungsi untuk membuat *field* pada *form* agar pemesan bisa menginputkan data di dalam *form*. Dihalaman ini penulis juga menggunakan perulangan *foreach* yang berfungsi untuk menampilkan data dari *database model* dengan *variable* tertentu. *Source code* dari halaman Pembayaran dapat dilihat pada gambar 4.18 dan gambar 4. 19

```

<section class="cart_section">
    <div class="container d-md-flex justify-content-between align-content-start">
        <div class="cart_main">
            <h3 class="cart_main-header d-flex align-items-center justify-content-between">
                Paket
                <span>({ $keranjang->count() }) Paket</span>
            </h3>
            <ul class="cart_main-list">
                <php
                    $jumlah_all_paket_value = 0;
                >
                @foreach ($keranjang as $data)
                    @foreach ($data->paket_wedding->get_first_foto as $foto)
                        <php
                            $jumlah_all_paket_value += (int) $data->jumlah_paket * (int) $data->paket_wedding->harga_paket;
                        >

                        <php
                            $jumlah_paket_value = (int) $data->jumlah_paket * (int) $data->paket_wedding->harga_paket;
                        >
                        <input type="hidden" name="paket_wedding_id_all[]" class="paket_wedding_id_all"
                               value="{{ $data->id }}"/>
                        <input type="hidden" name="jumlah_paket_bayar[]" class="jumlah_paket_bayar"
                               value="{{ $data->jumlah_paket }}"/>
                    <li class="cart_main-list-item">
                        <div>
                            class="wrapper d-flex flex-wrap flex-xl nowrap align-items-center justify-content-between">
                                <div class="wrapper_item d-flex align-items-center">
                                    <div class="wrapper_item-media">
                                        <picture>
                                            <source data-srcset="{{ asset('paket_wedding/' . $foto->url) }}"
                                                    srcset="{{ asset('paket_wedding/' . $foto->url) }}"
                                                    type="image/webp" style="object-fit: cover !important;"/>
                                            <img alt="Media placeholder" style="height: 140px; object-fit: cover !important;"/>
                                        </picture>
                                    </div>
                                    <div class="wrapper_item-info">
                                        <h4 class="title">{{ $data->paket_wedding->nama_paket }}</h4>
                                    </div>
                                </div>
                                <div class="price_wrapper d-flex flex-column">
                                    <span class="price">Rp .
                                        {{ number_format((float) $data->paket_wedding->harga_paket, 0, ',', '.') }}</span>
                                </div>
                                <div class="price_wrapper price_wrapper--subtotal d-flex flex-column">
                                    <h5 class="title">Subtotal</h5>
                                    <span class="price price--total" id="sub_total{{ $data->id }}">Rp .
                                        {{ number_format((float) $jumlah_paket_value, 0, ',', '.') }}</span>
                                </div>
                            </div>
                        </div>
                    </li>
                @endforeach
            </ul>
        </div>
    </div>

```

Gambar 4.18 . Source Code Halaman Pembayaran

```

<span class="d-flex align-items-center"
      onclick="kurangInCard('{{ $data->id }}")">
    <i class="icon-minus"></i>
</span>
<input class="qty_amount" type="number" readonly
       value="{{ $data->jumlah_paket }}" min="1" max="99"
       id="quantity{{ $data->id }}"/>
<span class="d-flex align-items-center"
      onclick="tambahInCard('{{ $data->id }}")">
    <i class="icon-plus"></i>
</span>
</div>
<input type="hidden" id="keranjangIdCard{{ $data->id }}"
       value="{{ $data->id }}"/>
<button type="button" class="close d-flex align-items-center align-items-sm-start"
        onclick="deleteData(this, {{ $data->id }})">
    <i class="icon-close"></i>
</button>
</div>
</li>
@endforeach
@endforeach
</ul>
</div>
<div class="cart_main-action d-flex flex-column flex-sm-row align-items-center justify-content-sm-between">
    <a class="btn--underline" href="{{ route('paket-wedding') }}>Tetap berbelanja</a>
</div>
</div>
<div class="cart_summary">
    <h3 class="cart_summary-header">Ringkasan Pesanan</h3>
    <form class="cart_summary-form" action="#" method="post" id="bayarPaketForm">
        <input class="cart_summary-form-field" type="hidden" name="user_id"
              value="{{ Auth::guard('web')->user()->id }}"/>
        <input class="cart_summary-form-field" type="text" name="tempat_acara"
              placeholder="Tempat Acara" />
        <input class="cart_summary-form-field" type="date" id="tanggal_acara" name="tanggal_acara"
              placeholder="Tanggal Acara" style="margin-top: 15px; />
    </form>
    <div class="cart_summary-table">
        <div class="cart_summary-table_row d-flex justify-content-between">
            <span class="property">{{ count($keranjang) }} Paket</span>
            <span class="value" id="paketTotal">Rp. @php echo isset($jumlah_all_paket_value) ? number_format((float)$jumlah_all_paket_value, 0, ',', '.') : '' @endphp</span>
        </div>
        <div class="cart_summary-table_row cart_summary-table_row--total d-flex justify-content-between">
            <span class="property">Total</span>
            <span class="value" id="totalharga">Rp. @php echo isset($jumlah_all_paket_value) ? number_format((float)$jumlah_all_paket_value, 0, ',', '.') : '' @endphp</span>
        </div>
    </div>
    <button type="button" class="cart_summary-btn btn" id="bayarPaket">Bayar</button>
</div>
</div>
</section>

```

Gambar 4.19 Source Code Halaman Pembayaran

4.1.17. Halaman *Login*

Halaman *Login* merupakan tampilan yang memiliki fungsi sebagai media awal pemesan / *user* sebelum dapat masuk atau mengakses beberapa halaman. Dengan memasukkan *username* dan *password* yang sudah terdaftar, maka pemesan / *user* akan bisa masuk untuk mendapatkan akses. Tampilan halaman *Login* dapat dilihat pada gambar 4.20



Gambar 4.20 Tampilan Halaman *Login*

4.1.18. Source Code Tampilan Halaman *Login*

Pada halaman *Login* ini penulis menggunakan beberapa *tag* HTML seperti *tag form* yang berfungsi untuk membuat formulir *login*, *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag label* yang berfungsi untuk memberi label pada *form*, *tag input* yang berfungsi untuk membuat *field* sebagai tempat pemesan / *user* mengisi *form*, *tag button* yang berfungsi untuk membuat tombol agar data yang di isi oleh pemesan / *user* dapat di proses. *Source code* dari tampilan halaman *Login* dapat dilihat pada gambar 4.21

```

<div class="cartOffcanvas offcanvas offcanvas-end" tabindex="-1" id="loginCanvas">
  <div class="cartOffcanvas_header d-flex align-items-center justify-content-between">
    <h2 class="cartOffcanvas_header-title" id="cartOffcanvasLabel" style="color: #765827;">Login

```

Gambar 4.21 Source Code Halaman Login

4.1.19. Tampilan Halaman Registrasi

Halaman Registrasi berfungsi untuk registrasi pemesan / *user* yang belum mempunyai akun, dengan memasukkan nama lengkap, *username*, *email*, nomor hp, alamat, *password*, dan konfirmasi *password* lalu klik tombol registrasi. Setelah melakukan registrasi, pemesan dapat melakukan login. Tampilan halaman Registrasi dapat dilihat pada gambar 4.22

The screenshot shows a registration form titled "Registrasi" in a dark brown header bar. The form consists of several input fields and a "Registrasi" button at the bottom. The fields are labeled as follows:

- Nama Lengkap
- Username
- Email
- No Hp
- Alamat
- Password
- Konfirmasi Password

Each field has a placeholder text inside a rounded rectangular input box. The "Registrasi" button is located at the bottom left of the form area.

Gambar 4.22 Tampilan Halaman Registrasi

4.1.20. *Source Code Halaman Registrasi*

Pada halaman Registrasi ini penulis menggunakan beberapa *tag* HTML seperti *tag form* yang berfungsi untuk membuat formulir *login*, *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag label* yang berfungsi untuk memberi label pada *form*, *tag input* yang berfungsi untuk membuat *field* sebagai tempat pemesan / user mengisi *form*, *tag button* yang berfungsi untuk membuat tombol agar data yang di isi oleh pemesan / user dapat di proses. *Source code* dari tampilan halaman Registrasi dapat dilihat pada gambar 4.23

```

<form action="{{ route('user.store') }}" method="post" id="registrasiForm" autocomplete="off">
  @csrf
  <ul class="cartOffcanvas_body-list">
    <div class="form-group">
      <label for="nama">Nama Lengkap</label> <br>
      <input name="register_nama" class="cart_summary-form_field field required" type="text"
        autocomplete="new-nama" placeholder="Nama Lengkap..." style="margin-top: 10px; width: 100%;"/>
    </div>

    <div class="form-group" style="margin-top: 15px;">
      <label for="username">Username</label> <br>
      <input name="register_username" class="cart_summary-form_field field required" type="text"
        autocomplete="new-username" placeholder="Username..." style="margin-top: 10px; width: 100%;"/>
    </div>

    <div class="form-group" style="margin-top: 15px;">
      <label for="register_email">Email</label> <br>
      <input name="register_email" class="cart_summary-form_field field required" type="email"
        autocomplete="new-email" placeholder="Email..." style="margin-top: 10px; width: 100%;"/>
    </div>

    <div class="form-group" style="margin-top: 15px;">
      <label for="register_no_hp">No Hp</label> <br>
      <input name="register_no_hp" class="cart_summary-form_field field required" type="text"
        autocomplete="new-no-hp" placeholder="Nomor Hp..." style="margin-top: 10px; width: 100%;"/>
    </div>

    <div class="form-group" style="margin-top: 15px;">
      <label for="register_alamat">Alamat</label> <br>
      <input name="register_alamat" class="cart_summary-form_field field required" type="text"
        autocomplete="new-alamat" placeholder="Alamat..." style="margin-top: 10px; width: 100%;"/>
    </div>

    <div class="form-group mt-4">
      <label for="password">Password</label> <br>
      <input name="register_password" class="cart_summary-form_field field required"
        id="konfirmasi_password" type="password" autocomplete="new-password"
        placeholder="Password..." style="margin-top: 10px; width: 100%;"/>
    </div>

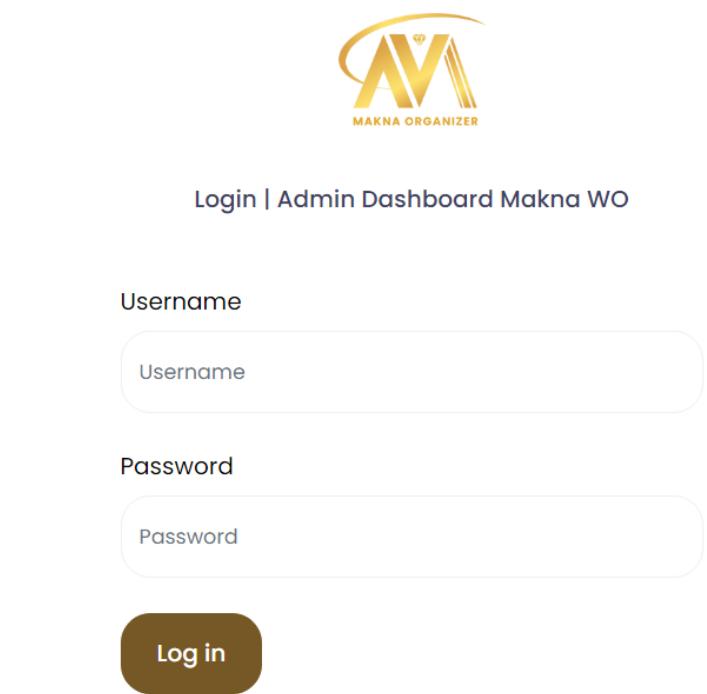
    <div class="form-group mt-4">
      <label for="password">Konfirmasi Password</label> <br>
      <input name="register_konfirmasi_password" class="cart_summary-form_field field required"
        type="password" autocomplete="new-password" placeholder="Password..." style="margin-top: 10px; width: 100%;"/>
    </div>
  </ul>
  <button type="button" class="cartOffcanvas_body-btn btn" id="regisitrasiButton">Registrasi</button>
</form>
</div>

