

Rancang Bangun Sistem Informasi Manajemen Proyek Event Organizer Berbasis Software as a Service Menggunakan Laravel

DOI: <http://dx.doi.org/10.35889/jutisi.v15i3.3678>

Creative Commons License 4.0 (CC BY – NC)

Salman Fahrurroji^{1*}, Yopi Nugraha²

Sistem Informasi, Institut Pendidikan Indonesia, Garut, Indonesia

*e-mail Corresponding Author: salmanfahrurrozy@gmail.com

Abstract

Limited transparency and sub-optimal real-time monitoring within the Event Organizer (EO) industry represent significant challenges that diminish Client trust. This research aims to develop a web-based project management system using a Software as a Service (SaaS) approach to bridge the information gap between admins and Clients. The system is developed using the Agile Scrum methodology, leveraging the Laravel framework and a multi-tenant database architecture to ensure scalability and secure data isolation among Users. Black Box Testing results indicate that all system functionalities operate successfully with a 100% success rate. Performance evaluation Validates the efficiency of the multi-tenant architecture, yielding an average response time of under 3 seconds. Furthermore, the User Acceptance Test (UAT) results achieved a score of 4.4 out of 5.0, with the highest rating in the transparency and monitoring aspect. The implementation of this system not only enhances the operational accountability of EOs and mitigates communication risks but also provides digital infrastructure cost efficiency for MSMEs through a centralized SaaS ecosystem.

Keywords: Project Management; Event Organizer; Saas; Laravel; Multi-tenant

Abstrak

Keterbatasan transparansi dan belum optimalnya pemantauan secara waktu nyata (*real-time*) dalam industri *Event Organizer* (EO) menjadi kendala utama yang menurunkan kepercayaan client. Penelitian ini bertujuan mengembangkan sistem manajemen proyek berbasis web dengan pendekatan *Software as a Service* (SaaS) guna menjembatani kesenjangan informasi (*information gap*) antara admin dan client. Sistem dibangun menggunakan metode *Agile Scrum* dengan memanfaatkan *framework* Laravel dan arsitektur basis data *multi-tenant* untuk menjamin skalabilitas serta isolasi data antar pengguna secara aman. Hasil pengujian *Black Box* menunjukkan fungsionalitas sistem berjalan 100%. Evaluasi performa membuktikan efisiensi arsitektur *multi-tenant* dengan waktu respons rata-rata di bawah 3 detik. Sementara itu, *User Acceptance Test* (UAT) memperoleh skor 4,4 dari skala 5,0, dengan penerimaan tertinggi pada aspek transparansi pemantauan. Implementasi sistem ini tidak hanya memperkuat akuntabilitas operasional EO dan mengurangi risiko miskomunikasi, tetapi juga menawarkan efisiensi biaya infrastruktur digital bagi UMKM melalui ekosistem SaaS terpusat.

Kata kunci: Manajemen Proyek; Event Organizer; Saas; Laravel; Multi-tenant

1. Pendahuluan

Perkembangan teknologi informasi mendorong berbagai sektor industri untuk melakukan transformasi digital, termasuk industri *Event Organizer* (EO) yang memiliki karakteristik proyek dinamis, kolaboratif, dan membutuhkan pengelolaan data secara cepat serta akurat. Sistem informasi manajemen proyek menjadi penting karena mampu membantu organisasi dalam mengelola jadwal, *vendor*, pembayaran, dokumentasi, hingga komunikasi dengan client secara terintegrasi. Selain meningkatkan efisiensi operasional, penerapan sistem berbasis web juga

dapat meningkatkan transparansi dan kualitas layanan kepada pelanggan, sehingga menjadi kebutuhan strategis bagi perusahaan EO di era digital saat ini.

Pada objek kajian penelitian ini, proses manajemen proyek *Event Organizer* masih banyak dilakukan secara manual menggunakan spreadsheet, dokumen terpisah, dan komunikasi melalui aplikasi pesan instan. Kondisi tersebut menimbulkan beberapa permasalahan, seperti kesulitan memantau progres proyek secara *real-time*, risiko kehilangan data, keterlambatan koordinasi antar tim, serta kurangnya transparansi informasi kepada client. Selain itu, proses pelaporan keuangan dan pengelolaan *vendor* belum terintegrasi dalam satu sistem, sehingga aktivitas operasional menjadi kurang efektif dan rentan terjadi kesalahan pencatatan.

Beberapa penelitian sebelumnya telah berupaya menyelesaikan permasalahan serupa terkait sistem informasi manajemen proyek EO. Penelitian oleh [1] mengembangkan sistem informasi *Event Organizer* berbasis *web* dengan metode Waterfall yang berfokus pada pengelolaan data event dan transaksi pelanggan. [2] merancang sistem informasi EO berbasis *web* untuk mendukung administrasi proyek dan pengelolaan jadwal kegiatan. Sementara itu, penelitian oleh [3][4] mengembangkan *E-Marketplace Event Organizer* berbasis *Progressive Web Application* menggunakan metode *Agile Scrum* dengan fitur pemesanan layanan EO secara online. Penelitian-penelitian tersebut menunjukkan bahwa digitalisasi proses EO mampu meningkatkan efisiensi kerja, namun sebagian besar masih berfokus pada kebutuhan internal organisasi (*single-tenant*), belum menyediakan mekanisme transparansi proyek bagi client secara *real-time*, dan belum menerapkan arsitektur *Software as a Service* (SaaS) yang dapat digunakan oleh banyak EO dalam satu platform.

