

**APLIKASI MAKAH WEDDING SERVICE
BERBASIS WEBSITE DENGAN
FRAMEWORK LARAVEL**

TUGAS AKHIR

Sebagai salah satu syarat untuk menyelesaikan
Program Diploma III Teknik Informatika
Politeknik Negeri Indramayu



Oleh :
CAMELIA
NIM 1803006

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
AGUSTUS 2021**

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

Nama : Camelia

NIM : 1803006

Program Studi : Diploma III Teknik Informatika

Judul : Aplikasi Makah *Wedding Service Berbasis Website*
Dengan *Framework Laravel*

Pembimbing : I. Muhamad Mustamiin, S.Pd., M.Kom.
NIP. 199205052019031011

Pembimbing : II. Alifia Puspaningrum, S.Pd., M.Kom.
NIP. 199305282019032024

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal 23 Agustus 2021 dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Diploma III Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

Ketua Penguji : Iryanto, S.Si., M.Si.
NIP. 199008012019031014

Anggota Penguji I : Darsih, S.Kom., M.Kom.
NIPPK. 198109062021212004

Anggota Penguji II : Muhamad Mustamiin, S.Pd., M.Kom.
NIP. 199205052019031011

Indramayu, September 2021
Ketua Jurusan Teknik Informatika



PERNYATAAN KEASLIAN

Dengan ini saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri serta Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam naskah ini dan disebutkan dalam daftar pustaka.

Indramayu, 23 Agustus 2021

Yang menyatakan



Camelia

NIM 1803006

ABSTRAK

Wedding organizer adalah jasa layanan yang bekerja secara personal untuk membantu calon pengantin dalam perencanaan dan mengawali penyelenggaraan pesta pernikahan sesuai dengan jadwal dan *budget* yang telah diatur. Pada saat pandemi Covid-19 jasa layanan pengelola *Wedding Organizer* menjadi salah satu yang paling terdampak, beralih dari pemasaran yg bersifat konvensional ke digital adalah sebagai salah satu cara untuk beradaptasi pada saat pandemi covid-19. Makah *Wedding Service* merupakan penyedia jasa layanan *wedding organizer*, *make up wedding*, *prewedding*, *graduation*, pesta, kelas *make up* pemula, kelas *make up* lanjutan, dekorasi dan alat pesta. Namun informasi jasa yang disediakan pada *wedding service* masih kurang diketahui oleh masyarakat, lalu proses pemesanannya masih bersifat konvensional. Penggunaan *website* bisa digunakan sebagai salah satu cara untuk pemilik dalam menerapkan strategi pemasaran secara *online* selama masa pandemi Covid-19. Dalam membangun aplikasi ini penulis menggunakan metode *waterfall* dari tahapan analisis, *design*, implementasi, pengujian dan pemeliharaan. Aplikasi ini mengelola data pegawai, *client*, pesanan, paket, pembayaran, jadwal, transportasi, dan testimoni. Hasil pengujian yang diperoleh pada website ini berdasarkan penilaian kuesioner dari responden yang mewakili pengguna, yaitu pemilik dan pegawai Makah *Wedding Service* kemudian masyarakat mencapai presentase 94,75%. Dengan demikian *website* ini dapat dinilai sudah baik untuk digunakan.

Kata Kunci : *Wedding organizer*, Makah *Wedding Service*, *Make Up*, *Website*, Metode *Waterfall*.

ABSTRACT

Wedding organizer is a service that works personally to assist the bride and groom in planning and supervising the implementation of the wedding party according to the schedule and budget that has been arranged. During the Covid-19 pandemic, Wedding Organizer management services were one of the most affected, switching from conventional to digital marketing as a way to adapt during the COVID-19 pandemic. Makah Wedding Service is a provider of wedding organizer services, wedding make-up, pre-wedding, graduation, parties, beginner make-up classes, advanced make-up classes, decorations and party supplies. However, the information about the services provided at the wedding service is still not known by the public, so the ordering process is still conventional. The use of the website can be used as a way for owners to implement online marketing strategies during the Covid-19 pandemic. In building this application the author uses the waterfall method from the stages of analysis, design, implementation, testing and maintenance. This application manages employee data, clients, orders, packages, payments, schedules, transportation, and testimonials. The test results obtained on this website are based on questionnaire assessments from respondents who represent users, namely owners and employees of Makah Wedding Service, then the community reaches a percentage of 94,75%. Thus this website can be considered as good to use.

Keywords : *Wedding Organizer, Makah Wedding Service, Make up, Website, Waterfall Method.*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah Subhanahu Wa Ta'ala yang Maha kuasa atas limpahan rahmat, hidayah dan kasih sayang-Nya. Tidak lupa pula shalawat dan salam senantiasa tercurahkan bagi Nabi Muhammad Shallallahu Alaihi Wasallam yang telah memberikan syafaatnya. Sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework Laravel*”.

Tugas Akhir ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma III Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

Kelancaran kegiatan Tugas Akhir tidak terlepas dari dukungan dan bimbingan berbagai pihak, Oleh karena itu, dalam kesempatan ini penulis ingin mengucapkan rasa terimakasih yang sebanyak-banyaknya kepada:

1. Bapak Casiman Sukardi, S.T., M.T selaku Direktur Politeknik Negeri Indramayu.
2. Bapak Iryanto S.Si., M.Si selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Indramayu.
3. Muhammad Anis Al Hilmi, S.Si., M.T., selaku Koordinator Tugas Akhir jurusan Teknik Informatika Politeknik Negeri Indramayu.
4. Bapak Muhamad Mustamiin, S.Pd.,M.Kom., selaku Dosen pembimbing 1.
5. Ibu Alifia Puspaningrum, S.Pd., M.Kom., selaku Dosen pembimbing 2.
6. Bapak Eka Ismantohadi, S.Kom., M.Eng., selaku Dosen wali kelas D3TI3A.
7. Kedua orangtua penulis yang selalu mendoakan dan memberikan dukungan kepada penulis agar bisa menyelesaikan Tugas Akhir dengan lancar.
8. Ibu Nur khalimah S.Pd., selaku pemilik dari Makah *Wedding Service*.
9. Semua pihak yang telah membantu dalam penyelesaian Tugas Akhir ini dari awal hingga akhir yang penulis tidak bisa sebutkan satu persatu.

Penulis menyadari adanya kekurangan dalam proses penulisan laporan ini. Sebagai bentuk perbaikan, penulis terbuka pada saran dan masukan dari pembaca.

Indramayu, 23 Agustus 2021

Penulis

DAFTAR ISI

	Halaman
SAMPUL DEPAN	i
HALAMAN JUDUL	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1. <i>Wedding Organizer</i>	6
2.2. Makah Wedding Service	6
2.3. <i>Website</i>	7
2.4. <i>Framework Laravel</i>	7
2.5. Visual Studio Code.....	8
2.6. <i>Personal Home Page (PHP)</i>	8
2.7. <i>Web Server</i>	9
2.8. Figma.....	9
2.9. <i>Flowchart</i>	9
2.10. <i>Unified Modeling Language (UML)</i>	10
2.11. <i>Entity Relationship Diagram (ERD)</i>	14
2.12. Metode waterfall	15
2.13. Kuesioner	17
BAB III METODE PELAKSANAAN.....	18
3.1. Metode Pelaksanaan.....	18
3.2. Metode Pengumpulan Data	19
3.3. Analisis Kebutuhan	20
3.4. <i>Flowchart</i>	22
3.5. Perancangan Sistem.....	26
3.5.1. Rancangan UML (<i>Undefined Modelling Language</i>).....	26
3.6. Rancangan ERD (<i>Entity Relationship Diagram</i>)	59
3.7. Rancangan Database.....	61
3.8. Rancangan Interface.....	68
BAB IV HASIL DAN PEMBAHASAN.....	80
4.1. Hasil	80

4.2. Pembahasan.....	80
4.2.1. Implementasi	80
4.3. Pengujian Aplikasi	95
BAB V PENUTUP.....	101
5.1. Kesimpulan.....	101
5.2. Saran.....	102
DAFTAR PUSTAKA	103
LAMPIRAN.....	105

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Logo Makah <i>Wedding Service</i>	6
Gambar 2.2 Logo Framework Laravel.....	7
Gambar 2.3 Logo Visual Studio Code	8
Gambar 2.4 Logo PHP	8
Gambar 2.5 Logo XAMPP	9
Gambar 2.6 Logo Figma	9
Gambar 2.7 Metode Waterfall.....	15
Gambar 3.1 Metode Waterfall.....	18
Gambar 3.2 <i>Flowchart Admin</i>	23
Gambar 3.3 <i>Flowchart Pegawai</i>	24
Gambar 3.4 <i>Flowchart Client</i>	25
Gambar 3.5 <i>Use Case Diagram</i>	26
Gambar 3.6 <i>Activity Diagram Melihat Lending Page</i>	27
Gambar 3.7 <i>Activity Diagram Registrasi Client</i>	28
Gambar 3.8 <i>Activity Diagram Login Client</i>	28
Gambar 3.9 <i>Activity Diagram Login Admin</i>	29
Gambar 3.10 <i>Activity Diagram Login Pegawai</i>	29
Gambar 3.11 <i>Activity Diagram Melihat Profile</i>	30
Gambar 3.12 <i>Activity Diagram Melihat Daftar Paket</i>	30
Gambar 3.13 <i>Activity Diagram Mengelola Data Jadwal Client</i>	31
Gambar 3.14 <i>Activity Diagram Mengelola Data Jadwal Admin</i>	32
Gambar 3.15 <i>Activity Diagram Mengelola Data Testimoni Client</i>	33
Gambar 3.16 <i>Activity Diagram Mengelola Data Testimoni Admin</i>	34
Gambar 3.17 <i>Activity Diagram Mengelola Data Pemesanan Client</i>	35
Gambar 3.18 <i>Activity Diagram Mengelola Data Pemesanan Admin</i>	36
Gambar 3.19 <i>Activity Diagram Mengelola Data Pesanan Pegawai</i>	37
Gambar 3.20 <i>Activity Diagram Melihat Data Jadwal Pegawai</i>	38
Gambar 3.21 <i>Activity Diagram Mengelola Data Paket Admin</i>	39
Gambar 3.22 <i>Activity Diagram Mengelola Data Pegawai Admin</i>	40
Gambar 3.23 <i>Activity Diagram Mengelola Data Client Admin</i>	41
Gambar 3.24 <i>Activity Diagram Mengelola Data Pembayaran Client</i>	42
Gambar 3.25 <i>Activity Diagram Mengelola Data Pembayaran Admin</i>	43
Gambar 3.26 <i>Activity Diagram Mengelola Data Pengeluaran Admin</i>	44
Gambar 3.27 <i>Activity Diagram Mengelola Data Transportasi Admin</i>	45
Gambar 3.28 <i>Activity Diagram Mengelola Data Landing Page Admin</i>	46
Gambar 3.29 <i>Class Diagram</i>	47
Gambar 3.30 <i>Sequence Diagram Melihat Landing Page</i>	48
Gambar 3.31 <i>Sequence Diagram Registrasi</i>	48
Gambar 3.32 <i>Sequence Diagram Login</i>	49
Gambar 3.33 <i>Sequence Diagram Melihat Profile</i>	49
Gambar 3.34 <i>Sequence Diagram Melihat Daftar Paket</i>	50
Gambar 3.35 <i>Sequence Diagram Mengelola Data Jadwal</i>	50
Gambar 3.36 <i>Sequence Diagram Mengelola Data Testimoni</i>	51
Gambar 3.37 <i>Sequence Diagram Mengelola Data Pesanan</i>	52

Gambar 3.38 <i>Sequence Diagram</i> Melihat Data Jadwal	52
Gambar 3. 39 <i>Sequence Diagram</i> Mengelola Data Paket.....	53
Gambar 3.40 <i>Sequence Diagram</i> Mengelola Data Pegawai	54
Gambar 3.41 <i>Sequence Diagram</i> Mengelola Data <i>Client</i>	55
Gambar 3.42 <i>Sequence Diagram</i> Mengelola Data Pembayaran.....	56
Gambar 3.43 <i>Sequence Diagram</i> Mengelola Data Pengeluaran.....	57
Gambar 3.44 <i>Sequence Diagram</i> Mengelola Data Transportasi.....	58
Gambar 3.45 <i>Sequence Diagram</i> Mengelola Data <i>Landing Page</i>	59
Gambar 3.46 Rancangan ERD (<i>Entity Relationship Diagram</i>)	60
Gambar 3.47 Rancangan Tampilan Halaman <i>Landing Page</i>	69
Gambar 3.48 Rancangan Tampilan Halaman <i>Registrasi</i>	69
Gambar 3.49 Rancangan Tampilan Halaman <i>Login</i>	70
Gambar 3.50 Rancangan Tampilan Halaman Admin	70
Gambar 3.51 Rancangan Tampilan Halaman Data Pegawai Admin	71
Gambar 3.52 Rancangan Tampilan Halaman Data <i>Client</i> Admin.....	71
Gambar 3.53 Rancangan Tampilan Halaman Data Pesanan Admin.....	72
Gambar 3.54 Rancangan Tampilan Halaman Data Transportasi Admin.....	73
Gambar 3.55 Rancangan Tampilan Halaman Data Jadwal Admin.....	73
Gambar 3.56 Rancangan Tampilan Halaman Data Testimoni Admin	74
Gambar 3.57 Rancangan Tampilan Halaman Data Pengeluaran Admin	74
Gambar 3.58 Rancangan Tampilan Halaman Data <i>Landing Page</i> Admin.....	75
Gambar 3.59 Rancangan Tampilan Halaman Pegawai.....	76
Gambar 3.60 Rancangan Tampilan Halaman Data Pesanan Pegawai	76
Gambar 3.61 Rancangan Tampilan Halaman Data Jadwal Pegawai	77
Gambar 3.62 Rancangan Tampilan Halaman <i>Client</i>	77
Gambar 3.63 Rancangan Tampilan Halaman Pesan Paket <i>Client</i>	78
Gambar 3.64 Rancangan Tampilan Halaman Daftar Paket <i>Client</i>	78
Gambar 3.65 Rancangan Tampilan Halaman Jadwal <i>Client</i>	79
Gambar 3.66 Rancangan Tampilan Halaman Testimoni <i>Client</i>	79
Gambar 4.1 Implementasi Tampilan Halaman <i>Landing Page</i>	85
Gambar 4.2 Implementasi Tampilan Halaman <i>Registrasi</i>	85
Gambar 4.3 Implementasi Tampilan Halaman <i>Login</i>	86
Gambar 4.4 Implementasi Tampilan Halaman <i>Admin</i>	86
Gambar 4.5 Implementasi Tampilan Halaman Data Pegawai Admin	87
Gambar 4.6 Implementasi Tampilan Halaman Data <i>Client</i> Admin.....	87
Gambar 4.7 Implementasi Tampilan Halaman Data Pesanan Admin.....	88
Gambar 4.8 Implementasi Tampilan Halaman Data Paket Admin	88
Gambar 4.9 Implementasi Tampilan Halaman Data Transportasi Admin.....	89
Gambar 4.10 Implementasi Tampilan Halaman Data Jadwal Admin.....	89
Gambar 4.11 Implementasi Tampilan Halaman Data Testimoni Admin	90
Gambar 4.12 Implementasi Tampilan Halaman Data Pengeluaran Admin	90
Gambar 4.13 Implementasi Tampilan Halaman Data <i>Landing Page</i> Admin	91
Gambar 4.14 Implementasi Tampilan Halaman Pegawai.....	91
Gambar 4.15 Implementasi Tampilan Halaman Pesanan Pegawai.....	92
Gambar 4.16 Implementasi Tampilan Halaman Jadwal Pegawai.....	92
Gambar 4.17 Implementasi Tampilan Halaman <i>Client</i>	93
Gambar 4.18 Implementasi Tampilan Halaman pesan paket <i>Client</i>	93
Gambar 4.19 Implementasi Tampilan Halaman Daftar Paket <i>Client</i>	94

Gambar 4.20 Implementasi Tampilan Halaman Atur Jadwal <i>Client</i>	94
Gambar 4.21 Implementasi Tampilan Halaman Testimoni <i>Client</i>	95

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-Simbol <i>Folwchart</i>	10
Tabel 2.2 Simbol-Simbol <i>Use Case Diagram</i>	11
Tabel 2.3 Simbol-Simbol <i>Activity Diagram</i>	12
Tabel 2.4 Simbol-Simbol <i>Class Diagram</i>	13
Tabel 2.5 Simbol-Simbol <i>Squence Diagram</i>	13
Tabel 2.7 Simbol-Simbol ERD	14
Tabel 3.1 Kebutuhan <i>Hardware</i>	22
Tabel 3.2 Kebutuhan <i>Software</i>	22
Tabel 3.3 Rancangan Tabel Admin.....	61
Tabel 3.4 Rancangan Tabel Pegawai	61
Tabel 3.5 Rancangan Tabel Konsumen	62
Tabel 3.6 Rancangan Tabel Pesanan.....	63
Tabel 3.7 Rancangan Tabel Paket.....	63
Tabel 3.8 Rancangan Tabel Kabupaten	64
Tabel 3.9 Rancangan Tabel Kecamatan.....	64
Tabel 3.10 Rancangan Tabel Transportasi.....	65
Tabel 3.11 Rancangan Tabel Transaksi	65
Tabel 3.12 Rancangan Tabel Jadwal.....	66
Tabel 3.13 Rancangan Tabel Testimoni.....	67
Tabel 3.14 Rancangan Tabel <i>Landing</i>	67
Tabel 3.15 Rancangan Tabel Pengeluaran	68
Tabel 4.1 Implementasi Tabel Admin.....	81
Tabel 4. 2 Implementasi Tabel Pegawai	81
Tabel 4. 3 Implementasi Tabel Konsumen	81
Tabel 4.4 Implementasi Tabel Pesanan.....	82
Tabel 4.5 Implementasi Tabel Paket.....	82
Tabel 4. 6 Implementasi Tabel Kabupaten	82
Tabel 4.7 Implementasi Tabel Kecamatan.....	82
Tabel 4.8 Implementasi Tabel Transportasi.....	83
Tabel 4.9 Implementasi Tabel Transaksi	83
Tabel 4.10 Implementasi Tabel Jadwal.....	83
Tabel 4.11 mplementasi Tabel Testimoni	84
Tabel 4.12 mplementasi Tabel <i>Landing</i>	84
Tabel 4.13 Implementasi Tabel Pengeluaran	84
Tabel 4.14 Keterangan Nilai Kuesioner.....	96
Tabel 4.15 Pertanyaan Kuesioner	96
Tabel 4.16 Jawaban Kuesioner	97
Tabel 4.17 Hasil Perhitungan Kuesioner	98

DAFTAR LAMPIRAN

Lampiran 1. *Source Code*

Lampiran 2. *Biodata Penulis*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Wedding organizer adalah jasa layanan yang bekerja secara personal untuk membantu calon pengantin dalam perencanaan dan supervisi penyelenggaraan pesta pernikahan sesuai dengan jadwal dan *budget* yang telah diatur. *Wedding Organizer* sering kali dicari oleh masyarakat karena selain untuk mempermudah dalam melakukan perencanaan pesta pernikahan (Maiyana dkk, 2019).

Pada saat pandemi Covid-19 Jasa layanan pengelola *Wedding Organizer* menjadi salah satu yang paling terdampak. Dikarenakan Selama masa pandemi Covid-19 ada larangan berkumpul dan mengadakan kegiatan yang menyebabkan kerumunan orang. Namun, bukan berarti tidak ada harapan bagi pelaku bisnis jasa layanan *Wedding Organizer*. Perlahan-lahan harapan muncul bagi para pelaku bisnis jasa *layanan Wedding Organizer* seiring diperbolehkannya kembali mengadakan acara pernikahan dengan menerapkan protokol kesehatan yang ketat Seperti memakai masker, menjaga jarak, mengecek suhu tubuh, mencuci tangan, dan pembatasan jumlah tamu undangan. Beralih dari pemasaran yg bersifat konvensional ke digital adalah sebagai salah satu cara untuk beradaptasi pada saat pandemi covid-19. Para pelaku bisnis harus bisa mengikuti tren yang sedang terjadi di masyarakat secara daring dikarenakan masyarakat akan tetap membatasi kegiatan mereka dengan tetap melakukan aktifitas secara *online* (Wijayanto, 2021).

Makah *Wedding Service* (MWS) merupakan salah satu penyedia jasa *wedding organizer* yang berada di kota Indramayu. Jasa yang ditawarkan oleh MWS yang terdiri dari *wedding organizer*, *make up wedding*, *prewedding*, *graduation*, pesta, kelas *meke up* pemula, kelas *make up* lanjutan, dekorasi dan alat pesta. Namun informasi jasa yang disediakan pada MWS masih kurang diketahui oleh masyarakat, lalu proses pemesanannya masih bersifat konvensional yaitu bertemu langsung antara pemilik MWS dengan *client*. Proses pemesanan yang dilakukan adalah *client* terlebih dahulu menemui pemilik MWS kemudian melakukan konsultasi dengan pemilik lalu memilih paket yang disediakan pada MWS setalah

client setuju memilih paket yang telah dipilih, *client* menentukan tanggal untuk melakukan pemesanan tersebut namun apabila pada tanggal yang telah ditentukan oleh *client* sudah ada yang memesan maka *client* menentukan tanggal yang berbeda dan disetujui oleh pemilik, kemudian *client* melakukan pembayaran dengan pemilik, setelah semua proses sudah disetujui maka proses pemesanan telah selesai.

Setelah mengetahui adanya masalah yang telah dipaparkan diatas bahwa informasi jasa yang disediakan pada MWS dan proses pemesanan masih bersifat konvensional, maka akan dilakukan penelitian untuk membuat sebuah aplikasi MWS berbasis *website* yang berfungsi sebagai media yang memfasilitasi masyarakat agar mendapatkan informasi jasa yang disediakan oleh MWS dan *client* agar bisa melakukan proses pemesanan dengan tanpa bertemu langsung dengan pemilik jasa MWS. Aplikasi ini juga menampilkan informasi paket-paket yang tersedia pada MWS beserta harganya.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan sebelumnya maka diperoleh suatu rumusan permasalahan yang menjadi dasar pembuatan Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel, yakni sebagai berikut :

1. Bagaimana merancang dan membangun Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel?
2. Bagaimana masyarakat dapat mengetahui informasi jasa yang disediakan pada Makah *Wedding Service*?
3. Bagaimana cara mempermudah *client* untuk melakukan proses pemesanan jasa yang disediakan yang disediakan pada Makah *Wedding Service*?
4. Bagaimana cara mempermudah admin dalam mengelola pemesanan jasa pada Makah *Wedding Service*?

1.3. Batasan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya dapat diperoleh beberapa batasan masalah, diantaranya:

1. Informasi yang dapat diketahui oleh masyarakat di *website* Makah *Wedding Service* hanya sebatas informasi *profile*, jasa yang disediakan, paket-paket

beserta harganya, testimoni *client*, dan *contact* yang bisa dihubungi pada Makah *Wedding Service*.

2. Proses pemesanan oleh *client* hanya dapat dilakukan apabila *client* sudah mendaftar akun di *website* Makah *Wedding Service* kemudian proses pemesanan akan dikonfirmasi oleh admin.
3. Proses pengelolahan pemesanan oleh admin di *website* Makah *Wedding Service* hanya mencangkup mengkonfirmasi pemesanan yang sudah masuk, kemudian menambahakan pemesanan, mengedit pemesanan, melihat *detail* pemesanan, melakukan pembayaran pemesanan yang sudah masuk, dan menghapus pemesanan.
4. Sistem dibangun menggunakan bahasa pemrograman PHP dengan *framework* Laravel *versi* 8 dan database MySQL.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disampaikan sebelumnya, maka tujuan dari pembuatan sistem ini adalah :

1. Dapat merancang dan membangun Aplikasi Makah *Wedding Service* dengan menggunakan *Framework* Laravel.
2. Masyarakat dapat mengetahui informasi jasa yang disediakan pada Makah *wedding Service*.
3. *Client* mendapatkan kemudahan untuk melakukan proses pemesanan jasa yang disediakan pada Makah *Wedding Service*.
4. Mempermudah admin dalam mengelola pemesanan jasa yang disediakan pada Makah *Wedding Service*.

1.5. Manfaat Penelitian

Berdasarkan tujuan tersebut maka diharapkan dapat memberikan manfaat sebagai berikut :

1. Memberikan kemudahan bagi masyarakat dalam mendapatkan informasi jasa yang disediakan pada Makah *Wedding Service*.
2. Memberikan kemudahan bagi *client* untuk melakukan proses pemesanan jasa yang disediakan pada Makah *Wedding Service*.
3. Memberikan kemudahan bagi admin dalam mengelola pemesanan jasa

Makah *Wedding Service*.

4. Membangun Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel.

1.6. Sistematika Penulisan

Dalam sistematika penulisan laporan Tugas Akhir ini dibagi menjadi lima bab, yang dimana setiap bagian memiliki pembahasan yang berbeda-beda tetapi saling terkait antara satu dengan yang lainnya. Untuk memudahkan penulisan laporan Tugas Akhir ini, maka akan diurutkan dan dijabarkan setiap bagian secara sistematis. Adapun urutan sistematika penulisan laporan Tugas Akhir ini sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini membahas mengenai isi dari latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, beserta sistematika penulisan Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel.

BAB II LANDASAN TEORI

Pada bab II landasan teori menjelaskan definisi ataupun konsep yang telah disusun berdasarkan dari sumber referensi yang terpercaya mengenai teori-teori yang terkait dengan penyusunan penulisan laporan Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel. kemudian sumber yang dijadikan referensi didapatkan dari jurnal dan buku.

BAB III METODE PELAKSANAAN

Bab ini membahas mengenai tahapan beserta metode penelitian yang digunakan dalam Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel. Bagian ini berisikan perancangan website yang terdiri dari flowchart, UML, ERD dan rancangan *design interface* yang telah dibuat untuk menjelaskan tahapan alur proses.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini memaparkan mengenai implementasi dalam Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel berdasarkan perancangan yang telah dijelaskan beserta membahas mengenai implementasi

database, implementasi *design interface*, dan pengujian sistem dengan menggunakan metode kuesioner.

BAB V PENUTUP

Pada bab v penutup membahas mengenai kesimpulan dan saran yang berkaitan dengan analisa dan optimasi pada website berdasarkan yang telash diuraikan pada bab-bab sebelumnya. Kesimpulan berisikan dari hasil rangkuman berdasarkan point-point yang dibutuhkan pada penulisan laporan Tugas Akhir Aplikasi Makah *Wedding Service* Berbasis Website Dengan Framework Laravel. Saran berisikan mengenai pendapat dan masukan dari penulis kepada pembaca.

BAB II

LANDASAN TEORI

2.1. *Wedding Organizer*

Wedding organizer adalah suatu jasa yang secara pribadi membantu calon pengantin dalam menentukan perencanaan dan mengawasi setiap pelaksanaan rangkaian acara pernikahan agar bisa sesuai dengan jadwal yang sudah ditetapkan, seperti memberikan informasi menegenai berbagai macam hal yang berhubungan dengan acara pernikahan, memberikan bantuan untuk bisa merumuskan konsep pernikahan, memberikan negosiasi dan koordinasi dengan pihak gedung dan *vendor*, mengatur acara dan seluruh rekanan pada hari pernikahan, mengawasi proses dan *detail* acara (Maiyana dkk, 2019).

2.2. *Makah Wedding Service*



Gambar 2.1 Logo Makah Wedding Service (Khalimah, 2021)

Makah Wedding Service (MWS) merupakan suatu usaha yang bergerak dalam bidang *Wedding Organizer* atau jasa pengelolaan acara pernikahan. Usaha ini dikelola oleh Makhrus S.Si dan Nur khalimah S.Pd, pasangan suami-istri yang sudah ahli dalam bidang pengelolaan jasa pernikahan (Khalimah, 2016). Logo *Makah Wedding Service* dapat dilihat pada Gambar 2.1.

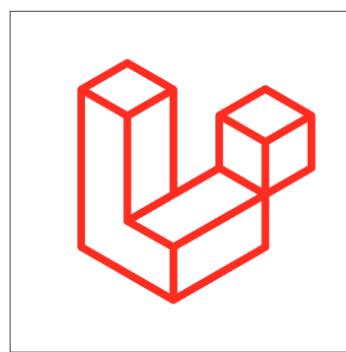
Didirikan sejak 20 oktober 2012 jasa MWS yang cakupan kerjanya meliputi daerah Indramayu, Subang, Majalengka, Cirebon, dan Kuningan, sudah banyak diminati oleh para calon pengantin karena kualitasnya yang unggul dengan harga yang bisa dinegosiasi (Khalimah, 2016).

Setiap *Wedding Organizer* pastinya memiliki keunggulan dan kelemahan masing-masing. Harga yang terlalu tinggi, kualitas yang belum memadai, dan pengelolaan yang belum baik merupakan kekurangan yang pada umumnya mendominasi jasa *Wedding Organizer*. Hal inilah yang memotivasi didirikannya MWS. Dengan perhitungan yang matang, komunikasi yang baik bersama *client* dan kedisiplinan merupakan kunci dari MWS guna meminimalisir kelemahan-kelemahan yang terjadi pada bisnis *Wedding Organizer*. Dengan demikian, MWS melayani sejak perencanaan praacara sampai pada evaluasi pascaacara dengan jasa yang disediakan meliputi Rias Pengantin, Dekorasi (Pelaminan, tenda, kursi, prasmanan, panggung, sound system, lampu, diesel, blower, karpet), Foto dan video shooting, perawatan pengantin, kartu undangan, hiburan modern dan tradisional (sintren, jaran lumping, jaipong, organ pop, dangdut, nasyid, qasidah), upacara adat, dan keperluan pernikahan lainnya (Khalimah, 2016).

2.3. Website

Website adalah sebuah kumpulan halaman pada suatu domain pada internet yang dibuat dengan tujuan tertentu dan saling berhubungan kemudian dapat diakses secara luas melalui halaman depan menggunakan browser dengan menggunakan URL website (Triwahyuni dkk, 2021).

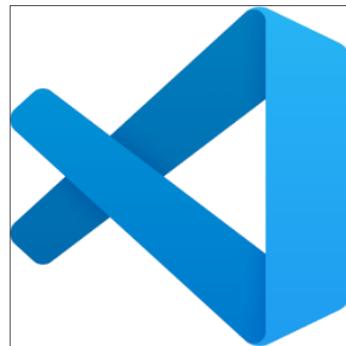
2.4. Framework Laravel



Gambar 2.2 Logo *Framework Laravel* (Anonim, 2021)

Laravel adalah kerangka aplikasi web dengan sintaks yang ekspresif dan elegan. Kami telah meletakkan fondasinya - membebaskan Anda untuk berkreasi tanpa memusingkan hal-hal kecil (Anonim, 2021). Logo *Framenwork Laravel* dapat dilihat pada Gambar 2.2.

2.5. Visual Studio Code



Gambar 2.3 Logo Visual Studio Code (Anonim, 2021)

Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop Anda dan tersedia untuk Windows, macOS, dan Linux. Muncul dengan dukungan bawaan untuk JavaScript, TypeScript dan Node.js dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti C++, C#, Java, Python, PHP, Go) dan runtime (seperti .NET dan Unit) (Anonim, 2021). Logo Visual Studio Code dapat dilihat pada Gambar 2.3.

2.6. Personal Home Page (PHP)



Gambar 2.4 Logo PHP (Anonim, 2021)

PHP adalah bahasa *scripting* tujuan umum populer yang sangat cocok untuk pengembangan *web*. Cepat, fleksibel, dan pragmatis, PHP telah mendukung segalanya mulai dari *blog* Anda hingga situs *web* paling populer di dunia (Anonim, 2020-2021). Logo PHP dapat dilihat pada Gambar 2.3.

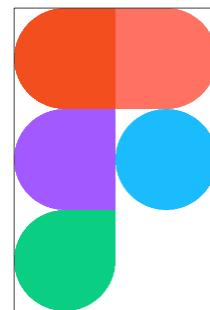
2.7. Web Server



Gambar 2.5 Logo XAMPP (Anonim, 2021)

XAMPP adalah distribusi Apache yang sepenuhnya gratis dan mudah dipasang yang berisi *MariaDB*, *PHP*, dan *Perl*. Paket *open source* XAMPP telah diatur agar sangat mudah dipasang dan digunakan (Anonim, 2021). Logo Xampp dapat dilihat pada gambar 2.6.

2.8. Figma



Gambar 2.6 Logo Figma (Anonim, 2021)

Figma adalah alat UI online gratis untuk membuat, berkolaborasi, membuat prototipe, dan menyerahkan. Figma adalah satu-satunya platform yang menyatukan fitur desain canggih yang sudah Anda sukai dan alur kerja yang lebih efisien untuk boot (Anonim, 2021). Logo Figma dapat dilihat pada gambar 2.6.

2.9. Flowchart

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. Flowchart merupakan gambar yang memperlihatkan langkah-langkah dari suatu prograrm dan hubungan antar proses beserta pernyataanya. Gambaran ini dinyatakan dengan simbol menggambarkan proses tertentu. Sedangkan proses digambarkan dengan penghubung (Johan, 2020).

Flowchart terdiri dari simbol-simbol tertentu, beberapa simbol-simbol yang digunakan dalam *flowchart* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-Simbol *Flowchart*

Simbol	Nama	Keterangan
	Garis Alir (<i>Flow Line</i>)	Menunjukkan arah aliran program.
	Decision	Menunjukkan perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	Terminator	Menunjukkan permulaan / akhir program (<i>start/end</i>).
	Manual Input	Menunjukkan pemasukan data secara manual <i>on-line</i> keyboard.
	Proses	Merupakan proses perhitungan/proses pengolahan data.
	<i>Input/Output Data</i>	Menunjukkan proses penginputan/output data, parameter, dan informasi.
	Display	Menunjukkan peralatan output yang digunakan yaitu layar, plotter, printer, dan sebagainya.
	Manual Operation	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/sistem.
	Database	Menunjukkan data-data yang di akses atau disimpan ke <i>database</i> .
	Disk Storage	Menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .

Sumber: Malabay, 2016.

2.10. *Unified Modeling Language* (UML)

Unified Modelling Language (UML) adalah bahasa standar yang digunakan untuk menjelaskan dan menyesuaikan artifak dari proses analisis dan *design* berorientasi objek. UML menyediakan standar pada notasi dan *diagram* yang bisa digunakan untuk memodelkan suatu sistem. UML menjadi bahasa yang bisa

digunakan untuk berkomunikasi dalam prespektif objek atau *user* dengan *developer* pemrograman dengan *developer design* (Efendi, 2018)

UML terdiri atas pengelompokan diagram-diagram sistem atau sudut pandang tertentu. Contoh dari diagram UML diantaranya yaitu (Efendi, 2018):

a. *Use Case Diagram*

Use case diagram adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor (Tabrani, 2019).

Simbol-simbol *Use case diagram* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Berfungsi sebagai peran pengguna ketika berinteraksi dengan use case.
	<i>Dependency</i>	Pada suatu elemen mandiri atau independent akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri atau independent.
	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
	<i>Include</i>	Menspesifikasi bahwa Usecase sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasi bahwa Usecase target memperluas perilaku dari Usecase sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasi paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

Sumber: Ahaddin, 2015.

b. *Activity Diagram*

Activity diagram menampilkan kegiatan dan perubahan dari satu aktivitas ke aktivitas lain dengan peristiwa yang terjadi di beberapa bagian dari sistem. *Activity diagram* merupakan state diagram khusus yang menggambarkan berbagai aliran aktivitas sistem yang sedang dirancang, sehingga mengetahui bagaimana masing-masing aliran itu berawal dan berakhir. Bahkan mungkin terjadi decision didalamnya (Pratama, 2019)

Berikut merupakan simbol-simbol pada *activity diagram* yang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Initial Node</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	<i>Activity</i>	Aktivitas dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu dari aktivitas digabungkan menjadi akhir.
	<i>Activity Final Node</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

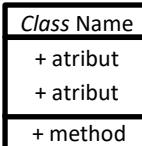
Sumber: Idris, 2018.

c. *Class Diagram*

Class diagram adalah sebuah hubungan antar class dan penjelasan detail pada tiap-tiap class di dalam model design dari suatu sistem, juga memperhatikan aturan dan tanggungjawab entitas yang menentukan perilaku sistem (Irawan, 2018).

Simbol-simbol *class diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-Simbol *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	Merupakan blok - blok pembangun pada pemrograman berorientasi obyek. Terdiri dari nama <i>class</i> , atribut dan <i>method</i> pada <i>class</i>
	<i>Association</i>	Merupakan sebuah <i>relationship</i> paling umum antara dua <i>class</i> . <i>Association</i> melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship.
	<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri <i>sendiri</i> dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi Composition terhadap <i>class</i> tempat dia bergantung tersebut.
	<i>Depedency</i>	Umumnya dependency digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.
	<i>Aggregation</i>	Mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi.

Sumber: Munjeri, 2017

d. *Sequence Diagram*

Sequence diagram adalah *diagram* yang digunakan untuk menggambarkan perilaku aktor pada sebuah sistem secara *detail* menurut waktu. *Sequence diagram* menunjukkan sejumlah contoh objek dan *message* yang diletakkan diantara objek-objek di dalam *use case* (Siregar, 2018).

Sequence diagram dapat digambarkan dengan simbol-simbol tertentu, simbol-simbol pada *sequence diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan seseorang yang berinteraksi dengan aplikasi.

Sumber: Astuti, 2017.

Tabel 2.6 Simbol-Simbol Sequence Diagram (Lanjutan)

	<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Object Message</i>	Menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
	<i>Message to Self</i>	Menggambarkan pesan/hubungan objek itu sendiri yang menunjukkan urutan kejadian yang terjadi.

Sumber: Astuti, 2017.

2.11. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu model yang menjelaskan hubungan antara data dalam database berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi (Setiadi, 2017)

Pemodelan struktur data dan hubungan antar data pada ERD digambarkan dengan beberapa notasi dan simbol yang dapat dilihat pada Tabel 2.6.

Tabel 2.7 Simbol-Simbol ERD

Simbol	Nama	Keterangan
	<i>Entity</i>	Suatu obyek yang didefinisikan dalam lingkungan pemakai dalam konteks sistem yang telah dibuat
	<i>Relation</i>	Menunjukkan adanya hubungan di antara sejumlah <i>entity</i> yang berbeda.
	<i>Atribute</i>	Mendeskripsikan sebuah karakter <i>entity</i>
	<i>Line</i>	Sebagai penghubung antara <i>entity</i> dengan <i>entity</i> maupun dengan atribut

Sumber: ryal, 2018.

Entity Relationship Diagram (ERD) hubungan dapat terdiri dari beberapa entitas yang disebut dengan derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas, sehingga kardinalitas relasi menunjukkan jumlah maksimum entitas

yang dapat berelasi dengan entitas pada himpunan entitas lain. kardinalitas yang terjadi antara diantara dua himpunan entitas dapat berupa (Setiadi, 2017):

a. *One To One*

Tipe relasi ini satu entitas diasosiasikan dengan satu entitas lainnya. Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak satu entitas pada himpunan entitas B, demikian juga sebaliknya.

b. *One To Many*

Tipe relasi ini, satu entitas diasosiasikan dengan banyak entitas lainnya. Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.