```

Gambar 4.23 Source Code Halaman Registrasi

4.1.21. Tampilan Halaman *Login* Admin

Halaman *Login* Admin merupakan tampilan yang memiliki fungsi sebagai media awal admin sebelum dapat masuk atau mengakses beberapa halaman dan mengelola data. Dengan memasukkan *username* dan *password* yang sudah terdaftar, maka admin akan bisa masuk untuk mendapatkan akses. Tampilan halaman *Login* admin dapat dilihat pada gambar 4.24



Gambar 4.25 Tampilan Halaman *Login* admin

4.1.22. Source Code Halaman *Login* Admin

Pada halaman *Login* Admin ini penulis menggunakan beberapa *tag* HTML seperti *tag form* yang berfungsi untuk membuat formulir *login*, *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag label* yang berfungsi untuk memberi label pada *form*, *tag input* yang berfungsi untuk membuat *field* sebagai tempat pemesan / user mengisi *form*, *tag button* yang berfungsi untuk membuat tombol agar data yang di isi oleh pemesan / user dapat di proses. *Source code* dari tampilan halaman *Login* dapat dilihat pada gambar 4.26

```

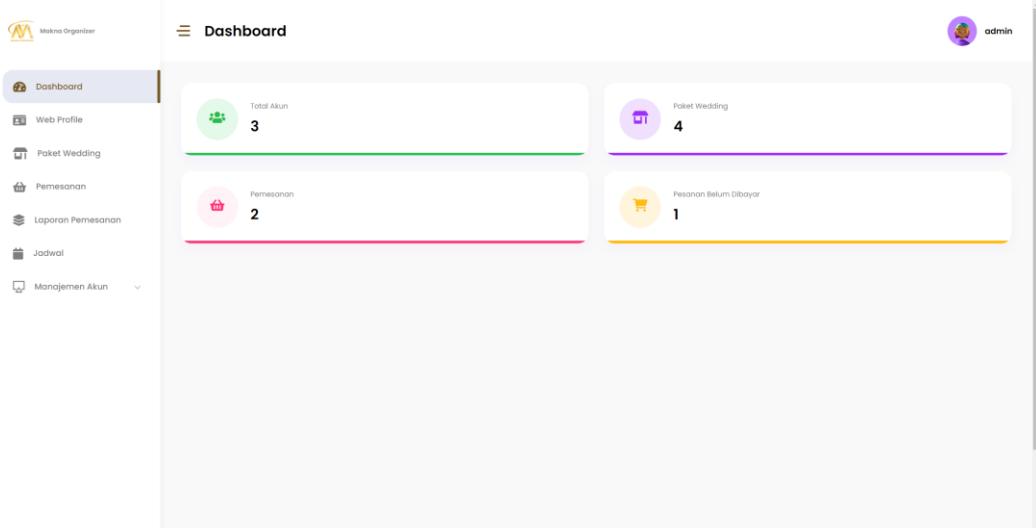
<body class="h-100">
  <section class="h-100 gradient-form" style="background-color: #rgb(248, 248, 248);">
    <div class="container py-5 h-100">
      <div class="row d-flex justify-content-center align-items-center h-100">
        <div class="col-xl-6">
          <div class="card rounded-3 text-black">
            <div class="row g-0">
              <div class="col-lg-12">
                <div class="card-body p-md-5 mx-md-4">
                  <div class="text-center">
                    @foreach($web as $data)
                      
                    @endforeach
                    <br><br>
                    <h5 class="login-heading mt-3">Login | Admin Dashboard Makna WD</h5>
                  </div>
                  <form action="{{ route('admin-login.store') }}" method="POST" class="mt-5">
                    @csrf
                    <div class="form-outline mb-4">
                      <label class="form-label" for="username">Username</label>
                      <input type="text" name="username" class="form-control" placeholder="Username"/>
                    </div>
                    <div class="form-outline mb-4">
                      <label class="form-label" for="password">Password</label>
                      <input type="password" name="password" class="form-control" placeholder="Password" />
                    </div>
                    <div class="d-grid">
                      <button class="btn btn-primary" type="submit">Log in</button>
                    </div>
                  </form>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>

```

Gambar 4.26 Source Code Halaman Login Admin

4.1.23. Tampilan Halaman *Dashboard* Admin

Halaman *Dashboard* Admin merupakan halaman yang akan menampilkan ringkasan dari informasi terkait data yang sudah di *input* kan ke dalam *database*, seperti total akun, paket wedding, pemesanan, pesanan yang belum dibayar. Tampilan halaman *Dashboard* Admin dapat dilihat pada gambar 4.27



Gambar 4.27 Tampilan Halaman *Dashboard* Admin

4.1.24. Source Code Halaman Dashboard Admin

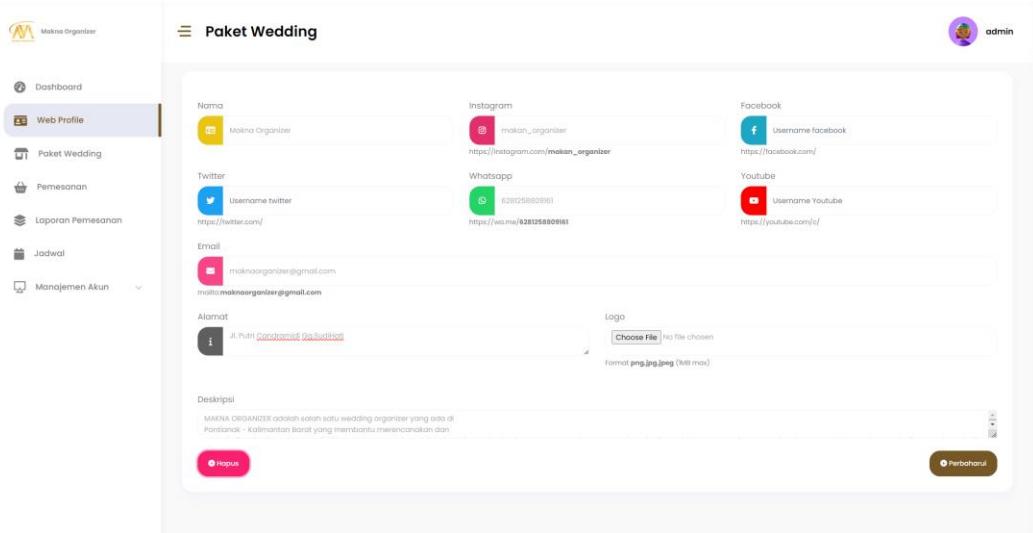
Pada halaman *Dashboard* admin, penulis menggunakan beberapa *tag* HTML seperti, *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag span* yang berfungsi untuk memberi gaya ke *text*. *Source code* halaman *Dashboard Admin* dapat dilihat pada gambar 4.28

```
<div class="row">
    <div class="col-sm-12 col-xl-12 col-xxl-12">
        <div class="row">
            <div class="col-sm-6">
                <div class="card activity-card">
                    <div class="card-body">
                        <div class="media align-items-center">
                            <span class="activity-icon bg-success mr-md-4 mr-3">
                                <i class="fas fa-users" style="color: #27BC48;font-size: 25px;"></i>
                            </span>
                            <div class="media-body">
                                <p class="fs-14 mb-2">Total Akun</p>
                                <span class="title text-black font-w600">{{ $total_akun }}</span>
                            </div>
                        </div>
                        <div class="progress" style="height:5px;">
                            <div class="progress-bar bg-success" style="width: 100%; height:5px;" role="progressbar">
                                <span class="sr-only">42% Complete</span>
                            </div>
                        </div>
                        <div class="effect bg-success" style="top: 104px; left: -28px;"></div>
                    </div>
                </div>
            <div class="col-sm-6">
                <div class="card activity-card">
                    <div class="card-body">
                        <div class="media align-items-center">
                            <span class="activity-icon bgl-secondary mr-md-4 mr-3">
                                <i class="fas fa-store-alt" style="color: #A02CFA;font-size: 25px;"></i>
                            </span>
                            <div class="media-body">
                                <p class="fs-14 mb-2">Paket Wedding</p>
                                <span class="title text-black font-w600">{{ $total_paket_wedding }}</span>
                            </div>
                        </div>
                        <div class="progress" style="height:5px;">
                            <div class="progress-bar bg-secondary" style="width: 100%; height:5px;" role="progressbar">
                                <span class="sr-only">42% Complete</span>
                            </div>
                        </div>
                        <div class="effect bg-secondary" style="top: 104px; left: -28px;"></div>
                    </div>
                </div>
            <div class="col-sm-6">
                <div class="card activity-card">
                    <div class="card-body">
                        <div class="media align-items-center">
```

Gambar 4.28 *Source Code* Halaman *Dashboard Admin*

4.1.25. Tampilan Halaman Kelola *Web Profile*

Halaman Kelola *Web Profile* merupakan halaman yang berfungsi untuk mengedit data terkait informasi *web* yang nanti akan muncul dibagian *footer* pada halaman pemesan / *user*. Pada halaman ini admin dapat mengedit data seperti nama dari *Wedding Organizer*, sosial media, alamat, logo, dan deskripsi. Terdapat juga tombol perbarui yang berfungsi untuk menginputkan data yang sudah diisi oleh admin agar informasi mengenai *web* di perbarui, terdapat juga tombol hapus, yang berfungsi untuk menghapus semua data terkait informasi *web*. Tampilan Halaman Kelola *Web Profile* dapat dilihat pada gambar 4.29



