BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

Perancangan ini disusun sebagai respons strategis terhadap permasalahan operasional yang telah diidentifikasi pada analisis sistem berjalan di Ade Decoration, di mana ketergantungan pada proses pencatatan manual telah menyebabkan inefisiensi operasional, risiko redundansi dan kehilangan data pesanan, serta kurangnya transparansi progres acara bagi klien. Berlandaskan pada metodologi pengembangan perangkat lunak yang diadopsi dalam penelitian ini, bab ini mencakup fase-fase kritis mulai dari analisis kebutuhan (software requirement analysis), desain sistem (design), penulisan kode program (code generation), hingga pengujian (testing) dan dukungan implementasi (support).

4.1 Analisa Kebutuhan Software

Analisis kebutuhan perangkat lunak dilakukan berdasarkan hasil observasi dan wawancara dengan pemilik, admin, marketing, dan klien Ade Decoration. Tahap ini bertujuan untuk mengidentifikasi kebutuhan sistem yang akan dikembangkan agar sesuai dengan proses bisnis yang sedang berjalan. Hasil analisis digunakan sebagai dasar dalam perancangan Sistem Informasi Manajemen Transaksi, terutama untuk mengatasi kendala dalam pengelolaan pesanan, pemantauan pembayaran, serta penyimpanan data transaksi yang masih dilakukan secara terpisah.

4.1.1 Kebutuhan Fungsional

Sistem informasi manajemen transaksi pada Ade Decoration adalah sistem berbasis *web* di mana setiap aktor memiliki hak akses yang berbeda sesuai dengan

perannya untuk menjaga keamanan dan keutuhan data. Berikut adalah spesifikasi kebutuhan fungsional yang diusulkan:

1. **Kebutuhan Fungsional Administrator (Admin)**

Admin berperan sebagai pengendali operasional harian dan manajemen pesanan dengan rincian fungsional sebagai berikut:

a. Melakukan autentikasi masuk dan keluar sistem (Login dan Logout).
b. Melakukan monitoring operasional harian melalui halaman *dashboard* statistik.
c. Mengelola transaksi pemesanan klien, termasuk melakukan negosiasi, perhitungan ulang biaya (*recalculation*), dan finalisasi pesanan.
d. Melakukan verifikasi bukti pembayaran klien dan menerbitkan *invoice* resmi.
e. Mencetak laporan operasional rutin untuk diserahkan kepada pihak manajemen.

2. **Kebutuhan Fungsional Klien (Client)**

Klien adalah pengguna layanan (calon pengantin atau penyelenggara acara) yang melakukan pemesanan dan pemantauan progres acara, dengan rincian fungsional sebagai berikut:

a. Melakukan registrasi akun baru dan autentikasi (Login dan Logout).
b. Melakukan eksplorasi layanan publik dengan melihat profil perusahaan, paket pernikahan, dan galeri portofolio.
c. Melakukan pemesanan paket layanan (*checkout*), termasuk mengisi formulir detail acara seperti tanggal, lokasi, dan catatan.
d. Melakukan manajemen pembayaran dengan mengunggah bukti transfer atau membayar melalui tautan pembayaran otomatis.
e. Memantau status pesanan terkini dan mengunduh dokumen kontrak serta *invoice*.

3. Kebutuhan Fungsional Pemilik (Owner)

Pemilik bertindak sebagai pengambil keputusan strategis (*decision maker*) yang melakukan pengawasan operasional dan konfigurasi tingkat tinggi, dengan rincian fungsional sebagai berikut:

a. Melakukan autentikasi masuk dan keluar sistem (Login dan Logout).
b. monitoring eksekutif melalui *dashboard* khusus yang menampilkan total pendapatan (*revenue*), total klien, dan rasio pesanan masuk.
c. Melakukan audit rekam data penjualan dan arus keuangan (*cash flow*) perusahaan.

4.1.2 Kebutuhan Non Fungsional

Selain memenuhi kebutuhan fungsional, sistem juga harus memenuhi kebutuhan non fungsional agar dapat digunakan dengan baik selama proses operasional. Kebutuhan non fungsional pada sistem yang diusulkan meliputi aspek performa, keamanan, dan ketersediaan sistem.

1. Kebutuhan Performa (Performance)

Sistem diharapkan mampu memberikan waktu respons yang baik ketika menerima permintaan dari pengguna. Seluruh proses, seperti login, pemesanan, pembayaran, penerbitan invoice, dan pembuatan laporan harus dapat diproses dengan stabil tanpa mengalami kegagalan. Untuk mengetahui kemampuan sistem dalam menangani sejumlah permintaan secara bersamaan, dilakukan pengujian performa menggunakan Apache JMeter.

2. Kebutuhan Keamanan (Security)

Sistem harus mampu menjaga keamanan data pengguna dan data transaksi dari akses yang tidak sah. Setiap pengguna hanya dapat mengakses fitur sesuai dengan hak

akses yang dimiliki. Selain itu, sistem diharapkan memiliki konfigurasi keamanan dasar pada aplikasi web sehingga dapat meminimalkan potensi kerentanan keamanan. Untuk mengetahui tingkat keamanan aplikasi, dilakukan pengujian menggunakan OWASP Zed Attack Proxy (OWASP ZAP).

3. Kebutuhan Ketersediaan (Availability)

Sistem diharapkan dapat diakses oleh pengguna selama layanan aplikasi berjalan sehingga proses operasional dapat dilakukan tanpa kendala. Seluruh fitur utama, seperti pengelolaan paket layanan, pemesanan, pembayaran, penerbitan invoice, dan laporan transaksi harus dapat digunakan sesuai dengan kebutuhan pengguna. Untuk memastikan sistem dapat digunakan sebagaimana mestinya, dilakukan User Acceptance Test (UAT) yang melibatkan pengguna sesuai dengan perannya.

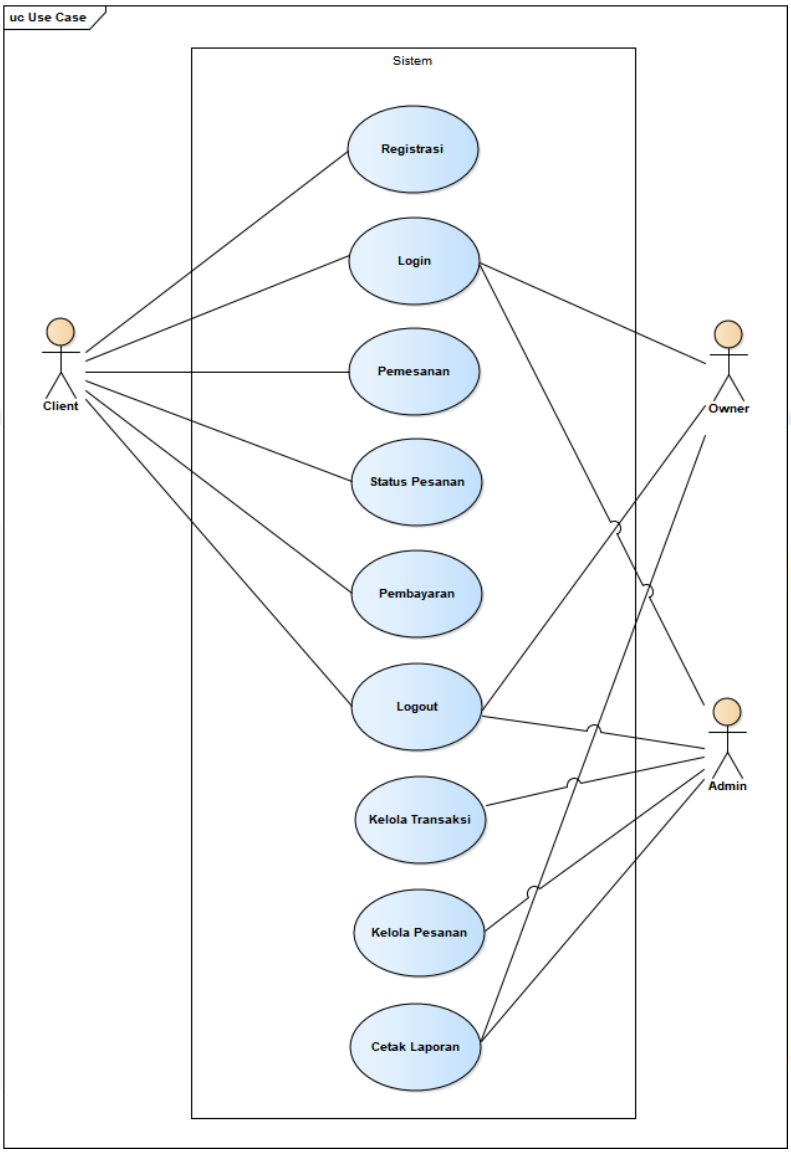
4.2 Desain

Tahap desain menerjemahkan kebutuhan sistem ke dalam representasi teknis menggunakan pemodelan UML (*Unified Modeling Language*) dan perancangan basis data.

4.2.1 Desain Pemodelan Sistem

A. Pemodelan *Use Case* Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan bagaimana aktor berinteraksi dengannya. Diagram di bawah ini menggunakan notasi standar UML untuk menunjukkan batasan sistem *(system boundary)* dan hubungan antara aktor dengan fungsionalitas utama sistem.



Sumber: Hasil Penelitian, 2026

Gambar IV. 1 Use Case Diagram

Tabel VI. 1 Deskripsi User Case Registrasi

Use Case Name	Registrasi
Use Case Description	Klien membuat akun baru agar dapat masuk dan menggunakan layanan pada sistem.
Actors	Client
Pre-Condition	Klien belum memiliki akun dan berada di halaman registrasi.
Post-Condition	Data klien tersimpan ke dalam *database*, dan klien siap untuk melakukan proses *login*.

Fault Condition	Format email tidak valid, kata sandi kurang dari ketentuan batas minimal, atau email sudah terdaftar di sistem.
Main Scenarios	1. Klien membuka halaman registrasi. 2. Klien mengisi *form* data diri (nama, email, *password*, dll). 3. Klien menekan tombol daftar. 4. Sistem memvalidasi data dan menyimpannya ke *database*.
Extension	Jika data tidak valid atau email sudah digunakan, sistem menahan proses simpan dan menampilkan pesan peringatan di bawah *field* yang bersangkutan.

Sumber: Hasil Penelitian, 2026

Tabel VI. 2 Deskripsi User Case Login

Use Case Name	Login
Use Case Description	Autentikasi pengguna untuk masuk ke dalam sistem sesuai dengan hak akses (*role*) masing-masing.
Actors	Client, Admin, Marketing, Owner
Pre-Condition	Pengguna sudah memiliki akun yang terdaftar dan aktif.
Post-Condition	Pengguna berhasil masuk dan sistem menampilkan halaman *dashboard* sesuai dengan *role* pengguna.
Fault Condition	Kombinasi email dan *password* salah, atau akun sedang dinonaktifkan.
Main Scenarios	1. Pengguna membuka halaman *login*. 2. Pengguna memasukkan email dan *password*. 3. Pengguna menekan tombol masuk. 4. Sistem memvalidasi kredensial dan mengarahkan ke halaman *dashboard*.
Extensions	Jika kredensial salah, sistem mengembalikan pengguna ke halaman *login* dengan memunculkan pesan peringatan "Email atau kata sandi salah".

Sumber: Hasil Penelitian, 2026

Tabel VI. 3 Deskripsi User Case Pemesanan

Use Case Name	Pemesanan
Use Case Description	Klien memilih paket *wedding/decoration* dan mengisi detail rencana acara.
Actors	Client
Pre-Condition	Klien sudah *login* dan berada di halaman detail paket layanan.
Post-Condition	Data pesanan baru berhasil dibuat dengan status awal (*pending*) dan menunggu konfirmasi pengelola.
Fault Condition	*Field mandatory* (seperti tanggal acara atau lokasi) kosong, atau tanggal yang dipilih bentrok dengan jadwal yang sudah penuh.
Main Scenarios	1. Klien memilih paket yang diinginkan. 2. Klien mengisi *form* detail acara (tanggal, lokasi, tambahan catatan). 3. Klien menekan tombol pesan / *checkout*. 4. Sistem menghitung total biaya sementara dan menyimpan data pesanan.
Extensions	Jika sistem mendeteksi jadwal yang dipilih sudah tidak tersedia, sistem menampilkan notifikasi ketersediaan jadwal.

Sumber: Hasil Penelitian, 2026

Tabel VI. 4 Deskripsi User Case Status Pesanan

Use Case Name	Status Pesanan
Use Case Description	Klien memantau pergerakan progres dari pesanannya (*pending*, disetujui, proses pengerjaan, atau selesai).
Actors	Client
Pre-Condition	Klien sudah *login* dan memiliki setidaknya satu riwayat pesanan.
Post-Condition	Klien mendapatkan informasi detail mengenai status terkini, *timeline*, dan tugas-tugas terkait acara miliknya.
Fault Condition	Gangguan koneksi saat memuat data, atau ID pesanan tidak ditemukan.
Main Scenarios	1. Klien membuka menu status order.

	2. Sistem menampilkan daftar pesanan beserta indikator statusnya. 3. Klien mengklik salah satu pesanan untuk melihat detail *timeline* dan *rundown* acara.
Extensions	Jika belum ada pesanan, sistem menampilkan antarmuka *empty state* yang mengarahkan klien untuk melihat paket layanan.

Sumber: Hasil Penelitian, 2026

Tabel VI. 5 Deskripsi User Case Pembayaran

Use Case Name	Pembayaran
Use Case Description	Proses Klien melakukan konfirmasi pembayaran, baik melalui unggah bukti transfer maupun tautan pembayaran otomatis.
Actors	Client
Pre-Condition	Klien memiliki pesanan dengan tagihan aktif yang belum dilunasi.
Post-Condition	Bukti pembayaran berhasil diserahkan ke sistem, dan status pesanan berubah menjadi menunggu verifikasi.
Fault Condition	*File* bukti transfer rusak/tidak terbaca, nominal *input* manual tidak sesuai tagihan.
Main Scenarios	1. Klien membuka detail tagihan/pesanan. 2. Klien melihat instruksi dan nominal pembayaran. 3. Klien mengunggah *file* bukti transfer dan memasukkan data rekening pengirim. 4. Sistem menyimpan *file* dan memperbarui status tagihan.
Extensions	Jika format *file* bukti bayar tidak sesuai (misal: PDF/JPG/PNG), sistem menolak unggahan dan meminta *file* yang valid.

Sumber: Hasil Penelitian, 2026

Tabel VI. 6 Deskripsi User Case Logout

Use Case Name	Logout
Use Case Description	Mengakhiri sesi pengguna dengan aman dari sistem.
Actors	Client, Admin, Marketing, Owner

Pre-Condition	Pengguna sedang dalam kondisi *login* ke dalam sistem.
Post-Condition	Sesi akses dicabut (*session destroyed*) dan pengguna dikembalikan ke halaman *landing page* atau *login*.
Fault Condition	Sesi *error* karena *timeout* sebelum tombol *logout* ditekan.
Main Scenarios	1. Pengguna mengklik *icon* profil atau menu pengaturan. 2. Pengguna memilih opsi *Logout* / Keluar. 3. Sistem menghapus sesi autentikasi dan memindahkan pengguna ke halaman awal.
Extensions	-

Sumber: Hasil Penelitian, 2026

Tabel VI. 7 Deskripsi User Case Kelola Transaksi

Use Case Name	Kelola Transaksi
Use Case Description	Admin memverifikasi pembayaran dari klien, menerbitkan *invoice*, dan melakukan finalisasi biaya.
Actors	Admin
Pre-Condition	Admin sudah *login* dan terdapat transaksi masuk yang perlu diproses.
Post-Condition	Tagihan diperbarui, pembayaran diverifikasi, dan status pesanan klien berubah menjadi aktif/dikonfirmasi.
Fault Condition	Bukti pembayaran klien manipulatif atau perhitungan *generate invoice* mengalami *error logic*.
Main Scenarios	1. Admin membuka menu transaksi/pembayaran. 2. Admin meninjau bukti transfer yang diunggah klien. 3. Admin mengubah status tagihan menjadi Lunas/Valid. 4. Sistem men-terbitkan bukti *invoice* resmi dan mengirimkan notifikasi kepada klien.
Extensions	Jika bukti pembayaran tidak valid, Admin dapat menolak transaksi dan sistem akan memberi opsi kepada Admin untuk mengirimkan catatan perbaikan ke Klien.

Sumber: Hasil Penelitian, 2026

Tabel VI. 8 Deskripsi User Case Kelola Pesanan

Use Case Name	Kelola Pesanan
Use Case Description	Admin memproses pesanan masuk, menyesuaikan jadwal (*rundown*)dan mengonfirmasi pesanan ke klien.
Actors	Admin
Pre-Condition	Admin sudah *login* dan terdapat pesanan baru dari klien.
Post-Condition	Kesepakatan acara (tanggal, tugas, rincian teknis) ditetapkan dan pesanan siap dieksekusi.
Fault Condition	Kesalahan sinkronisasi tanggal acara.
Main Scenarios	1. Admin membuka daftar pesanan klien yang masuk. 2. Admin meninjau detail acara dan melakukan negosiasi jika diperlukan. 3. Admin menyetujui pesanan.
Extensions	Jika terdapat revisi atau tambahan layanan *add-ons* (misal: tambah porsi katering), Admin melakukan *recalculation* harga sebelum mengonfirmasi pesanan.