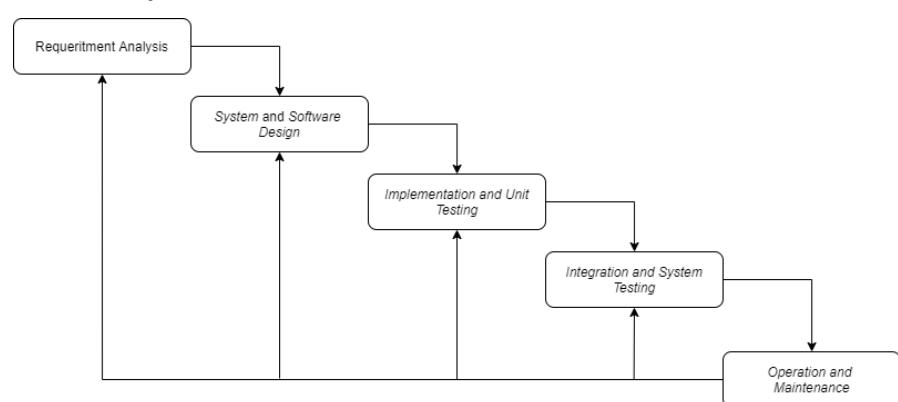
c. *Many To One*

Tipe relasi ini, banyak entitas berkaitan dengan satu entitas. Setiap entitas pada himpunan entitas A hanya dapat berelasi dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya.

d. *Many To Many*

Tipe relasi ini, terdiri banyak entitas berkaitan dengan banyak entitas yang lainnya. Yang dimana entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, sama juga dengan sebaliknya.

2.12. Metode waterfall



Gambar 2.7 Metode Waterfall (Tristianto, 2018).

Metode *waterfall* adalah suatu proses pengembangan perangkat lunak yang berurutan, dimana kemajuan terus dipandang sebagai terus mengalir kebawah seperti air terjun yang melewati fase-fase perencanaan, pemodelan, implemtasi, dan pengujian (Tristianto, 2018).

Tahap-Tahap metode *waterfall* sebagai berikut (Tristianto, 2018):

1. *Requirement analysis*

Pada langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahapan ini bisa melakukan sebuah penelitian, wawancara ataupun *study literatur*. Seorang *system* analis akan mencari informasi yang sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user*. Tahapan ini akan menghasilkan dokumen *user requirement* atau dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Kemudian dokumen inilah yang akan menjadi acuan system analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. *System and Software Design*

Pada proses *design* akan menterjemahkan syarat kebutuhan kedalam perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus kepada struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Pada tahapan ini akan menghasilkan dokumen yang disebut software requirement. Dokumen ini yang akan digunakan oleh programmer untuk melakukan aktivitas pembuatan sistemnya.

3. *Implementation and Unit Testing*

Coding merupakan penerjemah *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang telah diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan pada tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan melakukan *testing* adalah untuk menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa untuk diperbaiki.

4. *Integration and System Testing*

Pada tahapan ini bisa dikatakan *final* dalam membuat sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

5. *Operation and Maintenance*

Perangkat lunak yang sudah disampaikan kepada pelanggan akan mengalami sebuah perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau system operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

2.13. Kuesioner

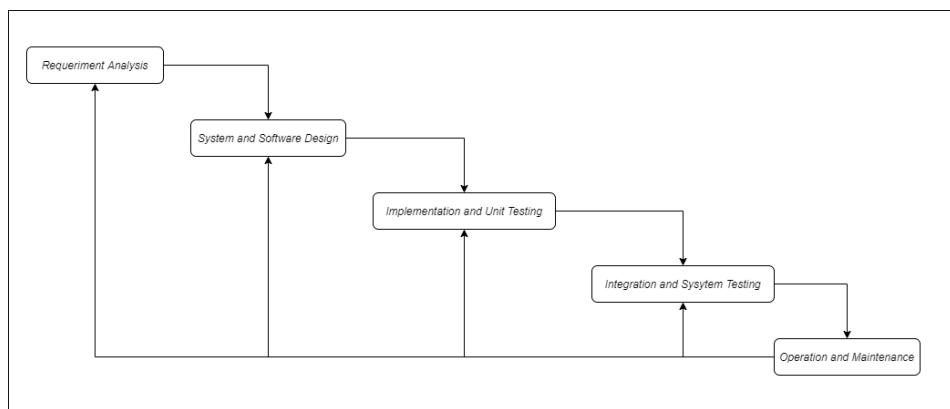
Kuesioner merupakan metode pengumpulan data yang dilakukan dengan cara memberikan pertanyaan kepada responden untuk diberikan respon sesuai dengan permintaan pengguna (Widoyoko, 2016)

BAB III

METODE PELAKSANAAN

3.1. Metode Pelaksanaan

Pembuatan Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel menggunakan metode *waterfall*. Untuk tahapan metode *waterfall* dapat dilihat pada gambar 3.1.



Gambar 3.1 Metode *Waterfall*

Penjelasan pada tahapan metode pelaksanaan dari penelitian yang telah dilakukan.

1. *Requeriment Analysis*

Hal yang pertama kali dilakukan adalah melakukan wawancara terhadap pemilik jasa Makah *Wedding Service* sebagai sumber untuk menentukan tujuan dari penelitian yang akan dilaksakan sesuai dengan kebutuhan dari mitra agar dapat bermanfaat untuk penelitian. Kemudian menakukan pengumpulan data yang dibutuhkan untuk penelitian.

2. *System and Software Design*

Setelah melakukan analisa kebutuhan yang ada pada jasa Makah *Wedding Service*. Lalu hasil dari analisa tersebut digunakan untuk membuat diagram alir data, merancang database Aplikasi Makah *Wedding Service* Berbasis *Website*.

3. *Implementation and Unit Testing*

Kemudian pada tahapan ini akan dilakukannya implementasi dari rancangan sistem yang sebelumnya telah dibuat. Implementasi pada Aplikasi Makah *Wedding Service* Berbasis *Website* terdiri dari implementasi untuk halaman *landing page*, halaman admin, halaman pegawai, dan halaman untuk *client*.

4. *Integration and System Testing*

Pada tahapan ini akan dilakukan uji coba kepada pemilik jasa Makah *Wedding Service* agar mengetahui manfaat dari sistem yang telah dibuat. Pengujian dilakukan dengan menggunakan kuesioner kepada pemilik dan pegawai Makah *Wedding Service* kemudian kepada masyarakat sebagai penguji.

5. *Operation and Maintenance*

Tahapan ini dilakukan proses pemeliharaan dan pemeriksaan secara berkala dari sistem yang sudah dibuat dengan tujuan agar mengetahui fungsi dari sistem tersebut masih bisa berjalan atau tidak.

3.2. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel menggunakan studi literatur serta observasi dan wawancara. Penelitian yang dilakukan dengan metode tersebut dapat menghasilkan data yang relevan sehingga selanjutnya dapat dilakukan analisa kebutuhan sistem serta perancangan aplikasi yang akan dibuat.

a. Studi Literatur

Pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel menggunakan studi literatur sebagai salah satu teknik yang digunakan untuk mengumpulkan data-data yang diperlukan berdasarkan sumber dari internet untuk dijadikan referensi.

b. Observasi dan Wawancara

Dalam pengumpulan data juga melelakukan observasi dan wawancara terhadap pemilik jasa Makah *Wedding Service* sebagai sumber untuk memperoleh data mengenai kebutuhan sistem dari permasalahan yang terjadi pada jasa Makah *Wedding Service* terkait informasi dan proses pemesanan pada Makah *Wedding Service*, serta bagaimana solusi untuk mengatasi permasalahan yang ada melalui suatu aplikasi berbasis *website*. Selanjutnya hasil dari pengamatan dan wawancara tersebut yang kemudian disimpulkan ke dalam proses yang terstruktur sehingga dapat diimplementasikan ke dalam Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel.

3.3. Analisis Kebutuhan

Tahapan selanjutnya setelah melakukan pengumpulan data adalah melakukan analisis kebutuhan pada aplikasi. Pada tahapan ini bertujuan untuk memastikan bahwa aplikasi ini dapat digunakan dengan tepat dan baik dengan sasaran dalam mengatasi permasalahan yang terjadi pada informasi dan proses pemesanan pada Makah *Wedding Service*. Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel. Analisis kebutuhan *website* ini terdiri dari kebutuhan *hardware* dan kebutuhan *software*.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan-kebutuhan yang memiliki keterkaitan langsung antara aplikasi dengan pengguna. Adapun kebutuhan fungsional pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel meliputi:

a. Kebutuhan Admin

Kebutuhan admin dalam Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel yaitu:

1. Melakukan *login* pada *website*.
2. Melihat halaman admin
3. Mengelola data pegawai.
4. Mengelola data *client*.
5. Mengelola data pesanan.
6. Mengelola data pembayaran.
7. Mengelola data *client*.
8. Mengelola data wilayah.
9. Mengelola data transportasi.
10. Mengelola data jadwal.
11. Mengelola data testimoni.
12. Mengelola data pengeluaran.
13. Mengelola data keuangan bulanan.
14. Mengelola data *landing page*.
15. Menghubungi *client*.
16. Melakukan *logout*.

b. Kebutuhan Pegawai

Kebutuhan pegawai dalam Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel yaitu:

1. Melakukan *login* pada website.
2. Melihat halaman pegawai
3. Mengelola data pesanan.
4. Melihat data jadwal.
5. Menghubungi *client*
6. Melakukan *logout*.

c. Kebutuhan *Client*

Kebutuhan *client* dalam Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel yaitu:

1. Melihat *landing page* aplikasi
2. Melakukan *registrasi* pada website.
3. Melakukan *login* pada website.
4. Melihat profile *client*
5. Melakukan pemesanan jasa pada Makah *Wedding Service*.
6. Melakukan pembayaran jika sudah memesan jasa Makah *Wedding Service*.
7. Melihat paket-paket yang tersedia.
8. Mengajukan jadwal untuk melakukan konsultasi ke admin.
9. Memberikan testimoni.
10. Melakukan *logout*

3.3.2 Kebutuhan *Non Fungsional*

Pada Kebutuhan *non fungsional* memaparkan kebutuhan perangkat keras (*hardware*) dan (*software*) diantaranya yaitu:

a. Kebutuhan pengkak keras (*hardware*)

Kebutuhan *hardware* yang dibutuhkan untuk pembuatan Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan *Hardware*

No	Jenis	Kebutuhan	Keterangan
1	<i>Processor</i>	AMD Radeon Graphics	Digunakan untuk mendukung berjalannya aplikasi pembuat sistem
2	RAM	4 GB	Minimal
3	Monitor	14inch	Untuk menampilkan <i>output</i> data secara grafis.

b. Kebutuhan perangkat lunak (*software*)

Kebutuhan perangkat lunak adalah kebutuhan yang harus dimiliki oleh perangkat lunak untuk memenuhi kebutuhan pengguna. Kebutuhan perangkat lunak dalam Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* ini dapat dilihat pada Tabel 3.2

Tabel 3.2 Kebutuhan *Software*

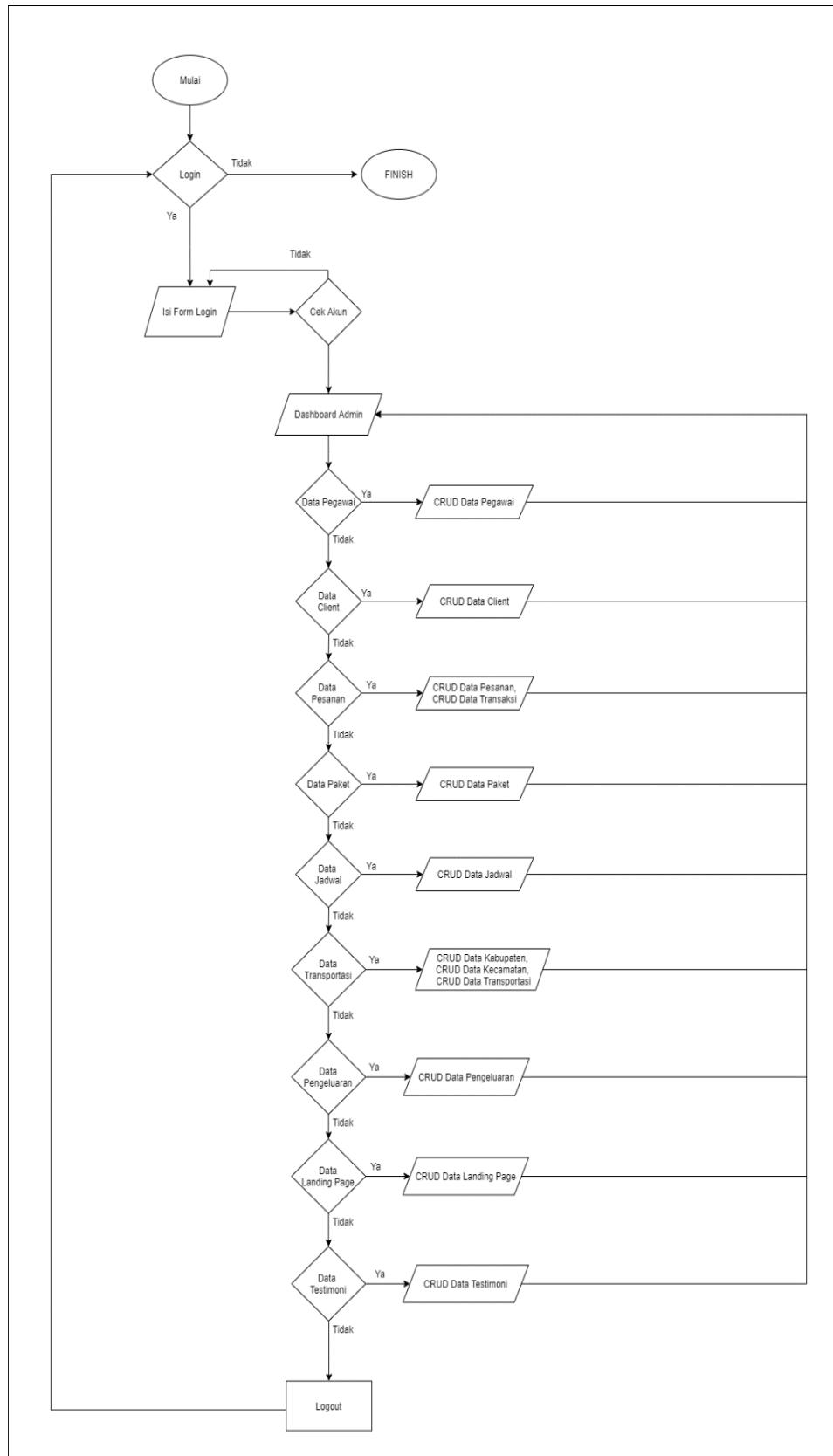
No	Jenis	Kebutuhan
1	Perangkat lunak pengolah	Visual Studio Code
2	Bahasa <i>Scripting</i>	PHP, HTML 5, CSS
3	Penyimpanan Data	Database (MySQL)
4	Web Browser	Chrome, Microsoft Edge, dan lainnya

3.4. Flowchart

Flowchart dibuat untuk menjelaskan alur pada sistem Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel. Terdapat *flowchart* admin, *flowchart* pegawai dan *flowchart* clinet Pada Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel .

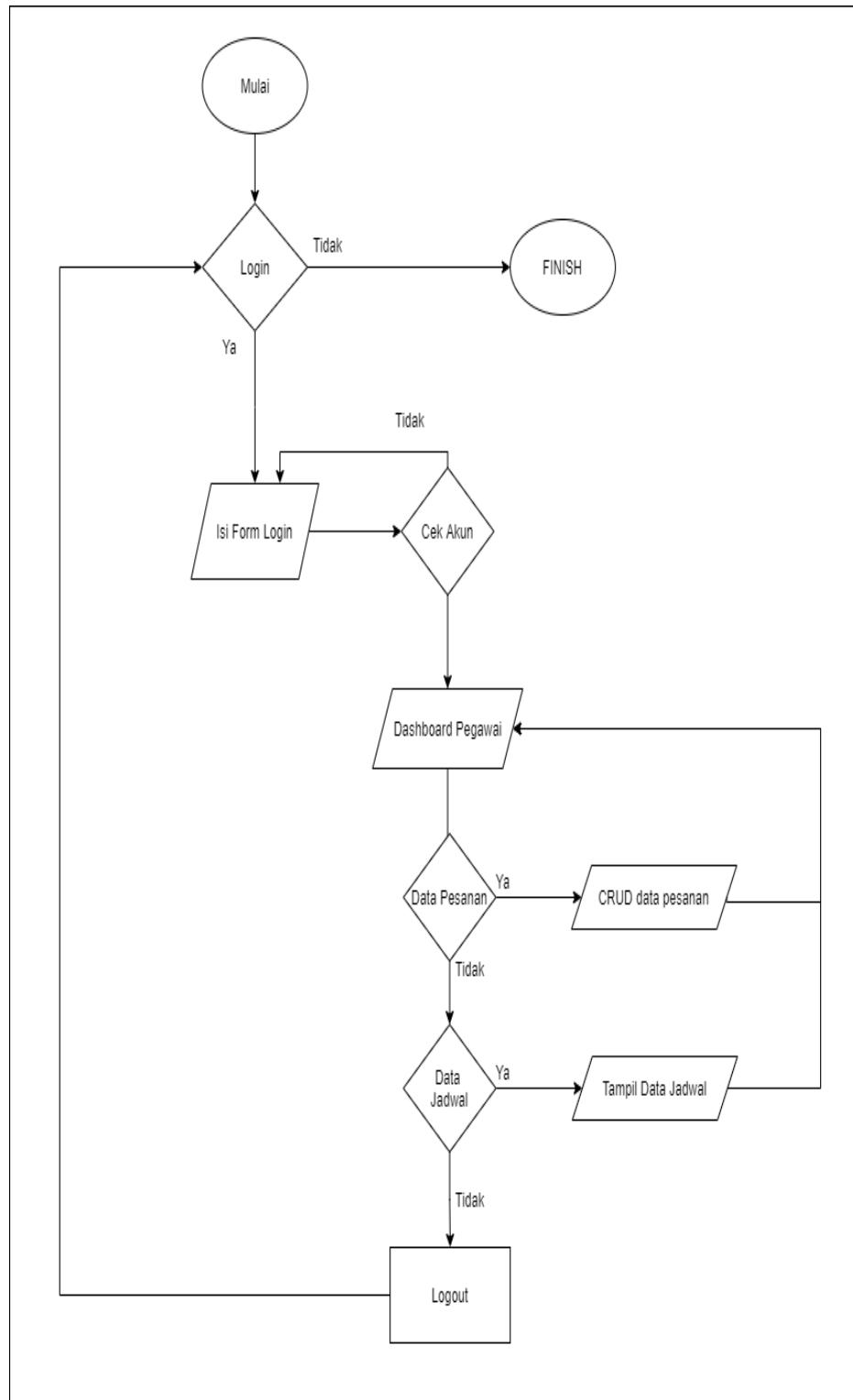
a. *Flowchart* Admin

Flowchart admin pada Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel dapat dilihat pada Gambar 3.2.

Gambar 3.2 *Flowchart Admin*

b. *Flowchart Pegawai*

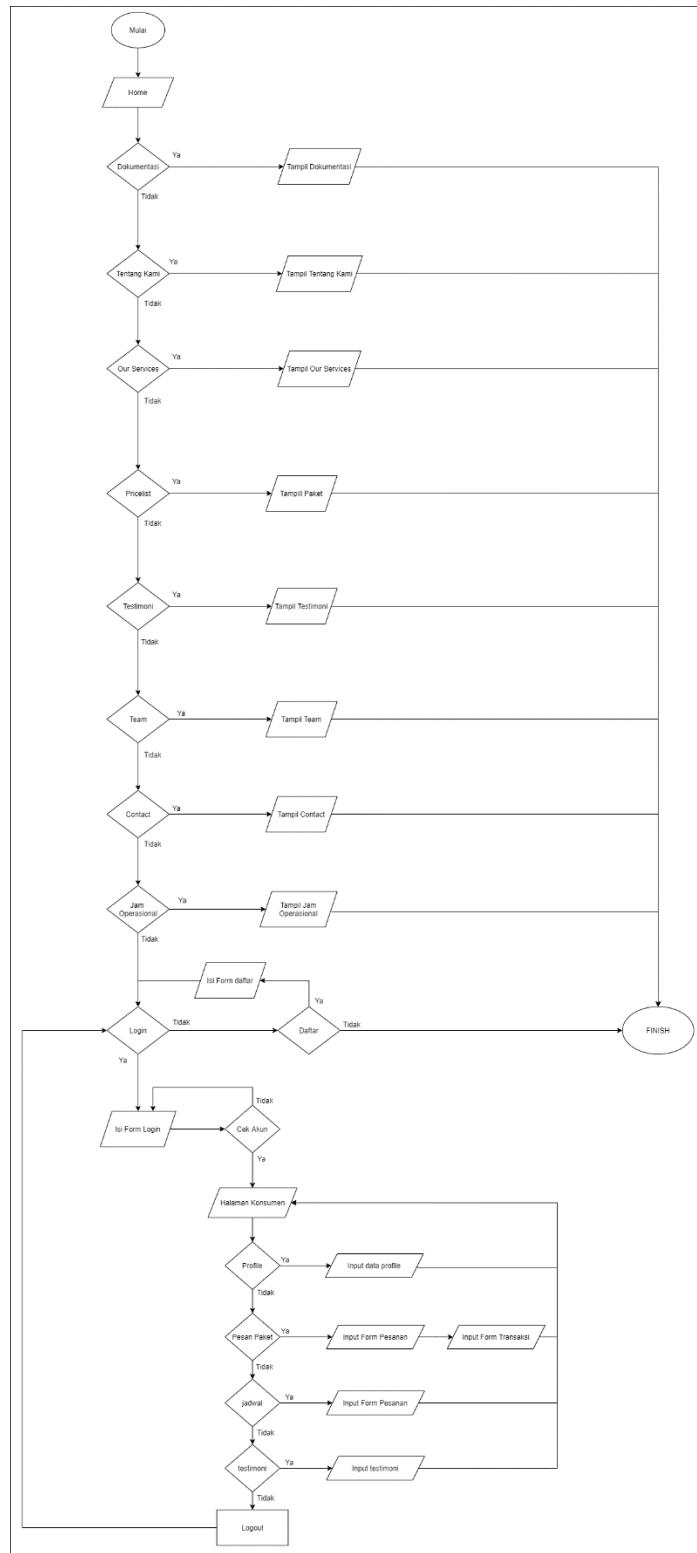
Flowchart pegawai pada Aplikasi Makah Wedding Service Berbasis Website Dengan Framework Laravel dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Flowchart Pegawai*

c. *Flowchart client*

Flowchart client pada Aplikasi Makah Wedding Service Berbasis Website Dengan Framework Laravel dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Flowchart Client*

3.5. Perancangan Sistem

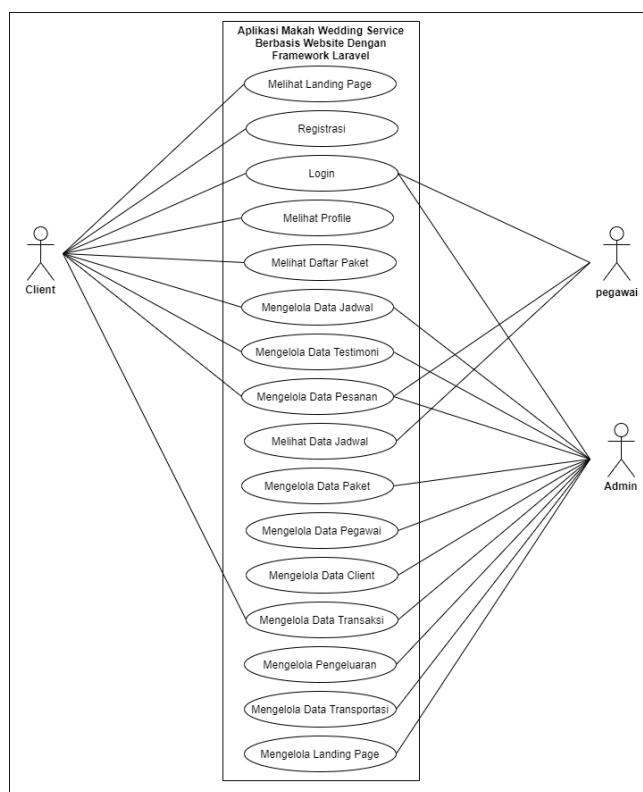
Perancangan sistem akan menjelaskan mengenai rancangan dalam proses Aplikasi Makah Wedding Service Berbasis Website Dengan *Framework* Laravel. Perancangan ini akan menjelaskan mengenai bagaimana sistem akan berjalan sesuai dengan kebutuhan yang dibutuhkan untuk mengatasi permasalahan yang terjadi. Perancangan sistem aplikasi ini akan dijelaskan melalui UML.

35.1. Rancangan UML (*Undefined Modelling Language*)

UML akan menyesuaikan *design* sistem dengan suatu bahasa yang sudah menjadi standar perancangan dan pendokumentasian sistem *software*. Aplikasi Makah Wedding Service Berbasis Website Dengan *Framework* Laravel ini dijelasakan melalui tiga jenis UML, yaitu terdiri dari *use case diagram*, *activity diagram*, dan *sequence diagram*.

a. *Use Case Diagram*

Use case diagram digunakan untuk menampilkan gambaran terkait interaksi *actor* dan sistem untuk memperkenalkan sistem yang akan dibangun, digunakan untuk menjelaskan bagaimana langkah-langkah yang dikerjakan oleh sistem. *Use case* dapat dilihat pada Gambar 3.5.



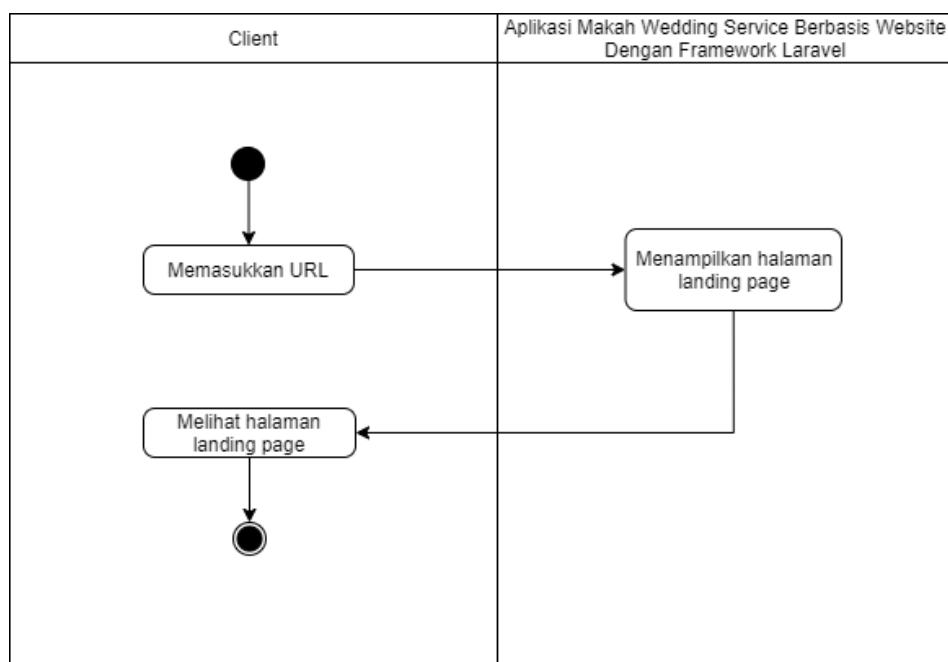
Gambar 3.5 *Use Case Diagram*

b. *Activity Diagram*

Activity diagram merupakan *state diagram* khusus yang menggambarkan berbagai aliran aktivitas didalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal dan berakhir. Pada Aplikasi Makah *Wedding Service Berbasis Website Dengan Framework Laravel* terdiri dari beberapa *activity diagram*, yaitu sebagai berikut:

1. *Activity Diagram Melihat Landing Page Oleh Client*

Client dapat melihat halaman *landing page* apabila *client* memasukkan URL, lalu *client* akan diarahkan pada halaman *landing page* dan *client* bisa melihat halaman *landing page*. *Activity diagram* melihat *landing page* oleh *client* dapat dilihat pada Gambar 3.6.

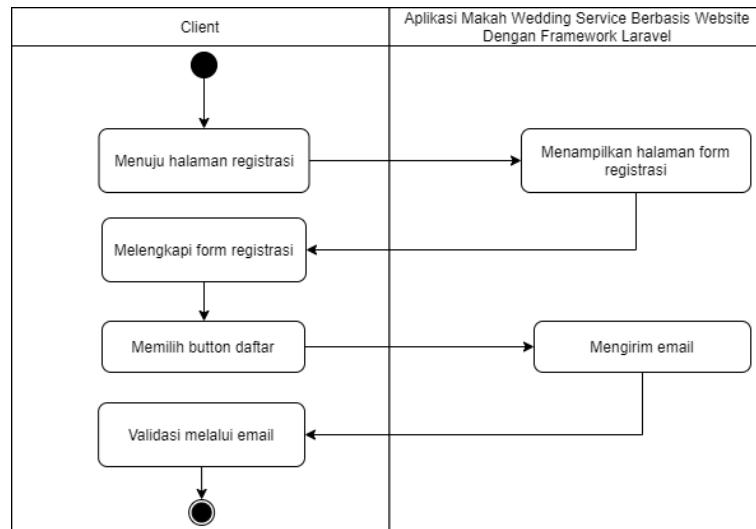


Gambar 3.6 *Activity Diagram Melihat Landing Page*

2. *Activity Diagram Registrasi Oleh Client*

Peroses *registrasi* dapat dilakukan apabila *client* telah menuju halaman *registrasi*, kemudian *client* melengkapi *form registrasi* yang telah disediakan, setelah melengkapi *form registrasi*, *client* memilih button daftar, apabila *client* sudah melengkapi *form registrasi* maka *client* akan mendapatkan pesan pada email untuk validasi akun *client* dan *client* melakukan validasi akun dengan cara memilih

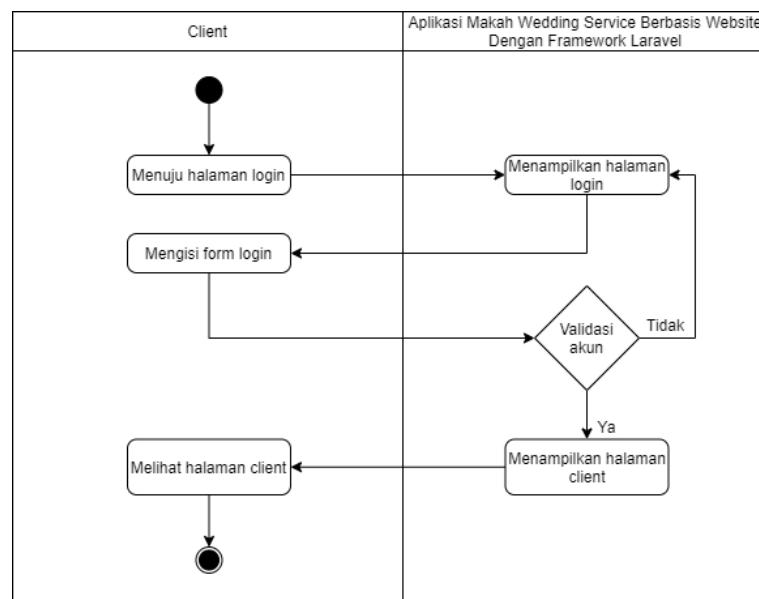
link yang ada pada pesan email. *Activity diagram registrasi* oleh *client* dapat dilihat pada Gambar 3.7.



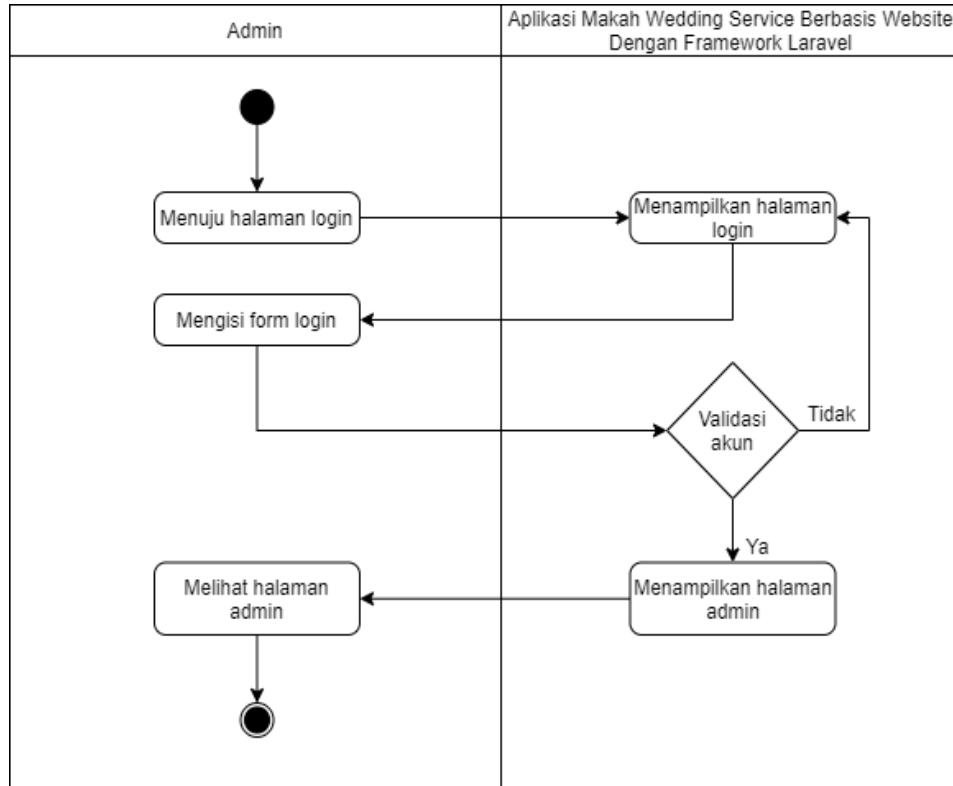
Gambar 3.7 *Activity Diagram Registrasi Client*

3. *Activity Diagram Login* Oleh *Client*, Admin, Pegawai

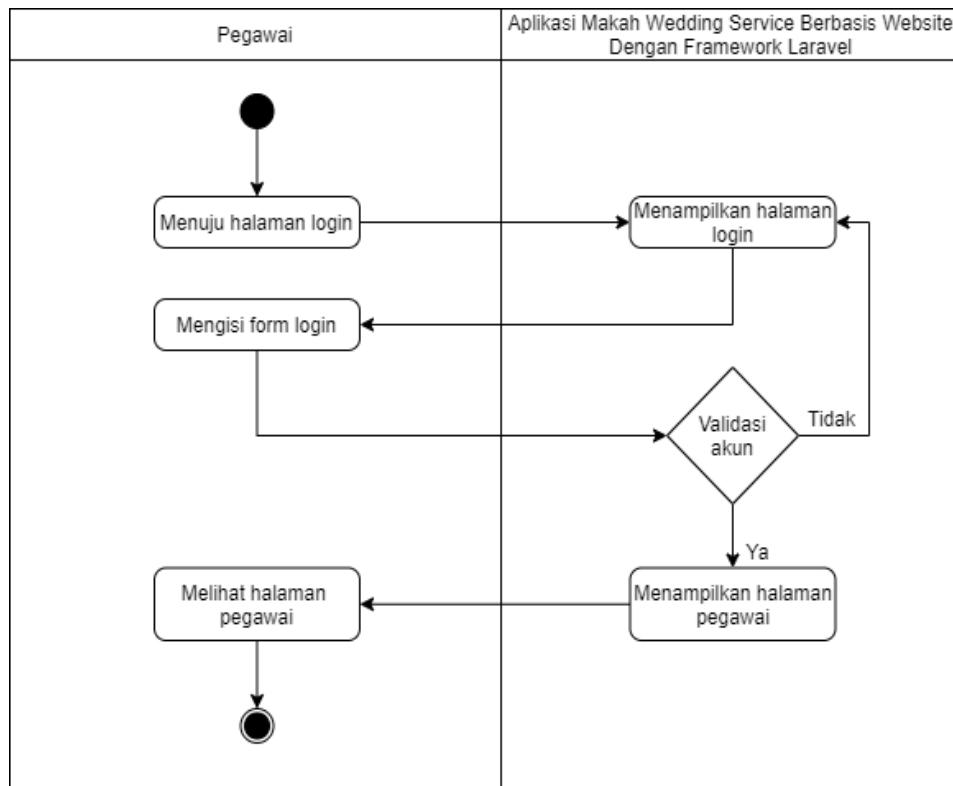
Proses *login* pengguna dimulai dari pengguna menuju halaman *login*, kemudian pengguna melengkapi *form login*, apabila pengguna berhasil melakukan *login* maka akan muncul halaman pengguna. *Activity diagram login* oleh *client* dapat dilihat pada Gambar 3.8, *Activity diagram login* oleh admin dapat dilihat pada Gambar 3.9, dan *Activity diagram login* oleh pegawai dapat dilihat pada Gambar 3.10.



Gambar 3.8 *Activity Diagram Login Client*



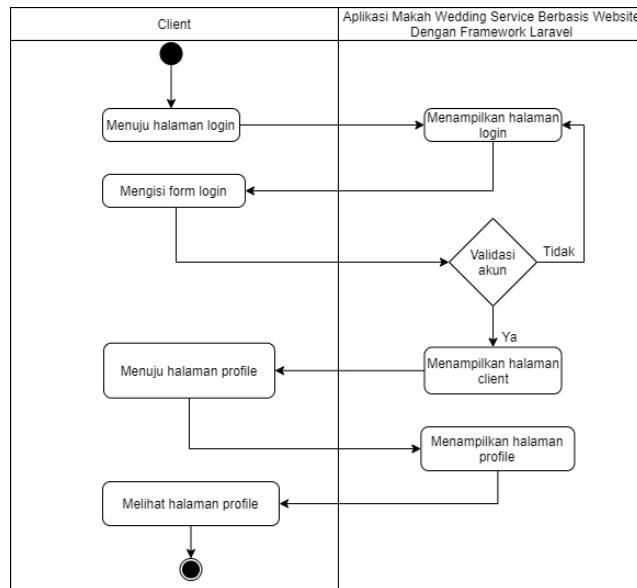
Gambar 3.9 Activity Diagram Login Admin



Gambar 3.10 Activity Diagram Login Pegawai

4. Activity Diagram Melihat Profile Oleh Client

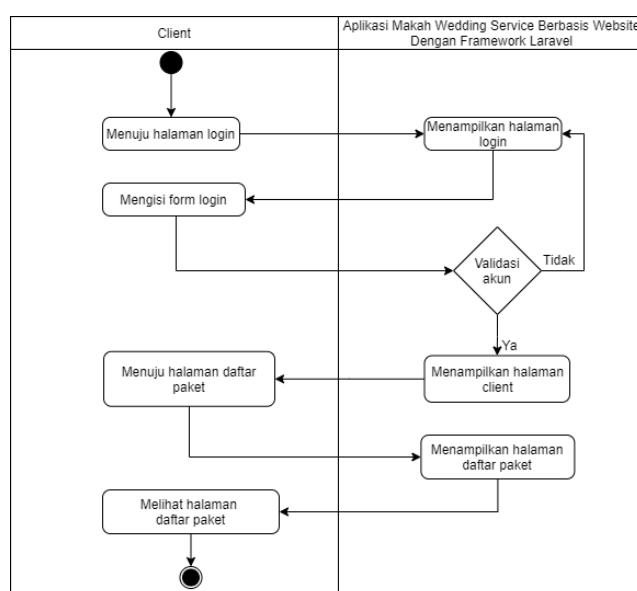
Profile *client* dapat dilihat apabila *client* telah berhasil melakukan login, lalu *client* menuju pada halaman profile, selanjutnya *client* dapat melihat halaman profile *client*. *Activity diagram* profile oleh *client* dapat dilihat pada Gambar 3.11.



