

**PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN TRANSAKSI
BERBASIS *WEB* PADA MUTIARA STUDIO PHOTO PADANG**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Program Strata-I
Pada Departemen Sistem Informasi Fakultas Teknologi Informasi Universitas
Andalas

Oleh:

Muhammad Al Hamid Afiza

1511522003

Pembimbing:

Ricky Akbar, M.Kom.

198410062012121001



**DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2023**

HALAMAN PENGESAHAN

**PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN TRANSAKSI
BERBASIS *WEB* PADA MUTIARA STUDIO FOTO PADANG**

Oleh

Muhammad Al Hamid Afiza
UNIVERSITAS ANDALAS
1511522003



PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa laporan Tugas Akhir yang berjudul **“Pembangunan Sistem Informasi Pengelolaan Transaksi Berbasis Web pada Mutiara Studio Foto Padang”** belum pernah diajukan sebagai salah satu syarat menyelesaikan mata kuliah Tugas Akhir di suatu Perguruan Tinggi dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat identik yang pernah ditulis atau dipublikasikan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini dan sebutkan dalam daftar pustaka.



Padang, Januari 2023

Muhammad Al Hamid Afiza

NIM. 1511522003

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan proposal tugas akhir ini. Proposal ini disusun berdasarkan penelitian yang penulis lakukan pada Mutiara Studio Photo dengan judul “**Pembangunan Sistem Informasi Pengelolaan Studio Foto Berbasis *Web* Pada Mutiara Studio Photo Padang**”.

Proposal tugas akhir ini disusun dengan tujuan untuk memenuhi persyaratan akademis dalam menyelesaikan mata kuliah Seminar Proposal di Jurusan Sistem Informasi Fakultas Teknologi Informasi Universitas Andalas. Untuk itu, izinkan penulis menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orangtua serta saudara-saudara saya yang selalu memberikan doa dan dukungan yang tiada habisnya,
2. Bapak Husnil Kamil, M.T., selaku Ketua Departemen Sistem Informasi, Fakultas Teknologi Informasi Universitas Andalas,
3. Bapak Ricky Akbar, M.Kom. selaku pembimbing Tugas Akhir,

Penulis menyadari bahwa dalam penyusunan proposal tugas akhir ini masih terdapat banyak kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari pembaca melalui *email* hamidafiza15@gmail.com agar penelitian selanjutnya dapat dilakukan lebih baik lagi.

Padang, Januari 2023

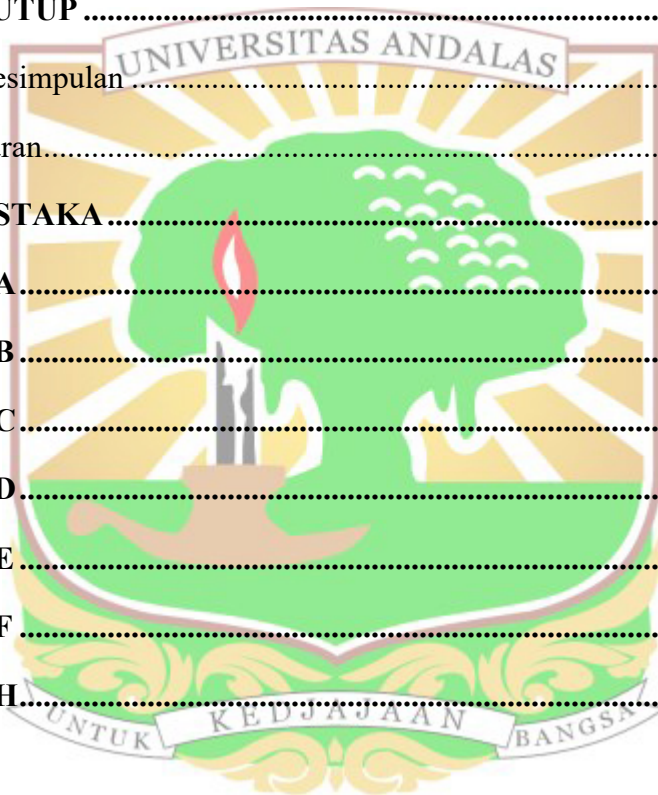
Penulis,

Muhammad Al Hamid Afiza

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
PERNYATAAN.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	vi
DAFTAR GAMBAR.....	vii
ABSTRAK	ix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Fotografi.....	5
2.2 Sistem Informasi	6
2.3 Perangkat Lunak Pendukung.....	6
2.4 Alat Pemodelan Aplikasi.....	9
2.5 Referensi Penelitian	14
BAB III METODOLOGI PENELITIAN	19
3.1 Objek Penelitian	19
3.2 Metode Pengumpulan Data	19
3.3 Metode Pengembangan Sistem	20
3.4 <i>Flowchart</i> Penelitian	21

BAB IV ANALISIS DAN PERANCANGAN SISTEM	25
4.1 Analisis Sistem.....	25
4.2 Perancangan Sistem	46
BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM.....	53
5.1 Implementasi Sistem	53
5.2 Pengujian Sistem.....	61
5.3 Kesimpulan Hasil Pengujian.....	71
BAB VI PENUTUP	73
6.1 Kesimpulan	73
6.2 Saran.....	73
DAFTAR PUSTAKA	74
LAMPIRAN A	76
LAMPIRAN B	88
LAMPIRAN C	99
LAMPIRAN D	108
LAMPIRAN E	111
LAMPIRAN F	121
LAMPIRAN H	156



DAFTAR TABEL

Tabel 2.1 Simbol usecase (Artina, 2006).....	12
Tabel 2.2 Referensi Penelitian	15
Tabel 4.1 Deskripsi Tugas Aktor	38
Tabel 4.2 Use scenario melihat galeri foto.....	40
Tabel 4.3 Use case scenario melihat daftar paket foto.....	40
Tabel 4.4 Use case scenario booking paket foto	41
Tabel 4.5 Tabel packages	48
Tabel 4.6 Tabel members	49
Tabel 4.7 Tabel booking	49
Tabel 5.1 Fokus Pengujian.....	62
Tabel 5.2 Mengubah Data Pasien (Benar)	63
Tabel 5.3 Mengubah Data Pasien (Benar)	65
Tabel 5.4 Mengubah galery (Benar)	66
Tabel 5.5 Kasus Pengujian Data Nama Paket (Salah)	68
Tabel 5.6 Pengujian Booking Studio Musik (Benar)	69
Tabel 5.7 Kasus Pengujian Tanggal Bulan Kosong (Salah)	70
Tabel 5.8 Hasil Pengujian	72



DAFTAR GAMBAR

Gambar 2.1 Simbol Event (Dewi, Indahyanti, & S, 2012)	9
Gambar 2.2 Simbol Activity dan Gateway (Dewi, Indahyanti, & S, 2012)	10
Gambar 2.3 Tipe-Tipe Connecting Object (Dewi, Indahyanti, & S, 2012)	10
Gambar 2.4 Simbol Pool dan Lane dalam Swimlane (Dewi, Indahyanti, & S, 2012)	11
Gambar 2.5 Use Case (Artina, 2006)	12
Gambar 3.1 Waterfall Model (Pressman, 2010)	20
Gambar 3.2 Flowchart Penelitian	22
Gambar 4.1 BPMN Proses booking yang sedang berjalan	26
Gambar 4.1 BPMN Proses <i>booking</i> yang sedang berjalan	26
Gambar 4.2 BPMN Proses Cetak Foto yang Sedang Berjalan	28
Gambar 4.2 BPMN Proses Cetak Foto yang Sedang Berjalan	28
Gambar 4.3 BPMN Proses Laporan Penjualan yang Sedang Berjalan	30
Gambar 4.3 BPMN Proses Laporan Penjualan yang Sedang Berjalan	30
Gambar 4.4 BPMN proses booking yang diusulkan	32
Gambar 4.4 BPMN proses booking yang diusulkan	32
Gambar 4.5 BPMN cetak foto yang diusulkan	34
Gambar 4.5 BPMN cetak foto yang diusulkan	34
Gambar 4.6 BPMN cetak laporan penjualan yang diusulkan	36
Gambar 4.7 Use Case Diagram	38
Gambar 4.8 Sequence diagram melihat galeri foto	42
Gambar 4.9 Sequence diagram melihat daftar paket	43
Gambar 4.10 Sequence diagram booking studio foto	43
Gambar 4.11 <i>Class analysis</i> pada melihat galeri foto	44
Gambar 4.12 <i>Class analysis</i> melihat daftar paket	44
Gambar 4.13 <i>Class analysis</i> booking	45
Gambar 4.14 <i>Class analysis</i> booking	46
Gambar 4.15 Arsitektur teknologi aplikasi	47
Gambar 4.16 Perancangan basis data	48
Gambar 4.17 Antarmuka halaman login	50

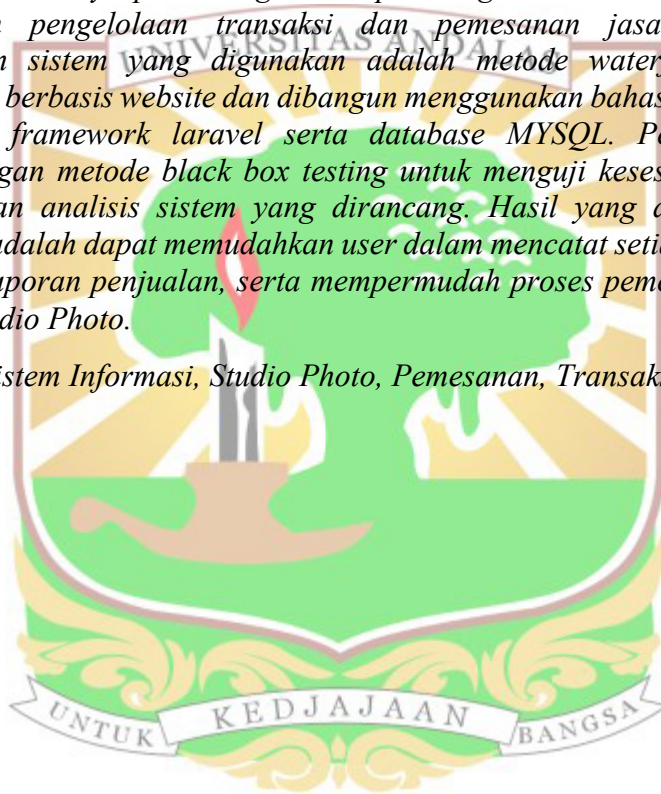
Gambar 4.18 Antarmuka halaman daftar paket	51
Gambar 4.19 Antarmuka halaman booking	52
Gambar 5.1 Kode program <i>routing</i> aplikasi	56
Gambar 5.2 Kode program <i>GaleryController</i>	57
Gambar 5.3 Kode program <i>model</i>	58
Gambar 5.4 Kode Program <i>View</i> Transaksi.....	59
Gambar 5.5 Implementasi tampilan halaman login	60
Gambar 5.6 Implementasi tampilan halaman admin.....	60
Gambar 5.7 Membuka Daftar Paket.....	64
Gambar 5.8 Mengubah Data Paket.....	64
Gambar 5.9 Data Paket Telah Berubah (Benar).....	64
Gambar 5.10 Membuka Data Galeri	67
Gambar 5.11 Memilih Salah Satu Data Galeri.....	67
Gambar 5.12 Data Galeri Telah Berubah.....	67
Gambar 5.13 Memilih Waktu booking	69
Gambar 5.14 Paket Berhasil di Booking.....	70
Gambar 5.15 Menampilkan Detail Booking	70
Gambar 5.16 Halaman Mengubah Detail Package	71
Gambar 5.17 Halaman Field Warning Jika Belum Diisi	71



ABSTRAK

Teknologi informasi sudah menjadi sarana utama dalam membantu manusia melakukan aktivitas sehari-hari termasuk usaha yang bergerak dibidang fotografi. Salah satunya adalah Mutiara Studio Photo. Saat ini proses pemesanan dan pengelolaan di Mutiara Studio Photo masih menggunakan cara konvensional. Semua pemesanan dicatat dalam sebuah buku secara manual sehingga membutuhkan waktu yang lama untuk rekapan transaksi dan menyebabkan besarnya peluang terjadi kesalahan pencatatan transaksi atau hilangnya bukti fisik transaksi. Cara pemesanan pun dilakukan dengan cara datang langsung ke studio foto atau pesan secara online melalui whatsapp bussines. Cara ini juga kurang efektif untuk merekap pesanan yang masuk karena masih dicatat secara manual dan belum ada sistem yang dapat menyimpan informasinya secara terstruktur. Untuk itu perlu adanya perancangan dan pembangunan sistem informasi untuk mempermudah pengelolaan transaksi dan pemesanan jasa foto. Metode pengembangan sistem yang digunakan adalah metode waterfall. Sistem ini dikembangkan berbasis website dan dibangun menggunakan bahasa pemrograman PHP dengan framework laravel serta database MYSQL. Pengujian sistem dilakukan dengan metode black box testing untuk menguji kesesuaian fungsional aplikasi dengan analisis sistem yang dirancang. Hasil yang diharapkan dari penelitian ini adalah dapat memudahkan user dalam mencatat setiap transaksi dan pengelolaan laporan penjualan, serta mempermudah proses pemesanan jasa foto di Mutiara Studio Photo.

Kata Kunci: Sistem Informasi, Studio Photo, Pemesanan, Transaksi.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi sudah menjadi sarana utama dalam membantu manusia melakukan aktivitas sehari-hari. Salah satu bentuk teknologi informasi yaitu adanya sebuah sistem yang sudah terkomputerisasi. Sistem ini dimanfaatkan demi terjaminnya kerja yang lebih efektif dan efisien. Sistem komputerisasi sudah banyak diterapkan diberbagai bidang, salah satunya di bidang fotografi dan videografi.

Fotografi merupakan sebuah media komunikasi, yakni media yang bisa digunakan untuk menyampaikan pesan atau ide kepada orang lain. Fotografi juga berguna untuk mendokumentasikan suatu momen atau peristiwa penting (Sudarma, 2014). Orang yang mengambil atau mendokumentasikan sebuah momen disebut fotografer. Fotografer merupakan salah satu jasa yang dibutuhkan banyak orang. Terutama dalam mengabadikan momen-momen dan acara penting seperti foto acara pernikahan, acara keluarga, acara wisuda, atau foto sebuah organisasi. Oleh sebab itu banyak fotografer atau orang-orang yang memiliki ketertarikan di dunia fotografi membangun sebuah studio foto untuk menghemat tempat dan memudahkan para pelanggannya mencari dan menggunakan jasa mereka. Salah satunya adalah Mutiara Studio Photo. Mutiara Studio Photo merupakan studio foto yang beralamat di Jalan Durian Tarung no.11, Kecamatan Kuranji. Sama seperti studio photo pada umumnya, Mutiara Studio Photo juga menyediakan layanan jasa foto, cetak foto, dan pemesanan bingkai. Jasa yang paling sering dipakai oleh pelanggan adalah jasa foto untuk foto wisuda, pre-wedding, post-wedding, foto keluarga, dan acara lainnya.

Saat ini proses pemesanan dan pengelolaan di Mutiara Studio Photo masih menggunakan cara konvensional. Semua pemesanan dicatat dalam sebuah buku secara manual sehingga membutuhkan waktu yang lama untuk rekapan transaksi yang pernah dilakukan. Hal ini menyebabkan terlambatnya pelaporan dan besarnya peluang terjadi kesalahan pencatatan transaksi atau hilangnya bukti fisik transaksi

seperti nota pembayaran. Cara pemesanan pun dilakukan dengan cara datang langsung ke studio foto atau pesan secara online melalui aplikasi chatting. Cara ini juga kurang efektif untuk merekap pesanan yang masuk karena masih dicatat secara manual dan belum ada system yang dapat menyimpan informasinya secara terstruktur. Untuk itu perlu adanya perancangan dan membangun sistem informasi untuk mempermudah dalam pengelolaan transaksi dan pemesanan jasa foto. Selain itu dengan adanya sistem informasi berbasis web diharapkan dapat menampilkan informasi lengkap terkait fasilitas dan paket foto yang dibutuhkan pelanggan sebelum melakukan pemesanan. Hal ini juga akan mempermudah pemasaran.

Sebagai referensi untuk memberikan wawasan pengetahuan mengenai aplikasi ini penulis merujuk kepada beberapa penelitian lain yang sudah pernah dilakukan sebelumnya dengan topik yang berkaitan dengan penelitian ini. Penelitian sebelumnya dilakukan oleh Arman pada tahun 2019 yang berjudul *Perancangan Antar Muka Aplikasi Pemesanan Fotografi Make Up Berbasis Online Pada Queen Photo Studio Padang*. Tujuan penelitian tersebut adalah untuk memudahkan pihak Queen Photo Studio dan Pelanggan. Hasil dari penelitian ini yaitu bagaimana sebuah sistem manual yang telah bertransformasi menjadi sebuah sistem online yang terkomputerisasi dan dengan adanya pemesanan online ini pihak Queen photo studio sudah tidak kesulitan lagi untuk merekap data pelanggan karena sudah menggunakan aplikasi yang terkomputerisasi (Arman,2019). Selanjutnya penelitian yang dilakukan oleh Eko Budi Mulyono pada tahun 2019 dengan judul *Sistem Informasi Jasa Fotografi Berbasis Web Pada Green Photography Jambi*. Penelitian ini bertujuan untuk merancang sebuah sistem informasi berbasis web yang mudah digunakan untuk mengelola berbagai data jasa yang ada(Eko Budi Mulyono, 2019). Kemudian penelitian lain nya dilakukan Afriyonna tahun 2014 dengan judul *Perancangan Sistem Informasi Administrasi Jasa Foto Pernikahan Berbasis Web Pada Euphoria Photo Studio*. Penelitian ini bertujuan untuk merancang website Euphoria Photo Studio, dan merancang sebuah website yang dapat memberikan kemudahan bagi pelanggan dalam memperkirakan harga paket foto pernikahan dan dapat memberikan kemudahan bagi admin dalam mengolah data(Afriyonna, 2014).

Berdasarkan uraian latar belakang tersebut, untuk mempercepat proses bisnis dan pengelolaan transaksi, laporan, dan pemesanan pada Mutiara Studio Photo, penulis mengangkat penelitian tugas akhir dengan judul “**Pembangunan Sistem Informasi Pengelolaan Transaksi Berbasis *Web* Pada Mutiara Studio Photo Padang**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka didapatkan rumusan masalah dalam penelitian ini adalah bagaimana melakukan pembangunan Sistem Informasi Pengelolaan Studio Foto Berbasis Web Pada Mutiara Studio Photo Padang.

1.3 Batasan Masalah

Agar permasalahan yang di angkat pada penelitian ini tidak meluas, perlu dibatasi yaitu sebagai berikut:

1. Sistem yang dirancang dan dibangun pada penelitian adalah khusus untuk mengatasi permasalahan dalam pengelolaan transaksi, laporan, dan pemesanan pada Studio Photo Mutiara.
2. Sistem diakses melalui web hanya dapat digunakan oleh user yang sudah terdaftar dan memiliki hak akses didalam aplikasi.
3. Aplikasi ini dibangun menggunakan bahasa pemrograman PHP dan database MYSQL.
4. Metode pengujian yang digunakan adalah *Black box testing* untuk menguji ketersediaan fungsional dan kesesuaian system dengan rancangan sistem yang diusulkan.

1.4 Tujuan Penelitian

Tujuan dari pembangunan sistem informasi pengelolaan studio foto adalah sebagai berikut.

1. Membangun sebuah sistem informasi yang dapat memudahkan user dalam mencatat setiap transaksi yang terjadi pada Studio Foto Mutiara.

2. Mempermudah proses booking dan memudahkan pengelolaan laporan penjualan serta dengan adanya sistem informasi ini dapat mengurangi penggunaan kertas pada Studio Foto Mutiara.

1.5 Sistematika Penulisan

Penulisan laporan Tugas Akhir ini secara sistematis dibagi menjadi enam bab, yaitu:

1. BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan laporan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang landasan teori dan informasi pendukung yang digunakan untuk penelitian ini.

3. BAB III MOTODOLOGI PENELITIAN

Bab ini menjelaskan tentang objek penelitian, metode pengumpulan data, metode pengembangan sistem yang digunakan, *flow chart* dan jadwal penelitian.

4. BAB IV ANALISIS DAN PERANCANGAN

Bab ini berisi tentang pemodelan analisis sistem menggunakan *tools* seperti, *Business Process Model Notation* (BPMN), *usecase diagram*, *usecase scenario*, *sequence diagram*, perancangan basis data, struktur basis data dan tabel, *class diagram*, arsitektur aplikasi dan perancangan antarmuka.

5. BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi tentang pengimplementasian aplikasi ke dalam bahasa pemrograman berdasarkan analisis dan perancangan, serta pengujian terhadap hasil implementasi sistem.

6. BAB VI PENUTUP

Berisi tentang kesimpulan terhadap hasil penelitian dan saran untuk pengembangan sistem kedepannya.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan tentang landasan teori dan informasi pendukung yang digunakan dalam penelitian. Dilengkapi dengan teori yang mendasari penelitian meliputi penjelasan tentang fotografi, studio foto, sistem informasi, studi literatur terkait pembangunan sistem informasi pengelolaan Studio Photo Mutiara berbasis web, perangkat lunak pendukung serta gambaran mengenai sistem yang sedang berjalan.

2.1 Fotografi

Fotografi berarti proses atau metode untuk menghasilkan gambar atau foto dari suatu objek dengan merekam pantulan cahaya yang mengenai objek tersebut pada media yang peka cahaya. Kata dari fotografi berasal dari dua istilah yunani: photo dari *phos* (cahaya) dan graphy dari *graphe* (tulisan atau gambar). Maka makna harfiah fotografi adalah menulis atau menggambar dengan cahaya. Dengan ini maka identitas fotografi bisa digabungkan menjadi kombinasi dari sesuatu yang terjadi secara alamiah (cahaya) dengan kegiatan yang diciptakan oleh manusia dengan budaya (menulis dan menggambar/melukis). Fotografi (*Photography*) berasal dari bahasa yunani yang berarti menggambar dengan cahaya. *Photography* adalah melukis dengan cahaya, prosesnya adalah foto tersebut merekam pantulan cahaya yang mengenai objek pada media yang peka cahaya (Kusumo, Bangsa, & Hosana, 2019).

Menurut Elliott Erwitt, Fotografi merupakan sebuah seni observasi. Ini tentang menemukan suatu hal yang menyenangkan di tempat biasanya. hal itu tidak akan ada hubungannya dengan hal-hal yang kamu lihat dan semua harus dilakukan dengan cara kamu melihat mereka. hal itu tidak akan ada hubungannya dengan hal-hal yang kamu lihat dan semua harus dilakukan dengan cara kamu melihat mereka.

Menurut Ansel Adams, Fotografi sebagai media berkesepsi dan komunikasi yang kuat, menawarkan berbagai persepsi, interpretasi dan eksekusi yang tak terbatas (Kusumo, Bangsa, & Hosana, 2019).

2.2 Sistem Informasi

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan. Tapi tidak semua data yang mengalir digunakan sebagai bahan pertimbangan pengambilan keputusan. Oleh karena itu dibutuhkan, suatu sistem informasi yang dapat mengelola data. Sistem itu harus dirancang sedemikian rupa agar dapat menentukan validitas data yang berasal dari berbagai sumber. Muhammad Sadeli (2011 : 13) mendefinisikan sistem informasi sebagai berikut : “ Sistem informasi merupakan merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi”. Jurnal Ilmiah Media Processor Vol.9 No.2, Juni 2014 ISSN 1907-6738 200 Budi Sutedjo Dharma Oetomo (2006 : 11) mendefinisikan sistem informasi sebagai berikut : “ Sistem informasi merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Hanif Al Fatta (2007 : 9) mendefinisikan sistem informasi sebagai berikut : “ Sistem informasi di definisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya”. Dari beberapa definisi di atas dapat disimpulkan bahwa sistem informasi merupakan sejumlah komponen yang terdiri dari manusia, komputer, hardware, software, jaringan komunikasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai tujuan tertentu (Afriyonna, Hendrawan, & Nugroho, 2019).

2.3 Perangkat Lunak Pendukung

Berikut ini merupakan penjelasan mengenai perangkat lunak pendukung yang digunakan dalam pembangunan sistem informasi pengelolaan studio foto pada Mutiara Studio Photo.

2.3.1 Bahasa Pemrograman PHP

Komputer adalah alat bantu untuk menjalankan perintah-perintah di dalam algoritma yang telah dimasukkan ke dalamnya. Agar komputer mengerti perintah yang dimaksudkan, maka perintah tersebut harus ditulis dalam bahasa khusus, yaitu bahasa komputer. Oleh karena itu algoritma yang ditulis dalam bahasa komputer

dinamakan program. Bahasa yang digunakan dalam menulis program komputer dinamakan dengan bahasa pemrograman. PHP (*Perl Hypertext Preprocessor*) adalah salah satu bahasa pemrograman yang digunakan untuk pengembangan web, seperti yang digunakan dalam penelitian ini. Berikut beberapa keunggulan PHP (Solichin, 2016)

1. Gratis

PHP dapat diunduh dan dipergunakan secara gratis. PHP dikembangkan oleh komunitas open source dan akan selalu didistribusikan secara gratis.

2. PHP berlisensi GNU General Public License (GPL)

Hal tersebut menjadi jaminan bahwa semua versi PHP akan selalu didistribusikan secara gratis. Walaupun demikian, untuk setiap program dan aplikasi yang dihasilkan menggunakan PHP, pengembang dapat menentukan lisensinya sendiri. Selain itu dalam proses distribusinya tetap diperkenankan menarik biaya dalam jumlah yang wajar, misalnya sebagai biaya packaging atau burning ke CD ROM.

3. Performa Handal

PHP sangatlah efisien, dengan sebuah spesifikasi server yang tidak mahal dapat melayani jutaan akses per hari.

4. Dukungan Basis Data

PHP mendukung hampir semua perangkat basisdata. Mulai dari MySQL, Oracle, PostgreSQL, Informix, Interbase, hingga SQLite. Bahkan beberapa diantaranya sudah terhubung dengan PHP secara native (default).

5. Pustaka bawaan

PHP dibangun secara khusus untuk aplikasi berbasis web. Oleh karena itu, disediakan banyak sekali pustaka bawaan yang terkait dengan web yang dapat langsung digunakan.

6. *Cross Platform*

PHP dapat dijalankan dengan baik pada hampir semua sistem operasi.

7. Mudah dipelajari

Perintah-perintah PHP sangat mudah dipelajari, karena sebagian besar perintah PHP diadopsi dari bahasa pemrograman populer seperti C/C++, Java, dan Perl.

2.3.2 Database MySQL

MySQL adalah sebuah sistem manajemen database yang bersifat *open source*. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia (Solichin, 2016). MySQL merupakan *database* yang dikembangkan dari bahasa SQL (Structure Query Language). SQL sendiri merupakan bahasa yang terstruktur yang digunakan untuk interaksi antara *script* program dengan *database server* dalam hal pengolahan data. Dengan SQL, kita dapat membuat tabel yang nantinya akan diisi dengan data, manipulasi data (misalnya menambah data, menghapus data dan memperbaharui data), serta membuat suatu perhitungan dengan berdasarkan data yang ditemukan. MySQL merupakan *software* resmi yang dikembangkan oleh perusahaan Swedia bernama MySQL AB, yang waktu itu bernama TcX Data Konsult AB.

Pada awalnya MySQL memakai nama mSQL atau “mini SQL” sebagai antarmuka yang digunakan, ternyata dengan menggunakan mSQL itu mengalami banyak hambatan, yaitu sangat lambat dan tidak fleksibel. Oleh karena itu, Michael Widenius berusaha mengembangkan *interface* yang tersebut hingga ditemukan MySQL. Kala itu, MySQL didistribusikan secara khusus, yakni untuk keperluan nonkomersial bersifat gratis, sedangkan untuk kebutuhan komersial diharuskan membayar lisensi. Barulah sejak versi 3.23.19, MySQL dikategorikan *software* berlisensi GPL, yakni dapat dipakai tanpa biaya untuk kebutuhan apapun. MySQL memiliki keuntungan dan juga kekurangan (Schwartz, Peter, & Tkachenko, 2012):

Keuntungan MySQL:

1. MySQL dapat dijalankan di banyak platform: saat ini dapat dijalankan di Linux, Windows, Mac, Solaris, OS2.
2. Banyak API bahasa pemrograman yang terintegrasi dengan MySQL: MySQL merupakan database sistem yang memiliki performa yang baik, sehingga banyak bahasa pemrograman yang memiliki API untuk mengintegrasikan MySQL untuk dipakai sebagai database.
3. Performa dapat dioptimalkan: dengan arsitektur yang dimiliki MySQL, pengguna dapat melakukan konfigurasi MySQL untuk mendapatkan performa yang optimal.

4. MySQL merupakan DBMS yang *open-source & freeware*: perbaikan *error &* pengembangan lebih mudah karena adanya forum yang aktif.
5. MySQL irit *resources*: MySQL menggunakan *engine default* bernama MyISAM menggunakan sedikit *disk space, memory* dan *CPU usage*.

Kekurangan MySQL adalah:

1. MySQL tidak memiliki fitur-fitur seperti *Analysis Services, Integration Services* dan *service-service* lainnya yang disediakan dari database sistem yang berbayar.
2. MySQL tidak memiliki algoritma join selengkap Oracle, SQL Server dan PostgreSQL.

2.4 Alat Pemodelan Aplikasi

Berikut ini penjelasan mengenai alat pemodelan aplikasi yang digunakan dalam pembangunan sistem informasi pengelolaan studio foto berbasis web pada Mutiara Studio Photo Padang.

2.4.1 BPMN (*Business Process Model Notation*)

Business Process Modeling Notation (BPMN) menggambarkan suatu diagram proses bisnis berdasarkan teknik diagram alur, dirangkai untuk membuat model-model grafis dari operasi-operasi bisnis dimana terdapat aktivitas-aktivitas dan kontrol-kontrol alur yang mendefinisikan urutan kerja.

1. *Flow Objects*

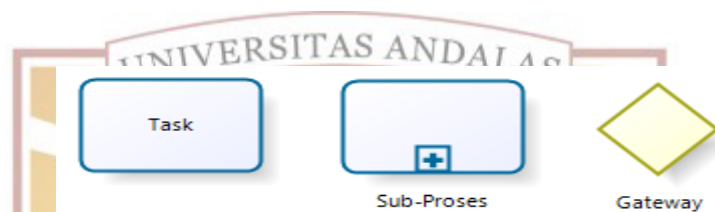
Flow objects dibagi menjadi 3, yaitu *event, activity* dan *gateway*.

- a. *Event*, digambarkan dengan sebuah lingkaran dan merupakan sesuatu yang “terjadi” selama berlangsungnya proses bisnis. *Event* mempengaruhi aliran proses dan memiliki penyebab (*trigger*) atau hasil (*result*). *Event* terbagi atas 3 tipe yaitu *start, intermediate*, dan *end*. Simbol *event* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Simbol Event (Dewi, Indahyanti, & S, 2012)

- b. *Activity*, ditunjukkan dengan persegi panjang dengan ujung-ujung bulat dan merupakan bentuk umum untuk pekerjaan yang dilakukan oleh perusahaan. Tipe dari aktivitas adalah *task* dan *sub process* yang dibedakan dengan tanda + pada bagian tengah bawah dari bentuk tersebut.
- c. *Gateway*, digambarkan dengan bentuk seperti belah ketupat dan digunakan untuk mengontrol percabangan dan penggabungan *Sequence Flow*. Jadi, *gateway* menentukan keputusan tradisional, penggabungan, dan penggabungan aliran. Internal Markers akan menentukan perilaku dari kontrol. Simbol *activity* dan *gateway* dapat dilihat pada Gambar 2.2



Gambar 2.2 Simbol Activity dan Gateway (Dewi, Indahyanti, & S, 2012)

2. *Connecting Objects*

Connecting Objects adalah elemen yang menghubungkan *flow object*. *Connecting Objects* memiliki 3 jenis elemen, yaitu:

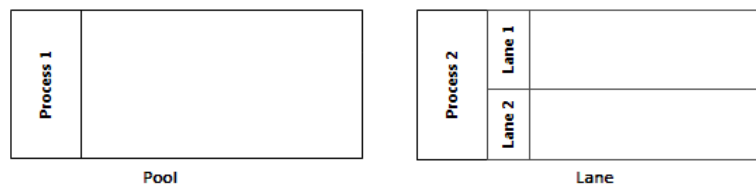
- a. Alur *Sequence* (*Sequence Flow*)
Digunakan untuk menunjukkan urutan yang kegiatannya dilakukan dalam sebuah proses.
- b. Alur Pesan (*Message Flow*)
Digunakan untuk menunjukkan aliran pesan antara dua entitas yang siap untuk mengirim dan menerima.
- c. Asosiasi (*Association*)
Digunakan untuk asosiasi data, informasi dan artefak dengan aliran benda. Tipe-tipe *connecting object* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Tipe-Tipe Connecting Object (Dewi, Indahyanti, & S, 2012)

3. *Swimlanes*

Swimlanes digambarkan dengan bentuk garis yang memisahkan dan mengelompokkan aktor. *Swimlanes* digunakan sebagai mekanisme untuk membagi kategori visual yang menggambarkan kemampuan fungsional atau tanggung jawab yang berbeda. Lanes digunakan untuk mengatur dan mengkategorikan aktivitas. Simbol *pool* dan *lane* dalam *swimlane* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Simbol Pool dan Lane dalam Swimlane (Dewi, Indahyanti, & S, 2012)

4. *Artifacts*

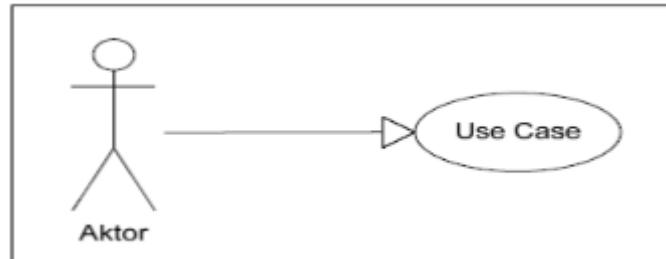
Artifacts adalah elemen yang digunakan untuk memberikan informasi tambahan dari sebuah proses. Versi BPMN saat ini memiliki 3 tipe *artifacts*, yaitu:

- Data object*: mekanisme untuk menunjukkan bagaimana data dibutuhkan atau diproduksi oleh aktivitas. *Data object* dihubungkan dengan aktivitas melalui *associations*.
- Group*: diwakili dengan persegi panjang dengan ujung bulat yang digambarkan dengan garis putus-putus. *Group* dapat digunakan untuk tujuan dokumentasi atau analisis, tetapi tidak mempengaruhi *Sequence Flow*.
- Annotation*: mekanisme untuk pemodelan memberikan informasi teks tambahan untuk pembaca dari diagram BPMN

2.4.2 *Use Case Diagram*

Use Case Diagram merupakan teknik menangkap kebutuhan-kebutuhan fungsional dari sistem baru atau sistem yang diubah. Setiap *use case* terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. *Use*

case diagram menggambarkan secara grafis hubungan *actor* dan satu atau lebih *use case*. Penggambarannya menggunakan notasi gambar orang, anak panah, dan elips (Artina, 2006). Contoh dari *use case* sederhana terlihat pada Gambar 2.5.



Gambar 2.5 Use Case (Artina, 2006)

Use Case Diagram juga merupakan teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case Diagram* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Simbol-simbol pada *use case diagram* dapat dilihat pada Tabel 2.1 berikut ini.

Tabel 2.1 Simbol usecase (Artina, 2006)

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen lain.
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

Tabel 2.1 Simbol *usecase* (Artina, 2006) (Lanjutan)

4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

Masing-masing *use case diagram* akan dijabarkan ke dalam *use case scenario*. Di dalam *use case scenario* akan diberi uraian nama *actor* yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, pra-kondisi yang harus dipenuhi, dan pasca-kondisi yang diharapkan setelah berjalannya fungsional *use case*. Selain itu juga diberikan usulan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh actor, baik sebagai proses utama yang pasti dilakukan oleh sistem maupun proses alternatif (Joubert & Swart, 2019).

2.4.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *Message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja

yang terjadi secara internal dan *output* apa yang dihasilkan (Dharwiyanti & Wahono, 2003). Masing-masing objek, termasuk aktor memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek yang lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

2.4.4 ERD (*Entity Relationship Diagram*)

ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Pada akhirnya ERD bisa juga digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang akan dibangun (Fatta, 2007). Dalam pembentukan ERD terdapat 3 komponen yang akan dibentuk, yaitu:

1. Entitas

Entitas merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang

2. Hubungan (*relationship*)

Berfungsi untuk menggambarkan hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.

3. Atribut

Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data.

2.5 Referensi Penelitian

Sebagai referensi untuk memberikan wawasan pengetahuan mengenai aplikasi ini penulis merujuk kepada beberapa penelitian lain yang sudah pernah dilakukan sebelumnya dengan topik yang berkaitan dengan penelitian ini yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 Referensi Penelitian

Penulis	Judul	Metode	Tujuan	Kesimpulan
1. Eko Budi Mulyono 2. Benni Purnama 3. Effiyaldi	Sistem Informasi Jasa Fotografi Berbasis Web Pada Green Photografi Jambi	<i>Waterfall</i>	1. Agar dapat merancang sebuah sistem informasi berbasis web yang mudah digunakan. 2. Dalam penulisan ini penulis merancang sistem informasi pada green photography yang memudahkan user untuk mengelola berbagai data jasa yang ada	1. Penelitian ini menghasilkan sistem informasi pemesanan paket foto berbasis website pada Greenphotography sebagai media promosi agar dapat di kenal oleh masyarakat. 2. Aplikasi yang telah dirancang dapat membantu meningkatkan produktivitas dan efisiensi kerja dan menambah tingkat promosi 3. Dengan adanya sistem pemesanan ini dapat memberikan pelayanan yang cepat dan tepat bagi konsumen dan juga dengan adanya sistem ini konsumen tidak perlu menghubungi pihak pemilik fotografi untuk mengklarifikasi

Tabel 2.2 Referensi Penelitian (lanjutan)

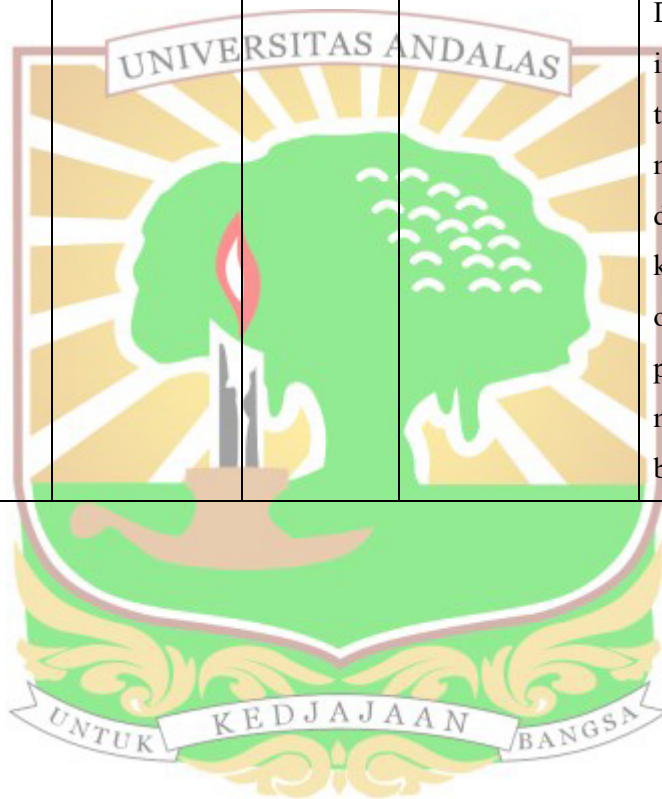
				pembayaran, karena di sistem telah disediakan menu untuk mengklarifikasi pembayarannya.
1. Arman 2. Liranti Rahmelina 3. Sotar 4. Yuliana Siska	Perancangan Antar Muka Aplikasi Pemesanan Fotografi Dan Make Up Berbasis Online Pada Queen Photo Studio Padang	Metode OOP dengan model air terjun (<i>waterfall</i>)	Mempermudah pihak Queen dan pelanggan.	1. Dengan adanya aplikasi ini, Queen photo studio sudah tidak menggunakan media kertas lagi dalam menerima pemesanan dari pelanggan. 2. Jadi Pihak Queen photo studio sudah tidak kesulitan lagi untuk merekap data pelanggan karena sudah menggunakan aplikasi yang terkomputerisasi. Serta tidak ada kemungkinan data rusak atau hilang karena sudah terminimalisir dengan aplikasi ini. 3. Pelanggan sudah tidak kesulitan lagi untuk mengantri dengan pelanggan lain ketika memesan

Tabel 2.2 Referensi Penelitian (lanjutan)

				jasa layanan fotografi maupun makeup pada Queen photo studio serta penggunaan waktu menjadi lebih efektif dan efisien, sehingga ke depan pelanggan Queen Photo studio semakin banyak dan omsetnya juga semakin meningkat, dan Queen juga dapat bersaing dengan perusahaan sejenis maupun dengan perusahaan yang lain.
1. Afriyonna 2. Hendrawan 3. Agus Nugroho	Perancangan Sistem Informasi administrasi Jasa Foto Pernikahan Berbasis Web Pada Euphoria Photo Studio	<i>Waterfall</i>	1. Merancang website pada Euphoria Photo Studio. 2. Merancang sebuah website yang dapat memberikan kemudahan bagi pelanggan dalam memperkirakan harga paket foto pernikahan dan dapat memberikan	1. Penelitian ini menghasilkan sistem informasi administrasi jasa foto berbasis web pada Euphoria Photo Studio yang dapat memudahkan dalam pengelolaan transaksi pemesanan jasa foto. 2. Dengan adanya sistem informasi ini pemesanan paket foto dapat dilakukan secara online.