Sumber: Hasil Penelitian, 2026

Tabel VI. 9 Deskripsi User Case Cetak Laporan

Use Case Name	Cetak Laporan
Use Case Description	Mengekspor data operasional, penjualan, dan keuangan sistem menjadi format dokumen laporan.
Actors	Admin, Owner
Pre-Condition	Aktor sudah *login* dan memiliki akses ke menu analitik/laporan.
Post-Condition	Dokumen laporan (berupa *file* Excel atau PDF) berhasil diunduh ke perangkat pengguna.
Fault Condition	Tidak ada data transaksi yang ditemukan pada rentang tanggal filter yang dipilih.
Main Scenarios	1. Aktor masuk ke menu pelaporan. 2. Aktor memilih jenis laporan dan menentukan filter rentang bulan/tanggal.

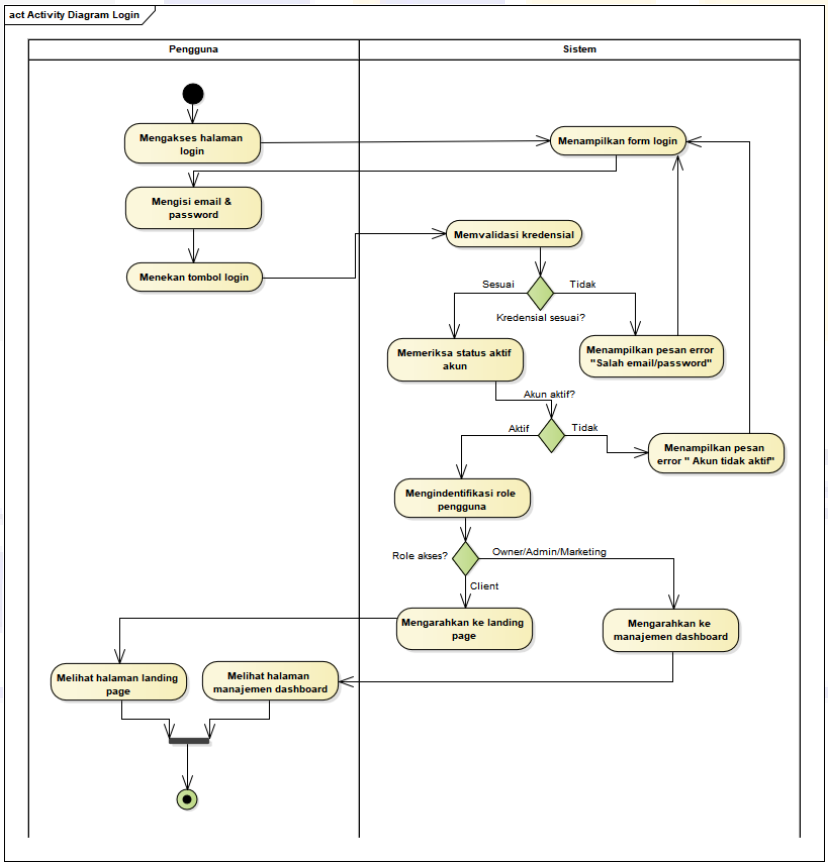
	3. Aktor menekan tombol *Export* atau Cetak. 4. Sistem mengompilasi data *database* dan mengunduh *file* dokumen.
Extensions	Jika *database* kosong pada tanggal yang di-filter, sistem menampilkan peringatan "Data pada periode tersebut tidak tersedia".

Sumber: Hasil Penelitian, 2026

B. Pemodelan Activity Diagram

1. Aktivitas Diagram Login Pengguna

Aktivitas ini mengilustrasikan alur otentikasi sistem yang diikat dengan validasi ganda, yaitu pencocokan kredensial dan pengecekan status aktif akun pengguna.



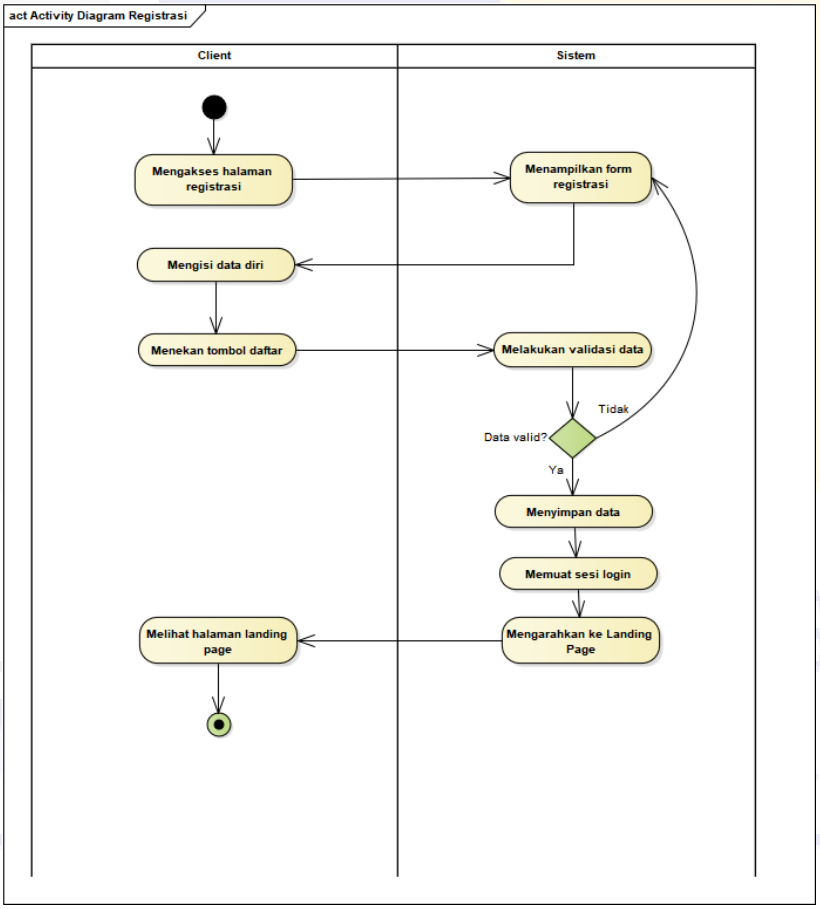
Sumber: Hasil Penelitian, 2026

Gambar IV. 2 Activity Diagram Login

Setelah otentikasi dinyatakan valid, sistem secara otonom melakukan routing berdasarkan hak akses (role), di mana Klien akan diarahkan ke landing page sedangkan staf operasional langsung masuk ke dashboard manajemen.

2. Aktivitas Diagram Registrasi Client

Activity diagram ini memetakan alur pendaftaran akun klien baru, di mana sistem bertugas memvalidasi kelengkapan *input* sebelum menyimpannya secara permanen ke *database*.



Sumber: Hasil Penelitian, 2026

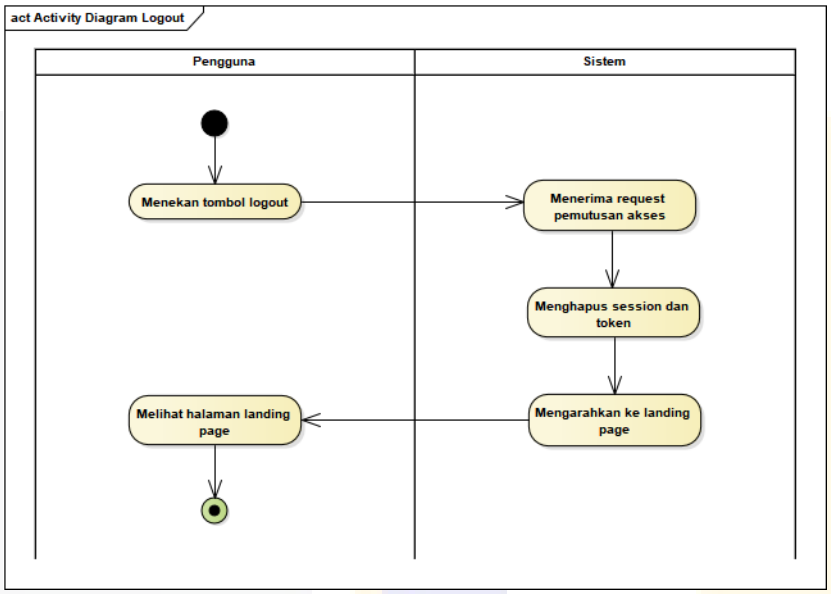
Gambar IV. 3 Activity Diagram Registrasi

Poin teknis utamanya ada pada efisiensi *User Experience* (UX), karena setelah data dinyatakan valid, sistem dirancang untuk langsung mengeksekusi sesi otentikasi

(*auto-login*) dan melempar klien ke *landing page* tanpa menuntut mereka melakukan *login* ulang.

3. Aktivitas Diagram Logout

Activity diagram ini mendeskripsikan mekanisme terminasi akses (*logout*) yang secara ketat memberlakukan pembersihan data autentikasi.



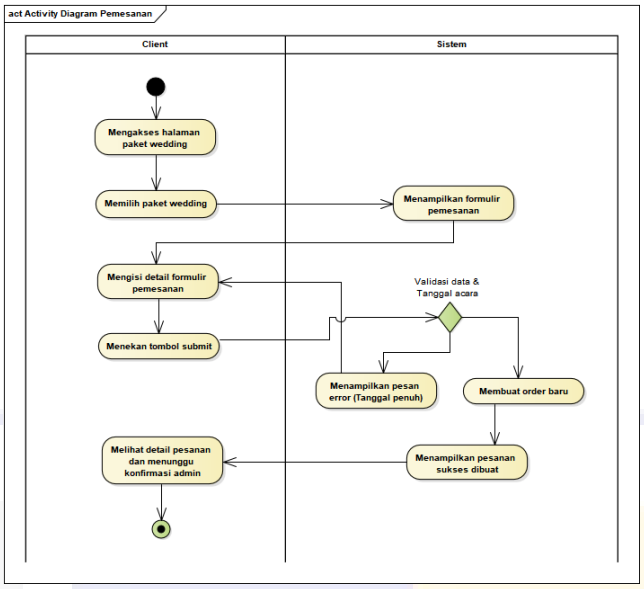
Sumber: Hasil Penelitian, 2026

Gambar IV. 4 Activity Diagram Logout

Sistem diwajibkan untuk menghancurkan *session* dan *token* aktif di *back-end* sebelum melempar pengguna kembali ke *landing page*.

4. Aktivitas Diagram Pemesanan

Activity diagram ini menguraikan alur reservasi paket pernikahan, di mana fokus utamanya ditekankan pada mekanisme sistem dalam mencegah terjadinya *double-booking*.



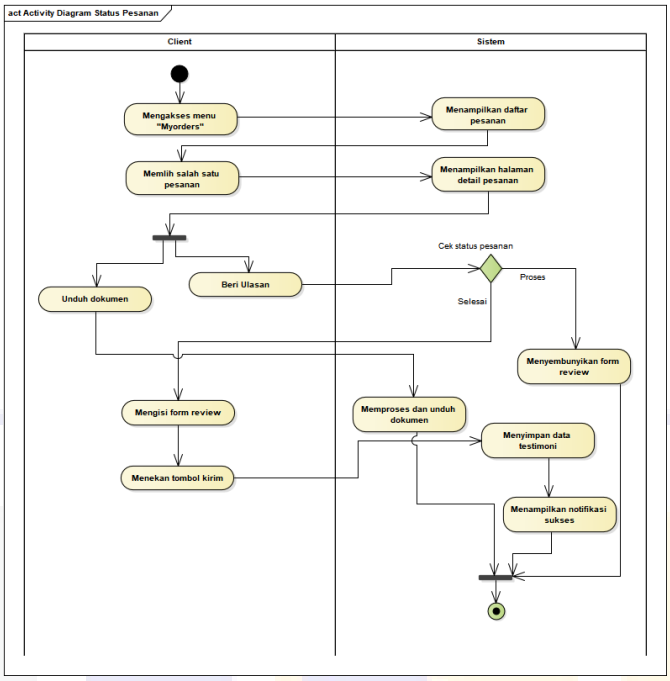
Sumber: Hasil Penelitian, 2026

Gambar IV. 5 Activity Diagram Pemesanan

Begitu klien mengirimkan formulir pemesanan, sistem akan langsung memvalidasi ketersediaan jadwal operasional pada tanggal acara yang dipilih. Jika slot di hari tersebut masih kosong, pesanan akan direkam dan klien diarahkan untuk menunggu konfirmasi admin, namun jika sudah penuh, sistem secara otomatis menolak dan menahan klien untuk menyesuaikan kembali tanggalnya.

5. Aktivitas Diagram Status Pesanan

Activity diagram ini memvisualisasikan alur interaksi klien di halaman riwayat pesanan saat mereka ingin mengunduh dokumen transaksi atau memberikan testimoni.



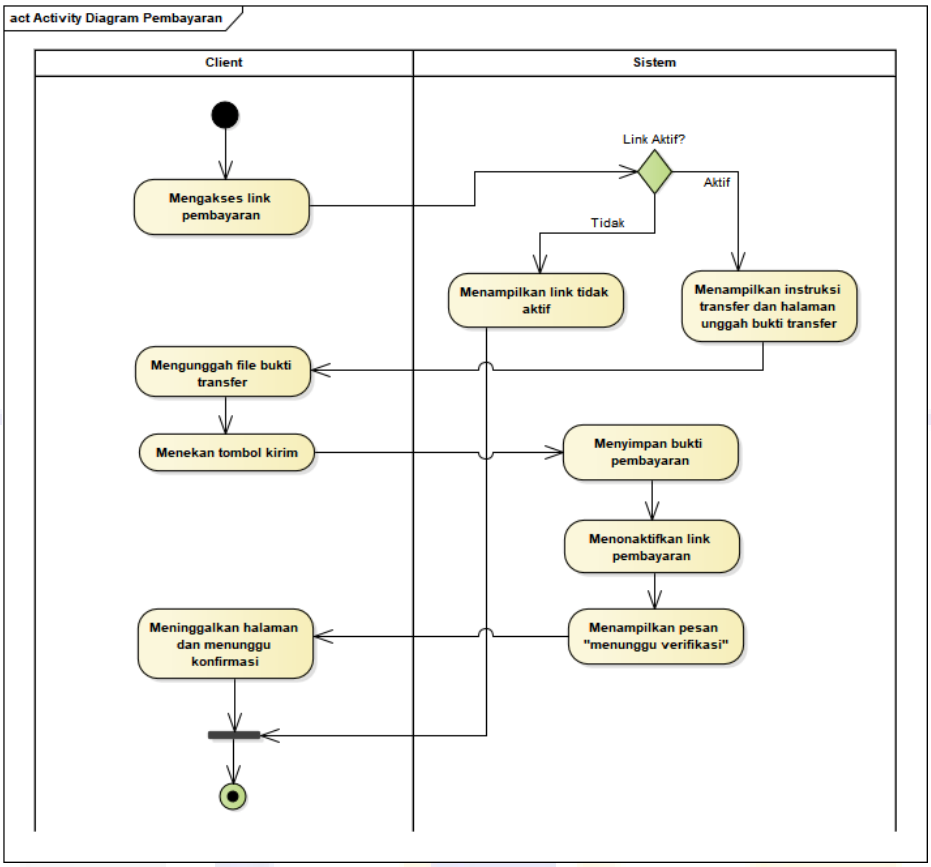
Sumber: Hasil Penelitian, 2026

Gambar IV. 6 Activity Diagram Status Pesanan

Aturan bisnis yang ditekankan pada alur ini ada di pembatasan fitur ulasan, di mana sistem hanya akan memunculkan *form review* jika status pesanan sudah benar-benar dinyatakan selesai. Logika pembatasan akses ini sengaja dirancang untuk menjaga keakuratan penilaian, sehingga klien tidak bisa asal memberikan *rating* saat proyek dekorasinya sendiri masih berstatus dikerjakan.

6. Aktivitas Diagram Pembayaran

Activity diagram ini memetakan alur konfirmasi pembayaran, di mana sistem akan memvalidasi status *link* terlebih dahulu sebelum memberikan akses kepada klien untuk mengunggah bukti transfer.