Gambar 4.29 Tampilan Halaman Kelola *Web Profile*

4.1.26. Source Code Halaman Kelola *Web Profile*

Pada halaman Kelola *Web Profile*, penulis menggunakan beberapa *tag* HTML, seperti *tag div* yang berfungsi untuk memanggil *class* dari CSS, *tag label* yang berfungsi untuk memberi label atau keterangan, *tag span* yang berfungsi untuk memberi gaya ke *text*. Potongan *Source Code* Halamaan Kelola *Web Profile* dapat dilihat pada gambar 4.30

```

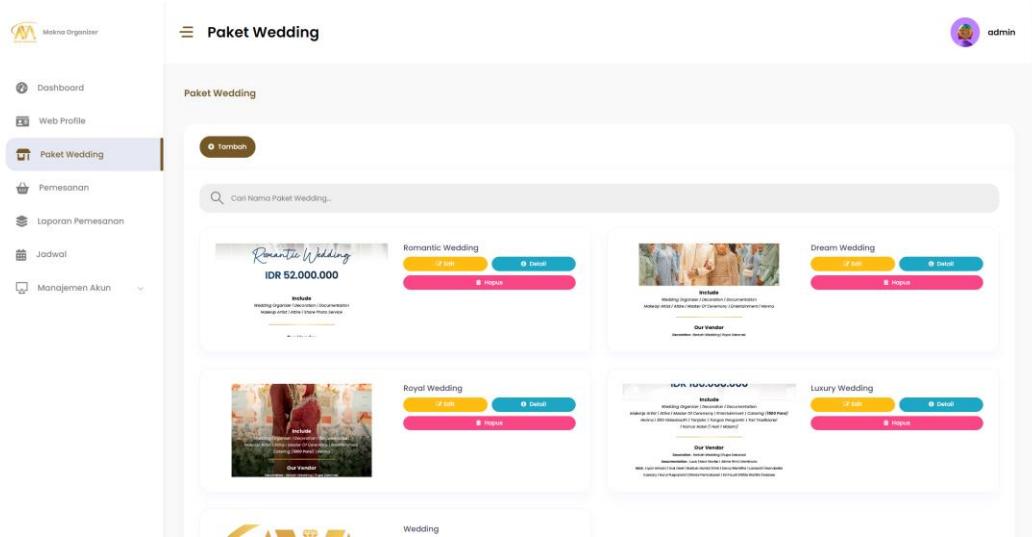
<div class="row">
    <div class="col-sm">
        <label for="name">Nama</label>
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text text-white" style="background-color: #rgb(235, 196, 21)" id="inputGroupPrepend3"><i class="far fa-id-card"></i></span>
            </div>
            <input type="text" id="edit_name" name="edit_name" value="{{old('edit_name') ?? $webs->name}}" class="form-control text-dark" placeholder="Nama">
            <div>
                <label for="edit_name" generated="true" class="error" style="display: none;"></label>
            </div>
        </div>
        <div class="col-sm">
            <label for="instagram">Instagram</label>
            <div class="input-group">
                <div class="input-group-prepend">
                    <span class="input-group-text text-white" style="background-color: #rgb(225,48,108)" id="inputGroupPrepend3"><i class="fab fa-instagram"></i></span>
                </div>
                <input type="text" id="edit_instagram" name="edit_instagram" value="{{old('edit_instagram') ?? $webs->instagram}}" class="form-control text-dark" placeholder="Username Instagram">
                <div>
                    <small>https://instagram.com/<b>{{old('edit_instagram') ?? $webs->instagram}}</b></small>
                </div>
            </div>
            <div class="col-sm">
                <label for="facebook">Facebook</label>
                <div class="input-group">
                    <div class="input-group-prepend">
                        <span class="input-group-text bg-info text-white" id="inputGroupPrepend3"><i class="fab fa-facebook-f"></i></span>
                    </div>
                    <input type="text" id="facebook" onkeyup="keyup('facebook')" name="edit_facebook" class="form-control text-dark" value="{{old('edit_facebook') ?? $webs->facebook}}" placeholder="Username Facebook">
                    <div>
                        <small>https://facebook.com/<b>{{old('edit_facebook') ?? $webs->facebook}}</b></small>
                    </div>
                </div>
                <br>
            </div>
            <div class="row">
                <div class="col-4">
                    <label for="twitter">Twitter</label>
                    <div class="input-group">
                        <div class="input-group-prepend">
                            <span class="input-group-text text-white" style="background-color: #rgb(29, 161, 242)" id="inputGroupPrepend3"><i class="fab fa-twitter"></i></span>
                        </div>
                        <input type="text" id="twitter" onkeyup="keyup('twitter')" name="edit_twitter" value="{{old('edit_twitter') ?? $webs->twitter}}" class="form-control text-dark" placeholder="Username Twitter">
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Gambar 4.30 Source Code Halaman Kelola Web Profile

4.1.27. Tampilan Halaman Kelola Paket Wedding

Halaman Kelola Paket *Wedding* merupakan halaman yang berfungsi untuk menambah, mengedit, dan menghapus data paket *wedding*. Terdapat tombol tambah yang apabila ditekan muncul tampilan modal *form* untuk menambahkan paket *wedding* dengan memasukkan nama paket, harga paket, deskripsi paket, dan foto paket, lalu tombol edit untuk mengedit data paket *wedding* yang dipilih, tombol *detail* untuk melihat *detail* dari paket yang dipilih, tombol hapus untuk menghapus paket *wedding* yang dipilih. Tampilan Halaman Kelola Paket *Wedding* dapat dilihat pada gambar 4.31 sampai gambar 4.34



Gambar 4.31 Tampilan Halaman Kelola Paket *Wedding*

Tambah Paket Wedding

Nama Paket *

Harga Paket *

Deskripsi Paket *

Foto Paket *

Upload Foto Paket Wedding

[Tutup](#)
Submit

Gambar 4.32 Tampilan Halaman Kelola Paket *Wedding* -form tambah paket

Edit Paket Wedding

Nama Paket *

Romantic Wedding

Harga Paket *

52000000

Deskripsi Paket *

Wedding Organizer | Decoration | Documentation
MakeUp Artist | Attire | Share Photo Service

Our Vendor
Decoration : Rupa Dekorasi
Documentation : Frame Like | Moonlight
MUA : Devvy Rismitha | Larasati | ENA | Nurul

Foto Paket *

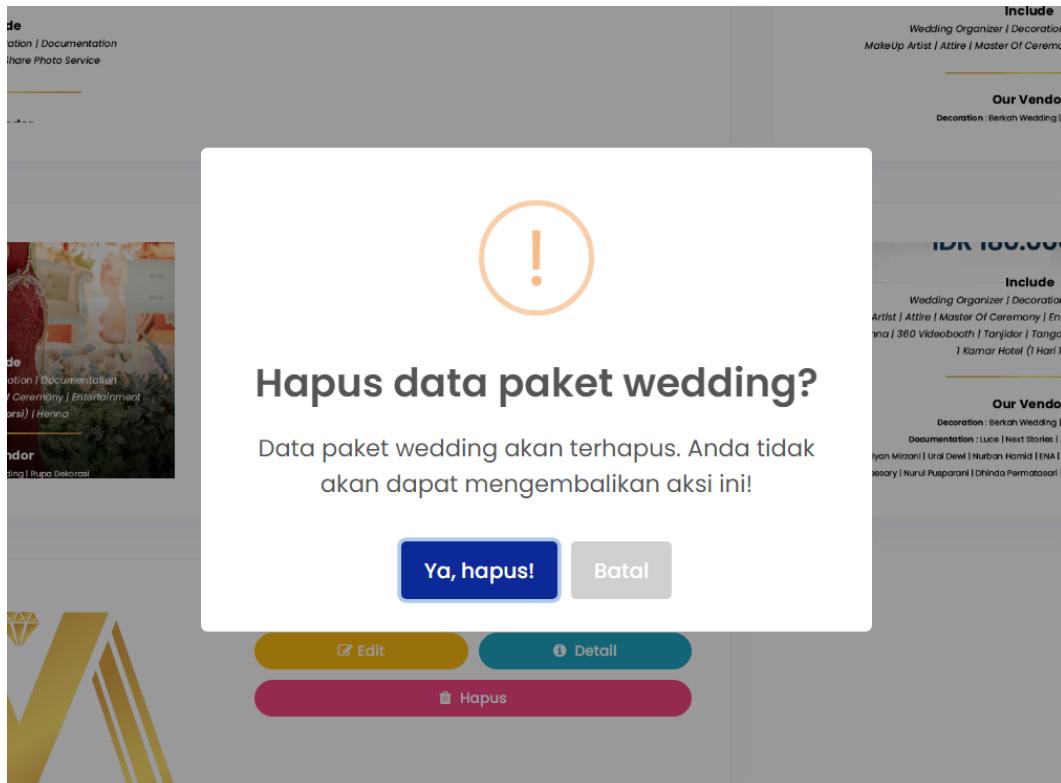


Hapus Foto

Tutup

Simpan Perubahan

Gambar 4.33 Tampilan Halaman Kelola Paket *Wedding* -form edit paket



Gambar 4.34 Tampilan Halaman Kelola Paket Wedding -hapus paket

4.1.28. *Source Code Halaman Kelola Paket Wedding*

Pada halaman Kelola Paket *Wedding*, penulis menggunakan beberapa *tag* HTML, seperti *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag img* yang berfungsi untuk menampung dan membaca data berupa gambar, *tag button* yang berfungsi untuk membuat tombol sebagai konfirmasi tindakan. Untuk penginputan data ke *database* menggunakan *crud ajax*. *Source code* halaman Kelola Paket *Wedding* dapat dilihat pada gambar 4.34 Sampai gambar 4.35

```

<div class="row" id="wrapVerified">
    <?php foreach($paket_wedding as $data): ?>
    <div class="col-sm-6">
        <div class="card" style="border: 1px solid #f0f1f5">
            <div class="card-body">
                <div class="row">
                    <div class="col-6">
                        @foreach($data->get_first_foto as $foto)
                            <?php if(!empty($foto->url)) : ?>
                            
                            <?php else : ?>
                            
                            <?php endif; ?>
                        @endforeach
                    </div>
                    <div class="col-6">
                        <h6><?= $data->nama_paket ?></h6>
                        <div class="button-list">
                            <div class="buttons d-flex justify-content-between">
                                <button type="button" data-toggle="modal" style="width: 100%;" data-target="#editPaketWeddingModal"
                                    onclick="editPaketWeddingFunction('{{ $data }})" class="btn btn-warning btn-xxs text-white mr-2"><i class="fa fa-edit mr-1"></i> Edit
                                </button>
                                <button type="button" style="width: 100%;" onclick="detailPaketWeddingFunction('{{ $data }})" data-toggle="modal" data-target="#detailPaketWeddingModal" class="btn btn-xxs btn-info" id="detailButton"><i class="fa fa-info-circle mr-1"></i> Detail
                                </button>
                            </div>
                            <form action="{{ route('paket-wedding.destroy', $data->id) }}" method="post">
                                @csrf
                                @method('DELETE')
                                <button type="button" style="width: 100% !important;" onclick="deletePaketWeddingFunction(this)" class="btn btn-xxs btn-danger mt-2" id="deleteButton"><i class="fa fa-trash mr-1"></i> Hapus</button>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