Gambar 3.11 *Activity Diagram* Melihat Profile

5. Activity Diagram Melihat Daftar Paket Oleh Client

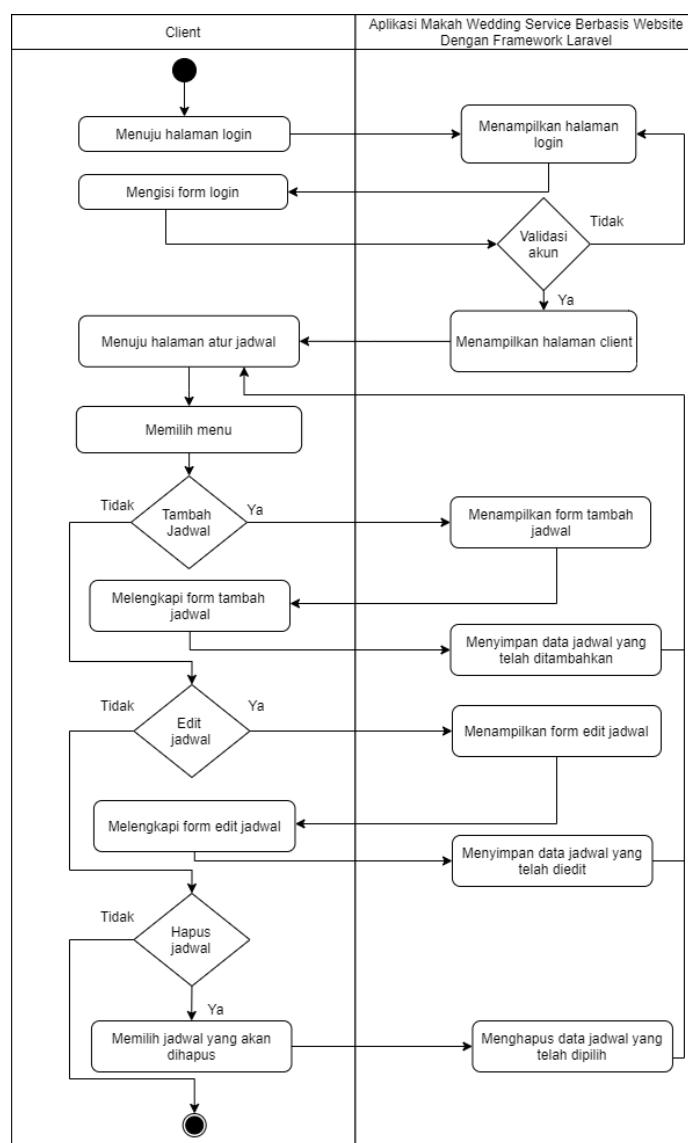
Client dapat melihat daftar paket apabila *client* telah berhasil melakukan *login*, kemudian menuju pada halaman daftar paket, lalu *client* dapat melihat halaman daftar paket. *Activity diagram* daftar paket oleh *client* dapat dilihat pada Gambar 3.12.



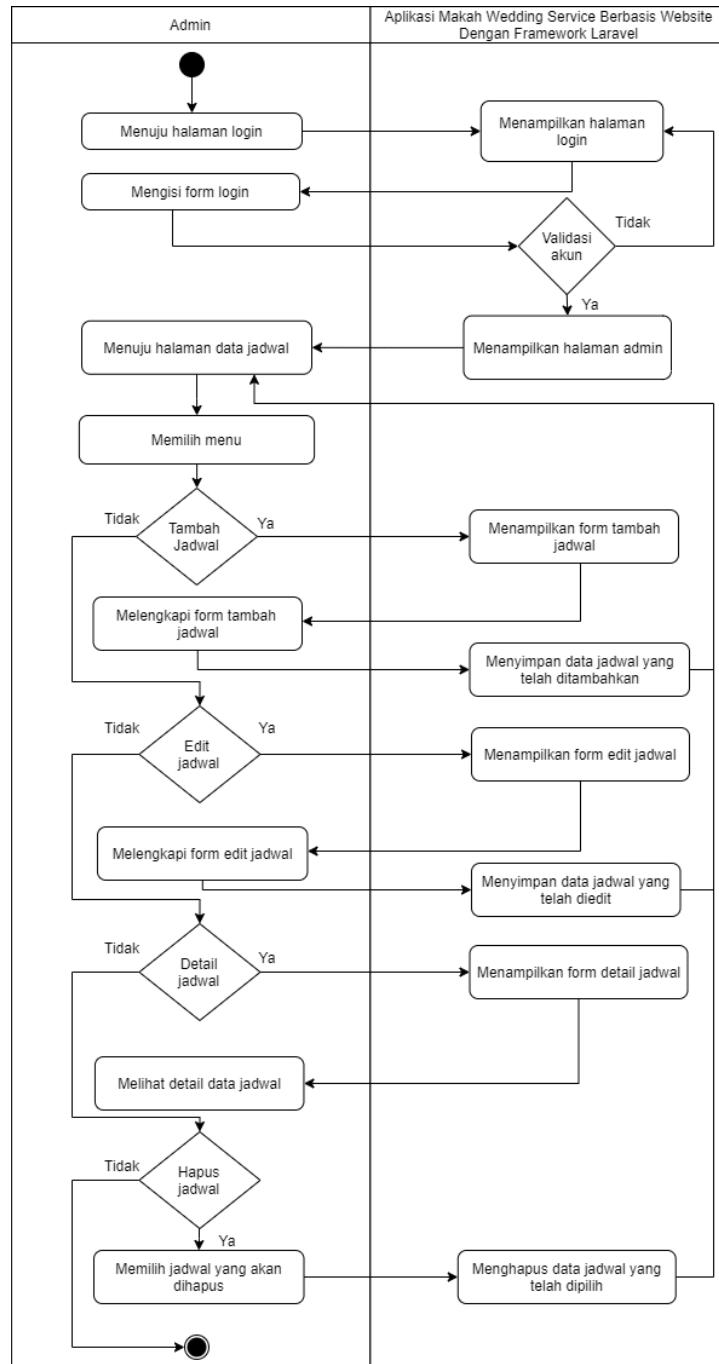
Gambar 3.12 *Activity Diagram* Melihat Daftar Paket

6. Activity Diagram Mengelola Data Jadwal Oleh Client dan Admin

Proses mengelola data jadwal bisa dilakukan jika pengguna telah berhasil melakukan *login*, kemudian menuju pada halaman data jadwal, lalu memilih menu tambah jadwal, edit jadwal, dan hapus jadwal. Jika memilih menu tambah jadwal maka pengguna melengkapi *form* tambah jadwal, jika memilih menu edit jadwal maka pengguna melengkapi *form* edit jadwal, dan jika memilih menu jadwal hapus maka data jadwal akan dihapus. *Activity diagram* mengelola data jadwal oleh *client* dapat dilihat pada Gambar 3.13 dan *Activity diagram* mengelola data jadwal oleh admin dapat dilihat pada Gambar 3.14.



Gambar 3.13 *Activity Diagram* Mengelola Data Jadwal Client

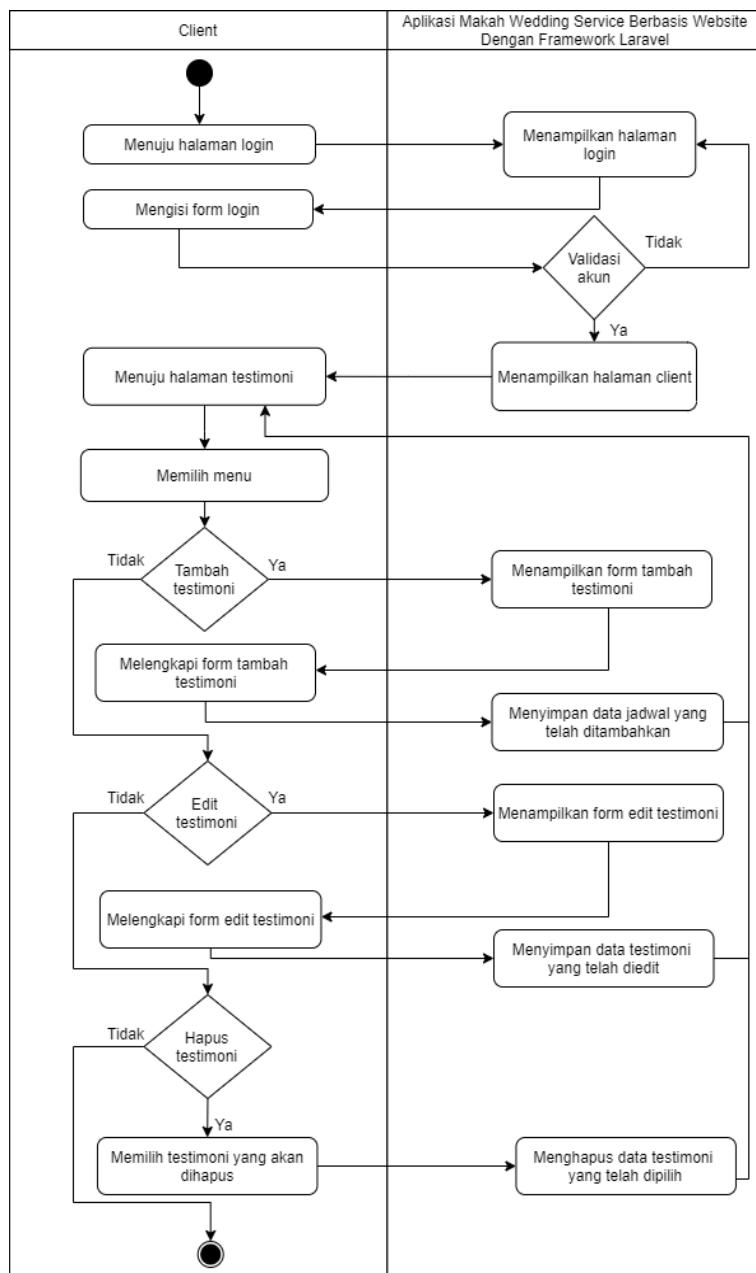


Gambar 3.14 *Activity Diagram* Mengelola Data Jadwal Admin

7. *Activity Diagram* Mengelola Data Testmoni Oleh Client

Client bisa mengelola testimoni jika sudah berhasil melakukan *login* dan menuju pada halaman testimoni, selanjutnya memilih menu tambah testimoni, edit testimoni, dan hapus testimoni. Bila memilih tambah testimoni maka akan *client* akan melengkapi *form* tambah testimoni, bila memilih edit testimoni maka *client* akan melengkapi *form* edit testimoni, bila memilih hapus maka data testimoni yang

dipilih untuk dihapus akan terhapus. *Activity diagram* mengelola data testimoni oleh *client* dapat dilihat pada Gambar 3.15.

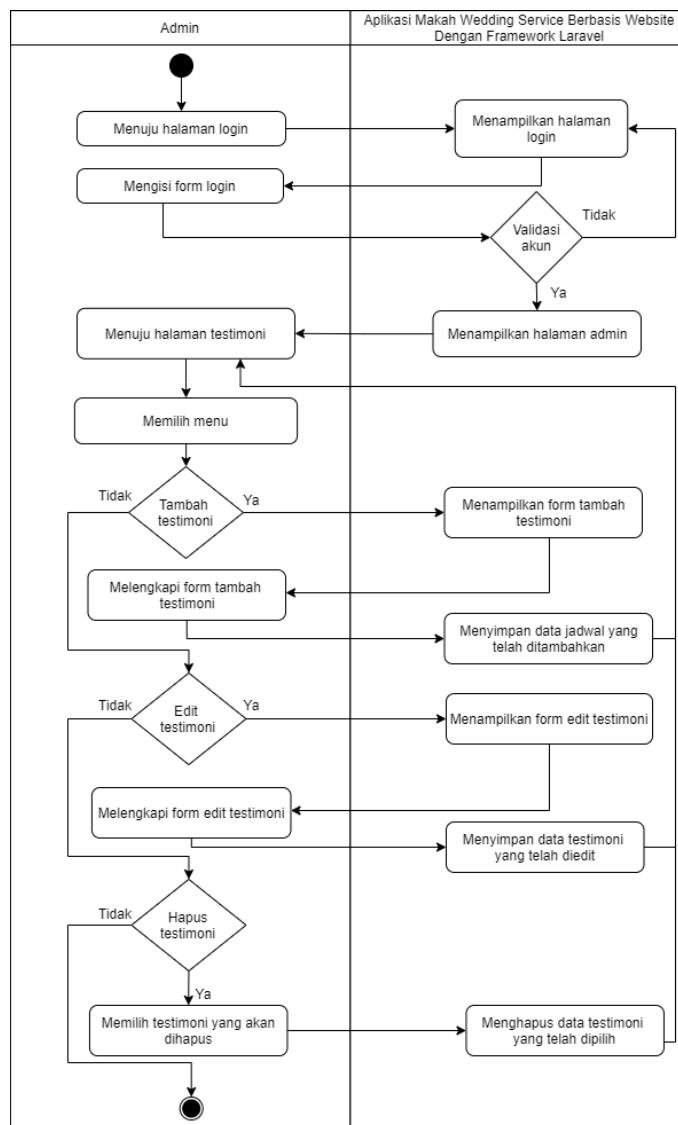


Gambar 3.15 *Activity Diagram* Mengelola Data Testimoni *Client*

8. *Activity Diagram* Mengelola Data Testmoni Oleh Admin

Pada proses mengelola data testimoni oleh admin bisa dilakukan ketika admin sudah berhasil melakukan *login*, kemudian admin menuju halaman testimoni dan memilih menu tambah testimoni, edit testimoni, hapus testimoni. Ketika memilih menu tambah testimoni maka admin akan melengkapi *form* tambah testimoni, ketika memilih menu edit maka admin akan melengkapi *form* edit testimoni, ketika

admin memilih hapus testimoni maka data yang telah dipilih akan terhapus. *Activity diagram* mengelola data testimoni oleh admin dapat dilihat pada Gambar 3.16.

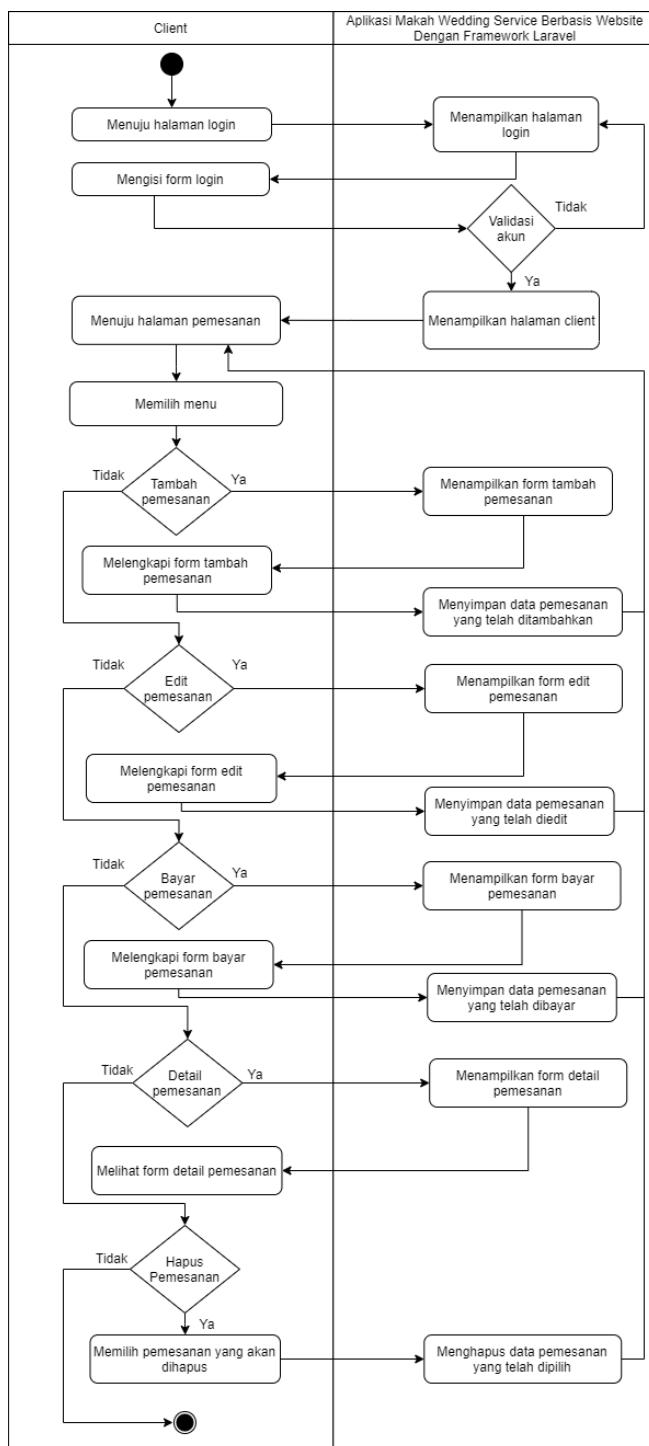


Gambar 3.16 *Activity Diagram* Mengelola Data Testimoni Admin

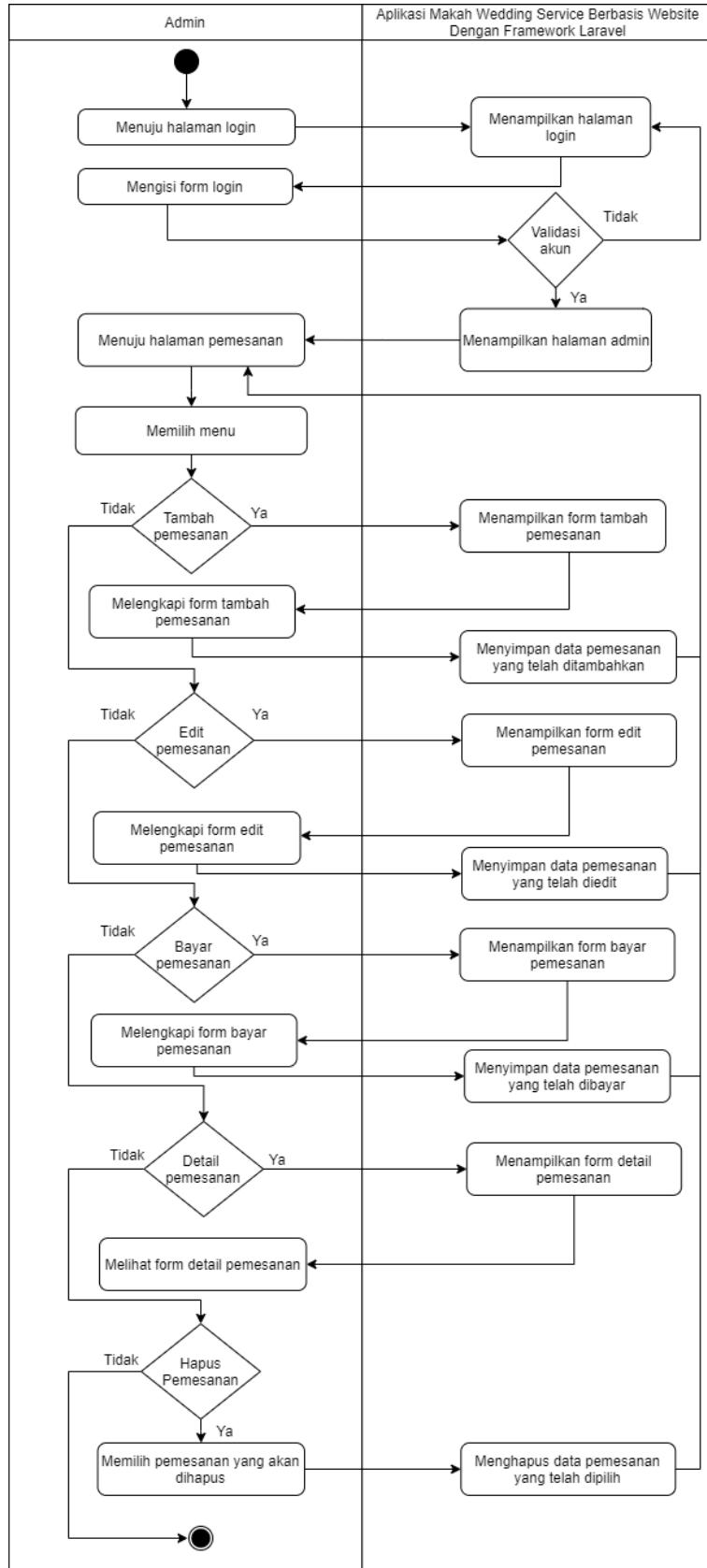
9. *Activity Diagram* Mengelola Data Pemesanan Oleh *Client* dan Admin

Proses pemesanan dilakukan ketika pengguna telah berhasil melakukan *login* dan menuju halaman pemesanan. Kemudian pengguna memilih menu yang terdiri dari tambah pemesanan, edit pemesanan, bayar pemesanan, detail pemesanan., dan hapus pemesanan. Apabila pengguna memilih tambah pemesanan maka akan muncul *form* tambah pemesanan yang dilengkapi oleh pengguna, apabila pengguna memilih edit pemesanan maka akan muncul *form* edit pemesanan yang dilengkapi oleh pengguna, apabila pengguna memilih bayar pemesanan maka akan muncul *form* bayar pemesanan yang dilengkapi oleh pengguna, apabila pengguna memilih

detail pemesanan maka akan muncul *form detail* data pemesanan, dan apabila pengguna memilih hapus data pemesanan maka data pemesanan yang telah dipilih akan terhapus. *Activity diagram* mengelola data pesanan oleh *client* dapat dilihat pada Gambar 3.17 dan *Activity diagram* mengelola data pesanan oleh admin dapat dilihat pada Gambar 3.18.



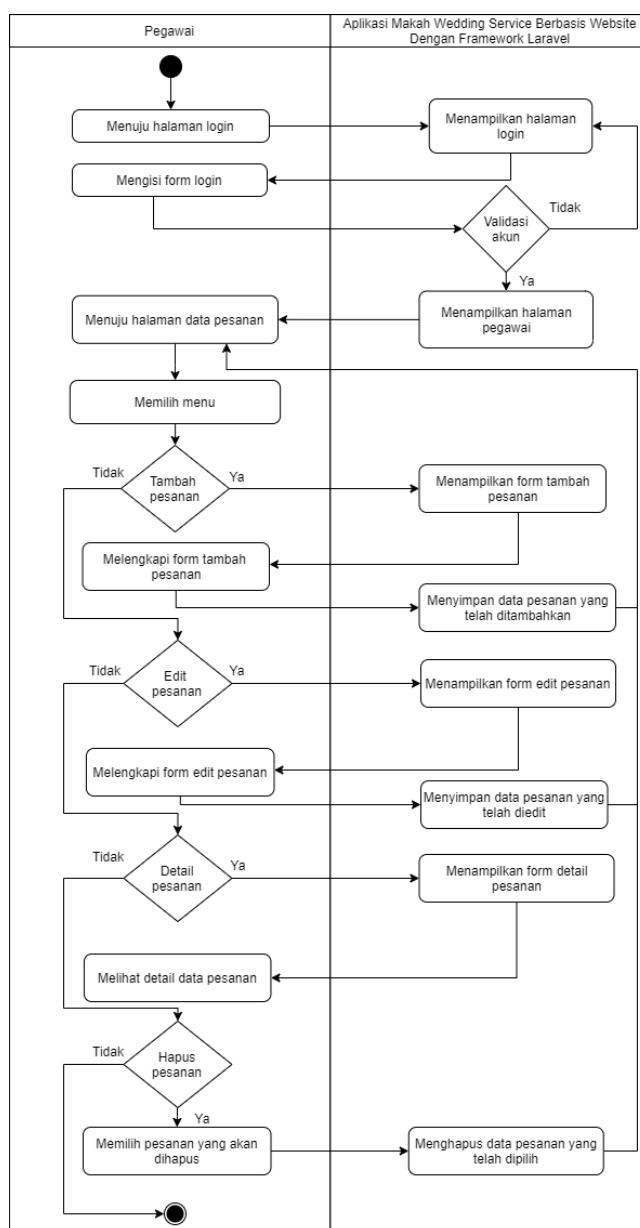
Gambar 3.17 *Activity Diagram* Mengelola Data Pemesanan *Client*



Gambar 3.18 Activity Diagram Mengelola Data Pemesanan Admin

10. Activity Diagram Mengelola Data Pesanan Oleh Pegawai

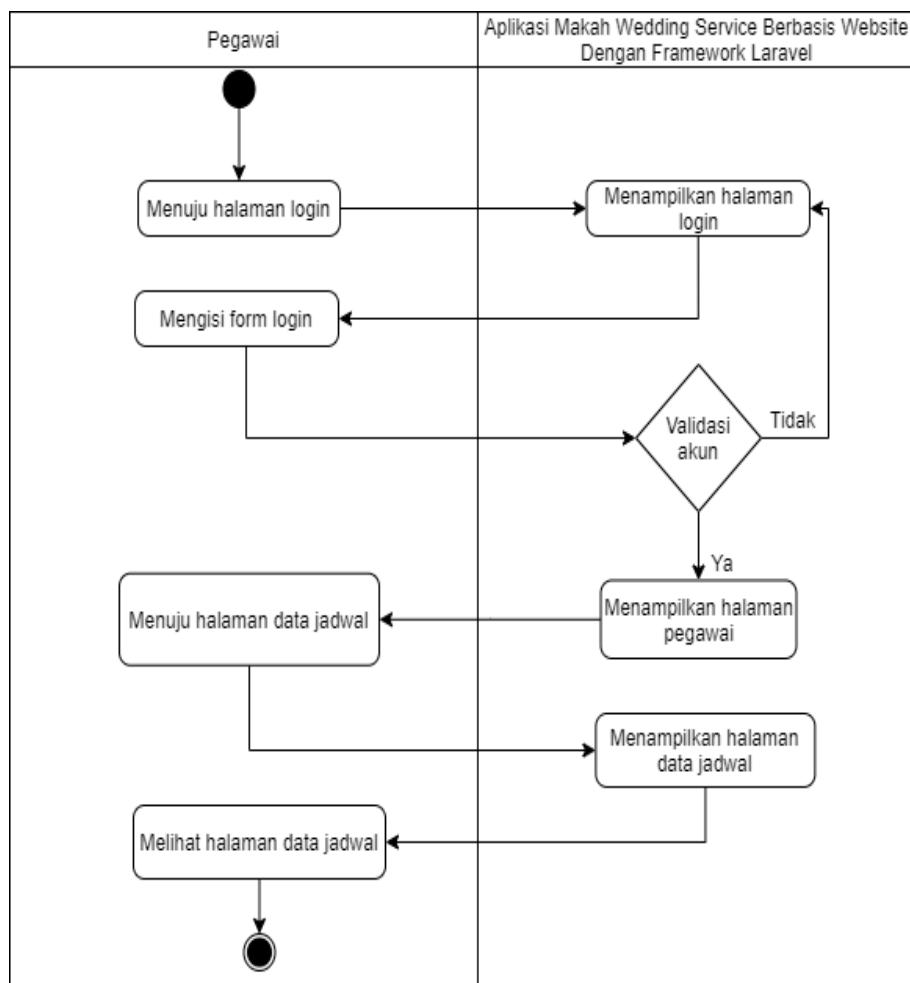
Pegawai bisa mengelola data pesanan apabila sudah berhasil melakukan *login* dan menuju pada halaman data pesanan. Kemudian pegawai memilih menu tambah pesanan, edit pesanan, detail pesanan, dan hapus pesanan. Jika pegawai memilih tambah pesanan maka melengkapi *form* tambah pesanan, jika pegawai memilih edit pesanan maka akan melengkapi *form* edit pesanan, jika pegawai memilih detail pesanan maka akan melihat *form detail* pesanan, dan jika memilih hapus pesanan maka data pesanan yang telah dipilih akan terhapus. *Activity diagram* mengelola data pesanan oleh pegawai dapat dilihat pada Gambar 3.19.



Gambar 3.19 *Activity Diagram* Mengelola Data Pesanan Pegawai

11. *Activity Diagram* Melihat Data Jadwal Oleh Pegawai

Data jadwal bisa dilihat oleh pegawai jika pegawai sudah berhasil melakukan *login*, kemudian menuju pada halaman data jadwal, setelah itu pegawai dapat melihat data jadwal. *Activity diagram* melihat data jadwal oleh pegawai dapat dilihat pada Gambar 3.20.

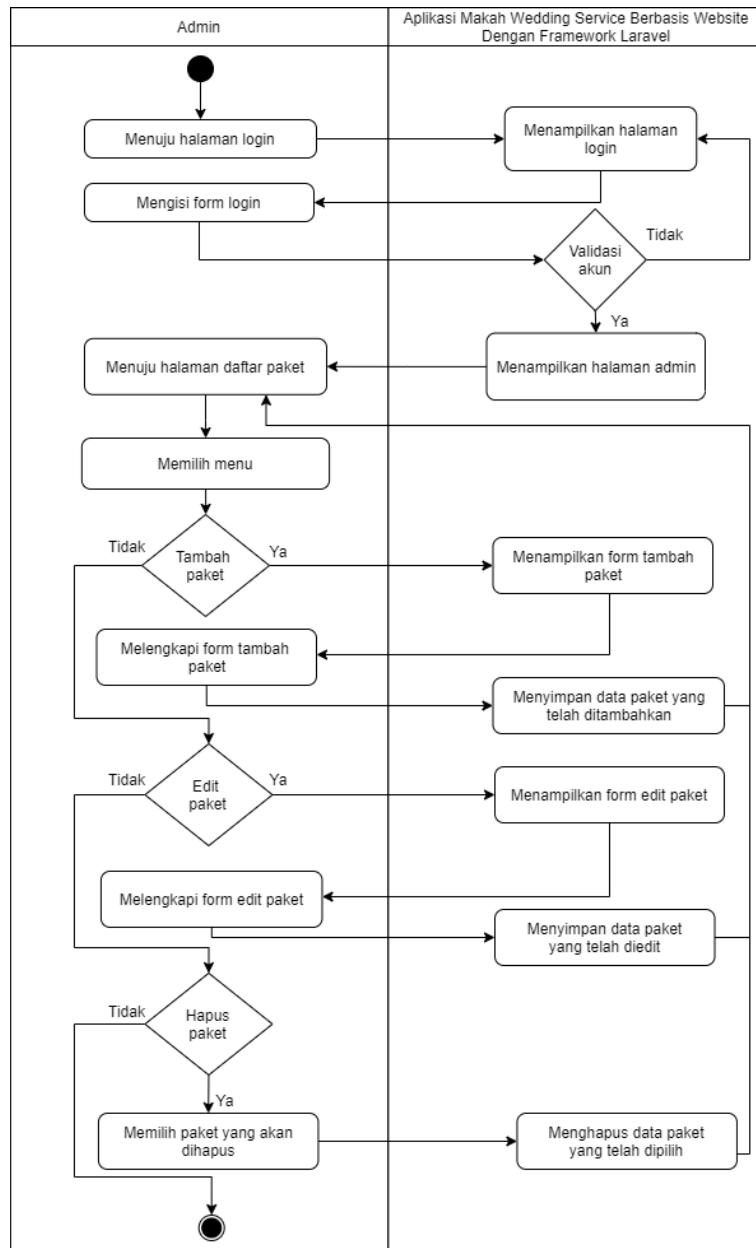


Gambar 3.20 *Activity Diagram* Melihat Data Jadwal Pegawai

12. *Activity Diagram* Mengelola Data Paket Oleh Admin

Admin dapat mengelola data paket ketika sudah melakukan login lalu menuju pada halaman data paket, kemudian admin memilih menu tambah paket, edit paket, dan hapus paket. Bila admin memilih tambah data paket maka muncul form tambah data paket yang akan dilengkapi oleh admin, Bila admin memilih edit data paket maka muncul form edit data paket yang akan dilengkapi oleh admin, dan Bila admin memilih hapus data paket maka data paket yang telah dipilih untuk

dihapus akan terhapus. *Activity diagram* mengelola data paket oleh admin dapat dilihat pada Gambar 3.21.

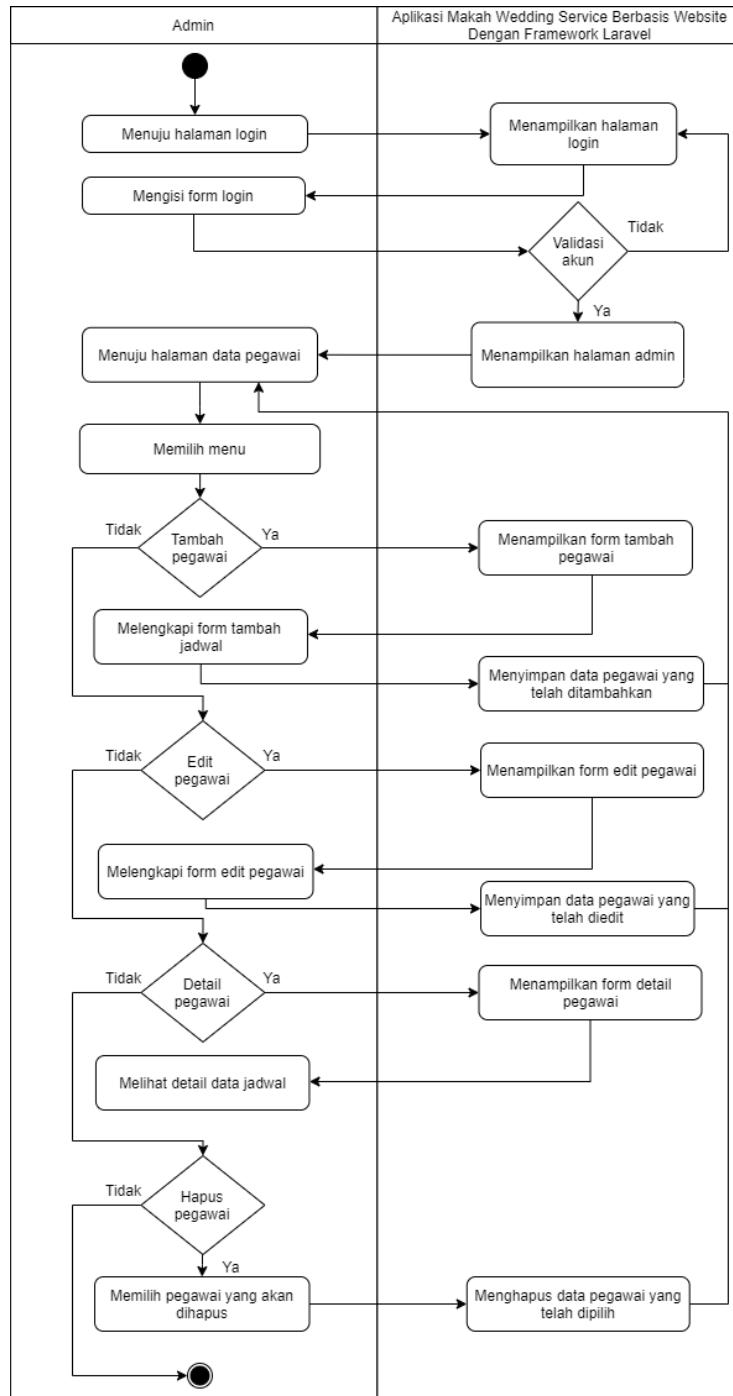


Gambar 3.21 *Activity Diagram* Mengelola Data Paket Admin

13. *Activity Diagram* Mengelola Data Pegawai Oleh Admin

Proses mengelola data pegawai dapat dilakukan ketika admin sudah berhasil melakukan *login*. Lalu admin menuju pada halaman data pegawai, setelah itu admin memilih menu pada data pegawai yang terdiri dari tambah pegawai, edit pegawai, *detail* pegawai, dan hapus pegawai. Apabila admin memilih menu tambah pegawai maka admin akan melengkapi *form* tambah data pegawai, apabila admin memilih

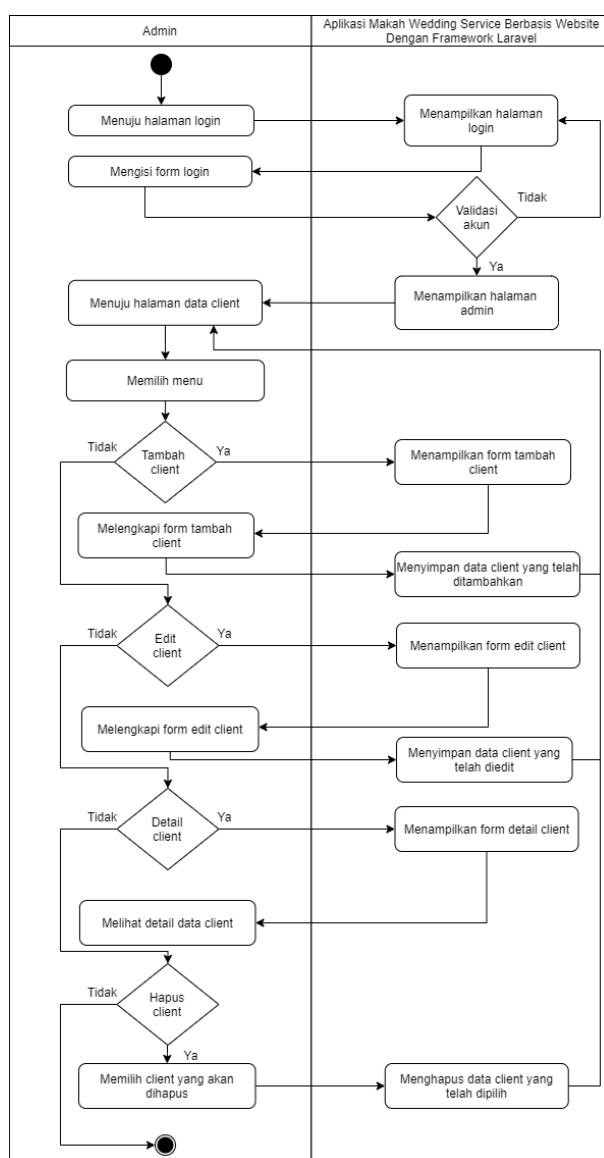
menu edit data pegawai maka admin akan melengkapi *form* edit data pegawai, apabila admin memilih menu detail pegawai maka akan menampilkan *form detail* pegawai, dan apabila admin memilih hapus data pegawai maka data pegawai yang telah dipilih untuk dihapus akan terhapus. *Activity diagram* mengelola data paket oleh admin dapat dilihat pada Gambar 3.22.