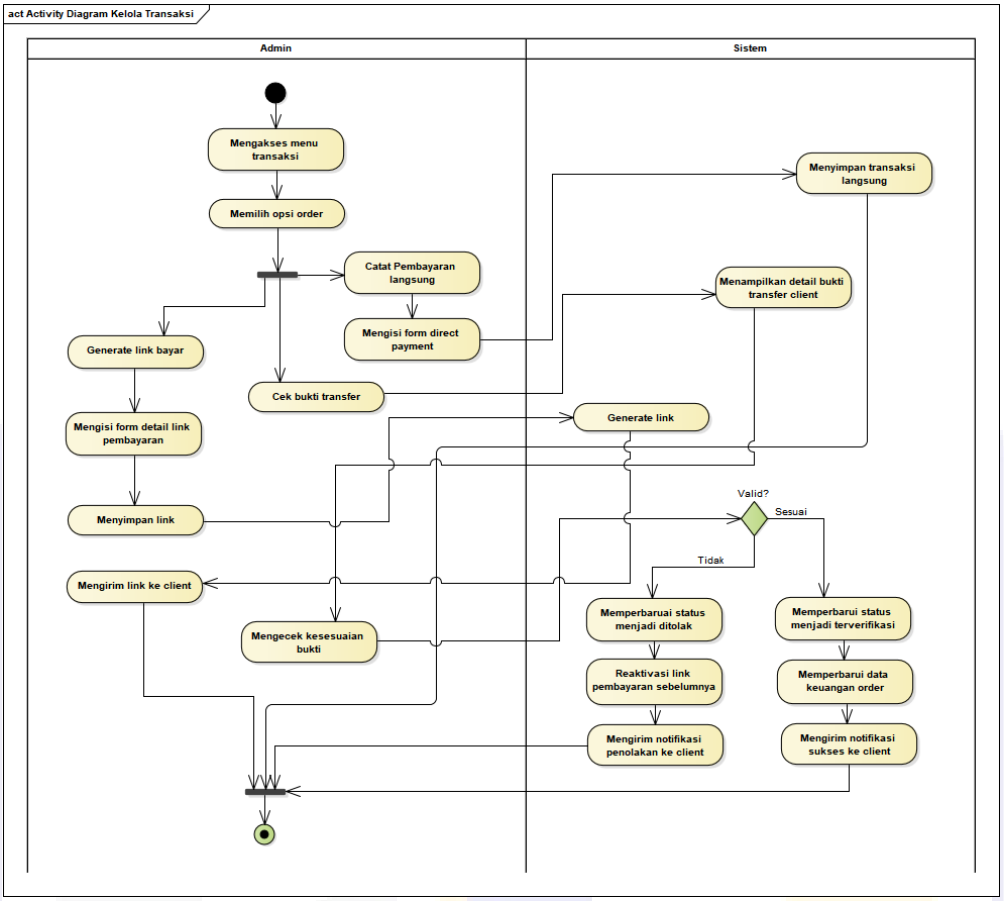
Sumber: Hasil Penelitian, 2026

Gambar IV. 7 Activity Diagram Pembayaran

Poin krusialnya ada pada langkah pengamanan sistem yang langsung menonaktifkan *link* tersebut sesaat setelah *file* berhasil tersimpan, sehingga secara efektif memblokir celah *double submit* atau pengunggahan data ganda dari sisi klien.

7. Aktivitas Diagram Kelola Transaksi

Activity diagram ini memetakan fleksibilitas manajemen transaksi admin yang mampu mengakomodasi tiga skenario operasional sekaligus, yakni pembuatan *link* tagihan, verifikasi unggahan transfer, dan pencatatan pembayaran langsung (*on-site*).



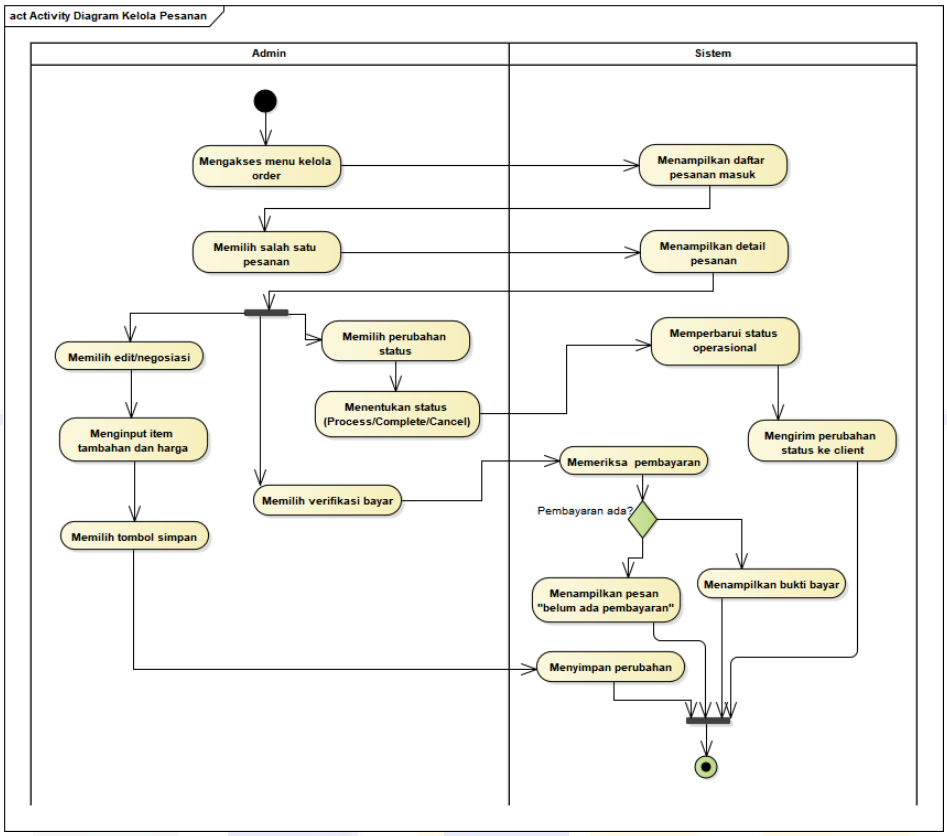
Sumber: Hasil Penelitian, 2026

Gambar IV. 8 Activity Diagram Kelola Transaksi

Nilai kepraktisan sistem ini paling terlihat pada jalur penolakan bukti bayar, di mana aplikasi dirancang untuk otomatis mereaktivasi *link* sebelumnya sehingga klien bisa langsung revisi unggahan tanpa admin perlu repot membuat tautan baru dari awal.

8. Aktivitas Diagram Kelola Pesanan

Activity diagram ini memetakan fleksibilitas admin dalam mengontrol siklus hidup pesanan melalui tiga cabang aksi sekaligus, yakni melakukan negosiasi harga, mengecek riwayat pembayaran, atau mengeksekusi pembaruan status operasional.



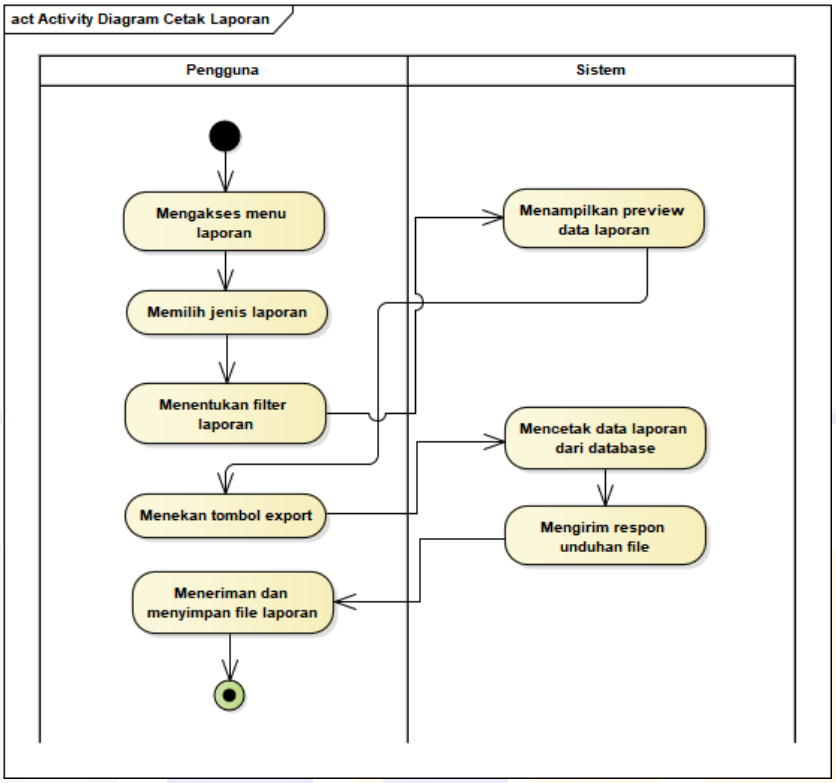
Sumber: Hasil Penelitian, 2026

Gambar IV. 9 Activity Diagram Kelola Pesanan

Nilai praktis dari rancangan ini sangat terlihat pada alur perubahan status, di mana sistem ditugaskan untuk otomatis menembakkan notifikasi progres proyek ke sisi klien agar transparansi tetap terjaga tanpa mengharuskan admin melakukan *follow-up* manual.

9. Aktivitas Diagram Cetak Laporan

Activity diagram ini memetakan alur generasi laporan berkala yang diawali dengan penyajian pratinjau data secara dinamis sesuai dengan kriteria filter yang ditentukan pengguna



Sumber: Hasil Penelitian, 2026

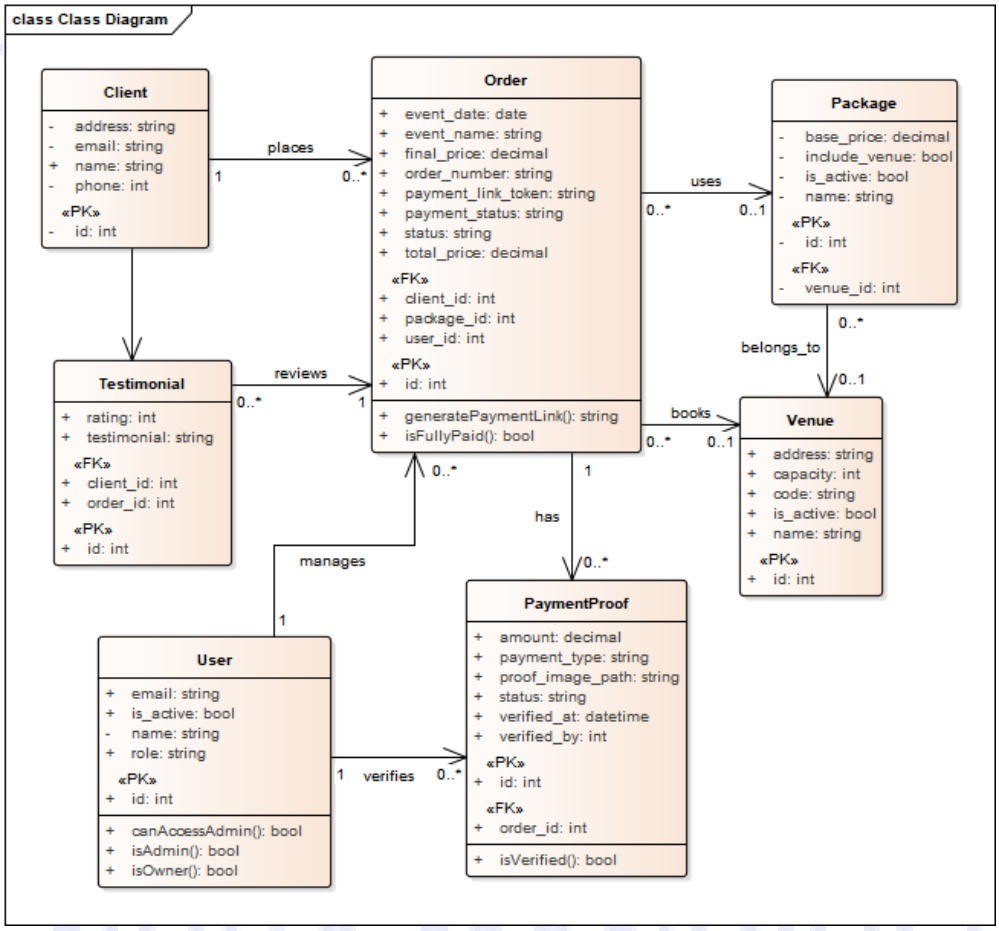
Gambar IV. 10 Activity Diagram Cetak Laporan

Ketika proses ekspor dieksekusi, sistem secara otonom menarik dataset dari basis data untuk langsung dikonversi menjadi berkas unduhan yang siap disimpan pada perangkat lokal.

C. Pemodelan Class Diagram

Class Diagram (Diagram Kelas) merupakan salah satu diagram struktural dalam UML yang menggambarkan struktur statis dari sistem yang akan dibangun. Diagram ini mendefinisikan himpunan kelas-kelas, atribut (data), metode (operasi), serta hubungan atau relasi antar objek di dalamnya. Dalam perancangan Sistem Informasi Wedding Organizer , *Class Diagram* memegang peranan krusial sebagai landasan pembentukan skema basis data (*database schema*).

Diagram ini dirancang untuk memetakan bagaimana entitas-entitas utama seperti Client, Admin, dan Owner saling berinteraksi dalam proses bisnis. Pemodelan ini memastikan bahwa setiap data yang disimpan memiliki struktur yang jelas, terorganisir, dan memiliki integritas referensial yang kuat guna mendukung fitur-fitur utama. Berikut adalah rancangan *Class Diagram* untuk sistem usulan:



Sumber: Hasil Penelitian, 2026

Gambar IV. 11 Class Diagram

Class Diagram pada Sistem Informasi Manajemen dan Pemesanan *Wedding Organizer* Berbasis Web ini menggambarkan struktur statis sistem yang terdiri dari sejumlah kelas utama beserta atribut, operasi, dan hubungan antar entitas. Diagram ini dirancang untuk merepresentasikan bagaimana aliran data dikelola secara terintegrasi

guna mendukung proses bisnis perusahaan, mulai dari pemesanan paket oleh klien, manajemen operasional lapangan, hingga pelaporan finansial dan penggajian pegawai.

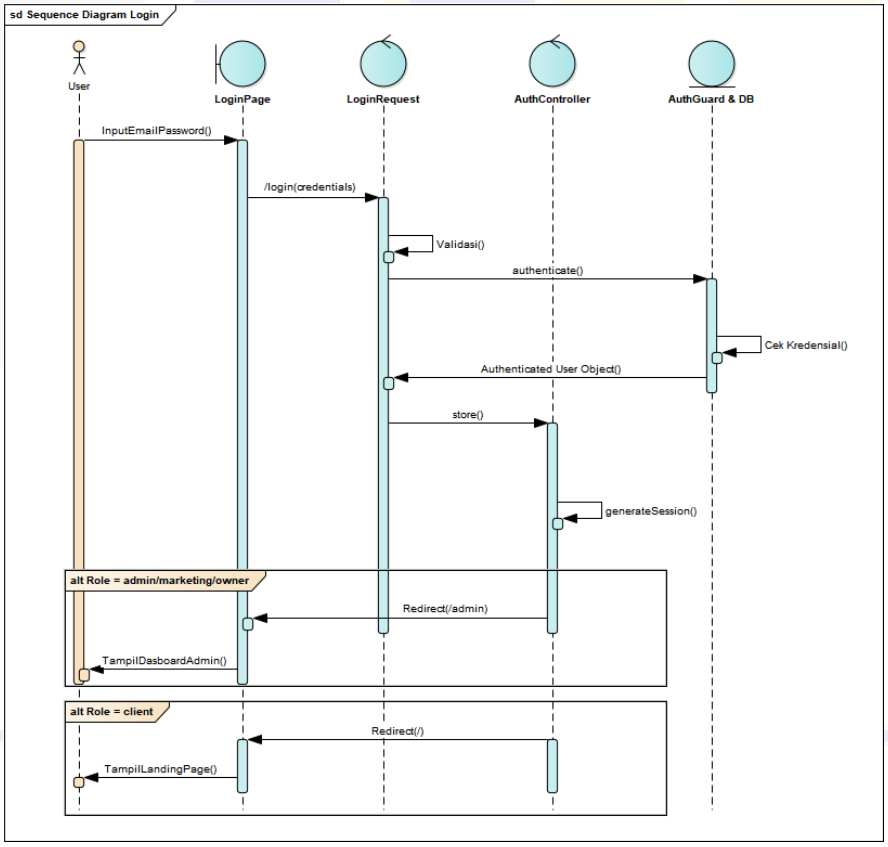
1. Kelas User berperan sebagai kelas inti (*core class*) yang merepresentasikan staf pengelola sistem, seperti Admin, Marketing, dan Owner. Kelas ini memiliki atribut otorisasi seperti email, role, dan is_active. Selain itu, kelas User dilengkapi dengan metode fungsional seperti isAdmin() dan canAccessAdmin() untuk memvalidasi hak akses. Kelas ini memiliki kendali manajerial yang luas, dibuktikan dengan relasinya terhadap kelas *Order* dan *PaymentProof*.
2. Kelas Client berfungsi untuk menyimpan data profil pelanggan dengan atribut seperti name, email, phone, dan address. Kelas ini merupakan titik awal dari siklus transaksi dan memiliki relasi *one-to-many* dengan kelas *Order*, yang berarti satu klien dapat melakukan banyak pesanan. Selain itu, kelas ini juga terhubung dengan kelas Testimonial, yang memungkinkan klien memberikan *rating* dan ulasan setelah acara selesai berdasarkan pesanan (order_id) mereka.
3. Kelas Order merepresentasikan modul transaksi utama pemesanan layanan *wedding*. Kelas ini menyimpan detail finansial dan administratif melalui atribut seperti event_date, total_price, payment_status, dan payment_link_token. Kelas ini memiliki operasi logika seperti generatePaymentLink() dan isFullyPaid() untuk otomatisasi tagihan. *Order* menjadi pusat relasi yang mengikat kelas *Client*, *Package*, serta berelasi langsung dengan kelas PaymentProof untuk melacak bukti transfer yang diunggah klien guna divalidasi oleh Admin.
4. Kelas Package dan Venue saling berelasi untuk membentuk modul katalog layanan. Kelas *Package* menyimpan harga dasar (base_price) dan rincian paket, yang dapat dikaitkan dengan lokasi acara pada kelas *Venue*.

D. Pemodelan 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. Berikut adalah spesifikasi perancangan *Sequence Diagram* yang dikelompokkan berdasarkan aktor pengguna sistem:

1. Sequence Diagram Login

Sequence Diagram ini mengilustrasikan alur interaksi saat pengguna (seperti Admin, Marketing, Owner, maupun Klien) melakukan proses otentikasi untuk masuk ke dalam sistem.