Berdasarkan celah penelitian tersebut, penelitian ini mengusulkan rancang bangun sistem informasi manajemen proyek *Event Organizer* berbasis *Software as a Service* (SaaS) menggunakan *framework* Laravel. Pendekatan SaaS dipilih karena memungkinkan beberapa perusahaan menggunakan satu aplikasi yang sama melalui layanan berbasis internet, sehingga mampu mengurangi kebutuhan infrastruktur, mempermudah pemeliharaan sistem, serta meningkatkan fleksibilitas dan skalabilitas layanan. Penelitian [5] menunjukkan bahwa penerapan model *multi-tenancy* pada aplikasi berbasis Laravel mampu mendukung pengembangan sistem SaaS dengan tetap menjaga isolasi data setiap pengguna, meningkatkan efisiensi penggunaan sumber daya, serta memudahkan pengelolaan banyak pelanggan dalam satu platform. Selain itu, penggunaan *framework* Laravel dinilai mampu menghasilkan aplikasi yang lebih terstruktur melalui penerapan arsitektur *Model-View-Controller* (MVC), sehingga proses pengembangan, pemeliharaan, dan pengembangan lanjutan sistem menjadi lebih efektif. Sejalan dengan itu, penelitian [6] membuktikan bahwa aplikasi monitoring proyek berbasis Laravel mampu meningkatkan integrasi data, mempermudah pemantauan progres pekerjaan secara *real-time*, serta mendukung proses pengambilan keputusan yang lebih cepat dan akurat. Berdasarkan hasil penelitian tersebut, sistem yang diusulkan juga dilengkapi dengan fitur *Client Portal* yang memungkinkan client memantau progres proyek, status pembayaran, serta dokumentasi kegiatan secara *real-time*, sehingga transparansi informasi dan kualitas komunikasi antara penyedia jasa dengan pelanggan dapat ditingkatkan. Oleh karena itu, penelitian ini menawarkan kebaruan berupa integrasi pengelolaan proyek internal *Event Organizer* dengan layanan pemantauan proyek bagi client dalam satu ekosistem SaaS berbasis Laravel yang mendukung penggunaan oleh banyak perusahaan secara bersamaan, sehingga diharapkan mampu mengatasi keterbatasan penelitian sebelumnya yang masih berorientasi pada sistem *single-tenant* dan belum menyediakan transparansi informasi proyek secara optimal.

2. Metodologi

Penelitian ini menggunakan pendekatan *Agile Scrum* dalam proses perancangan dan pengembangan sistem manajemen proyek *Event Organizer* (EO). Metode ini dipilih karena memiliki fleksibilitas tinggi dalam menyesuaikan perubahan kebutuhan serta mendukung proses pengujian fitur secara bertahap. Pengembangan sistem dilakukan melalui siklus iteratif yang disebut *sprint*, dengan durasi masing-masing sekitar dua minggu.

Tahapan penelitian dijalankan secara sistematis sebagai berikut:

1) Analisis Persyaratan (*Product Backlog*)

Tahap inisiasi difokuskan pada identifikasi persyaratan bisnis dan penetapan peran pengguna secara mendalam. Peneliti merumuskan daftar prioritas fitur (*product backlog*) yang mencakup pengelolaan entitas (client, *vendor*, produk), portal pemantauan progres mandiri bagi pelanggan, hingga modul manajemen arus kas. Fokus utama pada fase ini

adalah memastikan kesiapan arsitektur sistem dalam mendukung model *multi-tenant* yang skalabel.

- 2) **Perencanaan Iterasi (*Sprint Planning*)**
Setiap siklus diawali dengan sesi perencanaan untuk menetapkan *item backlog* yang akan dieksekusi berdasarkan skala prioritas. Tahapan ini melibatkan pembagian tugas teknis pada *framework* Laravel serta perancangan skema basis data MySQL yang dikonfigurasi khusus untuk menjamin isolasi data antar *tenant* secara aman dan terpisah[7].
- 3) **Pengembangan dan Pengujian Berkelanjutan (*Sprint Execution*)**
Realisasi fitur, seperti *Dashboard* dan sistem penagihan (*Invoicing*), dilakukan secara iteratif menggunakan bahasa pemrograman PHP di dalam lingkungan *framework* Laravel. Secara simultan, pengujian fungsional diterapkan pada setiap modul guna memvalidasi integritas serta logika sistem sebelum memasuki tahap integrasi yang lebih kompleks[8].
- 4) **Tinjauan dan Mekanisme Umpan Balik (*Sprint Review & Feedback*)**
Evaluasi dilaksanakan pada akhir setiap *sprint* dengan mempresentasikan *prototipe* fungsional kepada para pemangku kepentingan. Masukan yang diperoleh selama sesi tinjauan ini menjadi landasan fundamental dalam melakukan perbaikan serta penyesuaian fitur pada iterasi berikutnya[9].
- 5) **Implementasi Bertahap (*Incremental Deployment*)**
Setelah sistem dinyatakan memenuhi standar penagihan *User Acceptance Test* (UAT), modul-modul fungsional disebarkan ke lingkungan *server cloud*. Tahap akhir penelitian mencakup penyusunan dokumentasi teknis yang komprehensif serta pelaksanaan pelatihan pengguna (*user training*) untuk menjamin keberlanjutan serta optimalisasi operasional sistem[10].