Gambar 4.34 Source Code Halaman Kelola Paket Wedding -tampilan

```

<script>
var uploadedDocumentMap = {}
Dropzone.options.documentDropzone = {
  url: `({ route('paket-wedding.store') })`,
  maxFilesize: 20, // MB
  addRemovelinks: true,
  autoProcessQueue: false,
  parallelUploads: 10,
  autoDiscover: false,
  uploadMultiple: true,
  acceptedFiles: ".jpeg,.jpg,.png,.gif",
  processData: false,
  contentType: false,
  dictDefaultMessage: "Upload Foto Paket Wedding",
  headers: {
    'X-CSRF-TOKEN': "{{ csrf_token() }}"
  },
  success: function(file, response) {
    Swal.fire({
      title: 'Berhasil',
      text: "Data Paket Wedding telah berhasil ditambahkan",
      icon: 'success',
      confirmButtonColor: '#7066E0',
      confirmButtonText: 'Ok'
    }).then(result => {
      if (result.isConfirmed) {
        location.reload();
      } else {
        location.reload();
      }
    })
  },
  init: function() {
    var myDropzone = this;

    $("#addPaketWeddingbutton").click(function() {
      if (myDropzone.getQueuedFiles().length > 0) {
        $(".dropzone").css("cssText",
          "border: 1px solid #e6e6e6 !important; color: #c8c8c9 !important;");
        $(".errorAddImage").css("cssText", "display: none !important");
      } else {
        $(".dropzone").css("cssText",
          "border: 1px solid #f1556c !important; color: #f1556c !important;");
        $(".errorAddImage").css("cssText",
          "display: block !important; color: #f1556c; font-size: .875rem; font-weight: 400; line-height: 1.5; margin-top: 5px; padding: 0;");
      }
    });
    if (myDropzone.getQueuedFiles().length > 0 && $('#addPaketweddingForm').valid()) {
  
```

Gambar 4.35 Source Code Halaman Kelola Paket Wedding -CRUD ajax

4.1.29. Tampilan Halaman Kelola Pemesanan

Halaman Kelola Pemesanan merupakan halaman yang berfungsi konfirmasi pemesanan dari pesanan. Terdapat daftar data pesanan seperti nama pemesan, nama paket yang dipilih, jumlah paket, status pembayaran, dan aksi. Terdapat tombol detail yang berfungsi untuk menampilkan detail dari pesanan, tombol edit untuk mengedit data pesanan, dan tombol hapus untuk menghapus data pesanan. Tampilan halaman Kelola Pemesanan dapat dilihat pada gambar 4.36 Sampai gambar 4.39

| # | Pemesan | Nama Paket | Jumlah Paket | Status Pembayaran | Aksi |
|---|---------|------------------|--------------|-------------------|--|
| 1 | apip | Luxury Wedding | 6 | Sudah | <button>Detail</button> <button>Edit</button> <button>Hapus</button> |
| 2 | rahmat | Romantic Wedding | 1 | Belum | <button>Detail</button> <button>Edit</button> <button>Hapus</button> |

Showing 1 to 2 of 2 entries

Gambar 4.36 Tampilan Halaman Kelola Pemesanan

Pemesan
apip

Nama Paket
Luxury Wedding

Jumlah Paket
6

Tempat Acara
PCC

Tanggal Acara
2023-08-21

Total Harga
Rp. 1080000000

Status Pembayaran
sudah

Tutup

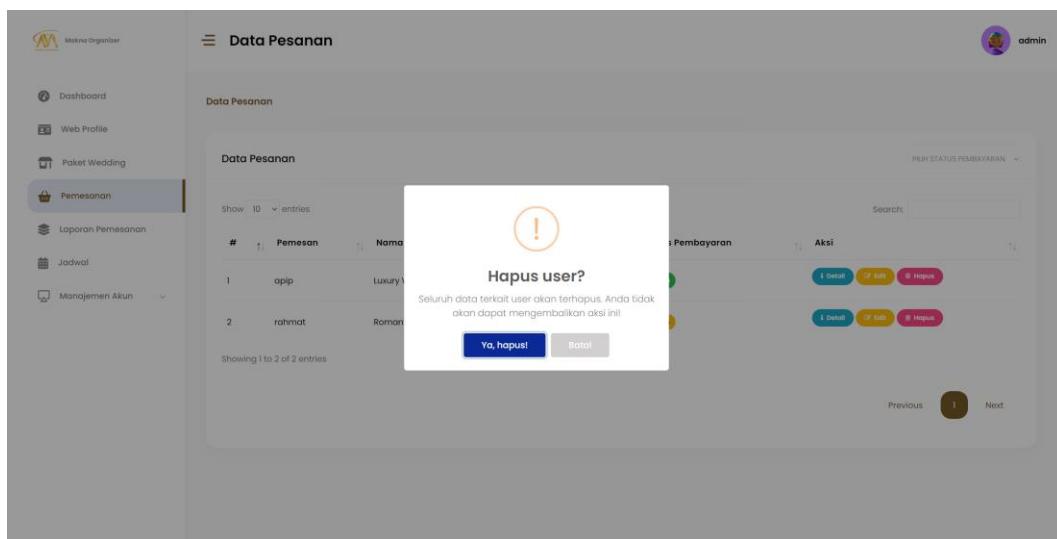
Gambar 4.37 Tampilan Halaman Kelola Pemesanan -detail pemesanan

Pemesan
apip

Status Pembayaran
Sudah

Tutup **Simpan Perubahan**

Gambar 4.38 Tampilan Halaman Kelola Pemesanan -form edit pemesanan



Gambar 4.39 Tampilan Halaman Kelola Pemesanan -hapus pesanan

4.1.30. Source Code Halaman Kelola Pemesanan

Pada Halaman Kelola Pemesanan ini, penulis menggunakan beberapa *tag* HTML dalam membuat tampilannya seperti, *tag table* yang berfungsi untuk membuat tabel dan menampung data yang dimana didalam *tag table* terkait dengan *tag thead*, *tag tr*, *tag th*, *tag tbody*, *tag td*, terdapat *tag div* yang berfungsi untuk memanggil *class* dari *class CSS*, *tag button* yang berfungsi untuk membuat tombol.

Pada halaman ini juga menggunakan *CRUD ajax* dalam mengedit data. *Source code* Halaman kelola pemesanan dapat dilihat pada gambar 4.40 Sampai gambar 4.41

```
<table id="data_pesanan_table" class="table dt-responsive" style="width:100%">
    <thead>
        <tr>
            <th>#</th>
            <th>Pemesan</th>
            <th>Nama Paket</th>
            <th>Jumlah Paket</th>
            <th>Status Pembayaran</th>
            <th>Aksi</th>
        </tr>
    </thead>

    <tbody>

        @php
        $increment = 1;
        @endphp

        @foreach($data_pesanan as $data)
        <tr>
            <td>{{ $increment++ }}</td>
            <td>{{ $data->users->nama }}</td>
            <td>{{ $data->paket_wedding->nama_paket }}</td>
            <td>{{ $data->jumlah_paket }}</td>
            <td>
                @php
                    $color = 'secondary';
                    if($data->status_pembayaran == 'belum'){
                        $color = 'warning';
                    }elseif ($data->status_pembayaran == 'sudah') {
                        $color = 'success';
                    }
                @endphp
                <span class="badge badge-{{ $color }} font-15">{{ ucwords($data->status_pembayaran) }}</span>
            </td>
            <td>
                <div class="form-group">
                    <button type="button" data-toggle="modal" data-target="#detailModal"
                        data-ids="{{ $data->id }}" data-role="{{ $data->role }}"
                        onclick="detailData('{{ $data }}')"
                        class="btn btn-info btn-xs text-white"><i class="fa fa-info mr-1"></i> Detail</button>
                    <button type="button" data-toggle="modal" data-target="#editModal"
                        data-ids="{{ $data->id }}" data-role="{{ $data->role }}"
                        onclick="setEditData('{{ $data }}')"
                        class="btn btn-warning btn-xs text-white"><i class="fa fa-edit mr-1"></i> Edit</button>
                    <form style="display: inline" action="{{ route('manajemen-akun.destroy', $data) }}"
                        method="post">
                        @csrf
                        @method('delete')
                        <button type="button"
                            class="btn btn-danger btn-xs rounded waves-light waves-effect"
                            onclick="deleteAlert(this)"><i class="fa fa-trash-o mr-1"></i> Hapus
                        </button>
                    </form>
                </div>
            </td>
        @endforeach
    </tbody>
</table>
```

Gambar 4.40 *Source Code* Membuat Halaman Kelola Pemesanan

```

<script>
    $("#pilihStatusPembayaran").change(function() {
        var status_pembayaran = $("#pilihStatusPembayaran").val();

        $.ajax({
            url: "{{ route('data-pesanan.select') }}",
            method: "POST",
            data: {
                "_token": "{{ csrf_token() }}",
                status_pembayaran: status_pembayaran
            },
            success: function(data) {
                $('#wrap').html(data);
                $('#data_pesanan_table').DataTable();
            }
        });
    });
</script>

{{-- Validate Modal Edit User --}}

```

Gambar 4.41 *Source Code* Halaman Kelola Pemesanan -CRUD ajax

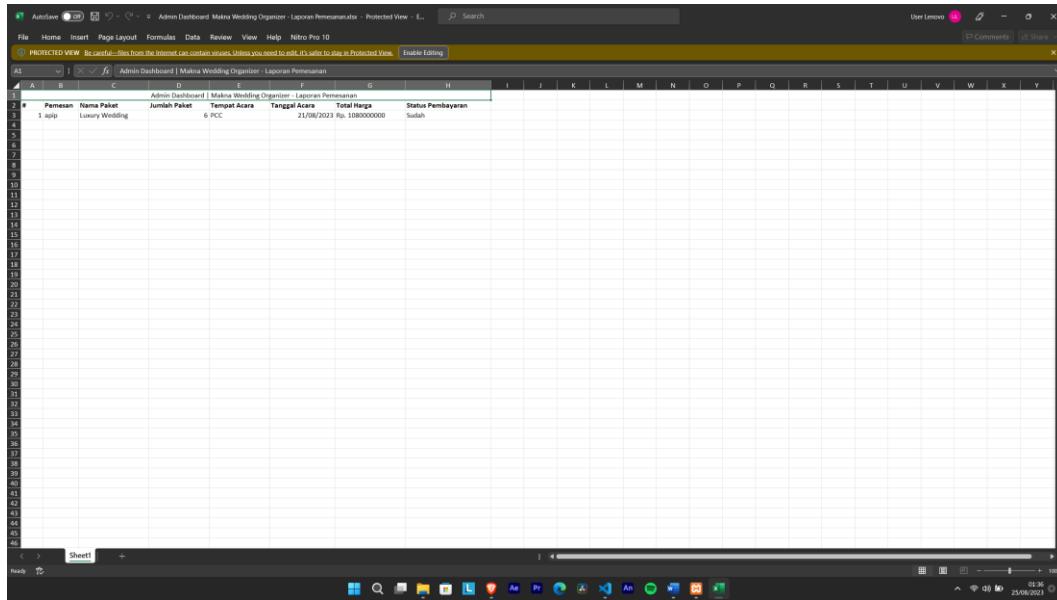
4.1.31. Tampilan Halaman Kelola Laporan Pemesanan