Sumber: Hasil Penelitian, 2026

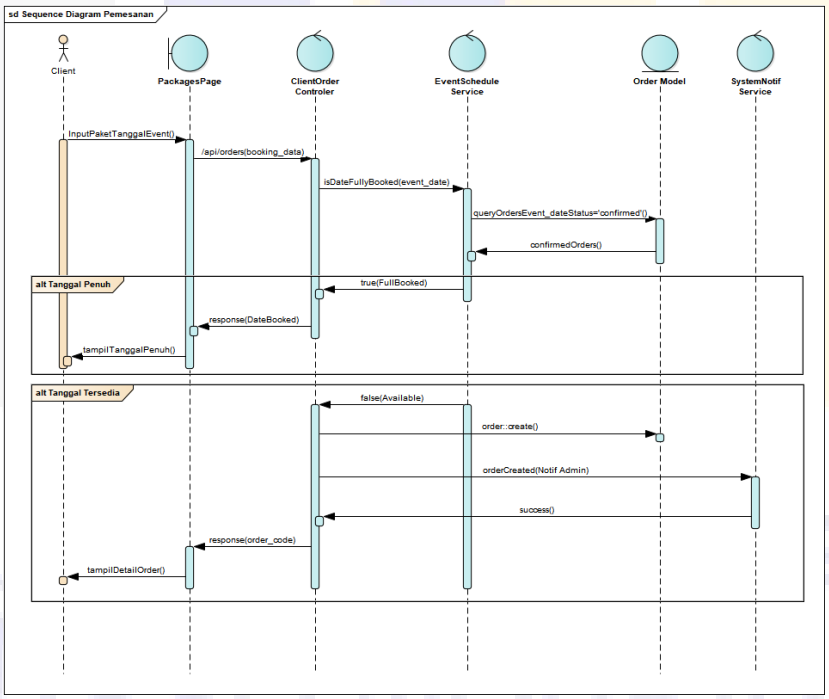
Gambar IV. 12 Sequence Diagram Login

Pada proses ini, pengguna menginputkan email dan kata sandi melalui antarmuka halaman *login*. Sistem kemudian menangkap *request* tersebut untuk divalidasi

formatnya sebelum diteruskan ke *database* guna pengecekan kecocokan kredensial. Apabila otentikasi dinyatakan valid, sistem akan membangun sesi (*generate session*) yang baru. Setelah itu, sistem secara otomatis mengeksekusi percabangan arah halaman (redireksi) berdasarkan hak akses (*role*); pihak internal manajemen akan langsung diarahkan ke *dashboard* admin, sedangkan pelanggan akan dilempar menuju halaman *landing page* utama.

2. Sequence Diagram Pemesanan

Sequence Diagram ini mengilustrasikan alur interaksi sistem saat klien melakukan pemesanan paket pernikahan.



Sumber: Hasil Penelitian, 2026

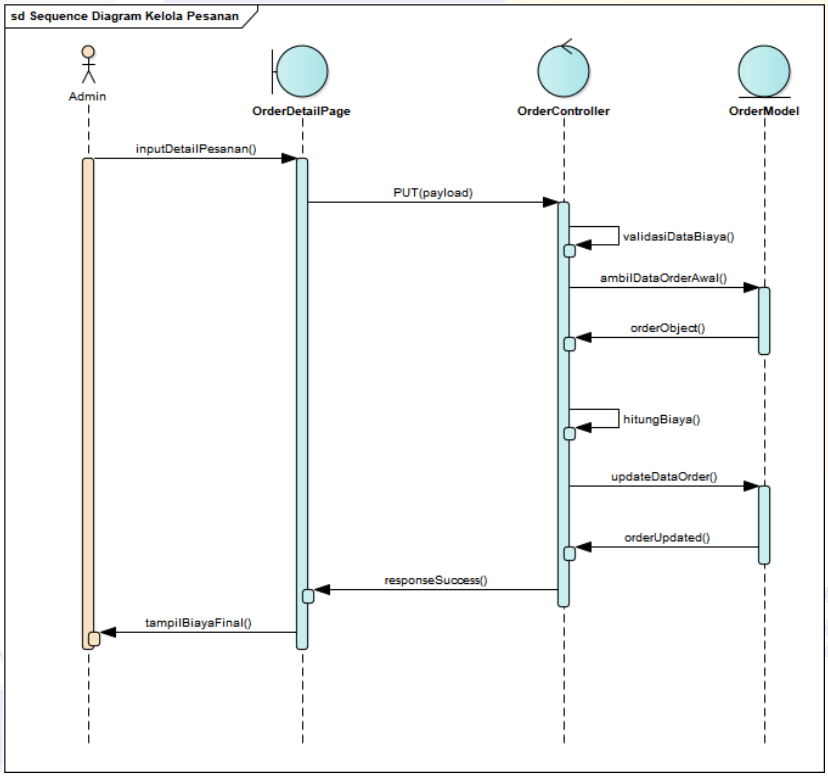
Gambar IV. 13 Sequence Diagram Pemesanan

Proses bermula ketika klien memilih paket dan menentukan tanggal acara melalui halaman antarmuka. Alih-alih langsung menyimpannya, sistem akan memvalidasi ketersediaan tanggal operasional tersebut di *database* terlebih dahulu untuk mencegah terjadinya bentrok jadwal (*double-booking*). Dari pengecekan ini, sistem

mengeksekusi dua skenario percabangan: jika kuota tanggal tersebut sudah penuh oleh pesanan lain, sistem otomatis menolak *request* dan memunculkan pesan peringatan kepada klien. Sebaliknya, jika tanggal masih tersedia, sistem akan merekam data pesanan secara permanen, memicu pengiriman notifikasi ke pihak admin, lalu mengarahkan klien menuju halaman detail transaksinya.

3. Sequence Diagram Kelola Pesanan

Sequence Diagram ini menggambarkan alur interaksi sistem saat admin memperbarui rincian tagihan atau melakukan negosiasi biaya pada suatu pesanan.



Sumber: Hasil Penelitian, 2026

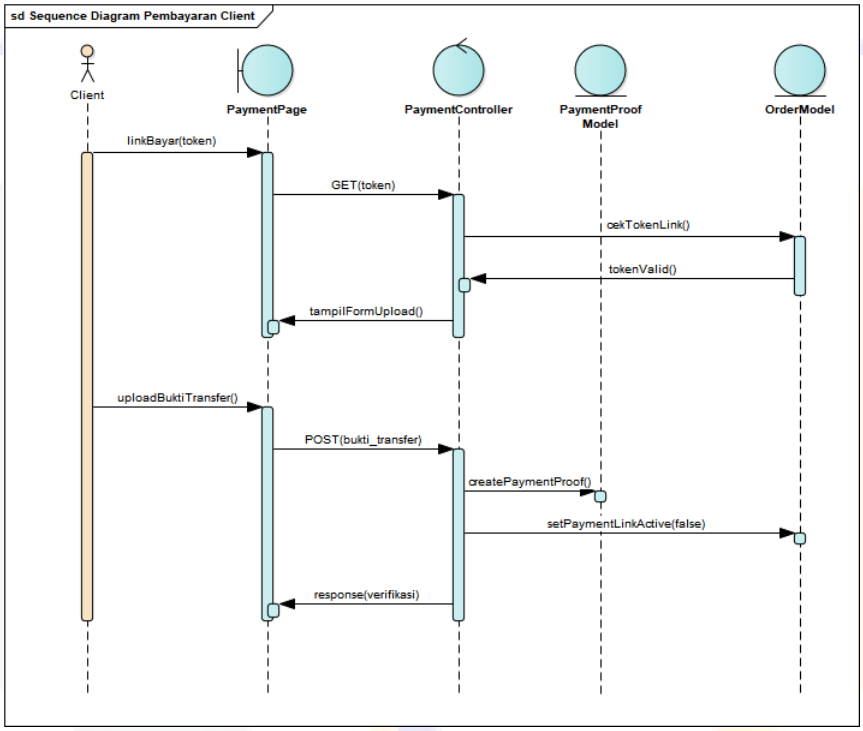
Gambar IV. 14 Sequence Diagram Kelola Pesanan

Untuk menjaga validitas perhitungan, sistem dirancang untuk menarik data pesanan awal dari *database* terlebih dahulu guna dikalkulasi ulang secara otomatis di sisi *server*. Setelah proses pembaruan harga tersebut divalidasi dan berhasil disimpan,

antarmuka aplikasi akan langsung menyajikan rincian biaya final yang paling baru kepada admin.

4. Sequence Diagram Pembayaran Client

Sequence Diagram ini memvisualisasikan alur interaksi klien saat melakukan konfirmasi pembayaran melalui tautan khusus ber-token.



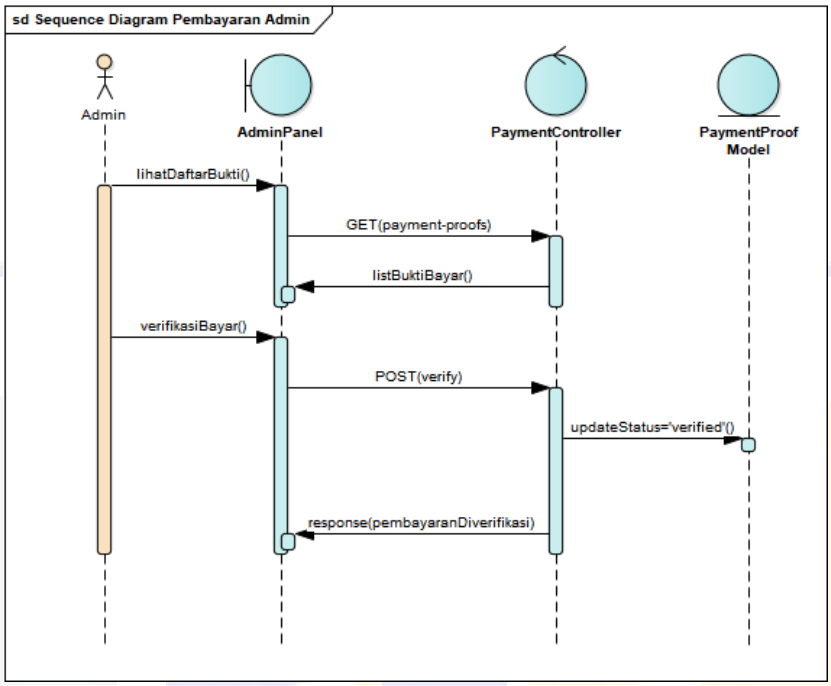
Sumber: Hasil Penelitian, 2026

Gambar IV. 15 Sequence Diagram Pembayaran

Proses diawali dengan validasi keamanan token terlebih dahulu; apabila tautan di *database* dinyatakan valid dan aktif, barulah antarmuka sistem menyajikan formulir unggahan kepada klien. Poin krusial pada alur ini terletak pada langkah mitigasi sistem sesaat setelah klien berhasil mengirimkan bukti transfer, di mana *controller* tidak hanya merekam dokumen tersebut, tetapi juga langsung menonaktifkan status tautan pembayarannya guna mencegah terjadinya celah pengiriman data ganda (*double-submit*).

5. Sequence Diagram Verifikasi Pembayaran

Sequence Diagram ini menguraikan alur kerja admin saat meninjau dan memvalidasi bukti transfer pembayaran yang masuk dari klien.



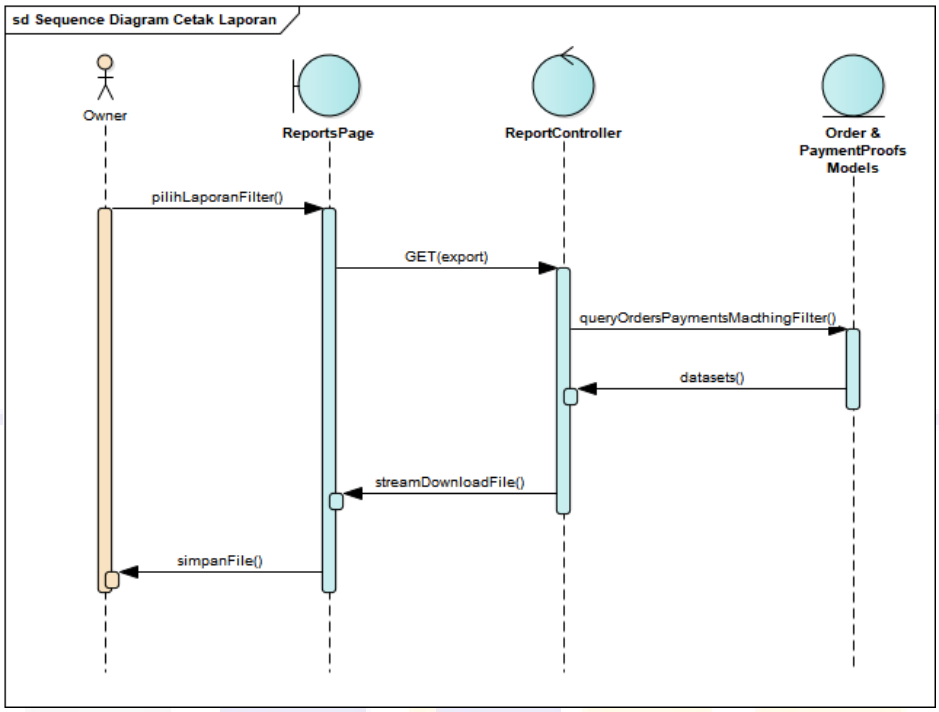
Sumber: Hasil Penelitian, 2026

Gambar IV. 16 Sequence Diagram Verifikasi Pembayaran

Prosesnya bermula ketika admin memuat daftar dokumen pembayaran melalui halaman antarmuka dasbor untuk dilakukan pengecekan silang. Setelah mutasi direkening dipastikan sesuai dan admin mengeksekusi aksi verifikasi, sistem akan meneruskan instruksi tersebut ke *controller* untuk memperbarui status dokumen di basis data menjadi terverifikasi. Alur ini kemudian ditutup dengan kembalian respons dari *server* yang memicu munculnya notifikasi keberhasilan validasi pada layar admin.

6. Sequence Diagram Cetak Laporan

Sequence Diagram ini merepresentasikan alur interaksi sistem saat *Owner* mengekspor data laporan operasional perusahaan.



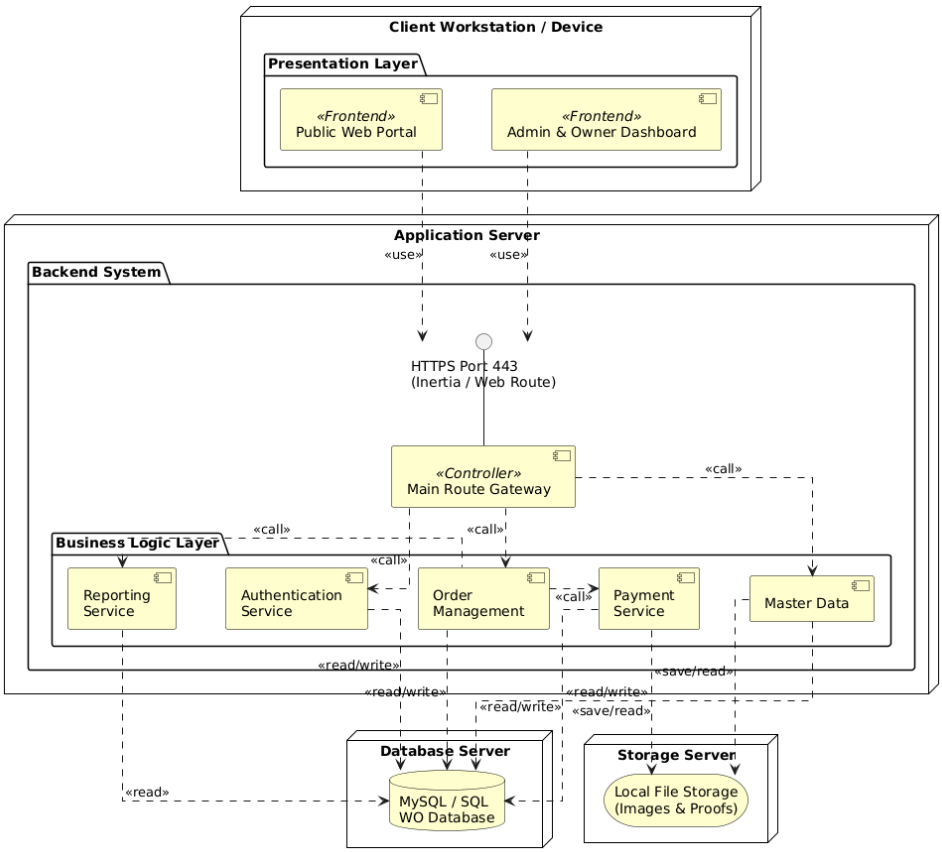
Sumber: Hasil Penelitian, 2026

Gambar IV. 17 Sequence Diagram Cetak Laporan

Proses diinisiasi ketika *Owner* menetapkan parameter filter pada antarmuka dasbor, yang kemudian memicu *controller* untuk menarik dataset spesifik dari basis data terkait riwayat pesanan dan pembayaran yang relevan. Nilai efisiensi teknis pada alur ini ditunjukkan oleh respons *server*; alih-alih merender tampilan halaman baru, *controller* bertugas langsung memproses dan menyalurkan (*stream*) kompilasi data tersebut menjadi format dokumen siap unduh agar bisa langsung disimpan ke penyimpanan lokal perangkat *Owner*.

E. Pemodelan Component Diagram

Component Diagram (Diagram Komponen) menggambarkan struktur fisik dari sistem, memperlihatkan bagaimana kode program dibagi menjadi modul-modul yang terpisah namun saling berinteraksi melalui antarmuka (*interface*) yang terdefinisi. Diagram ini memberikan gambaran tingkat tinggi mengenai arsitektur perangkat lunak yang dibangun untuk Wedding Organizer Ade Decoration.