Berdasarkan rangkaian metodologi yang telah dijabarkan, penerapan skema *Agile Scrum* memungkinkan proses desain dan implementasi berlangsung secara iteratif serta responsif terhadap perubahan kebutuhan. Melalui struktur tahapan yang terorganisir mulai dari analisis kebutuhan hingga penyebaran *system* pengembangan dapat dilakukan secara bertahap dengan tetap mengedepankan kualitas dan relevansi terhadap kebutuhan pengguna. Dengan demikian, metode ini dinilai sangat efektif dalam mendukung terciptanya sistem manajemen proyek *Event Organizer* berbasis SaaS yang fleksibel, terintegrasi, dan siap untuk diimplementasikan secara nyata.

3. Hasil dan Pembahasan

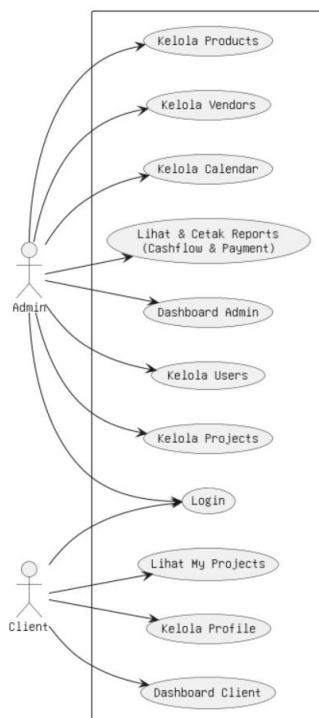
Bagian ini memaparkan luaran dari implementasi sistem manajemen proyek EO berbasis SaaS serta menyajikan analisis komprehensif terhadap serangkaian pengujian yang dilakukan untuk memvalidasi efektivitas sistem secara keseluruhan. Fokus utama evaluasi adalah pada integrasi fungsional antara admin dan client, yang dirancang untuk mengeliminasi hambatan komunikasi melalui sinkronisasi data secara otomatis.

3.1 Analisis Rancangan Sistem

Integrasi fungsional antara admin dan client direpresentasikan melalui pemodelan UML yang telah divalidasi. Fokus utama dari rancangan ini adalah sinkronisasi otomatis pembaruan progres oleh admin yang langsung terefleksi pada portal client, guna mengeliminasi hambatan komunikasi atau *information gap* yang umum ditemukan pada metode konvensional.

1) *Use Case Diagram*

Visualisasi pada Gambar 1 memetakan interaksi aktor utama, yakni Admin dan Client, dalam ekosistem SaaS manajemen proyek EO. Admin memegang otoritas penuh dalam pengelolaan sumber daya, termasuk data pengguna, proyek, *vendor*, serta modul keuangan.

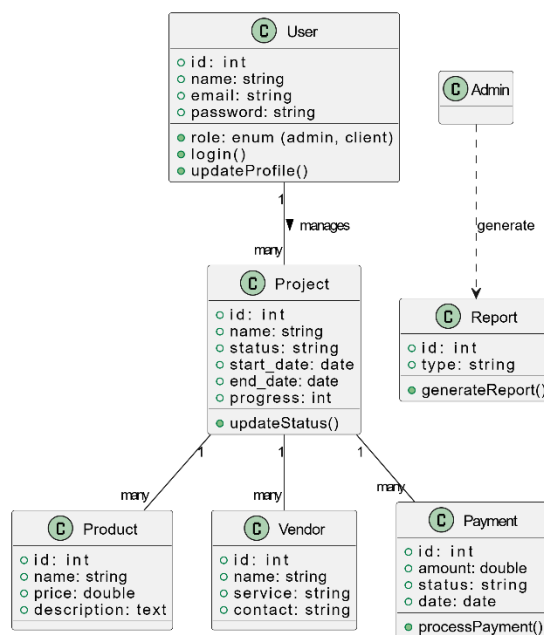


Gambar 1. Use Case Diagram Sistem Manajemen Proyek EO Berbasis SaaS

Di sisi lain, Client memiliki hak akses terbatas yang difokuskan pada pemantauan progres proyek dan manajemen profil personal. Protokol autentikasi melalui proses *Login* diterapkan secara ketat untuk menjamin keamanan dan privasi data di setiap tingkat pengguna.

2) Class Diagram

Struktur statis sistem pada Gambar 2 mengilustrasikan komponen utama seperti *User*, *Project*, *Product*, *Vendor*, *Payment*, dan *Report*. Setiap kelas dilengkapi dengan atribut dan metode spesifik yang mendukung operasional sistem secara keseluruhan.