Gambar 3.22 *Activity Diagram* Mengelola Data Pegawai Admin

14. Activity Diagram Mengelola Data Client Oleh Admin

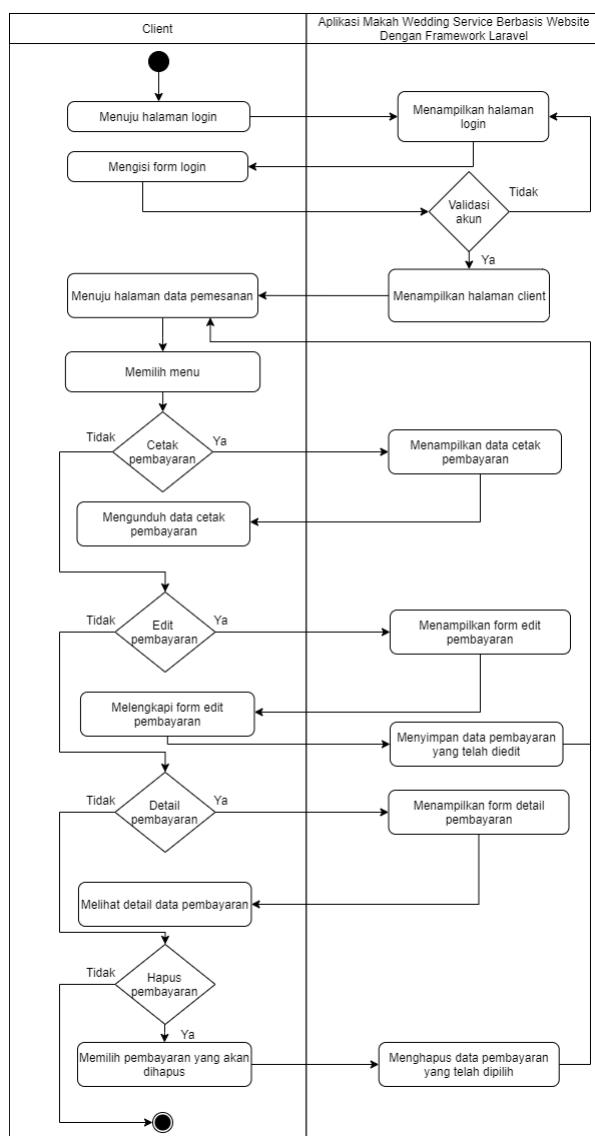
Admin dapat mengelola data *client* jika berhasil melakukan *login*. Kemudian menuju pada halaman data *client*. Setelah itu admin akan memilih menu tambah *client*, edit *client*, detail *client*, dan hapus *client*. Ketika admin memilih menu tambah *client* maka admin akan melengkapi *form* tambah data *client*, ketika admin memilih edit *client* maka admin akan melengkapi *form* edit data *client*, ketika admin memilih detail *client* maka admin akan melihat *form detail client*, dan ketika admin memilih hapus *client* maka data *client* yang telah dipilih untuk dihapus akan terhapus. *Activity diagram* mengelola data *client* oleh admin dapat dilihat pada Gambar 3.23.



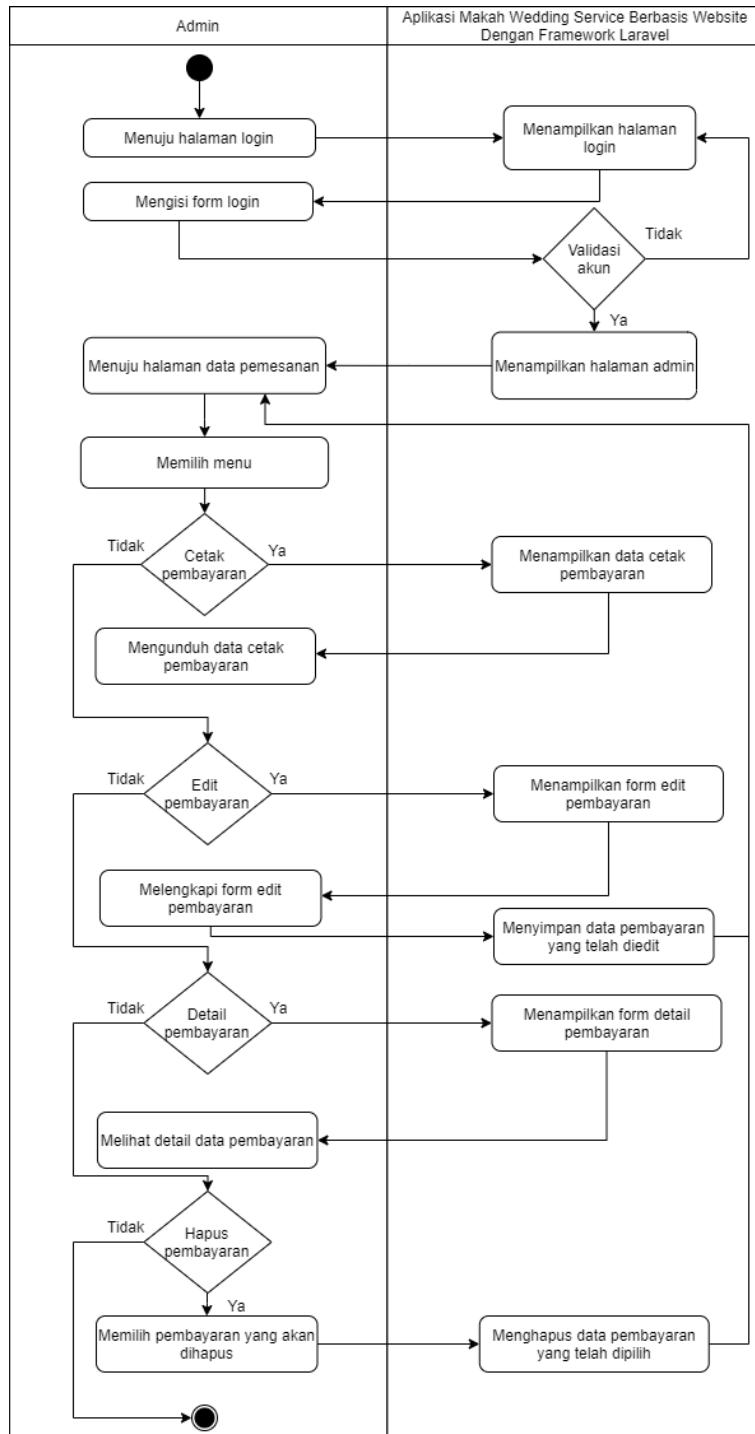
Gambar 3.23 *Activity Diagram* Mengelola Data *Client* Admin

15. *Activity Diagram* Mengelola Data Pembayaran Oleh *Client* dan Admin

Proses pembayaran dapat dilakukan apabila pengguna telah melakukan *login*. Setelah itu menuju pada halaman pemesanan dan menu cetak pembayaran, edit pembayaran, detail pembayaran, dan hapus pembayaran. Jika pengguna memilih menu cetak data pembayaran maka *client* akan mengunduh data pembayaran, jika pengguna memilih data edit pembayaran maka pengguna akan melengkapi *form* edit data pembayaran, jika pengguna memilih *detail* pembayaran maka akan menampilkan *form detail* data pembayaran, jika pengguna memilih hapus pembayaran maka data akan terhapus. *Activity diagram* mengelola data pembayaran oleh *client* dapat dilihat pada Gambar 3.24 dan *Activity diagram* mengelola data pembayaran oleh admin dapat dilihat pada Gambar 3.25.



Gambar 3.24 *Activity Diagram* Mengelola Data Pembayaran *Client*

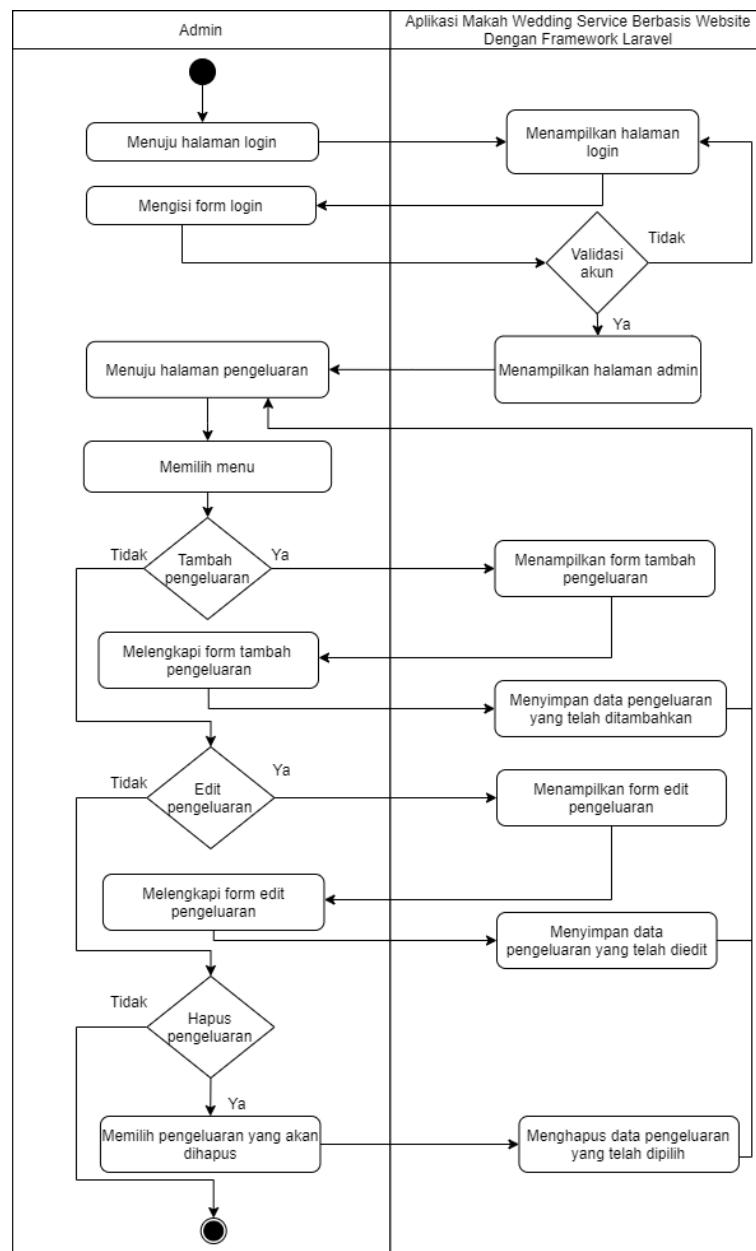


Gambar 3.25 *Activity Diagram* Mengelola Data Pembayaran Admin

16. *Activity Diagram* Mengelola Data Pengeluaran Oleh Admin

Admin dapat mengelola data pengeluaran apabila sudah berhasil melakukan *login* dan menuju pada halaman data pengeluaran, kemudian admin memilih menu tambah pengeluaran, edit pengeluaran, dan hapus pengeluaran. Bila admin memilih menu tambah pengeluaran maka admin akan melengkapi *form* tambah pegawai, bila

admin memilih edit pengeluaran maka admin akan melengkapi *form* edit data pegawai, dan jika admin memilih menu hapus pengeluaran maka data akan terhapus. *Activity diagram* mengelola data pengeluaran oleh admin dapat dilihat pada Gambar 3.26.

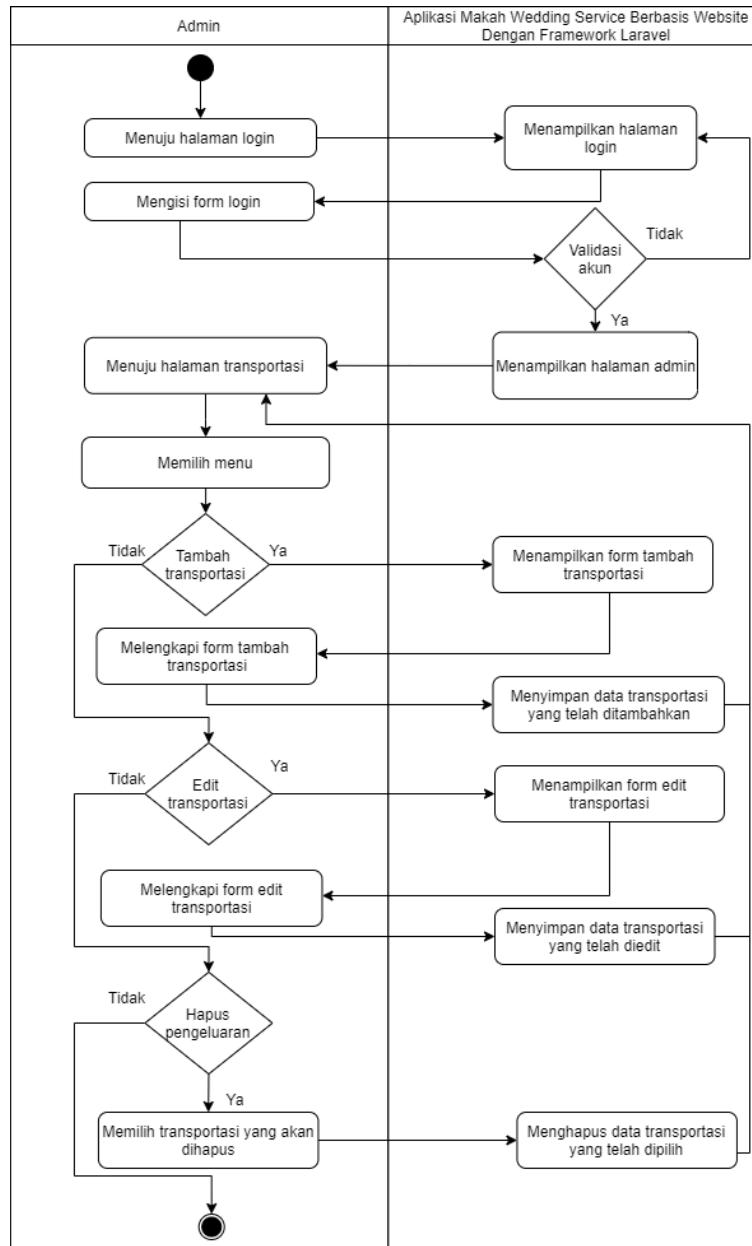


Gambar 3.26 *Activity Diagram* Mengelola Data Pengeluaran Admin

17. *Activity Diagram* Mengelola Data Transportasi Oleh Admin

Proses mengelola data transaksi dapat dikelola bila admin sudah berhasil melakukan *login*. Selanjutnya admin akan menuju halaman transaksi dan admin akan memilih menu tambah transaksi, edit transaksi, dan hapus transaksi. Bila admn

memilih menu tambah transaksi maka admin akan melengkapi *form* tambah pengeluaran, bila admin memilih menu edit transaksi maka admin akan melengkapi *form* edit data pengeluaran, dan jika admin memilih hapus transaksi maka data transportasi akan terhapus. *Activity diagram* mengelola data transportasi oleh admin dapat dilihat pada Gambar 3.27.

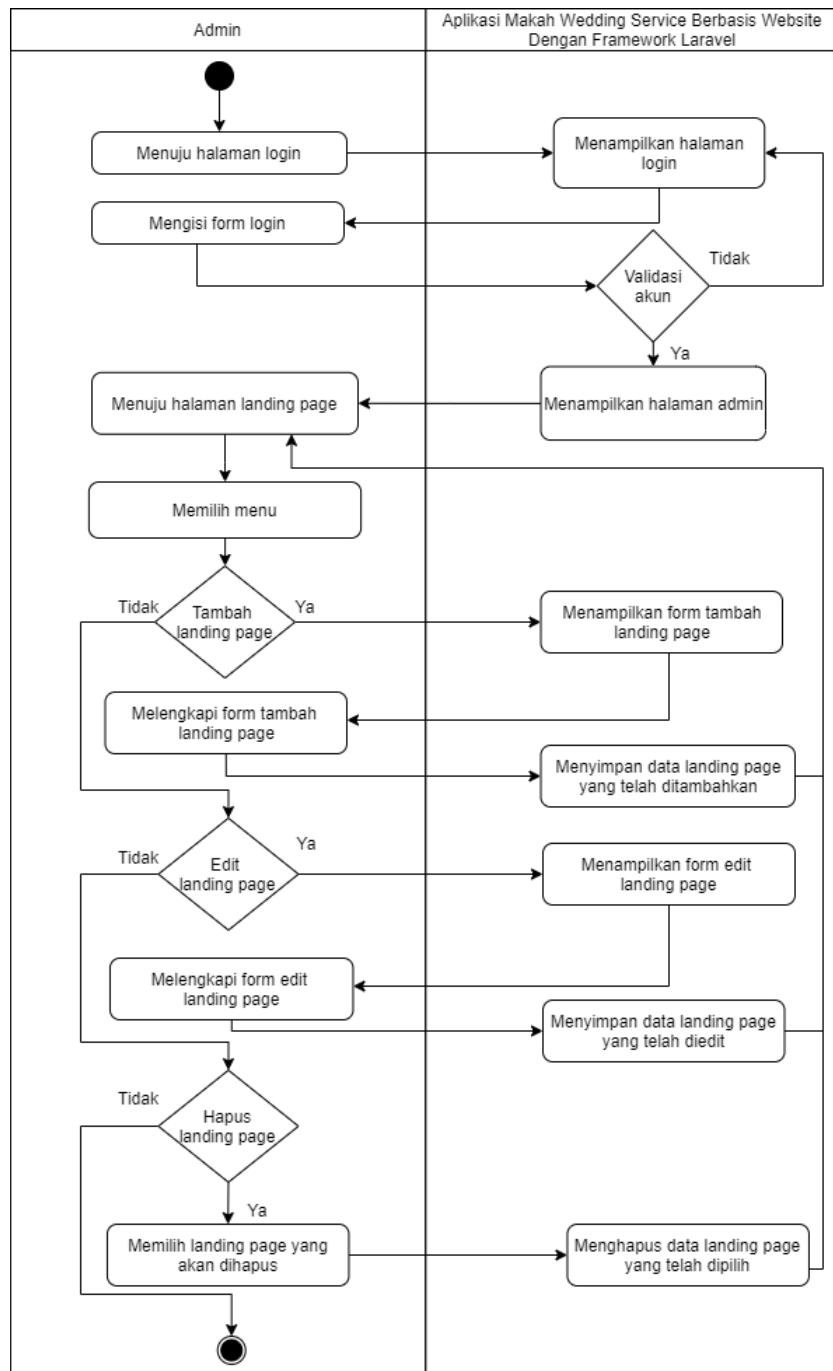


Gambar 3.27 *Activity Diagram* Mengelola Data Transportasi Admin

18. *Activity Diagram* Mengelola Data *Landing Page* Oleh Admin

Admin dapat mengelola data *landing page* jika sudah berhasil melakukan *login*. Lalu menuju pada halaman *landing page* dan memilih menu yang terdiri dari

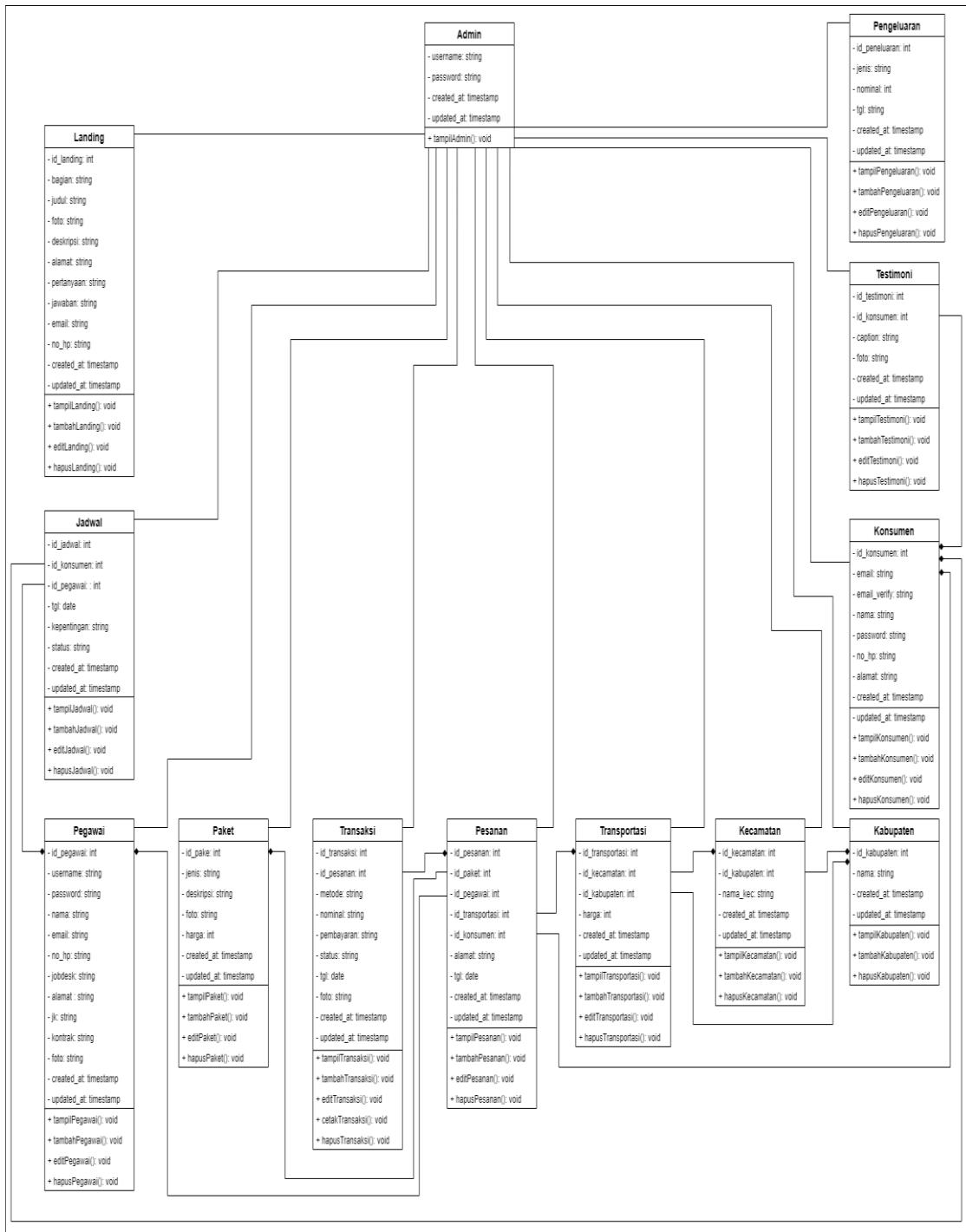
tambah landing page, edit landing page, dan hapus landing page. Ketika admin memilih menu tambah landing page maka admin akan melengkapi form tambah landing page, ketika admin memilih menu edit landing page maka admin akan melengkapi form edit landing page, dan ketika admin memilih hapus landing page maka data akan terhapus. *Activity diagram* mengelola data landing page oleh admin dapat dilihat pada Gambar 3.28.



Gambar 3.28 *Activity Diagram* Mengelola Data Landing Page Admin

c. Class Diagram

Class diagram yang ada pada Aplikasi Makah Wedding Service Berbasis Website Dengan Framework Laravel dapat dilihat pada Gambar 3.29.



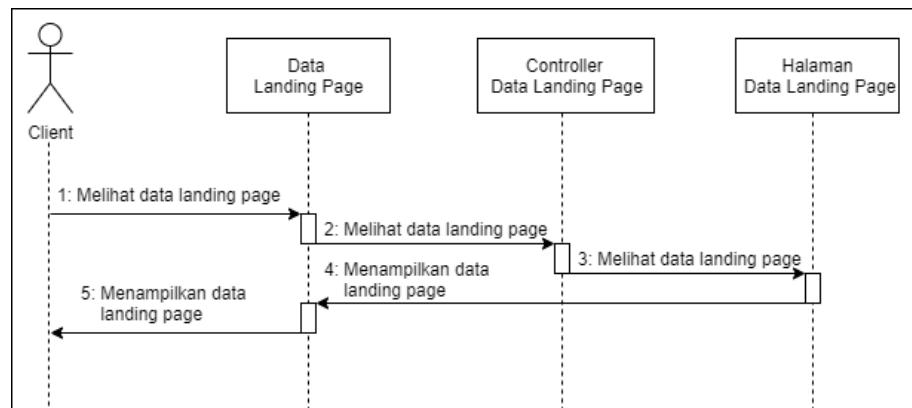
Gambar 3.29 Class Diagram

d. *Sequence Diagram*

Sequence diagram pada Aplikasi Makah Wedding Service Berbasis Website Dengan Framework Laravel yaitu sebagai berikut:

1. *Sequence Diagram Melihat Landing Page*

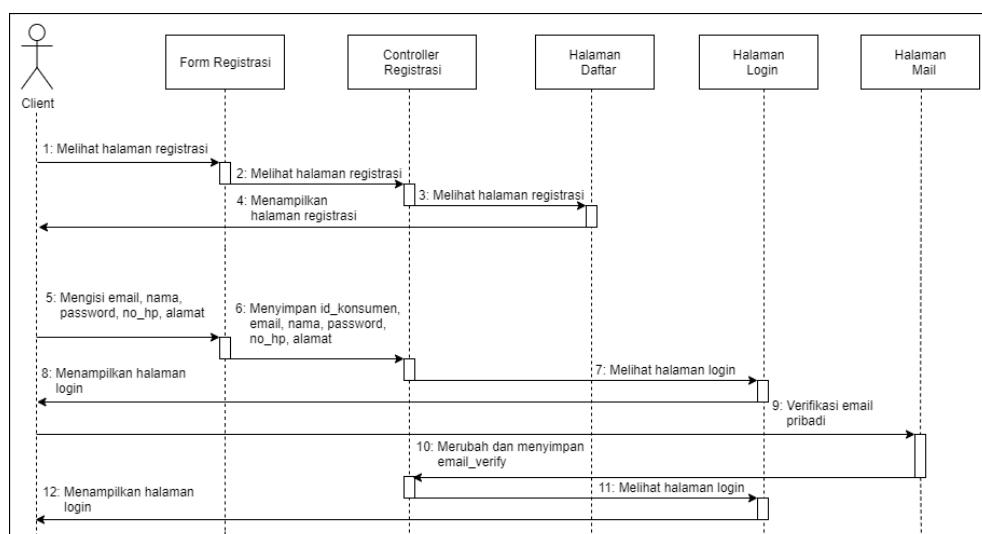
Menjelaskan mengenai urutan proses dalam website ketika *client* akan melihat *landing page*. Dimana *client* harus membuka halaman *landing page* agar bisa melihat halaman *landing page*. *Sequence diagram* melihat *landing page* dapat dilihat pada Gambar 3.30.



Gambar 3.30 *Sequence Diagram Melihat Landing Page*

2. *Sequence Diagram Registrasi*

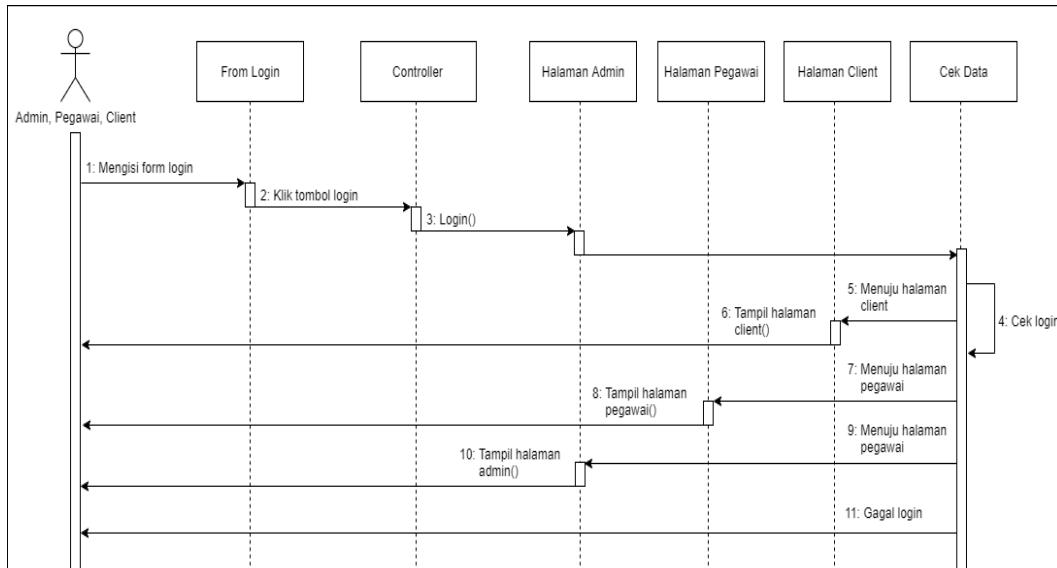
Sequence diagram registrasi menejlasakan tentang urutan proses dalam website ketika *client* akan melakukan registrasi. *Sequence diagram* melihat *registrasi* dapat dilihat pada Gambar 3.31.



Gambar 3.31 *Sequence Diagram Registrasi*

3. Sequence Diagram Login

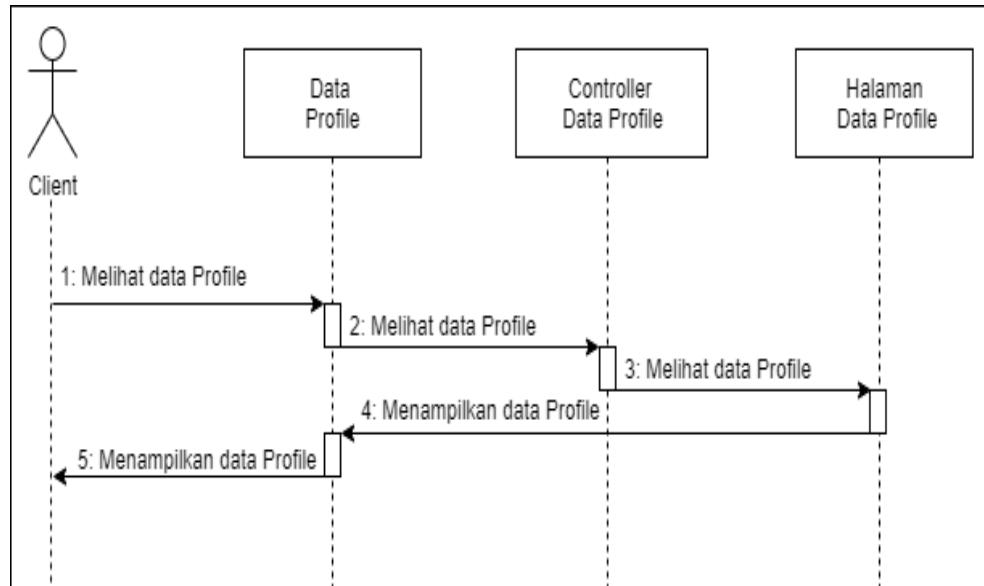
Sequence Diagram login menjelaskan mengenai urutan proses dalam website ketika pengguna melakukan *login*. *Sequence diagram login* dapat dilihat pada Gambar 3.32.



Gambar 3.32 Sequence Diagram Login

4. Sequence Diagram Melihat Profile

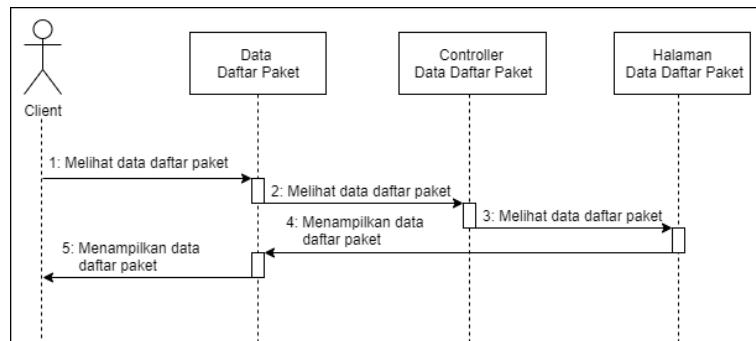
Sequence diagram melihat profile Menjelaskan mengenai urutan proses dalam website ketika *client* akan melihat *profile*. *Sequence diagram melihat profile* dapat dilihat pada Gambar 3.33.



Gambar 3.33 Sequence Diagram Melihat Profile

5. Sequence Diagram Melihat Daftar Paket

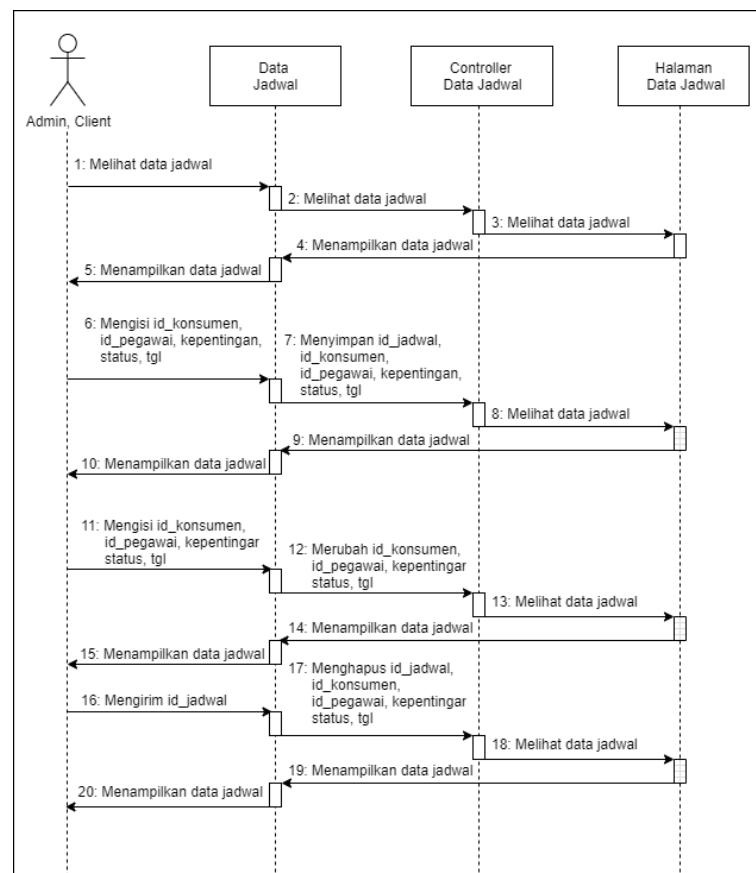
Sequence diagram melihat daftar paket menjelaskan tentang urutan proses dalam website ketika *client* akan melihat daftar paket. *Sequence diagram* melihat daftar paket dapat dilihat pada Gambar 3.34.



Gambar 3.34 *Sequence Diagram* Melihat Daftar Paket

6. Sequence Diagram Mengelola Data Jadwal

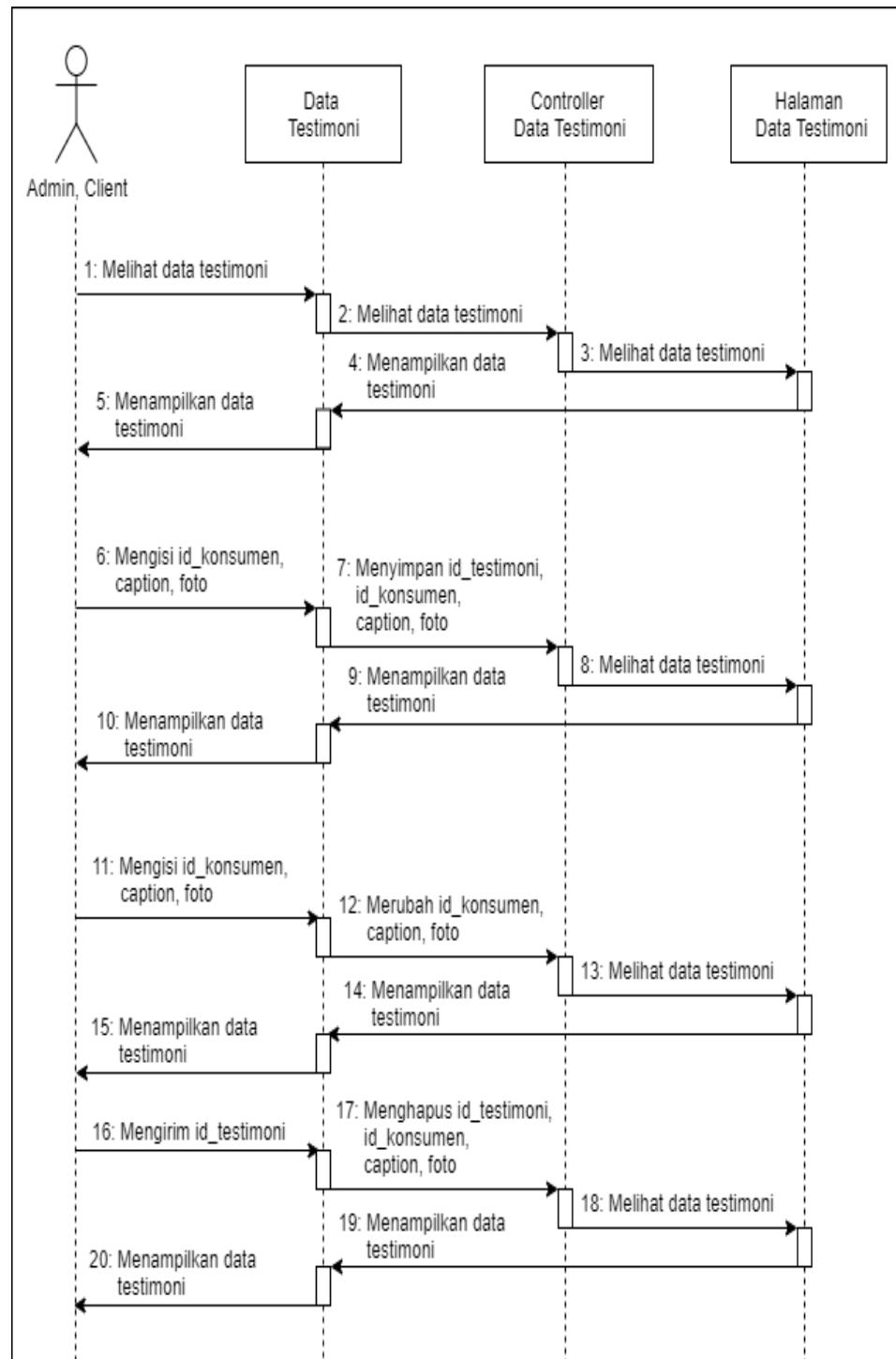
Menjelaskan mengenai bagaimana urutan proses dalam website ketika pengguna mengelola data jadwal. *Sequence diagram* mengelola data jadwal dapat dilihat pada Gambar 3.35.



Gambar 3.35 *Sequence Diagram* Mengelola Data Jadwal

7. Sequence Diagram Mengelola Data Testimoni

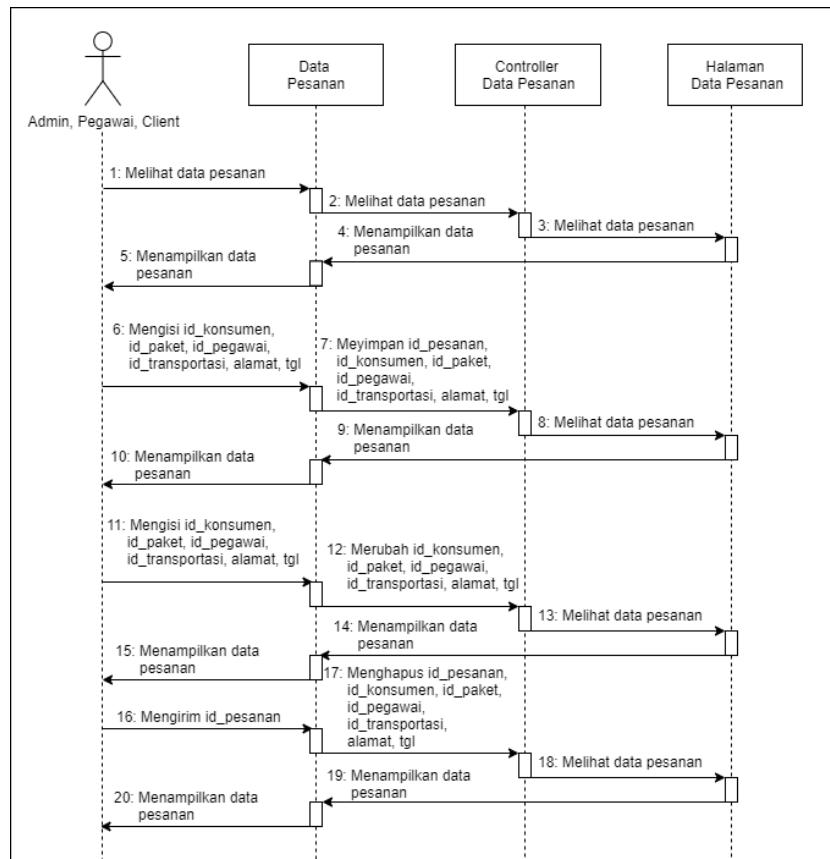
Sequence diagram mengelola data testimoni menjalaskan mengenai urutan proses dalam website ketika pengguna akan mengelola data testimoni. *Sequence diagram* mengelola data testimoni dapat dilihat pada Gambar 3.37.