Sumber: Hasil Penelitian, 2026

Gambar IV. 18 Component Diagram

Component Diagram ini memvisualisasikan arsitektur perangkat lunak pada sistem manajemen transaksi *Wedding Organizer* secara menyeluruh, memetakan bagaimana komponen-komponen fungsional didistribusikan ke dalam lingkungan fisik serta bagaimana mereka saling berinteraksi. Arsitektur sistem ini dibagi menjadi empat lingkungan utama, yaitu *Client Workstation*, *Application Server*, *Database Server*, dan *Storage Server*, yang saling terhubung untuk mendukung operasional bisnis secara terintegrasi.

Pada sisi pengguna atau *Client Workstation*, terdapat lapisan presentasi (*Presentation Layer*) yang terdiri dari dua antarmuka utama: *Public Web Portal* dan *Admin & Owner Dashboard*. *Public Web Portal* berfungsi sebagai etalase interaktif bagi klien untuk mengeksplorasi layanan dan melakukan pemesanan, sedangkan

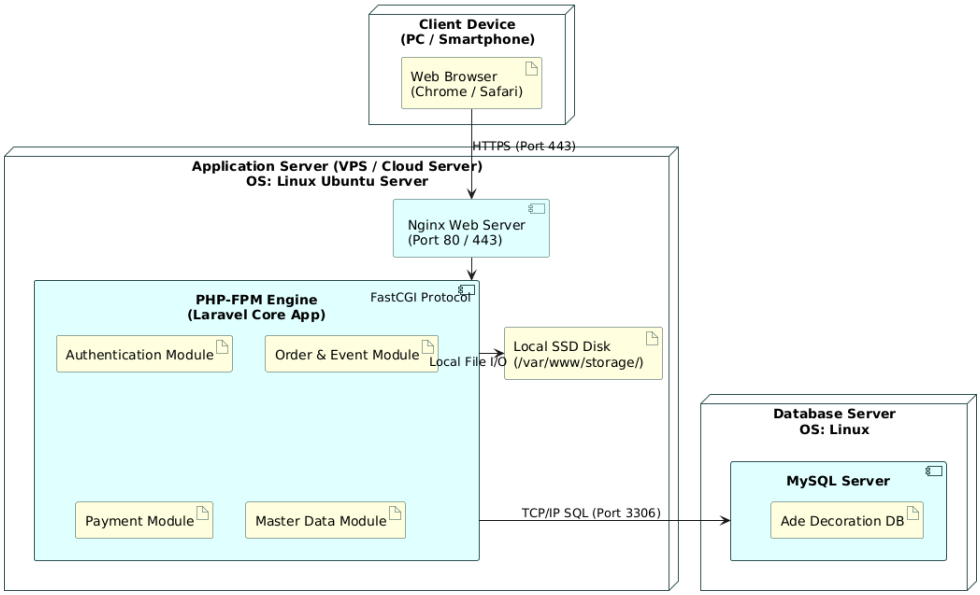
Admin & Owner Dashboard ditujukan secara eksklusif bagi staf internal untuk manajemen operasional harian. Kedua antarmuka ini berkomunikasi dengan sisi *server* melalui rute web berprotokol HTTPS yang aman.

Di jantung sistem terdapat *Application Server* yang menampung seluruh kendali logika bisnis (*back-end*). Pintu masuk utama untuk semua *request* dari sisi klien dikoordinasikan oleh *Main Route Gateway* (berperan sebagai *Controller*) yang mengekspos Port HTTPS 443, memanfaatkan teknologi *routing* modern (seperti Inertia). Di balik *gateway* ini, sistem mendelegasikan tugas ke serangkaian layanan internal (*Business Logic Layer*) yang bekerja secara modular. Layanan ini mencakup *Authentication Service* untuk mengamankan sesi otentikasi pengguna, *Order & Event Management* untuk mengawal siklus pesanan hingga penjadwalan acara, *Payment & Finance Service* untuk manajemen tagihan transaksi, *Master Data* untuk pengelolaan inventaris dekorasi dan lokasi (*venue*), serta *Reporting Service* untuk ekstraksi data laporan.

Untuk urusan persistensi data, sistem menerapkan pemisahan beban (*load separation*) antara data terstruktur dan berkas media. Seluruh layanan *backend* memiliki ketergantungan akses (*read/write*) secara langsung terhadap *Database Server* berbasis MySQL yang menampung seluruh relasi data operasional. Di sisi lain, sistem juga memanfaatkan *Storage Server* independen (*Local File Storage*) yang dikhususkan untuk menyimpan unggahan media statis, seperti gambar bukti transfer klien maupun foto galeri dekorasi. Pemisahan komponen penyimpanan ini memastikan aliran data tetap efisien, responsif, dan menjaga mesin *database* utama dari beban kerja yang berlebihan.

F. Pemodelan Deployment Diagram

Deployment Diagram digunakan untuk menggambarkan konfigurasi fisik sistem, termasuk perangkat keras (*hardware*), perangkat lunak (*software*), serta hubungan komunikasi antar node. Diagram ini menunjukkan bagaimana sistem diimplementasikan pada lingkungan nyata.



Sumber: Hasil Penelitian, 2026

Gambar IV. 19 Deployment Diagram

Deployment Diagram pada sistem manajemen transaksi *Wedding Organizer* ini memvisualisasikan arsitektur topologi jaringan dan konfigurasi fisik infrastruktur *server*. Diagram ini menunjukkan bagaimana komponen-komponen perangkat lunak dideploy dan dieksekusi di dalam *environment* produksi berbasis *client-server*. Secara garis besar, arsitektur fisik sistem ini didistribusikan ke dalam tiga *node* perangkat keras utama, yaitu *Client Device*, *Application Server* berbasis *Virtual Private Server* (VPS), dan *Database Server*.

Pada sisi *front-end*, *node Client Device* merepresentasikan gawai pengguna, seperti PC maupun *smartphone*, yang mengakses antarmuka sistem melalui *web browser*. Permintaan dari klien ini menjangkau *server* secara aman melalui jalur komunikasi berprotokol HTTPS. Rute *traffic* ini langsung diterima oleh *node*

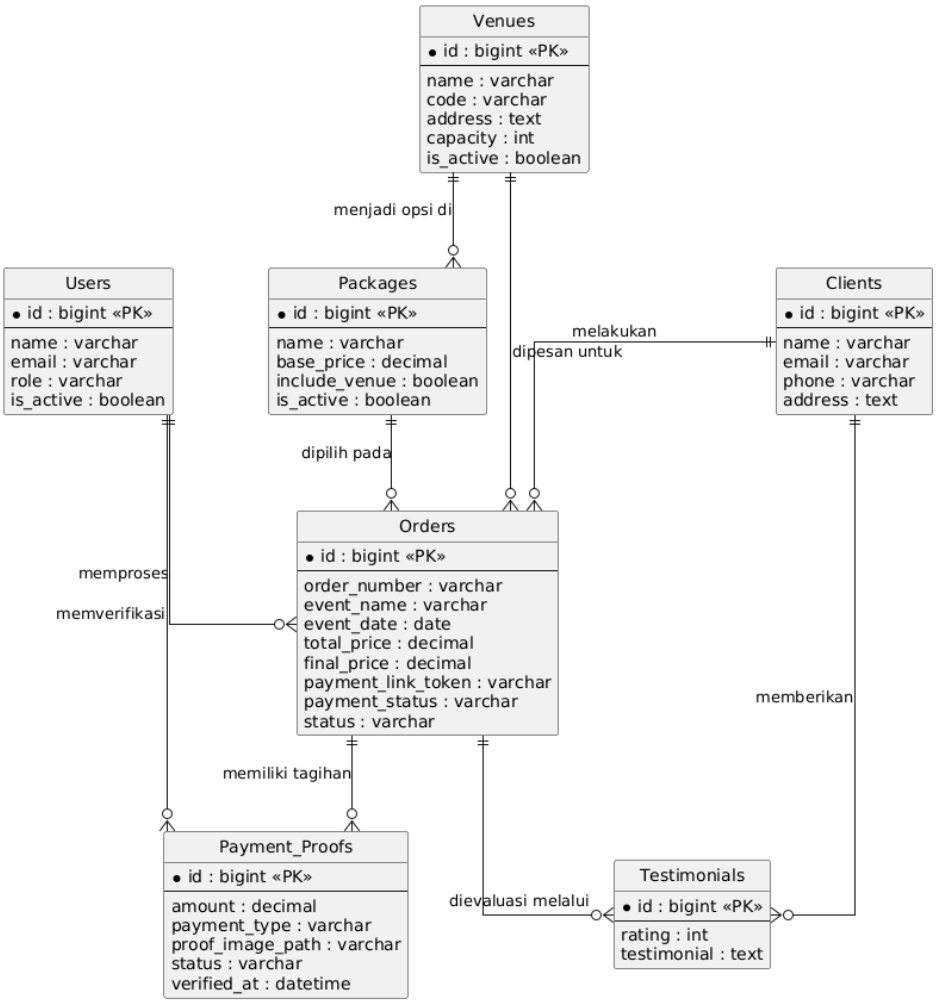
Application Server yang berjalan di atas sistem operasi Linux Ubuntu. Di dalam *server* aplikasi ini, *Nginx Web Server* bertugas sebagai penerima *request* awal yang kemudian mendelegasikan pemrosesan ke *PHP-FPM Engine* melalui protokol FastCGI. *Engine* inilah yang menjadi motor penggerak kerangka kerja Laravel (*Core App*) untuk mengeksekusi berbagai modul logika bisnis utama, mulai dari *Authentication Module*, *Order & Event Module*, *Payment Module*, hingga *Master Data Module*. Bersamaan dengan itu, *server* aplikasi ini juga mengandalkan *Local SSD Disk* untuk menangani proses baca-tulis penyimpanan *file* lokal.

Untuk manajemen data yang lebih krusial, sistem mendelegasikan beban penyimpanannya pada *node Database Server* terpisah yang menjalankan *Database Management System* (DBMS) MySQL. Seluruh aktivitas manipulasi data dari aplikasi Laravel akan dikirim ke "Ade Decoration DB" melalui jalur komunikasi *TCP/IP SQL* di Port standar 3306. Konfigurasi terpisah antara *server* aplikasi dan basis data ini merupakan langkah preventif arsitektural untuk memastikan beban komputasi terdistribusi secara efisien, menjaga skalabilitas aplikasi, serta mengamankan rekam jejak operasional *wedding organizer* dari potensi kelebihan beban server (*overload*).

4.2.2 Desain Pemodelan Data

A. Entity Relationship Diagram

Entity Relationship Diagram (ERD) memvisualisasikan struktur basis data sistem menggunakan notasi *Crow's Foot* untuk memperjelas kardinalitas dan hubungan antar entitas. Pusat dari arsitektur data ini adalah entitas users, yang bertindak sebagai tabel induk untuk menyimpan data seluruh pengguna sistem, baik pasien, dokter, maupun administrator. Entitas ini memiliki hubungan one-to-many dengan berbagai entitas transaksional, yang berarti satu data pengguna dapat melakukan banyak aktivitas.

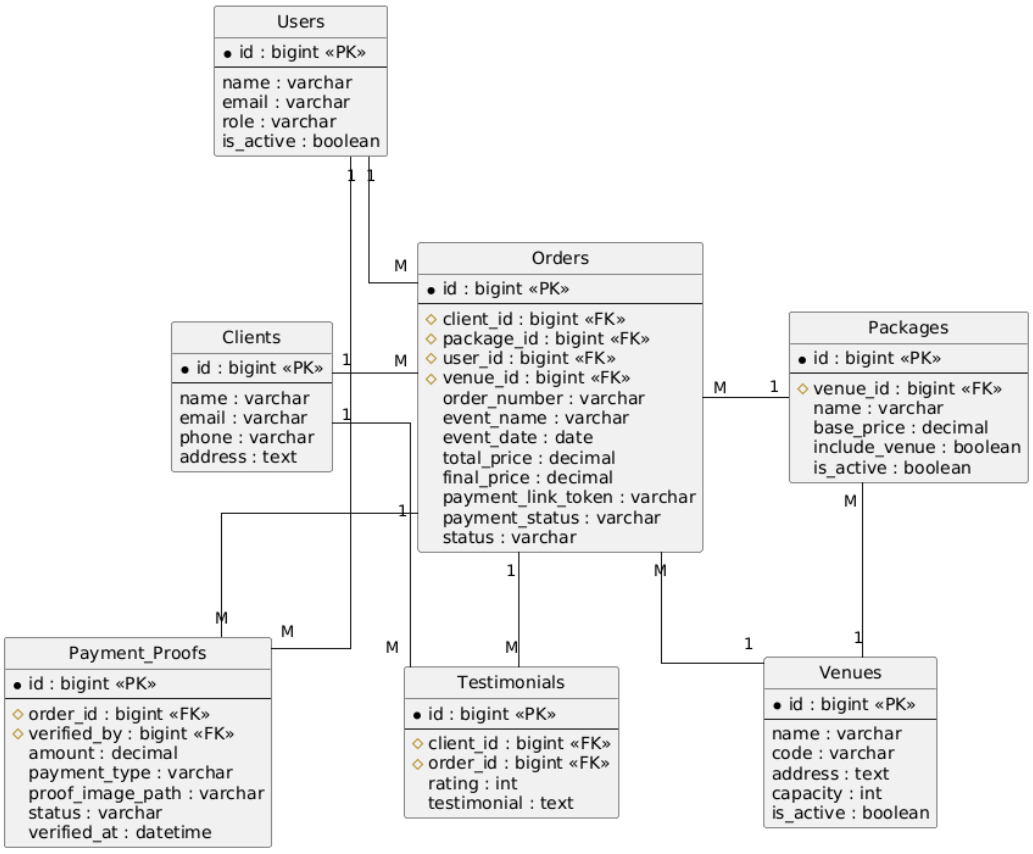


Sumber: Hasil Penelitian, 2026

Gambar IV. 20 Entity Relationship Diagram

B. Logical Record Structure

Logical Record Structure (LRS) merepresentasikan struktur logis basis data yang dirancang untuk mendukung integritas dan relasi antar data dalam sistem Wedding Organizer. Diagram ini memvisualisasikan bagaimana entitas-entitas data saling terhubung melalui atribut kunci, yaitu *Primary Key (PK)* sebagai identitas unik tabel dan *Foreign Key (FK)* sebagai penghubung referensial antar tabel. Struktur ini berpusat pada normalisasi data untuk meminimalkan redundansi dan menjaga konsistensi informasi.



Sumber: Hasil Penelitian, 2026

Gambar IV. 21 Logical Record Structure

Entitas Users bertindak sebagai tabel induk manajerial yang menampung seluruh otorisasi staf internal, mulai dari admin, marketing, hingga pihak *owner* Ade Decoration. Tabel ini memiliki relasi *one-to-many* (1:M) yang sangat krusial terhadap berbagai tabel operasional dan finansial. Sebagai contoh, relasi Users ke Payment_Proofs dan Operational_Costs menunjukkan rekam jejak (*audit trail*) yang jelas mengenai staf mana yang memverifikasi pembayaran klien atau mencatat pengeluaran kas. Di sisi lain, master data pelanggan dipusatkan pada entitas Clients, sedangkan kerangka layanan dasar dikelola melalui keterhubungan antara Venues, Packages, dan Inventory_Items.

Pusat aktivitas transaksional bisnis ini bermuara pada entitas Orders. Entitas ini merupakan tabel relasi sentral yang mengikat Clients (pemesan), Packages (paket yang

dipilih), Venues (lokasi acara), dan Users (staf penanggung jawab). Dari entitas Orders ini, alur data didistribusikan lebih lanjut; rincian tagihan diturunkan ke Order_Details, rekam riwayat transfer direkap dalam Payment_Proofs, dan evaluasi kualitas kepuasan ditampung pada Testimonials. Kardinalitas 1:M yang terbentuk pada relasi-relasi ini memastikan bahwa satu nomor pesanan dapat menampung banyak rincian *item* maupun mengakomodasi pembayaran termin (cicilan) tanpa merusak integritas referensial data.

Pada tahap eksekusi lapangan, operasional dikendalikan melalui entitas Events yang memiliki relasi *one-to-one* (1:1) dengan Orders. Pemetaan ini menegaskan bahwa satu pesanan yang sudah *deal* akan dikonversi menjadi satu jadwal acara spesifik. Acara ini kemudian dipecah ke dalam tabel Rundown_Items untuk menyusun susunan acara, dan diteruskan ke Task_Assignments untuk pembagian tugas spesifik kepada kru. Selain manajemen proyek, *Logical Record Structure* (LRS) ini juga mengakomodasi tata kelola *back-office* melalui entitas Employees dan Employee_Payrolls untuk pencatatan kompensasi gaji. Keseluruhan struktur relasi antar-tabel ini menjamin bahwa setiap siklus bisnis WO—mulai dari reservasi awal hingga eksekusi acara dan laporan keuangan—terkunci secara aman melalui mekanisme *Primary Key* (PK) dan *Foreign Key* (FK).

C. Spesifikasi File Database

Berikut adalah spesifikasi file (tabel) yang terdapat dalam basis data db_sistem_wo. Spesifikasi ini menjelaskan struktur penyimpanan fisik data.