Halaman Kelola Laporan Pemesanan merupakan halaman yang berfungsi untuk menampilkan detail berupa data pemesanan yang sudah di konfirmasi sebelumnya di halaman kelola pemesanan. Pada halaman ini akan menampilkan informasi seperti nama pemesan, nama paket, jumlah paket, tempat acara, tanggal acara, total harga, dan status pembayaran. Terdapat tombol *excel* yang berfungsi mengekspor data menjadi file *excel* atau data berformat *.xlsx*. Tampilan halaman Kelola Laporan Pemesanan dapat dilihat pada gambar 4.42 Sampai gambar 4.44

| # | Pesanan | Nama Paket | Jumlah Paket | Tempat Acara | Tanggal Acara | Total Harga | Status Pembayaran |
|---|---------|------------------|--------------|--------------|---------------|---------------|---|
| 2 | rahmat | Romantic Wedding | 1 | Audit Untan | 2023-08-25 | Rp. 52000000 | Belum |
| 1 | apip | Luxury Wedding | 6 | PCC | 2023-08-21 | Rp. 108000000 | Sudah |

Gambar 4.42 Tampilan Halaman Kelola Laporan Pemesanan

Gambar 4.43 Tampilan Halaman Kelola Laporan Pemesanan -ekspor menjadi excel



| Pemesan | Nama Paket | Jumlah Paket | Tempat Acara | Tanggal Acara | Total Harga | Status Pembayaran |
|---------|----------------|--------------|--------------|---------------|--------------|-------------------|
| 1 appr | Luxury Wedding | 0 Pcs. | | 21/08/2023 | Rp. 10000000 | Sudah |

Gambar 4.44 Tampilan Halaman Kelola Laporan Pemesanan -file .xlsx yang sudah di ekspor

4.1.31. *Source Code* Halaman Kelola Laporan Pemesanan

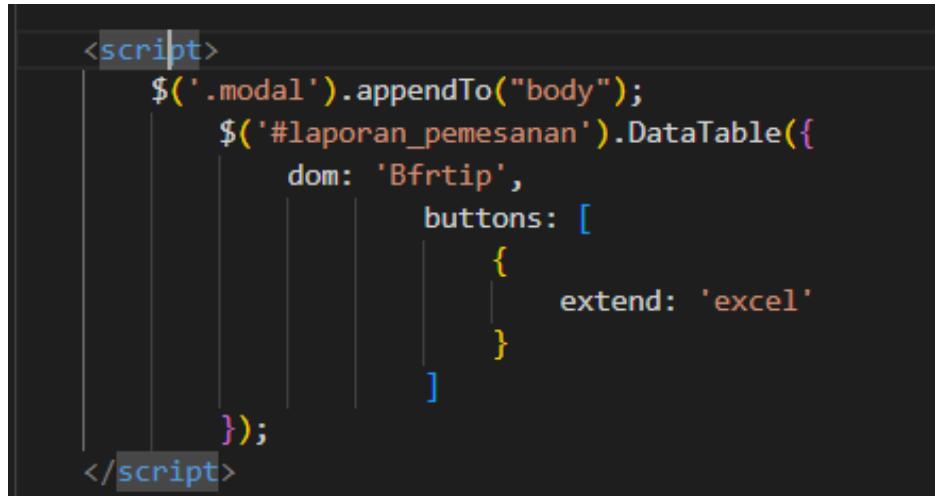
Pada Halaman Kelola Laporan Pemesanan ini, penulis menggunakan beberapa *tag* HTML dalam membuat tampilannya seperti, *tag table* yang berfungsi untuk membuat tabel dan menampung data yang dimana didalam *tag table* terkait dengan *tag thead*, *tag tr*; *tag th*, *tag tbody*, *tag td*, terdapat *tag div* yang berfungsi untuk memanggil *class* dari *class CSS*, *tag button* yang berfungsi untuk membuat tombol. Pada halaman ini untuk membuat eksport menjadi *excel* atau file .xlsx penulis menggunakan library *javascript* yaitu *pdfmake*. *Source code* Halaman kelola Laporan Pemesanan dapat dilihat pada gambar 4.45 Sampai gambar 4.46

```


| #                   | Pemesan                   | Nama Paket                                                                                                                                                                                                                                                                                                            | Jumlah Paket                                                                                                                                                                                                        | Tempat Acara               | Tanggal Acara               | Total Harga                                                                                                                                                                                                                                                                                             | Status Pembayaran                                                                                                                                                                                                                                                                 |
|---------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ \$increment++ }} | {{ \$data->users->nama }} | @php \$pemesanan = \App\Models\Pemesanan::where('user_id', \$data->user_id)->get();  foreach(\$pemesanan as \$item) {     \$data_paket = \App\Models\PaketWedding::where('id', \$item->paket_wedding_id)->first();     \$data_nama_paket[] = \$data_paket->nama_paket; } echo implode(",",\$data_nama_paket); @endphp | @php \$jumlah = \App\Models\Pemesanan::where('user_id', \$data->user_id)->get();  foreach(\$jumlah as \$item) {     \$data_jumlah_paket[] = \$item->jumlah_paket; }  echo implode(",",\$data_jumlah_paket); @endphp | {{ \$data->tempat_acara }} | {{ \$data->tanggal_acara }} | @php \$pemesanan = \App\Models\Pemesanan::where('user_id', \$data->user_id)->get();  \$jumlah = 0;  foreach(\$pemesanan as \$item) {     // \$data_jumlah_paket[] = \$item->jumlah_paket;     \$jumlah += \$item->jumlah_paket * \$item->paket_wedding->harga_paket; }  echo "Rp. " . \$jumlah; @endphp | @php \$color = 'secondary'; if(\$data->status_pembayaran == 'belum'){     \$color = 'warning'; } elseif (\$data->status_pembayaran == 'sudah') {     \$color = 'success'; } @endphp <span class="badge badge-{{\$color}} font-15">{{ ucwords(\$data->status_pembayaran) }}</span> |


```

Gambar 4.45 Source Code Halaman Kelola Laporan Pemesanan

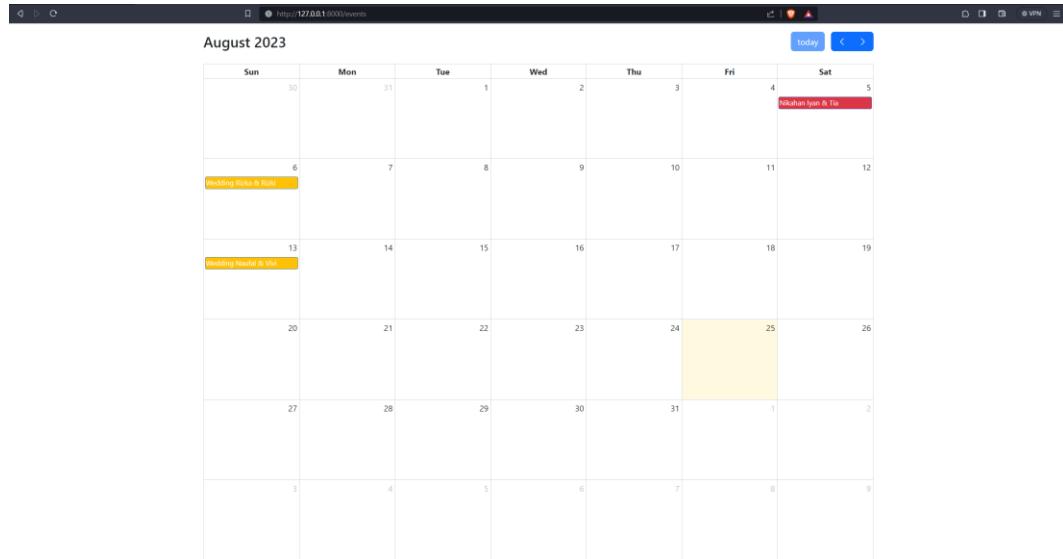


```
<script>
    $('.modal').appendTo("body");
    $('#laporan_pemesanan').DataTable({
        dom: 'Bfrtip',
        buttons: [
            {
                extend: 'excel'
            }
        ]
    });
</script>
```

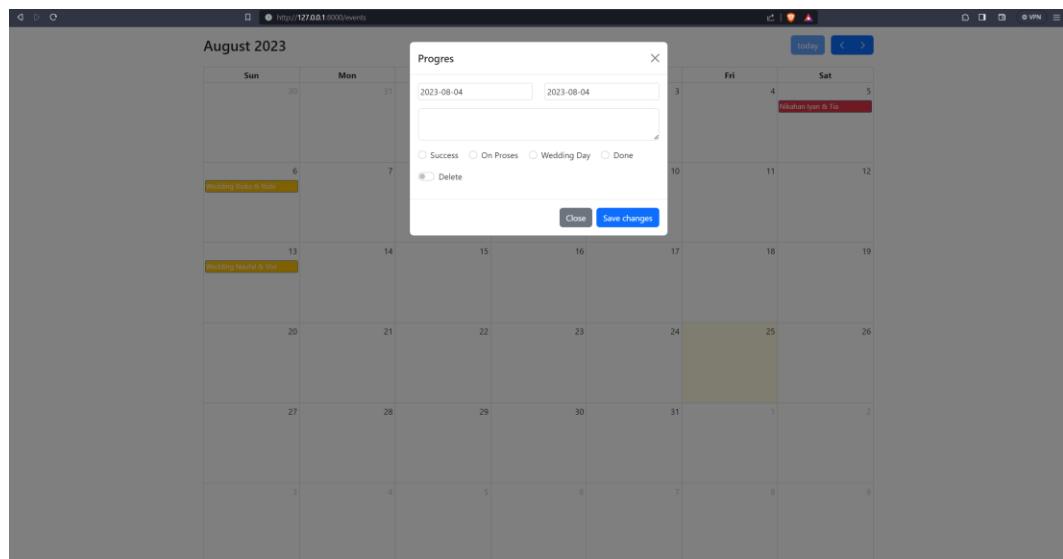
Gambar 4.46 *Source Code* Halaman Kelola Laporan Pemesanan -*script* ekspor ke *excel* atau file *.xlsx*

4.1.32. Tampilan Halaman Kelola Jadwal

Halaman Kelola Jadwal merupakan halaman yang berfungsi untuk menambah, mengedit, menghapus data jadwal. Terdapat kolom jadwal berupa kalender yang berisi tanggal, dan hari, admin dapat melakukan penginputan data dengan cara menekan kotak kalender maka akan muncul *form* modal yang berisi tanggal mulai, tanggal selesai, deskripsi, pilihan aksi, tombol delete yang berfungsi untuk menghapus data, terdapat dua tombol didalam modal *form*, tombol *close* untuk melakukan batal mengedit data, dan tombol *save changes* yang berfungsi menyimpan data yang sudah dimasukkan. Tampilan Halaman Kelola Jadwal dapat dilihat pada gambar 4.47 Sampai gambar 4.48