Gambar 3.36 Sequence Diagram Mengelola Data Testimoni

8. Sequence Diagram Mengelola Data Pesanan

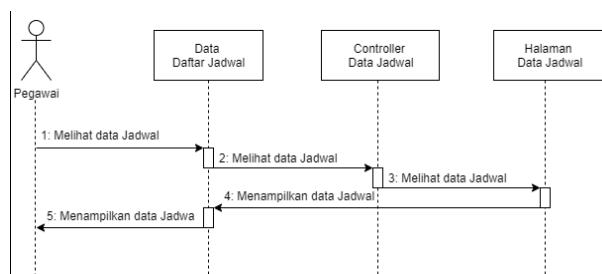
Menjelaskan tentang urutan proses dalam website ketika pengguna akan mengelola data pesanan. *Sequence diagram* mengelola data pesanan dapat dilihat pada Gambar 3.37.



Gambar 3.37 *Sequence Diagram* Mengelola Data Pesanan

9. Sequence Diagram Melihat Data Jadwal

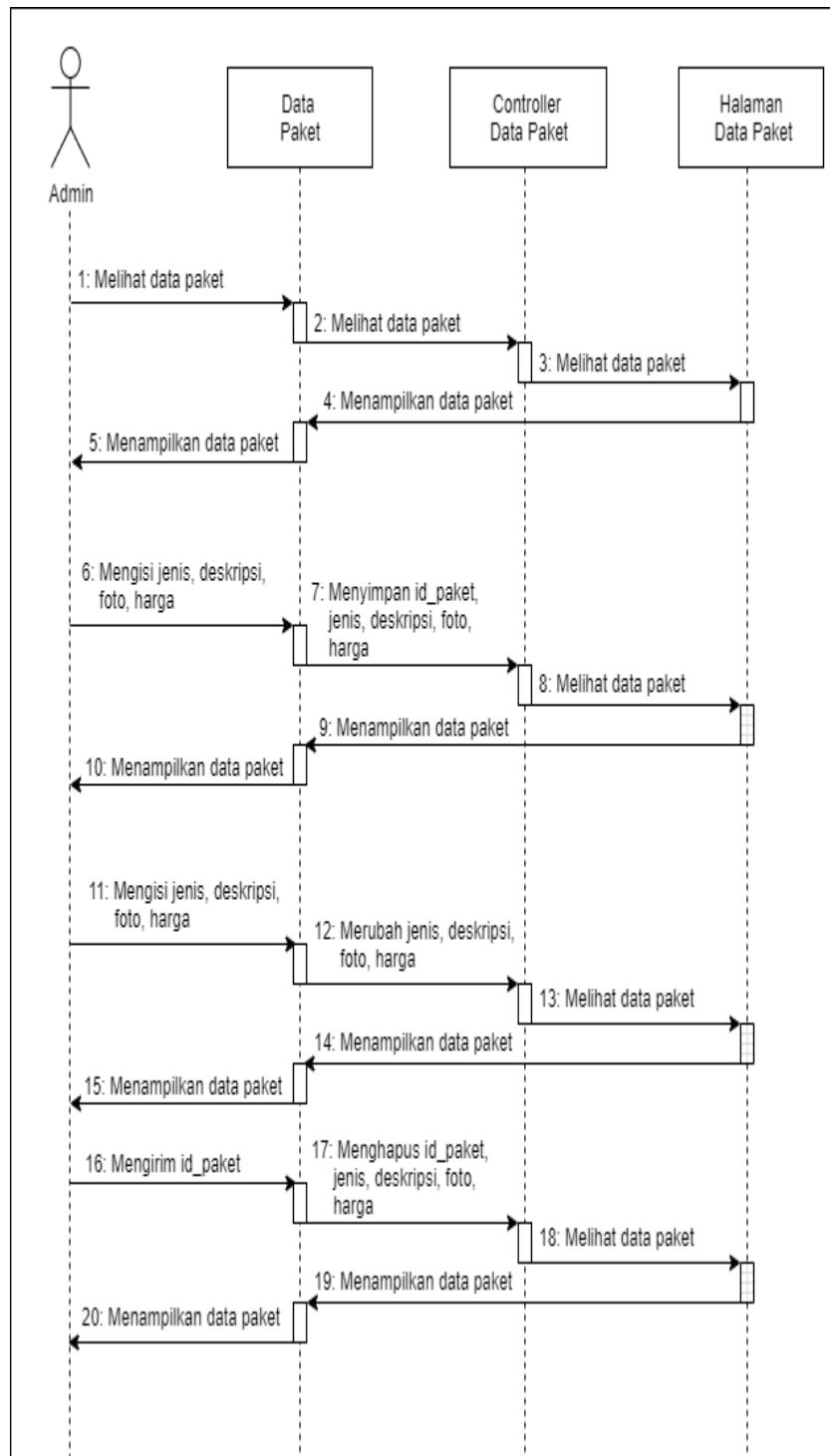
Sequence diagaram melihat data jadwal menguraikan mengenai proses dalam website ketika pegawai akan melihat data jadwal. *Sequence diagram* melihat data jadwal dapat dilihat pada Gambar 3.38.



Gambar 3.38 *Sequence Diagram* Melihat Data Jadwal

10. Sequence Diagram Mengelola Data Paket

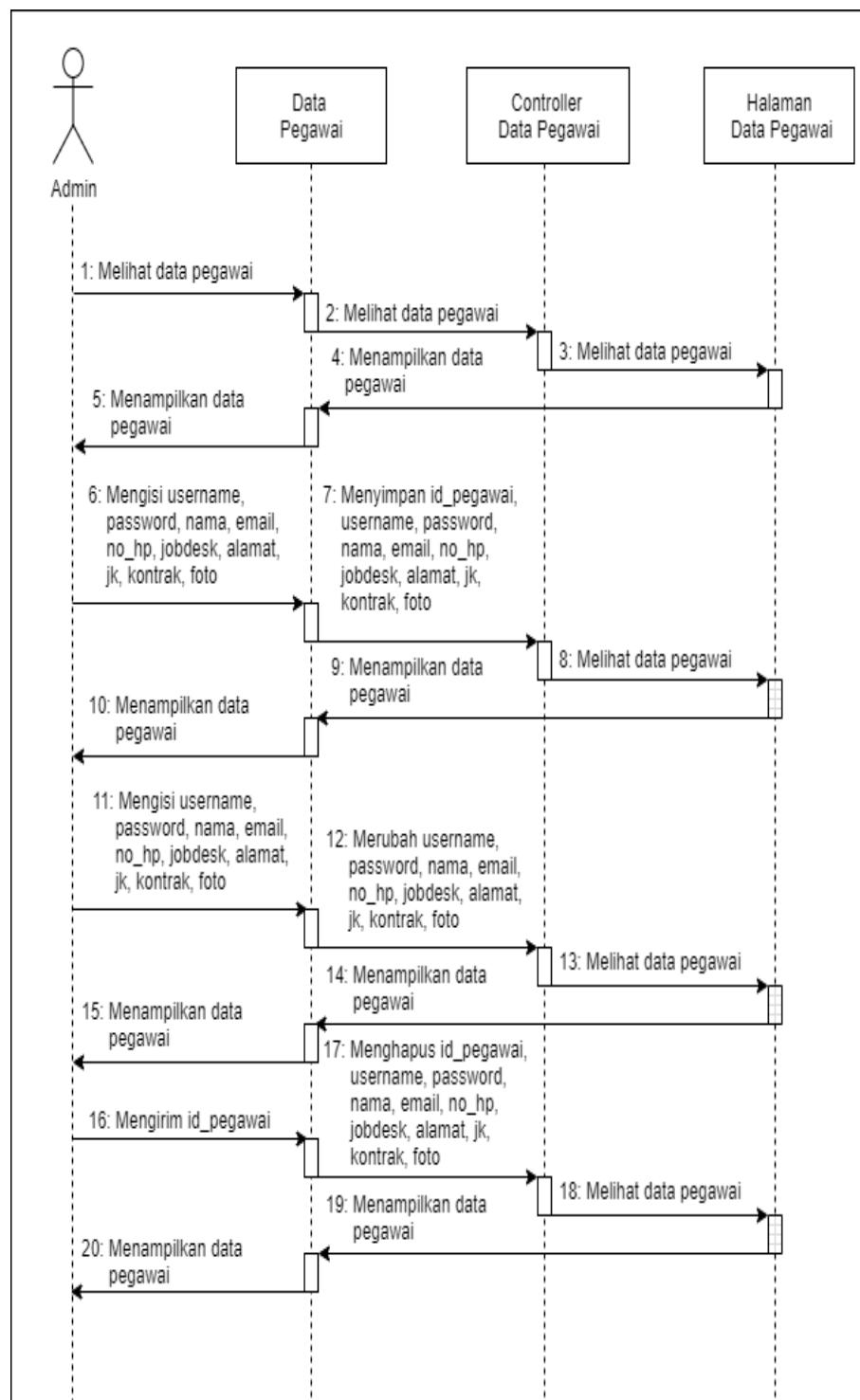
memberikan deskripsi mengenai proses dalam *website* ketika admin akan mengelola data paket. *Sequence diagram* mengelola data paket dapat dilihat pada Gambar 3.39.



Gambar 3. 39 *Sequence Diagram* Mengelola Data Paket

11. Sequence Diagram Mengelola Data Pegawai

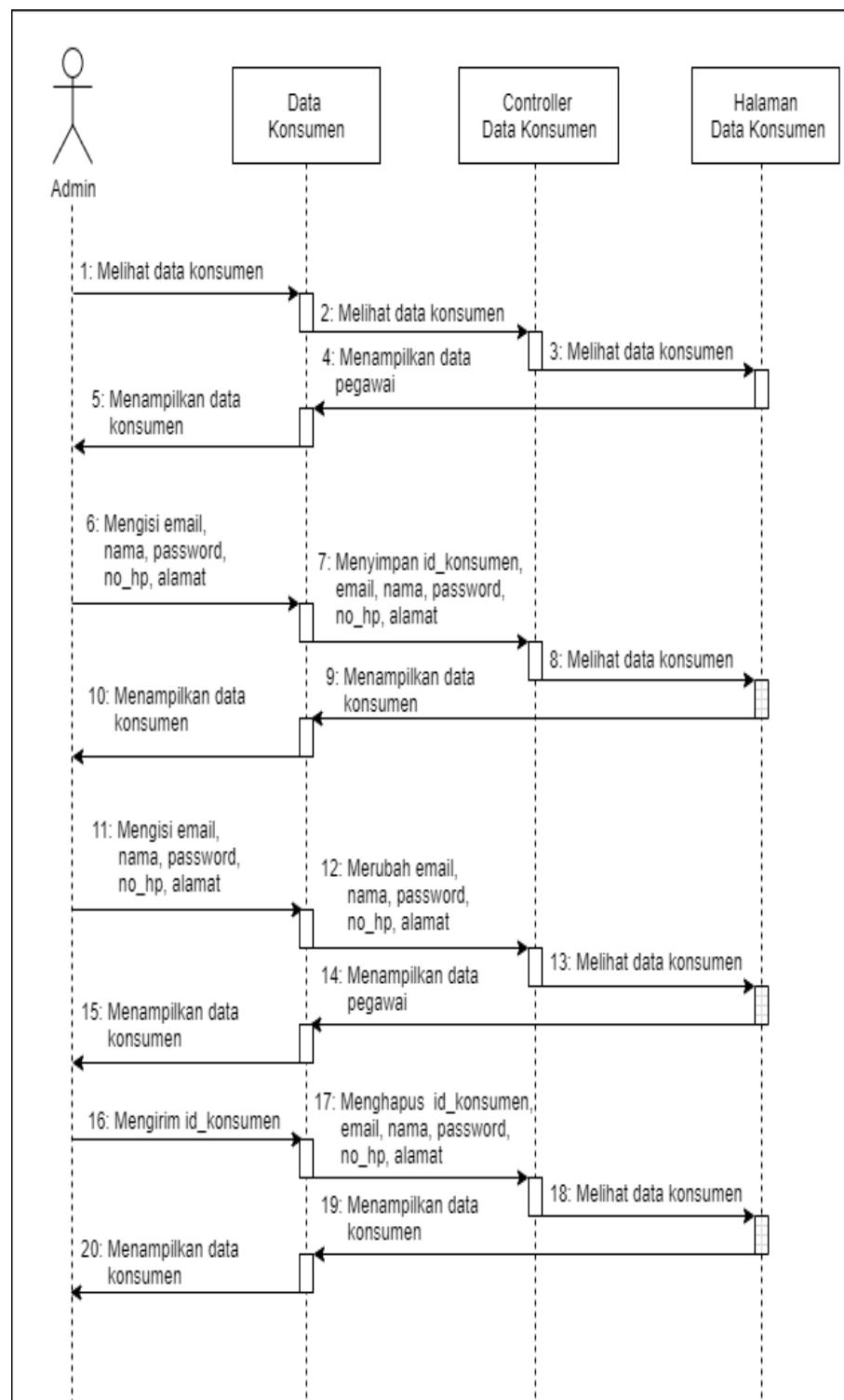
Sequence diagram mengelola data pegawai menjelaskan mengenai proses dalam *website* ketika admin akan mengelola data pegawai. *Sequence diagram* mengelola data pegawai dapat dilihat pada Gambar 3.40.



Gambar 3.40 Sequence Diagram Mengelola Data Pegawai

12. Sequence Diagram Mengelola Data Client

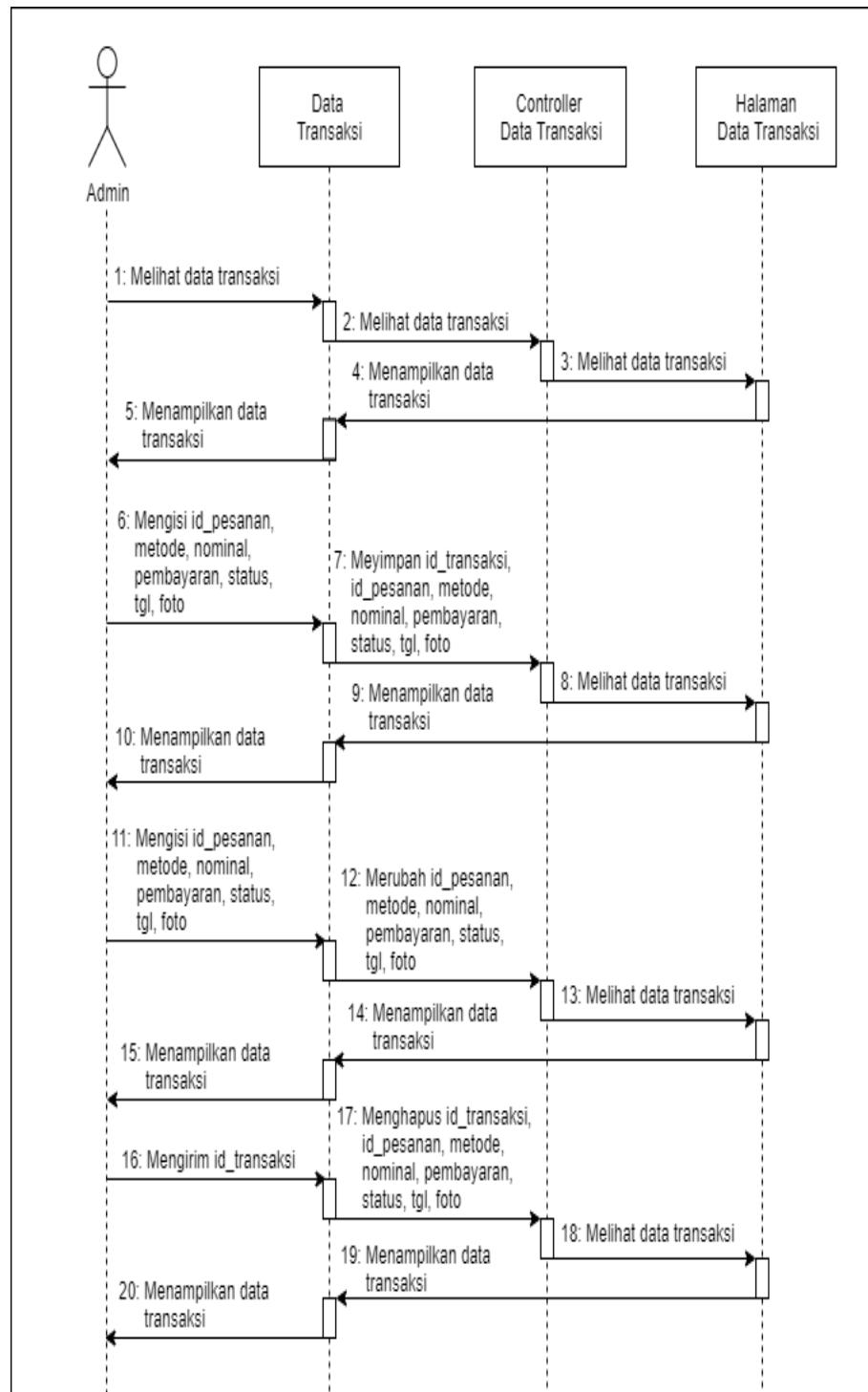
Sequence diagram mengelola data *client* menjelaskan mengenai proses dalam *website* ketika admin akan mengelola data *client*. *Sequence diagram* mengelola data *client* dapat dilihat pada Gambar 3.41.



Gambar 3.41 Sequence Diagram Mengelola Data Client

13. Sequence Diagram Mengelola Data Pembayaran

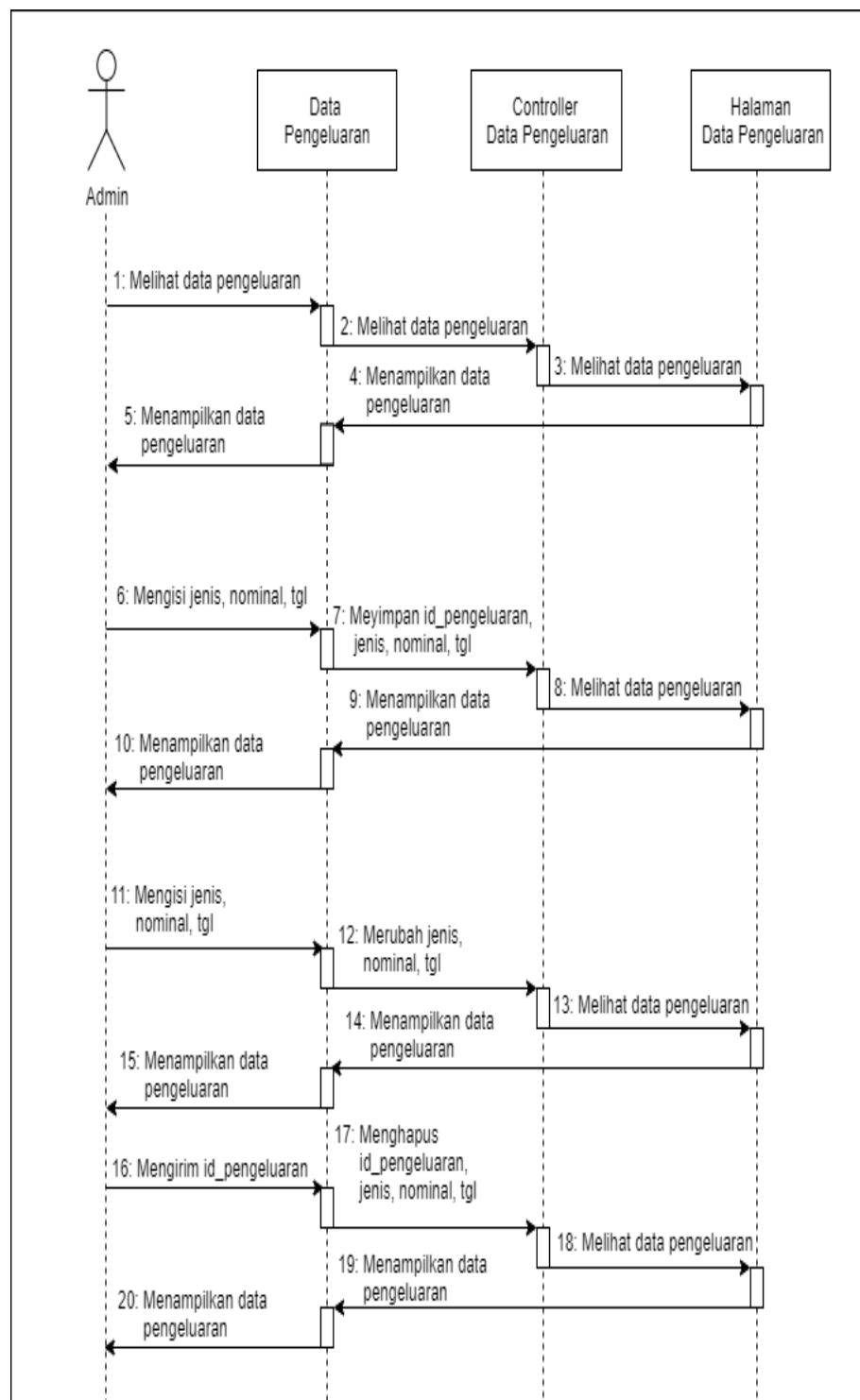
Sequence diagram mengelola data pembayaran menjelaskan mengenai proses dalam *website* ketika admin akan mengelola data pembayaran. *Sequence diagram* mengelola data pembayaran dapat dilihat pada Gambar 3.42.



Gambar 3.42 Sequence Diagram Mengelola Data Pembayaran

14. Sequence Diagram Mengelola Data Pengeluaran

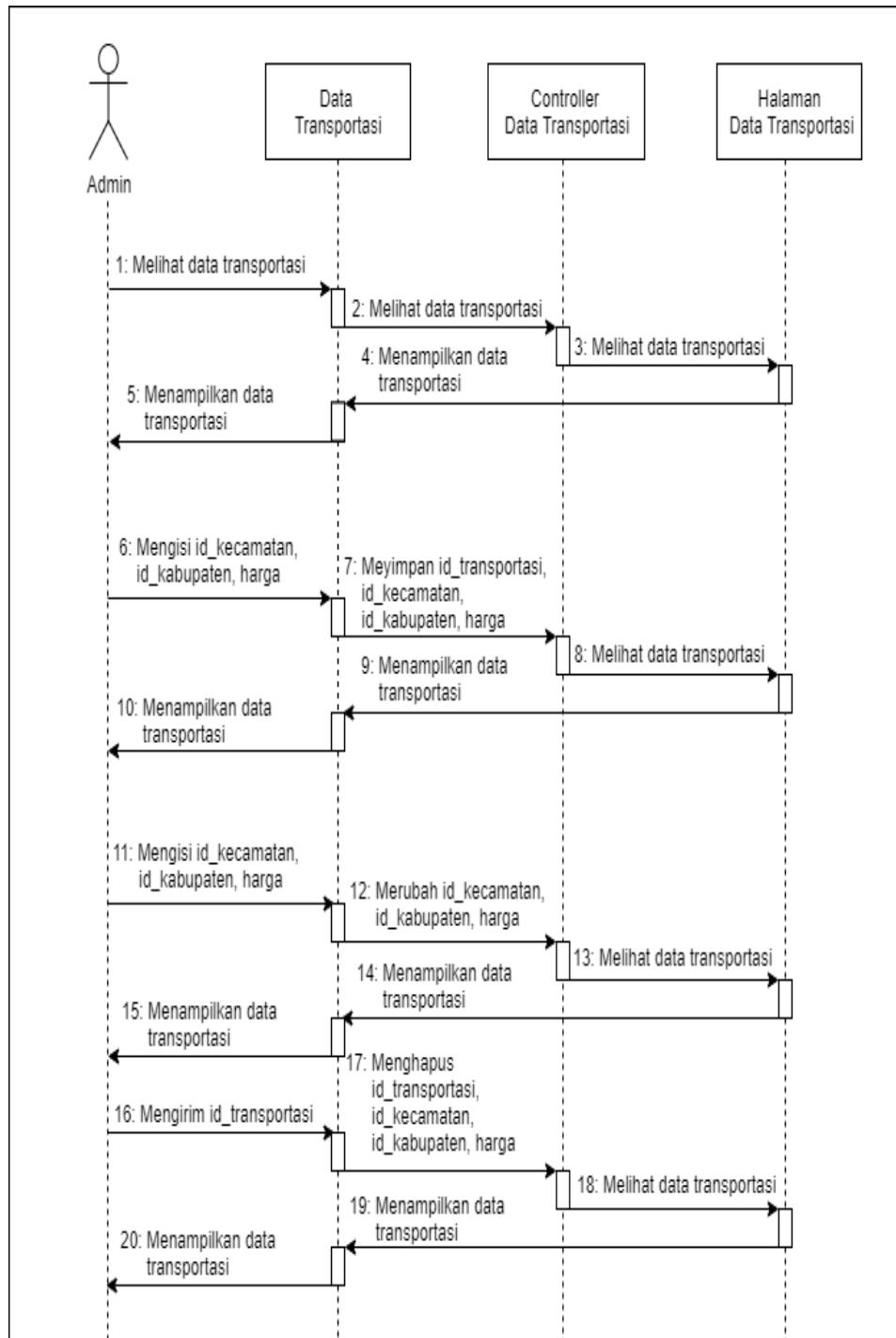
Sequence diagram mengelola data pengeluaran menjelaskan mengenai proses dalam *website* ketika admin akan mengelola data pembayaran. *Sequence diagram* mengelola data pengeluaran dapat dilihat pada Gambar 3.43.



Gambar 3.43 *Sequence Diagram* Mengelola Data Pengeluaran

15. Sequence Diagram Mengelola Data Transportasi

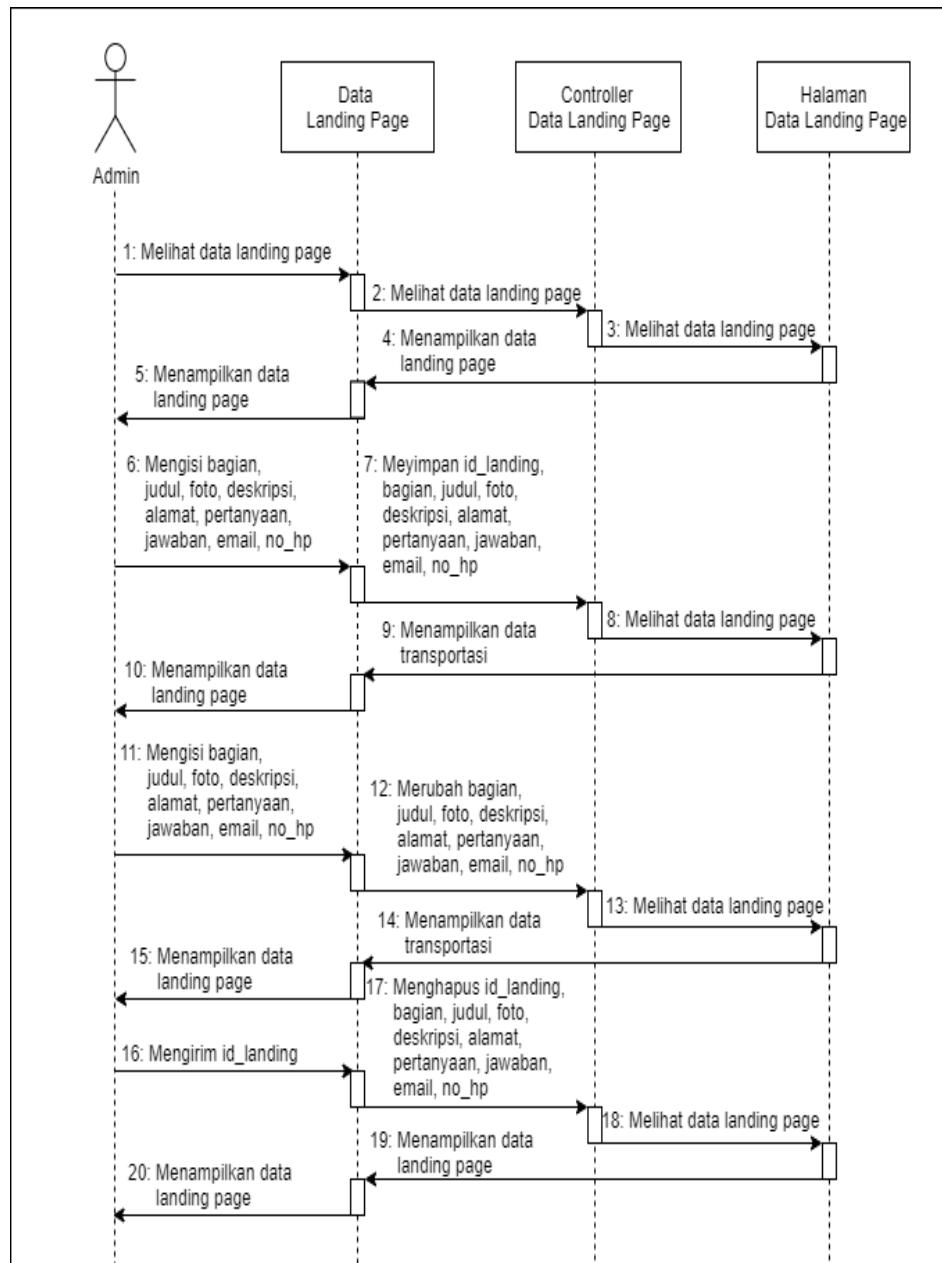
Sequence diagram mengelola data transportasi menjelaskan mengenai proses dalam *website* ketika admin akan mengelola data transportasi. *Sequence diagram* mengelola data transportasi dapat dilihat pada Gambar 3.44.



Gambar 3.44 *Sequence Diagram* Mengelola Data Transportasi

16. Sequence Diagram Mengelola Data Landing Page

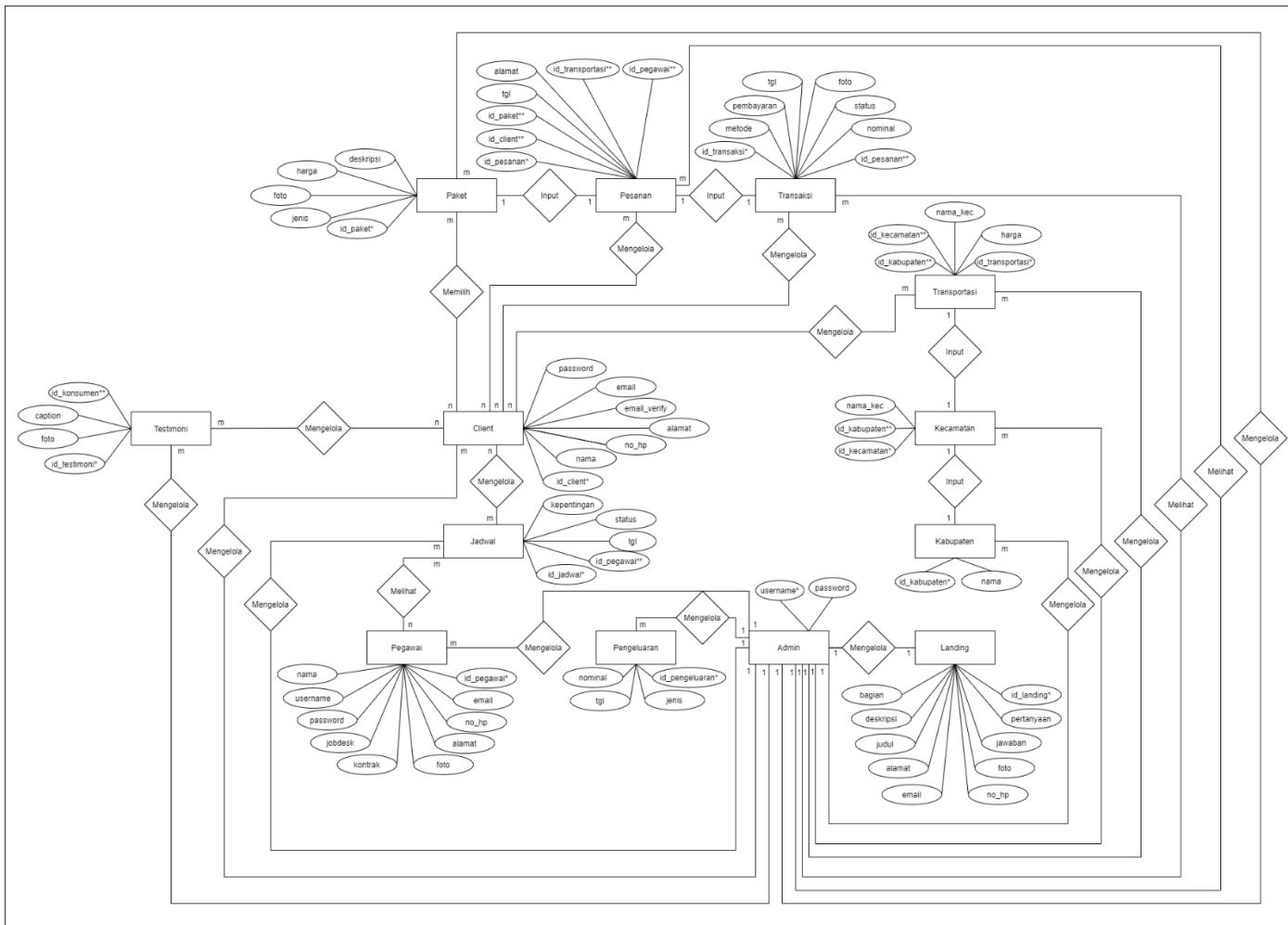
Sequence diagram mengelola data transportasi menjelaskan mengenai proses dalam *website* ketika admin akan mengelola data transportasi. *Sequence diagram* mengelola data transportasi dapat dilihat pada Gambar 3.45.



Gambar 3.45 Sequence Diagram Mengelola Data Landing Page

3.6. Rancangan ERD (*Entity Relationship Diagram*)

Rancangan ERD (*Entity Relationship Diagram*) yang dipakai dalam Aplikasi Makah Wedding Service Berbasis Website Dengan *Framework* Laravel dapat dilihat pada Gambar 3.46.



Gambar 3.46 Rancangan ERD (*Entity Relationship Diagram*)

3.7. Rancangan Database

Dalam membangun Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dibutuhkan sebuah rancangan *database*, dimana dalam *database* ini memerlukan 13 tabel yaitu tabel admin, pegawai, konsumen, pesanan, paket, kabupaten, kecamatan, transportasi, transaksi, jadwal, testimoni, landing, dan pemasukan.

a. Rancangan Tabel Admin

Tabel admin digunakan untuk menyimpan data admin. Pada tabel ini terdapat 4 field, yaitu *username*, *password*, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel admin dapat dilihat pada Tabel 3.3.

Tabel 3.3 Rancangan Tabel Admin

No	Nama Field	Tipe Data	Keterangan
1	<i>username</i>	varchar(255)	Untuk menyimpan <i>username</i> admin
2	<i>password</i>	varchar(255)	Untuk menyimpan <i>password</i> admin
3	<i>created_at</i>	timestamp	Tanggal dibuatnya data admin
4	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data admin

b. Rancangan Tabel Pegawai

Tabel pegawai digunakan untuk menyimpan data pegawai. Pada tabel ini terdapat 13 field, yaitu *id_pegawai*, *username*, *password*, *nama*, *email*, *no_hp*, *jobdesk*, *alamat*, *jk*, *kontrak*, *foto*, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel pegawai dapat dilihat pada Tabel 3.4.

Tabel 3.4 Rancangan Tabel Pegawai

No	Nama Field	Tipe Data	Keterangan
1	<i>id_pegawai</i>	int(10)	Untuk menyimpan id pegawai
2	<i>username</i>	varchar(255)	Untuk menyimpan <i>username</i> pegawai
3	<i>password</i>	varchar(255)	Untuk menyimpan <i>password</i> pegawai
4	<i>nama</i>	varchar(255)	Untuk menyimpan nama pegawai
5	<i>email</i>	varchar(255)	Untuk menyimpan email pegawai
6	<i>no_hp</i>	varchar(255)	Untuk menyimpan nomor hp pegawai
7	<i>jobdesk</i>	varchar(255)	Untuk menyimpan <i>job deskripsi</i> pegawai

8	alamat	varchar(255)	Untuk menyimpan alamat pegawai
9	jk	varchar(255)	Untuk menyimpan jenis kelamin pegawai
10	kontrak	varchar(255)	Untuk menyimpan kontrak kerja pegawai
11	foto	varchar(255)	Untuk menyimpan foto pegawai
12	<i>created_at</i>	timestamp	Tanggal dibuatnya data pegawai
13	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data pegawai

c. Rancangan Tabel Konsumen

Tabel konsumen digunakan untuk menyimpan data *client*. Pada tabel ini terdapat 8 field, yaitu *id_konsumen*, *email*, *nama*, *password*, *no_hp*, *alamat*, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel konsumen dapat dilihat pada Tabel 3.5.

Tabel 3.5 Rancangan Tabel Konsumen

No	Nama Field	Tipe Data	Keterangan
1	<i>id_konsumen</i>	int(10)	Untuk menyimpan id <i>client</i>
2	<i>email</i>	varchar(255)	Untuk menyimpan email <i>client</i>
3	<i>email_verify</i>	varchar(255)	Untuk validasi akun <i>client</i>
4	<i>nama</i>	varchar(255)	Untuk menyimpan nama <i>client</i>
5	<i>password</i>	varchar(255)	Untuk menyimpan <i>password client</i>
6	<i>no_hp</i>	varchar(255)	Untuk menyimpan nomor hp <i>client</i>
7	<i>alamat</i>	varchar(255)	Untuk menyimpan alamat <i>client</i>
8	<i>created_at</i>	timestamp	Tanggal dibuatnya data <i>client</i>
9	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data <i>client</i>

d. Rancangan Tabel Pesanan

Tabel pesanan digunakan untuk menyimpan data pesanan. Pada tabel ini terdapat 9 field, yaitu *id_pesanan*, *id_konsumen*, *id_paket*, *id_pegawai*,

`id_transportasi`, alamat, tgl, `created_at`, dan `updated_at`. Penjelasan mengenai field-field pada tabel pesanan dapat dilihat pada Tabel 3.6.