Tabel 2.2 Referensi Penelitian (lanjutan)

			kemudahan bagi admin dalam mengolah data.	Pelanggan dapat memilih paket foto yang sesuai dengan yang diinginkan dan melakukan kostumisasi terhadap paket foto yang di pesan. Dengan adanya sistem informasi ini proses transaksi menjadi lebih mudah, karena sistem dapat melakukan kalkulasi harga secara otomatis sehingga pelanggan bisa menyesuaikan dengan budget yang dimiliki.
--	--	--	---	---



BAB III

METODOLOGI PENELITIAN

Pada bab ini membahas mengenai objek kajian beserta lokasinya, pengumpulan data yang digunakan, metode pengembangan aplikasi, dan metode penelitian yang digunakan untuk merancang sistem informasi pengelolaan pada Mutiara Studio Photo berbasis web, sebagai objek penelitian.

3.1 Objek Penelitian

Objek kajian dan penelitian ini adalah Studio Foto Mutiara yang berlokasi di Jalan Durian Tarung no.11, Kecamatan Kuranji. Pemilihan objek ini dikarenakan pada studio foto masih menggunakan cara konvensional dan belum memiliki sistem informasi yang dapat mempermudah dalam pengelolaan studio foto.

3.2 Metode Pengumpulan Data

Metode-metode yang digunakan dalam pengumpulan data untuk membangun Sistem informasi pengelolaan studio foto pada Mutiara Studio Photo berbasis web ini adalah:

1. Observasi

Observasi atau survei yang dilakukan dengan pengamatan langsung terhadap objek penelitian yang telah ditetapkan. Seperti alur proses bisnis yang ada serta tata cara kerja aktor yang terlibat dalam proses pengelolaan pada Mutiara Studio Photo.

2. Wawancara

Wawancara dilakukan untuk mendapatkan informasi lebih detail tentang proses bisnis yang ada pada objek penelitian. Pada penelitian ini wawancara dilakukan dengan melakukan tanya-jawab dengan pemilik studio photo.

3. Studi Literatur

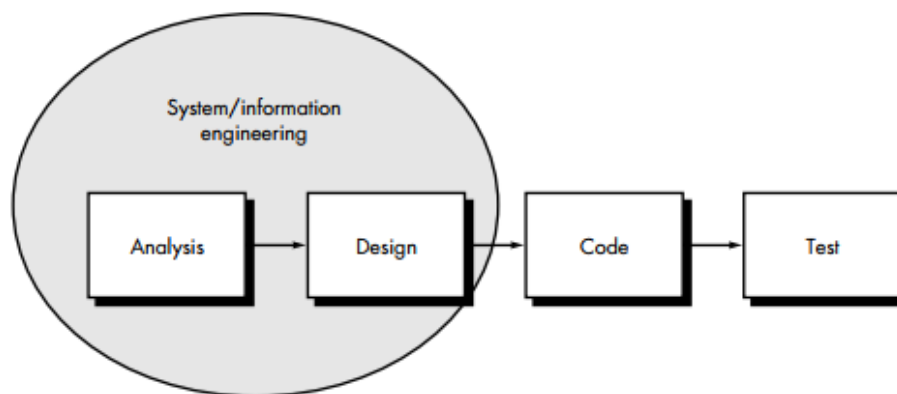
Pada tahapan ini studi literatur dilakukan dengan mempelajari dan memahami literatur yang berkaitan dengan teori dari berbagai sumber seperti situs internet, jurnal ilmiah dan bacaan lain yang mendukung penelitian.

4. Dokumentasi

Pada metode ini dilakukan pengumpulan data berupa dokumen-dokumen yang dilakukan dengan meminta ke studio foto Mutiara seperti nota/faktur transaksi dan data pelaporan keuangan.

3.3 Metode Pengembangan Sistem

Pada penelitian ini, aplikasi yang dikembangkan menggunakan metode pengembangan sistem model *waterfall*. Metode *waterfall* adalah proses pengembangan perangkat lunak tradisional yang umum digunakan dalam pembangunan proyek perangkat lunak. Dan alasan kenapa metode ini disebut *waterfall* adalah karena proses mengalir “secara sistematis dari satu tahap ke tahap lainnya dalam mode ke bawah”. Sehingga, *output* dari satu tahapan menjadi *input* untuk tahapan berikutnya, model sekuensial. Model *waterfall* adalah model klasik bersifat sistematis, berurutan dalam membangun *software* (Pressman, 2010). Berikut gambaran metode *waterfall* pada Gambar 3.1.



Gambar 3.1 Waterfall Model (Pressman, 2010)

Tahapan metode *waterfall* yang dilakukan pada penelitian ini hanya empat tahapan yaitu *analysis*, *design*, *coding* dan *testing*. Berikut penjelasan masing-masingnya:

1. *Analysis*

Pada tahap ini dilakukan identifikasi terhadap permasalahan yang terjadi pada Studio Foto Mutiara. Selanjutnya dilakukan analisa terhadap kebutuhan yang

diperlukan untuk pembangunan dan pengembangan aplikasi. Proses analisis ini digambarkan dengan BPMN (*Business Process Model Notation*).

2. *Design*

Pada tahap *design*, dilakukan perancangan semua kebutuhan yang telah dianalisa pada tahapan sebelumnya. Perancangan yang dilakukan terdiri dari perancangan ERD (*Entity Relational Diagram*), arsitektur perangkat lunak, perancangan antarmuka dengan menggunakan *tools* UML.

3. *Coding*

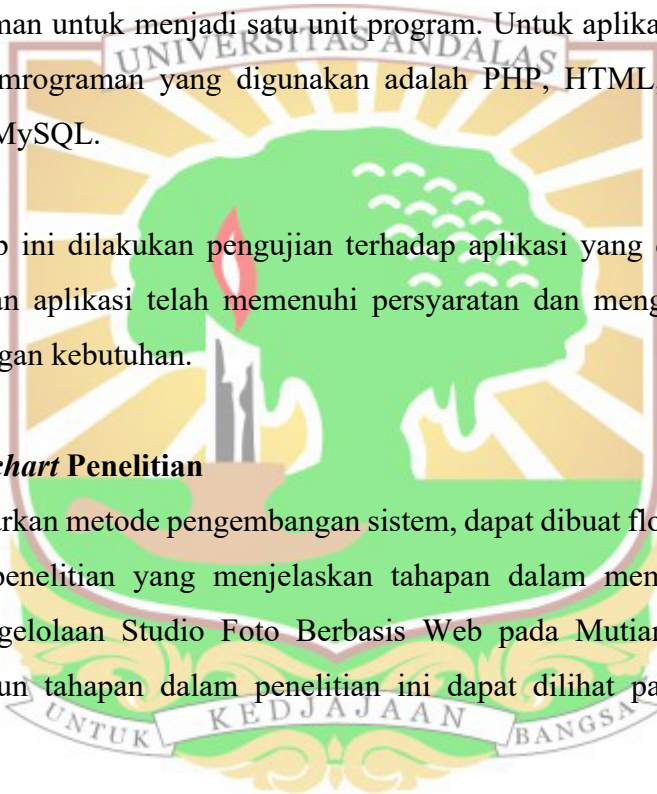
Pada tahap *coding*, semua *design* yang telah disiapkan diubah kedalam bahasa pemrograman untuk menjadi satu unit program. Untuk aplikasi berbasis web, bahasa pemrograman yang digunakan adalah PHP, HTML, *javascript* dan *database* MySQL.

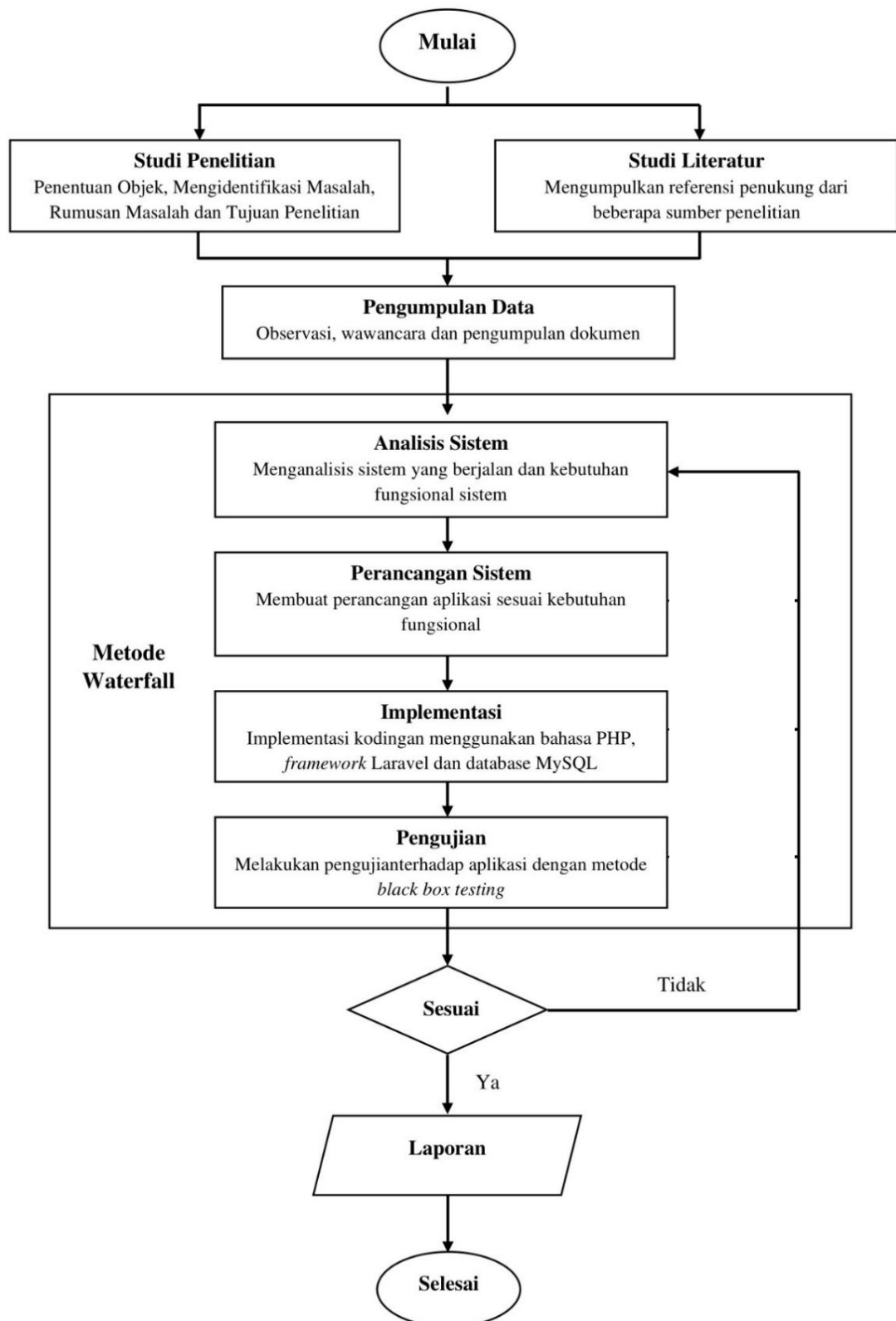
4. *Testing*

Pada tahap ini dilakukan pengujian terhadap aplikasi yang dirancang untuk memastikan aplikasi telah memenuhi persyaratan dan menghasilkan output sesuai dengan kebutuhan.

3.4 **Flowchart Penelitian**

Berdasarkan metode pengembangan sistem, dapat dibuat flowchart atau alur dalam proses penelitian yang menjelaskan tahapan dalam membangun Sistem Informasi Pengelolaan Studio Foto Berbasis Web pada Mutiara Studio Photo Padang. Adapun tahapan dalam penelitian ini dapat dilihat pada Gambar 3.2 berikut.





Gambar 3.2 Flowchart Penelitian

Berdasarkan Gambar 3.2, dapat dijelaskan flowchart penelitian ini sebagai berikut:

1. Studi Pendahuluan

Tahap awal dalam penelitian ini adalah studi pendahuluan. Studi pendahuluan berupa penetapan objek penelitian dan mengidentifikasi masalah yang ada pada objek penelitian yang kemudian dijadikan topik dalam penelitian ini. Adapun objek yang dikaji adalah sistem pengelolaan studio foto pada Studio Foto Mutiara.

2. Studi Literatur

Studi literatur dilakukan dengan cara mengumpulkan dan mempelajari literatur yang berkaitan dengan penelitian seperti buku, jurnal ilmiah, serta penelitian terkait yang telah dilakukan sebelumnya.

3. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data melalui observasi dan wawancara dengan pemilik Studio Foto Mutiara serta mengumpulkan data terkait yang berguna dalam menunjang penelitian.

4. Analisis Sistem

Pada tahap analisis sistem, dilakukan analisis kebutuhan fungsional user dan analisis proses bisnis yang sedang berjalan berdasarkan hasil wawancara yang telah dilakukan. Selanjutnya dibuat pemodelan proses bisnis menggunakan *tools* BPMN serta analisis fungsional sistem yang digambarkan menggunakan *use case diagram*, *use case scenario* dan *sequence diagram*.

5. Perancangan Sistem

Dalam tahap ini dilakukan perancangan terhadap sistem yang dibangun. Perancangan tersebut terdiri dari perancangan arsitektur aplikasi, perancangan *database*, dan perancangan *interface* aplikasi.

6. Implementasi

Pada tahap ini dilakukan implementasi dari perancangan sistem dengan melakukan pengkodean menggunakan bahasa pemrograman PHP dan menggunakan *framework* Laravel untuk aplikasi berbasis web dan MySQL untuk sistem penyimpanan data.

7. Pengujian

Dalam tahap pengujian, dilakukan pengujian terhadap aplikasi yang dibangun apakah telah sesuai dengan kebutuhan fungsional yang telah dirancang sebelumnya. Pengujian sistem dilakukan dengan menggunakan metode *black box testing*.

8. Laporan

Pada tahap ini dilakukan pembuatan laporan tentang analisis sistem, perancangan sistem dan pengujian terhadap sistem yang telah dibangun. Laporan ini menghasilkan dokumen yang dapat dijadikan acuan terhadap penelitian selanjutnya.



BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan hasil dari analisis pembangunan sistem informasi pengelolaan studio foto berbasis web pada Studio Foto Mutiara di Kota Padang. Pada tahapan analisis digambarkan menggunakan tools *Business Process Modeling Notation* (BPMN), analisis kebutuhan fungsional, *use case diagram*, *use case scenario*, *sequence diagram*, dan *class analysis*. Untuk tahap perancangan sistem digambarkan dengan *Entity Relationship Diagram*, *class diagram*, arsitektur aplikasi, dan *user interface*.

4.1 Analisis Sistem

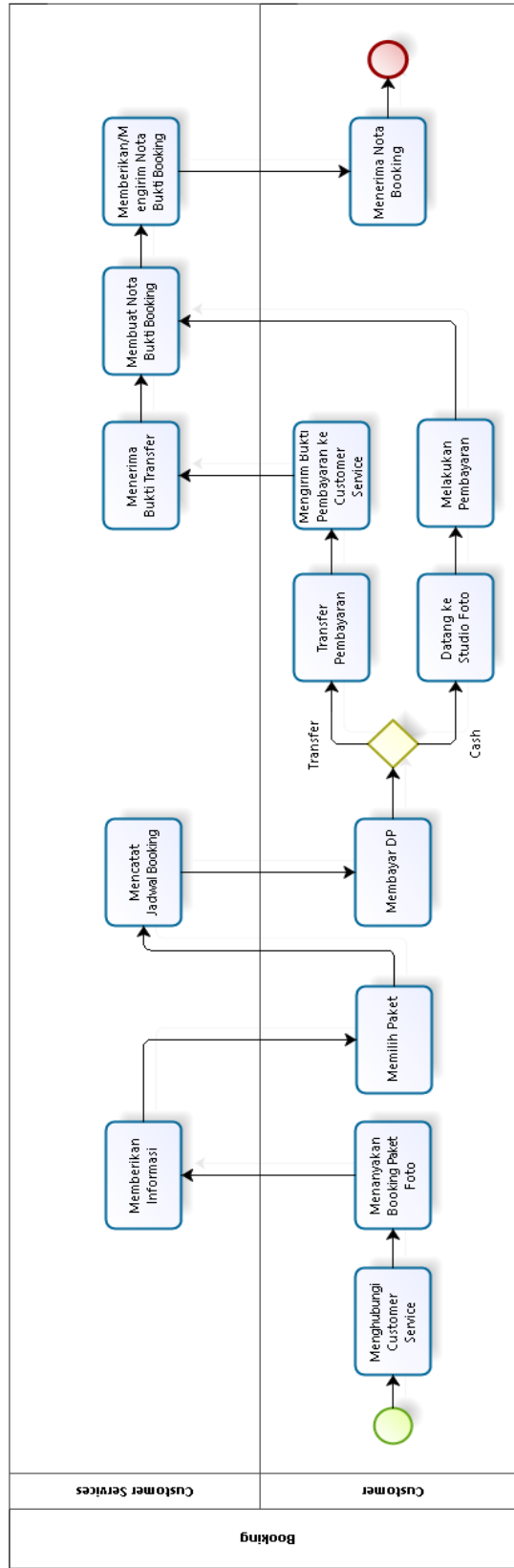
Analisis sistem dilakukan berdasarkan hasil observasi dan wawancara yang dilakukan langsung ke Studio Foto Mutiara Padang. Analisis sistem merupakan tahapan yang dilakukan dalam merencanakan sistem yang sesuai dengan kebutuhan fungsional sistem. Hasil dari pengumpulan data digambarkan menggunakan tools *Business Process Modeling Notation* (BPMN). Analisis sistem dimodelkan menggunakan *Unified Modelling Language* (UML) yang terdiri dari *use case diagram* dan *use case scenario*.

4.1.1 Sistem yang Sedang Berjalan

Pada bagian ini menjelaskan proses bisnis yang sedang berjalan pada Studio Foto Mutiara terkait proses pengelolaan dan pemesanan jasa foto. Berdasarkan hasil wawancara yang dilakukan dengan pihak Studio Foto Mutiara, proses tersebut terdiri dari proses booking, cetak foto dan laporan penjualan. Sistem tersebut dimodelkan menggunakan *Business Process Modelling Notation* (BPMN) dengan penjelasan sebagai berikut.

4.1.1.1 Proses bisnis booking yang sedang berjalan

Berikut ini adalah proses bisnis booking yang sedang berjalan pada Studio Foto Mutiara. Pemodelan proses booking ini dapat dilihat pada .



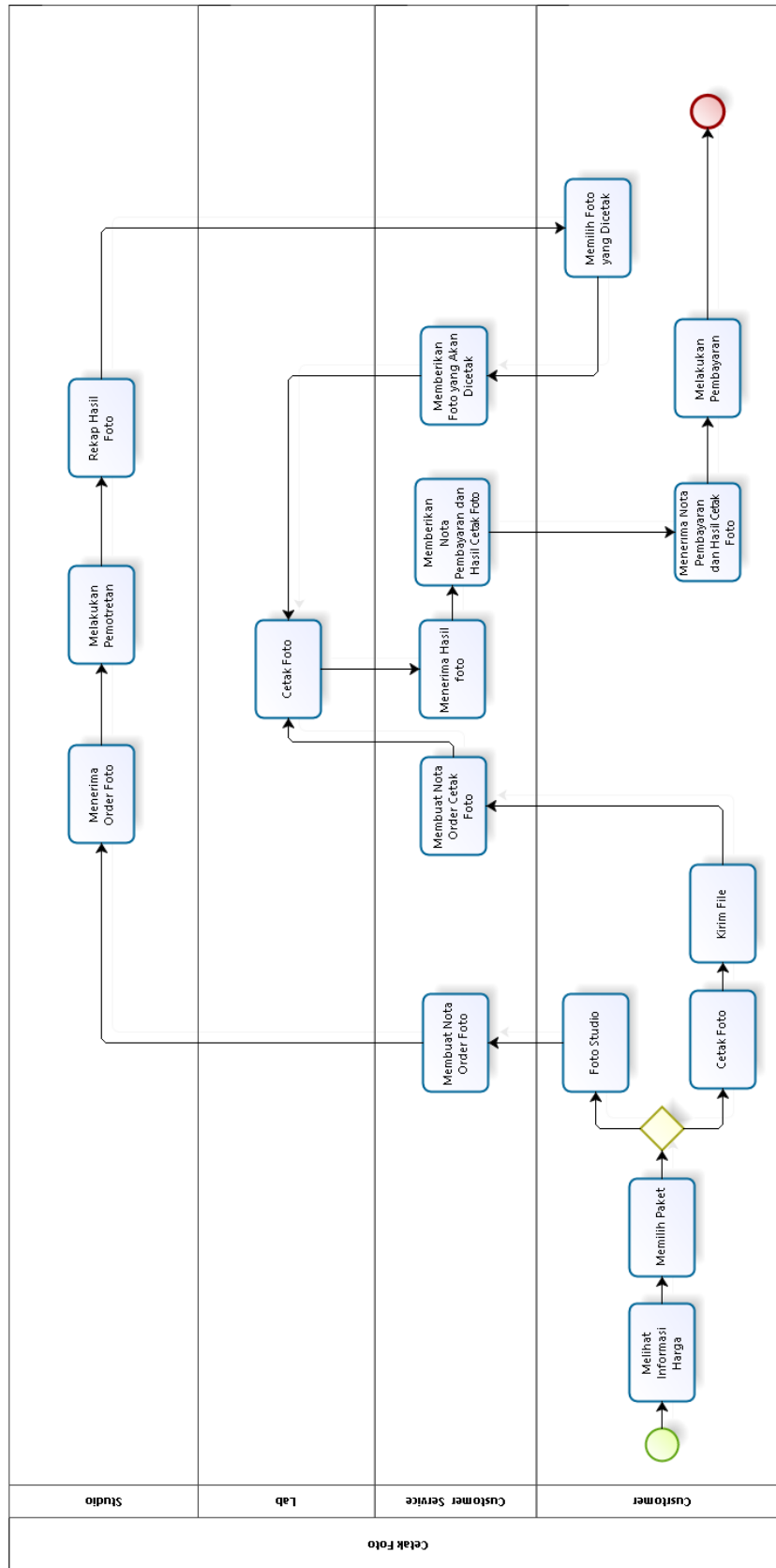
Gambar 4.1 BPMN Proses booking yang sedang berjalan

Berdasarkan BPMN proses booking yang sedang berjalan pada dapat diuraikan dengan penjelasan sebagai berikut.

1. *Customer* datang ke Mutiara Photo Studio kemudian menghubungi customer service (CS)
2. *Customer* menanyakan booking paket foto
3. CS memberikan informasi paket
4. *Customer* memilih paket
5. Kemudian CS mencatat jadwal booking
6. *Customer* melakukan pembayaran dp
7. Jika pembayaran melalui transfer, customer melakukan transfer pembayaran
8. *Customer* kemudian mengirim bukti pembayaran ke customer service
9. Kemudian CS menerima bukti transfer dan membuat nota bukti booking
10. CS mengirim nota bukti booking kepada customer
11. *Customer* menerima nota booking
12. Jika pembayaran cash, customer datang ke studio foto
13. Kemudian melakukan pembayaran
14. CS membuat nota booking
15. Kemudian CS memberikan nota bukti booking kepada customer
16. Customer menerima nota booking dari CS
17. Proses selesai

4.1.1.2 Proses Bisnis Cetak Foto yang Sedang Berjalan

Berikut ini merupakan proses bisnis cetak foto yang sedang berjalan di Studio Foto Mutiara. Pemodelan untuk proses bisnis ini dapat dilihat pada



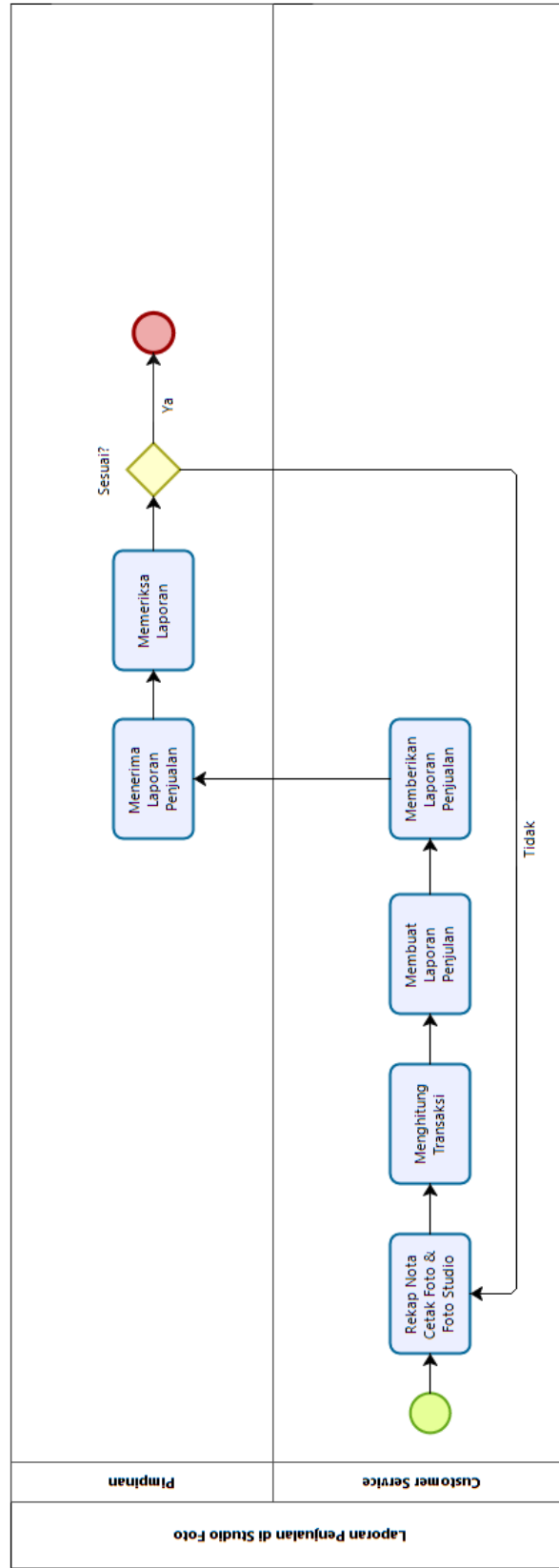
Gambar 4.3 BPMN Proses Cetak Foto yang Sedang Berjalan

Berdasarkan BPMN proses cetak foto yang sedang berjalan pada dapat diuraikan dengan penjelasan sebagai berikut.

1. *Customer* melihat informasi harga
2. Kemudian *customer* memilih paket yang tersedia
3. Terdapat dua jenis pilihan paket, yaitu foto studio dan cetak foto
4. Jika *customer* memilih paket foto studio
 - a. *Customer Service* (CS) membuat nota order foto studio
 - b. Studio Foto menerima order foto studio
 - c. Melakukan pemotretan
 - d. Studio merekap hasil pemotretan
 - e. *Customer* memilih foto yang akan dicetak
 - f. CS Memberikan foto yang akan dicetak kepada *Lab*
 - g. *Lab* menerima dan mencetak foto
 - h. CS menerima hasil cetak foto dan memberikan nota pembayaran
 - i. *Customer* menerima nota dan melakukan pembayaran
 - j. *Customer* menerima hasil foto
5. Jika *customer* memilih paket cetak foto
 - a. *Customer* mengirim file foto
 - b. *Customer Service* (CS) membuat nota order cetak foto
 - c. *Lab* menerima nota order dan mencetak foto
 - d. CS menerima hasil cetak foto
 - e. CS membuat nota pembayaran
 - f. *Customer* menerima nota dan melakukan pembayaran
6. Proses selesai

4.1.1.3 Proses Bisnis Laporan Penjualan yang Sedang Berjalan

Berikut ini adalah proses bisnis laporan penjualan yang sedang berjalan pada Studio Foto Mutiara. Pemodelan untuk proses bisnis ini dapat dilihat pada



Gambar 4.5 BPMN Proses Laporan Penjualan yang Sedang Berjalan

Berdasarkan BPMN proses laporan penjualan yang sedang berjalan pada dapat diuraikan dengan penjelasan sebagai berikut.

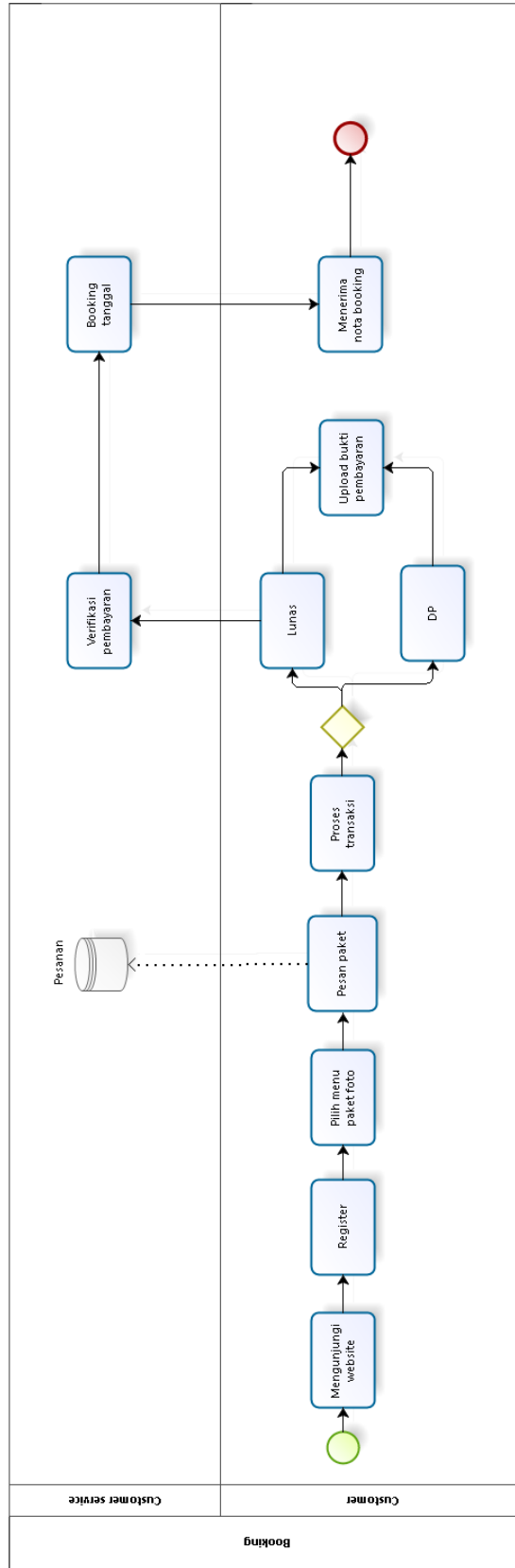
1. *Customer service* (CS) merekap nota cetak foto dan melakukan foto studio
2. CS menghitung jumlah transaksi yang ada
3. CS membuat laporan penjualan berdasarkan jumlah transaksi dan jumlah uang masuk
4. Setelah itu CS memberikan laporan penjualan
5. Pimpinan menerima laporan penjualan
6. Kemudian pimpinan memeriksa laporan
7. Jika sesuai maka laporan selesai
8. Jika tidak, pimpinan mengembalikan laporan ke CS
9. CS melakukan pemeriksaan ulang rekap nota dan cetak foto
10. CS memberikan laporan penjualan hasil pemeriksaan ulang
11. Pimpinan menerima laporan penjualan hasil pemeriksaan ulang
12. Jika sesuai, proses selesai

4.1.2 Sistem yang Diusulkan

Pembangunan sistem informasi pengelolaan studio foto berbasis web pada Studio Foto Mutiara yang diusulkan dalam analisis system ini juga dimodelkan dengan *Business Process Model and Notation* (BPMN). Berikut penjelasan dari sistem yang diusulkan untuk proses bisnis booking, cetak foto dan laporan penjualan pada Studio Foto Mutiara.

4.1.2.1 Proses Bisnis Booking yang diusulkan

Proses booking yang diusulkan dapat dilakukan oleh dua actor, yaitu: Customer dan customer service. Model proses bisnis booking yang diusulkan dapat dilihat pada



Gambar 4.7 BPMN proses booking yang diusulkan

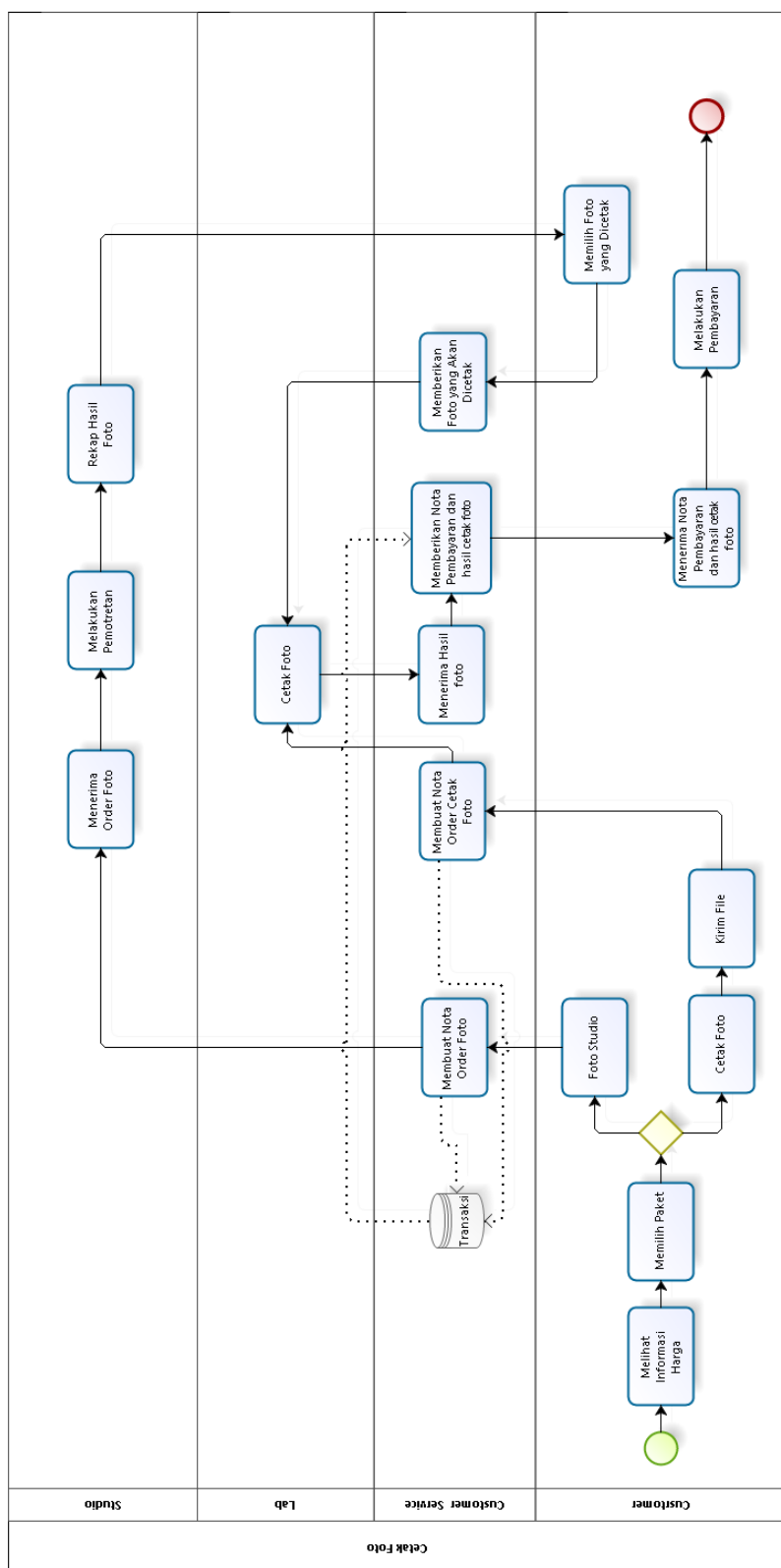
Berdasarkan BPMN proses booking yang diusulkan pada dapat
diuraikan dengan penjelasan sebagai berikut.

1. Customer Mengunjungi Website.
2. Customer Melakukan Register.
3. Customer memilih menu untuk memilih paket foto.
4. Setelah memilih paket foto customer melakukan pemesanan paket.
5. Customer service menginput ke Database pesanan.
6. Customer memproses transaksi.
7. Jika pembayaran lunas customer akan mengupload bukti pembayaran.
8. Kemudian customer service memverifikasi pembayaran.
9. Kemudian melakukan booking tanggal.
10. Customer menerima nota booking.
11. Jika pembayan DP, customer akan mengupload bukti pembayaran .
12. Kemudian customer service memverifikasi pembayaran.
13. Kemudian melakukan booking tanggal.
14. Customer menerima nota booking.

4.1.2.2 Proses Bisnis Cetak Foto yang Diusulkan

Proses cetak foto yang diusulkan dapat dilakukan oleh empat aktor, yaitu: Customer dan customer service, lab dan studio. Model proses bisnis cetak foto yang diusulkan dapat dilihat pada Gambar 4.10.

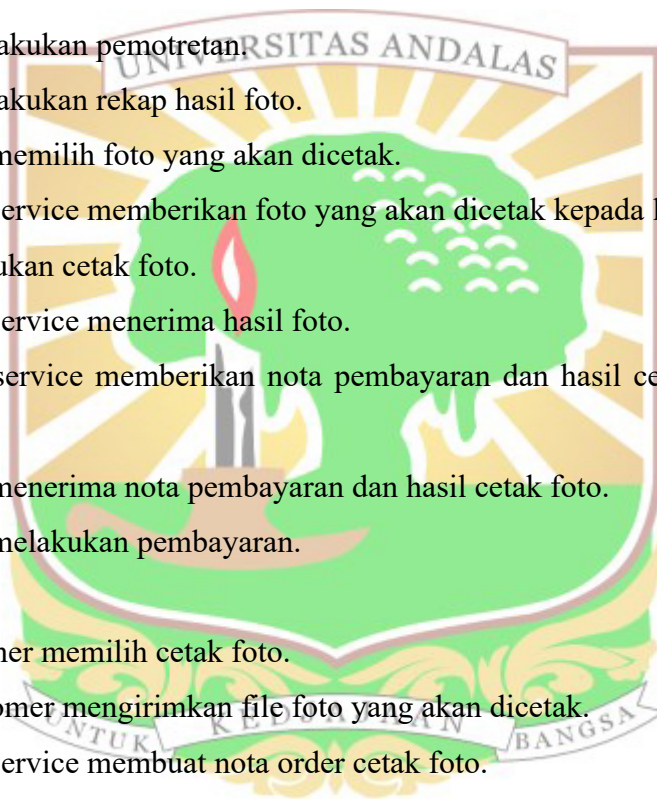




Gambar 4.9 BPMN cetak foto yang diusulkan

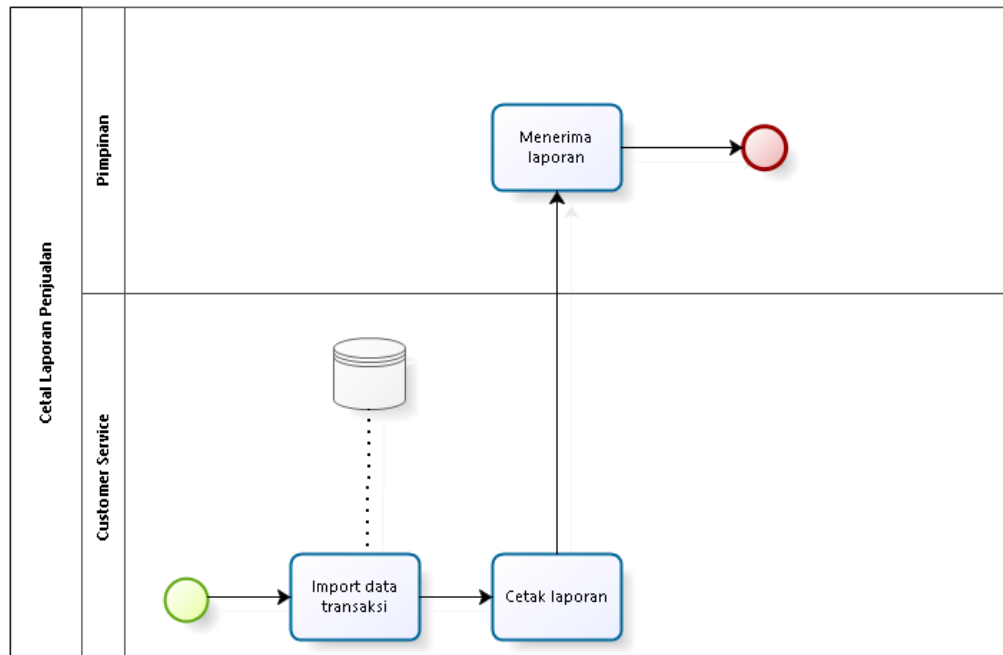
Berdasarkan BPMN proses cetak foto yang diusulkan pada dapat diuraikan dengan penjelasan sebagai berikut.

1. Customer melihat informasi harga.
2. Kemudian customer memilih paket.
3. Terdapat dua paket yaitu foto studio dan cetak foto.
4. Jika customer memilih foto studio.
5. Maka customer service membuat nota order foto.
6. Kemudian customer service menginputkan ke database transaksi.
7. Studio menerima order foto.
8. Studio melakukan pemotretan.
9. Studio melakukan rekap hasil foto.
10. Customer memilih foto yang akan dicetak.
11. Customer service memberikan foto yang akan dicetak kepada lab.
12. Lab melakukan cetak foto.
13. Customer service menerima hasil foto.
14. Customer service memberikan nota pembayaran dan hasil cetak foto kepada customer.
15. Customer menerima nota pembayaran dan hasil cetak foto.
16. Customer melakukan pembayaran.
17. Selesai
18. Jika customer memilih cetak foto.
19. Maka customer mengirimkan file foto yang akan dicetak.
20. Customer service membuat nota order cetak foto.
21. Customer service menginput kan kedalam database transaksi.
22. Lab melakukan cetak foto.
23. Customer service menerima hasil foto.
24. Customer service memberikan nota pembayaran dan hasil cetak foto.
25. Customer menerima nota pembayaran dan hasil cetak foto.
26. Customer melakukan pembayaran.
27. Proses selesai.



4.1.2.3 Proses Bisnis Cetak Laporan Penjualan yang Diusulkan

Proses cetak laporan penjualan yang diusulkan dapat dilakukan oleh dua actor, yaitu: Customer service dan pimpinan. Model proses bisnis cetak laporan penjualan yang diusulkan dapat dilihat pada Gambar 4.11.



Gambar 4.11 BPMN cetak laporan penjualan yang diusulkan

Berdasarkan BPMN proses cetak laporan penjualan yang diusulkan pada dapat diuraikan dengan penjelasan sebagai berikut.