Gambar 4.47 Tampilan Halaman Kelola Jadwal



Gambar 4.48 Tampilan Halaman Kelola Jadwal -form input data

4.1.33. *Source Code* Halaman Kelola Jadwal

Pada Halaman Kelola Jadwal, penulis menggunakan *library javascript* dan *bootstrap* berupa *datepicker* dan *fullcalendar* untuk membuat tampilan serta *input* data menggunakan *CRUD ajax*. Dalam pembuatan *form input* data penulis menggunakan beberapa *tag* HTML seperti, *tag div* yang berfungsi untuk memanggil *class* dari *CSS*, *tag input* untuk membuat *field* agar admin dapat memasukkan data. *Source code* Halaman Kelola Jadwal dapat dilihat pada gambar 4.49 Sampai gambar 4.50

```

<script>
    const modal = $('#modal-action')
    const csrfToken = $('meta[name=csrf_token]').attr('content')

    document.addEventListener('DOMContentLoaded', function() {
        var calendarEl = document.getElementById('calendar');
        var calendar = new FullCalendar.Calendar(calendarEl, {
            initialView: 'dayGridMonth',
            themeSystem: 'bootstrap5',
            events: `{{ route('events.list') }}`,
            editable: true,
            dateClick: function (info) {
                $.ajax({
                    url: `{{ route('events.create') }}`,
                    data: {
                        start_date: info.dateStr,
                        end_date: info.dateStr
                    },
                    success: function (res) {
                        modal.html(res).modal('show')
                        $('.datepicker').datepicker({
                            todayHighlight: true,
                            format: 'yyyy-mm-dd'
                        });
                    }
                });

                $('#form-action').on('submit', function(e) {
                    e.preventDefault()
                    const form = this
                    const formData = new FormData(form)
                    $.ajax({
                        url: form.action,
                        method: form.method,
                        data: formData,
                        processData: false,
                        contentType: false,
                        success: function (res) {
                            modal.modal('hide')
                            calendar.refetchEvents()
                        },
                        error: function (res) {
                            console.log(res)
                        }
                    })
                })
            }
        });
        eventClick: function ({event}) {
            $.ajax({
                url: `{{ url('events') }}/${event.id}/edit`,
                success: function (res) {
                    modal.html(res).modal('show')

                    $('#form-action').on('submit', function(e) {
                        e.preventDefault()
                        const form = this
                        const formData = new FormData(form)
                        $.ajax({
                            url: form.action,
                            method: form.method,
                            data: formData,
                            processData: false,
                            contentType: false,
                            success: function (res) {
                                modal.html(res).modal('show')
                            }
                        })
                    })
                }
            })
        }
    })
}

```

Gambar 4.49 Source Code Halaman Kelola Jadwal

```

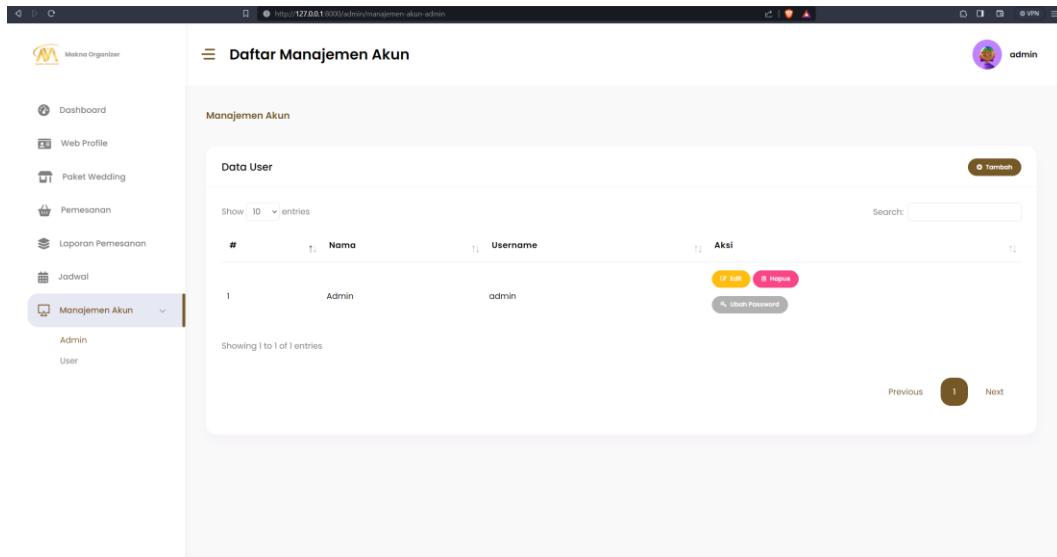
<x-modal-action action="{{ $action }}>
@if ($data->>id)
    @method('put')
@endif
</x-modal-action>
<div class="row">
    <div class="col-6">
        <div class="mb-3">
            <input type="text" name="start_date" readonly value="{{ $data->start_date ?? request()->start_date }}" class="form-control datepicker">
        </div>
    </div>
    <div class="col-6">
        <div class="mb-3">
            <input type="text" name="end_date" readonly value="{{ $data->end_date ?? request()->end_date }}" class="form-control datepicker">
        </div>
    </div>
    <div class="col-12">
        <div class="mb-3">
            <textarea name="title" class="form-control">{{ $data->title }}SuccessOn ProsesWedding DayDoneDelete

```

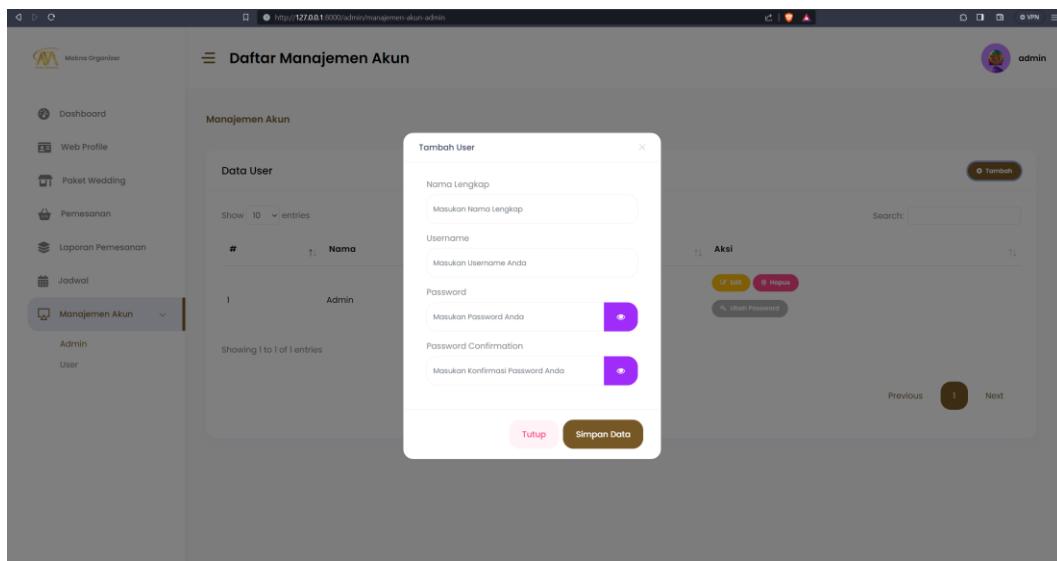
Gambar 4.50 Source Code Halaman Kelola Jadwal -*form input* data

4.1.34. Tampilan Halaman Kelola Manajemen Akun

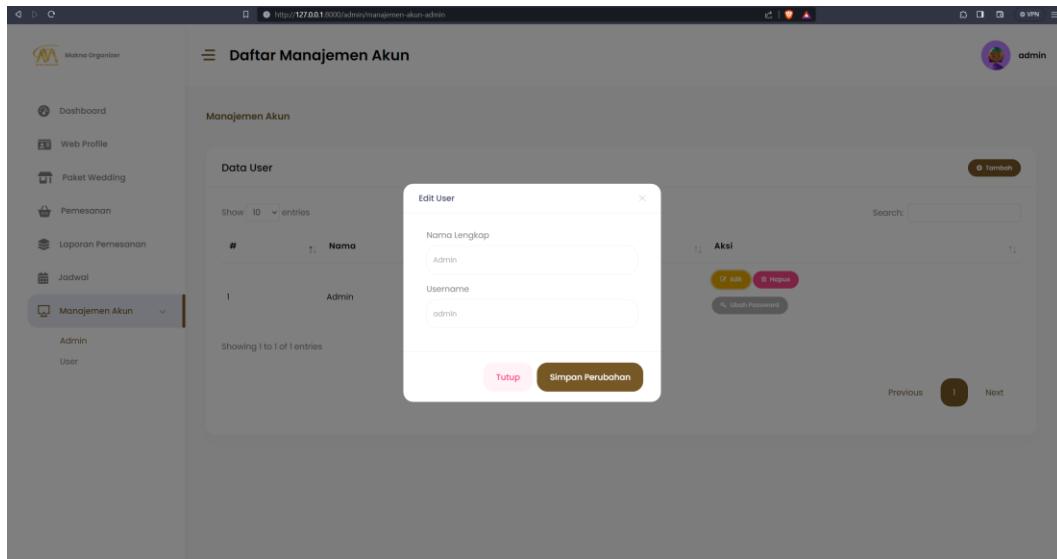
Halaman Kelola Manajemen Akun merupakan halaman yang berfungsi menambah, mengedit, menghapus, merubah *password*, terdapat dua pilihan untuk mengelola akun yaitu, admin dan *user*. Terdapat tombol tambah yang apabila ditekan akan muncul modal *form* untuk membuat akun, tombol *edit* untuk mengedit nama dan *username* akun, tombol hapus untuk menghapus akun, tombol ubah *password* untuk merubah *password* akun. Tampilan Halaman Kelola Manajemen Akun dapat dilihat pada gambar 4.51 Sampai gambar 4.55



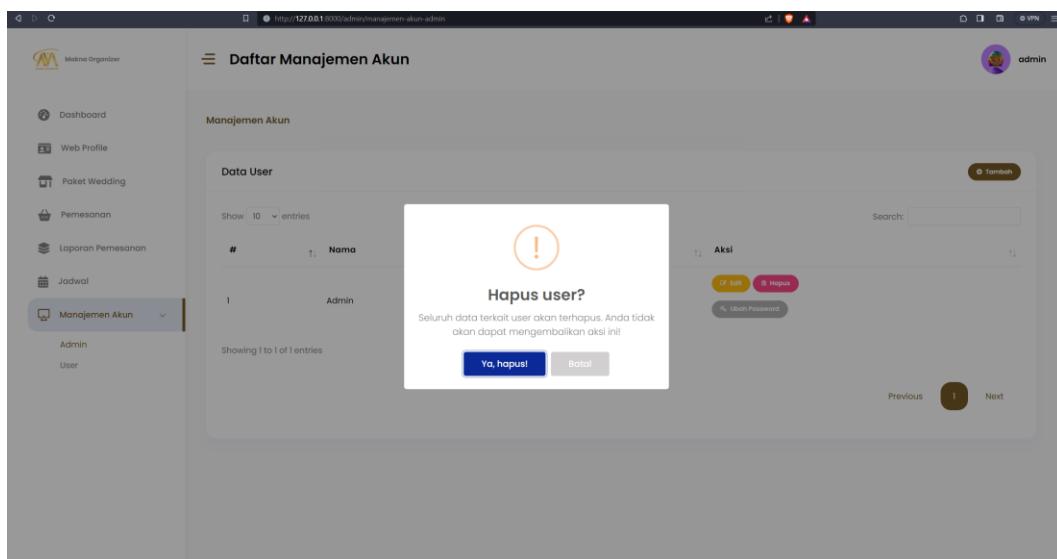
Gambar 4.51 Tampilan Halaman Kelola Manajemen Akun