Tabel 3.6 Rancangan Tabel Pesanan

No	Nama Field	Tipe Data	Keterangan
1	<code>id_pesanan</code>	int(10)	Untuk menyimpan id pesanan
2	<code>id_konsumen</code>	int(10)	Untuk menyimpan id_konsumen yang melakukan pesanan
3	<code>id_paket</code>	int(10)	Untuk menyimpan id_paket yang dipilih untuk pesanan
4	<code>id_pegawai</code>	int(10)	Untuk menyimpan id_pegawai yang dipilih untuk mengelola pesanan
5	<code>id_transportasi</code>	int(10)	Untuk menyimpan id_transportasi yang dipilih untuk pesanan yang dijadikan sebagai penentu harga pesanan yang dipesan
6	alamat	varchar(255)	Untuk menyimpan alamat <i>client</i>
7	tgl	date	Untuk menyimpan tanggal dilakukannya acara pesanan
8	<code>created_at</code>	timestamp	Tanggal dibuatnya data pesanan
9	<code>updated_at</code>	timestamp	Tanggal dilakukannya <i>update</i> data pesanan

e. Rancangan Tabel Paket

Tabel paket digunakan untuk menyimpan data paket. Pada tabel ini terdapat 7 field, yaitu `id_paket`, jenis, deskripsi, harga, foto, `created_at`, dan `updated_at`. Penjelasan mengenai field-field pada tabel paket dapat dilihat pada Tabel 3.7.

Tabel 3.7 Rancangan Tabel Paket

No	Nama Field	Tipe Data	Keterangan
1	<code>id_paket</code>	int(10)	Untuk menyimpan id paket
2	jenis	varchar(255)	Untuk menyimpan jenis paket
3	deskripsi	varchar(255)	Untuk menyimpan deskripsi paket
4	harga	int(11)	Untuk menyimpan harga paket
5	foto	varchar(255)	Untuk menyimpan foto deskripsi paket

6	<i>created_at</i>	timestamp	Tanggal dibuatnya data paket
7	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data paket

f. Rancangan Tabel Kabupaten

Tabel kabupaten digunakan untuk menyimpan data kabupaten dari cangkupan kerja jasa Makah *Wedding Service*. Pada tabel ini terdapat 4 field, yaitu *id_kabupaten*, *nama*, *created_at*, dan *updated_at*. Penjelasan mengenai filed-field pada tabel kabupaten dapat dilihat pada Tabel 3.8.

Tabel 3.8 Rancangan Tabel Kabupaten

No	Nama Field	Tipe Data	Keterangan
1	<i>id_kabupaten</i>	int(10)	Untuk menyimpan id kabupaten
2	<i>nama</i>	varchar(255)	Untuk menyimpan nama kabupaten
3	<i>created_at</i>	timestamp	Tanggal dibuatnya data kabupaten
4	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data kabupaten

g. Rancangan Tabel Kecamatan

Tabel kabupaten digunakan untuk menyimpan data kecamatan dari cangkupan kerja jasa Makah *Wedding Service*. Pada tabel ini terdapat 5 field, yaitu *id_kecamatan*, *id_kabupaten*, *nama_kec*, *created_at*, dan *updated_at*. Penjelasan mengenai filed-field pada tabel kecamatan dapat dilihat pada Tabel 3.9.

Tabel 3.9 Rancangan Tabel Kecamatan

No	Nama Field	Tipe Data	Keterangan
1	<i>id_kecamatan</i>	int(10)	Untuk menyimpan id kecamatan
2	<i>id_kabupaten</i>	int(10)	Untuk menyimpan id_kabupaten
3	<i>nama_kec</i>	varchar(255)	Untuk menyimpan nama kabupaten
4	<i>created_at</i>	timestamp	Tanggal dibuatnya data kecamatan
5	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data kecamatan

h. Rancangan Tabel Transportasi

Tabel transportasi digunakan untuk menyimpan data transportasi pesanan. Pada tabel ini terdapat 6 field yaitu, id_transportasi, id_kabupaten, nama_kec, harga, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel transportasi dapat dilihat pada Tabel 3.10.

Tabel 3.10 Rancangan Tabel Transportasi

No	Nama Field	Tipe Data	Keterangan
1	id_transportasi	int(10)	Untuk menyimpan id transportasi
2	id_kabupaten	int(10)	Untuk menyimpan id_kabupaten
3	Id_kecamatan	int(10)	Untuk menyimpan id_kecamatan
4	harga	int(11)	Untuk menyimpan harga transportasi
5	<i>created_at</i>	timestamp	Tanggal dibuatnya data transportasi
6	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data transportasi

i. Rancangan Tabel Transaksi

Tabel transaksi digunakan untuk menyimpan data transaksi. Pada tabel ini terdapat 10 field yaitu, id_transportasi, id_pesanan, metode, nominal, pembayaran, status, tgl, foto, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel transaksi dapat dilihat pada Tabel 3.11.

Tabel 3.11 Rancangan Tabel Transaksi

No	Nama Field	Tipe Data	Keterangan
1	id_transaksi	int(10)	Untuk menyimpan id transaksi
2	id_pesanan	int(10)	Untuk menyimpan id_pesanan yang melakukan transaksi
3	metode	varchar(255)	Untuk menyimpan metode yang dilakukan ketika melakukan transaksi
4	nominal	varchar(255)	Untuk menyimpan nominal harga ketika melakukan transaksi
5	pembayaran	varchar(255)	Untuk menyimpan pembayaran yang telah dilakukan ketika melakukan transaksi

6	status	varchar(255)	Untuk menyimpan status pembayaran yang telah melakukan transaksi
7	tgl	date	Untuk menyimpan tanggal pada dilakukannya transaksi
8	foto	varchar(255)	Untuk menyimpan bukti foto disaat dilakukannya transaksi
9	<i>created_at</i>	timestamp	Tanggal dibuatnya data transaksi
10	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data transaksi

j. Rancangan Tabel Jadwal

Tabel jadwal digunakan untuk menyimpan data jadwal. Pada tabel ini terdapat 8 field, yaitu id_jadwal, id_konsumen, id_pgawai, tgl, kepentingan, status, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel jadwal dapat dilihat pada Tabel 3.12.

Tabel 3.12 Rancangan Tabel Jadwal

No	Nama Field	Tipe Data	Keterangan
1	id_jadwal	int(10)	Untuk menyimpan id jadwal
2	id_konsumen	int(10)	Untuk menyimpan id_konsumen yang mengajukan jadwal
3	id_pgawai	int(10)	Untuk menyimpan id pegawai
4	tgl	varchar(255)	Untuk menyimpan tanggal pada mengajukan jadwal
5	kepentingan	varchar(255)	Untuk menyimpan kepentingan yang diajukan untuk mengajukan jadwal
6	status	varchar(255)	Untuk menyimpan status jadwal
7	<i>created_at</i>	timestamp	Tanggal dibuatnya data jadwal
8	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data jadwal

k. Rancangan Tabel Testimoni

Tabel testimoni digunakan untuk menyimpan data tesimoni. Pada tabel ini terdapat 6 field, yaitu id_testimoni, id_konsumen, caption, foto, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel testimoni dapat dilihat pada Tabel 3.13.

Tabel 3.13 Rancangan Tabel Testimoni

No	Nama Field	Tipe Data	Keterangan
1	id_testimoni	int(10)	Untuk menyimpan id testimoni
2	id_konsumen	int(10)	Untuk menyimpan id_konsumen yang memberikan testimoni
3	caption	varchar(255)	Untuk menyimpan kata-kata yang diberikan oleh <i>client</i>
4	foto	varchar(255)	Untuk menyimpan foto testimoni
5	<i>created_at</i>	timestamp	Tanggal dibuatnya data testimoni
6	<i>updated_at</i>	timestamp	Tanggal dilakukannya <i>update</i> data testimoni

1. Rancangan Tabel *Landing*

Tabel landing digunakan untuk menyimpan data *landing page* yang akan ditampilkan. Pada tabel ini terdapat 12 field yaitu, id_landing, bagian, judul, foto, deskripsi, alamat, pertanyaan, jawaban, email, no_hp, *created_at*, dan *updated_at*. Adapun penjelasan mengenai tabel landing dapat dilihat pada Tabel 3.14.

Tabel 3.14 Rancangan Tabel *Landing*

No	Nama Field	Tipe Data	Keterangan
1	id_landing	int(10)	Untuk menyimpan id <i>landing page</i>
2	bagian	varchar(255)	Untuk menyimpan bagian yang akan ditampilkan pada <i>landing page</i>
3	judul	varchar(255)	Untuk menyimpan judul dari <i>landing page</i>
4	foto	varchar(255)	Untuk menyimpan foto <i>landing page</i>
5	deskripsi	longtext	Untuk menyimpan deskripsi <i>landing page</i>
6	alamat	varchar(255)	Untuk menyimpan alamat dari Makah <i>Wedding Service</i> yang akan ditampilkan pada <i>landing page</i>
7	pertanyaan	varchar(255)	Untuk menyimpan pertanyaan yang akan ditampilkan pada <i>landing page</i>
8	jawaban	varchar(255)	Untuk menyimpan jawaban yang akan ditampilkan pada <i>landing page</i>

9	email	varchar(255)	Untuk menyimpan email dari Makah Wedding Service yang akan ditampilkan pada landing page
10	no_hp	varchar(255)	Untuk menyimpan nomor hp dari Makah Wedding Service yang akan ditampilkan pada landing page
11	<i>created_at</i>	timestamp	Tanggal dibuatnya data landing page
12	<i>updated_at</i>	timestamp	Tanggal dilakukannya update data landing page

m. Rancangan Tabel Pengeluaran

Tabel pengeluaran digunakan untuk menyimpan data pengeluaran keuangan pada Makah Wedding Service. Pada tabel ini terdapat 6 field, yaitu id_pengeluaran, jenis, nominal, tgl, *created_at*, dan *updated_at*. Penjelasan mengenai field-field pada tabel pengeluaran dapat dilihat pada Tabel 3.15.

Tabel 3.15 Rancangan Tabel Pengeluaran

No	Nama Field	Tipe Data	Keterangan
1	id_pengeluaran	int(10)	Untuk menyimpan id pengeluaran
2	jenis	varchar(255)	Untuk menyimpan jenis pengeluaran
3	nominal	int(10)	Untuk menyimpan nominal harga pengeluaran
4	tgl	date	Untuk menyimpan tanggal saat mendapatkan pengeluaran
5	<i>created_at</i>	timestamp	Tanggal dibuatnya data pengeluaran
6	<i>updated_at</i>	timestamp	Tanggal dilakukannya update data pengeluaran

3.8. Rancangan Interface

Rancangan *interface* pada Aplikasi Makah Wedding Service Berbasis Website Dengan *Framework* Laravel dibuat dengan menggunakan aplikasi Figma. Rancangan ini akan memberikan gambaran antarmuka sistem dengan pengguna, sehingga dapat disesuaikan oleh pengguna sebelum melakukan implementasi pada program. Beberapa rancangan *interface* yang dirancang dalam proses pembuatan Aplikasi Makah Wedding Service Berbasis Website Dengan *Framework* Laravel ini adalah sebagai berikut:

a. Rancangan Tampilan Halaman *Landing Page*

Saat membuka Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel maka akan menampilkan halaman *landing page* Aplikasi Makah *Wedding Service* yang menampilkan *company profile* dari makah *wedding service*. Kemudian juga ada *registrasi* untuk *client* yang akan membuat akun. Lalu ada *login* yang menghubungkan pada halaman *login* untuk *client*, pegawai, dan admin. Rancangan tampilan halaman *landing page* dapat dilihat pada Gambar 3.47.



Gambar 3.47 Rancangan Tampilan Halaman *Landing Page*

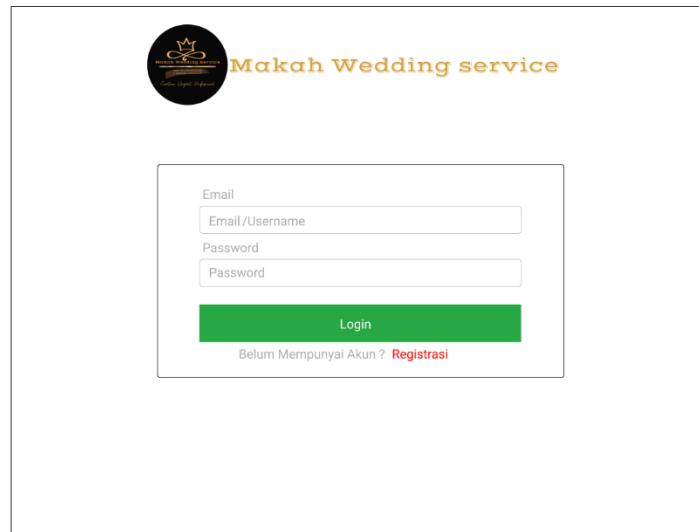
b. Rancangan Tampilan Halaman *Registrasi*

Rancangan tampilan halaman *registrasi* yang menampilkan form registrasi terdiri dari nama, email, *password*, nomor handphone, dan alamat yang harus dilengkapi oleh *client* agar berhasil membuat akun. Rancangan tampilan halaman *registrasi* dapat dilihat pada Gambar 3.48.

Gambar 3.48 Rancangan Tampilan Halaman *Registrasi*

c. Rancangan Tampilan Halaman Login

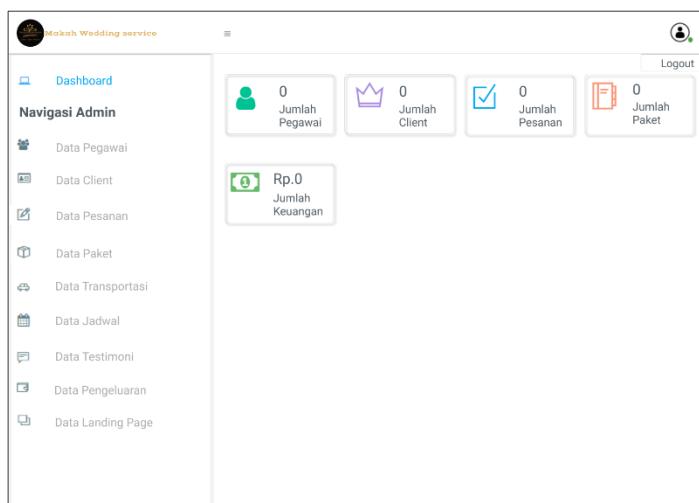
Rancangan tampilan halaman *login* akan menampilkan *form login* yang terdiri dari *username* atau email dan *password* yang harus dilengkapi oleh pengguna agar proses *login* berhasil dilakukan. Rancangan tampilan halaman *login* dapat dilihat pada Gambar 3.49.



Gambar 3.49 Rancangan Tampilan Halaman *Login*

d. Rancangan Tampilan Halaman Admin

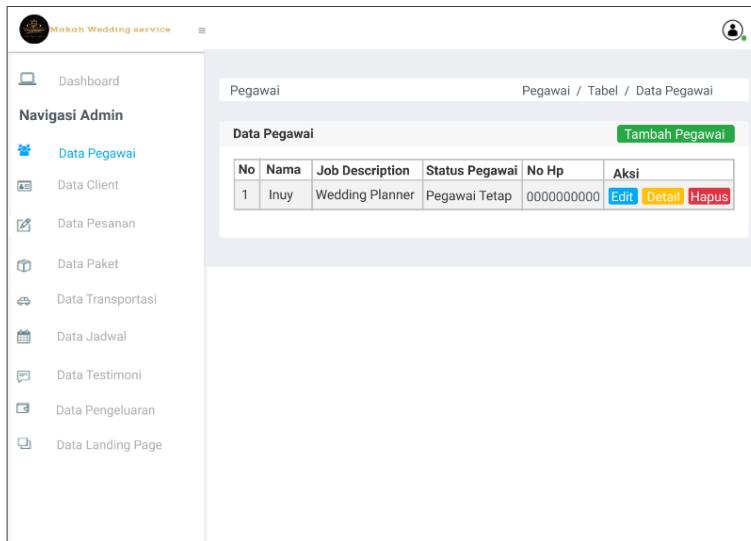
Rancangan tampilan halaman admin menampilkan data Jumlah Pegawai, jumlah *client*, jumlah pesanan, jumlah paket, dan jumlah keuangan. Kemudian juga ada navigasi admin yang terdiri dari data pegawai, data *client*, data pesanan, data paket, data transportasi, data jadwal, data testimoni, data pengeluaran, dan data *landing page*. Rancangan tampilan halaman admin dapat dilihat pada Gambar 3.50.



Gambar 3.50 Rancangan Tampilan Halaman Admin

1. Rancangan tampilan halaman data pegawai

Rancangan tampilan halaman data pegawai admin menampilkan tabel data pegawai yang terdiri dari *button* tambah pegawai, nama , *job deskripsi*, status pegawai, nomor handphone, dan aksi yang berisi *button* edit, *button* detail, *button* hapus. Rancangan tampilan halaman data pegawai dapat dilihat pada Gambar 3.51.

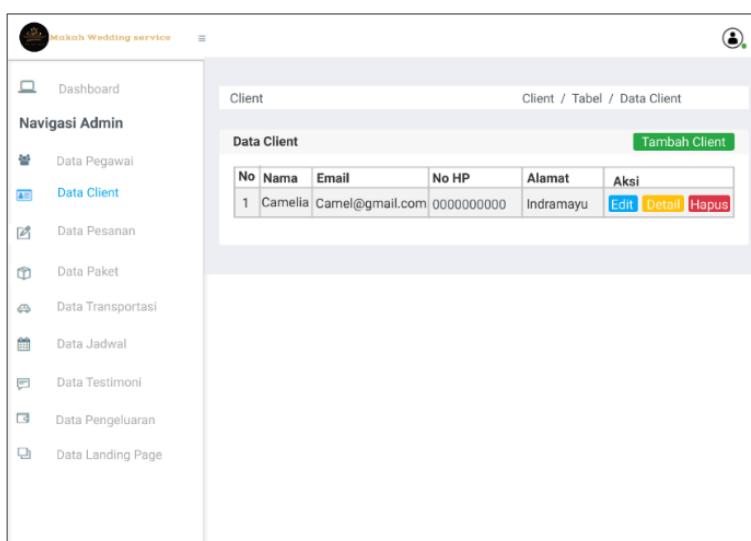


The screenshot shows a web application interface for managing employee data. On the left, there is a sidebar titled 'Navigasi Admin' with various menu items: Dashboard, Data Pegawai (selected), Data Client, Data Pesanan, Data Paket, Data Transportasi, Data Jadwal, Data Testimoni, Data Pengeluaran, and Data Landing Page. The main content area has a title 'Pegawai' and a subtitle 'Pegawai / Tabel / Data Pegawai'. Below this is a table titled 'Data Pegawai' with a green 'Tambah Pegawai' button. The table has columns: No, Nama, Job Description, Status Pegawai, No Hp, and Aksi. There is one row of data: No 1, Nama Inuy, Job Description Wedding Planner, Status Pegawai Pegawai Tetap, No Hp 0000000000, and Aksi with three buttons: Edit (blue), Detail (yellow), and Hapus (red).

Gambar 3.51 Rancangan Tampilan Halaman Data Pegawai Admin

2. Rancangan tampilan halaman data *client*

Rancangan tampilan halaman data *client* menampilkan tabel data *client* yang terdiri dari *button* tambah *client* nomor, nama , email, nomor handphone, alamat, dan aksi yang berisi *button* edit, *button* detail, *button* hapus. Rancangan tampilan halaman data *client* dapat dilihat pada Gambar 3.52.



The screenshot shows a web application interface for managing client data. The sidebar and main structure are identical to the 'Data Pegawai' page in Gambar 3.51. The main content area has a title 'Client' and a subtitle 'Client / Tabel / Data Client'. Below this is a table titled 'Data Client' with a green 'Tambah Client' button. The table has columns: No, Nama, Email, No HP, Alamat, and Aksi. There is one row of data: No 1, Nama Camelia, Email Camel@gmail.com, No HP 0000000000, Alamat Indramayu, and Aksi with three buttons: Edit (blue), Detail (yellow), and Hapus (red).

Gambar 3.52 Rancangan Tampilan Halaman Data *Client* Admin

3. Rancangan tampilan halaman data pesanan

Rancangan tampilan halaman data pesanan admin menampilkan 2 tabel data, yaitu tabel data pesanan dan tabel data pembayaran. pada tabel data pesanan terdiri dari *button* tambah pesanan , *client*, nomor handphone, paket, alamat total harga dan aksi yang berisi *button* bayar, *button* edit, *button* detail, *button* hapus. Kemudain pada tabel pembayaran terdiri dari nomor, *client*, metode pembayaran, nominal, bukti pembayaran, dan aksi berisi *button* cetak, *button* edit, *button* detail, *button* hapus. Rancangan tampilan halaman data pesanan dapat dilihat pada Gambar 3.53.

No	Client	No HP	Paket	Alamat	Total Harga	Aksi
1	Camelia	0000000000	Bronze	Indramayu	Rp. 00.0000	Bayar Edit Detail Hapus

No	Client	Metode Pembayaran	Paket	Nominal	Aksi
1	Camelia	Cash	Bronze	Rp. 00.0000	Cetak Edit Detail Hapus

Gambar 3.53 Rancangan Tampilan Halaman Data Pesanan Admin

4. Rancangan tampilan halaman data transportasi

Rancangan tampilan halaman data transportasi menampilkan 3 tabel data, yaitu tabel data kabupaten, tabel data transportasi, dan tabel data transportasi. pada tabel data kabupaten terdiri dari *button* tambah kabupaten nomor, nama kabupaten, dan aksi yang berisi *button* hapus. Lalu pada tabel data kecamatan terdiri dari *button* tambah kecamatan nomor, nama kecamatan, dan aksi yang berisi *button* hapus. kemudain pada tabel transaksi terdiri dari *button* tambah transportasi, nomor, nama

kabupaten, nama kecamatan, harga, dan aksi berisi *button edit* dan *button hapus*. Rancangan tampilan halaman data transportasi dapat dilihat pada Gambar 3.4.

The screenshot shows the 'Transportasi' section of the admin dashboard. It includes three tables:

- Data Kabupaten:** Shows a single entry for 'Indramayu' with an 'Aksi' button labeled 'Hapus'.
- Data Kecamatan:** Shows a single entry for 'Juntinyuat' with an 'Aksi' button labeled 'Hapus'.
- Data Transportasi:** Shows a single entry for 'Indramayu, Juntinyuat' with a price of 'Rp. 00.0000'. It has 'Edit' and 'Hapus' buttons.

Gambar 3.54 Rancangan Tampilan Halaman Data Transportasi Admin

5. Rancangan tampilan halaman data jadwal

Rancangan tampilan halaman data jadwal menampilkan tabel data jadwal yang terdiri dari nomor, nama *client*, nomor handphone *client*, kepentingan, pegawai, status, dan aksi yang berisi *button edit*, *button detail*, *button hapus*. Rancangan tampilan halaman data jadwal dapat dilihat pada Gambar 3.55.

The screenshot shows the 'Jadwal' section of the admin dashboard. It includes a table:

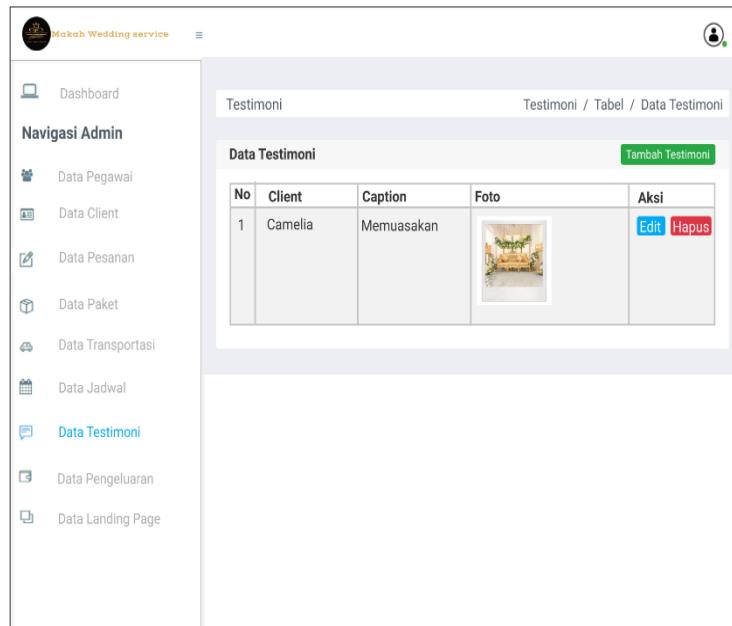
No	Client	No Hp	Kepentingan	Pegawai	Status	Aksi
1	Camelia	0000000000	Meninjau Lokasi	Inuy	Menunggu Informasi	Edit Detail Hapus

Gambar 3.55 Rancangan Tampilan Halaman Data Jadwal Admin

6. Rancangan tampilan halaman data testimoni

Rancangan tampilan halaman data testimoni menampilkan tabel data testimoni yang terdiri dari *button tambah testimoni*, nomor, nama *client*, *caption*

untuk testimoni *client*, foto testimoni, dan aksi yang berisi *button edit*, *button hapus*. Rancangan tampilan halaman data testimoni dapat dilihat pada Gambar 3.56.

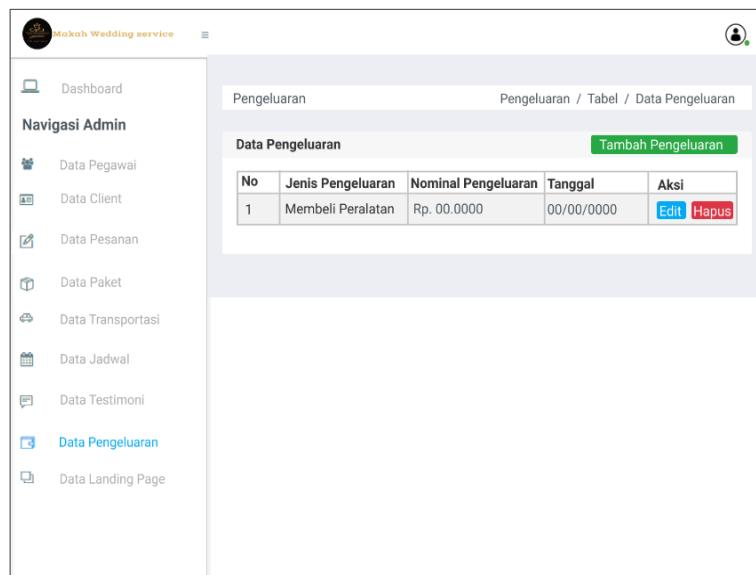


The screenshot shows a web application interface for 'Makah Wedding service'. On the left, there's a sidebar titled 'Navigasi Admin' with various menu items: Dashboard, Data Pegawai, Data Client, Data Pesanan, Data Paket, Data Transportasi, Data Jadwal, Data Testimoni (which is highlighted in blue), Data Pengeluaran, and Data Landing Page. The main content area has a header 'Testimoni' and a sub-header 'Testimoni / Tabel / Data Testimoni'. Below this is a table titled 'Data Testimoni' with columns: No, Client, Caption, Foto, and Aksi. There is one entry: No 1, Client Camelia, Caption Memuaskan, Foto (a small thumbnail image of a couple in a wedding setting), and Aksi (with 'Edit' and 'Hapus' buttons). At the top right of the main content area, there's a green button labeled 'Tambah Testimoni'.

Gambar 3.56 Rancangan Tampilan Halaman Data Testimoni Admin

7. Rancangan tampilan halaman data pengeluaran

Rancangan tampilan halaman data pengeluaran menampilkan tabel data pengeluaran yang terdiri dari *button tambah pengeluaran*, jumlah pengeluaran, nominal pengeluaran, tanggal, dan aksi yang berisi *button edit*, *button hapus*. Rancangan tampilan halaman data pengeluaran dapat dilihat pada Gambar 3.57.



The screenshot shows a web application interface for 'Makah Wedding service'. The sidebar and header are identical to the previous screenshot. The main content area has a header 'Pengeluaran' and a sub-header 'Pengeluaran / Tabel / Data Pengeluaran'. Below this is a table titled 'Data Pengeluaran' with columns: No, Jenis Pengeluaran, Nominal Pengeluaran, Tanggal, and Aksi. There is one entry: No 1, Jenis Pengeluaran Membeli Peralatan, Nominal Pengeluaran Rp. 00.0000, Tanggal 00/00/0000, and Aksi (with 'Edit' and 'Hapus' buttons). At the top right of the main content area, there's a green button labeled 'Tambah Pengeluaran'.

Gambar 3.57 Rancangan Tampilan Halaman Data Pengeluaran Admin

8. Rancangan tampilan halaman data *landing page*

Rancangan tampilan halaman data landing page menampilkan 6 tabel data, yaitu tabel data *background*, tabel data dokumentasi, tabel data tentang kami, tabel data *our services*, tabel data *question and answer*, dan tabel data *contact*. pada tabel data *background* terdiri judul, foto, dan aksi yang berisi *button edit*. kemudain pada tabel dokumentasi terdiri dari *button tambah dokumentasi*, judul, foto, dan aksi berisi *button edit*, *button hapus*. lalu pada tabel data tentang kami terdiri dari judul, deskripsi, foto, dan aksi yang berisi *button edit*. selanjutnya pada tabel *our services* terdiri dari *button tambah our services*, judul, deskripsi, dan aksi berisi *button edit*, *button hapus*. pada tabel *question and answer* terdiri dari *button tambah question and answer*, pertanyaan, jawaban, dan aksi berisi *button edit*, *button hapus*. lalu pada tabel data *contact* terdiri dari alamat, email, nomor handphone, dan Aksi yang berisi *button edit*. Rancangan tampilan halaman data transportasi dapat dilihat pada Gambar 3.58.

The screenshot shows the 'Landing Page' section of the 'Data Landing Page' in the 'Makah Wedding service' application. On the left, there's a sidebar titled 'Navigasi Admin' with various menu items. The main area displays two tables:

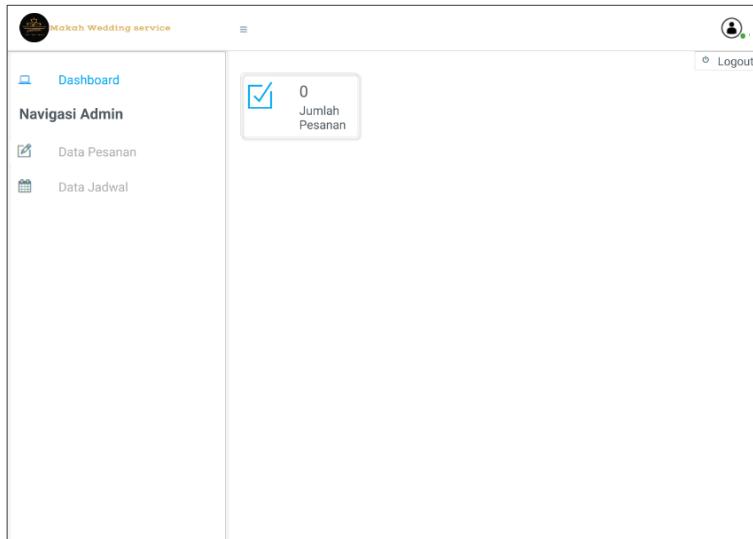
Judul	Foto	Aksi
Make Up Decoration Wedding Organizer		Edit

Judul	Foto	Aksi
Wedding Organizer		Edit Hapus

Gambar 3.58 Rancangan Tampilan Halaman Data *Landing Page* Admin

e. Rancangan Tampilan Halaman Pegawai

Rancangan tampilan halaman pegawai menampilkan data jumlah pesanan kemudian ada navigasi pegawai yang terdiri dari data pesanan dan data jadwal. Rancangan tampilan halaman pegawai dapat dilihat pada Gambar 3.59.



Gambar 3.59 Rancangan Tampilan Halaman Pegawai

1. Rancangan tampilan halaman data pesanan

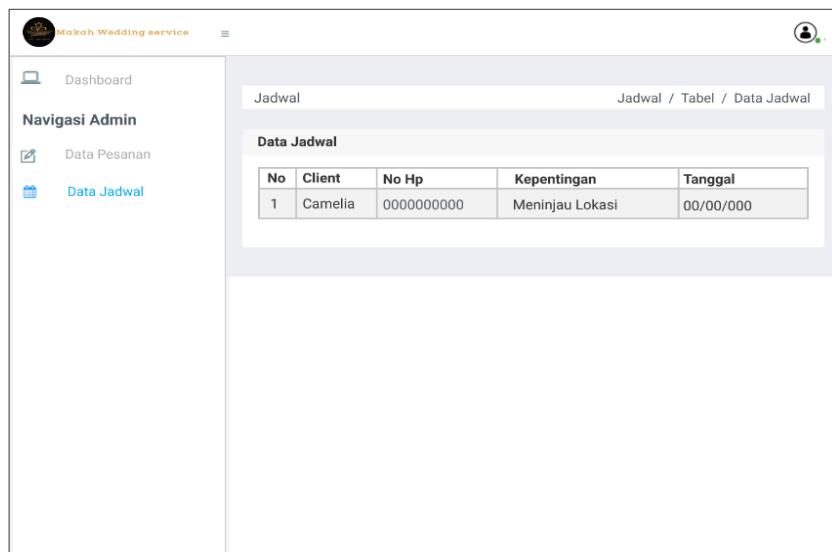
Rancangan tampilan halaman data pesanan menampilkan tabel data pesanan yang terdiri dari *button* tambah pesanan, nomor, *client*, nomor handphone, paket, kecamatan, alamat, dan aksi yang berisi *button* edit, *button* detail, *button* hapus. Rancangan tampilan halaman data pesanan dapat dilihat pada Gambar 3.60

Data Pesanan							Tambah Pesanan
No	Client	No Hp	Paket	Kecamatan	Alamat	Aksi	
1	Camelia	0000000000	Bronze	Juntinyuat	Indramayu	Edit Detail Hapus	

Gambar 3.60 Rancangan Tampilan Halaman Data Pesanan Pegawai

2. Rancangan tampilan halaman data jadwal

Rancangan tampilan halaman data jadwal menampilkan tabel data jadwal yang terdiri dari nomor, *client*, nomor handphone, kepentingan, dan tanggal. Rancangan tampilan halaman data jadwal dapat dilihat pada Gambar 3.61



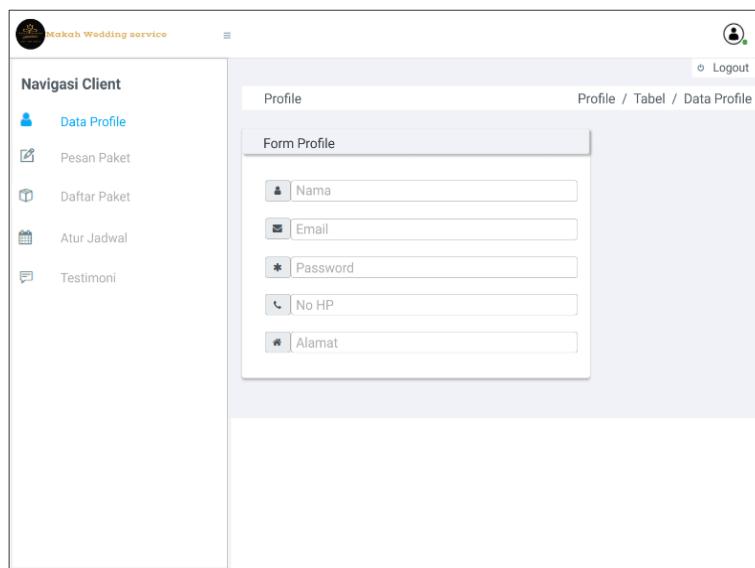
The screenshot shows a web application interface for 'Makah Wedding service'. On the left, there's a sidebar titled 'Navigasi Admin' with links for 'Dashboard', 'Data Pesanan', and 'Data Jadwal'. The main content area has a header 'Jadwal' and a breadcrumb 'Jadwal / Tabel / Data Jadwal'. Below this is a section titled 'Data Jadwal' containing a table with one row:

No	Client	No Hp	Kepentingan	Tanggal
1	Camelia	0000000000	Meninjau Lokasi	00/00/000

Gambar 3.61 Rancangan Tampilan Halaman Data Jadwal Pegawai

f. Rancangan Tampilan Halaman *Client*

Rancangan tampilan halaman *client* menampilkan data *from profile*. kemudian ada navigasi *client* yang terdiri dari data *profile*, pesan paket, daftar paket, atur jadwal, dan testimoni. Rancangan tampilan halaman *client* dapat dilihat pada Gambar 3.62.



The screenshot shows a web application interface for 'Makah Wedding service'. On the left, there's a sidebar titled 'Navigasi Client' with links for 'Data Profile', 'Pesan Paket', 'Daftar Paket', 'Atur Jadwal', and 'Testimoni'. The main content area has a header 'Profile' and a breadcrumb 'Profile / Tabel / Data Profile'. Below this is a form titled 'Form Profile' with fields for 'Nama', 'Email', 'Password', 'No HP', and 'Alamat'.

Gambar 3.62 Rancangan Tampilan Halaman *Client*

1. Rancangan tampilan halaman pesan paket

Rancangan tampilan halaman pesan paket menampilkan 2 tabel data, yaitu tabel data pemesanan dan tabel data pembayaran. Pada tabel data pesanan terdiri dari *button* tambah pemesanan nomor, nama, paket, alamat total harga dan aksi yang berisi *button* bayar, *button* edit, *button detail*, *button* hapus. Kemudian pada tabel pembayaran terdiri dari nomor, nama, metode pembayaran, pembayaran, nominal, dan aksi berisi *button* cetak, *button* edit, *button detail*, *button* hapus. Rancangan tampilan halaman data pesanan dapat dilihat pada Gambar 3.63.

No	Nama	Paket	Alamat	Total Harga	Aksi
1	Camelia	Bronze	Indramayu	Rp. 00.0000	Bayar Edit Detail Hapus

No	Nama	Metode Pembayaran	Pembayaran	Nominal	Aksi
1	Camelia	Transfer	DP 1 Untuk Blok Tanggal	Rp. 00.0000	Cetak Edit Detail Hapus

Gambar 3.63 Rancangan Tampilan Halaman Pesan Paket *Client*

2. Rancangan tampilan halaman daftar paket

Rancangan tampilan halaman daftar paket menampilkan tabel daftar paket yang terdiri dari nomor, jenis paket, nama paket, harga, dan aksi berisi deskripsi foto. Rancangan tampilan halaman daftar paket dapat dilihat pada Gambar 3.64.

No	Jenis Paket	Nama Paket	Harga	Deskripsi Foto
1	Wedding Organizer	Bronze	Rp. 00.0000	

Gambar 3.64 Rancangan Tampilan Halaman Daftar Paket *Client*

3. Rancangan tampilan halaman atur jadwal

Rancangan tampilan halaman atur jadwal menampilkan tabel data jadwal yang terdiri dari *button* tambah jadwal, nomor, nama, kepentingan, status, tanggal, alamat, dan aksi yang berisi *button* edit, *button* hapus. Rancangan tampilan halaman data jadwal dapat dilihat pada Gambar 3.65.

No	Nama	Kepentingan	Status	Tanggal	Aksi
1	Camelia	Meninjau Lokasi	Menunggu Informasi	00/00/000	Edit Hapus

Gambar 3.65 Rancangan Tampilan Halaman Jadwal *Client*

4. Rancangan tampilan halaman testimoni

Rancangan tampilan halaman testimoni menampilkan tabel data jadwal yang terdiri dari button tambah testimoni, nomor, nama, *caption*, foto, dan aksi yang berisi button edit, button hapus. Rancangan tampilan halaman data testimoni dapat dilihat pada Gambar 3.66.