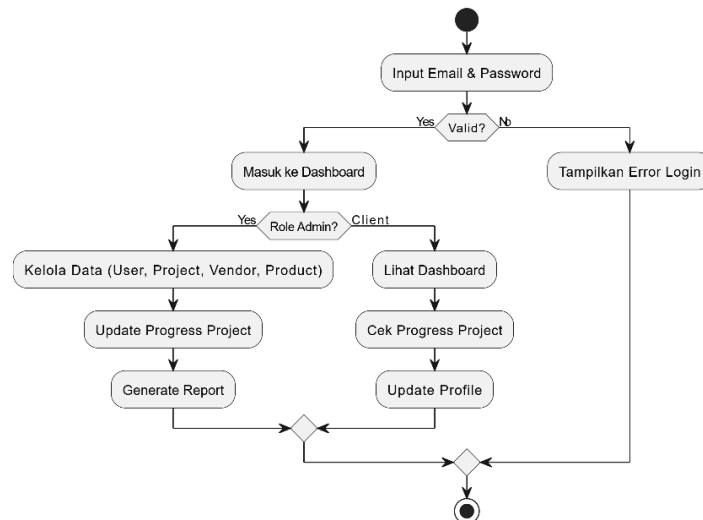


Gambar 2. Class Diagram

Arsitektur ini dirancang secara modular dengan pendekatan *multi-tenant*, yang memungkinkan isolasi data antar entitas tetap terjaga meskipun berada dalam satu platform terintegrasi. Pendekatan berbasis objek ini mempermudah pemeliharaan dan pengembangan fitur di masa depan.

3) Activity Diagram

Alur proses bisnis, mulai dari tahap autentikasi hingga manajemen proyek, digambarkan secara sistematis pada Gambar 3. Diagram ini merinci urutan aktivitas pengguna, *Validasi* data, hingga mekanisme penyajian *output* oleh sistem.

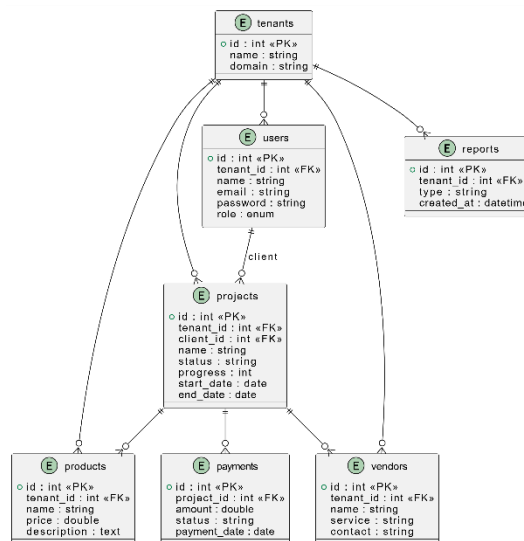


Gambar 3. Activity Diagram

Pemodelan alur yang terstruktur ini berfungsi untuk meminimalkan anomali pada tahap implementasi sekaligus mengidentifikasi titik-titik proses yang dapat dioptimalkan demi efisiensi operasional.

4) ERD (Entity Relationship Diagram)

Struktur basis data pada Gambar 4 mengadopsi skema *multi-tenant* untuk melayani lingkungan SaaS. Setiap tabel utama, seperti *Users*, *Projects*, dan *Vendors*, dihubungkan dengan tabel *tenants* melalui atribut *tenant_id* sebagai kunci isolasi data. Entitas *Projects* bertindak sebagai pusat relasi yang menghubungkan aktivitas manajemen dengan data *vendor*, produk, dan pembayaran, sehingga menjamin konsistensi informasi pada lingkungan data yang kompleks.

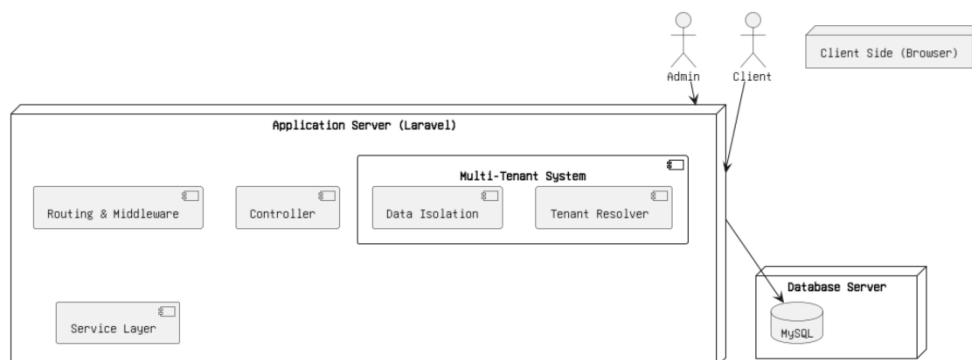


Gambar 4. ERD (Entity Relationship Diagram)

Perancangan ERD dalam penelitian ini mengadopsi konsep *multi-tenant*, di mana setiap data terikat pada identitas tenant tertentu. Pendekatan ini memungkinkan satu sistem digunakan oleh banyak pengguna tanpa mencampurkan data, sehingga meningkatkan efisiensi sekaligus menjaga keamanan dan konsistensi data.