1. Customer service mengimport data transaksi ke dalam database.
2. Customer service mencetak laporan.
3. Pimpinan menerima laporan.

4.1.3 Analisis Kebutuhan Fungsional

Berdasarkan analisis alur proses pengelolaan studio foto yang diusulkan, terdapat dua user yaitu admin dan customer. Maka dapat dirumuskan kebutuhan fungsionalnya sebagai berikut, yaitu:

1. Admin dapat melakukan login
2. Admin dapat mengelola halaman galeri
3. Admin dapat mengelola paket
4. Admin dapat mengelola transaksi pembayaran
5. Admin dapat melihat laporan transaksi

Kebutuhan fungsional pada Pimpinan

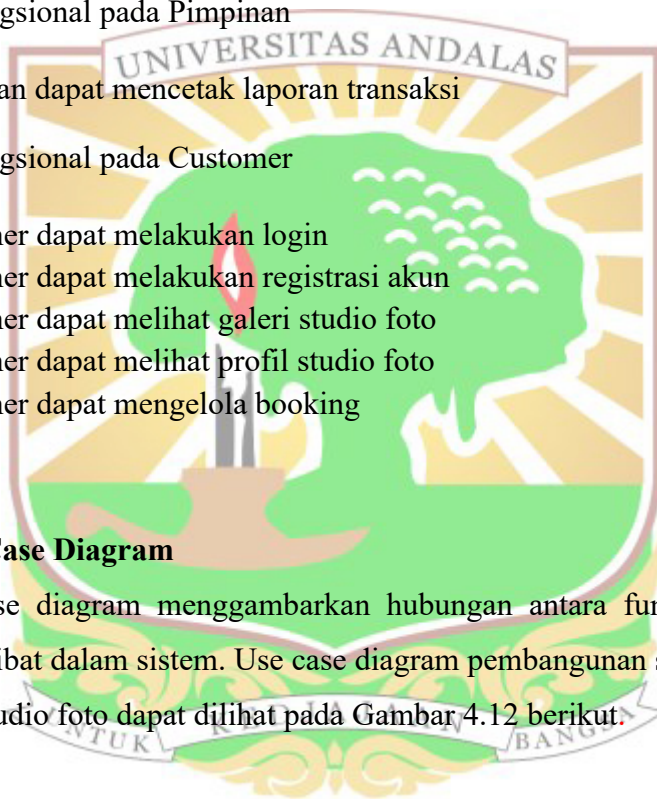
1. Pimpinan dapat mencetak laporan transaksi

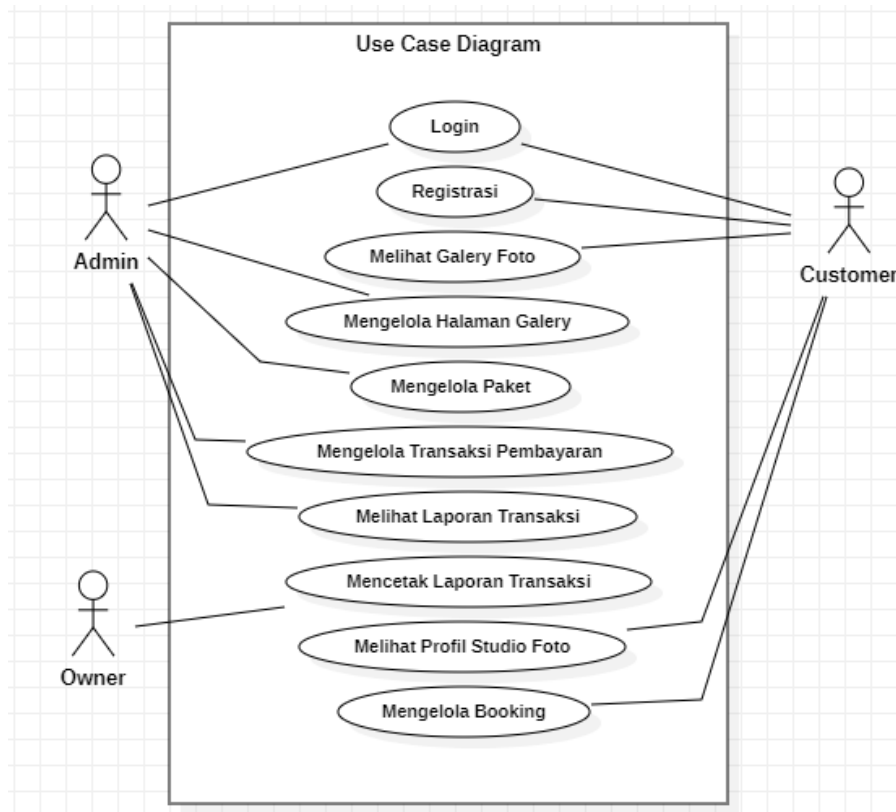
Kebutuhan fungsional pada Customer

1. Customer dapat melakukan login
2. Customer dapat melakukan registrasi akun
3. Customer dapat melihat galeri studio foto
4. Customer dapat melihat profil studio foto
5. Customer dapat mengelola booking

4.1.4 Use Case Diagram

Use case diagram menggambarkan hubungan antara fungsional dengan aktor yang terlibat dalam sistem. Use case diagram pembangunan sistem informasi pengelolaan studio foto dapat dilihat pada Gambar 4.12 berikut.





Gambar 4.12 Use Case Diagram

4.1.5 Deskripsi Tugas Aktor

Sesuai dengan *use case diagram* sebelumnya, terdapat dua aktor yang terlibat dalam sistem informasi pengelolaan studio foto berbasis *web*, yaitu admin dan customer. Masing-masing aktor memiliki beberapa tugas, deskripsi dari masing-masing tugas aktor tersebut dapat dilihat pada Tabel 4.1 berikut.

Tabel 4.1 Deskripsi Tugas Aktor

No	Aktor	Deskripsi Tugas
1.	Admin	1. Admin dapat melakukan login Admin dapat melakukan login pada aplikasi 2. Admin dapat mengelola halaman galeri Admin dapat menambah, mengubah, dan menghapus halaman galeri foto studio 3. Admin dapat mengelola paket

Tabel 4.1 Deskripsi Tugas Aktor (Lanjutan)

		Admin dapat menambah, melihat, mengubah, dan menghapus paket foto studio 4. Admin dapat mengelola transaksi pembayaran Admin dapat menerima dan menolak transaksi yang dilakukan oleh customer 5. Admin dapat melihat rekapan laporan transaksi Admin dapat melihat rekapan laporan transaksi customer
2	Pimpinan	1. Pimpinan dapat melihat laporan transaksi Pimpinan dapat melihat laporan transaksi
3	Customer	1. Customer dapat melakukan login Customer dapat melakukan login 2. Customer dapat melakukan registrasi akun Customer dapat melakukan registrasi akun 3. Customer dapat melihat galeri studio foto Customer dapat melihat galeri studio foto 4. Customer dapat melihat profil studio foto Customer dapat melihat profil studi foto 5. Customer dapat mengelola booking 6. Customer dapat menambah, melihat, mengubah, dan mencetak booking

4.1.6 Use Case Scenario

Use case scenario berfungsi untuk menjelaskan langkah -langkah *user* dalam menggunakan fungsional yang ada pada sistem. Pada bagian ini membahas beberapa skenario dari *use case* yang terdapat pada sistem informasi pengelolaan

studio foto secara online yang dibangun. Dalam bab ini hanya akan membahas mengenai use case scenario melihat galeri foto, melihat daftar paket dan booking studio foto. *Use case scenario* lainnya dijelaskan pada lampiran B.

4.1.6.1 Use Case Scenario Melihat Galeri Foto

Customer dapat melihat galeri foto melalui aplikasi web yang dibangun. Use case scenario melihat galeri foto dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2 Use scenario melihat galeri foto

Use Case	Melihat Galeri Foto
Aktor	Customer
Entry Condition	Aktor telah login
Flow of event	1. Aktor memilih menu galeri Foto 2. Sistem menampilkan halaman galery foto.
Exit Condition	Sistem menampilkan halaman galeri foto yang berisikan foto-foto dari studio foto

4.1.6.2 Use Case Scenario Melihat Daftar Paket foto

Customer dapat melihat daftar paket foto melalui aplikasi web yang dibangun. Use case scenario melihat galeri foto dapat dilihat pada Tabel 4.3 berikut.

Tabel 4.3 Use case scenario melihat daftar paket foto

Use Case	Melihat Daftar Paket Foto
Aktor	Customer
Entry Condition	Aktor telah login
Flow of event	1. Aktor memilih menu daftar paket 2. Sistem menampilkan halaman daftar paket.
Exit Condition	Sistem menampilkan halaman daftar paket yang ada di studio foto

4.1.6.3 Use Case Scenario Booking Paket Foto

Customer dapat booking paket foto melalui aplikasi web yang dibangun. Use case scenario melihat galeri foto dapat dilihat pada Tabel 4.4 berikut.

Tabel 4.4 Use case scenario booking paket foto

Use Case	Booking Paket Foto
Aktor	Customer
Entry Condition	Aktor telah login
Flow of event	1. Aktor memilih menu Booking 2. Sistem menampilkan halaman booking foto. 3. Aktor mengisi form booking yang terdiri dari tanggal sewa, alamat dan nama paket. 4. Sistem melakukan pengecekan data 5. sistem menampilkan pesan booking berhasil
Scenario Alternatif	1. Aktor menginputkan data kosong atau data yang salah 2. Sistem menampilkan pesan booking gagal
Exit Condition	Sistem menampilkan halaman riwayat booking terakhir

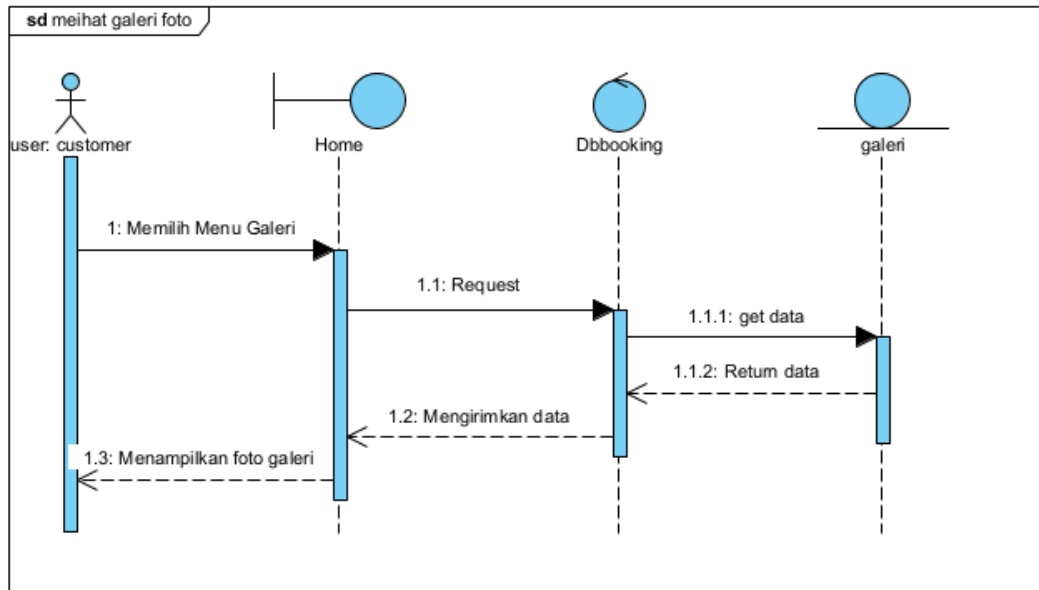
4.1.7 Sequence Diagram

Sub bab ini menjelaskan tentang sequence diagram dari beberapa proses yang dilakukan di dalam sistem. Sequence diagram mengacu pada use case diagram dan use case scenario sehingga memperjelas urutan proses yang terjadi pada sistem. Dalam sub bab ini hanya akan membahas mengenai Sequence Diagram melihat galeri foto, melihat daftar paket dan booking studio foto. Sequence Diagram lainnya dijelaskan pada lampiran B.

4.1.7.1 Sequence Diagram Melihat Galeri Foto

Proses melihat galeri foto dilakukan oleh customer melalui aplikasi web. Proses ini dimulai dengan customer yang sudah login ke aplikasi dan memilih

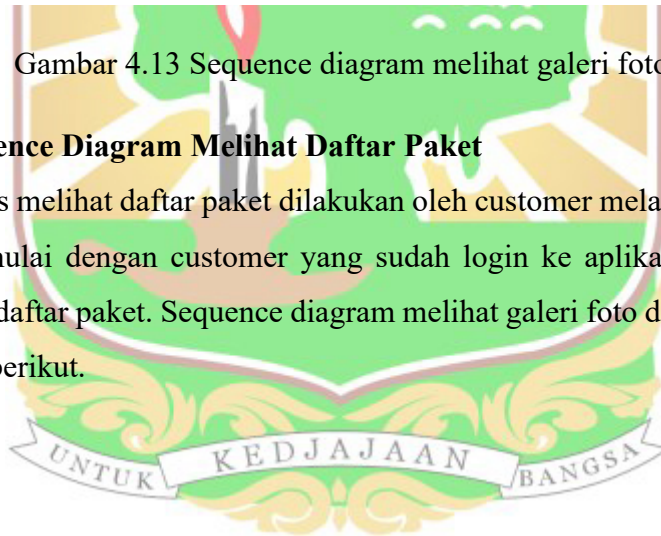
menu melihat galeri foto. Sequence diagram melihat galeri foto dapat dilihat pada Gambar 4.13 berikut.

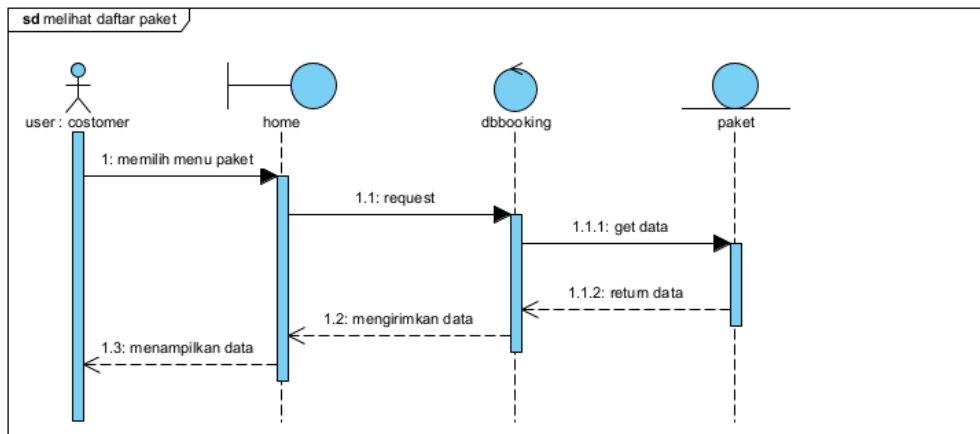


Gambar 4.13 Sequence diagram melihat galeri foto

4.1.7.2 Sequence Diagram Melihat Daftar Paket

Proses melihat daftar paket dilakukan oleh customer melalui aplikasi web. Proses ini dimulai dengan customer yang sudah login ke aplikasi dan memilih menu melihat daftar paket. Sequence diagram melihat galeri foto dapat dilihat pada Gambar 4.14 berikut.

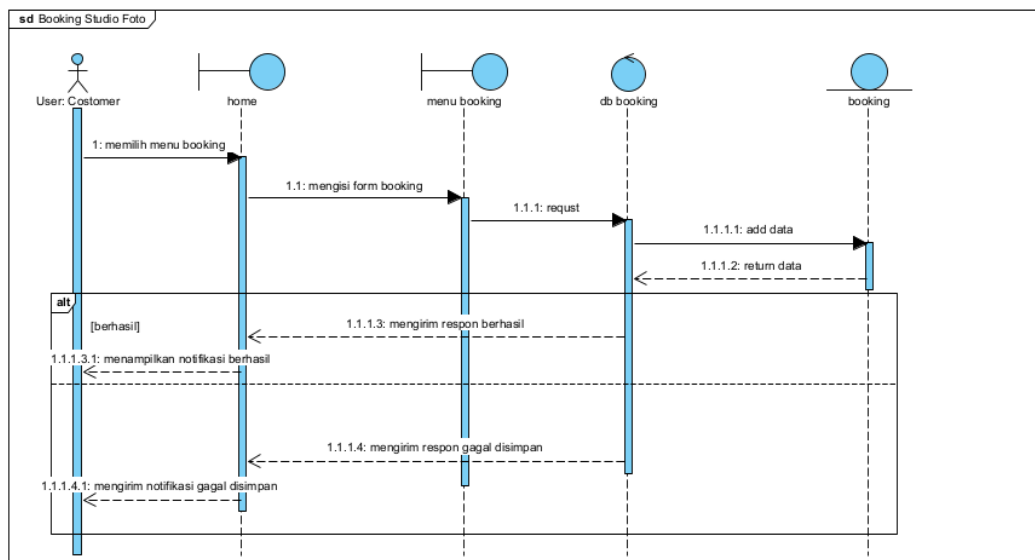




Gambar 4.14 Sequence diagram melihat daftar paket

4.1.7.3 Sequence Diagram Booking Studio Foto

Proses booking studio foto dilakukan oleh customer melalui aplikasi web. Proses ini dimulai dengan customer yang sudah login ke aplikasi dan memilih menu booking studio. Sequence diagram booking studio foto dapat dilihat pada Gambar 4.15 berikut.



Gambar 4.15 Sequence diagram booking studio foto

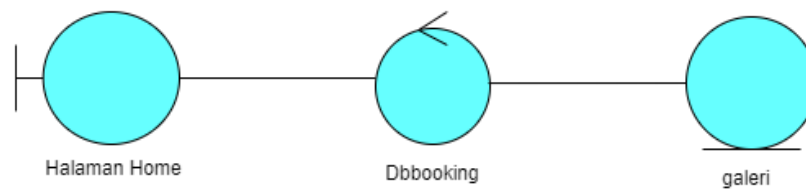
4.1.8 Class Analysis Aplikasi

Class analysis berfungsi untuk menggambarkan semua kelas yang terlibat di dalam setiap proses dan dibuat berdasarkan sequence diagram sistem yang telah

dirancang. *Class analysis* yang ditampilkan pada sub bab ini ada tiga, yaitu *class analysis* melihat galeri foto, melihat daftar paket, serta booking studio foto. Untuk *class analysis* lainnya dipaparkan pada bagian lampiran C.

4.1.8.1 *Class Analysis* melihat galeri foto

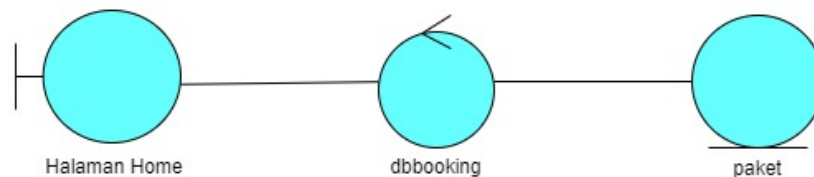
Class analysis untuk melihat galeri foto dapat dilihat pada Gambar 4.16 berikut ini.



Gambar 4.16 *Class analysis* pada melihat galeri foto

4.1.8.2 *Class Analysis* Melihat Daftar Paket

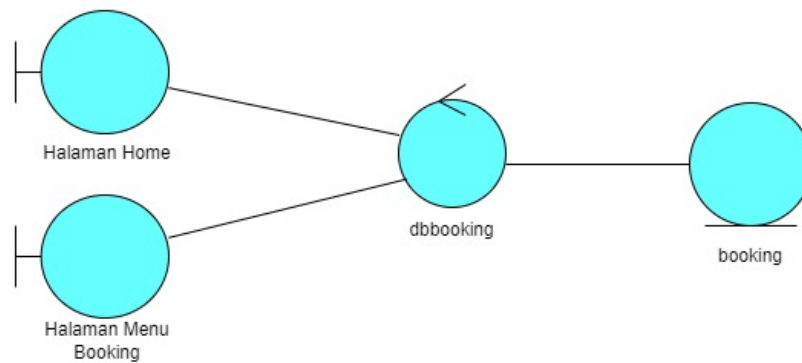
Class analysis untuk melihat daftar paket dapat dilihat pada Gambar 4.17 berikut ini.



Gambar 4.17 *Class analysis* melihat daftar paket

4.1.8.3 *Class Analysis* Booking

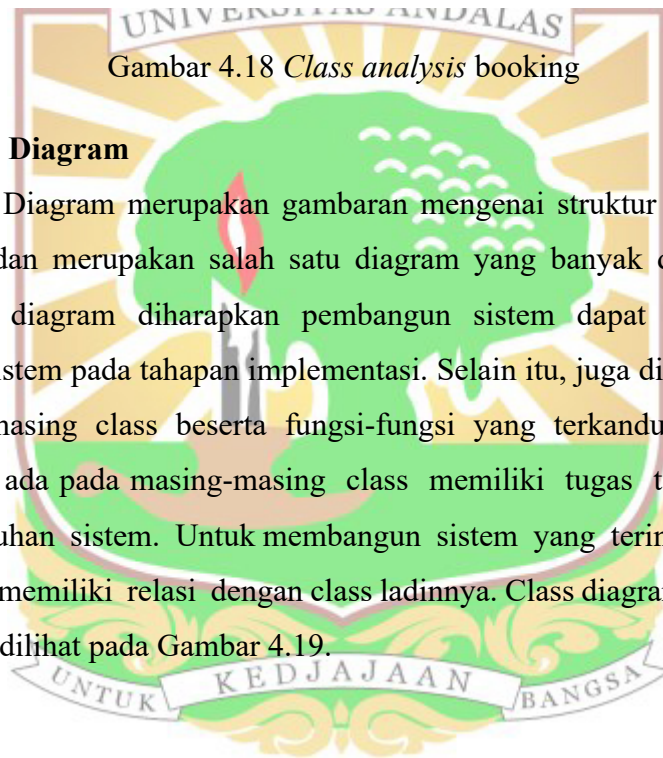
Class analysis untuk melihat proses booking dapat dilihat pada Gambar 4.18 berikut ini.

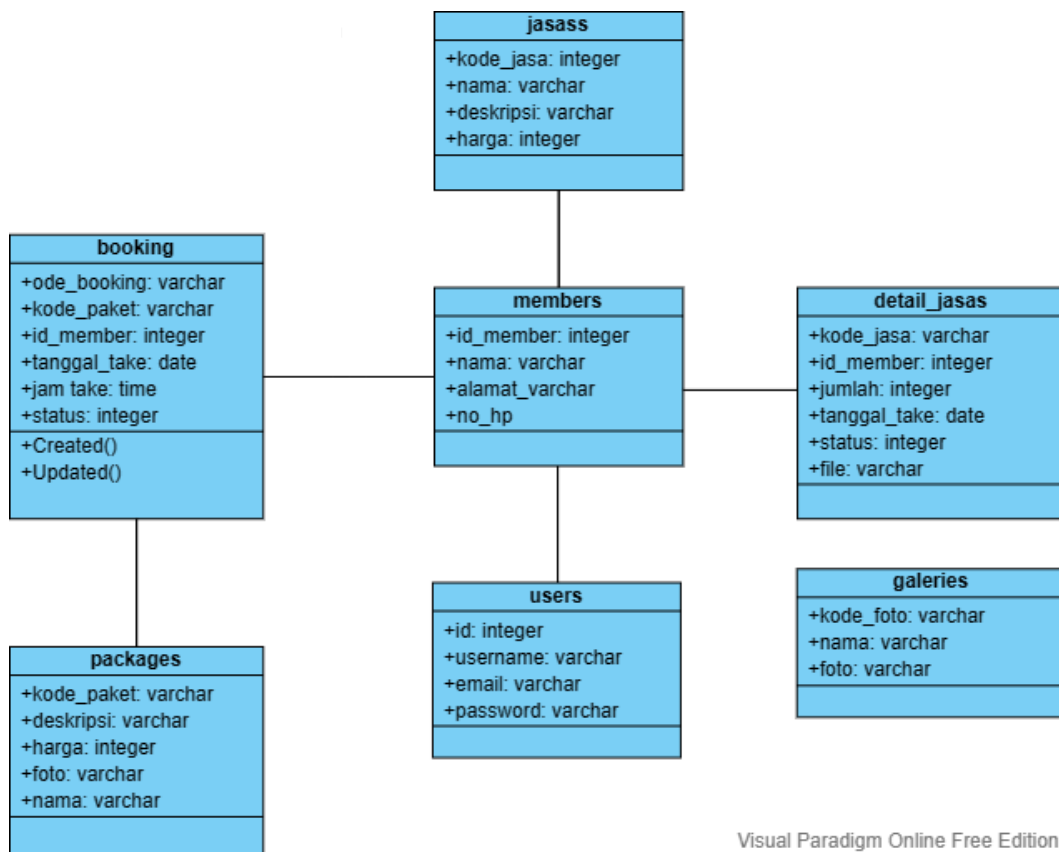


Gambar 4.18 *Class analysis booking*

4.1.9 Class Diagram

Class Diagram merupakan gambaran mengenai struktur kelas-kelas dari suatu sistem dan merupakan salah satu diagram yang banyak dipakai. Dengan adanya class diagram diharapkan pembangun sistem dapat terbantu dalam membangun sistem pada tahapan implementasi. Selain itu, juga dipaparkan atribut dari masing-masing class beserta fungsi-fungsi yang terkandung didalamnya. Fungsi yang ada pada masing-masing class memiliki tugas tersendiri sesuai dengan kebutuhan sistem. Untuk membangun sistem yang terintegrasi masing-masing class memiliki relasi dengan class lainnya. Class diagram pembangunan aplikasi dapat dilihat pada Gambar 4.19.





Gambar 4.19 *Class analysis* booking

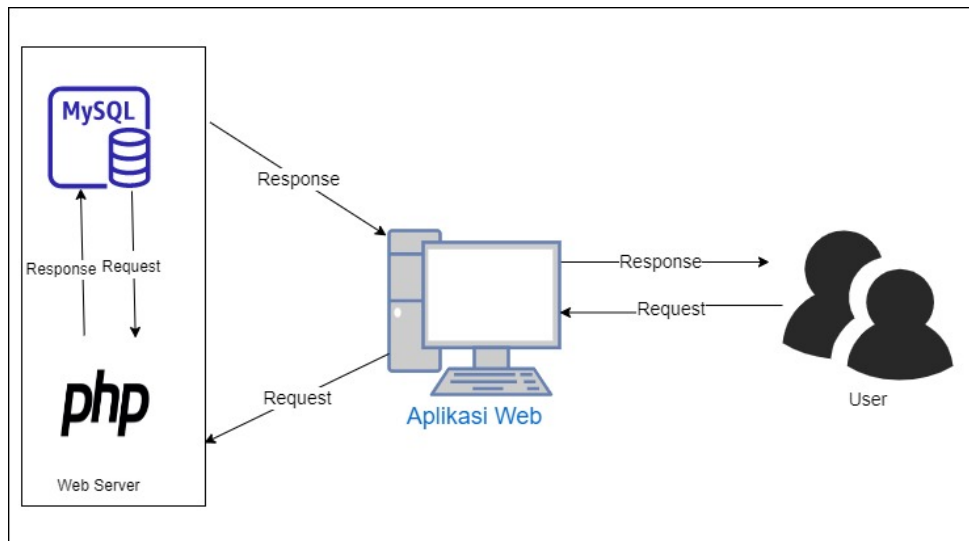
4.2 Perancangan Sistem

Berdasarkan tahapan analisis proses bisnis yang sedang berjalan, alur sistem dan kebutuhan fungsional sistem yang dibangun maka diperoleh hasil yang menjadi dasar dan tolok ukur untuk melakukan perancangan sistem. Perancangan sistem ini meliputi arsitektur aplikasi, perancangan basis data, struktur tabel *database* dan perancangan antarmuka.

4.2.1 Arsitektur Aplikasi

Perancangan arsitektur aplikasi adalah sebuah tahap penting dalam pembangunan. Arsitektur teknologi merupakan sebuah blue print dari elemen teknologi informasi dan sistem informasi bekerja sama dalam satu kesatuan. Arsitektur teknologi informasi berupa komponen-komponen dasar dalam mendukung sistem aplikasi perangkat lunak. Rancangan arsitektur aplikasi

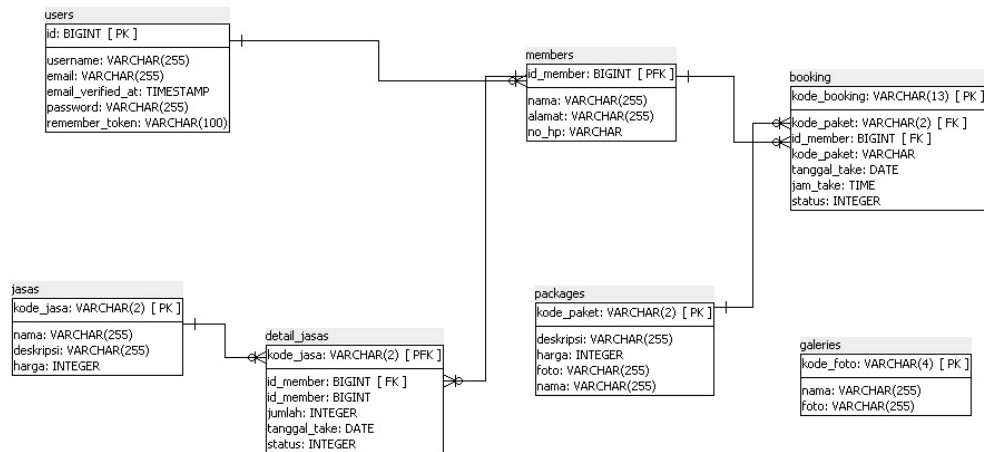
Pembangunan Sistem Informasi Pengelolaan Studio Foto Berbasis Web Pada Mutiara Studio Photo Padang seperti Gambar 4.20 berikut.



Gambar 4.20 Arsitektur teknologi aplikasi

4.2.2 Pearancangan Basis Data

Perancangan database dilakukan dengan Menyusun struktur database berdasarkan entitas yang digunakan dan hubungannya dengan entitas-entitas lain. Entitas beserta hubungannya digambarkan dalam sebuah ERD (*Entity Relationship Diagram*) yang mempresentasikan model data system. Database yang dirancang terdiri dari 7 tabel yaitu, table user, table jasa, table detail_jasa, table member, table paket, table booking dan table gallery. dari table-table yang telah dirancang tersebut, maka dapat digambarkan ERD dari perancangan aplikasi pengelolaan pada studio foto seperti Gambar 4.21 berikut.



Gambar 4.21 Perancangan basis data

4.2.3 Struktur Tabel Basis Data

Struktur table dan basis data merupakan representasi table pada masing-masing relasi beserta status atributnya. Status tersebut berupa *primary key*, *foreign key*, tipe data, nama atribut, dan nama table. Berikut merupakan uraian struktur tabel basis data dapat dilihat pada Tabel 4.5 - Tabel 4.7.

Tabel 4.5 Tabel packages

Nama Atribut	Tipe Data	Keterangan
Kode_paket	Varchar	PK
Deskripsi	Varchar(255)	
Harga	Integer	
Foto	Varchar(255)	
Nama	Varchar(255)	

Tabel 4.6 Tabel members

Nama Atribut	Tipe Data	Keterangan
Id_members	Bigint	PFK
Nama	Varchar (255)	
Alamat	Varchar (255)	
No_hp	Varchar	

Tabel 4.7 Tabel booking

Nama Atribut	Tipe Data	Keterangan
Kode_booking	Varchar (13)	PK
Kode_paket	Varchar (2)	FK
Id_member	Bigint	FK
Kode_paket	Varchar	
Tanggal_take	Date	
Jam_take	Date	
Status	Integer	

4.2.4 Perancangan Antarmuka

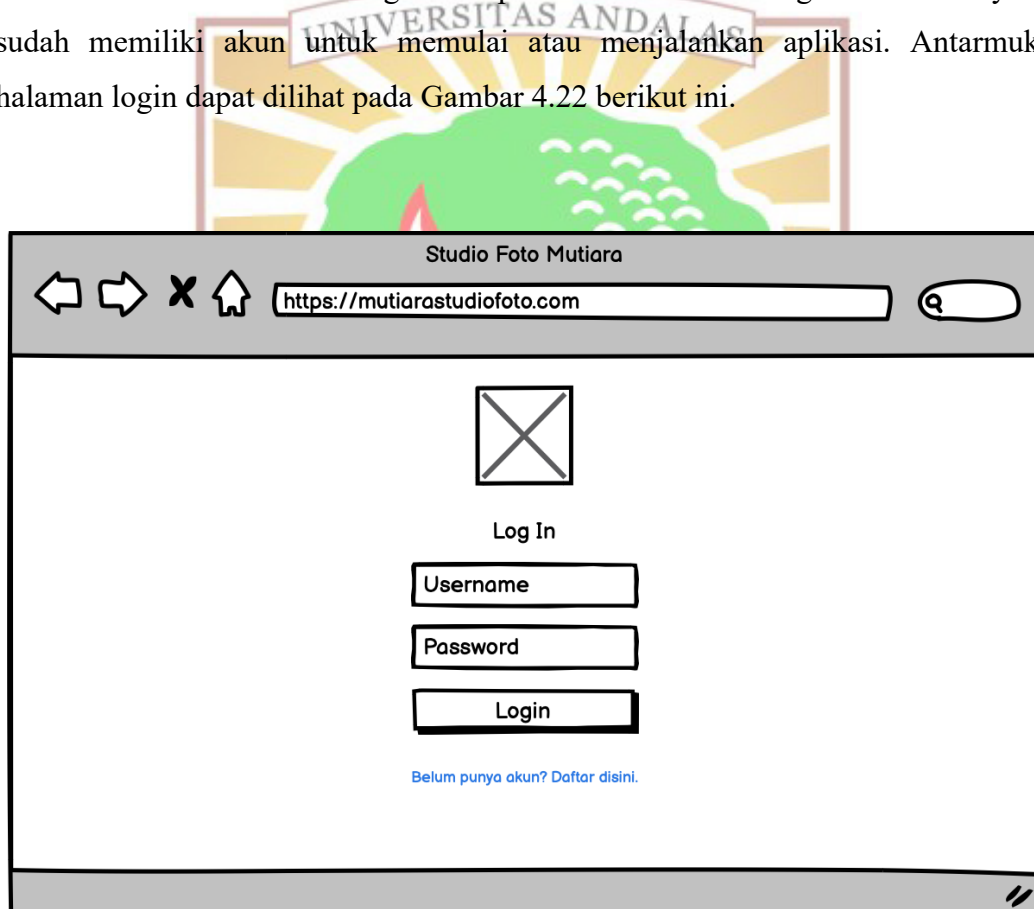
User interface digunakan untuk berinteraksi antara program dan pengguna. user interface memiliki fungsi untuk menghubungkan user dengan sistem informasi sehingga dapat digunakan nantinya. user interface dapat berupa Graphical User Interface (GUI) atau Command Line Interface (CLI). *User interface* bertujuan untuk menjadikan sistem yang dirancang bersifat *user friendly*. *User friendly* adalah karakter dari perangkat lunak yang mudah dioperasikan sehingga membuat

pengguna nyaman dan mudah dalam mengoperasikan perangkat lunak tersebut. Pada pembangunan aplikasi Pembangunan Sistem Informasi Pengelolaan Studio Foto Berbasis Web Pada Mutiara Studio Photo Padang.

Antarmuka aplikasi web yang dijelaskan pada bagian ini adalah halaman antarmuka halaman lihat galeri, halaman lihat daftar paket dan halaman booking studio foto. Untuk gambaran perancangan antarmuka aplikasi dapat dilihat pada penjabaran berikut.

4.2.4.1 Antarmuka Halaman Login

Antarmuka halaman login merupakan halaman awal bagi semua aktor yang sudah memiliki akun untuk memulai atau menjalankan aplikasi. Antarmuka halaman login dapat dilihat pada Gambar 4.22 berikut ini.



UNIVERSITAS ANDALAS

Studio Foto Mutiara

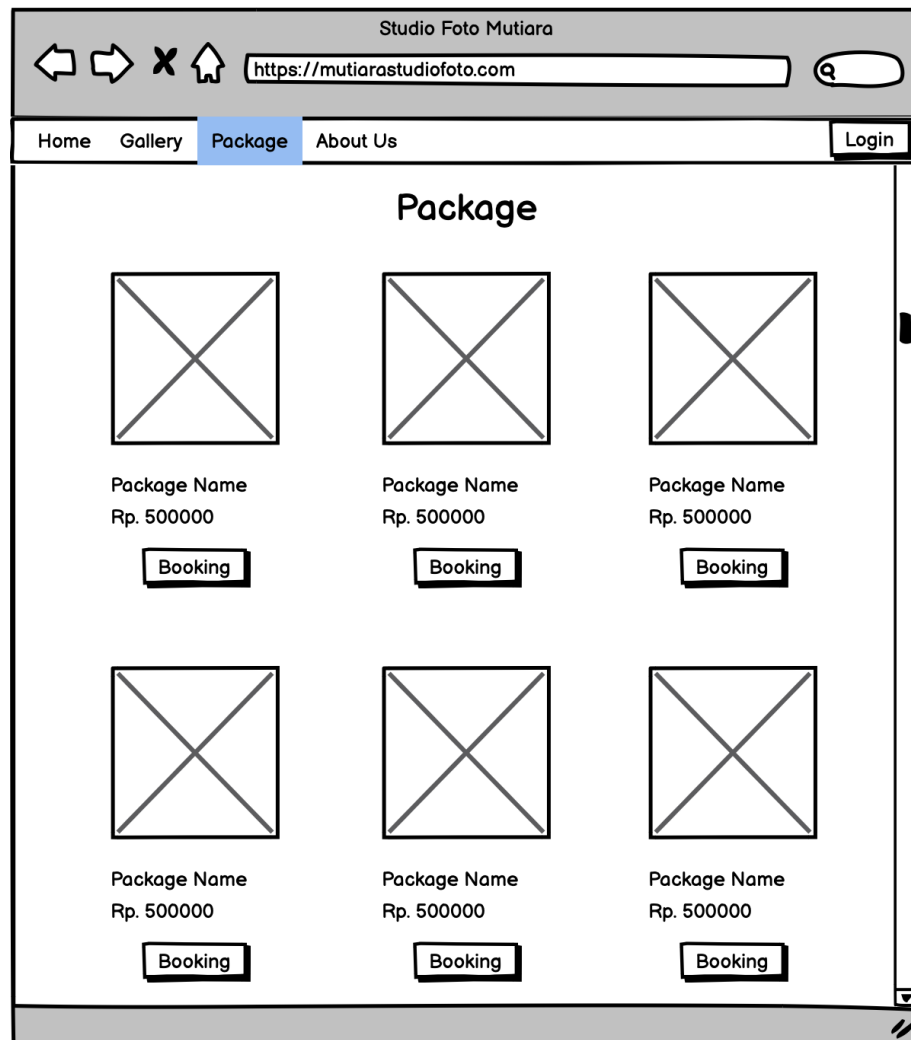
<https://mutiarastudiofoto.com>

[Belum punya akun? Daftar disini.](#)

Gambar 4.22 Antarmuka halaman login

4.2.4.2 Antarmuka Halaman Lihat Daftar Paket

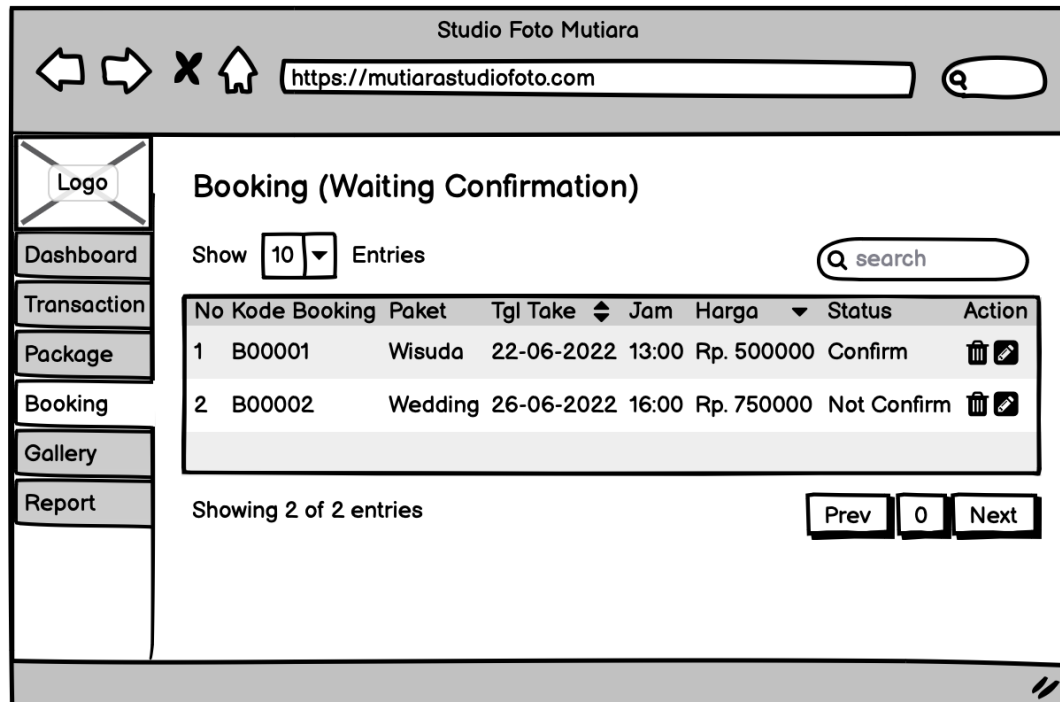
Antarmuka halaman lihat daftar paket merupakan halaman untuk pemilihan paket yang akan di booking oleh customer. Antarmuka halaman lihat daftar paket dapat dilihat pada Gambar 4.23 berikut ini



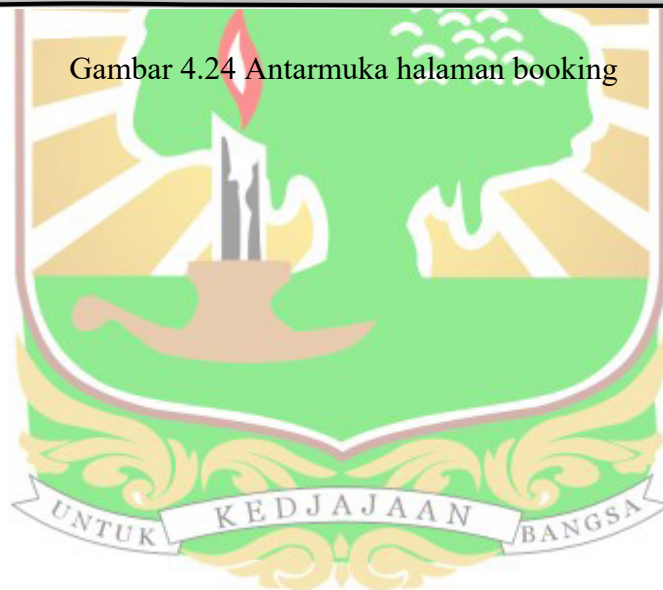
Gambar 4.23 Antarmuka halaman daftar paket

4.2.4.3 Antarmuka Halaman Booking

Antarmuka halaman booking merupakan halaman yang diakses oleh admin untuk mengelola data paket yang sudah di booking oleh customer. Antarmuka halaman booking data dilihat pada Gambar 4.24 berikut ini



Gambar 4.24 Antarmuka halaman booking



BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini dijelaskan tentang implementasi sistem sesuai dengan perancangan yang dilakukan pada bab sebelumnya dan pengujian kesesuaian sistem yang dibangun dengan alur proses pengelolaan studio foto yang diusulkan pada Studio Foto Mutiara di Kota Padang. Tahap implementasi dan tahap pengujian sistem dilakukan pada aplikasi *web*.