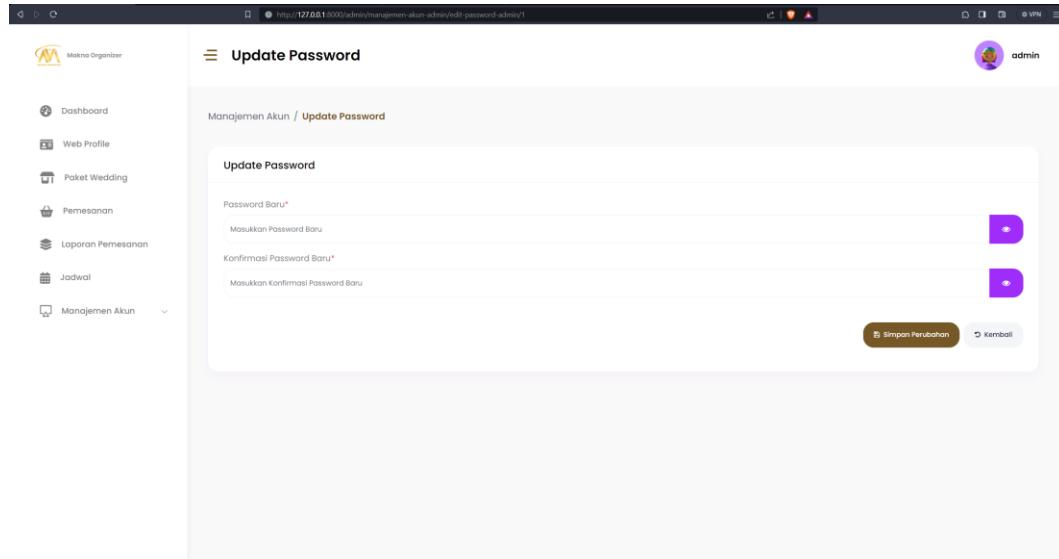
Gambar 4.52 Tampilan Halaman Kelola Manajemen Akun -form tambah akun



Gambar 4.53 Tampilan Halaman Kelola Manajemen Akun -form edit akun



Gambar 4.54 Tampilan Halaman Kelola Manajemen Akun -hapus akun



Gambar 4.55 Tampilan Halaman Kelola Manajemen Akun -ubah password

4.1.35. *Source Code Halaman Kelola Manajemen Akun*

Pada Halaman Kelola Manajemen Akun, penulis menggunakan beberapa *tag* HTML untuk membuat tampilan halaman dan *form*, seperti *tag table* yang berfungsi untuk membuat tabel dan menampung data yang dimana didalam *tag table* terkait dengan *tag thead*, *tag tr*; *tag th*, *tag tbody*, *tag td*, terdapat *tag div* yang berfungsi untuk memanggil *class* dari *class CSS*, *tag button* yang berfungsi untuk membuat tombol, *tag form* untuk membuat *form* modal. *Source code* Halaman Kelola Manajemen Akun dapat dilihat pada gambar 4.56 Sampai gambar 4.59

```

</div>
<div class="card-body">
    {{-- <div class="table-responsive"> --}}
    <table id="example4" class="table dt-responsive" style="width:100%">
        <thead>
            <tr>
                <th>#</th>
                <th>>Nama</th>
                <th>Username</th>
                <th>Email</th>
                <th>Nomor Hp</th>
                <th>Alamat</th>
                <th>Aksi</th>
            </tr>
        </thead>
        <tbody>
            @php
            $increment = 1;
            @endphp
            @foreach($users as $user)
            <tr>
                <td>{{ $increment++ }}</td>
                <td>{{ $user->nama }}<br>
                    @if(Cache::has('user-is-online-' . $user->id))
                        <span style="color: #28B62C;" class="ml-2">online</span>
                        <span class="indicator online"></span>
                    @endif
                </td>
                <td>{{ $user->username }}</td>
                <td>{{ $user->email }}</td>
                <td>{{ $user->no_hp }}</td>
                <td>{{ $user->alamat }}</td>
                <td>
                    <div class="form-group">
                        <button type="button" data-toggle="modal" data-target="#editUserModal"
                            data-id="{{ $user->id }}"
                            onclick="setEditData('{{ $user }}')"
                            class="btn btn-warning btn-xs text-white"><i class="fa fa-edit mr-1"></i> Edit</button>
                        <form style="display: inline" action="{{ route('manajemen-akun.destroy', $user) }}"
                            method="post">
                            @csrf
                            @method('delete')
                            <button type="button"
                                class="btn btn-danger btn-xs rounded waves-light waves-effect"
                                onclick="deleteAlert(this)"><i class="fa fa-trash-o mr-1"></i> Hapus
                            </button>
                        </form>
                    </div>
                    <div class="form-group">
                        <a href="{{ route('edit_password', $user->id) }}"
                            class="btn btn-dark btn-xs waves-effect waves-light" id="update-password"><i class="fa fa-key mr-1"></i> Ubah
                            Password
                        </a>
                    </div>
                </td>
            </tr>
            @endforeach
        </tbody>
    </table>

```

Gambar 4.56 Source Code Halaman Kelola Manajemen Akun

```

<div class="modal fade" id="addUserModal">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Tambah User</h5>
                <button type="button" class="close" data-dismiss="modal">&times;</span>
            </button>
        </div>
        <div class="modal-body">
            <form class="form-horizontal" action="{{ route('manajemen-akun.store') }}" method="post" id="tambahUs
                @csrf
                <input type="hidden" name="_token" value="{{ csrf_token() }}>

                <div class="form-group">
                    <div class="col-12">
                        <label for="election">Nama Lengkap</label>
                        <input class="form-control mb-1 @error('nama') is-invalid @enderror" type="text" id="nama
                            placeholder="Masukan Nama Lengkap" name="nama" value="{{ old('nama') }}>
                    </div>
                </div>

                <div class="form-group">
                    <div class="col-12">
                        <label for="period">Username</label>
                        <input class="form-control mb-1 @error('username') is-invalid @enderror" type="text"
                            id="username" placeholder="Masukan Username Anda" name="username"
                            value="{{ old('username') }}>
                    </div>
                </div>

                <div class="form-group">
                    <div class="col-12">
                        <label for="period">Email</label>
                        <input class="form-control mb-1 @error('email') is-invalid @enderror" type="text"
                            maxlength="35" type="email" id="email" placeholder="Masukan Email Anda"
                            name="email" value="{{ old('email') }}>
                    </div>
                </div>

                <div class="form-group">
                    <div class="col-12">
                        <label for="no_hp">Nomor Hp</label>
                        <input class="form-control mb-1 @error('no_hp') is-invalid @enderror" type="text"
                            id="no_hp" placeholder="Masukan Nomor Hp Anda" name="no_hp"
                            value="{{ old('no_hp') }}>
                    </div>
                </div>

                <div class="form-group">
                    <div class="col-12">
                        <label for="alamat">Alamat</label>
                        <input class="form-control mb-1 @error('alamat') is-invalid @enderror" type="text"
                            id="alamat" placeholder="Masukan Alamat Anda" name="alamat"
                            value="{{ old('alamat') }}>
                    </div>
                </div>

                <div class="form-group">
                    <div class="col-12">
                        <label for="period">Password</label>
                        <div class="input-group mb-1">
                            <input class="form-control @error('password') is-invalid @enderror" type="password"
                                name="password" id="passwordId" placeholder="Masukan Password Anda">
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

Gambar 4.57 Source Code Halaman Kelola Manajemen Akun -form tambah

```

<div class="modal fade" id="editUserModal">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Edit User</h5>
        <button type="button" class="close" data-dismiss="modal"><span>&times;</span></button>
      </div>
      <div class="modal-body">
        <form class="form-horizontal" action="{{ route('manajemen-akun.update', '') }}" id="editUser" method="POST">
          @csrf
          @method('put')
          <input type="hidden" name="_token" value="{{ csrf_token() }}>
          <input type="hidden" id="checkUsername">
          <input type="hidden" id="checkEmail">

          <div class="form-group">
            <div class="col-12">
              <label for="name">Nama Lengkap</label>
              <input class="form-control mb-1 @error('edit_nama') is-invalid @enderror" type="text"
                id="edit_nama" placeholder="Masukkan Nama Lengkap" name="edit_nama"
                value="{{ old('edit_nama') }}" required>
              @error('edit_nama')
                <span class="text-danger">{{ $message }}</span>
              @enderror
            </div>
          </div>

          <div class="form-group">
            <div class="col-12">
              <label for="username">Username</label>
              <input class="form-control mb-1 @error('edit_username') is-invalid @enderror" type="text"
                id="edit_username" placeholder="Masukkan Username" name="edit_username"
                value="{{ old('edit_username') }}" required>
              @error('edit_username')
                <span class="text-danger">{{ $message }}</span>
              @enderror
            </div>
          </div>

          <div class="form-group">
            <div class="col-12">
              <label for="email">Email</label>
              <input class="form-control mb-1 @error('edit_email') is-invalid @enderror" type="text"
                maxlength="35" type="email" id="edit_email" placeholder="Masukkan Email"
                name="edit_email" value="{{ old('edit_email') }}" required>
              @error('edit_email')
                <span class="text-danger">{{ $message }}</span>
              @enderror
            </div>
          </div>

          <div class="form-group">
            <div class="col-12">
              <label for="edit_no_hp">Nomor Hp</label>
              <input class="form-control mb-1 @error('edit_no_hp') is-invalid @enderror" type="number" id="edit_no_hp" placeholder="Masukkan No Hp"
                name="edit_no_hp" value="{{ old('edit_no_hp') }}" required>
              @error('edit_no_hp')
                <span class="text-danger">{{ $message }}</span>
              @enderror
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>

```

Gambar 4.58 Source Code Halaman Manajemen Akun -form edit

```

<div class="row">
    <div class="col-lg-12">
        <div class="card">
            <div class="card-header">
                <h4 class="card-title">Update Password</h4>
            </div>
            <div class="card-body">
                <form action="{{ route('manajemen-akun.updatePassword', $user) }}" method="POST">
                    @csrf

                    <div class="form-group">
                        <label for="passwordBaru">Password Baru<span class="text-danger">*</span></label>
                        <div class="input-group">
                            <input type="password" name="password_baru" parsley-trigger="change"
                                   placeholder="Masukkan Password Baru"
                                   class="form-control @error('password_baru') is-invalid @enderror" id="password_baru"
                                   value="{{ old('password_baru') }}">
                            <div class="input-group-append">
                                <button
                                    class="btn btn-secondary fa fa-eye @error('password_baru') btn-danger @enderror toggle-password-baru"
                                    type="button"></button>
                            </div>
                        </div>