1. Spesifikasi File Tabel Users (Pengguna)

Tabel ini menyimpan data seluruh pengguna sistem, termasuk Super Admin (Owner), Admin, Marketing, Sales, Staff, dan Klien.

a. Nama Database : sistedekor

b. Nama File : Tabel Users

c. Akronim : users.ibd

d. Tipe File : File Master

e. Akses File : Random

f. Panjang Record : 750 Byte (Estimasi)

g. Kunci Field : id

Tabel VI. 10 Spesifikasi File Tabel Users

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key	Keterangan
1	id	BigInt	20	Primary Key	ID Unik Pengguna
2	name	Varchar	255	-	Nama Lengkap Pengguna
3	email	Varchar	255	Unique Key	Alamat Email Pengguna
4	email_verified_at	Timestamp	-	-	Waktu Verifikasi Email
5	password	Varchar	255	-	Password Terenkripsi (Hash)
6	remember_token	Varchar	100	-	Token Pengingat Login
7	role	Enum	-	-	Role ('super_admin', 'admin', 'manager', 'staff', 'employee', 'sales', 'marketing', 'client', 'user')
8	is_active	TinyInt / Bool	1	-	Status Keaktifan Akun (1=Aktif, 0=Nonaktif)
9	created_at	Timestamp	-	-	Waktu Pembuatan Record
10	updated_at	Timestamp	-	-	Waktu Pembaruan Record

Sumber: Hasil Penelitian, 2026

2. Spesifikasi File Tabel Client (Pelanggan)

Tabel ini menyimpan data profil lengkap klien yang mendaftar atau melakukan pemesanan dekorasi.

a. Nama Database : sistedekor

b. Nama File : Tabel Clients

c. Akronim : clients.ibd

d. Tipe File : File Master

e. Akses File : Random

f. Panjang Record : 800 Byte (Estimasi)

g. Kunci Field : id

Tabel VI. 11 Spesifikasi File Tabel Client

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key	Keterangan
1	id	BigInt	20	Primary Key	ID Unik Klien
2	name	Varchar	255	-	Nama Lengkap Klien
3	phone	Varchar	50	-	Nomor Telepon Klien
4	email	Varchar	255	-	Alamat Email Klien
5	address	Text	-	-	Alamat Tempat Tinggal Klien
6	created_at	Timestamp	-	-	Waktu Pembuatan Record
7	updated_at	Timestamp	-	-	Waktu Pembaruan Record

Sumber: Hasil Penelitian, 2026

3. Spesifikasi File Tabel Packages (Paket Dekorasi)

Tabel ini menyimpan katalog paket dekorasi beserta harga dasar dan detail fasilitas venue.

a. Nama Database : sistedekor

b. Nama File : Tabel Packages

c. Akronim : packages.ibd

d. Tipe File : File Master

e. Akses File : Random

f. Panjang Record : 1200 Byte (Estimasi)

g. Kunci Field : id

Tabel VI. 12 Spesifikasi File Tabel Packages

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key	Keterangan
1	id	BigInt	20	Primary Key	ID Unik Paket Dekorasi
2	name	Varchar	255	-	Nama Paket Dekorasi
3	slug	Varchar	255	Unique Key	Slug URL Paket Dekorasi
4	description	Text	-	-	Deskripsi Kelengkapan Paket
5	base_price	Decimal	15,2	-	Harga Dasar Paket
6	includes_venue	TinyInt / Bool	1	-	Menyertakan Sewa Gedung (1=Ya, 0=Tidak)
7	venue_id	BigInt	20	Foreign Key	ID Gedung Bekerjasama
8	venue_price	Decimal	15,2	-	Harga Tambahan Gedung
9	is_active	TinyInt / Bool	1	-	Status Keaktifan Paket
10	created_at	Timestamp	-	-	Waktu Pembuatan Record
11	updated_at	Timestamp	-	-	Waktu Pembaruan Record

Sumber: Hasil Penelitian, 2026

4. Spesifikasi File Tabel Orders (Pemesanan)

Tabel ini menyimpan data transaksi pemesanan dekorasi beserta status negosiasi keuangan.

a. Nama Database : sistedekor

b. Nama File : Tabel Orders

c. Akronim : orders.ibd

d. Tipe File : File Transaksi

e. Akses File : Random

f. Panjang Record : 2500 Byte (Estimasi)

g. Kunci Field : id

Tabel VI. 13 Spesifikasi File Tabel Orders

No	Nama Field	Tipe Data	Ukuran Field	Jenis Key	Keterangan
1	id	BigInt	20	Primary Key	ID Unik Pemesanan
2	order_number	Varchar	100	Unique Key	Kode Nomor Transaksi Pemesanan
3	client_id	BigInt	20	Foreign Key	ID Klien Pemesan
4	package_id	BigInt	20	Foreign Key	ID Paket yang Dipesan
5	user_id	BigInt	20	Foreign Key	ID Karyawan Penerima Input
6	venue_id	BigInt	20	Foreign Key	ID Gedung yang Dipilih
7	event_name	Varchar	255	-	Nama Acara
8	event_date	Date	-	-	Tanggal Pelaksanaan Acara
9	final_price	Decimal	15,2	-	Harga Akhir Kesepakatan
10	remaining_amount	Decimal	15,2	-	Sisa Tagihan Pembayaran
11	status	Varchar	50	-	Status Progres Operasional Order
12	payment_status	Varchar	50	-	Status Verifikasi Keuangan
13	payment_link_token	Varchar	255	-	Token Khusus Link Pembayaran
14	payment_link_active	TinyInt / Bool	1	-	Status Keaktifan Link Bayar
15	notes	Text	-	-	Catatan Tambahan Pemesanan
16	created_at	Timestamp	-	-	Waktu Pembuatan Record

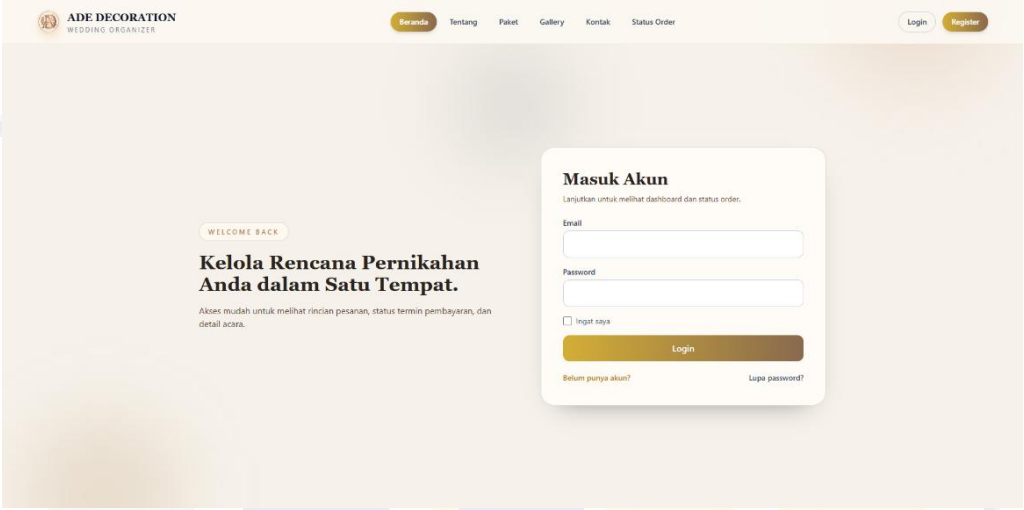
Sumber: Hasil Penelitian, 2026

4.2.3 Desain User Interface

Perancangan antarmuka pengguna (*User Interface Design*) bertujuan untuk memvisualisasikan tampilan sistem yang akan dibangun. Desain ini dirancang dengan prinsip *user-friendly* untuk memastikan kemudahan interaksi bagi seluruh pengguna, baik Administrator, Dokter, Pasien, maupun Pemilik, dalam mengakses fitur-fitur sistem Wedding Organizer. Berikut adalah rancangan antarmuka utama sistem:

1. Tampilan Halaman Login

Implementasi antarmuka halaman *login* pada gambar IV.22 dirancang sebagai gerbang utama sistem yang memuat fungsi krusial dalam hal otentikasi serta otorisasi hak akses pengguna sebelum diarahkan menuju halaman dasbor utama.



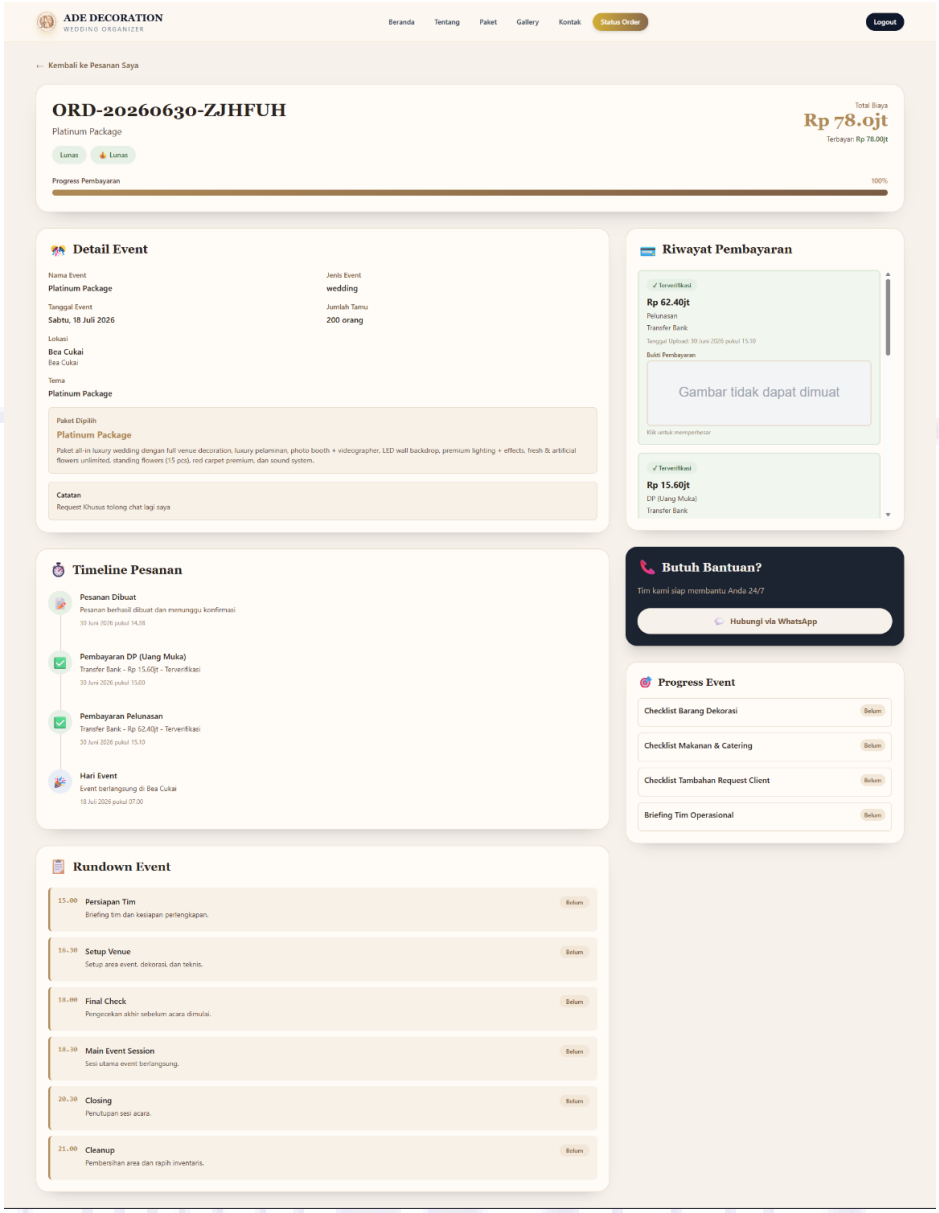
Sumber: Hasil Penelitian, 2026

Gambar IV. 22 Tampilan Halaman Login

Secara fungsional, antarmuka ini mengintegrasikan komponen formulir *input* berupa alamat email dan kata sandi yang terhubung langsung dengan basis data untuk memvalidasi kredensial pengguna, baik bagi pelanggan maupun manajemen Ade Decoration.

2. Tampilan Halaman Status Order (Client)

Implementasi antarmuka halaman detail status pesanan pada gambar IV.23 dirancang sebagai media transparansi informasi bagi pelanggan (*client*) untuk memantau seluruh perkembangan persiapan acara mereka secara *real-time*.



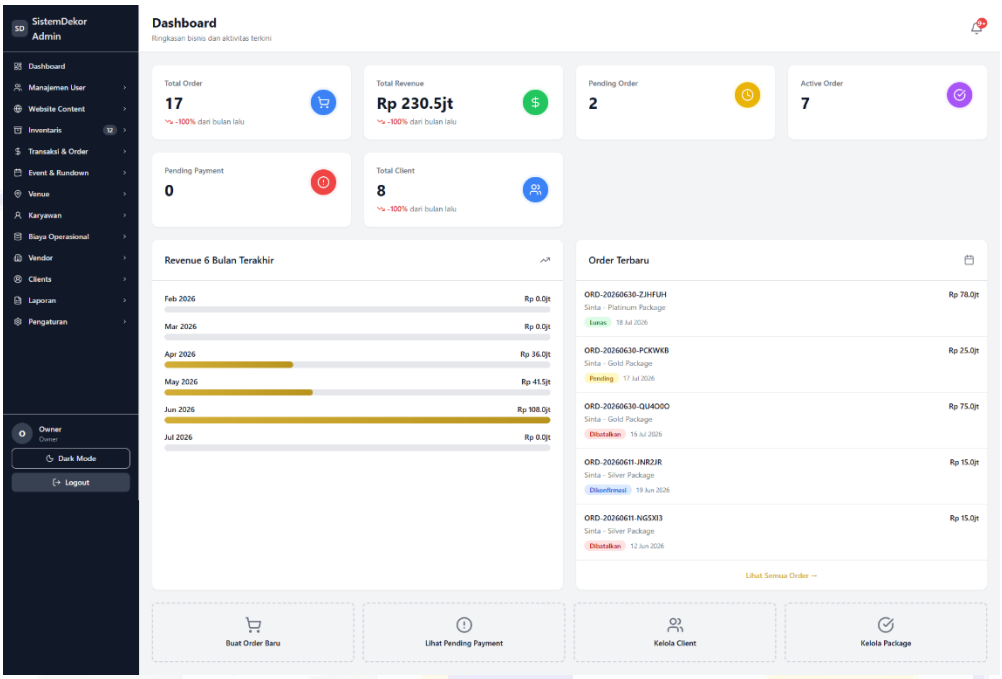
Sumber: Hasil Penelitian, 2026

Gambar IV. 23 Tampilan Halaman Status Order

Antarmuka halaman status pesanan klien secara komprehensif mengintegrasikan ringkasan finansial beserta *progress bar* pelunasan di bagian atas, detail paket dan lini masa kronologis acara di area kiri, serta riwayat pembayaran termin terverifikasi, kontrol logistik, dan tombol komunikasi WhatsApp darurat di area kanan untuk menyajikan transparansi data persiapan pernikahan secara *real-time*.

3. Tampilan Halaman Dashboard (Admin)

Implementasi antarmuka halaman dasbor admin pada gambar IV.24 dirancang sebagai pusat kendali operasional dan pemantauan kinerja bisnis secara terintegrasi bagi pihak manajemen Ade Decoration.



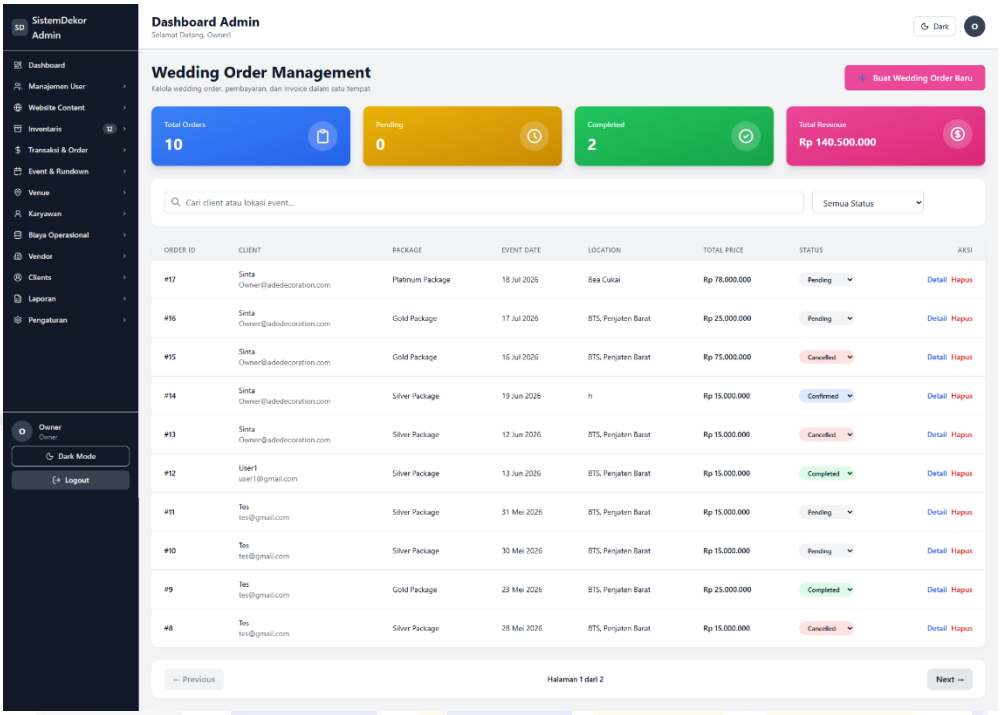
Sumber: Hasil Penelitian, 2026

Gambar IV. 24 Tampilan Halaman Dashboard Admin

Antarmuka halaman dasbor admin mengintegrasikan deretan kartu ringkasan eksekutif finansial dan operasional di bagian atas, grafik tren pendapatan bulanan serta panel aktivitas pemesanan terbaru di area tengah, serta baris tombol tindakan cepat dan bilah navigasi utama di bagian bawah dan kiri sistem untuk mengoptimalkan pusat kendali bisnis Ade Decoration secara terpadu.