5.1 Implementasi Sistem

Aplikasi pengelolaan studio foto berbasis web pada Studio Foto ini difungsikan sebagai sistem yang mengelola dan menyajikan data dan informasi yang berkaitan dengan pengelolaan data pada studio foto. Pada sistem ini terdapat dua aktor yang terlibat dan fungsional yang dapat dijalankan. Fungsional tersebut mewakili setiap tugas dari *user* dan diharapkan dapat memberikan efisiensi dan efektifitas terhadap semua proses pada sistem.

Implementasi aplikasi web dibangun menggunakan bahasa pemrograman PHP (Pearl Hypertext Preprocessor) dan framework Laravel. Framework Laravel yang digunakan adalah Laravel versi 5.7.28 dan web server apache XAMPP versi 3.2.3. Web server berkomunikasi dengan perangkat lunak yang disebut dengan *middleware* dan *middleware* yang berhubungan dengan database (Habibie, Purnama, & Triyono, 2014). Database yang digunakan sebagai tempat penyimpanan data adalah database MySQL.

Implementasi sistem ini dilakukan dengan menggunakan perangkat keras komputer dengan spesifikasi sebagai berikut:

1. Komputer dengan *processor* Intel core i5-4010U CPU 1.70 GHz.
2. Memori Komputer (RAM) 4 GB.
3. *Harddisk* Komputer dengan kapasitas 500 GB.

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem ini adalah sebagai berikut:

1. Sistem Operasi windows 10 pro
2. *Web browser* yang digunakan adalah google chrome versi 73.

Berikut ini spesifikasi *system requirement* untuk implementasi sistem:

1. Sistem operasi Windows 7, 8, 8.1, 10.
2. *Web server* dengan modul PHP 7.3.2 atau setelahnya
3. *Web browser* yang mendukung HTML 5

5.1.1 Pengkodean Program

Pada bagian pengkodean program ini dijelaskan kode program yang dibuat berdasarkan arsitektur aplikasi yang digunakan. Bagian *view*, *model*, dan *controller* dijelaskan secara terpisah berdasarkan fungsi dan kegunaan masing-masing. Berikut dijelaskan fungsi dan kegunaan masing-masing bagian pada aplikasi. Untuk penjelasan kode program lebih lanjut dapat dilihat pada lampiran G.

5.1.1.1 Kode Program *Routing* Aplikasi

Routing merupakan bagian yang mengatur arah perpindahan permintaan (*request*) dari *user*. *Route* dapat meng-*handle* semua perintah yang dideklarasikan dan mengirimkannya kepada *controller* serta *method* yang sesuai. Kode program *routing* aplikasi dapat dilihat pada Gambar 5.1.

```
<?php
use App\Model\Galery;

use App\Model\Package;

use App\Tracker;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Route;
```

```

Route::get('/', function () {

    //number of user connected or viewed

    Tracker::firstOrCreate([

        'ip'    => $_SERVER['REMOTE_ADDR']],

        ['ip'    => $_SERVER['REMOTE_ADDR'],

        'current_date' => date('Y-m-d')])->save();

    $packages = Package::all();

    $galery = Galery::all();

    return view('welcome',compact('packages','galery'));

})->name('welcome');

Route::group(['middleware' => ['role:admin']], function () {

    Route::resource('permissions', 'Admin\PermissionsController');

    Route::resource('roles', 'Admin\RolesController');

    Route::resource('users', 'Admin\UsersController');

    Route::get('login-activities',[

        'as' => 'login-activities',

        'uses' => 'Admin\UsersController@indexLoginLogs'

    ]);

    Route::resource('packages','Admin\PackageController');

    Route::resource('galleries', 'Admin\GaleryController');

    Route::resource('services', 'Admin\JasaController');

    Route::resource('transaction', 'Admin\TransactionController');

    Route::resource('report', 'Admin\ReportController');

});

Route::group(['middleware' => ['role:member']], function () {

    Route::resource('booking', 'Member\BookingController');

```

```

Route::resource('package', 'Member\PackageController');

Route::resource('service', 'Member\JasaController');

});

Route::group(['middleware' => ['auth']], function () {

    Route::resource('profile', 'Users\ProfileController');

});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

```

Gambar 5.1 Kode program *routing* aplikasi

5.1.1.2 Kode Program *Controller* Galeri Foto

Controller merupakan bagian yang mengatur seluruh alur dan *method* pada aplikasi sistem informasi dengan menggunakan arsitektur MVC. Salah satu *controller* yang dibangun adalah *GaleryController*. Kode program *GaleryController* dapat dilihat pada Gambar 5.2.

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Model\Galery;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class GaleryController extends Controller
{
    public function index()
    {
        $galleries = Galery::all();
        return view('admin.galery.index', compact('galleries'));
    }

    public function create()
    {
        return view('admin.galery.create');
    }

    public function store(Request $request)
    {
        // dd($request->all());
        $this->validate($request, [
            'kode_foto' => 'required|max:4|unique:galleries',
            'nama' => 'required',
            'foto' =>

```

```

'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
]);

$fotopath = null;
if ($request->hasFile('foto')) {
    $fotoname = $request->file('foto')->hashName();
    $fotopath = 'foto/'.$fotoname;
    Storage::disk('public')->put($fotopath,
file_get_contents($request->foto));
}
$galery = new Galery;
$galery->kode_foto = $request->kode_foto;
$galery->nama = $request->nama;
$galery->foto = $fotopath;
$galery->save();
return redirect()->route('galleries.index')-
>with('success','Data berhasil ditambahkan');
}

public function edit($id)
{
    $galery = Galery::find($id);
    return view('admin.galery.edit',compact('galery'));
}

public function update(Request $request, $id)
{
    $this->validate($request, [
        'nama' => 'required',
    ]);

    $galery = Galery::find($id);
    $galery->nama = $request->nama;
    if ($request->hasFile('foto')) {
        Storage::disk('public')->delete($galery->foto);
        $fotoname = $request->file('foto')->hashName();
        $fotopath = 'foto/'.$fotoname;
        Storage::disk('public')->put($fotopath,
file_get_contents($request->foto));
        $galery->foto = $fotopath;
    }
    $galery->save();
    return redirect()->route('galleries.index')-
>with('success','Data berhasil diubah');
}

public function destroy($id)
{
    $galery = Galery::find($id);
    Storage::disk('public')->delete($galery->foto);
    $galery->delete();
    return redirect()->route('galleries.index')-
>with('success','Data berhasil dihapus');
}
}

```

Gambar 5.2 Kode program *GaleryController*

5.1.1.3 Kode Program *Model Member*

Model merupakan bagian yang berfungsi memanggil data dari *database* serta mengirimkan data ke *controller* dengan menggunakan sintax SQL (*StructuredQuery Language*). Salah satu *model* yang dibangun adalah

model member untuk memanggil entitas data member pada *database*. Kode program *model* pendaftaran dapat dilihat pada Gambar 5.3.

```
<?php

namespace App\Model;

use Illuminate\Database\Eloquent\Model;

class Member extends Model
{
    protected $table = 'members';
    protected $fillable = [ 'nama', 'alamat', 'no_hp' ];

    public function user()
    {
        return $this->belongsTo(User::class, 'id');
    }
}
```

Gambar 5.3 Kode program *model*

5.1.1.4 Kode Program *View* Transaksi

View merupakan antar muka aplikasi yang berinteraksi langsung dengan *user*. Data dari *database* dan perintah dari *controller* kemudian ditampilkan pada *view*. Salah satu *view* yang dibangun adalah *view* transaksi yang berfungsi untuk menampilkan halaman “transaksi”. Kode program *view* daftar dapat dilihat pada Gambar 5.4.

```
@extends('layouts.app')

@section('content')

<div class="row">

<div class="col-sm-12">

<div class="card">
```

```

<div class="card-header">

<h3 class="pull-left">Transaksi</h3>

<h3 class="pull-right">

<a class="btn btn-sm btn-primary pull-right" href="{!!
route('transaksi.create') !!}">Add New</a>

</h3>
</div>

<div class="card-body">

v class="table-stats order-table ov-h">@include('transaksi.table')