5) Arsitektur SaaS

Penerapan arsitektur *multi-tenant* memberikan keunggulan dalam skalabilitas dan efisiensi manajemen sumber daya *server*.



Gambar 5. Arsitektur SaaS

Berbeda dengan model *single-tenant*, pendekatan ini memungkinkan satu aplikasi melayani banyak organisasi (EO) secara simultan tanpa memerlukan instalasi infrastruktur yang terpisah, sehingga memudahkan pemeliharaan sistem secara terpusat.

Berikut adalah tabel penjelasan detail mengenai aktivitas untuk aktor Admin EO berdasarkan diagram yang telah dibuat sebelumnya:

Tabel 1. Keterangan Aktivitas Admin EO

No	Nama Use Case	Deskripsi Aktivitas
1	<i>Login</i>	Admin memasukkan username dan password untuk memverifikasi identitas agar dapat masuk ke dalam sistem.
2	<i>Dashboard</i>	Admin melihat halaman utama yang berisi ringkasan statistik sistem, seperti jumlah project aktif, user terdaftar, dan notifikasi terbaru.
3	<i>Users</i>	Admin melakukan manajemen data pengguna (CRUD)
4	<i>Projects</i>	Admin mengelola data project acara. Admin dapat membuat project baru, memperbarui status (<i>Pending/In Progress/Done</i>), melihat detail project, serta menghapus project yang batal.
5	<i>Products</i>	Admin mengelola data produk atau paket layanan yang ditawarkan oleh EO (misal: Paket Wedding, Paket Ulang Tahun).
6	<i>Vendors</i>	Admin mengelola database <i>vendor</i> eksternal (pihak ketiga) seperti katering, dekorasi, atau sound system yang bekerja sama dengan EO.
7	<i>Calendar</i>	Admin melihat jadwal acara dalam tampilan kalender untuk memantau tanggal-tanggal penting dan menghindari bentrok jadwal antar project.
8	<i>Reports (Cashflow & Payment)</i>	Admin mencetak atau mengunduh laporan keuangan, yang terdiri dari laporan arus kas (<i>Cashflow</i>) operasional dan laporan status pembayaran client (<i>Payment</i>).

Berikut adalah tabel penjelasan detail mengenai aktivitas untuk *Client* berdasarkan diagram yang telah dibuat sebelumnya:

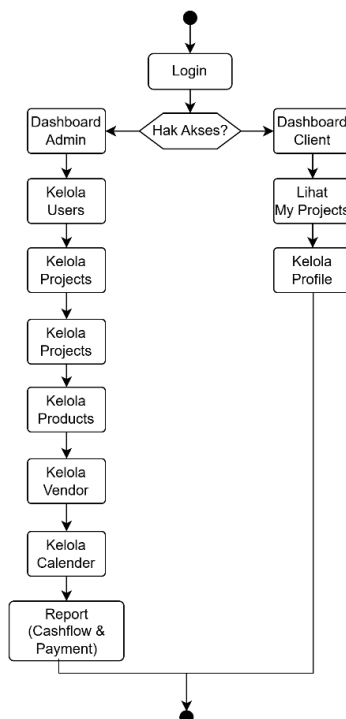
Tabel 2. Keterangan Aktivitas

No	Nama Use Case	Deskripsi Aktivitas
1	<i>Login</i>	<i>Client</i> memasukkan kredensial (<i>username/email</i> dan <i>password</i>) untuk masuk ke dalam area khusus client (<i>Client Portal</i>).

No	Nama Use Case	Deskripsi Aktivitas
2	<i>Dashboard</i>	<i>Client</i> melihat halaman beranda yang menampilkan ringkasan visual, seperti statistik jumlah project berdasarkan status (<i>Pending, In Progress, Done</i>) dan notifikasi terbaru terkait project mereka.
3	<i>My Projects</i>	<i>Client</i> mengakses menu proyek untuk memantau perkembangan acara mereka. Aktivitas ini mencakup melihat daftar <i>project</i> dan membuka Detail <i>Project</i> untuk melihat persentase progres (<i>progress bar</i>), status terkini, dan tanggal pelaksanaan.
4	<i>Profile</i>	<i>Client</i> mengelola informasi akun pribadi mereka. Ini meliputi mengganti foto profil, memperbarui nama, email, atau mengubah kata sandi untuk keamanan.

3.2 Implementasi

Implementasi antarmuka sistem disajikan berdasarkan tahapan operasional aplikasi yang mengacu pada *Use Case Diagram* serta proses bisnis yang telah dirancang sebelumnya. Penyajian dimulai dari halaman *Login* sebagai proses autentikasi pengguna, kemudian dilanjutkan dengan *Dashboard Admin*, pengelolaan data *Users, Projects, Products, Vendor, Calendar*, serta modul *Report* yang mencakup laporan *Cashflow* dan *Payment*. Selanjutnya, implementasi pada sisi client diawali dengan tampilan *Dashboard Client*, kemudian dilanjutkan ke menu *My Projects* dan *Profile*. Penyusunan antarmuka yang mengikuti alur proses bisnis tersebut bertujuan untuk memberikan gambaran yang sistematis mengenai mekanisme pengoperasian aplikasi sehingga setiap fungsi sistem dapat dipahami sesuai dengan kebutuhan pengguna.