4. Tampilan Halaman Wedding Order

Implementasi antarmuka halaman *Wedding Order Management* pada gambar IV.25 dirancang sebagai subsistem pengelolaan data transaksi dan reservasi acara secara terpusat untuk mempermudah admin dalam memantau siklus pemesanan klien Ade Decoration.



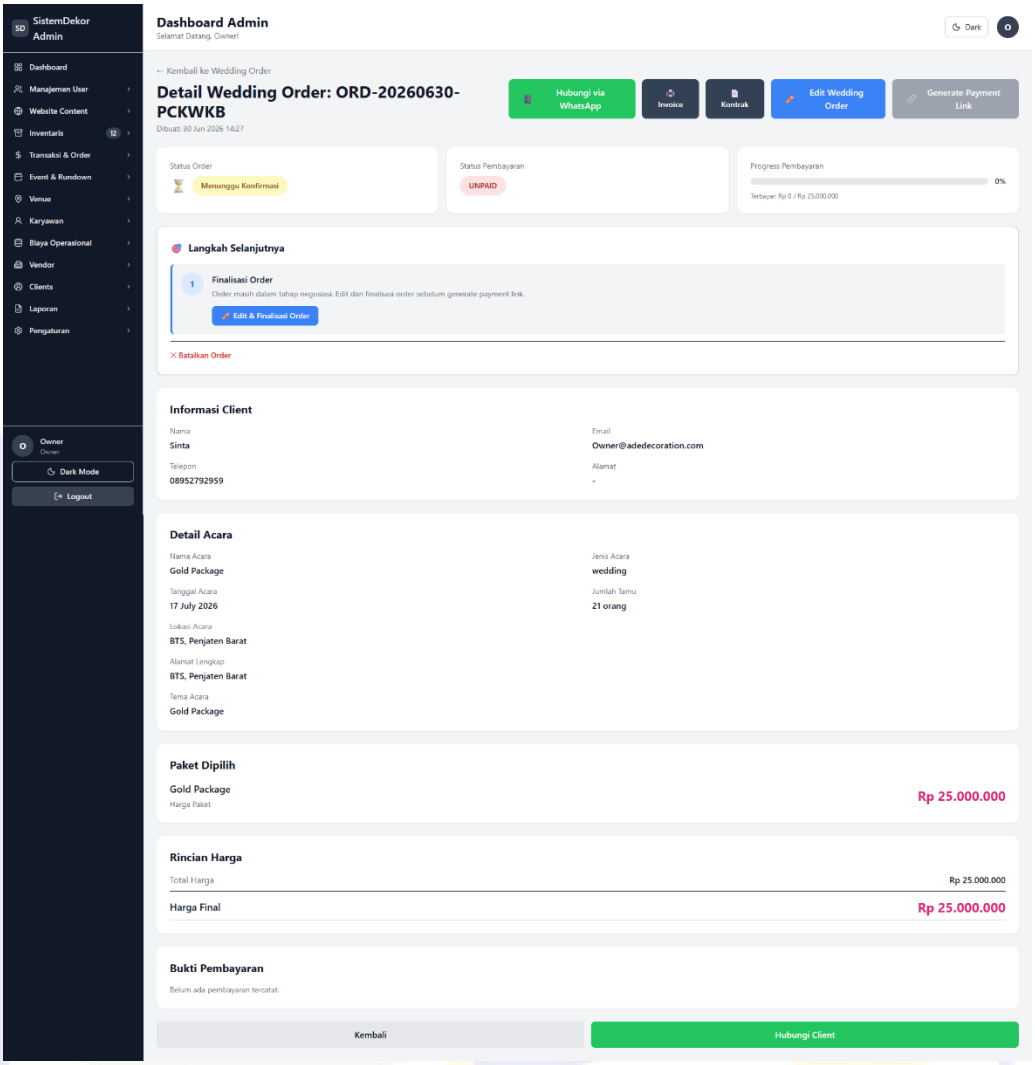
Sumber: Hasil Penelitian, 2026

Gambar IV. 25 Tampilan Halaman Wedding Order

Halaman ini secara menyajikan metrik ringkasan transaksi dan tombol pembuatan order baru pada bagian atas, bilah pencarian beserta filter status di area tengah, serta tabel transaksional komprehensif yang dilengkapi menu perubahan status instan dan fitur *pagination* di bagian bawah untuk mengoptimalkan efisiensi pengelolaan data reservasi.

5. Tampilan Halaman Detail Order

Implementasi antarmuka halaman detail pesanan pada gambar IV.26 dirancang sebagai panel kendali komprehensif bagi admin untuk mengelola seluruh tahapan operasional dan administratif dari suatu order spesifik milik klien.



Sumber: Hasil Penelitian, 2026

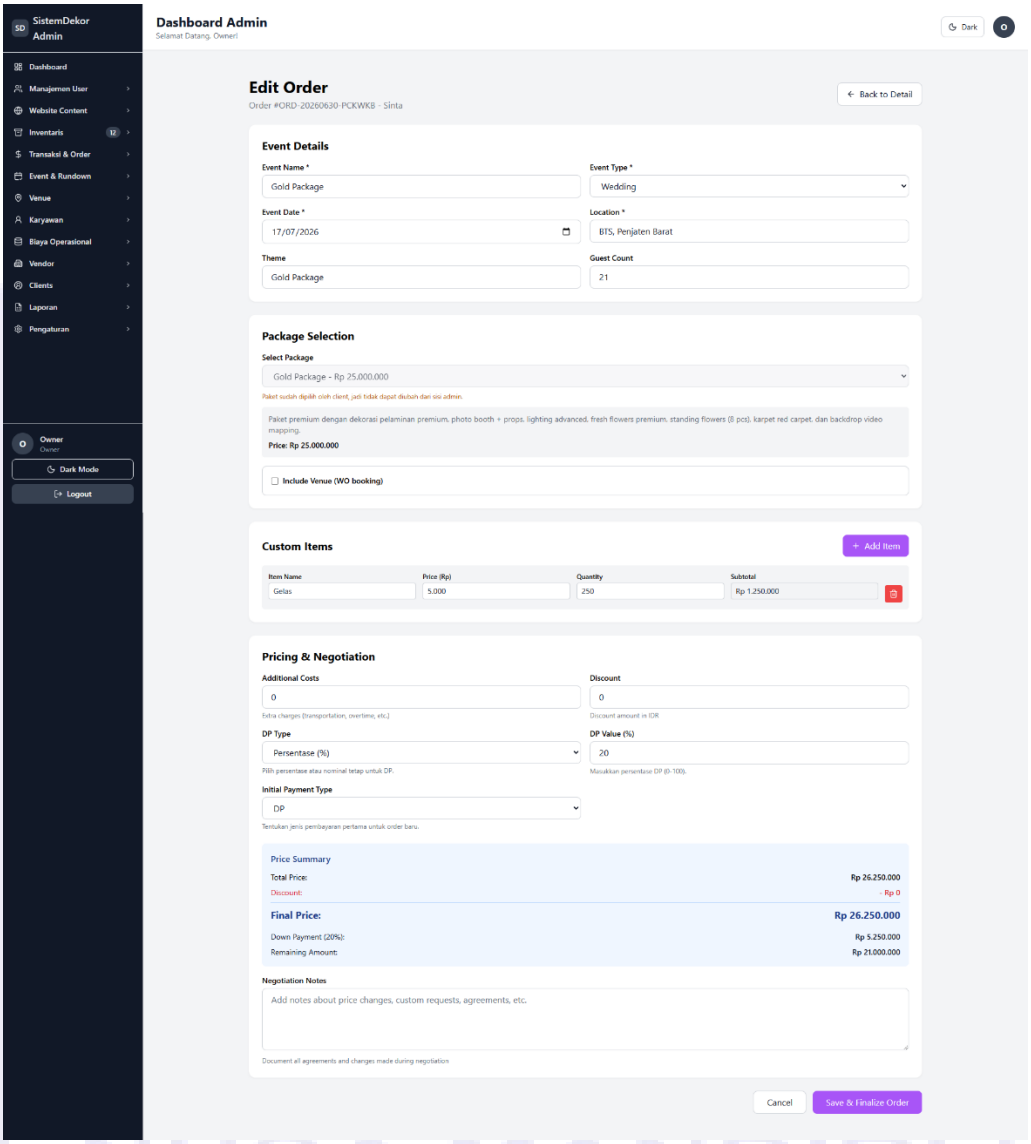
Gambar IV. 26 Tampilan Halaman Detail Order

Halaman detail pesanan secara komprehensif mengintegrasikan kode transaksi beserta tombol aksi dokumen dan komunikasi di bagian atas, kartu indikator status finansial serta panduan alur kerja finalisasi di area tengah, yang didukung oleh panel data terstruktur mengenai identitas klien, rincian teknis acara, kalkulasi harga final, dan riwayat pembayaran termin pada area inti untuk mempermudah kontrol operasional admin.

6. Tampilan Halaman Edit Order

Implementasi antarmuka halaman *Edit Order* pada gambar IV.27 dirancang sebagai modul manipulasi data tingkat lanjut dan pusat negosiasi harga finansial

(*recalculation*) yang digunakan oleh admin untuk memperbarui parameter pesanan secara spesifik.



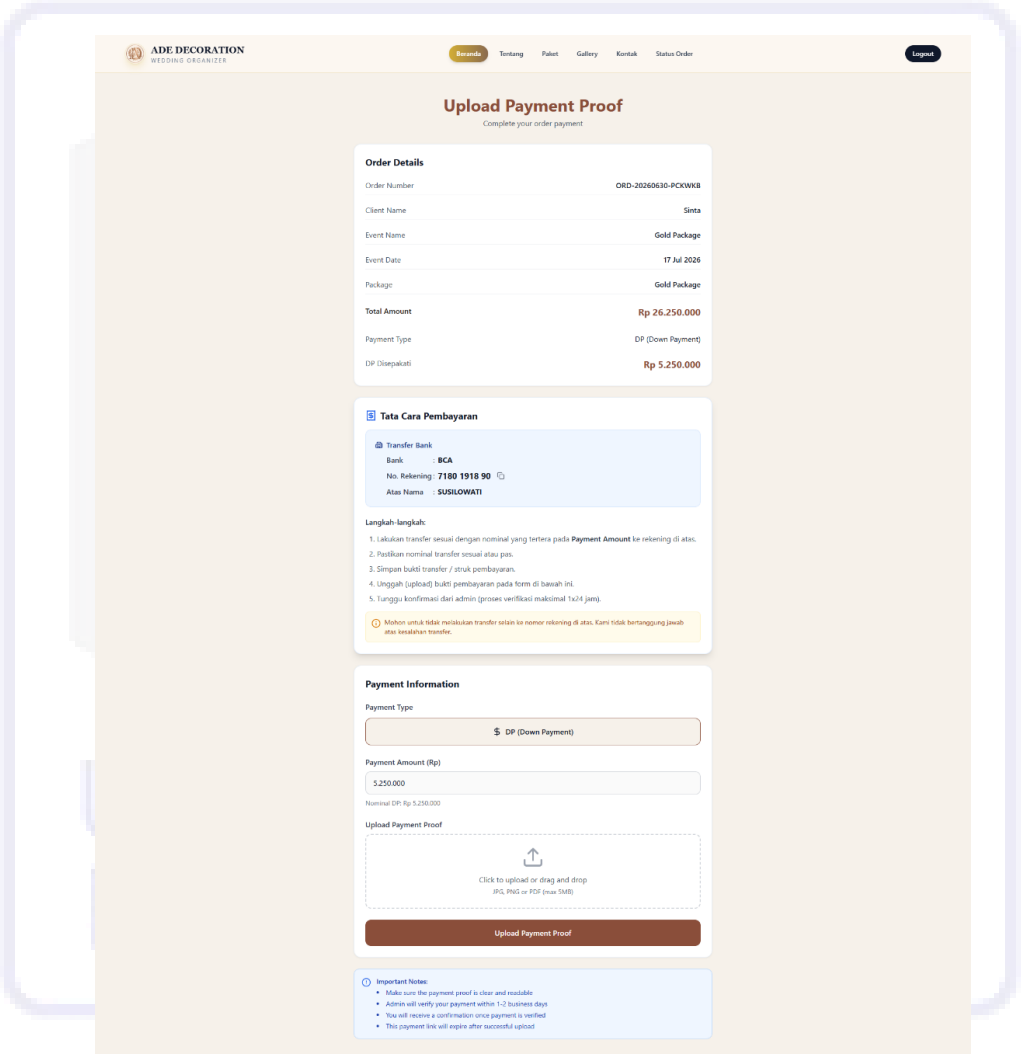
Sumber: Hasil Penelitian, 2026

Gambar IV. 27 Tampilan Halaman Edit Order

Halaman *Edit Order* secara komprehensif mengintegrasikan formulir dinamis detail acara dan pemilihan paket utama di bagian atas, kartu penambahan item kustom secara fleksibel di area tengah, serta panel *Pricing & Negotiation* di bagian bawah yang secara otomatis memicu fungsi kalkulasi ulang (*recalculation*) nominal uang muka dan sisa tagihan berdasarkan input biaya tambahan serta diskon sebelum disimpan ke dalam basis data.

7. Tampilan Halaman Upload Bukti Bayar

Implementasi antarmuka halaman *Upload Payment Proof* (Unggah Bukti Bayar) pada gambar IV.28 dirancang sebagai subsistem pemrosesan transaksi dari sisi pelanggan (*front-end*) yang berfungsi untuk mendokumentasikan serta memvalidasi bukti transfer pembayaran berdasarkan komitmen termin yang telah disepakati.



Sumber: Hasil Penelitian, 2026

Gambar IV. 28 Tampilan Halaman Upload Bukti Bayar

Halaman *Upload Payment Proof* secara terintegrasi menyajikan panel transparansi rincian nominal tagihan dan termin di bagian atas, panduan instruktif beserta informasi rekening bank tujuan transfer di area tengah, serta formulir

pengisian jumlah dana dan area unggah berkas bukti transfer dinamis di bagian bawah untuk memfasilitasi pelaporan transaksi pelanggan sebelum diverifikasi oleh admin.

4.3 Code Generation

Tahapan *Code Generation* atau implementasi kode merupakan fase di mana rancangan desain antarmuka dan logika sistem diterjemahkan ke dalam bahasa pemrograman. SistemWedding Organizer dibangun menggunakan kerangka kerja (*framework*) Laravel dengan bahasa pemrograman PHP untuk sisi *server* (backend) dan HTML/CSS/JavaScript untuk sisi antarmuka (frontend). Berikut adalah beberapa potongan kode utama (*listing program*) yang mengimplementasikan fungsi-fungsi krusial dalam sistem:

1. Implementasi Logika Autentikasi (Login)

```
// app/Http/Controllers/Auth/AuthenticatedSessionController.php
public function store(LoginRequest $request)
{
    // 1. Verifikasi email & password
    $request->authenticate();
    // 2. Regenerasi sesi (keamanan standar)
    $request->session()->regenerate();
    // 3. Redirect cerdas berdasarkan role
    if ($request->user()->role === 'client') {
        return redirect()->intended(route('client.dashboard'));
    }
    return redirect()->intended(route('admin.dashboard'));
}
```

2. Implementasi Logika Pembuatan Pesanan

```
// app/Http/Controllers/OrderController.php
public function store(Request $request)
{
    // 1. Buat pesanan dengan status awal 'pending'
    $order = Order::create([
        'order_code' => 'ORD-' . date('Ymd') . '-' . strtoupper(Str::random(6)),
        'client_id' => auth()->id(),
        'status' => 'pending',
        'event_name' => $request->event_name,
        'event_date' => $request->event_date,
        'total_price' => $request->total_price, // Harga mentah paket
    ]);
    // 2. Simpan item/dekorasi yang dipilih
    $order->items()->createMany($request->items);
    return redirect()->route('client.orders.show', $order);
}
```

3. Implementasi Logika Proses Pesanan

```
// app/Http/Controllers/OrderController.php
public function update(Request $request, Order $order)
{
    // 1. Update harga dan setujui pesanan
    $order->update([
        'status' => 'confirmed', // Menunggu pembayaran
        'final_price' => $request->final_price, // Harga setelah nego/diskon
```

UNIVERSITAS NUSA MANDIRI

```
        'dp_amount' => $request->dp_amount,     // Ketentuan DP
        // 2. Generate token unik untuk form upload struk klien
        'payment_link_token' => Str::random(64),
        'payment_link_active' => true,
    ]);
}
```

4. Implementasi Logika Simpan Bukti Bayar

```
// app/Http/Controllers/PaymentProofController.php
public function upload(Request $request, $token)
{
    $order = Order::where('payment_link_token', $token)->firstOrFail();
    // 1. Simpan gambar struk ke storage
    $path = $request->file('proof_file')->store('payment-proofs', 'public');
    // 2. Catat riwayat transaksi dengan status 'pending' (menunggu verifikasi Admin)
    $order->paymentTransactions()->create([
        'amount' => $request->amount,
        'proof_file_path' => $path,
        'payment_type' => $request->payment_type, // 'dp' atau 'pelunasan'
        'status' => 'pending',
    ]);
    // 3. Matikan sementara link pembayaran agar tidak dobel submit
    $order->update(['payment_link_active' => false]);
    return back()->with('success', 'Bukti pembayaran berhasil diunggah.');
}
```