</div>

</div>

</div>

</div>

</div>

</div>
@endsection

```

Gambar 5.4 Kode Program *View Transaksi*

5.1.2 Implementasi Antar Muka Aplikasi *Web*

Aplikasi *web* yang dibangun dapat diakses oleh *user* menggunakan *web browser*. Berikut dijelaskan tampilan aplikasi *web* pembangunan aplikasi pengelolaan studio foto berbasis web pada Studio Foto Mutiara sesuai dengan implementasi sistem. Untuk penjelasan antar muka lain dapat dilihat pada lampiran H.

5.1.2.1 Halaman *Login*

Halaman *login* merupakan halaman *website* yang pertama kali ditampilkanketika *user* membuka aplikasi *web*. Hal ini dimaksudkan agar aplikasi ini hanya dapat digunakan oleh *user* yang terlibat dalam aplikasi ini dan memiliki hakakses. Aplikasi ini memiliki satu level hak akses *user*, yaitu admin. Berikut tampilan halaman *login* dapat dilihat pada Gambar 5.5.

Gambar 5.5 Implementasi tampilan halaman login

Berdasarkan Gambar 5.5 *user* diharuskan untuk *login* agar dapat masuk kedalam aplikasi dengan men-input-kan *email* dan *password* yang telah ditentukan pada *database* lalu menekan tombol “LOGIN”. Jika *user* berhasil *login* maka sistem menampilkan halaman utama aplikasi.

5.1.2.2 Halaman Admin

Halaman admin diberikan kepada seorang *user* dengan level hak akses sebagai admin. Admin memiliki wewenang dalam mengelola data *user*, data paket, data galeri, data jasa, data transaksi dan data laporan. Tampilan halaman admin dapat dilihat pada Gambar 5.6.

Gambar 5.6 Implementasi tampilan halaman admin

Berdasarkan Gambar 5.6 dapat dijelaskan bahwa halaman admin memiliki enam menu utama dan satu menu *logout* yang diuraikan sebagai berikut:

1. Menu Beranda

Menu beranda merupakan halaman utama yang menampilkan informasi data pengunjung pada hari ini.

2. Menu *Package*

Menu *package* berfungsi sebagai menu untuk mengelola data paket yang ada pada studio foto. *User* dapat menambah data, merubah data dan menghapus data terpilih.

3. Menu *Galery*

Menu *galery* berfungsi sebagai menu untuk mengelola data galeri. *User* dapat menambah data, merubah data dan menghapus data terpilih.

4. Menu *Service*

Menu *service* berfungsi sebagai menu untuk mengelola data *service* yang ada pada studio foto. *User* dapat menambah, merubah dan menghapus data.

5. Menu Transaksi

Menu transaksi menampilkan halaman transaksi pada studio foto

6. Menu Laporan

Menu laporan berfungsi untuk menampilkan semua transaksi yang sudah berstatus selesai dalam kurun waktu tertentu.

7. Menu *Logout*

Menu *logout* berfungsi untuk menutup halaman utama dan menampilkan halaman *login*.

5.2 Pengujian Sistem

Pada tahapan ini dilakukan pengujian pada semua bagian aplikasi yang telah dibangun. Proses ini dilakukan agar aplikasi yang dibangun berjalan sesuai dengan perancangan sistem. Pengujian ini dilakukan dengan

menggunakan metode *black box testing*, yaitu metode pengujian yang berfokus pada pemeriksaan ketersediaan fungsional yang sudah dirancang pada aplikasi.

Dengan menggunakan metode ini diketahui apakah fungsional aplikasi sudah berjalan sebagaimana mestinya dan sesuai dengan harapan atau tidak. Kelebihan metode ini dibandingkan metode *white box testing* adalah pada metode *black box testing* tidak diperlukan pengetahuan khusus tentang kode program aplikasi, sedangkan pada metode *white box testin* penguji diharuskan untuk memeriksa komponen internal kode program aplikasi.

5.2.1 Fokus Pengujian

Fokus pengujian aplikasi ini menggunakan data uji berdasarkan data yang telah didapatkan dari aplikasi *web*. Pada pengujian ini terdapat dua belas item uji yang terdiri dari tujuh item uji aplikasi *web* dan lima item uji aplikasi *mobile*. Fokus Pengujian dapat dilihat pada Tabel 5.1 berikut ini.

Tabel 5.1 Fokus Pengujian

No	Item Uji	Detail Pengujian
1.	Login	Verifikasi data login
2.	Mengelola halaman galeri	Lihat, tambah, hapus, edit
3.	Mengelola paket	Lihat, tambah, hapus, edit
4.	Mengelola transaksi pembayaran	Lihat, tambah, edit, hapus
5.	Mengelola laporan transaksi	Lihat
6.	Booking studio foto	Mmilih jadwal, booking

5.2.2 Kasus Hasil Pengujian

Bagian ini membahas tentang kasus-kasus serta hasil dari pengujian sistem. Pengujian yang dilakukan berpedoman pada fokus pengujian yang telah ditentukan sebelumnya. Pengujian ini dilakukan berdasarkan fungsional dengan

memperhatikan masukan ke sistem dan keluaran yang dihasilkan sistem. Pada bagian ini dijelaskan tiga kasus pengujian, yaitu pengujian mengelola data pasien poliklinik mata (ubah data), pengujian mengelola status rawat pasien (ubah status) dan pengujian mengelola ruangan (menambah ruangan).

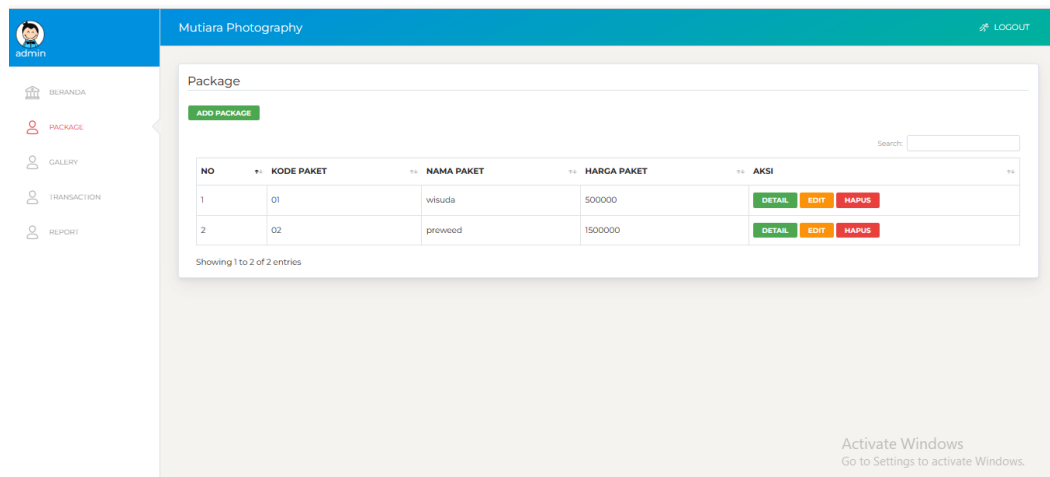
5.2.2.1 Mengelola Paket

Pada pengujian ini dilakukan ubah data pasien. Hasil pengujian mengubah data pasien dapat dilihat pada Tabel 5.2.

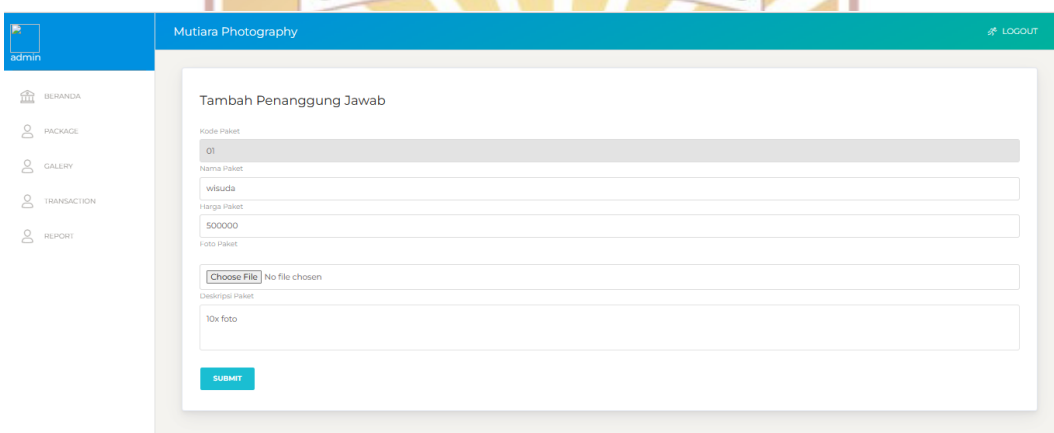
Tabel 5.2 Mengubah Data Pasien (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman mengelola paket
Yang diharapkan	Sistem menampilkan seluruh daftar paket yang tersedia, dapat mengubah data paket
Pengamatan	Menampilkan data paket dan mengubah data paket serta menekan tombol “submit”
Hasil	Sesuai

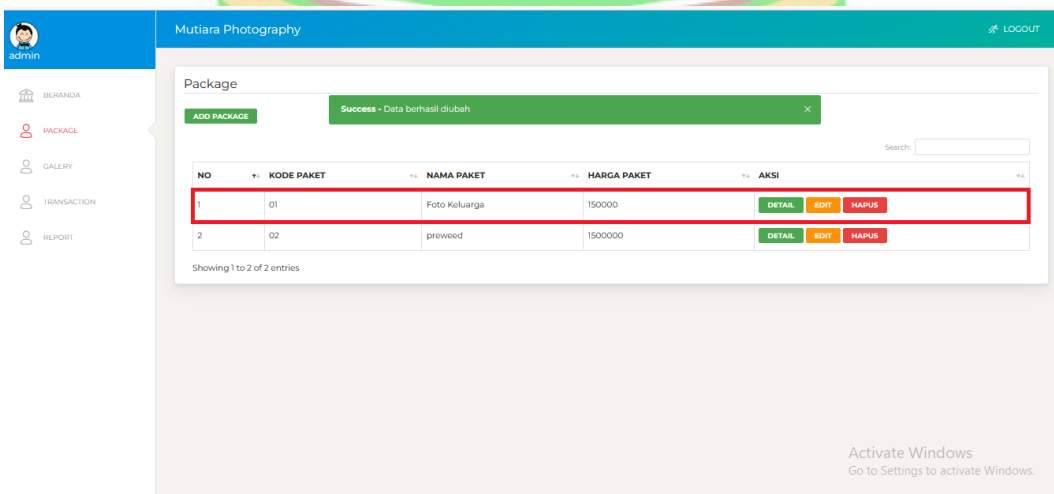
Pengujian aplikasi dilakukan dengan memilih dan menginput data paket pada halaman paket dan menekan tombol “Submit”. Jika berhasil maka sistem menampilkan halaman paket kembali dan menampilkan list paket yang tersedia. Hasil pengujian mengelola paket (benar) dapat dilihat pada Gambar 5.7 - Gambar 5.9.



Gambar 5.7 Membuka Daftar Paket



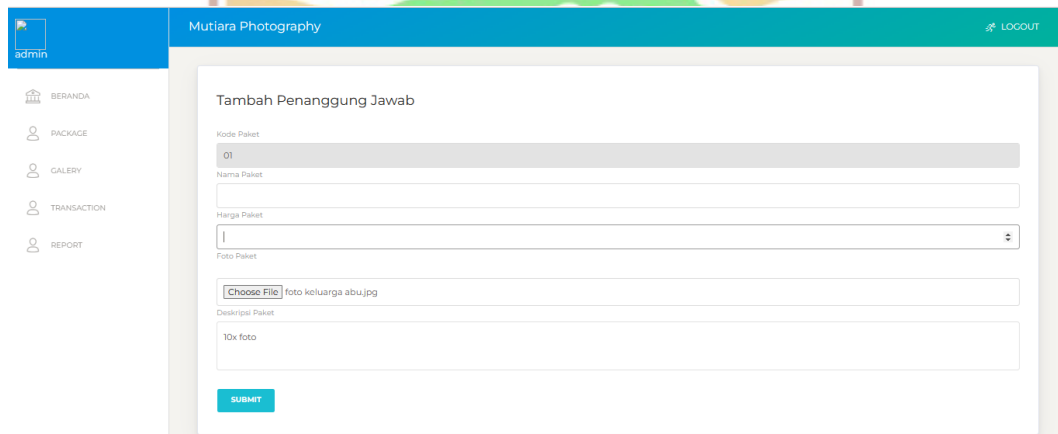
Gambar 5.8 Mengubah Data Paket



Gambar 5.9 Data Paket Telah Berubah (Benar)

Tabel 5.3 Mengubah Data Pasien (Benar)

Kasus dan Hasil Uji (Salah)	
Data masukan	Data nama paket dan harga paket salah atau kosong
Yang diharapkan	Sistem menampilkan halaman edit paket dan notifikasi “Please fill out this field”
Pengamatan	Notifikasi “Please fill out this field” ditampilkan di halaman edit paket
Hasil	Sesuai



Tambah Penanggung Jawab

Kode Paket
01

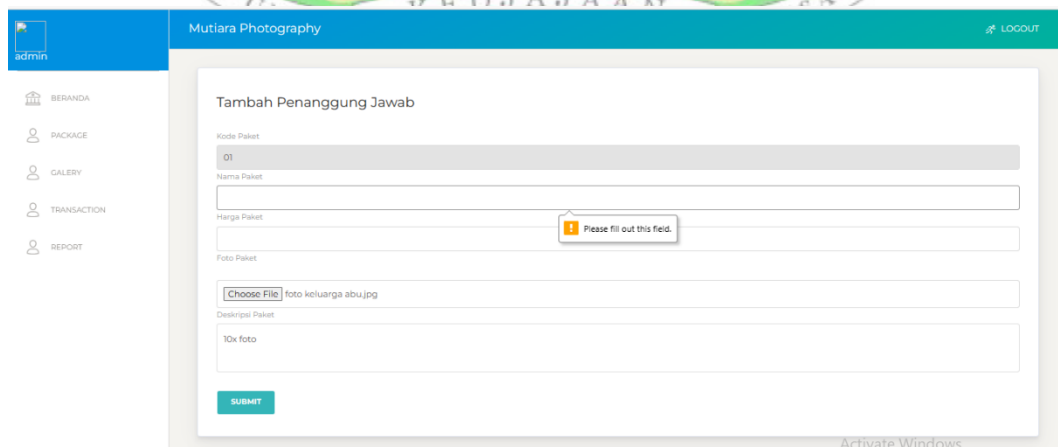
Nama Paket

Harga Paket

Foto Paket
foto keluarga abu.jpg

Deskripsi Paket
10x foto

SUBMIT



Tambah Penanggung Jawab

Kode Paket
01

Nama Paket

Harga Paket
Please fill out this field.

Foto Paket
foto keluarga abu.jpg

Deskripsi Paket
10x foto

SUBMIT

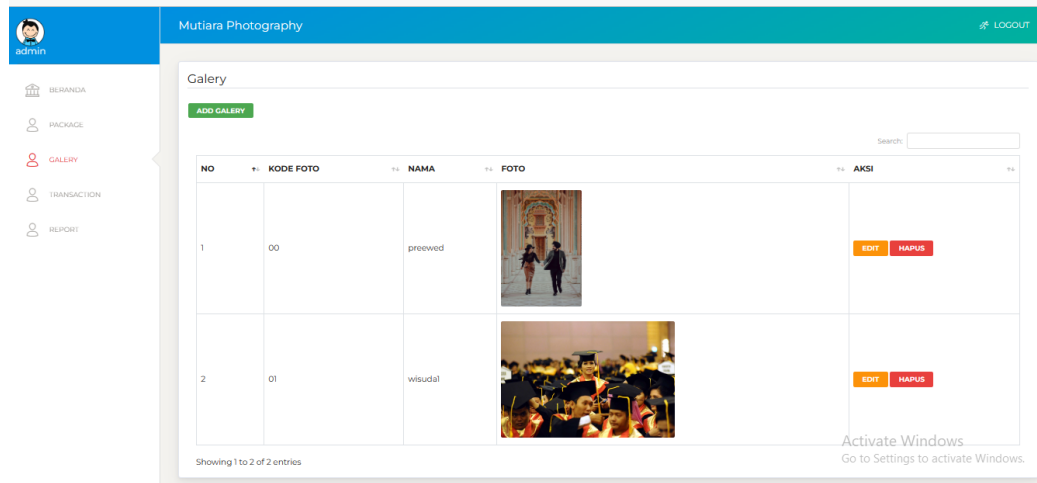
5.2.2.2 Pengujian Mengubah Galery

Pada pengujian ini dilakukan perubah data pada halaman galeri. Hasil pengujian perubahan halaman galeri dapat dilihat pada Tabel 5.4.

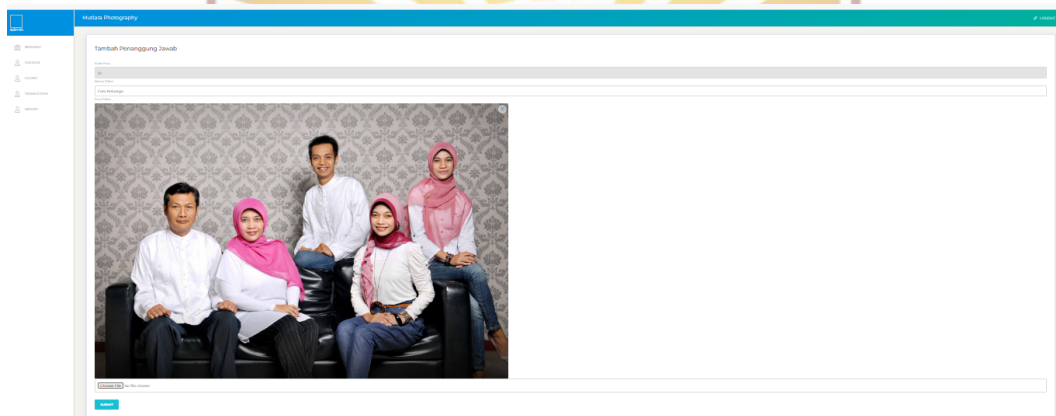
Tabel 5.4 Mengubah galery (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman galery
Yang diharapkan	Sistem menampilkan seluruh daftar galery yang tersedia, dapat mengubah gambar pada halaman gallery.
Pengamatan	Menampilkan gambar dan mengubah gambar serta menekan tombol “submit”
Hasil	Sesuai

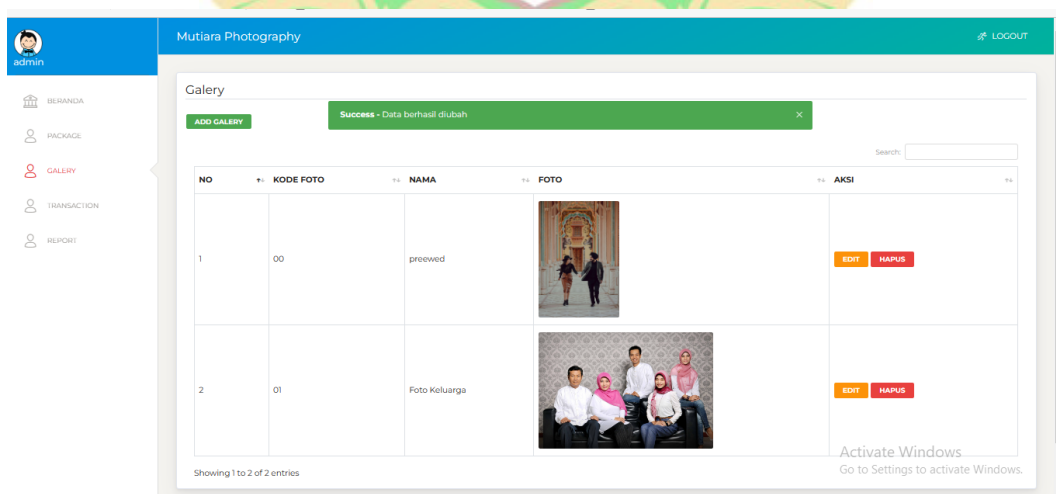
Pengujian aplikasi dilakukan dengan memilih salah satu data gambar yang ada pada halaman gallery, selanjutnya memilih tombol edit pada kolom aksi, maka akan tampil data gambar, kemudian mengisi data gambar dan mengubah file gambar, dan menekan tombol submit. Jika berhasil maka sistem menampilkan data gambar yang telah diperbarui. Hasil pengujian mengelola status rawat pasien (benar) dapat dilihat pada Gambar 5.10 - Gambar 5.12.



Gambar 5.10 Membuka Data Galeri



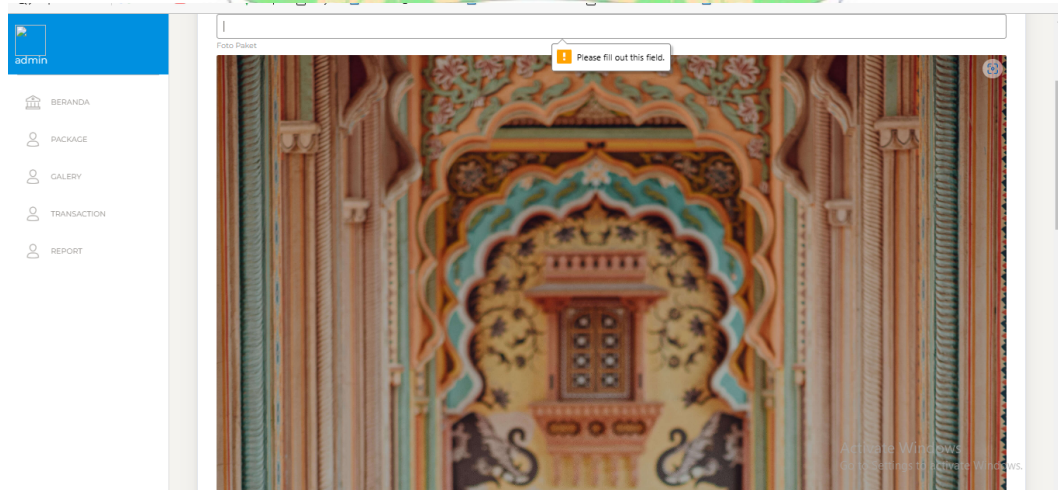
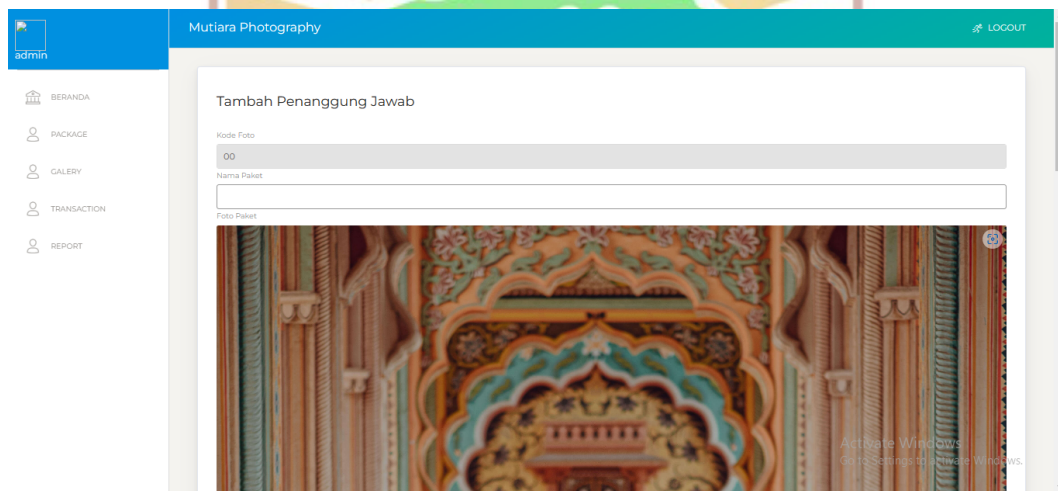
Gambar 5.11 Memilih Salah Satu Data Galeri



Gambar 5.12 Data Galeri Telah Berubah

Tabel 5.5 Kasus Pengujian Data Nama Paket (Salah)

Kasus dan Hasil Uji (Salah)	
Data masukan	Data nama paket salah atau kosong
Yang diharapkan	Sistem menampilkan halaman edit galeri dan notifikasi “Please fill out this field”
Pengamatan	Notifikasi “Please fill out this field” ditampilkan di halaman edit galeri
Hasil	Sesuai



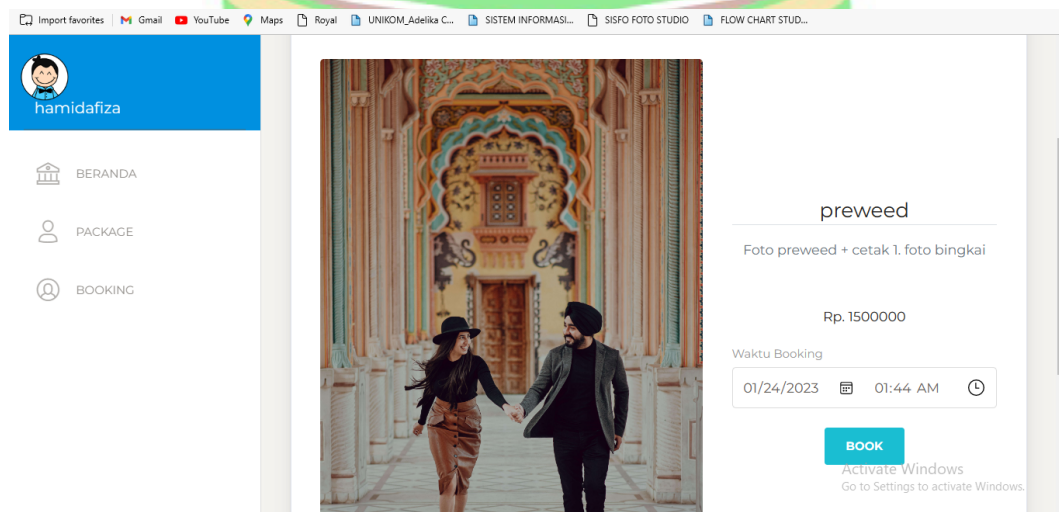
5.2.2.3 Pengujian Booking Studio Foto

Pada pengujian ini dilakukan booking Studio Foto. Hasil pengujian dapat dilihat pada Tabel 5.6.

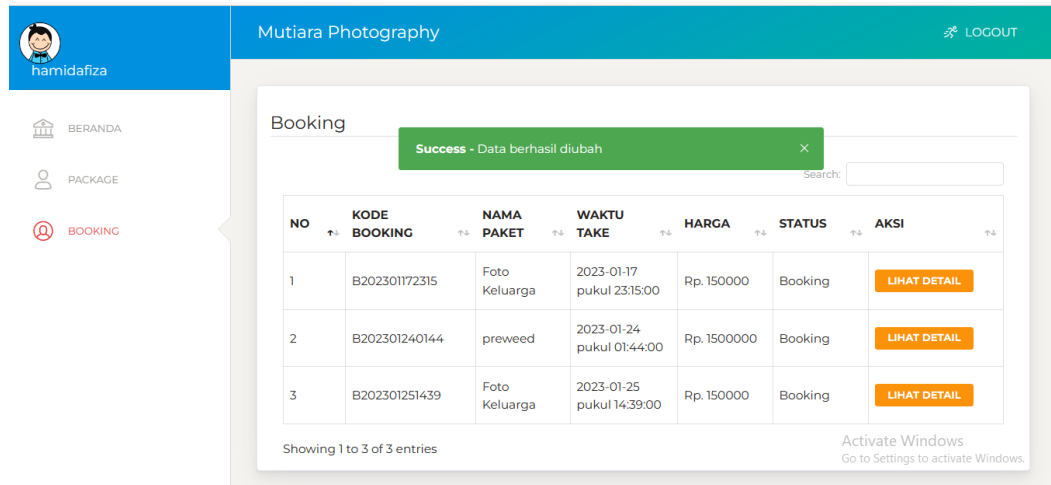
Tabel 5.6 Pengujian Booking Studio Musik (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Member memilih hari, tanggal dan jam <i>booking</i> dan mengisi <i>down payment</i>
Yang diharapkan	Sistem menampilkan halaman <i>booking</i>
Pengamatan	Dapat memilih tanggal dan jam yang diinginkan
Hasil	Sesuai

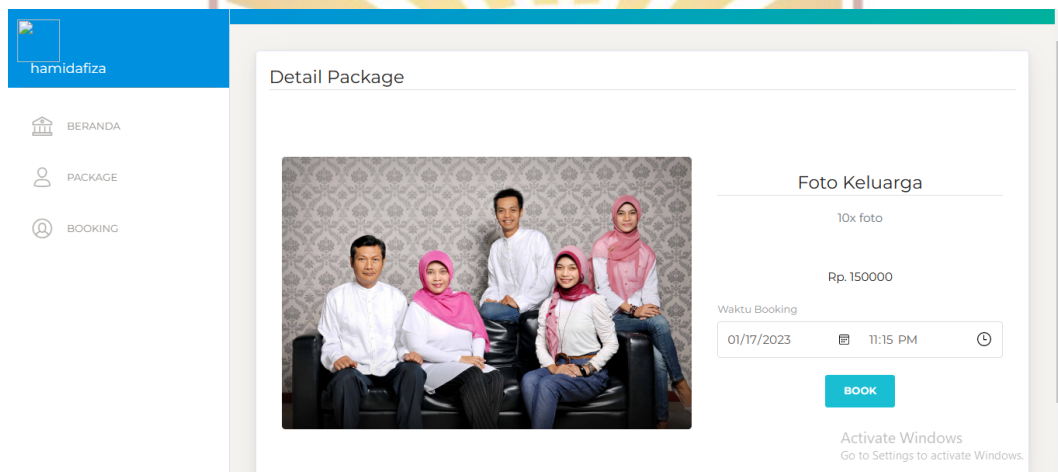
Pengujian aplikasi dilakukan dengan memilih salah satu data paket yang tersedia, selanjutnya memilih tombol lihat detail paket, maka akan tampil halaman detail paket yang akan di booking. Kemudian pilih hari, tanggal, dan jam yang akan di booking. Jika berhasil maka sistem menampilkan data paket yang sudah di booking. Hasil pengujian booking(benar) dapat dilihat pada Gambar 5.13 - Gambar 5.17.



Gambar 5.13 Memilih Waktu booking



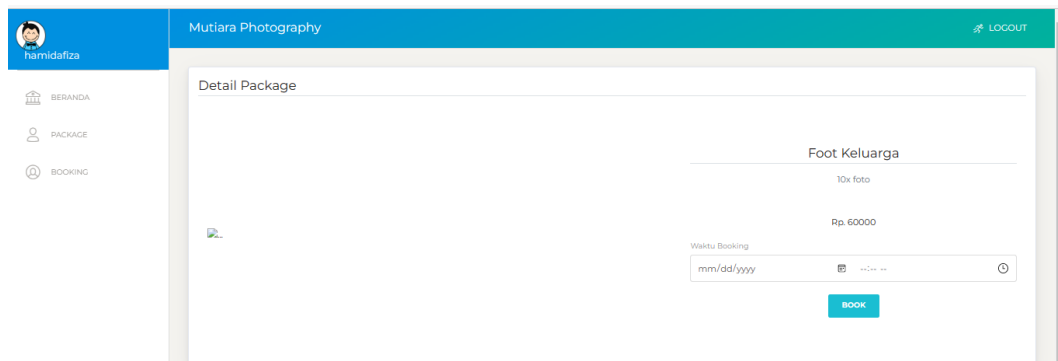
Gambar 5.14 Paket Berhasil di Booking



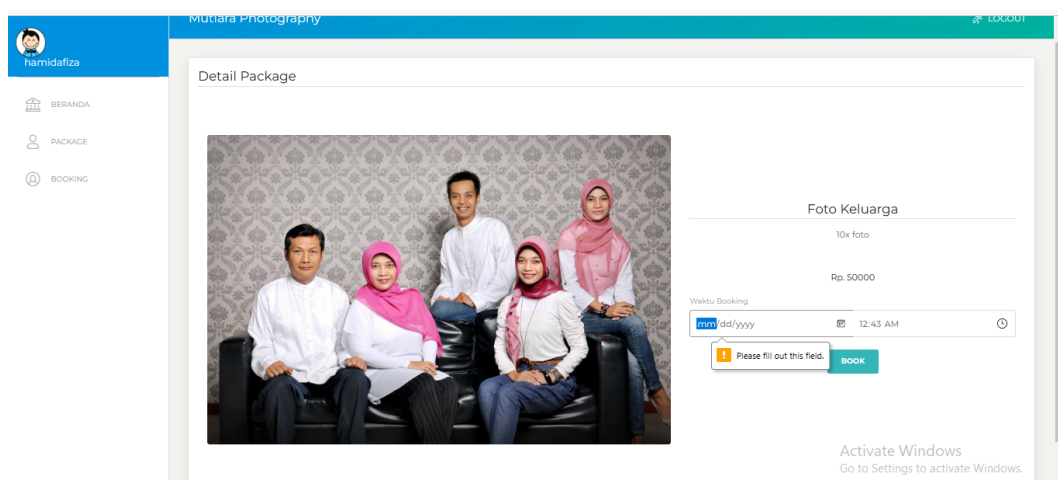
Gambar 5.15 Menampilkan Detail Booking

Tabel 5.7 Kasus Pengujian Tanggal Bulan Kosong (Salah)

Kasus dan Hasil Uji (Salah)	
Data masukan	Data tanggal, bulan, tahun salah atau kosong
Yang diharapkan	Sistem menampilkan halaman detail paket dan notifikasi "Please fill out this field"
Pengamatan	Notifikasi "Please fill out this field" ditampilkan di halaman detail paket
Hasil	Sesuai



Gambar 5.16 Halaman Mengubah Detail Package



Gambar 5.17 Halaman Field Warning Jika Belum Diisi

5.3 Kesimpulan Hasil Pengujian

Tahap pengujian yang dilakukan terhadap aplikasi yang dibangun berfokus pada ketersediaan dan kesesuaian fungsional sistem yang diuji secara manual. Berdasarkan pengujian yang telah dilakukan diperoleh kesesuaian hasil antara perancangan dengan keluaran sistem, selain itu juga tidak ditemukan kegagalan dalam setiap proses pada masing-masing fungsional. Dapat disimpulkan bahwa rancang aplikasi pengelolaan studio foto pada Mutiara Studio Foto berjalan sesuai dengan fungsional yang dirancang dan sesuai dengan kebutuhan yang ada sesuai dengan Tabel 5.8.

Tabel 5.8 Hasil Pengujian

No.	Fungsional	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Login system	▪ Tidak mengisi <i>username</i> dan <i>password</i> dengan benar ▪ Mengisi <i>username</i> dan <i>password</i> dengan benar	▪ Sistem akan menampilkan notifikasi “These credentials do not match our records.”. ▪ Sistem akan mengarahkan halaman ke <i>dashboard</i> admin.	Sesuai
2	Paket yang tersedia	Menambahkan paket, lihat detail paket, edit paket	Menampilkan halaman paket dan status paket sudah berhasil di tambah	Sesuai
3	Transaksi	Mencoba menu ambil dan tolak pemesanan	Sistem menampilkan list data yang sudah di booking, dan tekan menu ambil untuk meng accept pemesanan, dan menu tolak untuk membatalkan pemesanan	Sesuai
4	Laporan	Memilih waktu untuk mencatat laporan perhari, perbulan, mauoun pertahun dan tekan tombol submit	Sistem menampilkan hasil pemesanan selama waktu yang di tentukan	Sesuai
5	Logout	Memilih menu logout	Sistem keluar dari aplikasi dan menampilkan halaman login.	Sesuai

BAB VI

PENUTUP

Bab ini menjelaskan tentang kesimpulan dan saran pada laporan tugas akhir ini. Kesimpulan merupakan pencapaian dari tujuan yang telah ditetapkan pada penelitian, sedangkan saran merupakan harapan yang diinginkan kedepannya terkait penelitian yang telah dilakukan

6.1 Kesimpulan

Berdasarkan hasil penelitian dan perancangan terhadap Aplikasi Pembangunan Sistem Informasi Pengelolaan Studio Foto pada Mutiara Studio Photo dapat disimpulkan sebagai berikut:

1. Sistem informasi ini bertujuan untuk melakukan digitalisasi beberapa proses bisnis yang berjalan pada studio foto Mutiara. Digitalisasi proses bisnis terintegrasi menjadi satu sistem informasi pengelolaan studio foto.
2. Sistem yang dirancang dapat memudahkan customer dalam booking dan memilih paket foto secara online.
3. Sistem ini mempermudah rekapitulasi transaksi dan menghemat waktu dalam pembuatan laporan penjualan serta menghasilkan hasil yang akurat.
4. Sistem ini juga memudahkan admin dalam proses cetak foto dan pengelolaan pesanan

6.2 Saran

Berdasarkan penelitian yang telah dilakukan, penulis menyarankan agar Sistem Informasi Pengelolaan Studio Foto pada Studio Foto Mutiara dapat dilakukan pengembangan pada aplikasi mobile agar aplikasi menjadi lebih fleksibel dan akses data lebih cepat.

DAFTAR PUSTAKA

- Kusumo, R. O., Bangsa, P. G., & Hosana, M. (2019). Perancangan Fotografi sebagai Media Promosi Pariwisata Pantai Kedung Tumpang, Pucanglaban, Tulungagung. *Jurnal DKV Adiwarna, Universitas Kristen Petra*, 1-10.
- Afriyonza, Hendrawan, & Nugroho, A. (2019). PERANCANGAN SISTEM INFORMASI ADMINISTRASI JASA FOTO PERNIKAHAN BERBASIS WEB PADA EUPHORIA PHOTO STUDIO. *Jurnal Ilmiah Media Processor*, 196-209.
- Artina, N. (2006). Penerapan Analisis Kebutuhan Metode Use Case pada Metode Pengembangan Terstruktur . *Jurnal Ilmiah STMIK GI MDP*, 1-6.
- Dewi, L. P., Indahyanti, U., & S, Y. H. (2012). Pemodelan Proses Bisnis Menggunakan Activity Diagram UML DAN BPMN (Studi Kasus FRS ONLINE).
- Mulyono, E. B., Purnama, B., & Effiyaldi. (2019). SISTEM INFORMASI JASA FOTOGRAFI BERBASIS WEB PADA GREEN PHOTOGRAPHY JAMBI. *Jurnal Ilmiah Mahasiswa Sistem Informasi*, 317-330.
- Arman, Rahmelina, L., & Siska, Y. (2019). PERANCANGAN ANTAR MUKA APLIKASI PEMESANAN FOTOGRAFI DAN MAKE UP BERBASIS ONLINE PADA QUEEN PHOTO STUDIO PADANG . *ZONAsi: Jurnal Sistem Informasi*, 86-96.
- Solichin, A. (2016). *PEMROGRAMAN WEB DENGAN PHP DAN MUSQL*. Jakarta: Budi Luhur.
- Yulisman. (2019). Aplikasi Penyewaan Perlengkapan Studio Foto (Kamera dan Aksesoris) Berbasis Web di Cinema Kreatif Desain Pekanbaru. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 16-22.
- Pressman, R. S. (2010). *Rekayasa Perangkat Lunak*. ANDI.
- Sudarma, I. K. (2014). *Fotografi*. Yogyakarta: Graha Ilmu.

Schwartz, B., Peter, Z., & Tkachenko, V. (2012). *High Performance MySQL, 3rd Edition*. California: O'Reilly Media, Inc.

Joubert, Y. T., & Swart, C. (2019). Developing a team performance framework utilising Neethling's thinking style preference instrument. *SA Journal of Human Resource Management*, 50-62.

Dharwiyanti, S., & Wahono, R. S. (2003, June 10). *Pengantar Unified Modeling Language (UML)*. Retrieved from Ilmu Komputer: <http://www.ilmukomputer.com>

Fatta, H. A. (2007). *Analisis dan Perancangan Sistem Informasi*. Yogyakarta: ANDI.





Lampiran A

Usecase scenario aplikasi

A. Admin

1. Admin dapat melakukan login

<i>Use Case</i>	<i>Login</i>
<i>Actor</i>	Customer dan Admin
<i>Entry Condition</i>	Actor berada di halaman utama
<i>Flow of event</i>	1. Actor menginput <i>username</i> dan <i>password</i> pada <i>field</i> yang tersedia. 2. Actor mengklik tombol ' <i>login</i> '. 3. Sistem menampilkan notifikasi <i>login</i> berhasil
<i>Scenario Alternatif</i>	4. Actor mengklik tombol ' <i>login</i> ' 5. Sistem menampilkan notifikasi kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>dashboard</i> .

2. Admin dapat mengelola halaman galeri

a) Menambah

<i>Use Case</i>	Mengelola halaman Galeri (tambah)
<i>Actor</i>	Admin
<i>Entry Condition</i>	Actor telah melakukan login dan berada pada halaman beranda