                        @error('password_baru')
                        <div class="mt-1">
                            <span class="text-danger">{{ $message }}</span>
                        </div>
                        @enderror
                    </div>

                    <div class="form-group">
                        <label for="konfirmasiPasswordBaru">Konfirmasi Password Baru<span
                               class="text-danger">*</span></label>
                        <div class="input-group">
                            <input type="password" name="konfirmasi_password_baru" parsley-trigger="change"
                                   placeholder="Masukkan Konfirmasi Password Baru"
                                   class="form-control @error('konfirmasi_password_baru') is-invalid @enderror"
                                   id="konfirmasi_password_baru" value="{{ old('konfirmasi_password_baru') }}">
                            <div class="input-group-append">
                                <button
                                    class="btn btn-secondary fa fa-eye @error('konfirmasi_password_baru') btn-danger @enderror toggle-konfirmasi-password-baru"
                                    type="button"></button>
                            </div>
                        </div>

                        @error('konfirmasi_password_baru')
                        <div class="mt-1">
                            <span class="text-danger">{{ $message }}</span>
                        </div>
                        @enderror
                    </div>

                    <div class="form-group d-flex justify-content-end mt-5">
                        <button class="btn btn-md btn-primary waves-effect waves-light mr-2" type="submit"><i
                            class="fa fa-save mr-1"></i> Simpan Perubahan</button>
                        <a href="{{ route('manajemen-akun.index') }}"
                           class="btn btn-md btn-light waves-effect"><i class="fa fa-undo mr-1"></i> Kembali</a>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

Gambar 4.59 Source Code Halaman Kelola Manajemen Akun -ubah password

4.2. Pengujian Sistem Construction Of Prototype

Pengujian merupakan proses yang bertujuan untuk memastikan apakah semua fungsi sistem dapat bekerja dengan baik dan mencari kesalahan yang mungkin terjadi pada sistem sehingga sistem dapat berjalan sebagaimana mestinya. Pengujian yang dilakukan pada *website* ini menggunakan metode pengujian *black box* yaitu pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pada metode in website diuji dengan berbagai aspek seperti hak akses halaman, masukan user, dan hasil keluaran yang di berikan oleh sistem. Berikut daftar pengujian disajikan pada beberapa tabel dibawah ini:

Tabel 4. 1. Pengujian halaman admin

| No | Pertanyaan | Hasil | | | |
|----|---|-------|---|----|----|
| | | SB | B | KB | TB |
| 1. | Apakah menampilkan halaman <i>dashboard</i> admin ? | ✓ | | | |
| 2. | Apakah dapat menyimpan data di bagian admin ? | ✓ | | | |
| 3. | Apakah dapat menghapus data di bagian admin ? | ✓ | | | |
| 4. | Apakah dapat mengedit data di bagian admin ? | ✓ | | | |
| 5. | Apakah dapat menampilkan data di bagian admin ? | ✓ | | | |
| | Total | | | | |

Tabel 4. 2. Pengujian Halaman user/pemesanan

| No | Pertanyaan | Hasil | | | |
|----|---|-------|---|----|----|
| | | SB | B | KB | TB |
| 1. | Apakah menampilkan halaman beranda <i>user</i> ? | ✓ | | | |
| 2. | Apakah menampilkan halaman paket <i>wedding user</i> ? | ✓ | | | |
| 3. | Apakah menampilkan halaman tentang / <i>about user</i> ? | ✓ | | | |
| 4. | Apakah menampilkan halaman <i>client review user</i> ? | ✓ | | | |
| 5. | Apakah menampilkan halaman jadwal <i>user</i> ? | ✓ | | | |
| 6. | Apakah dapat melakukan login dan registrasi <i>user</i> ? | ✓ | | | |

| | | | | | |
|-------|--|-------------------------------------|--|--|--|
| 7. | Apakah dapat menampilkan halaman keranjang di <i>user</i> ? | <input checked="" type="checkbox"/> | | | |
| 8. | Apakah dapat menampilkan halaman pembayaran di <i>user</i> ? | <input checked="" type="checkbox"/> | | | |
| 9. | Apakah dapat menampilkan halaman detail paket di <i>user</i> ? | <input checked="" type="checkbox"/> | | | |
| Total | | | | | |

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pemaparan yang telah dituangkan penulis pada bab-bab sebelumnya terdapat beberapa poin kesimpulan yang dapat ditarik oleh penulis antara lain sebagai berikut:

1. Tahapan pembangunan *website* ini sesuai dengan rencana yang dirancang sebelumnya dimulai dari pengumpulan data, perancangan, pembuatan *web* menggunakan bahasa pemrograman PHP dan HTML MySql untuk basis datanya, serta menggunakan *framework laravel*.
2. *Website* ini menjadi wadah bagi Makna *Wedding Organizer* untuk memberikan informasi kepada pemesannya/
3. *Website* ini memberikan informasi umum mengenai *Wedding Organizer* seperti profil, galeri, jadwal, serta pemesanan untuk pemesan.

5.2. Saran

Adapun saran yang dapat membantu dalam pengembangan sistem adalah sebagai berikut :

Penambahan fitur pembayaran dengan *bank* untuk mempermudah pembayaran

DAFTAR PUSTAKA

- [1] R. Wahyuni and Y. Irawan, "APLIKASI-BOOK UNTUK ATURAN KERJA BERBASIS WEB DI PENGADILAN NEGERI MUARA BULIAN KELAS II JAMBI," *JURNAL ILMU KOMPUTER*, vol. 9, pp. 20-26, 2020.
- [2] R. Imam and A. Sari, "SISTEM INFORMASI PADA “MAYA” WEDDING ORGANIZER BERBASIS WEBSITE," *Surya Informatika*, vol. 2018, pp. 24-33, 2018.
- [3] R. H. Gustavianto, Rancang Bangun Website Penyewaan Peralatan Event Organizer Pada Wiraswara Production Di Kota Pontianak, Pontianak: Perpustakaan Teknik Elektro Politeknik Negeri Pontianak, 2020.
- [4] P. B. A. A. Putra, V. H. Pranatawijaya and N. N. K. Sari, "Implementasi Location Based Service Pada Aplikasi Mobile Penyajian Ruang Ujian," *JURNAL SAINS DAN INFORMATIKA*, pp. 26-30, 2020.
- [5] M. Efmi, M. Susanti and F. Rahayu, "Penyewaan Online Wedding Organizer Berbasis Web," *SAINS*, vol. 5, pp. 24-30, 2019.
- [6] D. Kribiantoro and P. D. Abda'u, Dasar Pemrograman Web dengan bahasa HTML, PHP, dan Database MySQL, vol. 13, Zahira Media Publisher, 2019, pp. 97-104.
- [7] R. Fitri, Pemrograman Basis Data Menggunakan MySQL, vol. 7, Banjarmasin: Poliban Press, 2020, pp. 232-238.
- [8] I. Styawantoro and A. Komarudin, Pemrograman berbasis web HTML, PHP 7, MySQLi, Dan Bootstrap 4, klaten: Penerbit Lakeisha, 2019.
- [9] A. N. Widhi, E. Sutanta and E. K. Nurwati, "PEMANFAATAN FRAMEWORK LARAVEL UNTUK PENGEMBANGAN SISTEM INFORMASI TOKO ONLINE DI TOKO NEW TREND BATURETNO," *Jurnal Script*, vol. 7, pp. 232-238, 2019.
- [10] D. F. Ananda, Rancang Bangun Aplikasi Penyedia Jasa Wedding Organizer Anugrah Kreasi Pontianak Berbasis Website, Pontianak: Perpustakaan Teknik Elektro Pontianak, 2021.
- [11] Anhar, Panduan Menguasai PHP dan MySql, Jakarta: Media Kita, 2010.
- [12] R. Hidayat, Cara Praktis Membangun Website Gratis, Jakarta: Elex Media Komputindo, 2010.

- [13] A. Kadir, Dasar Perancangan Dan Implementasi Database Relasional, Yogyakarta: Andi, 2009.
- [14] A. Kristanto., Perancangan Sistem Informasi Dan Aplikasinya, Yogyakarta: Gava Media, 2008.
- [15] E. Maiyana, "Perancangan Aplikasi Media Informasi Lowongan Kerja Perusahaan Bagi Pencari Kerja Berbasis Web," *Sains Dan Informatika*, vol. 3, pp. 118-125, 2017.
- [16] B. Nugroho., Analisis Dan Perancangan Sistem Informasi Dengan Metodelogi Berorientasi Objek, Bandung: Informatika Bandung, 2008.
- [17] D. Prabowo, "Webiste E-Commerce menggunakan model view controller (MVC) dengan framework laravel," *Sistem informasi*, p. 24, 2015.
- [18] I. K. A. Putra, D. Pramana and N. L. P. Srinadi, "Sistem Manajemen Arsip Menggunakan Framework Laravel dan Vue.Js (Studi Kasus : BPKAD Provinsi Bali)," *JURNAL SISTEM DAN INFORMATIKA*, vol. 13, pp. 97-104, 2019.
- [19] M. Shalahuddind, Pemrograman Objek Dengan Bahasa Pemrograman C++,PHP,danJava, Bandung: Modula, 2010.
- [20] M. H. R. Susanti, "Implementasi Sistem Informasi Penggajian Untuk Membantu Manajemen Keuangan Dalam Pengolahan Usaha," *RESTI*, vol. 33, pp. 416-421, 2018.
- [21] S. H. R. Zakir, "Webbased Learning model that can be implemented In Learning Settings Without Being Limited By Time, Place, And Space," *Theoritical and Applied Information Technology*, pp. 7996-8005, 2018.

LAMPIRAN



Lampiran 1. Foto Bersama Tim Makna Wedding Organizer untuk Melakukan Observasi dan Wawancara

**NEXT JADWAL PERSIAPAN/
PERTEMUAN**

1) Kiki & Ikbal - 08 Juli 2023 - Rumah Adat Melayu
 * 01 Juni = Ketemu vendor dekor (ferly dekorasi)
 Note - Jam 4 sore di para space ✓
 * 21 Juni = BP4 KUA & Rapat vendor
 * 27 Juni = Fitting final
 * 05 Juli = Gladi Pomfit di Rumah Adat Melayu
 * 06 Juli = Betangas pengantin (pagi)
 * 07 Juli = Betangas pengantin (pagi) & Gladi sangkur pora di halaman Rumah Adat Melayu (sore)

2) Nanda & Haris - 16 Juli 2023 - Gedung Korpri
 * 04 Juli = Rapat vendor dan keluarga
 * 05 Juli = BP4 KUA
 * 08/09 Juli = Client antar isi seserahan ke kantor makna
 * 10 Juli = Fitting final (Mysha Hamisah)

Lampiran 2. Contoh dari Manajemen Data Pemesanan Menggunakan Aplikasi Note dari Smarthphone