Gambar 6. Alur Implementasi Sistem Berdasarkan Proses Bisnis

Pada Gambar 6, Alur implementasi sistem yang disusun berdasarkan proses bisnis serta *Use Case Diagram* yang telah dirancang pada tahap perancangan sistem. Proses implementasi diawali dengan halaman *Login* sebagai mekanisme autentikasi pengguna sebelum memperoleh hak akses ke dalam aplikasi. Setelah proses autentikasi berhasil, sistem akan mengarahkan pengguna ke halaman sesuai dengan peran yang dimiliki. Pengguna dengan hak akses Admin akan diarahkan ke *Dashboard Admin* untuk melakukan pengelolaan data pengguna, proyek, produk, vendor, kalender kegiatan, serta laporan. Sementara itu, pengguna dengan hak akses

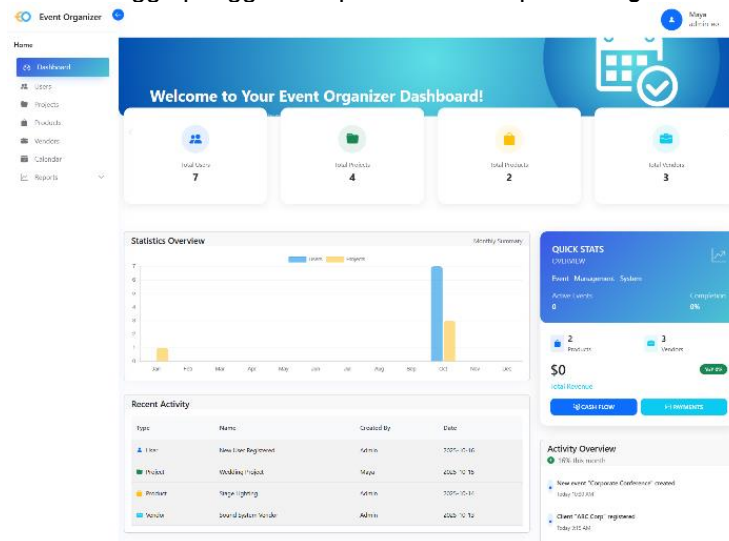
Client akan diarahkan ke *Dashboard Client* untuk memantau perkembangan proyek dan mengelola informasi akun. Penyusunan implementasi berdasarkan alur tersebut bertujuan untuk memberikan gambaran yang sistematis mengenai mekanisme operasional aplikasi sehingga setiap tahapan penggunaan sistem dapat dipahami dengan lebih mudah.



Gambar 7. Login

Pada Gambar 7, halaman *Login* merupakan antarmuka awal yang digunakan sebagai proses autentikasi sebelum pengguna mengakses Sistem Informasi Manajemen Proyek *Event Organizer* berbasis *Software as a Service* (SaaS). Halaman ini berfungsi untuk memverifikasi identitas pengguna melalui alamat *email* dan kata sandi yang telah terdaftar sehingga hanya pengguna yang memiliki hak akses yang dapat menggunakan sistem sesuai dengan perannya.

Proses penggunaan diawali dengan pengguna memasukkan alamat email dan kata sandi pada kolom yang tersedia, kemudian menekan tombol Sign In. Selanjutnya sistem melakukan validasi terhadap data yang dimasukkan. Apabila proses autentikasi berhasil, sistem akan mengarahkan pengguna ke halaman *Dashboard Admin* atau *Dashboard Client* sesuai dengan hak aksesnya. Sebaliknya, apabila data yang dimasukkan tidak valid, sistem akan menampilkan notifikasi kesalahan sehingga pengguna dapat melakukan proses *login* kembali.

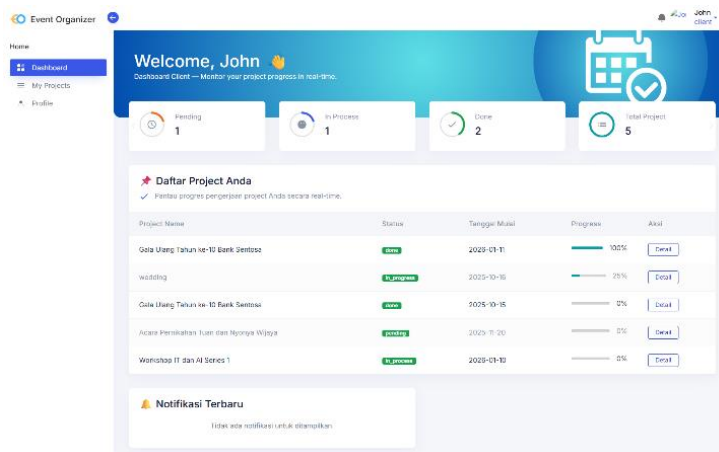


Gambar 8. Dashboard Admin EO

Pada Gambar 8, *Dashboard Admin* merupakan halaman utama yang ditampilkan setelah proses autentikasi berhasil dilakukan. Halaman ini berfungsi sebagai pusat pengelolaan dan pemantauan aktivitas sistem dengan menyajikan informasi secara *real-time*, seperti jumlah pengguna, jumlah proyek, produk atau layanan, vendor, statistik perkembangan proyek, aktivitas terbaru, serta ringkasan kondisi sistem. Informasi tersebut membantu admin dalam melakukan monitoring dan pengambilan keputusan selama proses pengelolaan proyek.