<i>Flow of event</i>	1. Actor mengklik menu ' <i>Gallery</i> ' 2. Sistem menampilkan halaman menu <i>Gallery</i> 3. Actor mengklik ' <i>tambah Gallery</i> ' 4. Sistem menampilkan halaman tambah <i>Gallery</i> 5. Aktor mengisi <i>form</i> tambah <i>Gallery</i> dan mengklik tombol ' <i>submit</i> '

	6. Sistem menyimpan hasil tambah <i>Gallery</i> 7. Sistem menampilkan notifikasi data berhasil ditambahkan
<i>Scenario Alternatif</i>	8. <i>Actor</i> mengklik tombol ' <i>submit</i> ' saat data yang diinputan kosong 9. Sistem menampilkan pesan kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Gallery</i>

b) Mengubah

<i>Use Case</i>	Mengelola halaman galeri (edit)
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Gallery</i>' 2. Sistem menampilkan halaman menu <i>Gallery</i> 3. <i>Actor</i> mengklik '<i>edit</i>' 4. Sistem menampilkan halaman <i>Gallery</i> yang akan diedit 5. <i>Actor</i> mengisi <i>form</i> tambah <i>Gallery</i> dan mengklik tombol '<i>submit</i>' 6. Sistem menyimpan hasil pembaruan <i>Gallery</i> 7. Sistem menampilkan notifikasi data berhasil diperbarui
<i>Scenario Alternatif</i>	8. <i>Actor</i> mengklik tombol ' <i>submit</i> ' saat data yang diperbarui kosong 9. Sistem menampilkan pesan kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Gallery</i>

c) Menghapus

<i>Use Case</i>	Mengelola halaman galeri (edit)
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu 'Gallery' 2. Sistem menampilkan halaman <i>Gallery</i> 3. <i>Actor</i> mengklik tombol 'hapus' 4. Sistem menampilkan <i>popup</i> konfirmasi hapus <i>Gallery</i> 5. <i>Actor</i> mengklik tombol 'OK' 6. Sistem menampilkan notifikasi <i>Gallery</i> telah terhapus
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Gallery</i>

3. Admin dapat mengelola paket

a) Menambah

<i>Use Case</i>	Mengelola paket (tambah)
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu 'Package' 2. Sistem menampilkan halaman menu <i>Package</i> 3. <i>Actor</i> mengklik 'tambah <i>Package</i> 4. Sistem menampilkan halaman tambah <i>Package</i> 5. Aktor mengisi <i>form</i> tambah <i>Package</i> dan mengklik tombol 'submit' 6. Sistem menyimpan hasil tambah <i>Package</i>

	7. Sistem menampilkan notifikasi data berhasil ditambahkan
<i>Scenario Alternatif</i>	8. <i>Actor</i> mengklik tombol ' <i>submit</i> ' saat data yang diinputan kosong 9. Sistem menampilkan pesan kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Package</i>

b) Melihat

<i>Use Case</i>	Mengelola paket (lihat)
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan login dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Package</i>' 2. Sistem menampilkan halaman menu <i>Package</i> 3. <i>Actor</i> mengklik '<i>detail Package</i>' 4. Sistem menampilkan halaman detail <i>Package</i>
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Package</i>

c) Mengubah

<i>Use Case</i>	Mengelola halaman paket (edit)
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Package</i>' 2. Sistem menampilkan halaman menu <i>Package</i> 3. <i>Actor</i> mengklik '<i>edit</i>'

	4. Sistem menampilkan halaman <i>Package</i> yang akan diedit 5. <i>Actor</i> mengisi <i>form</i> tambah <i>Package</i> dan mengklik tombol ' <i>submit</i> ' 6. Sistem menyimpan hasil pembaruan <i>Package</i> 7. Sistem menampilkan notifikasi data berhasil diperbarui
<i>Scenario Alternatif</i>	8. <i>Actor</i> mengklik tombol ' <i>submit</i> ' saat data yang diperbarui kosong 9. Sistem menampilkan pesan kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Package</i>

d) Menghapus

<i>Use Case</i>	Mengelola halaman paket (hapus)
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu ' <i>Package</i> ' 2. Sistem menampilkan halaman <i>Package</i> 3. <i>Actor</i> mengklik tombol 'hapus' 4. Sistem menampilkan <i>popup</i> konfirmasi hapus <i>Package</i> 5. <i>Actor</i> mengklik tombol 'OK' 6. Sistem menampilkan notifikasi <i>Package</i> telah terhapus
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Package</i>

4. Admin dapat mengelola transaksi pembayaran

a) Menerima

<i>Use Case</i>	Menerima transaksi pembayaran
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengkili menu '<i>Booking</i>' 2. Sistem menampilkan halaman '<i>Booking</i>' 3. <i>Actor</i> mengklik tombol 'Mengambil' 4. Sistem menampilkan <i>popup</i> notifikasi 5. <i>Actor</i> mengklik tombol 'OK' 6. Sistem menyimpan data
<i>Exit Condition</i>	Sistem menampilkan halaman

b) Menolak

<i>Use Case</i>	Menolak transaksi pembayaran
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengkili menu '<i>Booking</i>' 2. Sistem menampilkan halaman '<i>Booking</i>' 3. <i>Actor</i> mengklik tombol 'Menolak' 4. Sistem menampilkan <i>popup</i> notifikasi 5. <i>Actor</i> mengklik tombol 'OK' 6. Sistem menyimpan data
<i>Exit Condition</i>	Sistem menampilkan halaman <i>transaction</i>

5. Admin dapat melihat rekap laporan transaksi

<i>Use Case</i>	Melihat rekap laporan transaksi
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> memilih menu '<i>Transaction</i>' 2. Sistem menampilkan halaman <i>transaction</i> 3. <i>Actor</i> memilih <i>transaction</i> yang akan dilihat 4. Sistem menampilkan <i>transaction</i> yang dipilih
<i>Exit Condition</i>	Sistem menampilkan halaman <i>transaction</i>

B. Customer

1. Customer dapat melakukan login

<i>Use Case</i>	<i>Login</i>
<i>Actor</i>	<i>Customer</i> dan <i>customer service</i>
<i>Entry Condition</i>	<i>Actor</i> berada di halaman <i>login</i>
<i>Flow of event</i>	1. <i>Actor</i> menginput <i>username</i> dan <i>password</i> pada <i>field</i> yang tersedia. 2. <i>Actor</i> mengklik tombol <i>login</i>. 3. Sistem menampilkan notifikasi <i>login</i> berhasil
<i>Scenario Alternatif</i>	4. <i>Actor</i> mengklik tombol <i>login</i> 5. Sistem menampilkan notifikasi kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman dashboard.

2. Customer dapat melakukan registrasi akun

<i>Use Case</i>	Registrasi Akun
-----------------	-----------------

<i>Actor</i>	<i>Customer</i>
<i>Entry Condition</i>	<i>Actor</i> berada pada halaman utama
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu 'Register' pada halaman utama 2. Sistem menampilkan halaman <i>Register</i> 3. <i>Actor</i> menginputkan <i>form</i> register dan mengklik tombol 'Daftar' 4. Sistem menyimpan data dan menampilkan halaman Beranda
<i>Scenario Alternatif</i>	5. <i>Actor</i> mengklik tombol 'Daftar' saat data yang diperbarui kosong 6. Sistem menampilkan pesan kesalahan
<i>Scenario Alternatif</i>	7. <i>Actor</i> mengklik tombol 'Daftar' 8. Sistem menampilkan pesan kesalahan
<i>Exit Condition</i>	Sistem menampilkan halaman Beranda.

3. Customer dapat melihat galeri studio foto

<i>Use Case</i>	Melihat Galeri Studio Foto
<i>Actor</i>	<i>Customer</i>
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu 'Gallery' 2. Sistem menampilkan halaman <i>Galery</i>
<i>Exit Condition</i>	Sistem menampilkan halaman Beranda.

4. Customer dapat melihat profil studio foto

5. Customer dapat mengelola booking

a) Menambah

<i>Use Case</i>	Melakukan <i>Booking</i>
<i>Actor</i>	<i>Customer</i>
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Package</i>' 2. Sistem menampilkan halaman menu <i>Package</i> 3. <i>Actor</i> memilih <i>package</i> yang diinginkan 4. Sistem menampilkan halaman detail <i>package</i> 5. <i>Actor</i> mengklik tombol '<i>Book</i>' 6. Sistem menyimpan data 7. Sistem menampilkan notifikasi data berhasil dilakukan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>Booking</i>

b) Melihat

<i>Use Case</i>	Melihat riwayat <i>Booking</i>
<i>Actor</i>	<i>Customer</i>
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Booking</i>' 2. Sistem menampilkan halaman menu <i>Booking</i>
<i>Exit Condition</i>	Sistem menampilkan halaman menu <i>Booking</i>

c) Mengubah

<i>Use Case</i>	Mengubah <i>Booking</i>
<i>Actor</i>	<i>Customer</i>
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Booking</i>' 2. Sistem menampilkan halaman menu <i>Booking</i> 3. <i>Actor</i> memilih tombol '<i>Edit</i>' pada <i>Booking</i> yang diinginkan 4. Sistem menampilkan halaman <i>edit booking</i> 5. <i>Actor</i> memilih <i>package</i> dan mengklik tombol '<i>Book</i>' 6. Sistem menyimpan data 7. Sistem menampilkan notifikasi data berhasil dilakukan
<i>Exit Condition</i>	Sistem menampilkan halaman <i>booking</i>

d) Mencetak

<i>Use Case</i>	Melihat Galeri Studio Foto
<i>Actor</i>	<i>Customer</i>
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> mengklik menu '<i>Booking</i>' 2. Sistem menampilkan halaman menu <i>Booking</i> 3. <i>Actor</i> memilih tombol '<i>cetak</i>' pada <i>Booking</i> yang diinginkan 4. Sistem mencetak <i>booking</i> yang diinginkan
<i>Exit Condition</i>	Sistem mencetak <i>booking</i>

C. Pimpinan

1. Pimpinan dapat melihat laporan transaksi

<i>Use Case</i>	Melihat rekap laporan transaksi
<i>Actor</i>	Admin
<i>Entry Condition</i>	<i>Actor</i> telah melakukan <i>login</i> dan berada pada halaman beranda
<i>Flow of event</i>	1. <i>Actor</i> memilih menu '<i>Transaction</i>' 2. Sistem menampilkan halaman <i>transaction</i> 3. <i>Actor</i> memilih <i>transaction</i> yang akan dilihat 4. Sistem menampilkan <i>transaction</i> yang dipilih
<i>Exit Condition</i>	Sistem menampilkan halaman <i>transaction</i>



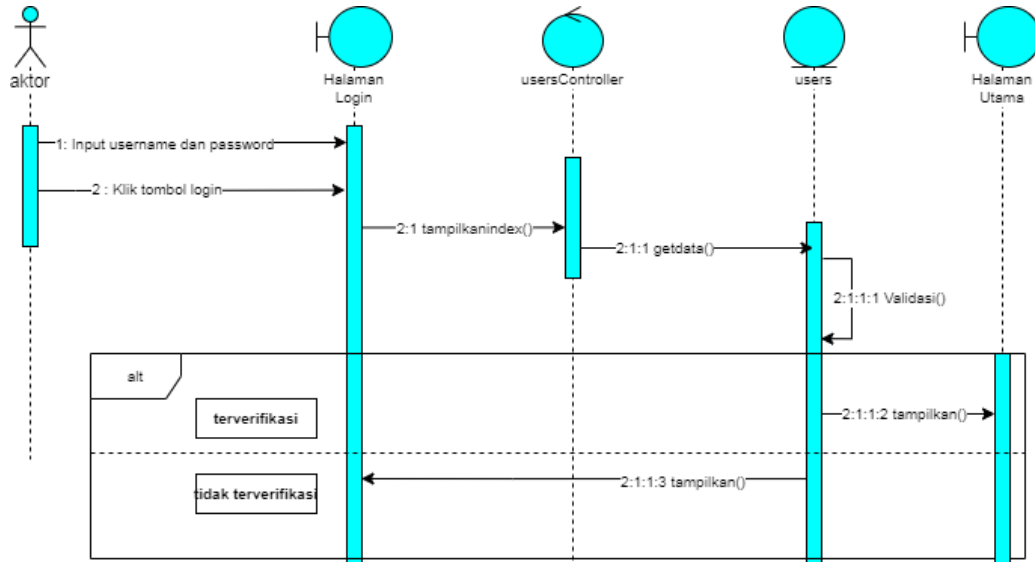


Lampiran B

Sequence Diagram

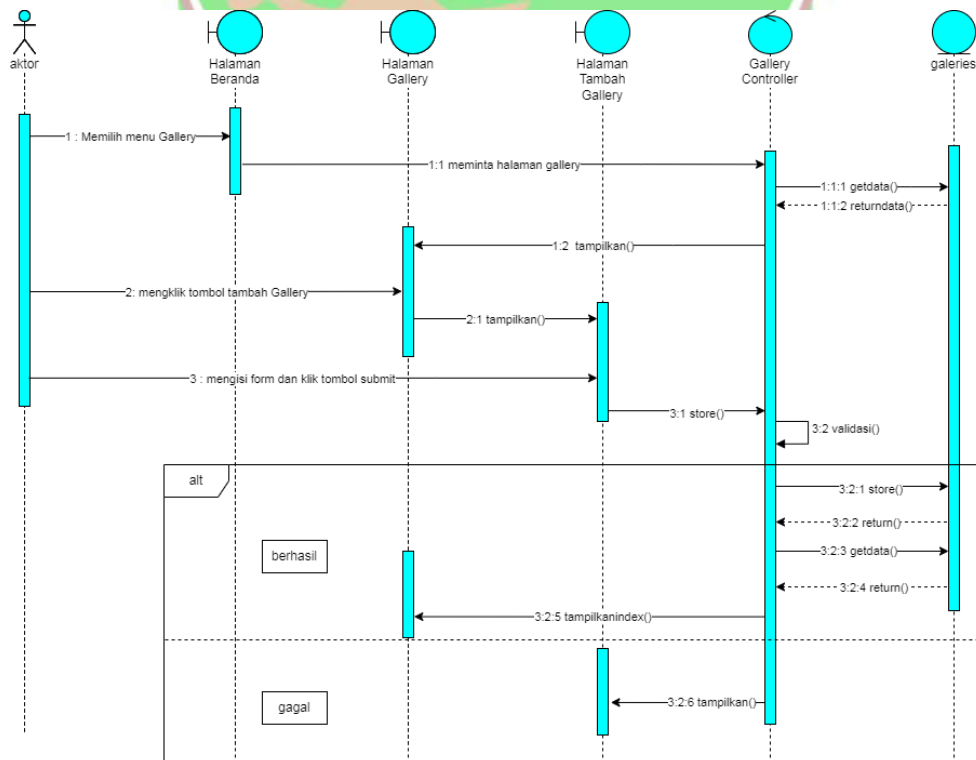
A. Admin

1. Admin dapat melakukan login

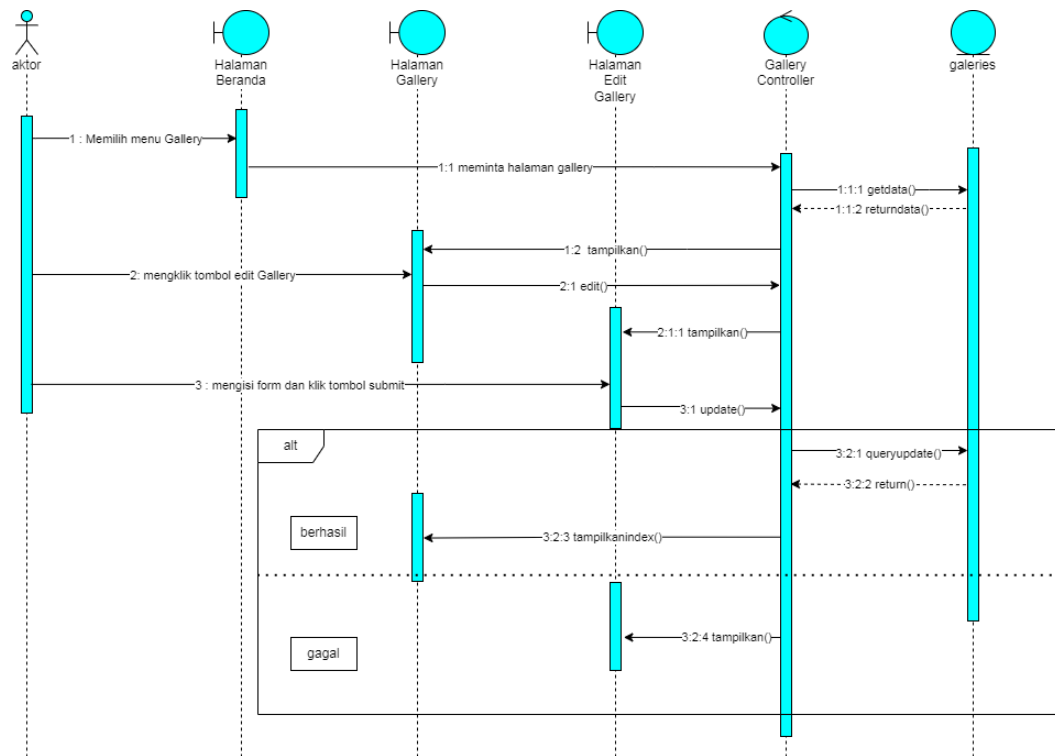


2. Admin dapat mengelola halaman galeri

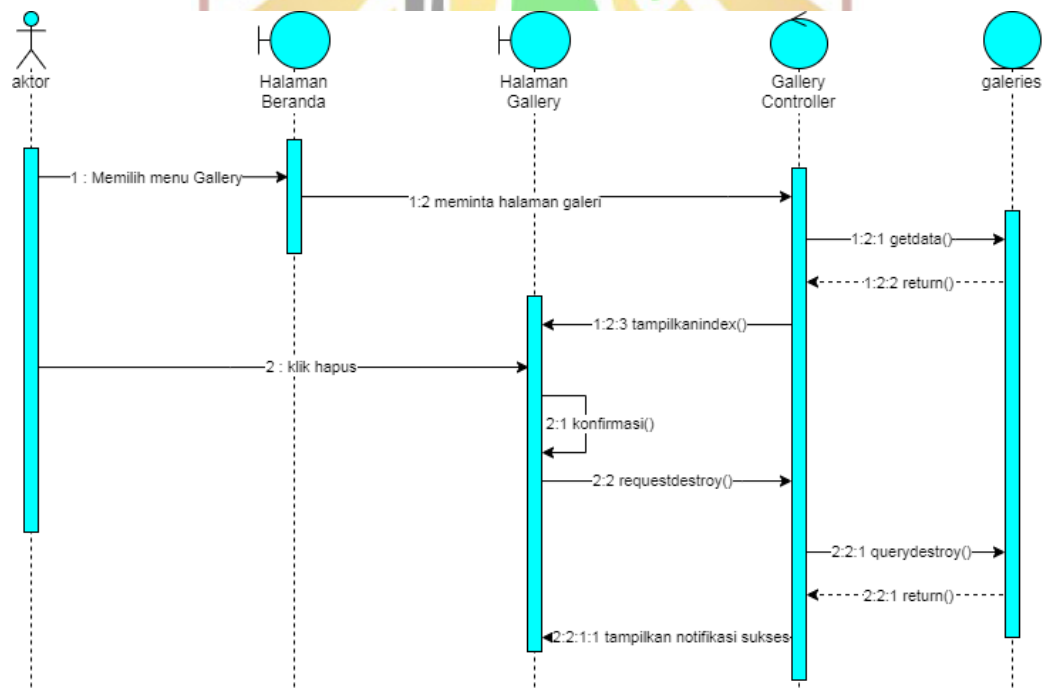
a) Menambah galeri



b) Mengubah galeri

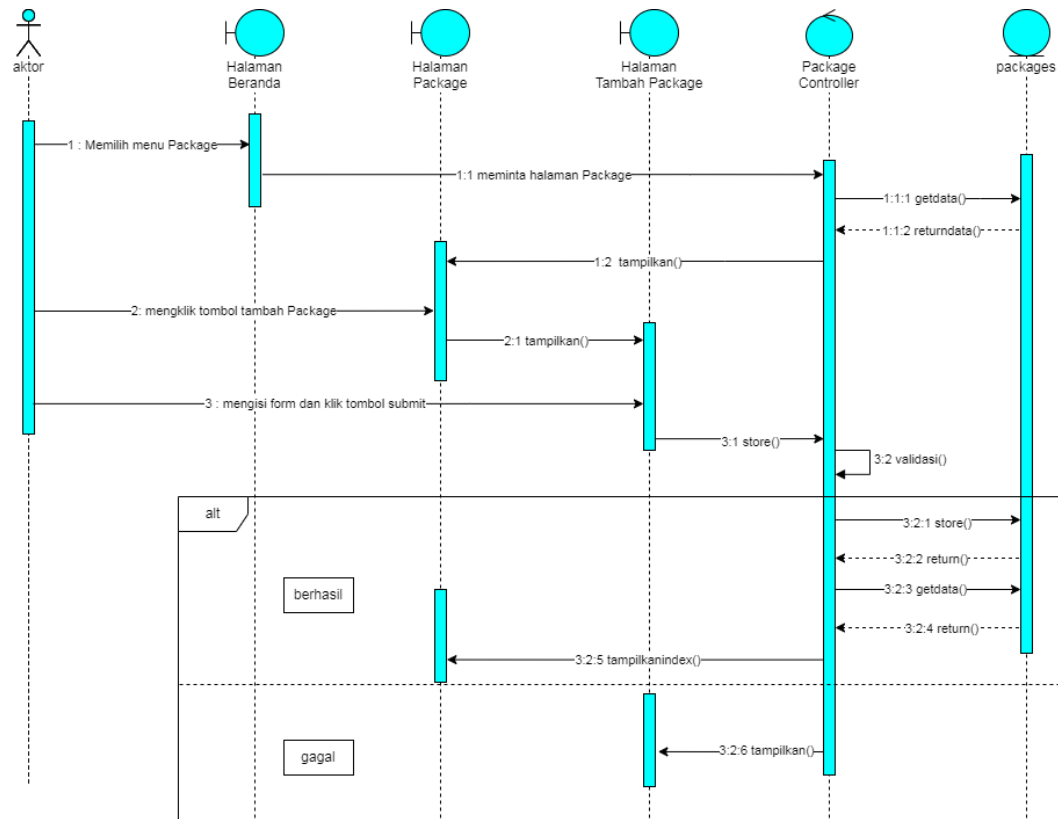


c) Menghapus galeri

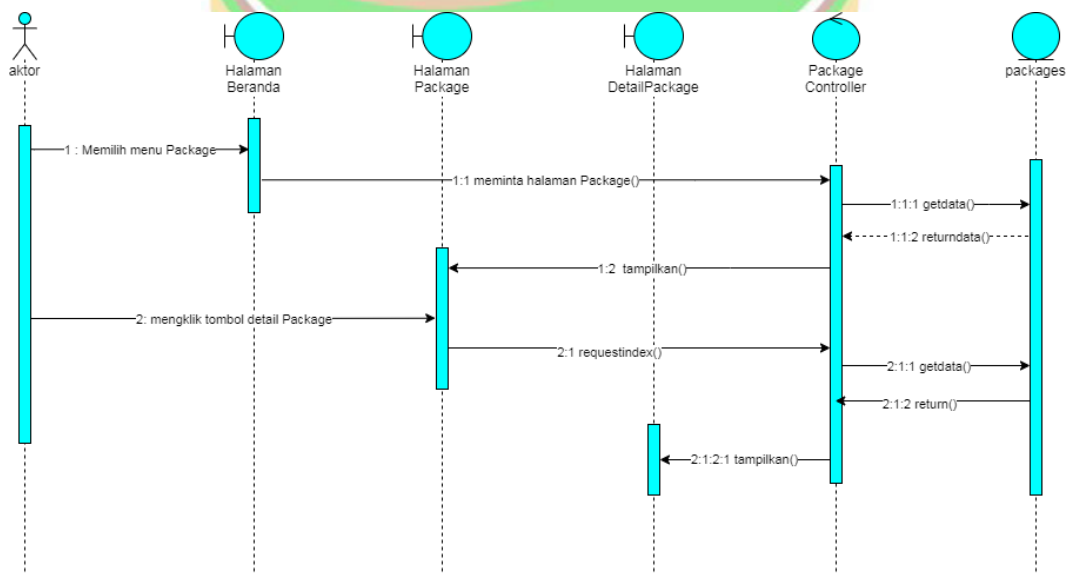


3. Admin dapat mengelola paket

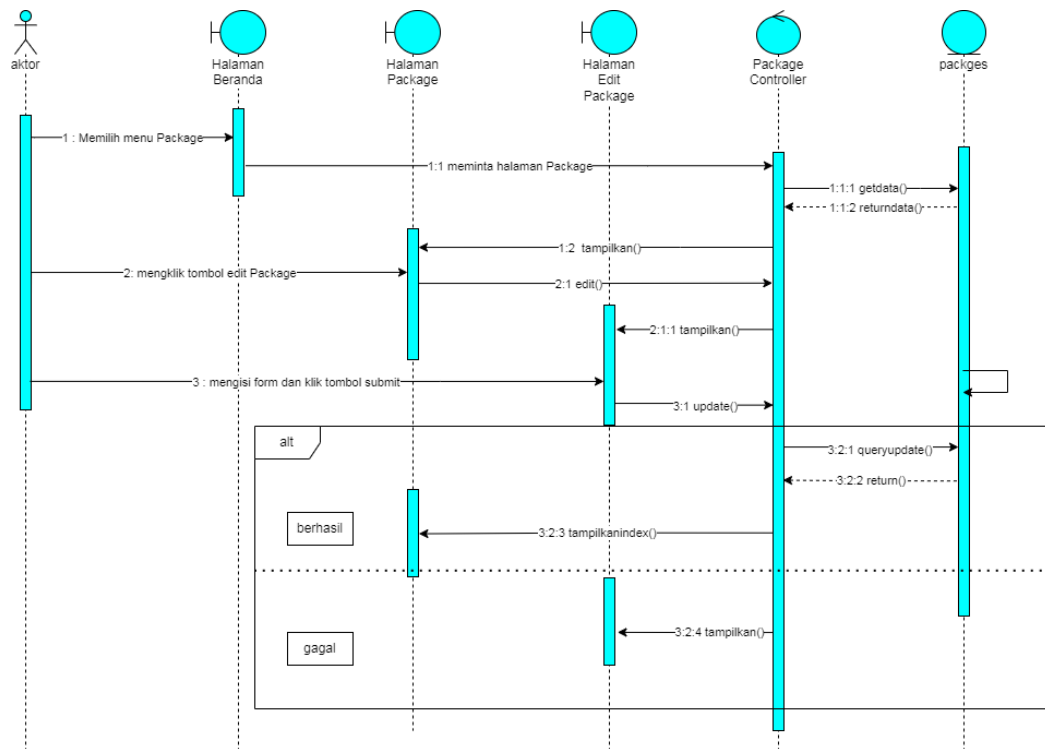
a) Menambah paket



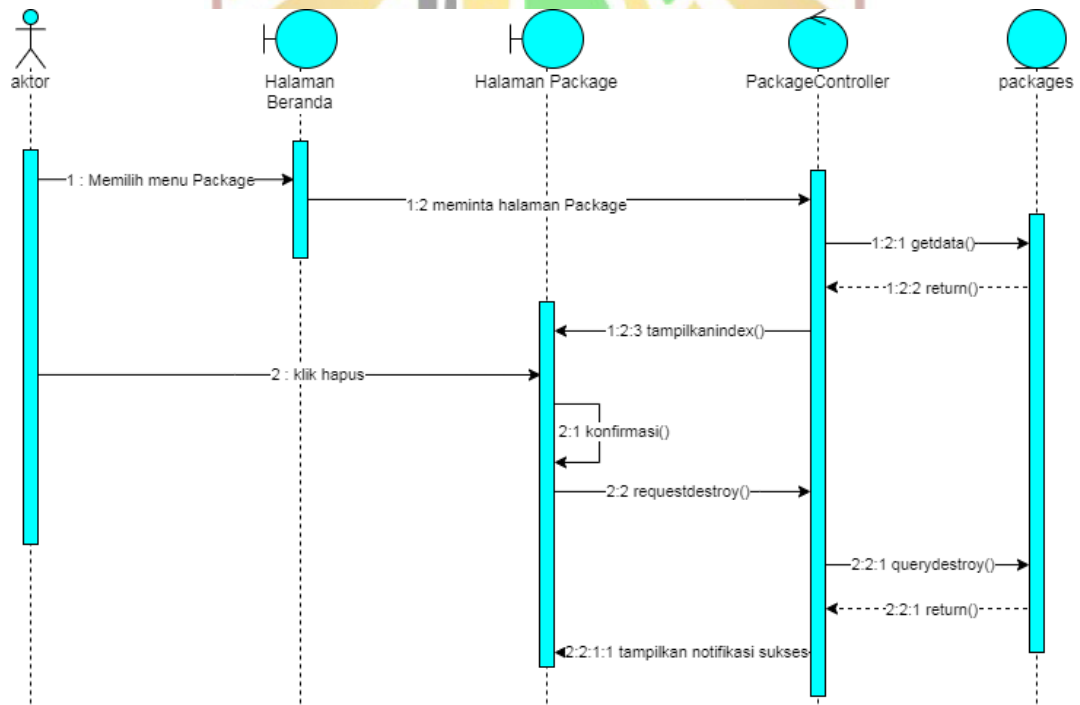
b) Melihat paket



c) Mengubah paket

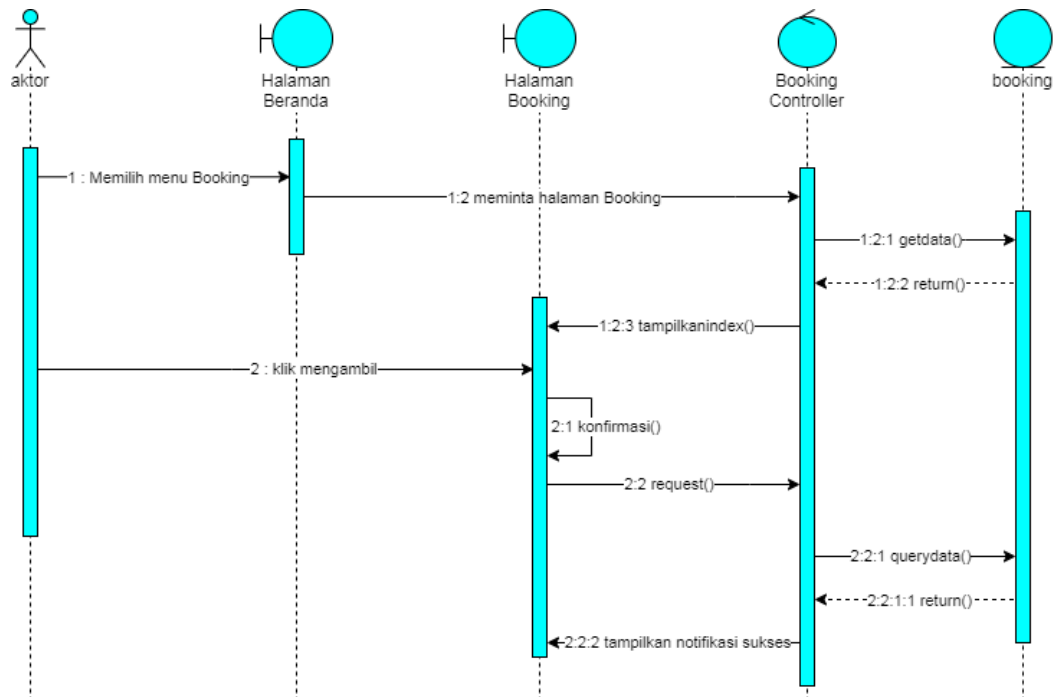


d) Menghapus paket

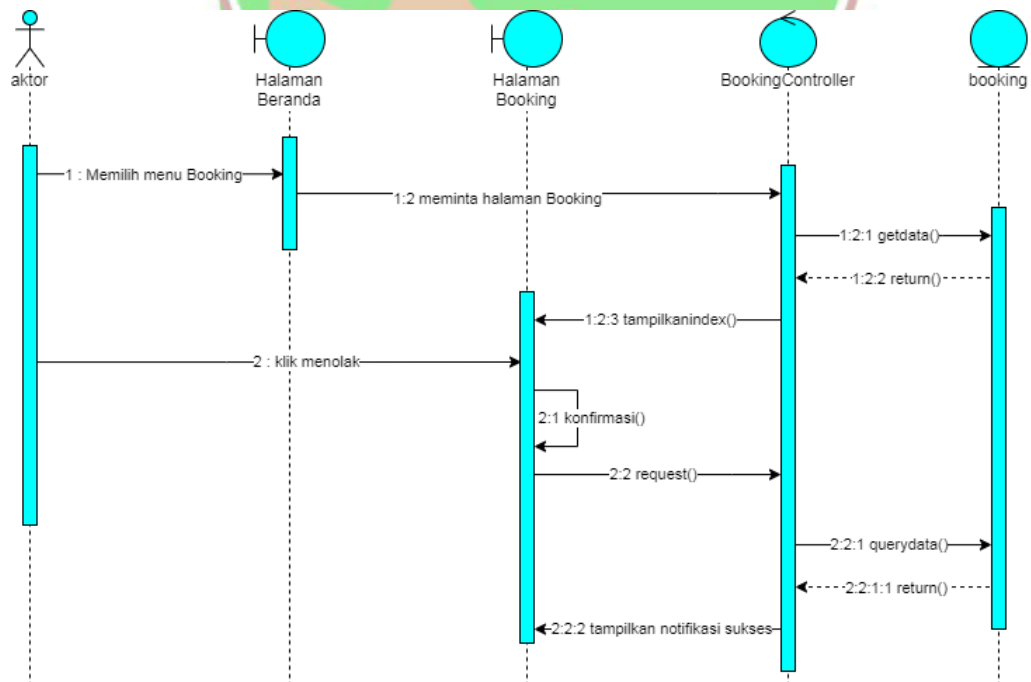


4. Admin dapat mengelola transaksi pembayaran

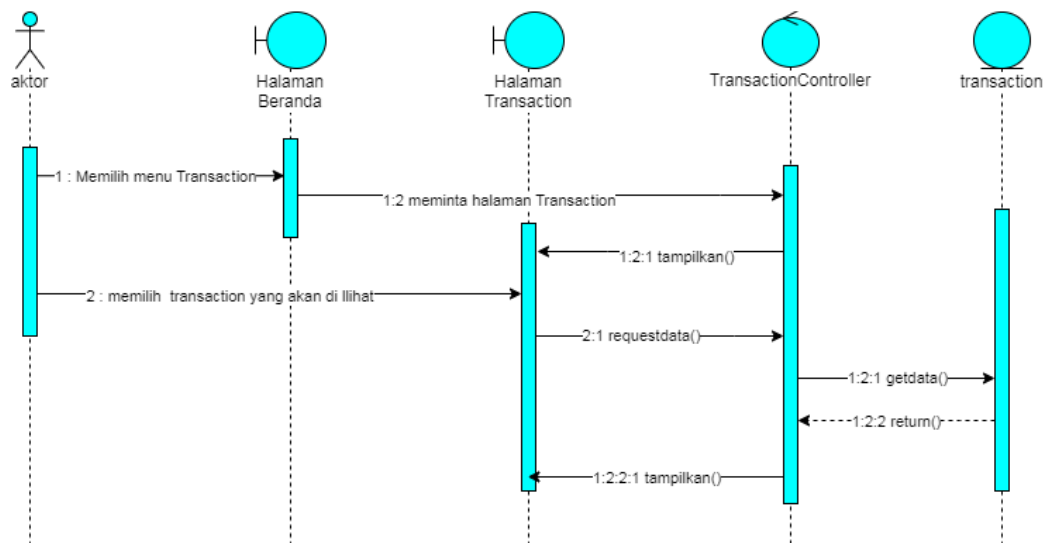
a) Menerima transaksi



b) Menolak transaksi

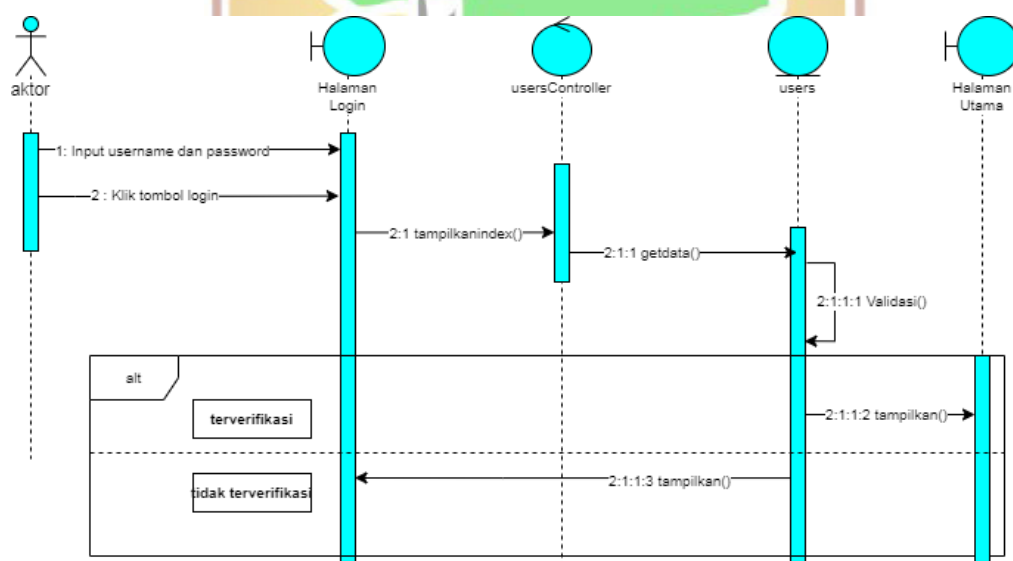


5. Admin dapat melihat rekapan laporan transaksi

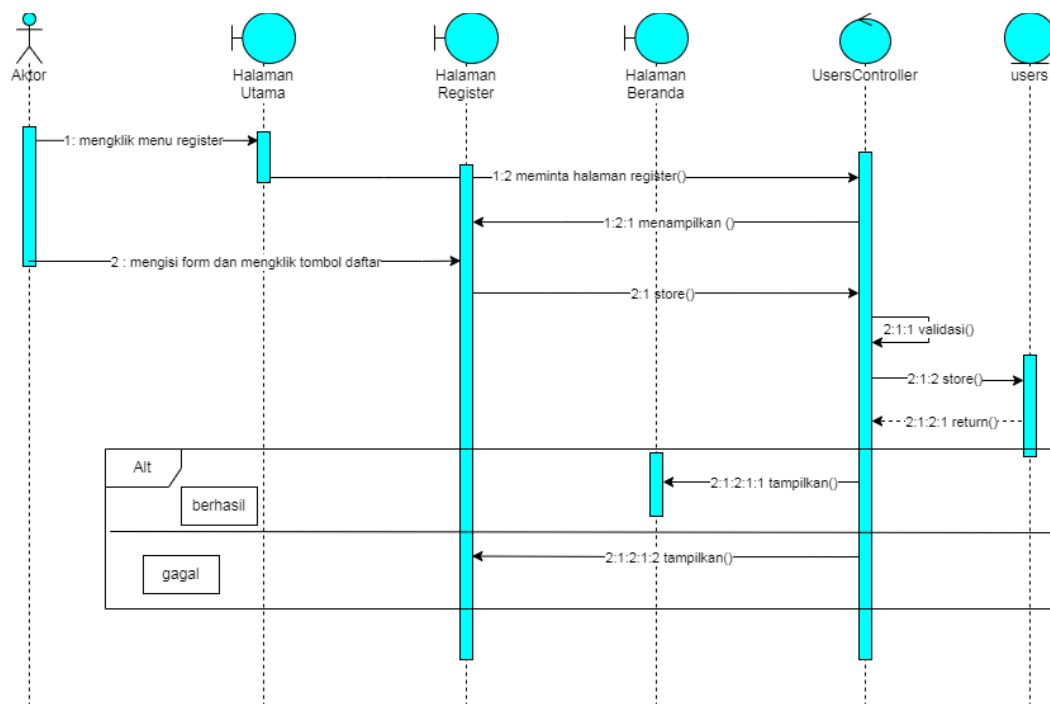


B. Customer

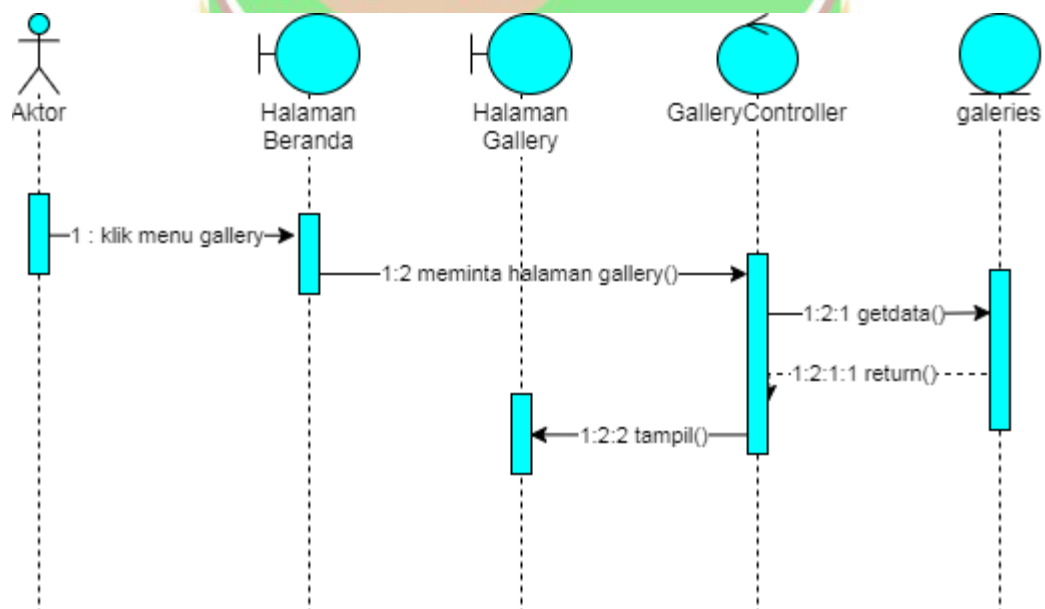
1. Customer dapat melakukan login



2. Customer dapat melakukan registrasi akun

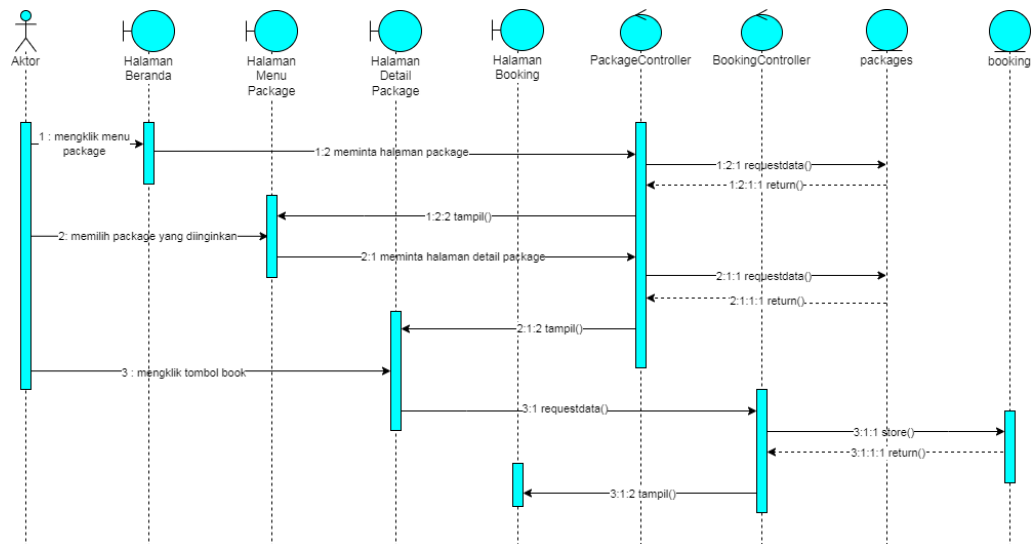


3. Customer dapat melihat galeri studio foto

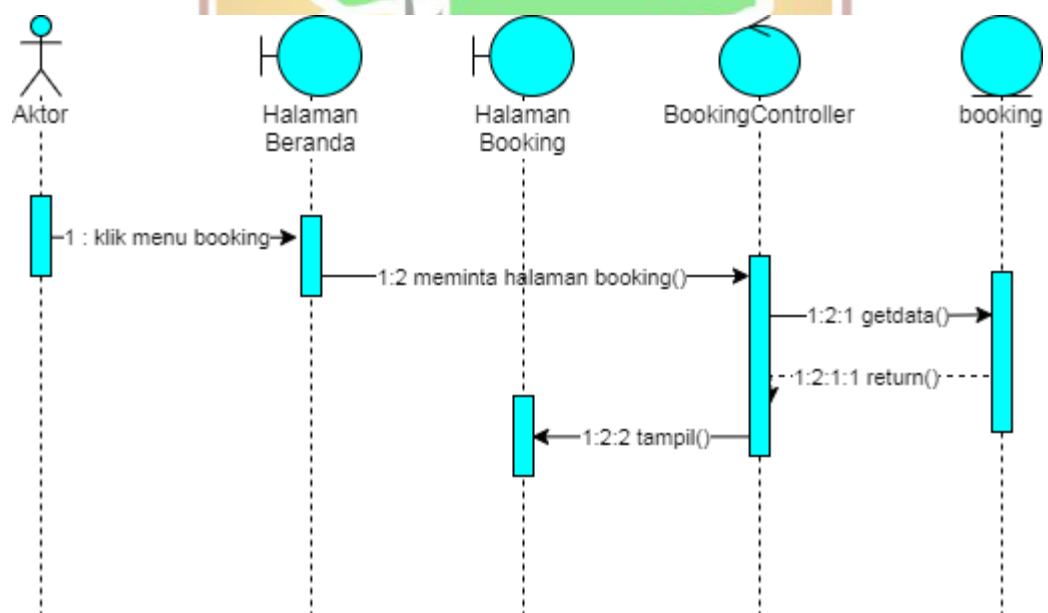


4. Customer dapat mengelola booking

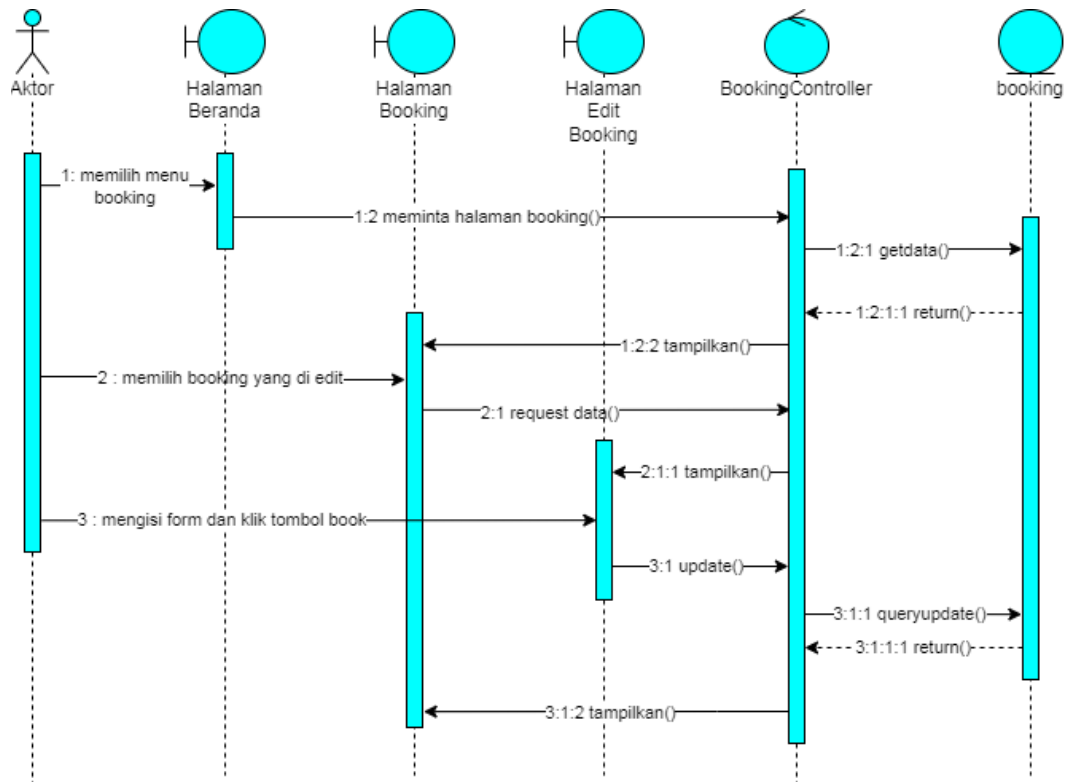
a) Menambah booking



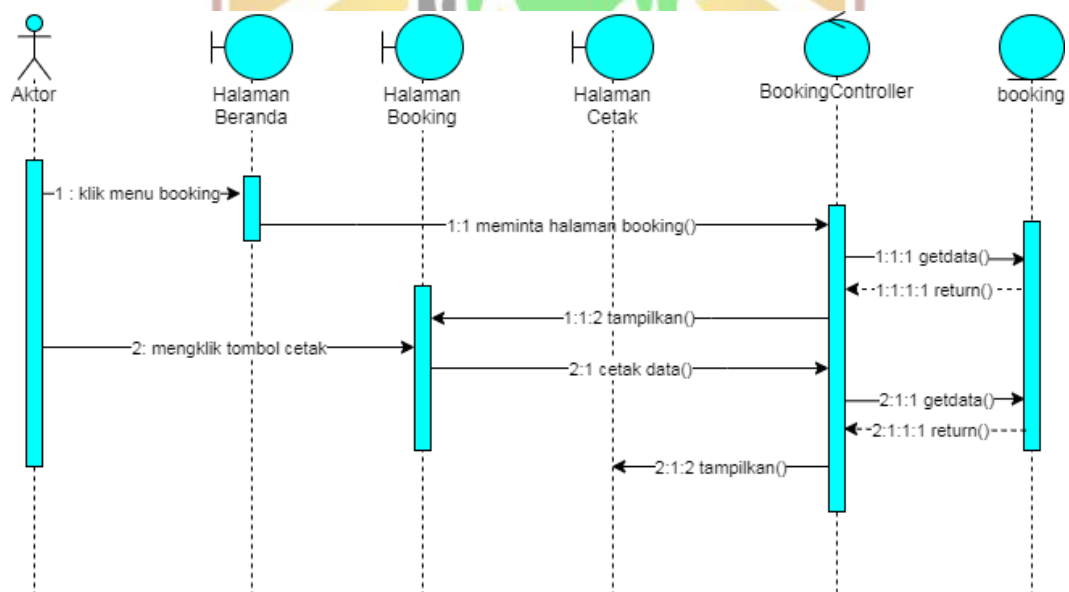
b) Melihat booking



c) Mengubah booking

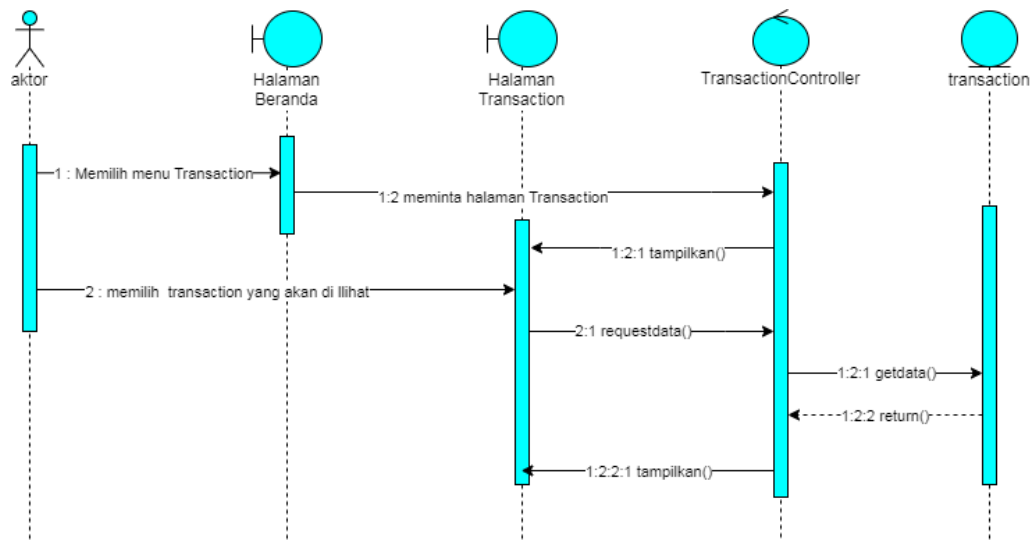


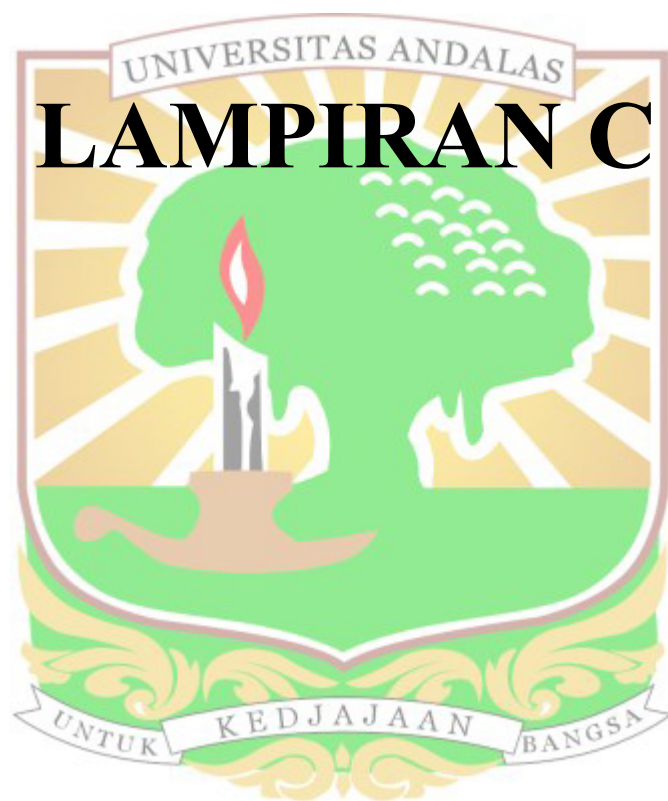
d) Mencetak *booking*



C. Pimpinan

1. Pimpinan dapat melihat laporan transaksi





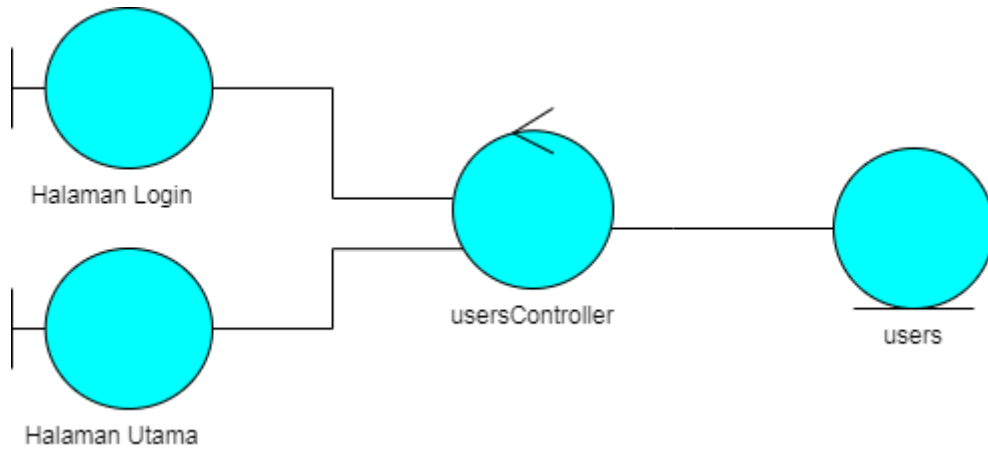
LAMPIRAN C

Lampiran C

Class Analysis

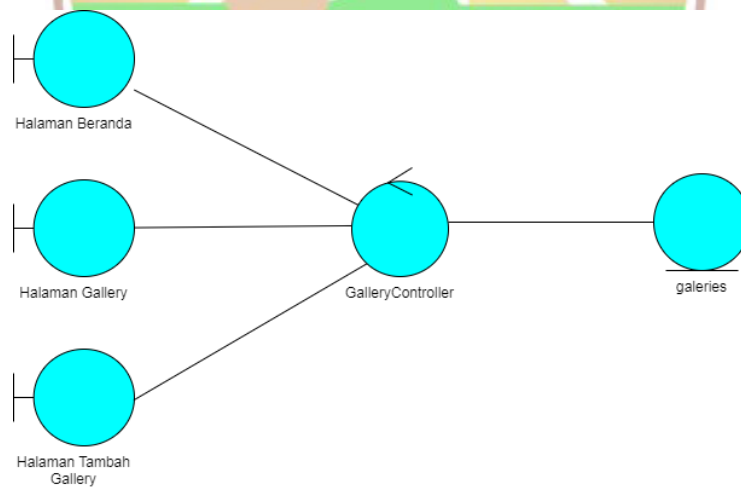
A. Admin

1. Admin dapat melakukan login

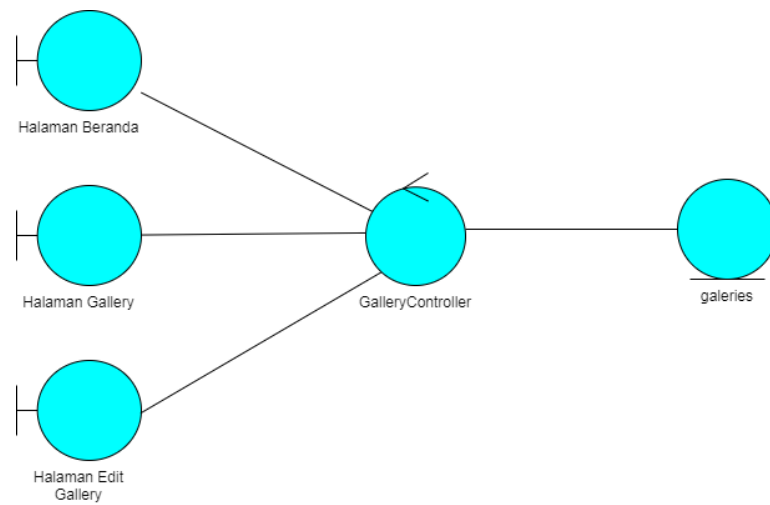


2. Admin dapat mengelola halaman galeri

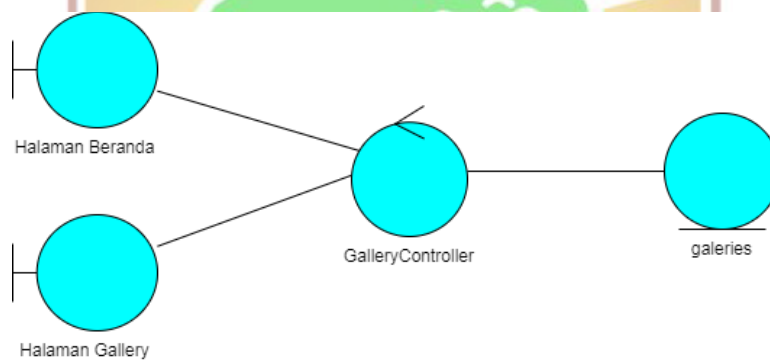
a) Menambah



b) Mengubah

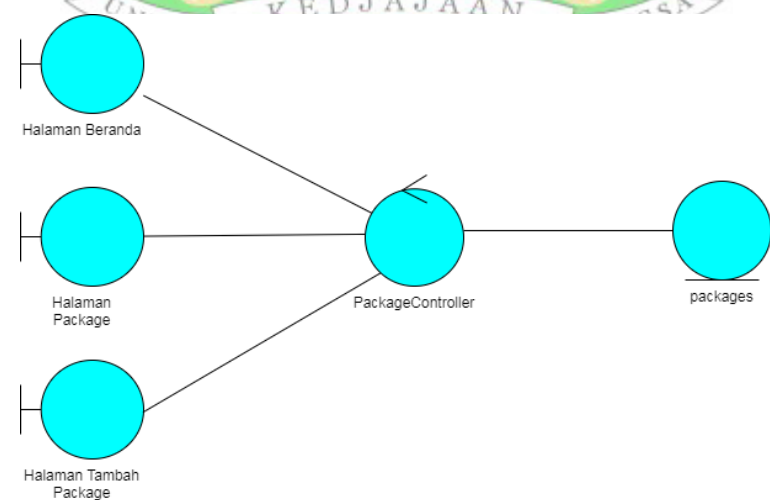


c) Menghapus

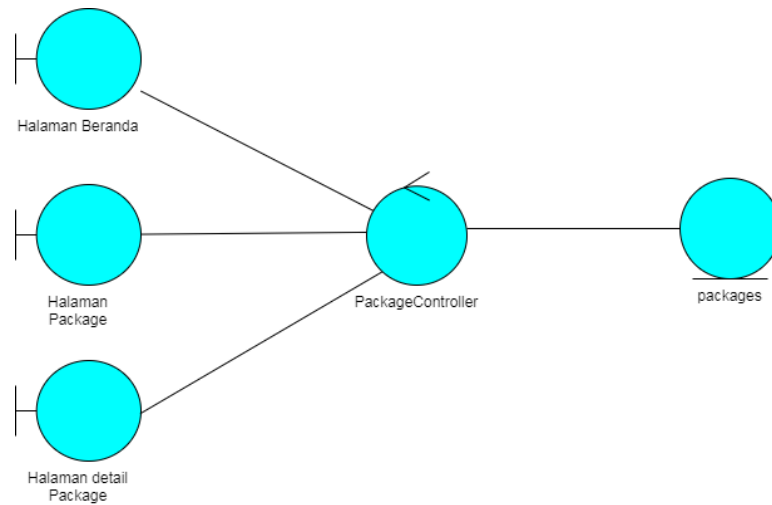


3. Admin dapat mengelola paket

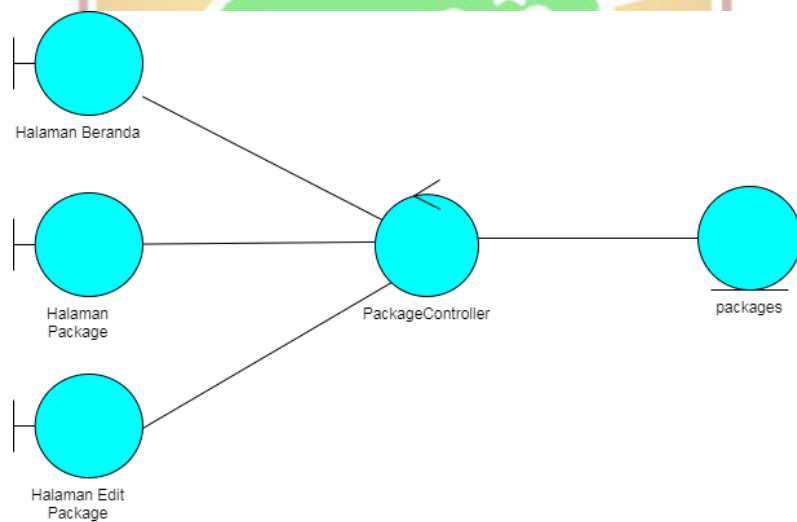
a) Menambah



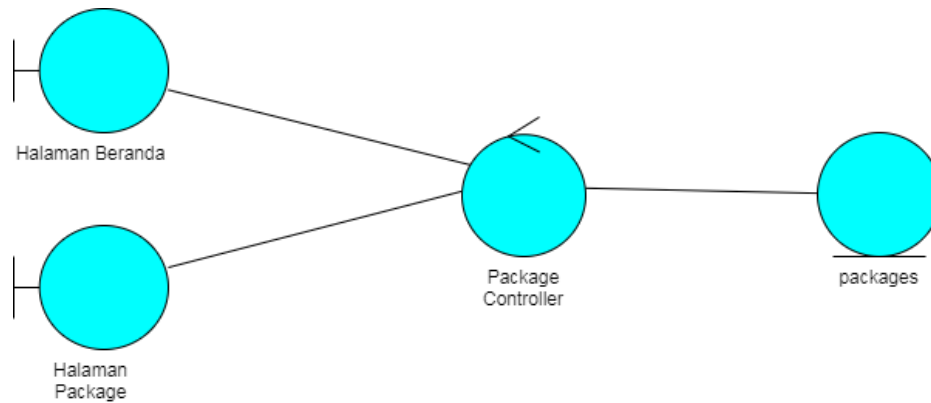
b) Melihat



c) Mengubah

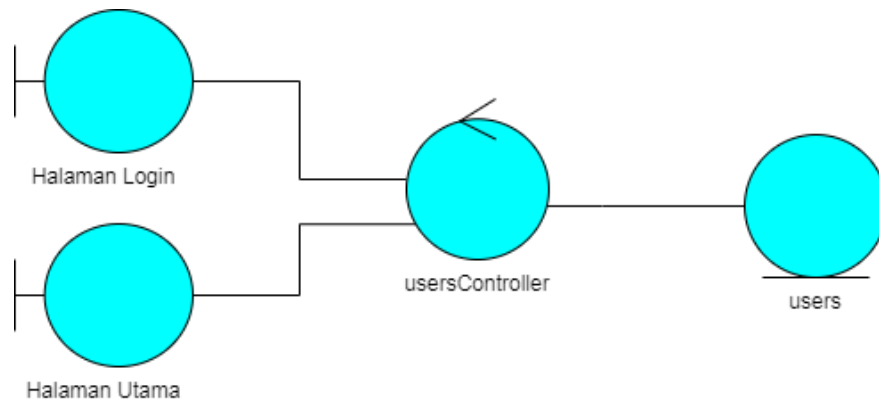


d) Menghapus

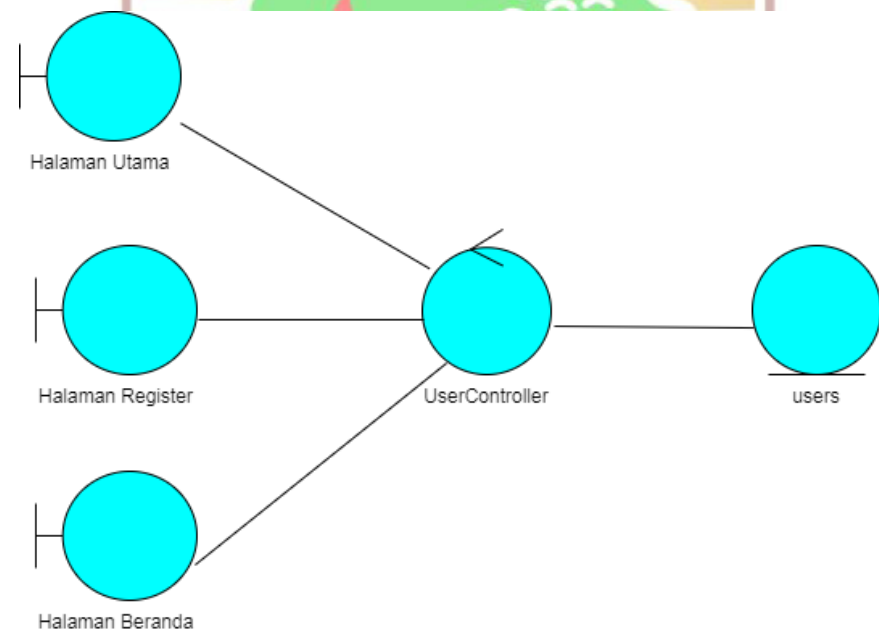


B. Customer

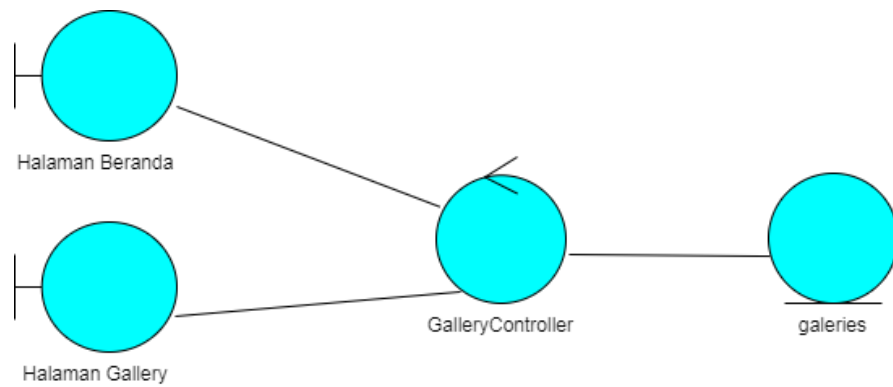
1. Customer dapat melakukan login



2. Customer dapat melakukan registrasi akun

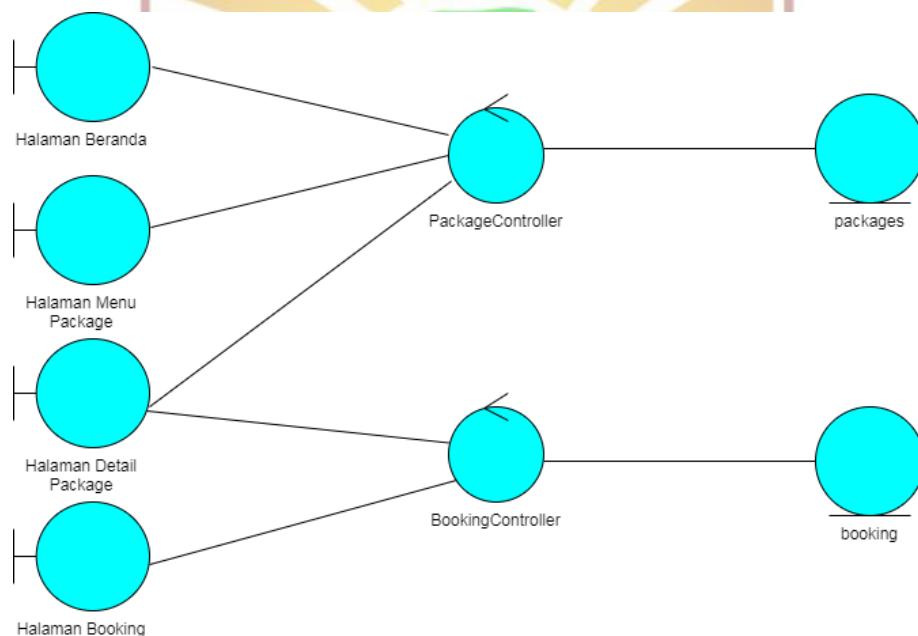


3. Customer dapat melihat galeri studio foto

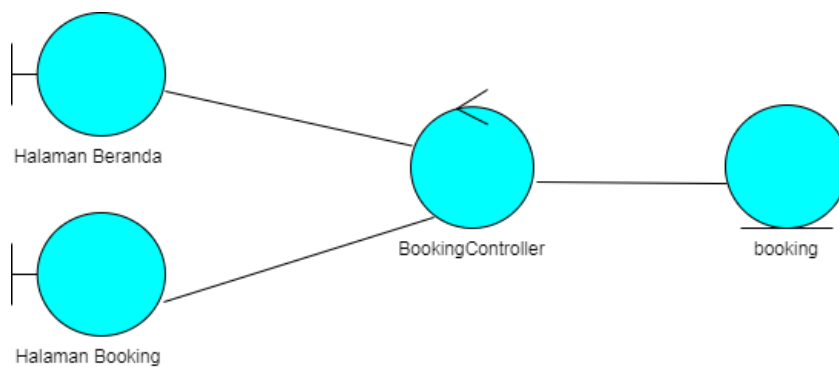


4. Customer dapat mengelola booking

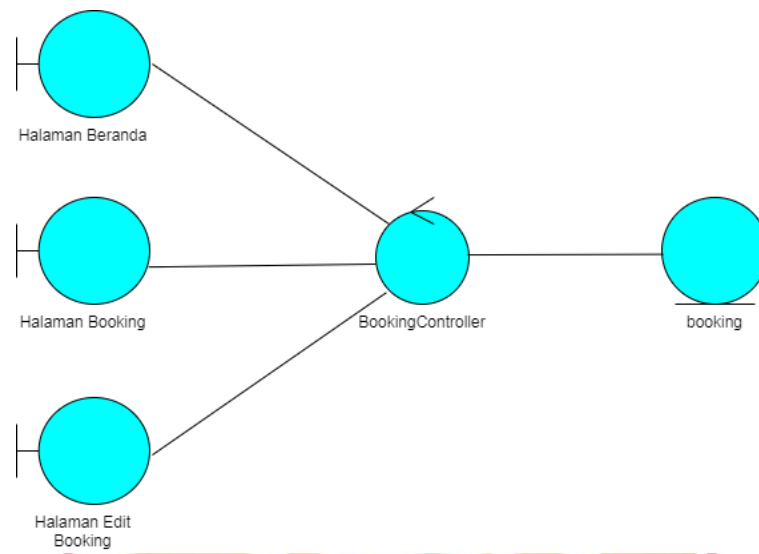
a) Menambah



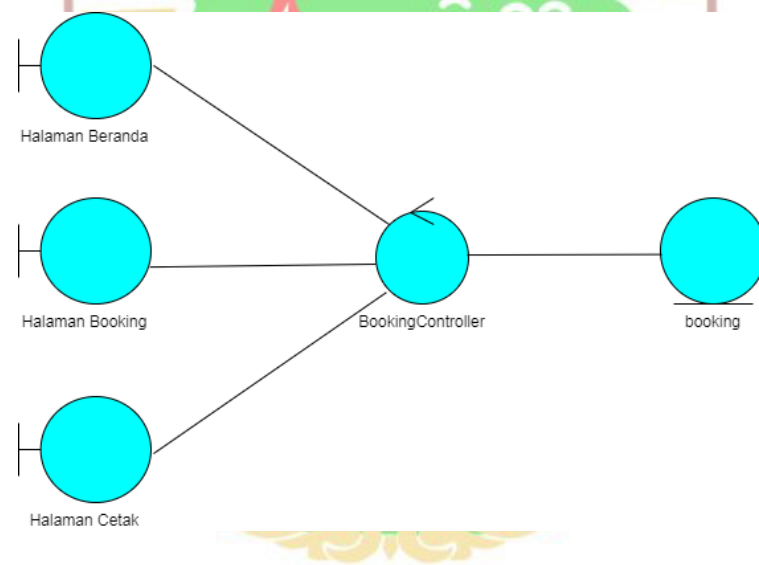
b) Melihat



c) Mengubah

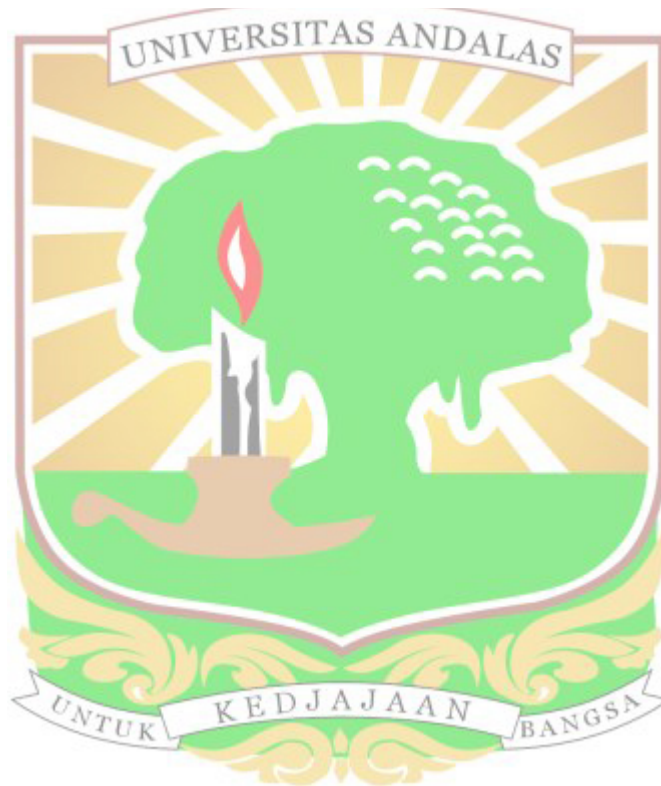
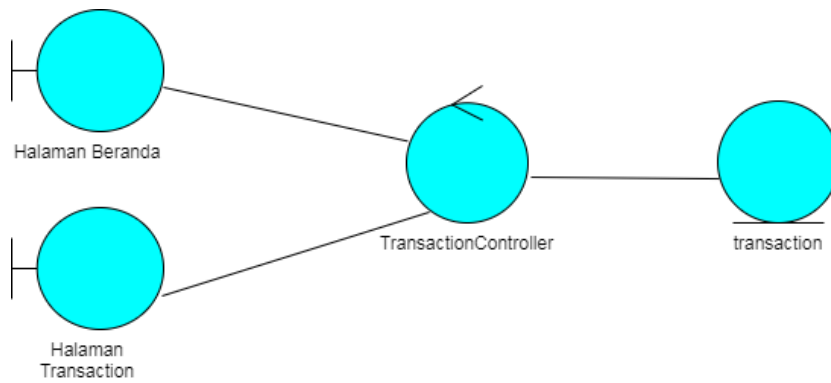


d) Mencetak



C. Pimpinan

1. Pimpinan dapat melihat laporan transaksi





LAMPIRAN D

Lampiran D

Struktur Tabel dan Basis Data Aplikasi

1. Tabel *Users*

Nama Atribut	Tipe Data	Keterangan
id	Bigint	PK
Username	Varchar (255)	
Email	Varchar (255)	
Email_ferified_at	Timestamp	
Password	Varchar (255)	
Remember_token	Varchar (100)	

2. Jasas

Nama Atribut	Tipe Data	Keterangan
Kode_jasa	Varchar (2)	PK
Nama	Varchar (255)	
Deskripsi	Varchar (255)	
Harga	Integer	

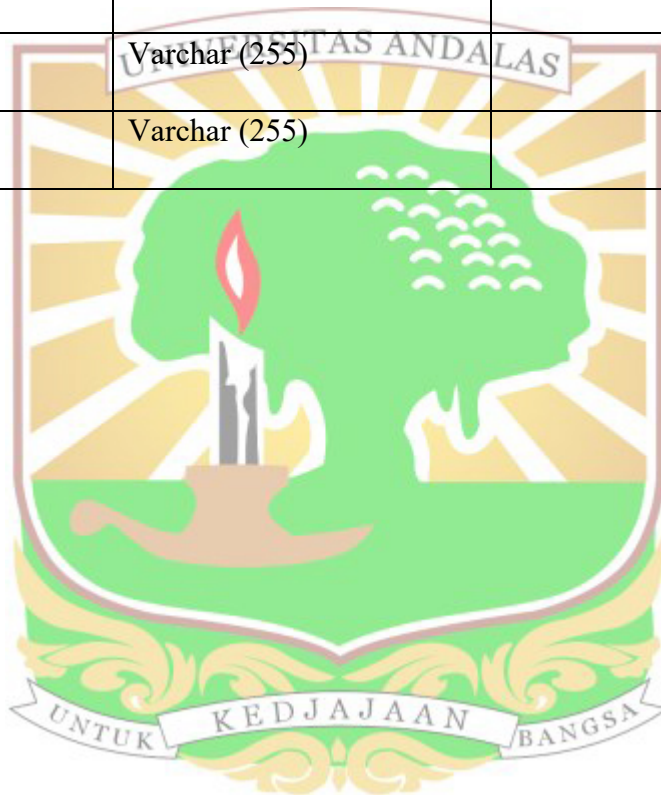
3. Detail_jasas

Nama Atribut	Tipe Data	Keterangan
Kode_jasa	Varchar (2)	PFK
Id_member	Bigint	FK
Id_member	Bigint	

Jumlah	Integer	
Tanggal_take	Date	
Status	Integer	

4. Galleries

Nama Atribut	Tipe Data	Keterangan
Kode_foto	Varchar (4)	PK
Nama	Varchar (255)	
Foto	Varchar (255)	





Lampiran E

User Interface Aplikasi

1. *User Interface Halaman Login User*

The image shows a web browser window titled "Studio Foto Mutiara". The address bar displays "https://mutiarastudiofoto.com". The main content area features a login form with the following elements:

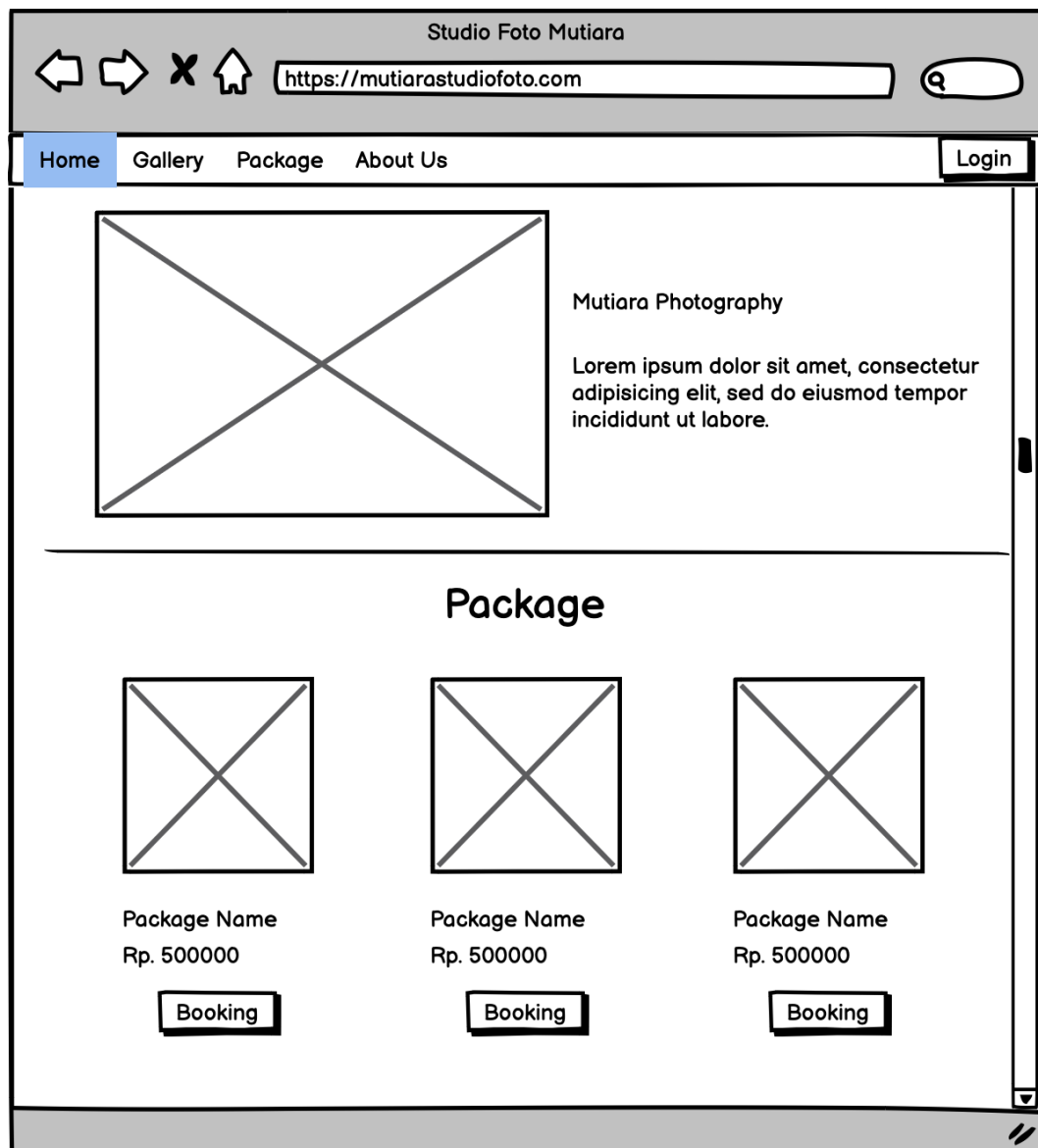
- A placeholder image represented by a square with an 'X' inside.
- The text "Log In" centered above the input fields.
- A text input field labeled "Username".
- A text input field labeled "Password".
- A "Login" button.
- A link below the button that reads "Belum punya akun? Daftar disini."

2. *User Interface Halaman Registrasi*

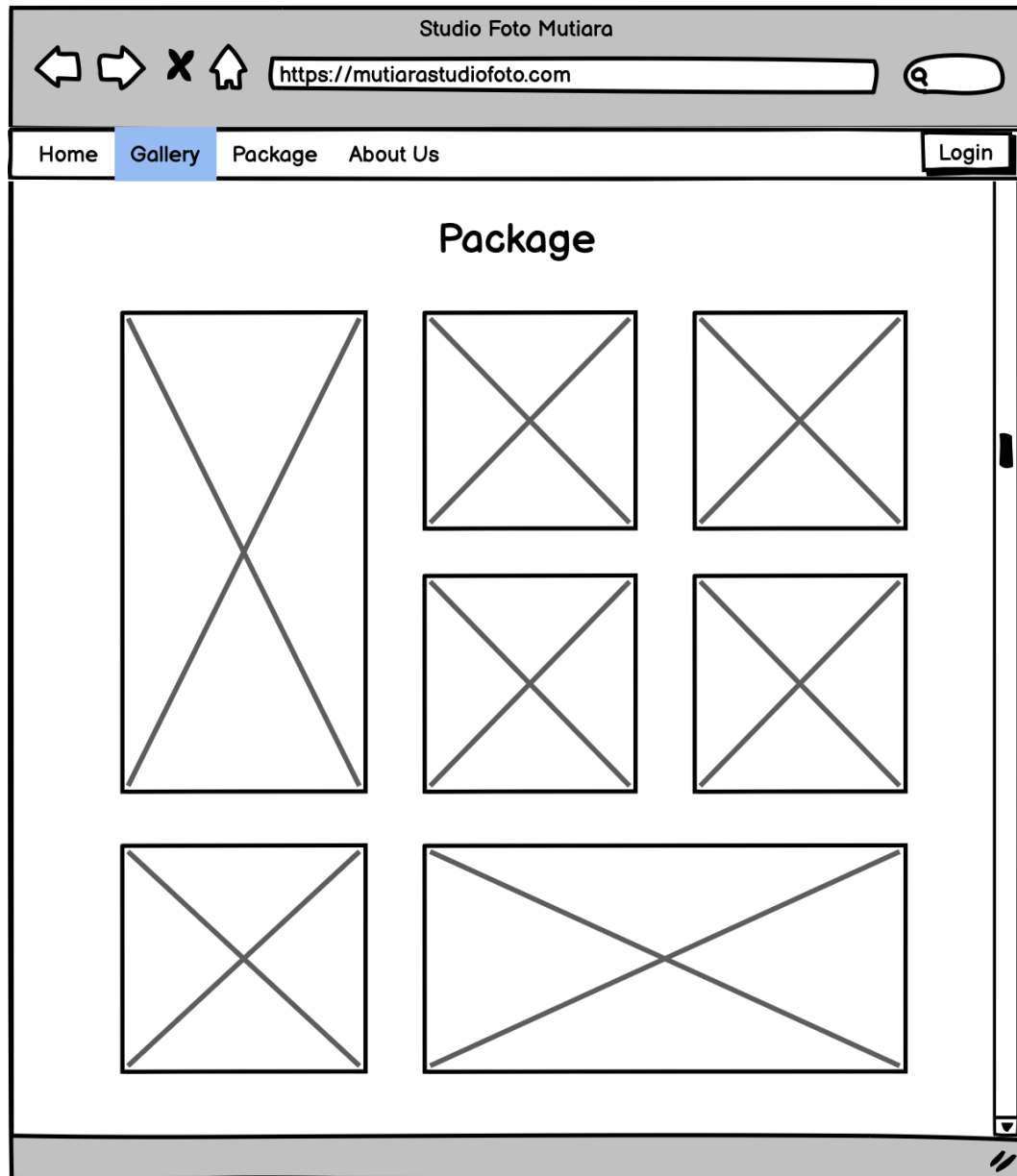
The image shows a web browser window titled "Studio Foto Mutiara". The address bar displays "https://mutiarastudiofoto.com". The main content area features a registration form with the following elements:

- The text "Registrasi" centered above the input fields.
- A text input field labeled "Nama".
- A text input field labeled "No. Hp".
- A text input field labeled "Alamat".
- A text input field labeled "Username".
- A text input field labeled "Password".
- A text input field labeled "Confirm Password".
- A "Sign up" button.
- A link below the button that reads "Sudah punya akun? Login."

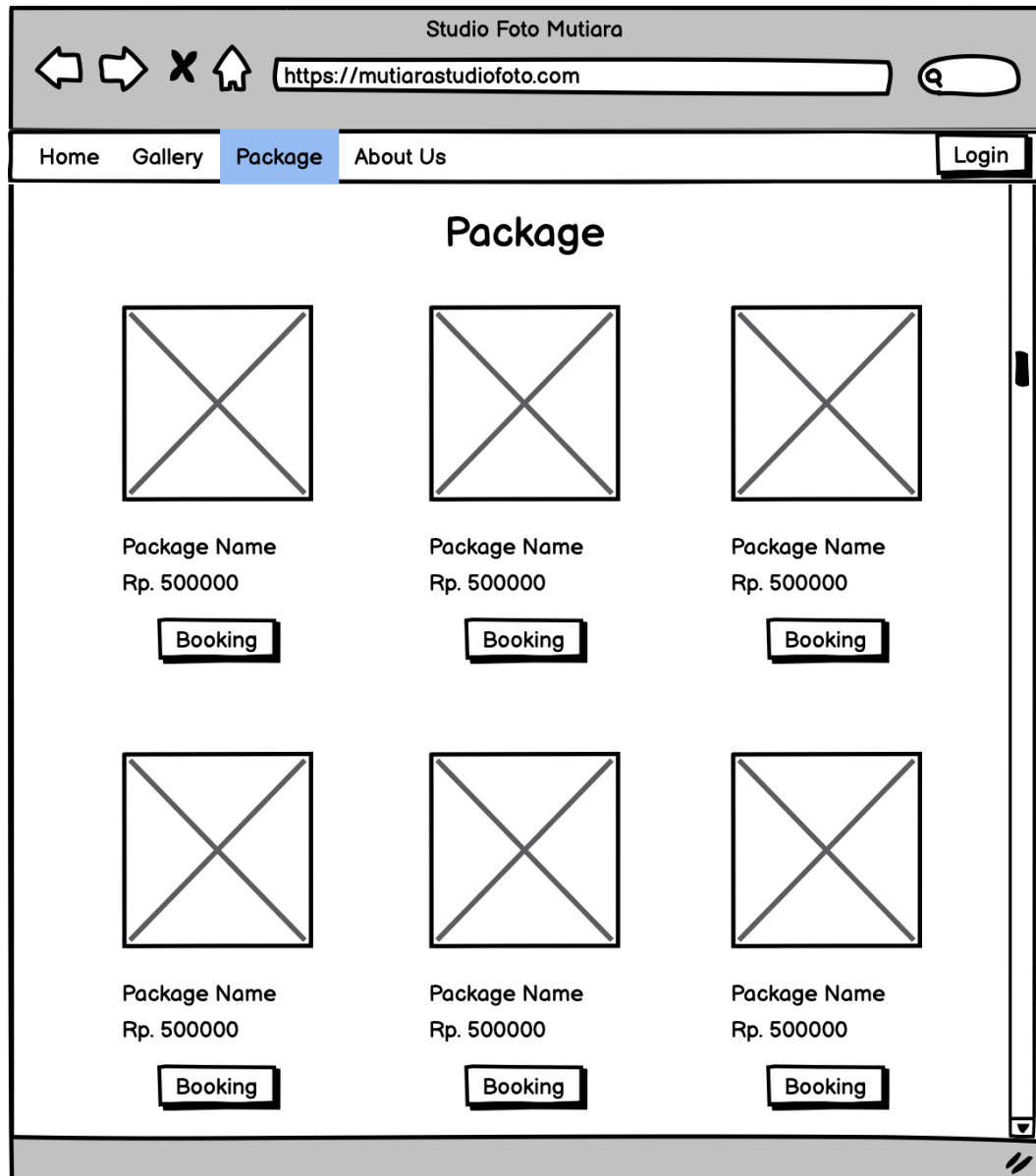
3. User Interface Halaman Home



4. *User Interface* Halaman Galeri



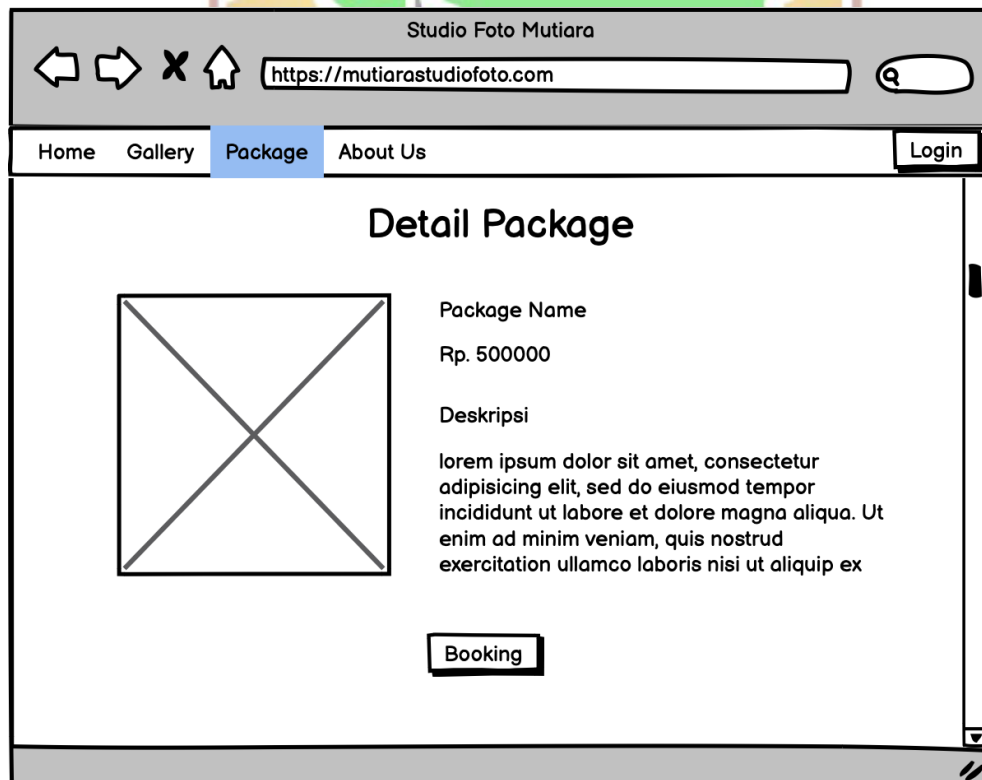
5. User Interface Halaman Packet



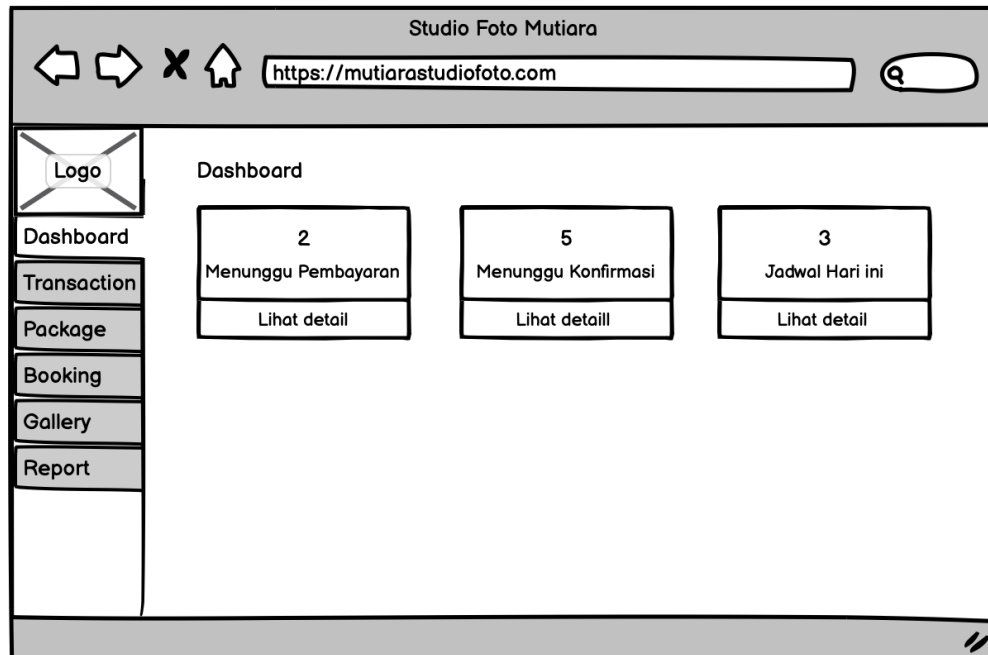
6. User Interface Halaman Tentang Studio Photo Mutiara



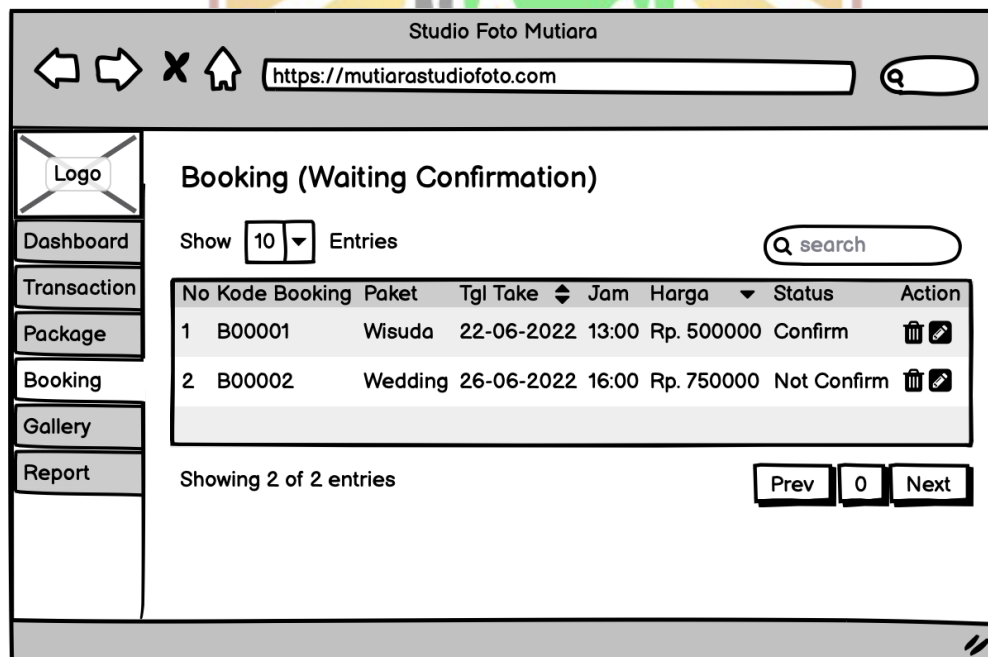
7. User Interface Halaman Detail Paket



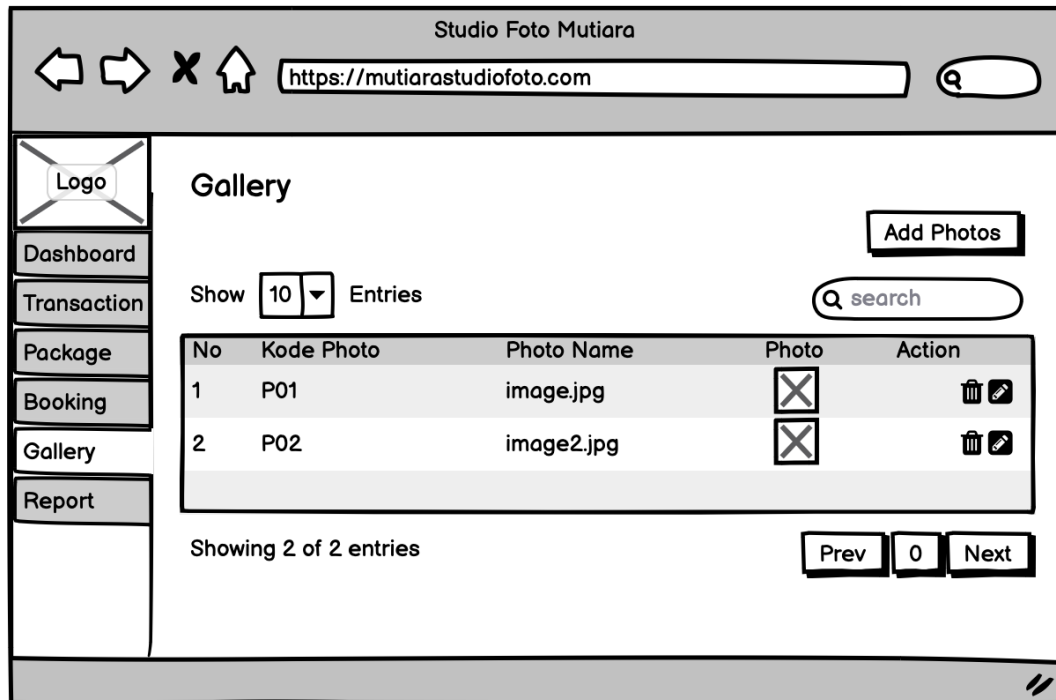
8. User Interface Halaman Beranda Admin



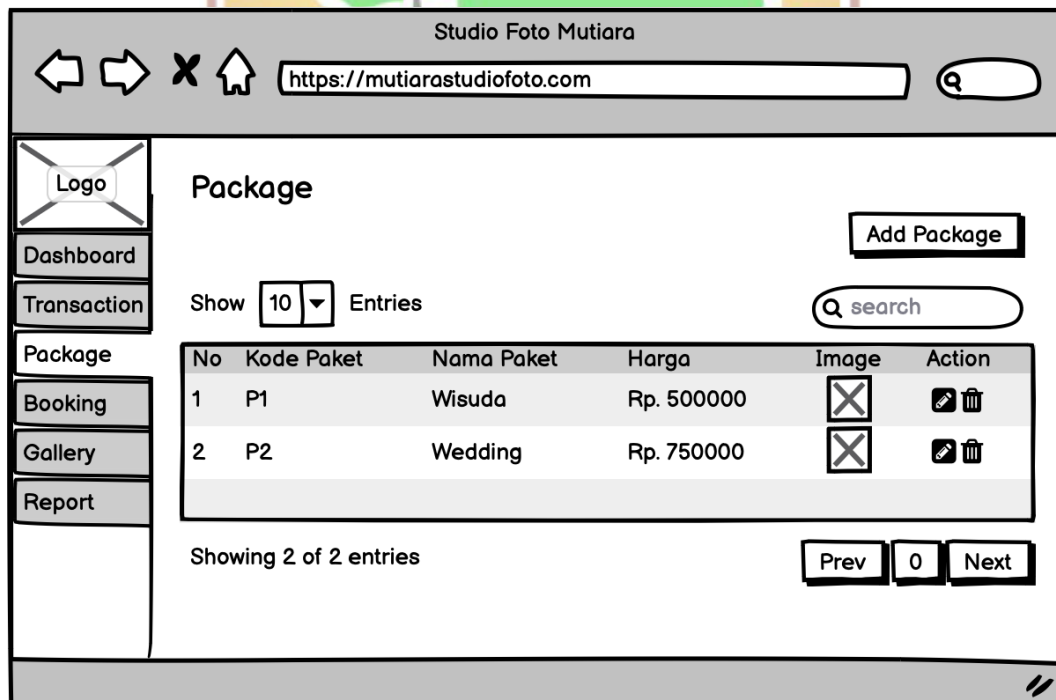
9. User Interface Halaman Mengelola Booking



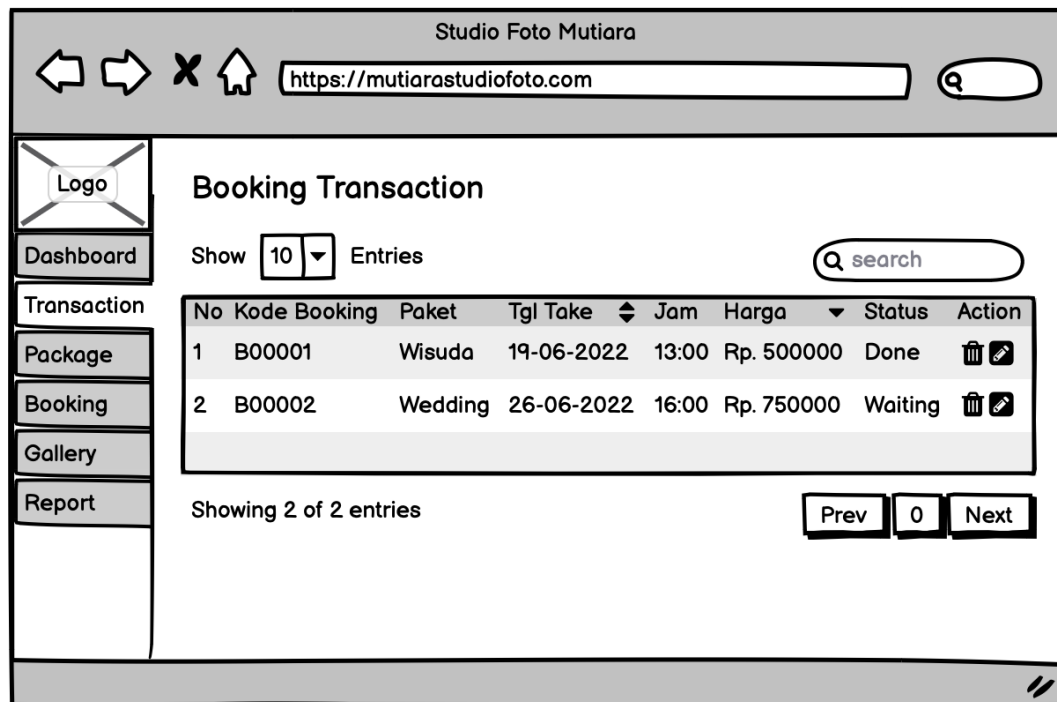
10. User Interface Halaman Mengelola Galeri



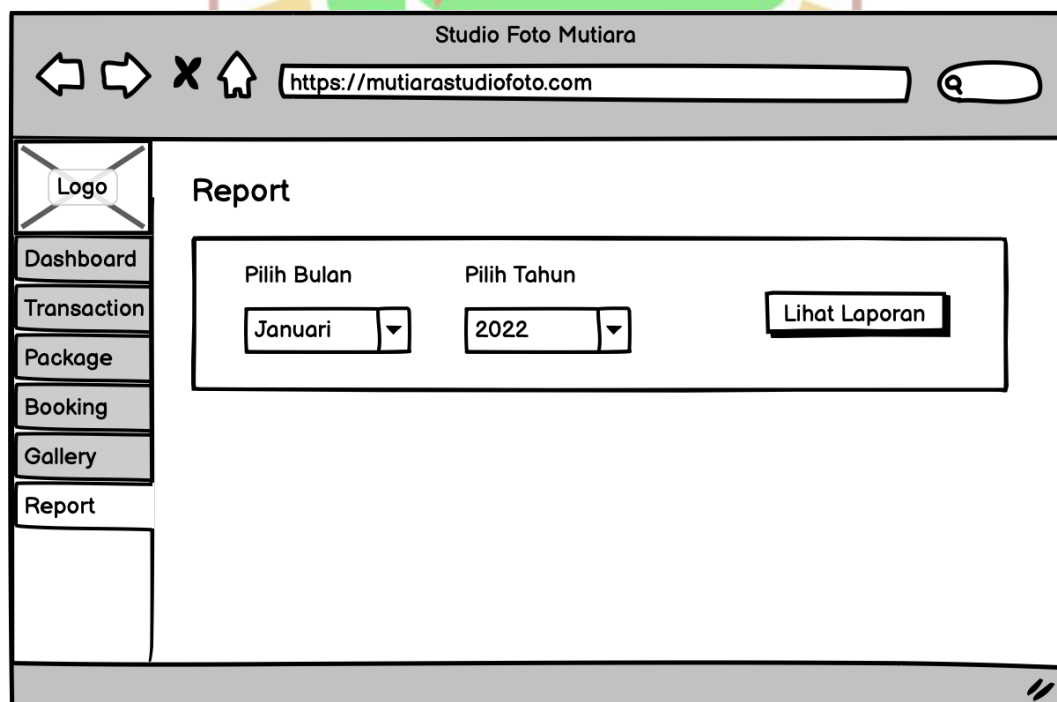
11. User Interface Halaman Mengelola Paket



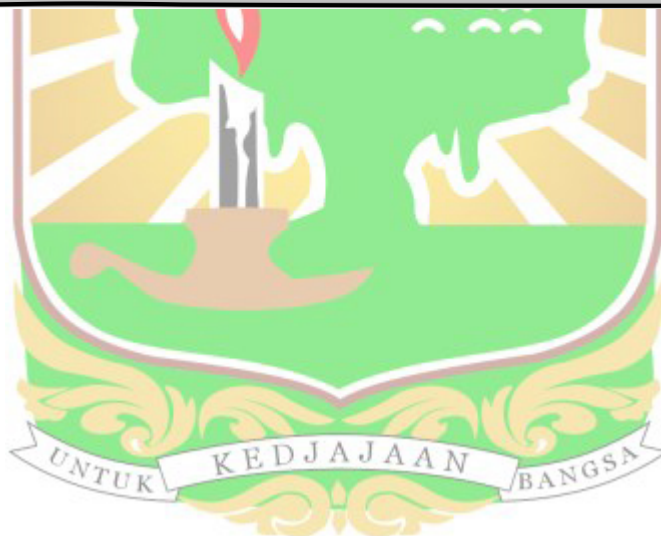
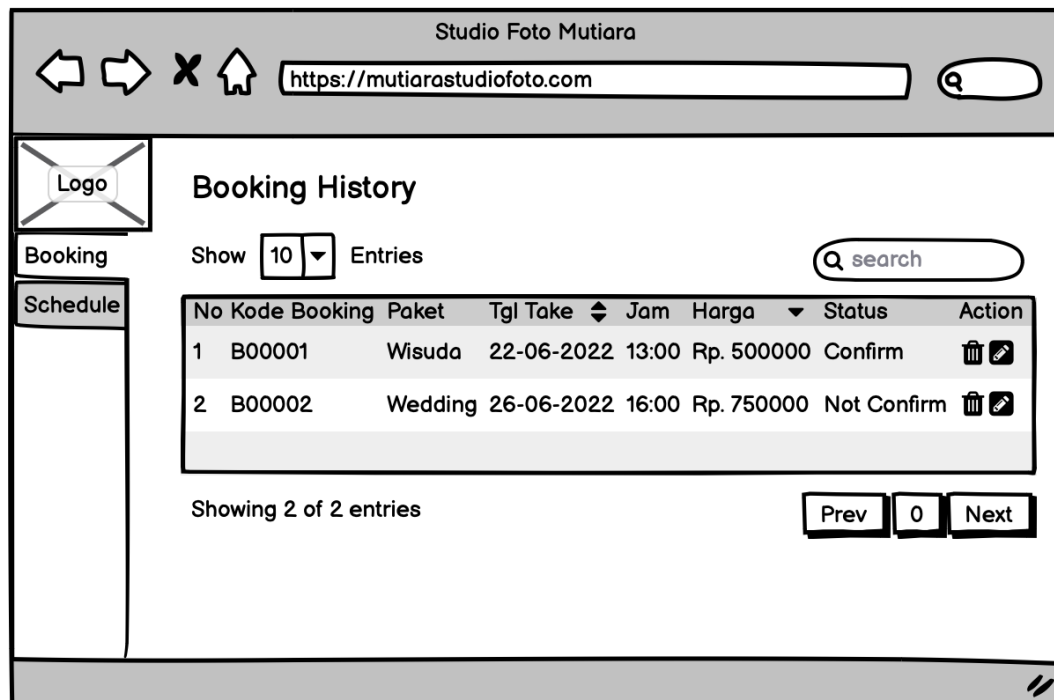
12. User Interface Halaman Mengelola Transaksi



13. User Interface Halaman Mengelola Laporan



14. *User Interface* Halaman Melihat Riwayat Booking





LAMPIRAN F

Lampiran F

Script Kode Program

1. Routing