No	Nama	Caption	Foto	Aksi
1	Camelia	Memuaskan		Edit Hapus

Gambar 3.66 Rancangan Tampilan Halaman Testimoni *Client*

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Penelitian ini menghasilkan sebuah Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel. Aplikasi ini digunakan untuk media informasi jasa yang disediakan pada Makah *Wedding Service* kemudian untuk membantu *client* untuk melakukan proses pemesanan pada Makah *Wedding Service* secara *online* agar tidak melakukan pertemuan secara langsung antara pemilik jasa Makah *Wedding Service* dengan *client*. Aplikasi ini juga bisa digunakan sebagai media promosi bisnis bagi Makah *Wedding Service*.

Dapat disimpulkan bahwa Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel yang telah dibuat merupakan media yang dapat dimanfaatkan bagi pemilik jasa Makah *Wedding Service* sebagai media informasi jasa yang disediakan pada Makah *Wedding Service*, dan untuk *client* dapat dimanfaatkan untuk melakukan pemesanan secara *online*.

4.2. Pembahasan

Tahapan ini membahas tahapan implementasi dari rancangan yang telah dibuat sebelumnya ke dalam program. Pada tahapan ini menjelaskan mengenai kesesuaian antara rancangan dengan hasil implementasi. Pada tahapan ini juga membahas mengenai hasil pengujian aplikasi, sehingga terciptalah Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel.

4.2.1. Implementasi

Setelah melakukan perancangan aplikasi, tahapan selanjutnya adalah implementasi. Pada tahapan ini rancangan yang telah dibuat akan diterjemahkan ke dalam bahasa pemrograman. Tahapan inilah dilakukan proses coding menggunakan bahasa pemrograman PHP dengan menggunakan *software* Visual Studio Code, kemudian pada tahapan ini juga diimplementasikan database melalui MySQL berdasarkan rancangan database dan ERD yang sudah dibuat sehingga menghasilkan Aplikasi Makah *Wedding Service* Berbasis *Website* yang dapat digunakan oleh pemilik jasa Makah *Wedding Service* dan *client*.

4.2.1.1. Implementasi *Database*

Terdapat 13 tabel yang digunakan untuk mengimplementasikan kebutuhan pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel. Dimana tabel-tabel tersebut merupakan tabel-tabel yang telah di bahas pada bagian rancangan *database*.

a. Implementasi Tabel Admin

Implementasi tabel admin pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.1.

Tabel 4.1 Implementasi Tabel Admin

1	username	varchar(255)	utf8mb4_unicode_ci	UNSIGNED
2	password	varchar(255)	utf8mb4_unicode_ci	
3	created_at	timestamp		
4	updated_at	timestamp		

b. Implementasi Tabel Pegawai

Implementasi tabel pegawai pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Implementasi Tabel Pegawai

1	id_pegawai	int(10)	UNSIGNED
2	username	varchar(255)	utf8mb4_unicode_ci
3	password	varchar(255)	utf8mb4_unicode_ci
4	nama	varchar(255)	utf8mb4_unicode_ci
5	email	varchar(255)	utf8mb4_unicode_ci
6	no_hp	varchar(255)	utf8mb4_unicode_ci
7	jobdesk	varchar(255)	utf8mb4_unicode_ci
8	alamat	varchar(255)	utf8mb4_unicode_ci
9	jk	varchar(255)	utf8mb4_unicode_ci
10	kontrak	varchar(255)	utf8mb4_unicode_ci
11	foto	varchar(255)	utf8mb4_unicode_ci
12	created_at	timestamp	
13	updated_at	timestamp	

c. Implementasi Tabel Konsumen

Implementasi tabel konsumen pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Implementasi Tabel Konsumen

1	id_konsumen	int(10)	UNSIGNED
2	email	varchar(255)	utf8mb4_unicode_ci
3	email_verify	varchar(255)	utf8mb4_unicode_ci
4	nama	varchar(255)	utf8mb4_unicode_ci
5	password	varchar(255)	utf8mb4_unicode_ci
6	no_hp	varchar(255)	utf8mb4_unicode_ci
7	alamat	varchar(255)	utf8mb4_unicode_ci
8	created_at	timestamp	
9	updated_at	timestamp	

d. Implementasi Tabel Pesanan

Pada tabel ini digunakan sebagai media untuk menyimpan data pesanan yang telah dipesan oleh *client*. Implementasi tabel pesanan pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.4.

Tabel 4.4 Implementasi Tabel Pesanan

1	id_pesanan	int(10)	UNSIGNED
2	id_konsumen	int(10)	UNSIGNED
3	id_paket	int(10)	UNSIGNED
4	id_pegawai	int(10)	UNSIGNED
5	id_transportasi	int(10)	UNSIGNED
6	alamat	varchar(255)	utf8mb4_unicode_ci
7	tgl	date	
8	created_at	timestamp	
9	updated_at	timestamp	

e. Implementasi Tabel Paket

Implementasi tabel paket pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi Tabel Paket

1	id_paket	int(10)	UNSIGNED
2	jenis	varchar(255)	utf8mb4_unicode_ci
3	deskripsi	varchar(255)	utf8mb4_unicode_ci
4	harga	int(11)	
5	foto	varchar(255)	utf8mb4_unicode_ci
6	created_at	timestamp	
7	updated_at	timestamp	

f. Implementasi Tabel Kabupaten

Implementasi tabel kabupaten pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.6.

Tabel 4.6 Implementasi Tabel Kabupaten

1	id_kabupaten	int(10)	UNSIGNED
2	nama	varchar(255)	utf8mb4_unicode_ci
3	created_at	timestamp	
4	updated_at	timestamp	

g. Implementasi Tabel Kecamatan

Implementasi tabel kecamatan pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.7.

Tabel 4.7 Implementasi Tabel Kecamatan

1	id_kecamatan	int(10)	UNSIGNED
2	id_kabupaten	int(10)	UNSIGNED
3	nama_kec	varchar(255)	utf8mb4_unicode_ci
4	created_at	timestamp	
5	updated_at	timestamp	

h. Implementasi Tabel Transportasi

Implementasi tabel transportasi pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.8.

Tabel 4.8 Implementasi Tabel Transportasi

1	id_transportasi	🔑 int(10)	UNSIGNED
2	id_kabupaten	int(10)	UNSIGNED
3	id_kecamatan	int(10)	UNSIGNED
4	harga	int(11)	
5	created_at	timestamp	
6	updated_at	timestamp	

i. Implementasi Tabel Transaksi

Implementasi tabel transaksi pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.9.

Tabel 4.9 Implementasi Tabel Transaksi

1	id_transaksi	🔑 int(10)	UNSIGNED
2	id_pesanan	🔑 int(10)	UNSIGNED
3	metode	varchar(255) utf8mb4_unicode_ci	
4	nominal	varchar(255) utf8mb4_unicode_ci	
5	pembayaran	varchar(255) utf8mb4_unicode_ci	
6	status	varchar(255) utf8mb4_unicode_ci	
7	tgl	date	
8	foto	varchar(255) utf8mb4_unicode_ci	
9	created_at	timestamp	
10	updated_at	timestamp	

j. Implementasi Tabel Jadwal

Implementasi tabel jadwal pada Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel dapat dilihat pada Tabel 4.10.

Tabel 4.10 Implementasi Tabel Jadwal

1	id_jadwal	🔑 int(10)	UNSIGNED
2	id_konsumen	🔑 int(10)	UNSIGNED
3	id_pegawai	🔑 int(10)	UNSIGNED
4	tgl	date	
5	kepentingan	varchar(255) utf8mb4_unicode_ci	
6	status	varchar(255) utf8mb4_unicode_ci	
7	created_at	timestamp	
8	updated_at	timestamp	

k. Implementasi Tabel Testimoni

Implementasi tabel testimoni pada Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel dapat dilihat pada Tabel 4.11.

Tabel 4.11 mplementasi Tabel Testimoni

1	id_testimoni	int(10)	UNSIGNED
2	id_konsumen	int(10)	UNSIGNED
3	caption	varchar(255)	utf8mb4_unicode_ci
4	foto	varchar(255)	utf8mb4_unicode_ci
5	created_at	timestamp	
6	updated_at	timestamp	

l. Implementasi Tabel *Landing*

Implementasi tabel *landing* pada Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel dapat dilihat pada Tabel 4.12.

Tabel 4.12 mplementasi Tabel *Landing*

1	id_landing	int(10)	UNSIGNED
2	bagian	varchar(255)	utf8mb4_unicode_ci
3	judul	varchar(255)	utf8mb4_unicode_ci
4	foto	varchar(255)	utf8mb4_unicode_ci
5	deskripsi	longtext	utf8mb4_unicode_ci
6	alamat	varchar(255)	utf8mb4_unicode_ci
7	pertanyaan	varchar(255)	utf8mb4_unicode_ci
8	jawaban	varchar(255)	utf8mb4_unicode_ci
9	email	varchar(255)	utf8mb4_unicode_ci
10	no_hp	varchar(255)	utf8mb4_unicode_ci
11	created_at	timestamp	
12	updated_at	timestamp	

m. Implementasi Tabel Pengeluaran

Implementasi tabel pengeluaran pada Aplikasi Makah *Wedding Service* Berbasis Website Dengan *Framework* Laravel dapat dilihat pada Tabel 4.13.

Tabel 4.13 Implementasi Tabel Pengeluaran

1	id_pengeluaran	int(10)	UNSIGNED
2	jenis	varchar(255)	utf8mb4_unicode_ci
3	nominal	int(11)	
4	tgl	date	
5	created_at	timestamp	
6	updated_at	timestamp	

4.2.1.2. Implementasi *Design Interface*

Implementasi *design interface* dan program merupakan tahap untuk mengimplementasikan *design* yang sudah dirancang sebelumnya. *design Interface* merupakan cara untuk memudahkan pengguna dalam melakukan aktivitas pada aplikasi.

a. Implementasi Tampilan Halaman *Landing Page*

Pada saat *website* dibuka maka yang pertama kali akan muncul adalah halaman *landing page*. Pada halaman *landing page* terdapat tampilan informasi *company profile*, jasa yang disediakan, paket-paket beserta harganya, testimoni dari *client*, dan data *contact* yang bisa dihubungi oleh Makah *Wedding Service*. Implementasi tampilan halaman *landing page* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi Tampilan Halaman *Landing Page*

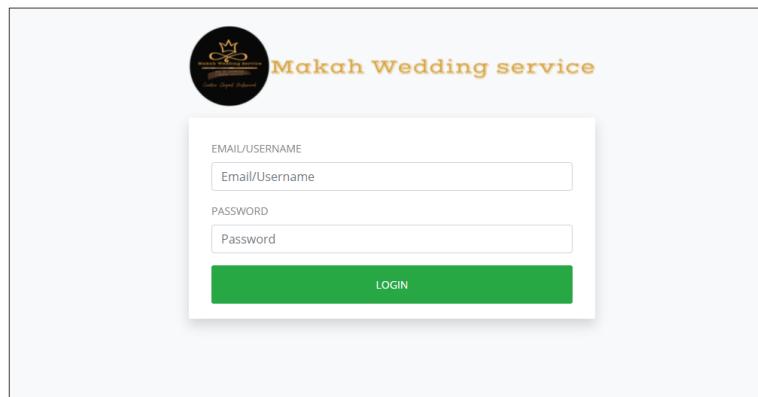
b. Implementasi Tampilan Halaman *Registrasi*

Halaman registrasi akan muncul apabila *client* ingin melakukan *login* jika belum membuat sebuah akun pada *website* ini maka tidak bisa melakukan *login* karena belum terdaftar, dan harus melakukan registrasi terlebih dahulu, halaman registrasi terdapat *form* registrasi yang terdiri dari nama, email, *password*, nomor handphone, dan alamat yang harus dilengkapi oleh *client* agar berhasil membuat akun. Implementasi tampilan halaman *regitrasи* dapat dilihat pada Gambar 4.2.

Gambar 4.2 Implementasi Tampilan Halaman *Registrasi*

c. Implementasi Tampilan Halaman *Login*

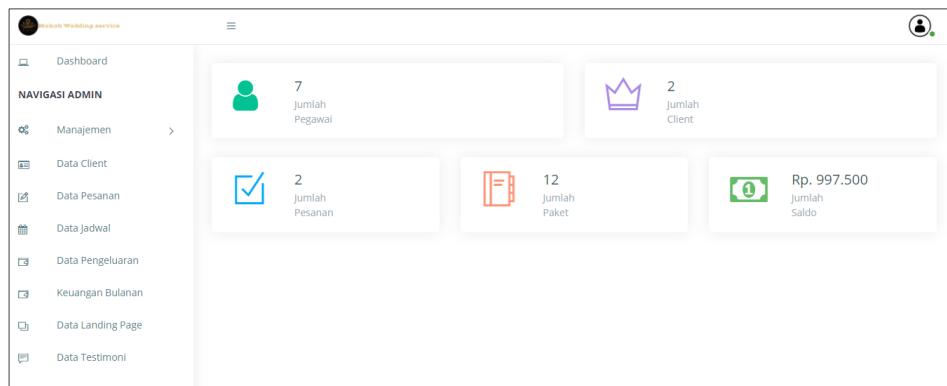
Setelah melakukan *registrasi* maka akan bisa melakukan *login* untuk *client*, halaman *login* terdapat *form login* yang terdiri dari email dengan *password* yang harus dilengkapi oleh *client* agar berhasil melakukan *login*, kemudian untuk admin dengan pegawai melakukan *login* dengan melengkapi *form login* yang terdiri dari *username* dengan *password* yang harus dilengkapi oleh admin dan pegawai agar berhasil melakukan *login*. Implementasi tampilan halaman *login* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Implementasi Tampilan Halaman *Login*

d. Implementasi Tampilan Halaman Admin

Setelah admin sudah berhasil melakukan *login* maka akan muncul halaman admin. Pada halaman admin menampilkan jumlah pegawai, jumlah *client*, jumlah pesanan, jumlah paket, dan jumlah keuangan. Kemudian ada juga navigasi admin yang terdiri dari manajemen, data *client*, data pesanan, data jadwal, data pengeluaran, data *landing page*, dan data testimoni. Implementasi tampilan halaman admin dapat dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi Tampilan Halaman *Admin*

1. Implementasi tampilan halaman data pegawai

Halaman data pegawai menampilkan data-data pegawai, seperti nama, *job description*, status pegawai, dan nomor handphone. Selain itu juga bisa menambahkan data pegawai, edit data pegawai, melihat *detail* data pegawai, dan hapus data pegawai. Implementasi tampilan halaman data pegawai dapat dilihat pada Gambar 4.5.

No	Nama	Job Description	Status Pegawai	No Hp	Aksi
1	Inuy	Consultant dan Wedding Planner	Pegawai Tetap	083822765867	<button>Edit</button> <button>Detail</button> <button>Hapus</button>
2	Makhrus	Wedding Project Manager	Pegawai Tetap	0897840000	<button>Edit</button> <button>Detail</button> <button>Hapus</button>
3	Desi	Pendamping Pengantin	Pegawai Tetap	0897843799	<button>Edit</button> <button>Detail</button> <button>Hapus</button>
4	Aep	Wedding Decorator	Pegawai Tetap	0897843700	<button>Edit</button> <button>Detail</button> <button>Hapus</button>
5	Gina	Wedding Organizer	Pegawai Tetap	0897843799	<button>Edit</button> <button>Detail</button> <button>Hapus</button>
6	Yulis	Wedding Stylist	Pegawai Tetap	0897843722	<button>Edit</button> <button>Detail</button> <button>Hapus</button>

Gambar 4.5 Implementasi Tampilan Halaman Data Pegawai Admin

2. Implementasi tampilan halaman data *client*

Halaman data *client* menampilkan data-data *client*, seperti nama, email, nomor handphone, dan alamat. Selain itu juga bisa menambahkan data *client*, edit data *client*, melihat *detail* data *client*, dan hapus data *client*. Implementasi tampilan halaman data *client* dapat dilihat pada Gambar 4.6.

No	Nama	Email	No HP	Alamat	Aksi
1	camelia	cameliacamelia231@gmail.com	083822765867	Indramayu	<button>Edit</button> <button>Detail</button> <button>Hapus</button>
2	Amelia	cameli14cmill@gmail.com	0897843700	Indramayu	<button>Edit</button> <button>Detail</button> <button>Hapus</button>

Gambar 4.6 Implementasi Tampilan Halaman Data *Client* Admin

3. Implementasi tampilan halaman data pesanan

Halaman data pesanan menampilkan data-data pesanan dan data-data pembayaran. Pada data-data pesanan menampilkan data-data seperti nama *client*,

nomor handphone *client*, paket yang telah dipilih oleh *client*, total harga yang harus dibayar *client*, dan kekurangan pembayaran pesanan. Selain itu juga bisa menambahkan data pesanan akan tetapi jika tanggal yang dipesan sudah *full booking* maka data pesanan tidak bisa ditambahkan, edit data pesanan, melihat *detail* data pesanan, dan hapus data pesanan. Kemudian pada data-data pembayaran menampilkan data-data pembayaran seperti nama *client*, status pembayaran, paket yang telah dipilih oleh *client*, dan nominal pembayaran. Selanjutnya bisa juga cetak data pembayaran, edit data pembayaran, melihat *detail* data pembayaran, dan hapus data pembayaran. Implementasi tampilan halaman data pesanan dapat dilihat pada Gambar 4.7.

The screenshot shows the 'Data Pesanan' section of the admin interface. It displays a table of order details with columns: No, Client, No Hp, Paket, Total Harga, Kurang, and Aksi. Below this is a 'Data Pembayaran' section with a table showing columns: No, Client, Status Pembayaran, Paket, Nominal, and Aksi. A red message box at the top states: 'Data Pesanan gagal ditambah karena tanggal telah full booking'. A green 'Tambah Pesanan' button is located at the top right of the 'Data Pesanan' section.

No	Client	No Hp	Paket	Total Harga	Kurang	Aksi
1	camelia	085822765867	Diamond -2 Package	Rp. 29.800.000	Rp. 28.800.000	<button>Bayar</button> <button>Edit</button> <button>Detail</button> <button>Hapus</button>
2	Pipit	085822765867	Gold -2 Package	Rp. 22.500.000	Rp. 22.500.000	<button>Bayar</button> <button>Edit</button> <button>Detail</button> <button>Hapus</button>
3	alvin	08765325902	Gold -2 Package	Rp. 21.500.000	Rp. 21.500.000	<button>Bayar</button> <button>Edit</button> <button>Detail</button> <button>Hapus</button>
4	Amelia	0897843700	Gold -2 Package	Rp. 22.500.000	Rp. 22.500.000	<button>Bayar</button> <button>Edit</button> <button>Detail</button> <button>Hapus</button>

No	Client	Status Pembayaran	Paket	Nominal	Aksi
----	--------	-------------------	-------	---------	------

Gambar 4.7 Implementasi Tampilan Halaman Data Pesanan Admin

4. Implementasi tampilan halaman data paket

Halaman data paket menampilkan data-data paket, seperti jenis paket, nama paket, dan harga paket. Selain itu juga bisa menambahkan data paket, edit data paket, dan hapus data paket. Implementasi tampilan halaman data paket dapat dilihat pada Gambar 4.8.

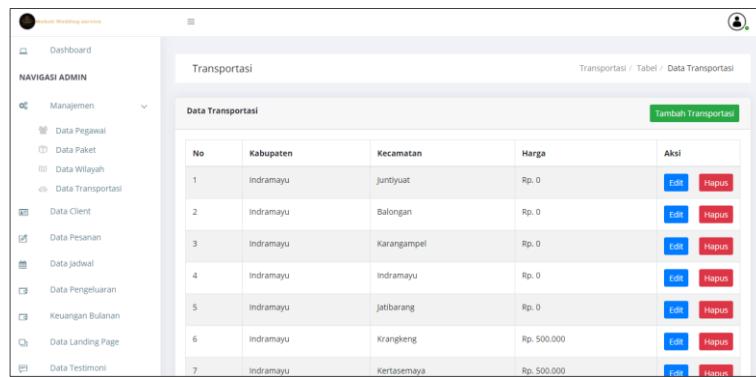
The screenshot shows the 'Paket' section of the admin interface. It displays a table of package details with columns: No, Jenis Paket, Nama Paket, Harga, Deskripsi Foto, and Aksi. Each row includes a thumbnail image of the package. A green 'Tambah Paket' button is located at the top right of the 'Data Paket' section.

No	Jenis Paket	Nama Paket	Harga	Deskripsi Foto	Aksi
1	Wedding Organizer	Mini dekor Package	Rp. 6.500.000		<button>Edit</button> <button>Hapus</button>
2	Wedding Organizer	Silver -2 Package	Rp. 17.000.000		<button>Edit</button> <button>Hapus</button>
3	Wedding Organizer	Silver -1 Package	Rp. 19.500.000		<button>Edit</button> <button>Hapus</button>

Gambar 4.8 Implementasi Tampilan Halaman Data Paket Admin

5. Implementasi tampilan halaman data transportasi

Halaman data transportasi menampilkan data-data transportasi, seperti nama kabupaten, nama kecamatan, dan harga. Selain itu juga bisa menambahkan data transportasi, edit data transportasi, dan hapus data transportasi. Implementasi tampilan halaman data transportasi dapat dilihat pada Gambar 4.9.

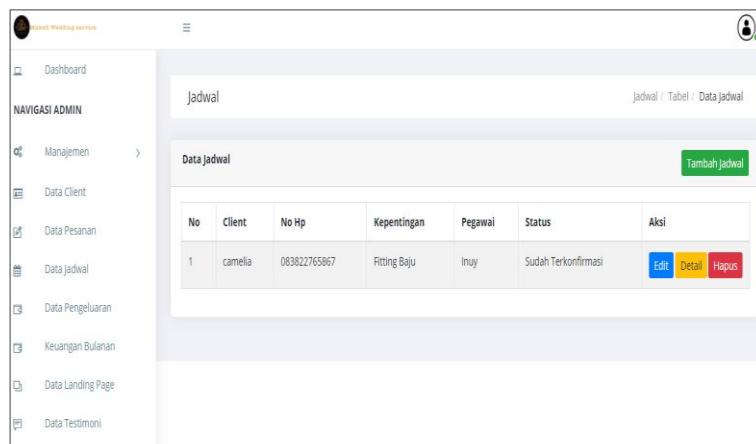


No	Kabupaten	Kecamatan	Harga	Aksi
1	Indramayu	Juntiyuat	Rp. 0	<button>Edit</button> <button>Hapus</button>
2	Indramayu	Balongan	Rp. 0	<button>Edit</button> <button>Hapus</button>
3	Indramayu	Karangampel	Rp. 0	<button>Edit</button> <button>Hapus</button>
4	Indramayu	Indramayu	Rp. 0	<button>Edit</button> <button>Hapus</button>
5	Indramayu	Jatibarang	Rp. 0	<button>Edit</button> <button>Hapus</button>
6	Indramayu	Krangkeng	Rp. 500.000	<button>Edit</button> <button>Hapus</button>
7	Indramayu	Kertasemaya	Rp. 500.000	<button>Edit</button> <button>Hapus</button>

Gambar 4.9 Implementasi Tampilan Halaman Data Transportasi Admin

6. Implementasi tampilan halaman data jadwal

Halaman data jadwal menampilkan data-data jadwal, seperti nama *client*, nomor handphone *client*, kepentingan, nama pegawai, dan status. Selain itu juga bisa menambahkan data jadwal, edit data jadwal, dan hapus data jadwal. Implementasi tampilan halaman data jadwal dapat dilihat pada Gambar 4.10.



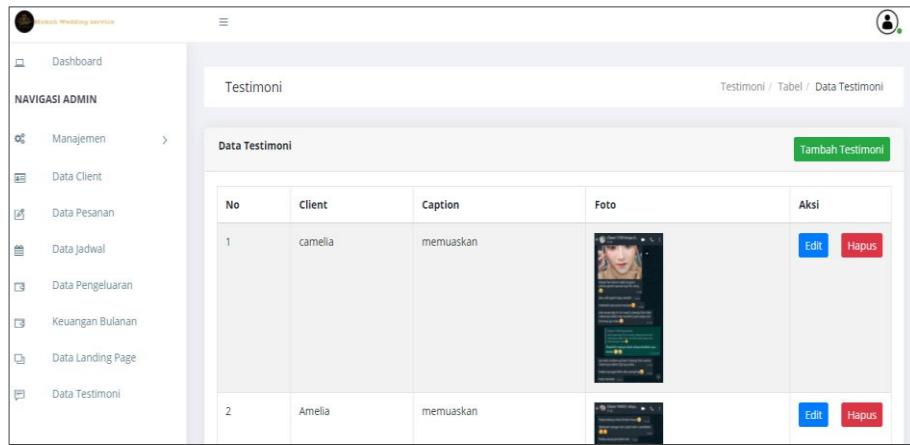
No	Client	No Hp	Kepentingan	Pegawai	Status	Aksi
1	camelia	083822765867	Fitting Baju	Inuy	Sudah Terkonfirmasi	<button>Edit</button> <button>Detail</button> <button>Hapus</button>

Gambar 4.10 Implementasi Tampilan Halaman Data Jadwal Admin

7. Implementasi tampilan halaman data testimoni

Halaman data testimoni menampilkan data-data testimoni, seperti nama *client*, *caption*, dan foto. Selain itu juga bisa menambahkan data testimoni, edit

data testimoni, dan hapus data testimoni. Implementasi tampilan halaman data testimoni dapat dilihat pada Gambar 4.11.

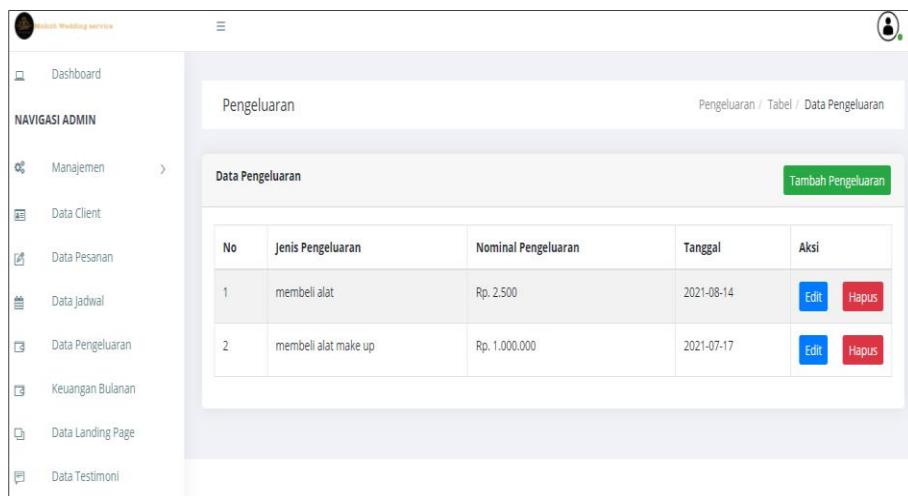


No	Client	Caption	Foto	Aksi
1	camelia	memuaskan		<button>Edit</button> <button>Hapus</button>
2	Amelia	memuaskan		<button>Edit</button> <button>Hapus</button>

Gambar 4.11 Implementasi Tampilan Halaman Data Testimoni Admin

8. Implementasi tampilan halaman data pengeluaran

Halaman data pengeluaran menampilkan data-data pengeluaran, seperti jenis pengeluaran, nominal pengeluaran, dan tanggal pengeluaran. Selain itu juga bisa menambahkan data pengeluaran, edit data pengeluaran, dan hapus data pengeluaran. Implementasi tampilan halaman data pengeluaran dapat dilihat pada Gambar 4.12.



No	Jenis Pengeluaran	Nominal Pengeluaran	Tanggal	Aksi
1	membeli alat	Rp. 2.500	2021-08-14	<button>Edit</button> <button>Hapus</button>
2	membeli alat make up	Rp. 1.000.000	2021-07-17	<button>Edit</button> <button>Hapus</button>

Gambar 4.12 Implementasi Tampilan Halaman Data Pengeluaran Admin

9. Implementasi tampilan halaman data landing page

Halaman data *landing page* menampilkan data-data tabel *landing page*. Terdiri dari tabel data *background*, tabel data dokumentasi, tabel data tentang kami,

tabel data *our service*, tabel data *question and answer*, dan tabel data *contact*. Implementasi tampilan halaman data pengeluaran dapat dilihat pada Gambar 4.13.

The screenshot shows the 'Landing' page for an admin. On the left, there's a sidebar with 'NAVIGASI ADMIN' containing links like 'Dashboard', 'Manajemen', 'Data Client', 'Data Pesanan', 'Data Jadwal', 'Data Pengeluaran', 'Keuangan Bulanan', 'Data Landing Page', and 'Data Testimoni'. The main area is titled 'Landing' and shows two tables:

- Data Background:** A table with columns 'Judul', 'Foto', and 'Aksi'. It contains one row for 'Make Up | Decoration | Wedding Organizer' with a photo thumbnail and 'Edit' button.
- Data Dokumentasi:** A table with columns 'Judul', 'Foto', and 'Aksi'. It contains one row for 'Wedding Organizer' with a photo thumbnail and 'Edit' and 'Hapus' buttons. There's also a green 'Tambah Dokumentasi' button.

Gambar 4.13 Implementasi Tampilan Halaman Data *Landing Page* Admin

e. Implementasi Tampilan Halaman Pegawai

Setelah pegawai sudah berhasil melakukan login maka akan muncul halaman pegawai. Pada halaman admin menampilkan jumlah pesanan. Kemudian ada juga navigasi pegawai yang terdiri dari data pesanan dan data jadwal. Implementasi tampilan halaman pegawai dapat dilihat pada Gambar 4.14.

The screenshot shows the 'Landing' page for an employee. On the left, there's a sidebar with 'NAVIGASI PEGAWAI' containing links like 'Dashboard', 'Data Pesanan', and 'Data Jadwal'. The main area shows a summary: a blue document icon followed by the number '1' and the text 'Jumlah Pesanan'.

Gambar 4.14 Implementasi Tampilan Halaman Pegawai

1. Implementasi tampilan halaman data pesanan pegawai

Halaman data pesanan pegawai menampilkan data-data pesanan, seperti nama *client*, nomor handphone *client*, nama paket, kecamatan *client* dan alamat

lengkap *client*. Selain itu juga bisa menambahkan data pesanan tetapi jika tanggal yang dipesan sudah *full booking* maka data pesanan tidak bisa ditambahkan, edit data pesanan, melihat *detail* pesanan, dan hapus data pesanan. Implementasi tampilan halaman data pesanan pegawai dapat dilihat pada Gambar 4.15.

No	Client	No Hp	Paket	Kecamatan	Alamat	Aksi
1	camelia	083822765867	Diamond -2 Package	Bongas	Indramayu	Edit Detail Hapus

Gambar 4.15 Implementasi Tampilan Halaman Pesanan Pegawai

2. Implementasi tampilan halaman data jadwal pegawai

Halaman data jadwal pegawai menampilkan data-data jadwal, seperti nama, nomor handphone, nama paket, kepentingan dan tanggal jadwal. Implementasi tampilan halaman data jadwal pegawai dapat dilihat pada Gambar 4.16.

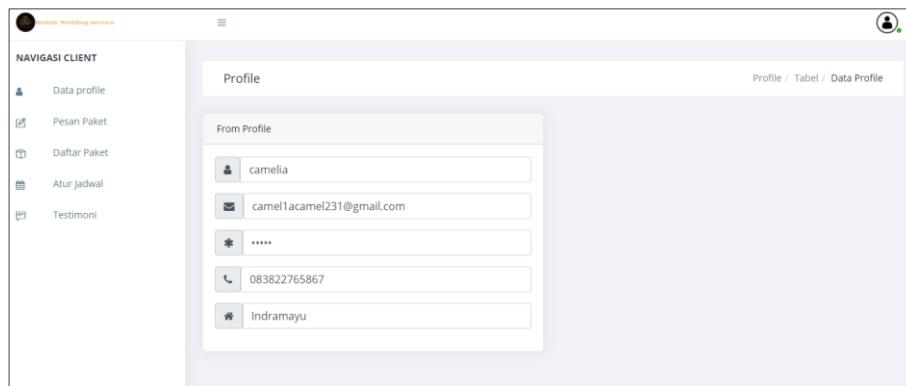
No	Client	No Hp	Kepentingan	Tanggal
1	camelia	083822765867	Fitting Baju	2021-08-18

Gambar 4.16 Implementasi Tampilan Halaman Jadwal Pegawai

f. Implementasi Tampilan Halaman *Client*

Setelah *client* sudah berhasil melakukan *login* maka akan muncul halaman *client*. Pada halaman admin menampilkan navigasi *client* yang terdiri dari data

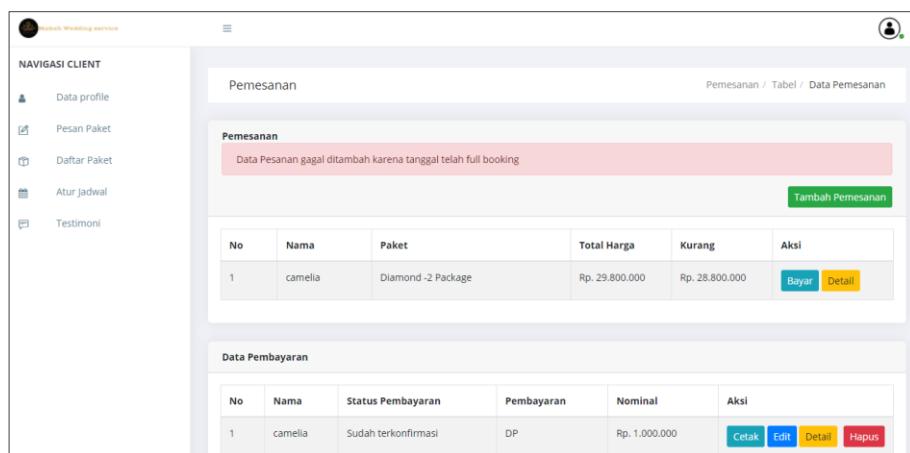
profile, pesan paket, daftar paket, atur jadwal, dan testimoni. Implementasi tampilan halaman pegawai dapat dilihat pada Gambar 4.16.



Gambar 4.17 Implementasi Tampilan Halaman *Client*

1. Implementasi tampilan halaman pesan paket *Client*

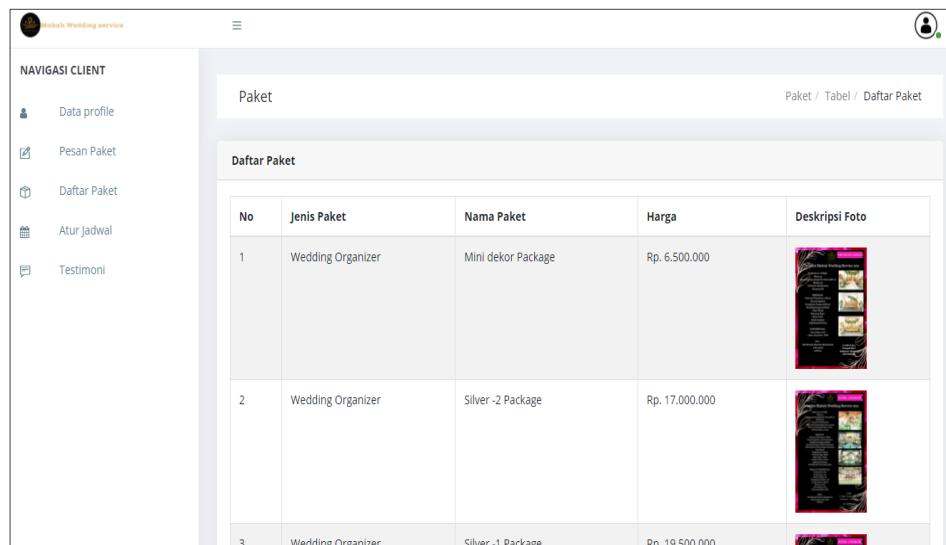
Halaman data pesan paket *client* menampilkan data-data pemesanan dan data-data pembayaran. Pada data-data pemesanan menampilkan data-data seperti nama *client*, paket yang telah dipilih oleh *client*, total harga pemesanan, dan kekurangan pembayaran. Selain itu juga bisa menambahkan data pemesanan akan tetapi jika tanggal yang dipesan sudah *full booking* maka data pesanan tidak bisa ditambahkan, edit data pemesanan, melihat *detail* data pemesanan, dan hapus data pemesanan. Kemudian pada data-data pembayaran menampilkan data-data pembayaran seperti nama *client*, status pembayaran, paket yang telah dipilih oleh *client*, dan nominal pembayaran. Selanjutnya bisa juga cetak data pembayaran, edit data pembayaran, melihat *detail* data pembayaran, dan hapus data pembayaran. Implementasi tampilan halaman data pesanan dapat dilihat pada Gambar 4.18.



Gambar 4.18 Implementasi Tampilan Halaman pesan paket *Client*

2. Implementasi tampilan halaman daftar paket *Client*

Halaman daftar paket *client* menampilkan data-data paket, seperti jenis paket, nama paket, harga paket, dan deskripsi foto. Implementasi tampilan halaman daftar paket *client* dapat dilihat pada Gambar 4.19.

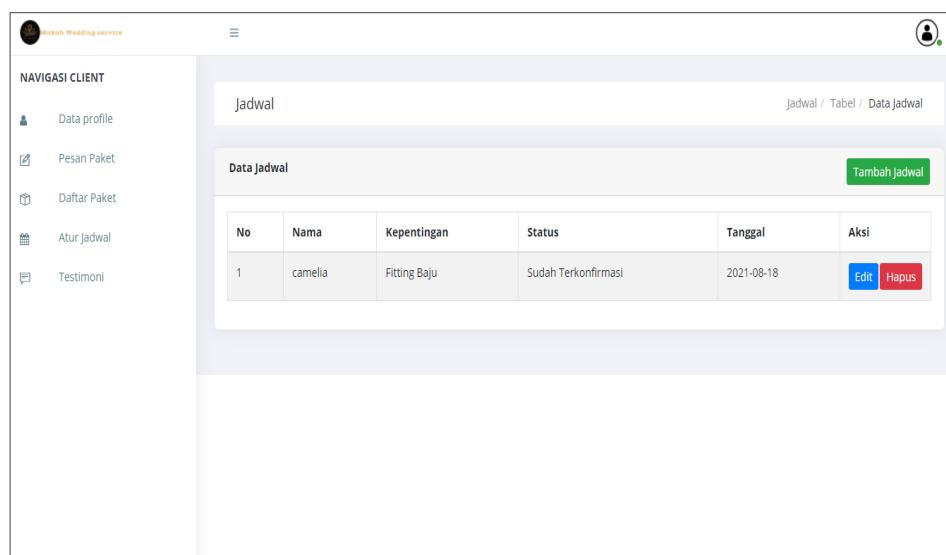


No	Jenis Paket	Nama Paket	Harga	Deskripsi Foto
1	Wedding Organizer	Mini dekor Package	Rp. 6.500.000	
2	Wedding Organizer	Silver -2 Package	Rp. 17.000.000	
3	Wedding Organizer	Silver -1 Package	Rp. 19.500.000	

Gambar 4.19 Implementasi Tampilan Halaman Daftar Paket *Client*

3. Implementasi tampilan halaman atur jadwal *Client*

Halaman atur jadwal *client* menampilkan data-data jadwal, seperti nama *client*, no.hp *client*, kepentingan, nama pegawai, dan status. Selain itu juga bisa menambahkan data jadwal, edit data jadwal, dan hapus data jadwal. Implementasi tampilan halaman atur jadwal *client* dapat dilihat pada Gambar 4.20.