Prosedur penggunaan dimulai ketika admin berhasil masuk ke dalam sistem. Dashboard akan menampilkan ringkasan informasi operasional yang dapat digunakan sebagai dasar untuk mengelola seluruh fitur aplikasi. Melalui menu navigasi yang tersedia di sisi kiri, admin dapat

mengakses modul *Users*, *Projects*, *Products*, *Vendor*, *Calendar*, dan *Report* sesuai dengan kebutuhan operasional. Setiap perubahan data yang dilakukan pada modul tersebut akan tersimpan pada basis data dan ditampilkan kembali secara otomatis sehingga informasi yang tersedia selalu diperbarui.



Gambar 9. Dashboard Client

Pada Gambar 9, *Dashboard Client* merupakan halaman utama yang ditampilkan setelah client berhasil melakukan proses *login*. Halaman ini berfungsi sebagai media monitoring yang memberikan informasi mengenai perkembangan proyek secara transparan dan *real-time*. Informasi yang disajikan meliputi jumlah proyek berdasarkan status *Pending*, *In Progress*, *Done*, dan *Total Project*, daftar proyek beserta progres pengerjaan, tanggal pelaksanaan, serta notifikasi aktivitas terbaru yang berkaitan dengan proyek.

Prosedur penggunaan dimulai setelah client berhasil melakukan autentikasi. Sistem akan menampilkan *Dashboard Client* yang berisi seluruh informasi proyek yang dimiliki. Selanjutnya client dapat memantau status dan persentase progres setiap proyek, memilih tombol *Detail* untuk melihat informasi proyek secara lebih lengkap, serta menggunakan menu navigasi *My Projects* dan *Profile* guna mengakses riwayat proyek maupun mengelola informasi akun. Melalui dashboard ini, client dapat memantau perkembangan proyek secara mandiri tanpa harus menghubungi pihak *Event Organizer*.

3.3 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi yang telah dirancang dapat berjalan sesuai dengan kebutuhan pengguna serta memenuhi standar kinerja yang diharapkan. Proses pengujian ini mencakup pengujian fungsional menggunakan metode *Black Box Testing*, evaluasi kinerja sistem, serta pengukuran tingkat kepuasan pengguna melalui *User Acceptance Test* (UAT). Dengan dilakukannya pengujian secara menyeluruh, diharapkan sistem yang dikembangkan memiliki tingkat keandalan, efisiensi, dan kualitas yang baik sebelum diimplementasikan secara nyata.

Black Box Testing

Metode *Black Box Testing* diterapkan untuk memverifikasi bahwa setiap modul fungsional dalam sistem beroperasi sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Berdasarkan hasil observasi, seluruh fitur utama mampu menjalankan logika bisnisnya secara akurat tanpa ditemukan anomali atau kesalahan teknis yang signifikan. Temuan ini mengonfirmasi bahwa sistem memiliki tingkat reliabilitas yang tinggi dalam melayani kebutuhan operasional pengguna.

Tabel 3. Testing Dashboard Admin EO

No	Fitur	Skenario Pengujian	Hasil
1	Login	Memasukkan <i>email</i> dan <i>password</i> yang benar (Admin).	Valid
2	Dashboard	Menampilkan jumlah total <i>Users</i> , <i>Projects</i> , <i>Products</i> , dan <i>Vendors</i> dengan benar.	Valid

3	Manajemen <i>Users</i>	CRUD	<i>Valid</i>
4	Manajemen <i>Projects</i>	CRUD	<i>Valid</i>
5	Manajemen <i>Products</i>	CRUD	<i>Valid</i>
6	<i>Calender</i>	Jadwal <i>event</i> tampil pada tanggal yang sesuai di kalender.	<i>Valid</i>
7	<i>Report-Cashflow</i>	Sistem menampilkan total saldo dan mengunduh file laporan.	<i>Valid</i>
8	<i>Report-Payment</i>	Data transaksi yang tampil sesuai dengan rentang tanggal yang dipilih.	<i>Valid</i>

Tabel 4. *Testing Dashboard*

No	Fitur	Skenario Pengujian	Hasil
1	<i>Login</i>	Memasukkan <i>email</i> dan <i>password</i> yang benar (<i>Client</i>).	<i>Valid</i>
2	<i>Dashboard</i>	Menampilkan <i>Dashboard Client</i>	<i>Valid</i>
3	<i>My Projects</i>	Detail <i>project</i> yang spesifik	<i>Valid</i>
4	<i>Profile</i>	Informasi akun <i>Client</i>	<i>Valid</i>

Seluruh skenario pengujian pada fitur *Login*, *Dashboard*, Manajemen Data (CRUD), Kalender, hingga Ekspor Laporan Keuangan menunjukkan status *Valid*. Sistem mampu memproses data dan menampilkan visualisasi sesuai logika bisnis.