```
<?php

use App\Model\Galery;

use App\Model\Package;

use App\Tracker;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Route;

Route::get('/', function () {

    //number of user connected or viewed

    Tracker::firstOrCreate([

        'ip' => $_SERVER['REMOTE_ADDR'],

        ['ip' => $_SERVER['REMOTE_ADDR'],

        'current_date' => date('Y-m-d')])->save();

    $packages = Package::all();

    $galery = Galery::all();

    return view('welcome', compact('packages', 'galery'));

})->name('welcome');

Route::group(['middleware' => ['role:admin|owner']], function () {

    Route::get('menunggu',

    'Admin\TransactionController@menungguPembayaran');

    Route::resource('permissions', 'Admin\PermissionsController');

    Route::resource('roles', 'Admin\RolesController');

    Route::resource('users', 'Admin\UsersController');

    Route::get('login-activities',[

        'as' => 'login-activities',
```

```

        'uses' => 'Admin\UsersController@indexLoginLogs'

    });

    Route::resource('packages', 'Admin\PackageController');

    Route::resource('galleries', 'Admin\GaleryController');

    Route::resource('services', 'Admin\JasaController');

    Route::resource('transaction', 'Admin\TransactionController');

    Route::resource('report', 'Admin\ReportController');

    Route::post('/reportresult', 'Admin\ReportController@getReport');

});

Route::group(['middleware' => ['role:member']], function () {

    Route::get('cetak/{id}',
'Member\BookingController@displayReport');

    Route::resource('booking', 'Member\BookingController');

    // Route::get('booking/detail', 'Member\BookingController');

    Route::resource('package', 'Member\PackageController');

    Route::resource('service', 'Member\JasaController');

    Route::get('/about', function(){

        return view('about');

    })->name('about');

    Route::get('/galery', function () {

        $galery = Galery::all();

        return view('galery',compact('galery'));

    })->name('galery');

});

Route::group(['middleware' => ['auth']], function () {

    Route::resource('profile', 'Users\ProfileController');

});

```

```
Auth::routes();

// Route::get('/home', 'HomeController@index')->name('home');

Route::get('/home', function () {

    $galery = Galery::all();

    return view('home',compact('galery'));

})->name('home');
```



2. Controller

a. GaleryController.php

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Model\Galery;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class GaleryController extends Controller
{
    public function index()
    {
        $galleries = Galery::all();

        return view('admin.galery.index', compact('galleries'));
    }

    public function create()
    {
        return view('admin.galery.create');
    }

    public function store(Request $request)
    {
        // dd($request->all());

        $this->validate($request, [

            'kode_foto' => 'required|max:4|unique:galleries',

            'nama' => 'required',

            'foto' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',

        ]);
    }
}

```



```

$fotopath = null;

    if ($request->hasFile('foto')) {

        $fotoname = $request->file('foto')->hashName();

        $fotopath = 'foto/'.$fotoname;

        Storage::disk('public')->put($fotopath,
file_get_contents($request->foto));

    }

    $galery = new Galery;

    $galery->kode_foto = $request->kode_foto;
    $galery->nama = $request->nama;
    $galery->foto = $fotopath;
    $galery->save();

    return redirect()->route('galleries.index')-
>with('success','Data berhasil ditambahkan');

}

public function edit($id)
{
    $galery = Galery::find($id);
    return view('admin.galery.edit',compact('galery'));
}

public function update(Request $request, $id)
{
    $this->validate($request, [

        'nama' => 'required',

    ]);

    $galery = Galery::find($id);

    $galery->nama = $request->nama;

```

```

        if ($request->hasFile('foto')) {

            Storage::disk('public')->delete($galery->foto);

            $fotoname = $request->file('foto')->hashName();

            $fotopath = 'foto/'.$fotoname;

            Storage::disk('public')->put($fotopath,
file_get_contents($request->foto));

            $galery->foto = $fotopath;

        }

        $galery->save();

        return redirect()->route('galleries.index')-
>with('success','Data berhasil diubah');

    }

    public function destroy($id)
    {

        $galery = Galery::find($id);

        Storage::disk('public')->delete($galery->foto);

        $galery->delete();

        return redirect()->route('galleries.index')-
>with('success','Data berhasil dihapus');

    }

}

```



b. PackageController.php

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Model\Booking;
use App\Model\Package;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class PackageController extends Controller
{
    public function index()
    {
        $packages = Package::all();
        return view('admin.package.index', compact('packages'));
    }

    public function create()
    {
        return view('admin.package.create');
    }

    public function store(Request $request)
    {
        $this->validate($request, [

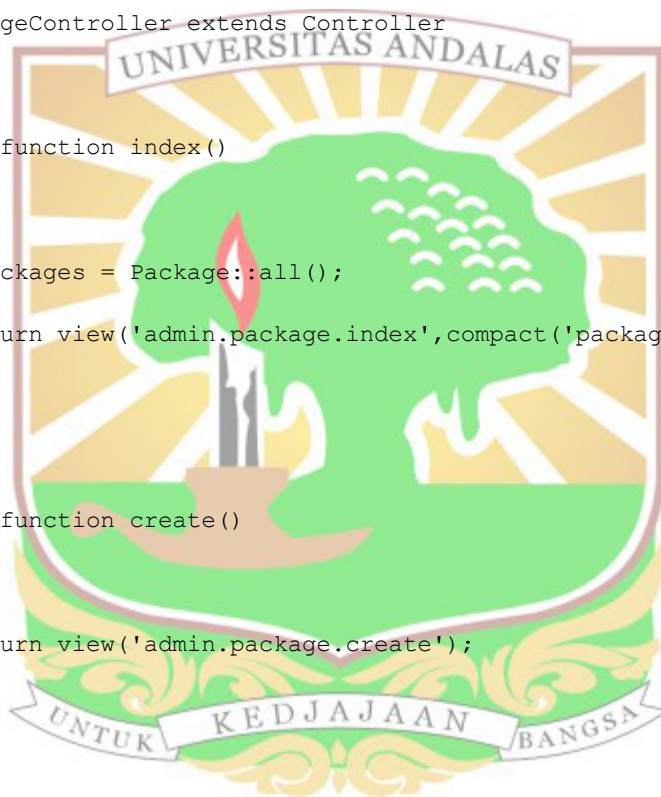
            'kode_paket' => 'required|max:2|unique:packages',

            'nama' => 'required',

            'deskripsi' => 'required',

            'harga' => 'required',

```



```

        'foto'
    'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',

    });

    $fotopath = null;

    if ($request->hasFile('foto')) {

        $fotoname = $request->file('foto')->hashName();

        $fotopath = 'foto/'.$fotoname;

        Storage::disk('public')->put($fotopath,
file_get_contents($request->foto));

    }

    $package = new Package;

    $package->kode_paket = $request->kode_paket;

    $package->nama = $request->nama;

    $package->deskripsi = $request->deskripsi;

    $package->harga = $request->harga;

    $package->foto = $fotopath;

    $package->save();

    return redirect()->route('packages.index')-
>with('success','Data berhasil ditambahkan');

}

public function show($id)
{

    $package = Package::find($id);

    return view('admin.package.show',compact('package'));

}

public function edit($id)
{

    $package = Package::find($id);

```

```

        return view('admin.package.edit', compact('package'));
    }

    public function update(Request $request, $id)
    {
        $this->validate($request, [

            'nama' => 'required',

            'deskripsi' => 'required',

            'harga' => 'required',
        ]);

        $package = Package::find($id);

        $package->kode_paket = $request->kode_paket;
        $package->nama = $request->nama;
        $package->deskripsi = $request->deskripsi;
        $package->harga = $request->harga;

        if ($request->hasFile('foto')) {
            Storage::disk('public')->delete($package->foto);

            $fotiname = $request->file('foto')->hashName();

            $fotopath = 'foto/'.$fotiname;

            Storage::disk('public')->put($fotopath,
            file_get_contents($request->foto));

            $package->foto = $fotopath;
        }

        $package->save();

        return redirect()->route('packages.index')-
        >with('success','Data berhasil diubah');
    }

    public function destroy($id)

```

```

{

    $package = Package::find($id);

    Storage::disk('public')->delete($package->foto);

    $package->delete();

    return redirect()->route('packages.index')-
>with('danger','Data berhasil dihapus');

}

}

```

c. *ReportController.php*



```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\DB;

use Response;

use App\Model\Booking;

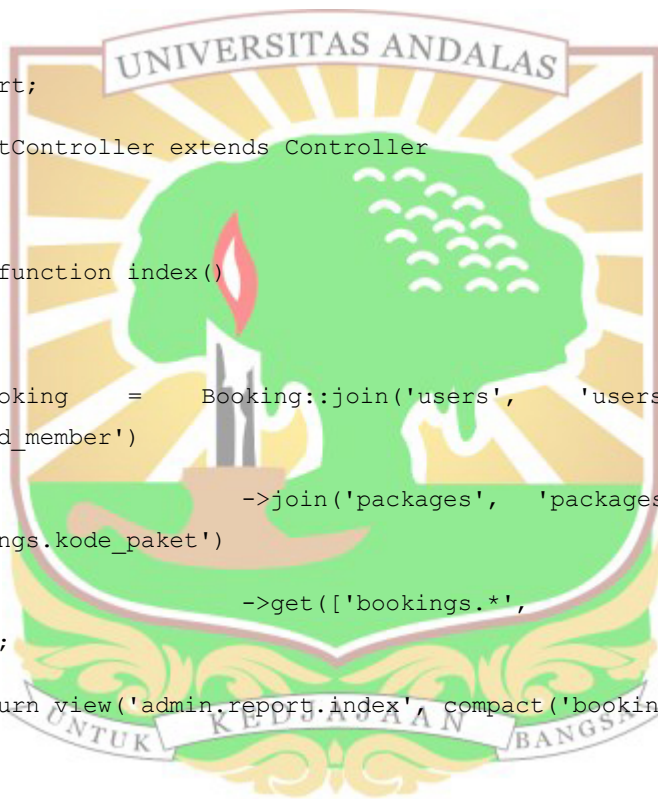
use Illuminate\Http\Request;

use PdfReport;

class ReportController extends Controller
{
    public function index()
    {
        $booking = Booking::join('users', 'users.id', '=',
'bookings.id_member')
->join('packages', 'packages.kode_paket',
'=', 'bookings.kode_paket')
->get(['bookings.*', 'packages.*',
'users.*']);
        return view('admin.report.index', compact('booking'));
    }

    public function getReport(Request $request){
        if($request->tipe == "harian"){
            $data = Booking::join('users', 'users.id', '=',
'bookings.id_member')
->join('packages', 'packages.kode_paket',
'=', 'bookings.kode_paket')

```



```

                                ->where('bookings.tanggal_take', '=',
$request->range)

                                ->get(['bookings.*', 'packages.*',
'users.*']);

        } elseif ($request->type == "bulanan") {

            $pieces = explode("-", $request->range);

            $data = Booking::join('users', 'users.id', '=',
'bookings.id_member')

                                ->join('packages', 'packages.kode_paket',
'=', 'bookings.kode_paket')

                                ->whereYear('bookings.tanggal_take',
$pieces[0])

                                ->whereMonth('bookings.tanggal_take',
$pieces[1])

                                ->get(['bookings.*', 'packages.*',
'users.*']);

        } elseif ($request->type == "tahunan") {

            $data = Booking::join('users', 'users.id', '=',
'bookings.id_member')

                                ->join('packages', 'packages.kode_paket',
'=', 'bookings.kode_paket')

                                ->whereYear('bookings.tanggal_take',
$request->range)

                                ->get(['bookings.*', 'packages.*',
'users.*']);

        }

        return response()->json(array('data'=> $data), 200);

    }

    public function displayReport(Request $request, $id)

    {

        dd($id);

        $fromDate = $request->input('from_date');

```

```

    $toDate = $request->input('to_date');

    $sortBy = $request->input('sort_by');

    $title = 'Registered User Report'; // Report title

    $meta = [ // For displaying filters description on header

        'Registered on' => $fromDate . ' To ' . $toDate,

        'Sort By' => $sortBy

    ];

    $queryBuilder = User::select(['name', 'balance', 'registered_at'])
    // Do some querying..