No	Nama	Kepentingan	Status	Tanggal	Aksi
1	camelia	Fitting Baju	Sudah Terkonfirmasi	2021-08-18	<button>Edit</button> <button>Hapus</button>

Gambar 4.20 Implementasi Tampilan Halaman Atur Jadwal *Client*

4. Implementasi tampilan halaman testimoni *Client*

Halaman testimoni *client* menampilkan data-data testimoni, seperti nama *client*, *caption*, dan foto. Selain itu juga bisa menambahkan data testimoni, edit data testimoni, dan hapus data testimoni. Implementasi tampilan halaman data testimoni dapat dilihat pada Gambar 4.21.

No	Nama	Caption	Foto	Aksi
1	camella	memuaskan		Edit Hapus

Gambar 4.21 Implementasi Tampilan Halaman Testimoni *Client*

4.3. Pengujian Aplikasi

Tahapan berikutnya adalah pengujian aplikasi. Pada tahapan ini dilakukan pengujian terhadap hasil implementasi pada Aplikasi Makah *Wedding Service* Berbasis *Website*. Tujuan tahapan pengujian aplikasi ini untuk menentukan bahwa setiap fitur yang ada pada aplikasi telah berjalan kemudian palikasi dapat mengatasi permasalahan yang ada pada Makah *Wedding Service*. Pengujian pada aplikasi ini dilakukan dengan metode pengujian kuesioner sebagai acuan keberhasilan pada aplikasi.

a. Kuesioner

Pengujian melalui kuesioner dilakukan kepada 20 orang sebagai pengujian yang menilai hasil implementasi *website* sebagai acuan keberhasilan pembuatan *website*. Dari 20 orang yang melakukan pengujian melalui kuesioner diantaranya merupakan pemilik dengan pegawai pada Makah *Wedding Service* sebagai mitra. Tujuan dari pengujian ini adalah untuk memastikan kesesuaian aplikasi yang telah dibuat dengan kebutuhan yang diperlukan oleh pengguna khususnya pemilik dengan pegawai pada Makah *Wedding Service*.

1. Keterangan Nilai

Pada penilaian yang akan dilakukan menggunakan keterangan index nilai A, B, C, dan D. keterangan nilai terdapat pada Tabel 4.17.

Tabel 4.14 Keterangan Nilai Kuesioner

Index Nilai	Keterangan
A	Sangat Baik
B	Baik
C	Cukup
D	Kurang

2. Pertanyaan Kuesioner

Pada pertanyaan kuesioner terdapat 10 daftar pertanyaan untuk melakukan pengujian melalui kuesioner ini yang meliputi dari keseluruhan komponen pengujian. Pertanyaan kuesioner tersebut terdapat pada Tabel 4.18.

Tabel 4.15 Pertanyaan Kuesioner

No	Pertanyaan	A	B	C	D
1	Bagaimana pendapat anda mengenai tampilan <i>interface website</i> ini?				
2	Apakah <i>website</i> ini membantu bagi calon pengantin dalam menentukan jasa layanan <i>Wedding Organizer</i> ?				
3	Apakah <i>website</i> ini memberikan kemudahan bagi <i>client</i> dalam proses pemesanan jasa yang disediakan pada Makah <i>Wedding Service</i> secara <i>online</i> ?				
4	Apakah <i>website</i> ini memberikan kemudahan bagi masyarakat luas dalam mendapatkan informasi jasa yang disediakan pada Makah <i>Wedding Service</i> ?				
5	Apakah <i>website</i> ini memberikan kemudahan bagi pemilik dalam memberikan informasi jasa yang disediakan pada Makah <i>Wedding Service</i> ?				
6	Apakah <i>website</i> ini menjadi salah satu sarana bagi pemilik dalam menerapkan strategi pemasaran secara <i>online</i> selama masa pandemi Covid-19?				
7	Apakah setiap fitur yang disediakan berfungsi dengan baik?				

8	Apakah <i>website</i> ini mudah digunakan?				
9	Apakah website ini memberikan manfaat bagi pengguna?				
10	Apakah website ini mudah dipahami bagi pengguna?				

3. Hasil Jawaban Kuesioner

Hasil dari jawaban kuesioner ini didapatkan berdasarkan penilaian yang telah diberikan oleh penguji terhadap *website* dari kuesioner. Kemudian hasil tersebut disajikan dalam bentuk tabel yang terdapat pada Tabel 4.19.

Tabel 4.16 Jawaban Kuesioner

No	Nama	Pertanyaan									
		1	2	3	4	5	6	7	8	9	10
1	Nur Khalimah (Pemilik)	A	A	A	A	A	B	A	A	A	A
2	Makhrus (Pemilik)	A	A	A	A	A	A	A	A	A	A
3	Gina Ari Amanda (Pegawai)	A	A	A	A	A	B	A	A	A	A
4	Nur Ismail (Pegawai)	A	A	A	A	A	A	A	A	A	A
5	Duranto A. (Pegawai)	A	A	A	A	A	A	B	B	A	B
6	Saefudin (Pegawai)	A	A	A	A	A	A	A	A	A	A
7	Abdul Syakur (Pegawai)	A	A	A	A	A	A	A	A	A	A
8	Ade Syaeful Anwar (Pegawai)	A	A	A	A	A	A	A	A	A	A
9	Zacxy (Pegawai)	A	A	A	A	A	A	A	A	A	A
10	Adday Agung Alfayni (Masyarakat)	A	A	A	A	A	A	B	A	A	B
11	Afriyani (Masyarakat)	A	A	A	B	A	B	A	A	A	A

No	Nama	Pertanyaan									
		1	2	3	4	5	6	7	8	9	10
12	Iif Khofifah (Masyarakat)	A	A	A	B	A	A	A	A	A	A
13	Egi Sofyan Haryadi (Masyarakat)	B	A	B	B	A	A	B	A	A	B
14	Siti Fajriyah (Masyarakat)	A	A	B	B	A	A	B	B	A	B
15	Rohmatul Jannah (Masyarakat)	A	A	A	A	A	A	A	A	A	A
16	Desi Sri Wahyuni (Masyarakat)	A	A	B	A	B	A	A	A	A	A
17	Oktoviani (Masyarakat)	A	B	B	A	B	B	B	C	B	B
18	Sri Devika Meina (Masyarakat)	A	B	B	A	A	B	B	B	B	B
19	Indri Suhartini (Masyarakat)	A	B	A	A	B	A	A	A	B	A
20	Padlan Raya (Masyarakat)	B	A	A	A	A	A	A	B	A	A

4. Hasil Perhitungan Kuesioner

Berikut ini adalah hasil dari perhitungan kuesioner dari Aplikasi Makah *Wedding Service Berbasis Website*. Kemudian hasil tersebut disajikan dalam bentuk tabel yang terdapat pada Tabel 4.20.

Tabel 4.17 Hasil Perhitungan Kuesioner

No	Pertanyaan	A	B	C	D
1	Bagaimana pendapat anda mengenai tampilan <i>interface website</i> ini?	18	2	0	0
2	Apakah <i>website</i> ini membantu bagi calon pengantin dalam menentukan jasa layanan <i>Wedding Organizer</i> ?	17	3	0	0

3	Apakah website ini memberikan kemudahan bagi client dalam proses pemesanan jasa yang disediakan pada Makah Wedding Service secara online?	15	5	0	0
4	Apakah website ini memberikan kemudahan bagi masyarakat luas dalam mendapatkan informasi jasa yang disediakan pada Makah Wedding Service?	16	4	0	0
5	Apakah website ini memberikan kemudahan bagi pemilik dalam memberikan informasi jasa yang disediakan pada Makah Wedding Service?	17	3	0	0
6	Apakah website ini menjadi salah satu sarana bagi pemilik dalam menerapkan strategi pemasaran secara online selama masa pandemi Covid-19?	15	5	0	0
7	Apakah setiap fitur yang disediakan berfungsi dengan baik?	14	6	0	0
8	Apakah website ini mudah digunakan?	16	3	1	0
9	Apakah website ini memberikan manfaat bagi pengguna?	17	3	0	0
10	Apakah website ini mudah dipahami bagi pengguna?	14	6	0	0
Total		159	40	1	0

Dari hasil perhitungan kuesioner tersebut dapat disimpulkan dengan persamaan 1, dengan menggunakan ketentuan skala sebagai berikut :

1. A (Sangat Baik) = 4
2. B (Baik) = 3
3. C (Cukup) = 2
4. D (Kurang) = 1

Dimana skala nilai maksimal adalah 4 yaitu sangat baik.

Berikut merupakan persamaan (1).

$$\text{Hasil Pengujian} = \sum_{i=1}^{i=4} H_i \times 100\% \quad (1)$$

Dimana H_i didapatkan dari persamaan (2) berikut.

$$H_i = \frac{\sum_{j=1}^n X_j}{Np} \quad (2)$$

Keterangan:

j = Index soal ke j.

X_j = Nilai atau skor pertanyaan ke j untuk index i.

n = Maksimal soal = 10

N_p = Total Poin = 800, didapatkan dari persamaan (3).

Untuk mencari nilai N_p dapat dilihat pada persamaan (3) dibawah ini.

$$N_p = n \times 4 \text{ (skala maksimal)} \times \text{jumlah responden.} \quad (3)$$

$$N_p = 10 \times 4 \times 20 = 800$$

Dengan menggunakan persamaan (1), (2), dan (3) maka akan dapat dihitung hasil pengujian sebagai berikut.

$$D \text{ (Kurang)} (0 \times \frac{1}{800}) \times 100\% = 0\%$$

$$C \text{ (Cukup)} (1 \times \frac{2}{800}) \times 100\% = 0,25\%$$

$$B \text{ (Baik)} (40 \times \frac{3}{800}) \times 100\% = 15\%$$

$$A \text{ (Sangat Baik)} (159 \times \frac{4}{800}) \times 100\% = 79,5\% \\ \text{Hasil} = 94,75\%$$

5. Kesimpulan Pengujian

Berdasarkan dari hasil pengujian pada Aplikasi Makah *Wedding Service* Berbasis *Website* sudah bisa berfungsi dengan baik. Akan tetapi ada salah satu hasil jawaban C untuk pertanyaan nomor 8 dengan pertanyaan Apakah *website* ini mudah digunakan?, kemudian penulis melakukan perbaikan dengan merubah tampilan *interface* agar pengguna lebih mudah untuk menggunakan *website* ini dan hasil pengujian pada website sudah sesuai dengan kebutuhan dan membantu permasalahan yang ada pada jasa Makah *Wedding Service*. Kemudian hasil dari presentase pengujian dengan menggunakan kuesioner yang telah dilakukan oleh 20 orang penguji terhadap *website* memperoleh hasil sebanyak 94,75%.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil dari pengumpulan data, analisa perancangan, implementasi beserta pengujian pada pembuatan Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel kemudian dapat disimpulkan sebagai berikut:

1. *Website* dapat memberikan kemudahan bagi masyarakat dalam mendapatkan informasi jasa yang disediakan pada Makah *Wedding Service* dilakukan dengan cara melakukan analisis kebutuhan sistem dengan proses observasi dan wawancara langsung dengan pemilik jasa Makah *Wedding Service*, perancangan *design* sistem dan implementasi *design* sistem. Pada tahapan perancangan *design* sistem terdiri dari pembuatan UML, *Flowchart*, ERD, Perancangan database dan perancangan *design interface*. Pada pembuatan Aplikasi Makah *Wedding Service* Berbasis *Website* Dengan *Framework* Laravel diimplementasikan menggunakan bahasa pemrograman PHP dengan *framework* Laravel *versi* 8 dan database MySQL.
2. Dapat memberikan kemudahan bagi *Client* untuk melakukan proses pemesanan jasa yang disediakan pada Makah *Wedding Service* yang terdapat pada fitur pemesanan di halaman *client*.
3. Dapat memberikan kemudahan bagi Admin dalam mengelola pemesanan, data pegawai, data *client*, data paket yang disediakan, data wilayah, data transportasi, data jadwal, data pengeluran, keuangan bulanan, landing page, dan data testimoni di website Makah *Wedding Service*.
4. Berdasarkan hasil dari kuesioner yang telah didapatkan dari 20 responden pada pembuatan Aplikasi Makah *Wedding Service* berbasis *website* Dengan *Framework* Laravel yang terdiri dari pemilik dengan pengawai Makah *Wedding Service* kemudian masyarakat luas didapatkan hasil presentase sebesar 94,75% sehingga bisa disimpulkan *website* ini sudah siap untuk digunakan.

5.2. Saran

Pada *website* ini masih belum sepenuhnya sempurna, oleh karena itu dipaparkan saran untuk pengembang selanjutnya adat dapat dijadikan pertimbangan, sehingga website dapat lebih baik dan lebih sempurna. Adapun saran-saran yang akan diberikan antara lain sebagai berikut:

1. Untuk selanjutnya bisa dikembangkan fitur laporan keuangan bulanan.
2. Diharapkan pengembang berikutnya agar bisa menambahkan fitur untuk *fitting* sebelum hari H.

DAFTAR PUSTAKA

- Ahaddin, Fachrul. (t.thn.). *Mengenal Use Case Diagram*. Diambil kembali dari
<https://www.dumetschool.com/blog/Mengenal-Use-Case-Diagram/>
- Anonim. (2021). *Getting Started*. Diambil kembali dari
<https://code.visualstudio.com/>
- Anonim. (2021). *PHP: Hypertext Preprocessor*. Diambil kembali dari
<https://www.php.net>
- Anonim. (2021). *The PHP Framework for Web Artisans*. Diambil kembali dari
<https://laravel.com/>
- Anonim. (2021). *UI Design Tool*. Diambil kembali dari
<https://www.figma.com/ui-design-tool/>
- Anonim. (2021). *XAMPP Apache + MariaDB + PHP + Perl*. Diambil kembali
dari <https://www.apachefriends.org/>
- Astuti, Novianti. (2017). Perancangan Sistem Informasi Pembelian Dan Penjualan
Pada Koperasi Ppdk Lapas Perempuan Klas Iia Tangerang. Skripsi S-1.
Tangerang: Sekolah Tinggi Manajemen Dan Ilmu Komputer STMIK
Raharja.
- Efendi, Ilham. (2018). *Pengertian dan Tipe Diagram UML*. Diambil kembali dari
<https://www.it-jurnal.com/pengertian-dan-tipe-diagram-uml/>
- Efmi Maiyana, M. S. (2019). Penyewaan Online Wedding Organizer Berbasis
Web. *JURNAL SAINS DAN INFORMATIKA* , 25.
- Idris, Mohamad Rivai. (2018). Pengembangan Sistem Informasi Pengelolaan Akta
Dan Laporan Pembayaran Pada Notaris Dan Ppat Siti Ubaidah S.H.
Skripsi S-1. Tangerang: Sekolah Tinggi Manajemen Dan Ilmu Komputer
STMIK Raharja.
- Inuy Khalimah. (2016). *Makah Wedding Service 2016*. Diambil kembali dari
<http://makahwedding.blogspot.com/2016/03/profil-makah-wedding-service.html?m=0>
- Irza Triwahyuni, Nevi Costari, M. Muzaki Wildan G. (2021). SISTEM
INFORMASI UMKM MARMER BERBASIS WEBSITE PADA DESA
GAMPING KECAMATAN CAMPURDARAT. *Artikel Sistem Informasi
Manajemen, Akuntansi* .
- Johan RZ. (2021). MAKALAH ALGORITMA DAN FLOWCHART.
ACADEMIA, 3-4.
- Malabay. (2016). Pemanfaatan Flowchart Untuk Kebutuhan Deskripsi Proses
Bisnis. *Pemanfaatan Flowchart untuk Kebutuhan Deskripsi Proses Bisnis*.
, Volume 12, 23.
- Muhammad Dedi Irawan, S. A. (2018). Implementasi E-Arsip Pada Program
Studi Teknik Informatika. *Jurnal Teknologi Informasi*, 78.

- Munjeri, Intan Juliani. (2017). Sistem Penunjang Keputusan Untuk Rekayasa Lalu Lintas Berbasis Optimasi Traffic Delay. Skripsi S-1. Jakarta: Universitas Islam Negeri Hidayatullah Jakarta.
- Nanang Wijayanto. (2021). *Mengubah Nasib, Bisnis Wedding Organizer Harus Beralih ke Digital*. Diambil kembali dari sindonews.com: <https://ekbis.sindonews.com/read/471904/34/mengubah-nasib-bisnis-wedding-organizer-harus-beralih-ke-digital-1625152025>
- Pratama, Aditya Rahmatullah. (2019). *Belajar UML - Sequence Diagram*. Diambil kembali dari <https://www.codepolitan.com/belajar-uml-sequence-diagram-57fdb1a5ba777-17044>
- ryal. (2017). *Entity Relationship Diagram (ERD)*. Diambil kembali dari steemit: <https://steemit.com/esteem/@ryal/entity-relationship-diagram-erd-ec3f63c1800ff>
- Setiadi, M. Fikri. (2017). *Kupas Tuntas Konsep Model Data Entity Relationship Diagram atau ERD*. Diambil kembali dari <http://mfikri.com/artikel/kupas-tuntas-konsep-model-data-entity-relationship-diagram-atau-erd.html>
- Siregar, V. M. (2018). SISTEM INFORMASI PENDATAAN LOGISTIK AKTIVA TETAP. *Jurnal SISTEMASI*, 252.
- Tabrani, Muhamad dan Insan Rezqy Aghniya. (2019). Implementasi Metode Waterfall Pada Program Simpan Pinjam Koperasi Subur Jaya Mandiri Subang. *Jurnal Interkom*, 14(1):44–53.
- Tristianto, C. (2018). PENGGUNAAN METODE WATERFALL UNTUK PENGEMBANGAN SISTEM MONITORING DAN EVALUASI PEMBANGUNAN PEDESAAN. *Jurnal Teknologi Informasi*, 13-14.
- Widoyoko, Eko Putro. (2015). Teknik Penyusunan Instrumen Penelitian. *Yogyakarta: Pustaka*.

LAMPIRAN

Lampiran 1. *Source Code*

Web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AdminDashboardController;
use App\Http\Controllers\AdminPegawaiController;
use App\Http\Controllers\AdminKonsumenController;
use App\Http\Controllers\AdminPesananController;
use App\Http\Controllers\AdminPaketController;
use App\Http\Controllers\AdminJadwalController;
use App\Http\Controllers\AdminTestimoniController;
use App\Http\Controllers\AdminPemasukanController;
use App\Http\Controllers\AdminPengeluaranController;
use App\Http\Controllers\AdminLandingController;
use App\Http\Controllers\AdminTransportasiController;
use App\Http\Controllers\AdminTransportasiHargaController;
use App\Http\Controllers\AdminKabupatenController;
use App\Http\Controllers\AdminTransaksiController;
use App\Http\Controllers\AdminKeuanganController;
use App\Http\Controllers\KonsumenJadwalController;
use App\Http\Controllers\KonsumenPaketController;
use App\Http\Controllers\KonsumenPesanController;
use App\Http\Controllers\KonsumenProfileController;
use App\Http\Controllers\KonsumenTestimoniController;
use App\Http\Controllers\KonsumenTransaksiController;
use App\Http\Controllers\PegawaiDashboardController;
use App\Http\Controllers\PegawaiPesananController;
use App\Http\Controllers\PegawaiJadwalController;
use App\Http\Controllers\login;
use App\Http\Controllers\RegistrasiController;
use App\Http\Controllers\LandingPageController;
use App\Http\Controllers\MailController;

/*
|--------------------------------------------------------------------------
|
| Web Routes
|
| Here is where you can register web routes for your
| application. These
| routes are loaded by the RouteServiceProvider within a group
| which
| contains the "web" middleware group. Now create something
| great!
|
*/
//ADMIN
Route::group(['prefix'=> 'admin', 'middleware'=> 'auth:admin'],
function()
{
```

```

Route::resource('dashboard', AdminDashboardController::class);
Route::resource('pegawai', AdminPegawaiController::class);
Route::resource('konsumen', AdminKonsumenController::class);
Route::resource('pesanan', AdminPesananController::class);
Route::resource('keuangan', AdminKeuanganController::class);

Route::get('detail/{id}', [AdminPesananController::class, 'show']);
;
Route::resource('paket', AdminPaketController::class);
Route::resource('jadwal', AdminJadwalController::class);
Route::resource('testimoni', AdminTestimoniController::class);
Route::resource('pemasukan', AdminPemasukanController::class);
Route::resource('pengeluaran', AdminPengeluaranController::class);
Route::resource('landing', AdminLandingController::class);
Route::resource('transportasi', AdminTransportasiController::class);
Route::resource('transportasi_harga', AdminTransportasiHargaController::class);
Route::resource('kabupaten', AdminKabupatenController::class);
Route::resource('transaksi', AdminTransaksiController::class);
Route::post('kecamatan', [AdminTransportasiController::class, 'storeKecamatan'])->name('kecamatan.store');
Route::delete('hapus_kecamatan/{id}', [AdminTransportasiController::class, 'destroyKecamatan'])->name('kecamatan.destroy');
});
//END ADMIN

//PEGAWAI
Route::group(['prefix'=> 'pegawai', 'middleware'=> 'auth:pegawai'], function()
{
Route::resource('dashboard-pegawai', PegawaiDashboardController::class);
Route::resource('pesanan-pegawai', PegawaiPesananController::class);
Route::resource('jadwal-pegawai', PegawaiJadwalController::class);

});
//END PEGAWAI

//KONSUMEN
Route::group(['prefix'=> 'konsumen', 'middleware'=> 'auth:konsumen'], function()
{
Route::resource('profile-konsumen', KonsumenProfileController::class);
Route::resource('paket-konsumen', KonsumenPaketController::class);
Route::resource('pesan-konsumen', KonsumenPesanController::class);
Route::resource('transaksi-konsumen', KonsumenTransaksiController::class);
Route::resource('jadwal-konsumen', KonsumenJadwalController::class);
Route::resource('testimoni-konsumen', KonsumenTestimoniController::class);
}
)

```

```

    });

//END KONSUMEN

//UMUM
Route::resource('/', LandingPageController::class);
Route::resource('register', RegistrasiController::class);
Route::get('login', function () {
    return view('login');
})->middleware('guest');
Route::get('admin-detail-pesanan', function () {
    return view('admin.admin-detail-pesanan');
});
Route::post('/kirimdata', [login::class, 'login'])->name('login');
Route::get('/keluar', [login::class, 'logout']);
Route::get('kirim-email', [MailController::class, 'index']);
Route::get('/validasi/{id_konsumen}', [RegistrasiController::class, 'ubahStatus']);
Route::get('/cetak/{id}', [KonsumenTransaksiController::class, 'cetak'])->name('cetak');
Route::get('/cetakAdmin/{id}', [AdminTransaksiController::class, 'cetakAdmin'])->name('cetakAdmin');
Route::post('/getkecamatan', [AdminPesananController::class, 'getKecamatan'])->name('getkecamatan');

//END UMUM

```

login.php (Controller)

```

<?php

namespace App\Http\Controllers;
use App\Models\Admin;
use Illuminate\Support\Facades\Auth;
use App\Models\Pegawai;
use App\Models\Konsumen;

use Illuminate\Http\Request;

class login extends Controller
{
    function login(Request $kiriman) {
        $dataadmin= Admin::where ('username', $kiriman->username) ->where('password', $kiriman->password)->get();
        $datapegawai= Pegawai::where ('username', $kiriman->username) ->where('password', $kiriman->password)->get();
        $datakonsumen= Konsumen::where ('email', $kiriman->username) ->where('password', $kiriman->password)->where('email_verify', 'sudah verifikasi')->get();

        if (count ($dataadmin)>0){
            Auth::guard('admin')->LoginUsingId($dataadmin[0] ['username']);
            return redirect('/admin/dashboard');

        }else if (count ($datapegawai)>0){


```

```

        Auth::guard('pegawai')-
>LoginUsingId($datapegawai[0]['id_pegawai']);
        return redirect('/pegawai/dashboard-pegawai');

    }else if (count ($datakonsumen)>0) {
        Auth::guard('konsumen')-
>LoginUsingId($datakonsumen[0]['id_konsumen']);
        return redirect('/konsumen/profile-konsumen');

    }else{
        return redirect('/login');
    }
}

function logout(){
    if(Auth::guard('admin')->check()){

        Auth::guard('admin')->logout();

    }else if(Auth::guard('pegawai')->check()){

        Auth::guard('pegawai')->logout();

    }else if (Auth::guard('konsumen')->check()){

        Auth::guard('konsumen')->logout();
    }
    return redirect('/');
}
}

```

RegistrasiController.php (Controller)

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Konsumen;

class RegistrasiController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return view('register');
    }
    public function store(Request $request)
    {
        $data = array(
            'email'=>$request->email,
            'nama'=>$request->nama,
            'password'=>$request->password,
        )
    }
}

```

```

        'no_hp'=>$request->no_hp,
        'alamat'=>$request->alamat,
        'email_verify'=>"belum verifikasi"
    );
    Konsumen::create($data);
    $data_konsumen=Konsumen::where('email',$request->email)->first();

        //---irim email
    $konsumen = Konsumen::find($data_konsumen->id_konsumen);
    $id_konsumen = $data_konsumen->id_konsumen;
    $pass = $request->password;
    //$jasa = jasaModel::find($tenagaKerja->id_jasa);
    $link =
    '1803006.web.ti.polindra.ac.id/validasi/'.$id_konsumen;
    $details = [
        'title' => 'Pemberitahuan validasi Registrasi Client
Makah Wedding Service.',
        'body' => 'Silahkan tekan link,',
        'link' => $link,
        'ket' => ' untuk validasi akun client.',
    ];

    \Mail::to($request->email)->send(new
\App\Mail\MyTestMail($details));
    //Kirim Email

    return redirect('login');
}

public function ubahStatus($id_konsumen)
{
    $konsumen = Konsumen::find($id_konsumen);
    if ($konsumen->email_verify == "belum verifikasi") {
        $data = array(
            'email_verify'=>"sudah verifikasi",
        );
        Konsumen::whereid_konsumen($id_konsumen)->update($data);
        return redirect('login')->with('success',
$konsumen->email);

    }else{
        return redirect()->back()->with('alert','Maaf
tidak bisa diakses');
    }
}
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**

```

```

 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    //
}
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    //
}
/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    //
}
}

```

AdminPesananController.php (Controller)

```

<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Pesanan;
use App\Models\Paket;
use App\Models\Konsumen;
use App\Models\Pegawai;
use App\Models\Transportasi;
use App\Models\Transaksi;
use App\Models\Kabupaten;
use App\Models\Kecamatan;

class AdminPesananController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $pesanan =
Pesanan::join('transportasis','transportasis.id_transportasi',

```

```

'pesanans.id_transportasi')
//with('transportasi.kabupaten','transportasi.kecamatan','paket'
,'pegawai',
'konsumen')->get();
    ->get();
$paket = Paket::all();
$konsumen = Konsumen::all();
$pegawai = Pegawai::all();
$transportasi = Transportasi::all();
$transaksi = Transaksi::all();
$kabupaten = Kabupaten::all();
$kecamatan = Kecamatan::all();
return view('Admin.data-
pesanans',compact('pesanans','paket','konsumen','pegawai','transpo
rtasi','transaksi','kabupaten','kecamatan'))->with('i');
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $cek= Pesanan::where('tgl',$request->tgl)->count();
    if($cek<3)
    {
        $transport =Transportasi::where('id_kabupaten',$request-
>id_kabupaten_pesanan)->where('id_kecamatan',$request-
>kecamatan)->first();
        // dd($transport);
        $data = array(
            'id_konsumen'=>$request->id_konsumen,
            'id_paket'=>$request->id_paket,
            'id_pegawai'=>$request->id_pegawai,
            'id_transportasi'=>$transport->id_transportasi,
            'alamat'=>$request->alamat,
            'status'=>$request->status,
            'tgl'=>$request->tgl
        );
        Pesanan::create($data);
        return redirect('admin\pesanans')->with('success','Data
Pesanan berhasil ditambah');
    } else{
        return redirect('admin\pesanans')->with('Error','Data
Pesanan gagal ditambah karena tanggal telah full booking');
    }
}

public function show($id)
{
    $data1= Pesanan::find($id);
    $data2= Transaksi::all();
    return view('Admin.admin-detail-
pesanans',compact('data1','data2'))->with('i');

}
/**
 * Display the specified resource.

```

```

/*
 * @param int $id
 * @return \Illuminate\Http\Response
 */
// untuk edit
public function update(Request $request, $id)
{
    $cek= Pesanan::where('tgl',$request->tgl)->where('id_pesanan','!=',$id)->count();

    if($cek<3)
    {

        $transport =
Transportasi::where('id_kabupaten',$request->id_kabupaten_pesanan)->where('id_kecamatan',$request->kecamatan)->first();
        // dd($request->kecamatan);
        $pesanan = Pesanan::find($id);

        $pesanan->id_konsumen = $request->id_konsumen;
        $pesanan->id_paket= $request->id_paket;
        $pesanan->id_pegawai= $request->id_pegawai;
        $pesanan->id_transportasi= $transport->id_transportasi;
        $pesanan->alamat= $request->alamat;

        if($request->status != null)
            $pesanan->status = $request->status;
        $pesanan->tgl= $request->tgl;

        $pesanan->save();

        return redirect('admin\pesan');
    } else{
        return redirect('admin\pesan')->with('Error','Data Pesanan gagal diedit karena tanggal telah full booking');
    }
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    try{
        $datas = Pesanan::findOrFail($id);
        $datas->delete();
        return redirect('admin\pesan')->with('success','pesan Berhasil Dihapus');
    }catch(\Throwable $th){
        return redirect('admin\pesan')->withErrors('Data gagal dihapus. Harap hapus data yang terkait');
    }
}

public function getKecamatan(Request $request)

```

```

    {
        $kec = Kecamatan::with('transportasi')-
>where('id_kabupaten',$request->id_kabupaten)->get();
        return response()->json($kec);
    }
}

```

PegawaiPesananController.php (Controller)

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Pesanan;
use App\Models\Paket;
use App\Models\Konsumen;
use App\Models\Transportasi;
use App\Models\Kabupaten;
use App\Models\Kecamatan;
use DB;

class PegawaiPesananController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $paket = Paket::all();
        $konsumen = Konsumen::all();
        $pesanan =
Pesanan::with('transportasi.kabupaten','transportasi.kecamatan',
'paket','konsumen')->get();
        $transportasi = Transportasi::all();
        $kabupaten = Kabupaten::all();
        $kecamatan = Kecamatan::all();

        return view('Pegawai.data-
pesanan',compact('pesanan','paket','konsumen','transportasi','ka
bupaten','kecamatan'))->with('i');
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $cek= Pesanan::where('tgl',$request->tgl)->count();
        if($cek<3)
        {

```

```

$transport =
Transportasi::where('id_kabupaten', $request-
>id_kabupaten_pesanan) -> where('id_kecamatan', $request-
>kecamatan) -> first();

$data = array(
    'id_pegawai' => auth() -> user() -> id_pegawai,
    'id_konsumen' => $request -> id_konsumen,
    'id_paket' => $request -> id_paket,
    'id_transportasi' => $transport -> id_transportasi,
    'alamat' => $request -> alamat,
    'status' => 'Telah Melakukan Pemesanan',
    'tgl' => $request -> tgl
);
Pesanan::create($data);
return redirect('pegawai\pesanant-pegawai')-
>with('success', 'Data Pegawai berhasil ditambah');
} else{
    return redirect('pegawai\pesanant-pegawai')-
>with('Error', 'Data Pesanan gagal ditambah karena tanggal telah
full booking');
}
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
// untuk edit
public function update(Request $request, $id)
{
    $cek= Pesanan::where('tgl', $request->tgl)-
>where('id_pesanan', '!=', $id) -> count();
    if($cek<3)
    {

        $transport =
Transportasi::where('id_kabupaten', $request->id_kabupaten) -
>where('id_kecamatan', $request->kecamatan) -> first();

        $data = array(
            'id_pegawai' => auth() -> user() -> id_pegawai,
            'id_konsumen' => $request -> id_konsumen,
            'id_paket' => $request -> id_paket,
            'id_transportasi' => $transport -> id_transportasi,
            'alamat' => $request -> alamat,
            'tgl' => $request -> tgl
        );
        Pesanan::whereid_pesanan($id) -> update($data);

        return redirect('pegawai\pesanant-pegawai');
    } else{
        return redirect('pegawai\pesanant-pegawai')-
>with('Error', 'Data Pesanan gagal diedit karena tanggal telah
full booking');
    }
}

```

```

    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        try{
            $datas = Pesanan::findOrFail($id);
            $datas->delete();
            return redirect('pegawai\pesan-an-pegawai')-
>with('success','pesan-an Berhasil Dihapus');
        }catch(\Throwable $th){
            return redirect('pegawai\pesan-an-pegawai')-
>withErrors('Data gagal dihapus. Harap hapus data yang
terkait');
        }
    }

    public function getKecamatan(Request $request)
    {
        $kec = Kecamatan::where('id_kabupaten',$request-
>id_kabupaten)->get();
        return response()->json($kec);
    }
}

```

KonsumenPesananController.php (Controller)

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Pesanan;
use App\Models\Paket;
use App\Models\Transportasi;
use App\Models\Transaksi;
use App\Models\Kabupaten;
use App\Models\Kecamatan;

class KonsumenPesananController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    // untuk menampilkan
    public function index()
    {

```

```

$pesan = 
Pesanan::with('transportasi.kecamatan','transportasi.kabupaten')
->get();
// dd($pesan);
$paket = Paket::all();
$transportasi = Transportasi::all();
$transaksi = Transaksi::all();
$kabupaten = Kabupaten::all();
$kecamatan = Kecamatan::all();

return view('Konsumen.pesan-
paket',compact('kecamatan','pesan','paket','transportasi','tra-
nsaksi','kabupaten'))->with('i');
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $cek= Pesanan::where('tgl',$request->tgl)->count();
    if($cek<3)
    {

        $transport =
Transportasi::where('id_kabupaten',$request-
>id_kabupaten_pesanan)->where('id_kecamatan',$request-
>kecamatan)->first();

        $data = array(
            'id_konsumen'=>auth()->user()->id_konsumen,
            'id_pegawai'=>1,
            'id_paket'=>$request->id_paket,
            'id_kabupaten'=>$request->id_kabupaten,
            'id_transportasi'=>$transport->id_transportasi,
            'alamat'=>$request->alamat,
            'status'=>'Telah Melakukan Pemesanan',
            'tgl'=>$request->tgl
        );
        Pesanan::create($data);
        return redirect('konsumen\pesan-konsumen')-
>with('success','Data Pesan paket berhasil ditambah');
    } else{
        return redirect('konsumen\pesan-konsumen')-
>with('Error','Data Pesanan gagal ditambah karena tanggal telah
full booking');
    }
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
// untuk edit

```

```

public function update(Request $request, $id)
{
    $cek= Pesanan::where('tgl',$request->tgl)-
>where('id_pesanan','!=',$id)->count();

    if($cek<3)
    {

        $transport =
Transportasi::where('id_kabupaten',$request->id_kabupaten)-
>where('id_kecamatan',$request->kecamatan)->first();

        $data = [
            'id_paket'=>$request->id_paket,
            'tgl'=>$request->tgl,
            'alamat'=>$request->alamat,
            'id_transportasi'=>$transport->id_transportasi
        ];
    }

    Pesanan::whereid_pesanan($id)->update($data);

    return redirect('konsumen\pesan-konsumen');
} else{
    return redirect('konsumen\pesan-konsumen')-
>with('Error','Data Pesanan gagal diedit karena tanggal telah
full booking');
}
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
// untuk hapus
public function destroy($id)
{
    try{
        $datas = Pesanan::findOrfail($id);
        $datas->delete();
        return redirect('konsumen\pesan-konsumen')-
>with('success','pesan paket Berhasil Dihapus');
    }catch(\Throwable $th){
        return redirect('konsumen\pesan-konsumen')-
>withErrors('Data gagal dihapus. Harap hapus data yang
terkait');
    }
}
public function getKecamatan(Request $request)
{
    $kec = Kecamatan::where('id_kabupaten',$request-
>id_kabupaten)->get();
    return response()->json($kec);
}
}

```

LandingPageController.php (Controller)

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Pegawai;
use App\Models\Paket;
use App\Models\Testimoni;
use DB;

class LandingPageController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $background=DB::table('landings')->where('bagian','=', 'Background')->get();
        $dokumentasi=DB::table('landings')->where('bagian','=', 'Dokumentasi')->get();
        $tentang_kami=DB::table('landings')->where('bagian','=', 'Tentang Kami')->get();
        $services=DB::table('landings')->where('bagian','=', 'Services')->get();
        $qna=DB::table('landings')->where('bagian','=', 'QnA')->get();
        $contact=DB::table('landings')->where('bagian','=', 'Contact')->get();
        $pegawai = Pegawai::all();
        $testimoni = Testimoni::all();
        $paket = Paket::all();
        return view('landing-page',compact('background','dokumentasi','tentang_kami','services','qna','contact','pegawai','testimoni','paket'))->with('i');

    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request

```

```
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    //
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    //
}

}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    //
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    //
}

}
```

Pesanan.php (Models)

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Pesanan extends Model
{
    protected $table='pesanans';
    protected $primaryKey = 'id_pesanan';
    protected $fillable =
['id_pesanan','id_konsumen','id_paket','id_pegawai','id_transportasi','tgl','alamat','status']; //field tabel
    public $timestamps = false;

    public function paket()
    {
        return $this->belongsTo('App\Models\Paket','id_paket');
    }

    public function konsumen()
    {
        return $this-
>belongsTo('App\Models\Konsumen','id_konsumen');
    }

    public function pegawai()
    {
        return $this-
>belongsTo('App\Models\Pegawai','id_pegawai');
    }

    public function transportasi()
    {
        return $this-
>belongsTo('App\Models\Transportasi','id_transportasi');
    }
}
```

Lampiran 2. Biodata Penulis

BIODATA PENULIS TUGAS AKHIR

Nama	:	Camelia		
Nomor Induk Mahasiswa (NIM)	:	1803006		
Jenis Kelamin	:	Perempuan		
Tempat & Tanggal Lahir	:	Indramayu, 23 Desember 2000		
Status	:	Mahasiswa		
Agama	:	Islam		
Asal Ijazah Sekolah	:	Nama Sekolah	Kota Sekolah	Tahun Ijazah
SD	:	Mi Al Mujahidin	Indramayu	2012
SLTP	:	Mts Al Mujahidin	Indramayu	2015
SLTA	:	SMKN 1 Balongan	Indramayu	2018
Program Studi / Jurusan	:	Teknik Informatika		
Alamat Rumah Asal	:	Blok Bacok Desa Segeran Lor Kecamatan Juntinyuat Kabupaten Indramayu Provinsi Jawa Barat.		
No. Telpon / HP	:	083822765867		
Email	:	camellacamel231@gmail.com		
Nama Orang Tua	:	Darus		
Pekerjaan Orang Tua	:	Wiraswasta		
Alamat Orang Tua	:	Blok Bacok Desa Segeran Lor Kecamatan Juntinyuat Kabupaten Indramayu Provinsi Jawa Barat.		
Judul Tugas Akhir	:	Aplikasi Makah <i>Wedding Service</i> Berbasis Website Dengan Framework Laravel		
Dosen Pembimbing	:	<ol style="list-style-type: none">1. Muhamad Mustamiin, S.Pd., M.Kom2. Alifia Puspaningrum, S.Pd., M.Kom		



Indramayu, 10 September 2021

Penulis,

Camelia

NIM 1803006