5. Implementasi Logika Verifikasi Pembayaran

```
// app/Http/Controllers/PaymentProofController.php
public function verify(Request $request, Order $order, PaymentTransaction $transaction)
{
    // 1. Ubah status struk transaksi menjadi 'Sah / Verified'
    $transaction->update(['status' => 'verified']);
    // 2. Hitung total semua uang yang sudah masuk secara sah
    $totalPaid = $order->paymentTransactions()->where('status', 'verified')->sum('amount');
    // 3. Update status Order berdasar total uang yang masuk
    if ($totalPaid >= $order->final_price) {
        $order->update(['status' => 'completed']); // Lunas
    } else {
        $order->update(['status' => 'processing']); // Baru DP (Sedang Digarap)
    }
    // 4. Otomatis menjadwalkan ke Kalender Event (Jika ini pembayaran pertama)
    Event::firstOrCreate(
        ['order_id' => $order->id],
        [
            'title' => 'Event: ' . $order->event_name,
            'start_date' => $order->event_date,
            'status' => 'scheduled',
        ]
    );
```

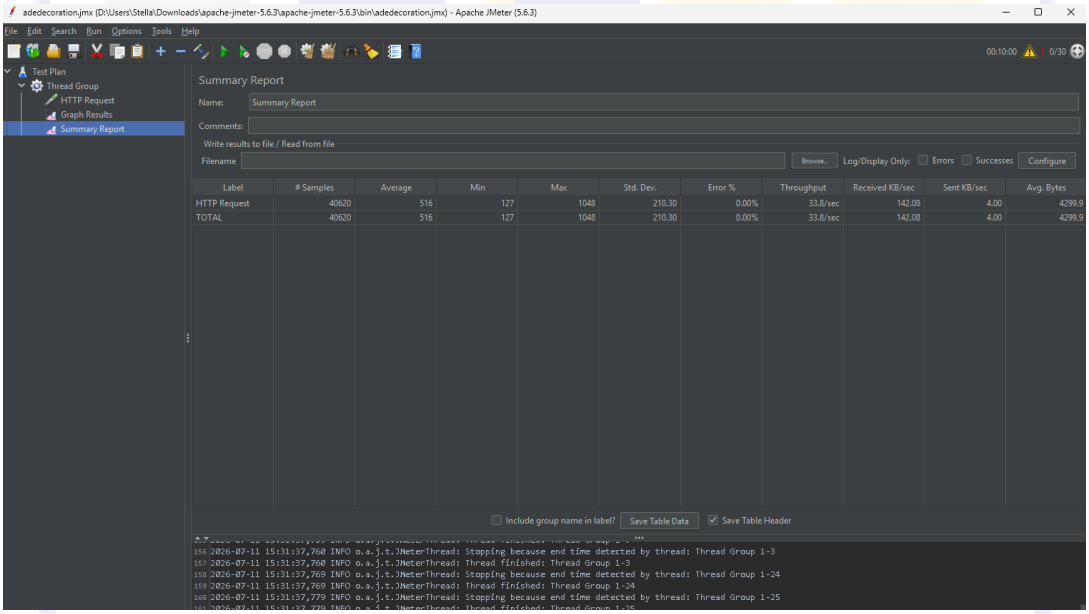
```
        return back()->with('success', 'Pembayaran dikonfirmasi!');
    }
```

4.4 Testing

Tahap pengujian dilakukan untuk memastikan sistem berjalan sesuai dengan kebutuhan fungsional yang telah didefinisikan sebelumnya.

4.4.1 Pengujian Performace Website

Pengujian performa dilakukan untuk mengetahui kemampuan sistem dalam menangani sejumlah permintaan (request) yang dikirimkan secara bersamaan. Pada penelitian ini, pengujian dilakukan menggunakan aplikasi Apache JMeter dengan metode HTTP Request terhadap halaman sistem yang telah dikembangkan.

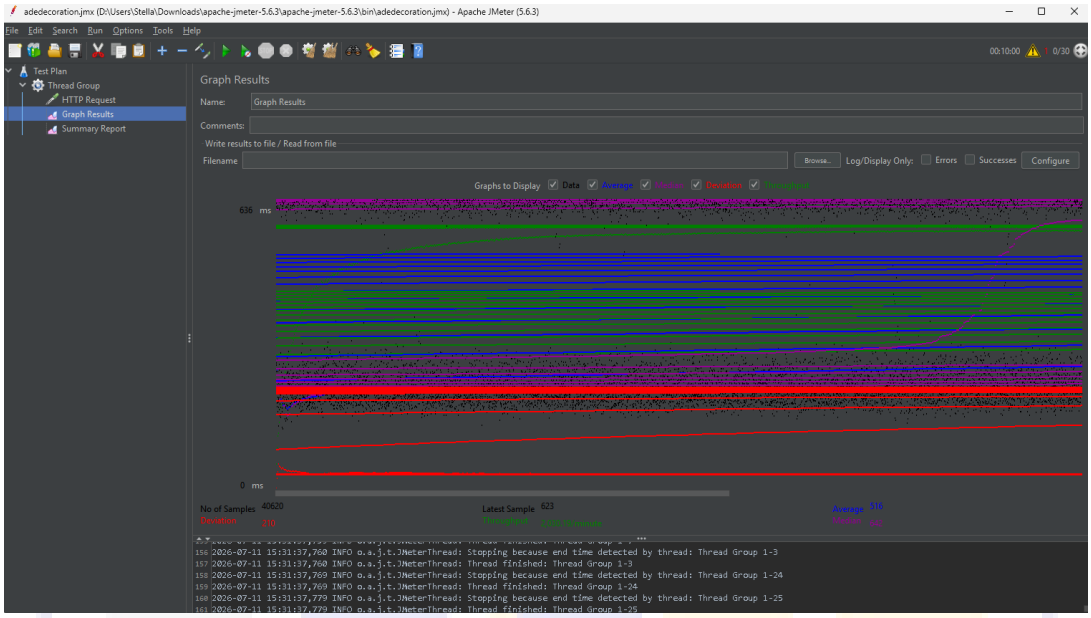


Sumber: Hasil Penelitan (2026)

Gambar IV. 29 Pengujian Apache JMeter (1)

Berdasarkan hasil pengujian, sistem berhasil memproses 40.620 request tanpa mengalami kegagalan (Error = 0,00%). Waktu respons rata-rata (Average) yang diperoleh sebesar 516 ms, dengan waktu respons tercepat 127 ms dan waktu respons

terlama 1.048 ms. Selain itu, nilai Throughput mencapai 33,8 request per detik, yang menunjukkan jumlah permintaan yang mampu diproses sistem setiap detiknya.



Sumber: Hasil Penelitian, 2026

Gambar IV. 30 Pengujian Apache JMeter (2)

Hasil tersebut menunjukkan bahwa aplikasi mampu melayani permintaan pengguna dengan baik selama proses pengujian berlangsung. Tidak ditemukannya error selama pengujian juga menunjukkan bahwa sistem dapat berjalan secara stabil pada skenario pengujian yang dilakukan.

4.4.2 Pengujian Keamanan Website

Pengujian keamanan dilakukan menggunakan OWASP Zed Attack Proxy (OWASP ZAP). Tujuan pengujian ini adalah untuk mengidentifikasi potensi kerentanan keamanan yang masih terdapat pada aplikasi web sehingga dapat diketahui risiko yang mungkin muncul ketika sistem digunakan.

Alert Counts by Alert Type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Medium	3 (50.0%)
Sub Resource Integrity Attribute Missing	Medium	1 (16.7%)
Cookie No HttpOnly Flag	Low	2 (33.3%)
X-Content-Type-Options Header Missing	Low	1 (16.7%)
Modern Web Application	Informational	3 (50.0%)
Session Management Response Identified	Informational	2 (33.3%)
Total		6

Sumber: Hasil Penelitian, 2026

Gambar IV. 31 Pengujian OWASP ZAP

Berdasarkan hasil tersebut, beberapa kerentanan yang ditemukan masih tergolong risiko Medium dan Low. Kerentanan seperti Content Security Policy (CSP) Header Not Set menunjukkan bahwa aplikasi belum menerapkan header keamanan CSP sehingga browser belum memiliki aturan untuk membatasi sumber konten yang dapat dijalankan. Selain itu, ditemukan juga Sub Resource Integrity Attribute Missing yang menunjukkan belum adanya mekanisme verifikasi integritas terhadap resource eksternal.

Pada tingkat risiko Low, OWASP ZAP menemukan bahwa cookie belum menggunakan atribut HttpOnly, sehingga dalam kondisi tertentu cookie masih dapat diakses melalui JavaScript. Selain itu, aplikasi juga belum menerapkan header X-Content-Type-Options, yang berfungsi untuk mencegah browser melakukan MIME type sniffing.

Sementara itu, alert dengan kategori Informational hanya memberikan informasi mengenai karakteristik aplikasi web, seperti penggunaan mekanisme session management dan identifikasi bahwa aplikasi merupakan aplikasi web modern. Alert pada kategori ini tidak menunjukkan adanya kerentanan keamanan yang perlu ditangani.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem telah berjalan dengan cukup baik dari sisi keamanan karena tidak ditemukan kerentanan dengan tingkat risiko High maupun Critical. Meskipun demikian, beberapa rekomendasi dari OWASP ZAP, seperti penambahan Content Security Policy (CSP), penggunaan atribut HttpOnly pada cookie, penerapan Subresource Integrity (SRI), serta penambahan header X-Content-Type-Options, dapat dipertimbangkan untuk meningkatkan keamanan sistem pada pengembangan berikutnya

4.4.3 Pengujian Keterimaan Sistem

Pengujian ini menggunakan metode *User Acceptance Test* (UAT) untuk memverifikasi apakah sistem dapat diterima oleh pengguna akhir.

Tabel VI. 14 Dokumen User Acceptace Test

Dokumen *User Acceptance Testing*						
Nama Proyek	:	Perancangan Sistem Informasi Manajemen Layanan Dan Konsultasi Berbasis Web Dengan Metode Extreme Programming	[LOGO]			
Studi Kasus/Mitra	:	Wedding Organizer Ade Decoration				
Manajer Proyek	:	Edom				
Proses Pengujian Admin						
No	***Use Case***	**Deskripsi Pengujian**	**Hasil Uji**	**Nama Penguji**	**Tanggal Pengujian**	**Catatan Penguji**
1	Login Admin	Admin melakukan login	Berhasil	Susi	24 Juni 2026	Berhasil masuk ke

		menggunakan akun yang telah terdaftar				dashboard admin sesuai hak akses.
2	Kelola Paket	Admin menambah, mengubah, menghapus, dan melihat data paket wedding	Berhasil	Susi	24 Juni 2026	Perubahan data paket tersimpan dengan baik
3	Kelola Data Client	Admin melihat serta mengelola data pelanggan yang telah melakukan registrasi	Berhasil	Susi	24 Juni 2026	Data pelanggan tampil sesuai database
4	Kelola Wedding Order	Admin menerima, mengubah, dan memperbarui data pesanan wedding	Berhasil	Susi	25 Juni 2026	Status pesanan berhasil diperbarui sesuai perubahan pelanggan
5	Kelola Mini Order	Admin membuat pesanan mini order tanpa proses registrasi client	Berhasil	Susi	25 Juni 2026	Mini order berhasil dibuat dan tersimpan
6	Generate Link Pembayaran	Admin membuat link pembayaran untuk client	Berhasil	Susi	25 Juni 2026	Link pembayaran berhasil dibuat dan dapat diakses client
7	Verifikasi Pembayaran	Admin memverifikasi bukti pembayaran yang diunggah client	Berhasil	Susi	25 Juni 2026	Status pembayaran berubah sesuai hasil verifikasi
8	Kelola Invoice	Admin membuat dan mencetak invoice transaksi	Berhasil	Susi	25 Juni 2026	Invoice berhasil dibuat dalam format PDF
9	Cetak Laporan Transaksi	Admin menampilkan dan mencetak laporan transaksi berdasarkan periode	Berhasil	Susi	25 Juni 2026	Laporan sesuai data transaksi yang tersimpan.

Proses Pengujian Client						
No	***Use Case***	**Deskripsi Pengujian**	**Hasil Uji**	**Nama Penguji**	**Tanggal Pengujian**	**Catatan Penguji**

1	Registrasi Akun	Client melakukan registrasi akun baru	Berhasil	Dhika	25 Juni 2026	Akun berhasil dibuat dan dapat digunakan untuk login
2	Login Client	Client melakukan login ke sistem	Berhasil	Dhika	25 Juni 2026	Login berhasil dan diarahkan ke dashboard client
3	Melakukan Pemesanan	Client memilih paket dan mengirimkan data pemesanan	Berhasil	Dhika	25 Juni 2026	Pesanan berhasil dikirim ke admin
4	Melihat Status Pesanan	Client melihat perkembangan status pemesanan	Berhasil	Dhika	25 Juni 2026	Status berubah sesuai proses admin
5	Upload Bukti Bayar	Client mengunggah bukti transfer pembayaran	Berhasil	Dhika	25 Juni 2026	File berhasil diunggah dan menunggu verifikasi admin
6	Melihat Invoice	Client melihat atau mengunduh invoice transaksi	Berhasil	Dhika	25 Juni 2026	Invoice dapat ditampilkan dan diunduh
Proses Pengujian Owner						
No	***Use Case***	**Deskripsi Pengujian**	**Hasil Uji**	**Nama Penguji**	**Tanggal Pengujian**	**Catatan Penguji**
1	Login Owner	Owner melakukan login ke sistem	Berhasil	Edom	25 Juni 2026	Owner berhasil masuk ke dashboard
2	Melihat Dashboard	Owner melihat ringkasan transaksi dan statistik	Berhasil	Edom	25 Juni 2026	Data dashboard sesuai kondisi sistem
3	Melihat Laporan Transaksi	Owner melihat laporan transaksi berdasarkan periode.	Berhasil	Edom	25 Juni 2026	Laporan sesuai dengan data transaksi
4	Mencetak Laporan	Owner mencetak laporan transaksi.	Berhasil	Edom	25 Juni 2026	Laporan berhasil dicetak dalam format PDF

Sumber: Hasil Penelitian, 2026

4.5 Support

Tahapan *support* atau pemeliharaan merupakan fase penting setelah sistem berhasil dikembangkan dan diuji. Fase ini bertujuan untuk memastikan sistem dapat berjalan stabil di lingkungan produksi, mengakomodasi perubahan teknologi, serta memenuhi kebutuhan pengguna yang mungkin berkembang seiring waktu.

4.5.1 Publikasi Web

Sistem Informasi Manajemen Transaksi Wedding Organizer yang berbasis web (Laravel) akan dipublikasikan (deploy) ke server publik agar dapat diakses secara daring oleh seluruh pengguna dari berbagai lokasi.

1. Domain & Hosting

Sistem akan menggunakan domain resmi (https://adedecoration.my.id) yang terhubung dengan self-fosted mini server yang terhubung ke jaringan internet publik.

2. Konfigurasi Server

Pengaturan permintaan web dikelola menggunakan web server Nginx untuk mendukung framework laravel.

3. Database Migration

Struktur dan skema database yang telah dirancang pada lingkungan pengembangan di migrasikan ke dalam sistem server database produksi.

4. Security (SSL)

Pemasangan sertifikat SSL (Secure Sockets Layer) memanfaatkan fitur universal SSL dari cloudflare, yang memaksakan pengalihan otomatis dari HTTP ke protokol HTTPS.

4.5.2 Spesifikasi Hardware dan Software

Untuk menjamin sistem dapat berjalan dengan optimal, baik pada lingkungan produksi (production environment) maupun saat diakses oleh pengguna, diperlukan pemenuhan kebutuhan spesifikasi perangkat keras dan perangkat lunak yang terstandarisasi.

Berikut adalah spesifikasi hardware dan software yang dibutuhkan untuk menjalankan sistem:

1. Spesifikasi Perangkat Keras (Hardware)

Kebutuhan perangkat keras dalam penelitian ini dibagi menjadi dua bagian, yaitu spesifikasi *Self-Hosted Mini Server* dan spesifikasi perangkat pengguna. Berikut spesifikasinya:

Tabel VI. 15 Spesifikasi Perangkat Keras

Komponen	Spesifikasi Server	Spesifikasi Pengguna
Prosesor	Intel Core i3 / AMD Ryzen 3	Intel Core i3 / AMD Ryzen 3
RAM	8 GB	4 GB
Penyimpanan	SSD 128 GB	HDD/SSD 100 MB
Koneksi Internet	20 Mbps	10 Mbps
Monitor	Opsional	Resolusi 1280 x 720 atau lebih
Perangkat Input	Standar	Standar/Touchscreen

Sumber: Hasil Penelitian, 2026

2. Spesifikasi Perangkat Lunak (Software)

Perangkat lunak digunakan untuk mendukung proses bisnis sistem. Berikut spesifikasinya:

Tabel VI. 16 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi Server	Spesifikasi Pengguna
Sistem Operasi	Linux (Ubuntu/Debian)	Windows 10/11, Linux, MacOS, Android, atau IOS
Web Server	Nginx	-
Database	MySQL / MariaDB	-
Bahasa Pemograman	PHP (8.3 atau terbaru)	-
Framework	Laravel (11.x)	-
Web Browser	-	Google Chrome, Mozzila Firefox, atau Safari

Sumber: Hasil Penelitian, 2026