        ->whereBetween('registered_at', [$fromDate,
$toDate])

        ->orderBy($sortBy);

    $columns = [ // Set Column to be displayed

        'Name' => 'name',

        'Registered At', // if no column_name specified, this will
        automatically search for snake_case of column name (will be
        registered_at) column from query result

        'Total Balance' => 'balance',

        'Status' => function($result) { // You can do if statement or
        any action do you want inside this closure

            return ($result->balance > 100000) ? 'Rich Man' : 'Normal
        Guy';

        }

    ];

    return PdfReport::of($title, $meta, $queryBuilder, $columns)

        ->editColumn('Registered At', [ // Change column
        class or manipulate its data for displaying to report

```

```

        'displayAs' => function($result) {

            return $result->registered_at->format('d M
Y');

        },

        'class' => 'left'

    ])

    ->editColumns(['Total Balance', 'Status'], [ //
Mass edit column

        'class' => 'right bold'

    ])

    ->showTotal([ // Used to sum all value on specified
column on the last table (except using groupBy method). 'point' is a
type for displaying total with a thousand separator

        'Total Balance' => 'point' // if you want to
show dollar sign ($) then use 'Total Balance' => '$'

    ])

    ->limit(20) // Limit record to be showed

    ->stream(); // other available method:
store('path/to/file.pdf') to save to disk, download('filename') to
download pdf / make() that will producing DomPDF / SnappyPdf instance
so you could do any other DomPDF / snappyPdf method such as stream() or
download()

}

}

```

d. RulesController.php

```

<?php

namespace App\Http\Controllers\Admin;

use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Gate;
use App\Http\Controllers\Controller;

class RolesController extends Controller
{
    /**
     *
     * allow admin only
     *
     */

    public function __construct() {
        $this->middleware('role:admin');
    }

    /**
     * Display a listing of Role.
     *
     * @return \Illuminate\Http\Response
     */

    public function index()
    {
        $roles = Role::all();
    }
}

```



```

        return view('admin.roles.index', compact('roles'));
    }

    /**
     * Show the form for creating new Role.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $permissions = Permission::get()->pluck('name', 'name');

        return view('admin.roles.create', compact('permissions'));
    }

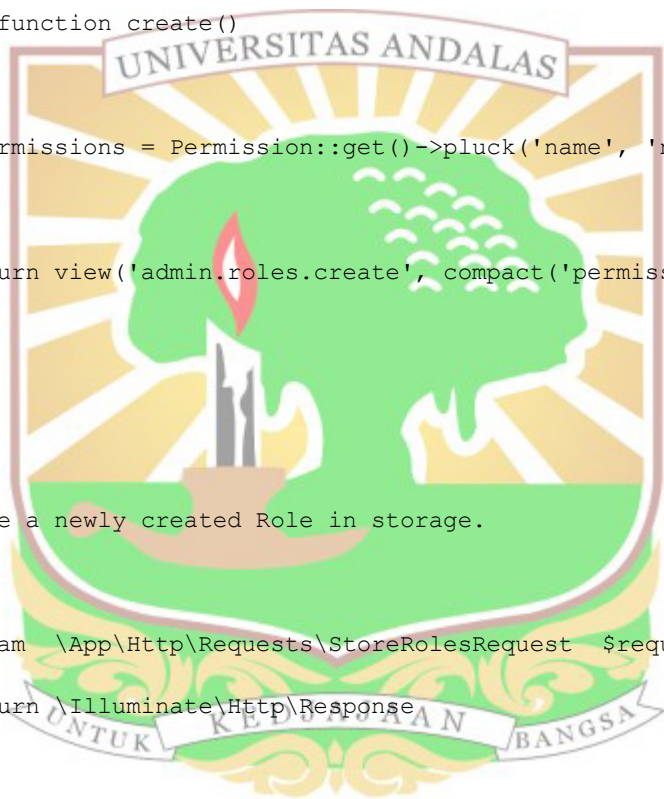
    /**
     * Store a newly created Role in storage.
     *
     * @param \App\Http\Requests\StoreRolesRequest $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([

            'name' => ['required', 'unique:roles', 'max:20'],

            'permission' => ['required'],

        ]);
    }

```



```

        $role = Role::create($request->except('permission'));

        $permissions = $request->input('permission') ? $request->
>input('permission') : [];

        $role->givePermissionTo($permissions);

        return redirect()->route('roles.index')->with('success', "The
$role->name was saved successfully");

    }

/**
 * Show the form for editing Role.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $permissions = Permission::get()->pluck('name', 'name');

    $role = Role::findOrFail($id);

    $data = $role->permissions()->pluck('name');

    $selectedPermissions = $data[0];

    return view('admin.roles.edit', compact('role',
'permissions','selectedPermissions'));
}

/**
 * Update Role in storage.
 *
 * @param \App\Http\Requests\UpdateRolesRequest $request
 * @param int $id

```

```

    * @return \Illuminate\Http\Response

    */

public function update(Request $request, $id)

{

    $request->validate([

        'name' => ['required','unique:roles','max:20'],

        'permission' => ['required'],

    ]);

    $role = Role::findOrFail($id);

    $role->update($request->except('permission'));

    $permissions = $request->input('permission') ? $request->input('permission') : [];

    $role->syncPermissions($permissions);

    return redirect()->route('admin.roles.index')->with('warning',

"The $role->name was updated successfully");

}

/**

 * Remove Role from storage.

 *

 * @param int $id

 * @return \Illuminate\Http\Response

 */

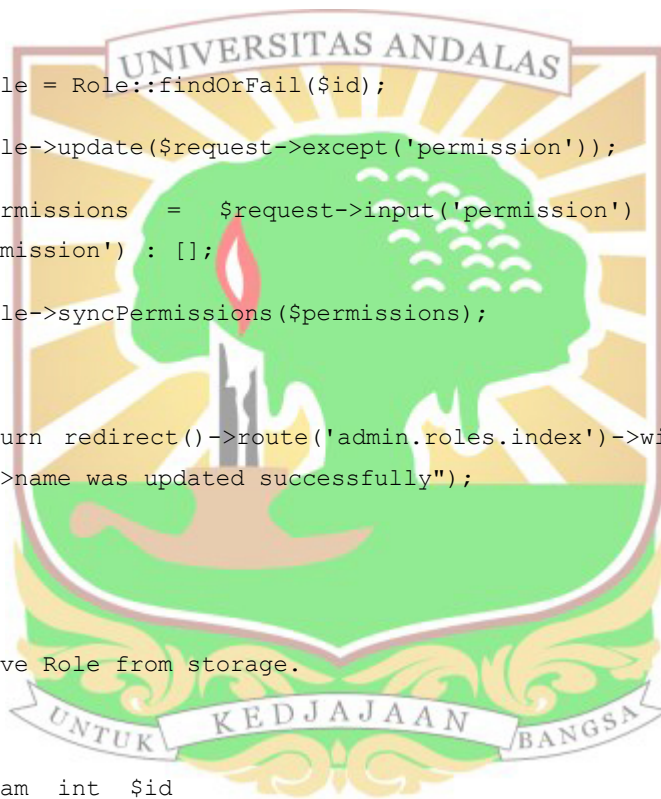
public function destroy($id)

{

    $role = Role::findOrFail($id);

    $role->delete();

```



```

        return redirect()->route('roles.index')->with('danger', "The
        $role->name was deleted successfully");

    }

    /**
     * Delete all selected Role at once.
     *
     * @param Request $request
     */
    public function massDestroy(Request $request)
    {
        if ($request->input('ids')) {
            $entries = Role::whereIn('id', $request->input('ids'))-
            >get();

            foreach ($entries as $entry) {
                $entry->delete();
            }
        }
    }
}

```



e. *TransactionController.php*

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\DB;


use App\Model\Booking;

use App\Model\Package;


use Illuminate\Http\Request;
class TransactionController extends Controller
{
    public function index()
    {
        $booking = Booking::join('users', 'users.id', '=',
'bookings.id_member')
->join('packages', 'packages.kode_paket',
'=', 'bookings.kode_paket')
->get(['bookings.*', 'packages.*',
'users.*']);

        return view('admin.transactions.index', compact('booking'));
    }

    public function show($id)
    {
        $booking = Booking::find($id);

        $package = Package::find($booking->kode_paket);

        return
view('admin.transactions.show', compact('booking', 'package'));
    }

    public function menungguPembayaran()

```

```

{
    $booking = Booking::join('users', 'users.id', '=',
'bookings.id_member')

                                ->join('packages', 'packages.kode_paket',
'=', 'bookings.kode_paket')

                                ->get();

    return view('admin.transactions.index', compact('booking'));
}

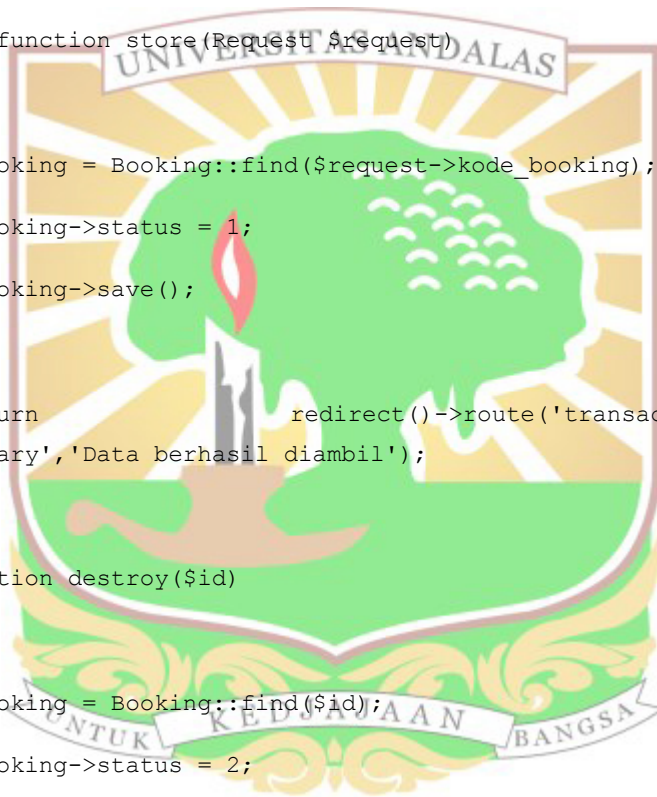
public function store(Request $request)
{
    $booking = Booking::find($request->kode_booking);
    $booking->status = 1;
    $booking->save();

    return redirect()->route('transaction.index')-
>with('primary','Data berhasil diambil');
}

public function destroy($id)
{
    $booking = Booking::find($id);
    $booking->status = 2;
    $booking->save();

    return redirect()->route('transaction.index')-
>with('danger','Data berhasil diambil');
}
}

```



f. UsersController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\User;
use Spatie\Permission\Models\Role;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Gate;
use App\Http\Controllers\Controller;
use Illuminate\Validation\Rule;
use App\UserLoginLog;
class UsersController extends Controller
{
    /**
     *
     * allow admin only
     *
     */
    public function __construct() {
        //$this->middleware(['role:admin|creator']);

        $this->middleware(['role:admin']);
    }

    /**
     * Display a listing of User.
     *
     * @return \Illuminate\Http\Response
```

```

*/

public function index()

{

    $users = User::paginate(10);

    return view('admin.users.index', compact('users'));

}

/**
 * Display a listing of User.
 *
 * @return \Illuminate\Http\Response
 */
public function indexLoginLogs()
{
    $userLoginActivities = UserLoginLog::paginate(10);

    return view('admin.activity.logs',
compact('userLoginActivities'));
}

/**
 * Show the form for creating new User.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $roles = Role::get()->pluck('name', 'name');

```



```

        return view('admin.users.create', compact('roles'));
    }

    /**
     * Store a newly created User in storage.
     *
     * @param \App\Http\Requests\StoreUsersRequest $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255',
'unique:users'],
            'mobile' => ['required', 'numeric', 'digits:10',
'unique:users'],
            'password' => ['required', 'min:5'],
            'roles.*' => ['required']
        ]);

        $user = User::create(array_merge($request->all(), ['password' =>
bcrypt($request->password)]));

        $roles = $request->input('roles') ? $request->input('roles') :
[];

        $user->assignRole($roles);

        return redirect()->route('users.index')->with('success', "The
$user->name was saved successfully");
    }

```

```

}

/**
 * Show the form for editing User.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $roles = Role::get()->pluck('name', 'name');

    $user = User::findOrFail($id);

    $data = $user->roles()->pluck('name');
    $selectedRoles = $data[0];

    return view('admin.users.edit', compact('user',
'roles', 'selectedRoles'));
}

/**
 * Update User in storage.
 *
 * @param \App\Http\Requests\UpdateUsersRequest $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)

```



```

{
    $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255',
Rule::unique('users')->ignore($id)],
        'mobile' => ['required', 'numeric', 'digits:10',
Rule::unique('users')->ignore($id)],
        'password' => ['sometimes', 'nullable', 'min:5'],
        'roles.*' => ['required']
    ]);

    $user = User::findOrFail($id);
    $user->update(array_merge($request->all(), ['password' =>
bcrypt($request->password)]));
    $roles = $request->input('roles') ? $request->input('roles') :
[];
    $user->syncRoles($roles);

    return redirect()->route('users.index')->with('warning', "The
$user->name was updated successfully");
}
/**
 * Remove User from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $user = User::findOrFail($id);

```

```

$user->delete();

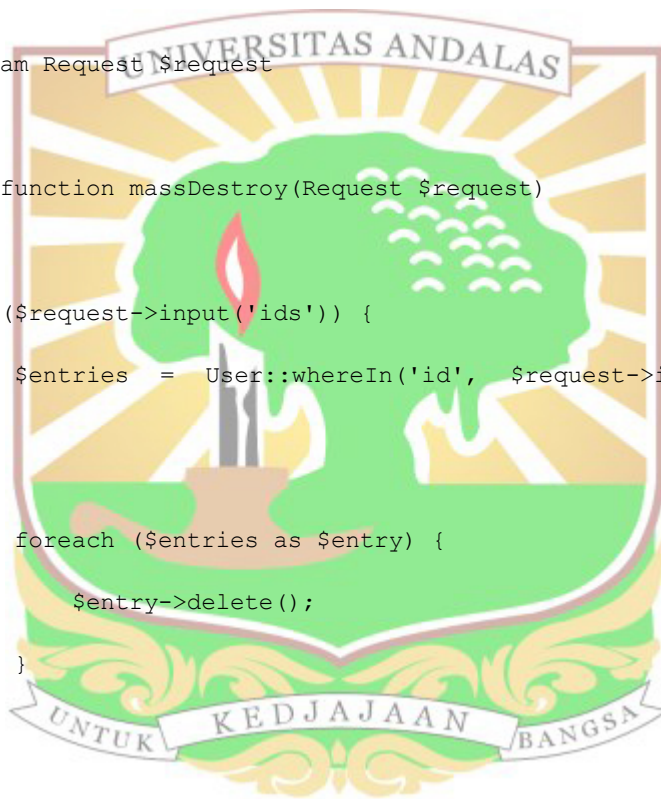
return redirect()->route('users.index')->with('danger', "The
$user->name was deleted successfully");

}

/**
 * Delete all selected User at once.
 *
 * @param Request $request
 */
public function massDestroy(Request $request)
{
    if ($request->input('ids')) {
        $entries = User::whereIn('id', $request->input('ids'))->get();

        foreach ($entries as $entry) {
            $entry->delete();
        }
    }
}
}

```



Model

a. *Booking*

```

<?php

namespace App\Model;

use Illuminate\Database\Eloquent\Model;

class Booking extends Model
{
    protected $table = 'bookings';

    protected $primaryKey = 'kode_booking';
    public $incrementing = false;

    protected $keyType = 'string';

    protected $fillable = ['kode_booking', 'kode_paket', 'id_member',
'tanggal_take', 'jam_take', 'status'];

    public static function bookingCount(){
        return Booking::count();
    }

    public static function terimaCount(){
        return Booking::where('status', '=', '1')->count();
    }

    public static function belumCount(){
        return Booking::where('status', '=', '0')->count();
    }
}

```

b. Package

```
<?php

namespace App\Model;

use Illuminate\Database\Eloquent\Model;

class Package extends Model
{
    protected $table = 'packages';

    protected $primaryKey = 'kode_paket';
    public $incrementing = false;

    protected $keyType = 'string';

    protected $fillable = ['kode_paket', 'nama', 'deskripsi',
    'harga', 'foto'];
}
```





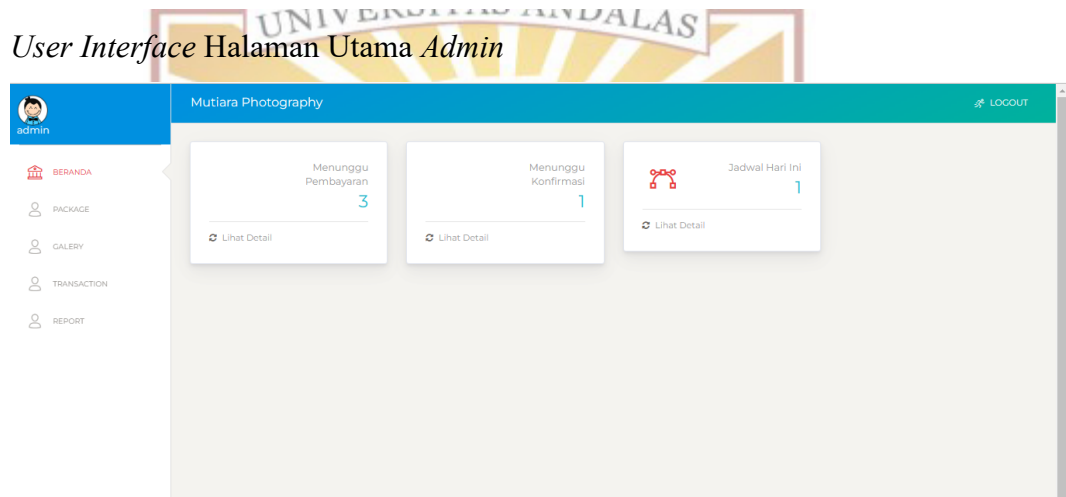
Implementasi *User Interface* Aplikasi

1. *User Interface* Halaman Login User



The screenshot shows a login form for 'Mutiara Photography'. The form has a title 'Login' and two input fields: 'Nama Pengguna' (Username) and 'Kata Sandi' (Password). Below the password field is a 'LOGIN' button and a link that says 'BELUM PUNYA AKUN? DAFTAR' (Don't have an account? Register).

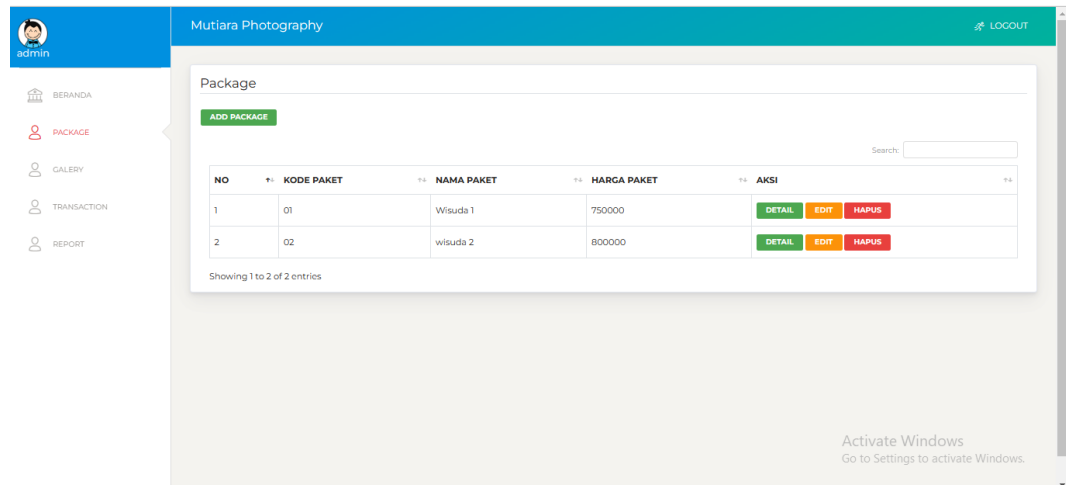
2. *User Interface* Halaman Utama Admin



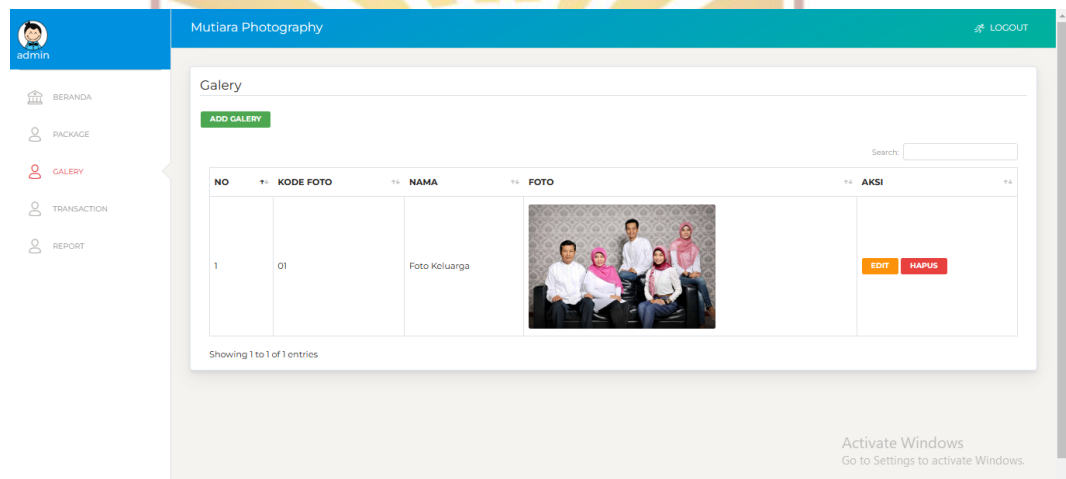
3. *User Interface* Halaman Utama Customer



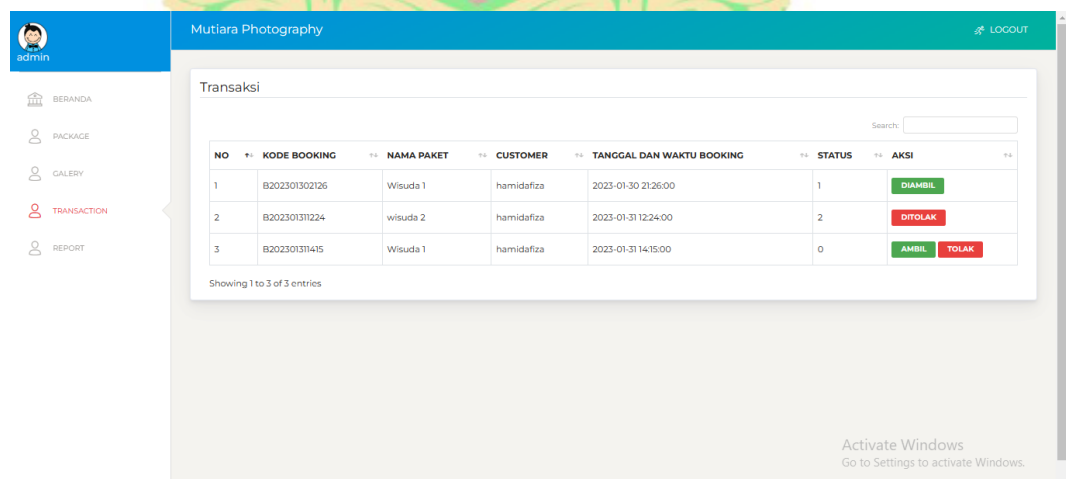
4. User Interface Halaman Mengelola Package



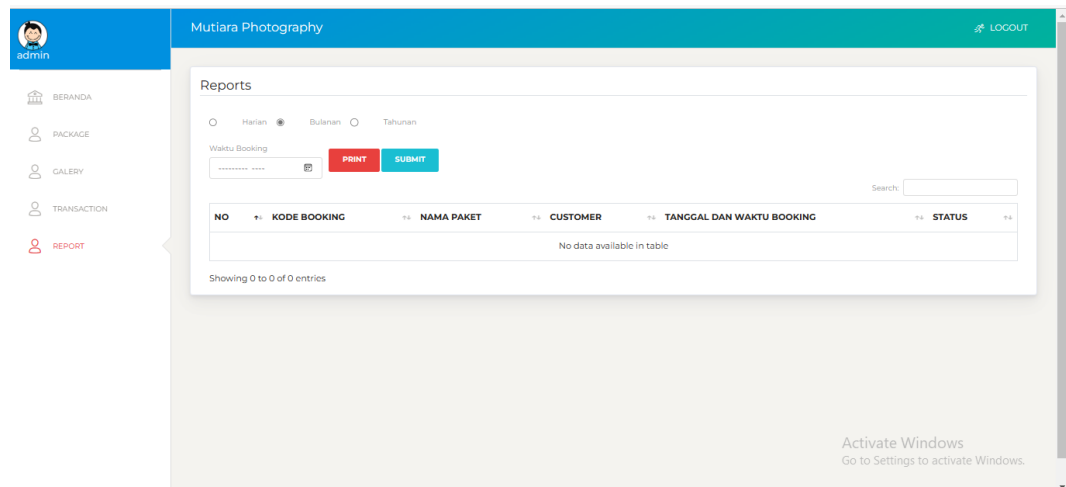
5. User Interface Mengelola Halaman Galeri



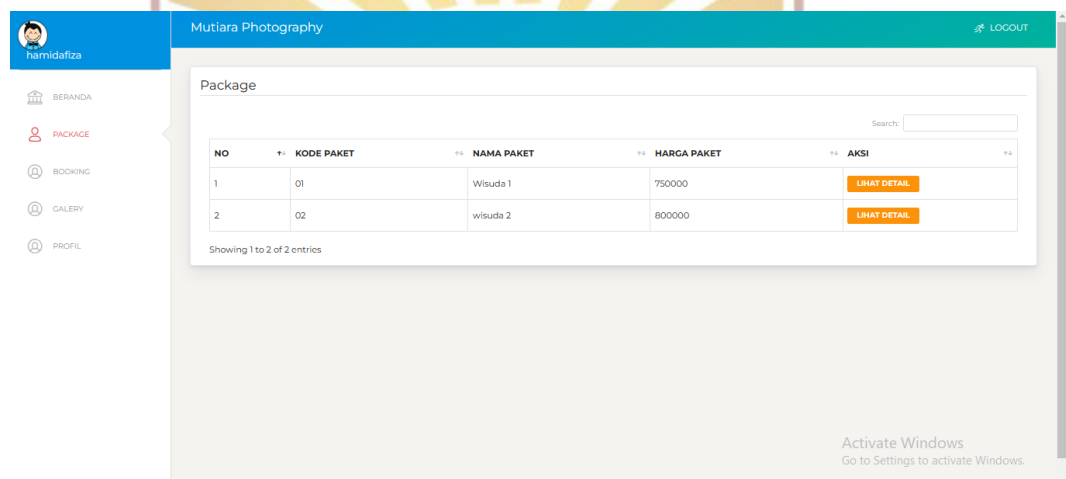
6. User Interface Mengelola Halaman Transaction



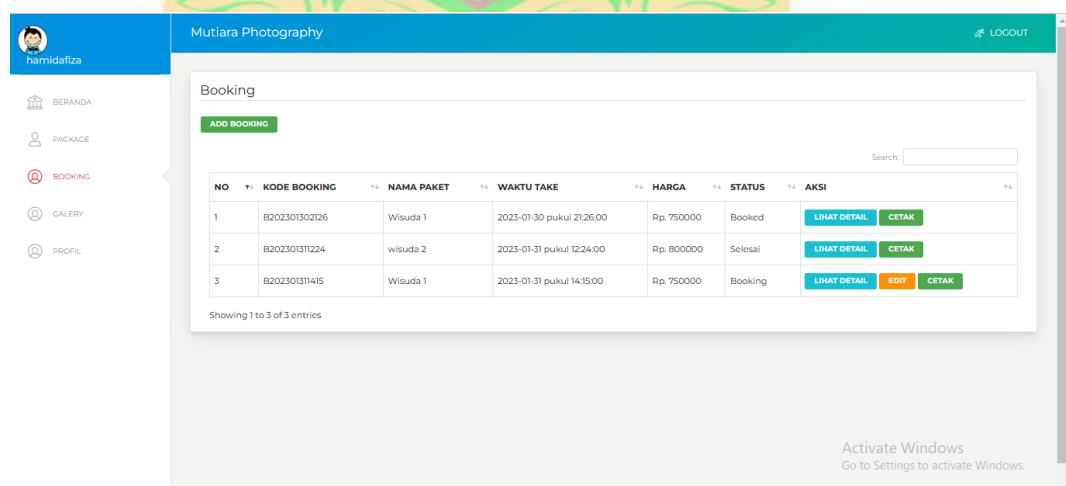
7. User Interface Mengelola Halaman Report



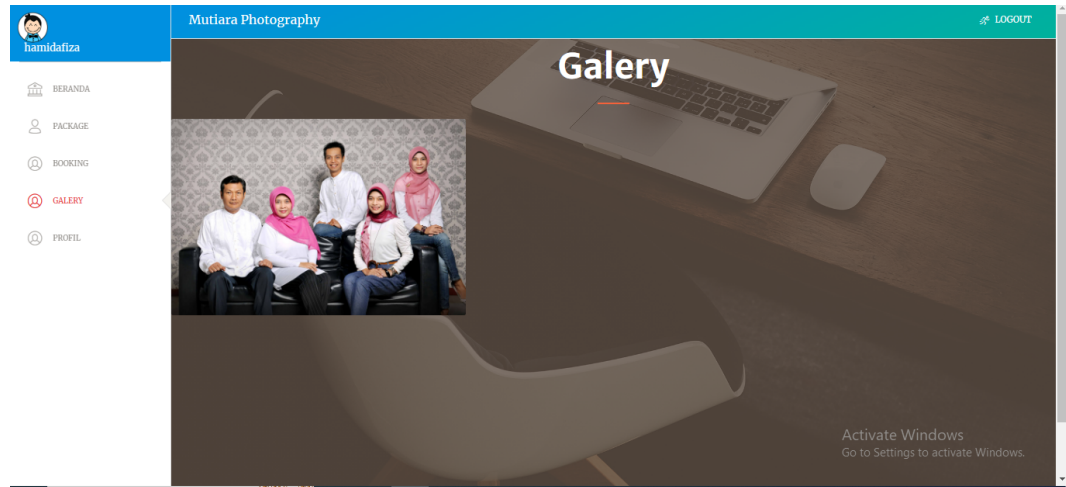
8. User Interface Melihat Daftar Package



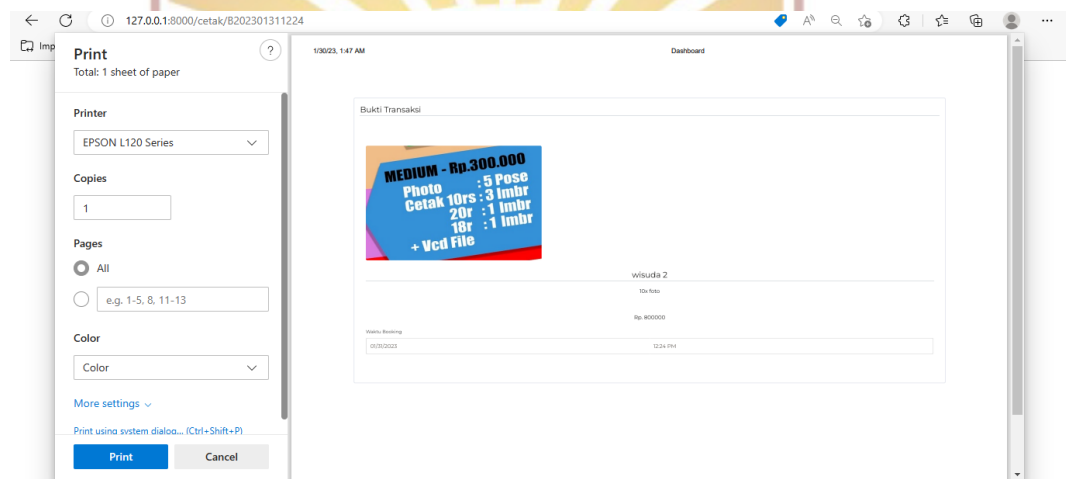
9. User Interface Mengelola Booking



10. User Interface Melihat Galery



11. User Interface Halaman Mencetak Riwayat Booking





LAMPIRAN H

Form Pengujian Sistem

Nama : Pembangunan Sistem Informasi Pengelolaan Transaksi Berbasis Web Pada Mutiara Studio Photo Padang

Nama Penguji :

Jabatan :

Pengujian Aplikasi sebagai Admin:

No	Fungsional	Scenario pengujian	Hasil yang diharapkan	Sesuai	Tidak Sesuai
1	Login	Masuk melalui web, mengisi username dan password admin	Sistem bisa masuk dalam user admin	✓	
2	Mengelola Paket	Memilih menu paket pada sidebar	Sistem menampilkan halaman paket	✓	
3	Tambah Paket	Menekan tombol add paket dan mengisi from tambah paket pada halaman tambah paket	Sistem menampilkan paket berhasil ditambahkan pada halaman paket	✓	
4	Mengelola Galeri	Memilih menu galeri pada sidebar	Sistem menampilkan halaman galeri	✓	
5	Tambah Galeri	Menekan tombol add galeri dan mengisi from tambah galeri pada halaman tambah galeri	Sistem menampilkan galeri berhasil ditambahkan pada halaman galeri	✓	
6	Mengelola Transaksi	Menekan menu transaksi pada sidebar, kemudian mengklik tombol ambil untuk mengkonfirmasi pesanan dari customer	Sistem menampilkan halaman transaksi dan menampilkan halaman detail booking untuk mengkonfirmasi pesanan customer	✓	
7	Melihat Laporan Transaksi	Memilih menu report pada sidebar, dan mengisi waktu booking, kemudian mengklik tombol submit	Sistem menampilkan halaman laporan, dan menampilkan hasil pemesanan pada waktu yang telah dipilih	✓	

Padang, Januari 2023

Penguji

Agum Sadewa

(Agum Sadewa)

Form Pengujian Sistem

Nama : Pembangunan Sistem Informasi Pengelolaan Transaksi Berbasis Web Pada Mutiara Studio Photo Padang

Nama Penguji :

Jabatan :

Pengujian Aplikasi sebagai Customer:

No	Fungsional	Scenario pengujian	Hasil yang diharapkan	Sesuai	Tidak Sesuai
1	Registrasi	Menginputkan data pada form registrasi, kemudian mengklik tombol submit	Sistem menampilkan halaman login sebagai customer	✓	
2	Memilih paket	Memilih menu paket pada sidebar dan mengklik tombol lihat detail paket untuk melakukan pemesanan	Sistem menampilkan halaman paket yang tersedia, dan menampilkan halaman detail paket	✓	
3	Mengelola booking	Memilih menu booking, dan mengklik tombol add booking untuk menambah booking	Sistem menampilkan halaman Kelola booking dan menampilkan halaman paket untuk menambah booking paket	✓	
4	Mencetak Riwayat booking	Mengklik tombol cetak pada halaman booking	Sistem menampilkan bukti Riwayat booking	✓	
5	Melihat galeri studio photo	Memilih menu galeri pada sidebar	Sistem menampilkan halaman galeri studio photo	✓	
6	Melihat profil studio photo	Memilih menu profile studio pada sidebar	Sistem menampilkan halaman tentang studio photo	✓	

Padang, Januari 2023

Penguji



(Fiqri Fajor Nugraha)

PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN TRANSAKSI BERBASIS WEB PADA MUTIARA STUDIO PHOTO PADANG

by Muhammad Al Hamid

Submission date: 01-Feb-2023 06:13PM (UTC+0800)

Submission ID: 2003987476

File name: Laporan_TA_Hamid_-_1_Feb_2023_OKE_2023.pdf (15.12M)

Word count: 15572

Character count: 104370

PEMBANGUNAN SISTEM INFORMASI PENGELOLAAN TRANSAKSI BERBASIS WEB PADA MUTIARA STUDIO PHOTO PADANG

ORIGINALITY REPORT

9%

SIMILARITY INDEX

9%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

scholar.unand.ac.id

Internet Source

6%

2

www.scribd.com

Internet Source

3%

Exclude quotes On

Exclude bibliography On

Exclude matches < 3%