Fitur akses Portal Client, pemantauan detail proyek, dan pembaruan profil berhasil dieksekusi tanpa kendala, dengan status hasil pengujian *Valid*.

Evaluasi Kepuasan Pengguna

Validasi kesesuaian sistem terhadap kebutuhan operasional industri diuji menggunakan metode *User Acceptance Test* (UAT).

Tabel 6. Hasil Kepuasan Pengguna

No	Aspek Penilaian	Skor Rata-rata
1	Kemudahan penggunaan	4.4
2	Kecepatan akses	4.2
3	Kejelasan informasi	4.5
4	Transparansi monitoring	4.6
5	Kepuasan keseluruhan	4.3

Berdasarkan hasil pengujian, rata-rata keseluruhan skor kepuasan mencapai 4,4 dari skala 5,0, yang mengindikasikan tingkat penerimaan yang sangat baik secara empiris. Skor tertinggi diraih oleh aspek transparansi monitoring (4,6), yang membuktikan bahwa fitur *Dashboard Client* secara efektif berhasil menjawab permasalahan mendasar terkait kesenjangan informasi (*information gap*) antara *vendor* dan *Client* yang selama ini terjadi pada proses manual. Meskipun demikian, skor pada kecepatan akses (4,2) dan kepuasan keseluruhan (4,3) memperlihatkan adanya ruang untuk penyempurnaan sistem. Berdasarkan umpan balik kualitatif, beberapa pengguna menilai bahwa antarmuka pelaporan kas menyajikan data yang cukup padat sehingga membutuhkan waktu adaptasi visual. Identifikasi batasan ini menjadi temuan penting bagi penelitian untuk iterasi pengembangan UI/UX di masa mendatang tanpa mengurangi komprehensivitas penyajian data.

4.4 Pembahasan

Hasil penelitian menunjukkan bahwa Sistem Informasi Manajemen Proyek *Event Organizer* berbasis *Software as a Service* (SaaS) yang dikembangkan menggunakan *framework* Laravel berhasil memenuhi tujuan penelitian, yaitu menyediakan platform yang mampu mengintegrasikan pengelolaan proyek, vendor, produk, transaksi, serta pemantauan progres proyek dalam satu sistem. Keberhasilan tersebut didukung oleh hasil *Black Box Testing* yang menunjukkan seluruh fitur utama, seperti login, manajemen pengguna, proyek, vendor, produk, kalender, laporan keuangan, dan portal client, berfungsi sesuai kebutuhan tanpa ditemukan kesalahan fungsional. Selain itu, hasil *User Acceptance Test* (UAT) memperoleh nilai rata-rata

4,4 dari skala 5, yang menunjukkan bahwa sistem memiliki tingkat penerimaan pengguna yang sangat baik. Hasil tersebut mengindikasikan bahwa sistem mampu menjawab permasalahan utama yang ditemukan pada proses bisnis *Event Organizer*, yaitu pengelolaan data yang masih terpisah, proses monitoring yang kurang transparan, serta keterlambatan penyampaian informasi kepada client.

Keberhasilan implementasi sistem dipengaruhi oleh penerapan arsitektur SaaS berbasis *multi-tenant* yang memungkinkan beberapa *Event Organizer* menggunakan satu aplikasi secara bersamaan dengan tetap menjaga isolasi data setiap tenant. Pendekatan ini memberikan efisiensi dalam pengelolaan aplikasi karena pemeliharaan sistem, pembaruan perangkat lunak, dan pengelolaan infrastruktur dilakukan secara terpusat. Selain itu, penggunaan *framework* Laravel memberikan struktur pengembangan yang terorganisasi, keamanan autentikasi yang lebih baik, serta kemudahan dalam implementasi hak akses berdasarkan peran pengguna. *Dashboard* yang menyajikan informasi proyek secara *real-time* juga membantu admin dan client memperoleh informasi secara cepat sehingga proses koordinasi menjadi lebih efektif.

Temuan penelitian ini sejalan dengan teori sistem informasi yang dikemukakan oleh [11] yang menyatakan bahwa penerapan teknologi informasi mampu meningkatkan efisiensi proses bisnis dan kualitas informasi yang dihasilkan organisasi. Hasil penelitian juga mendukung penelitian [12] yang menunjukkan bahwa sistem berbasis *web* dapat mempercepat pengolahan data sekaligus mengurangi kesalahan administrasi. Selain itu, penerapan konsep *Software as a Service* pada penelitian ini memperkuat hasil penelitian [13] yang menjelaskan bahwa model SaaS memberikan fleksibilitas akses, kemudahan pemeliharaan, serta efisiensi biaya infrastruktur karena seluruh layanan dijalankan melalui platform berbasis *cloud*.

Jika dibandingkan dengan penelitian [1][2], penelitian ini memiliki persamaan dalam pengembangan sistem informasi berbasis web untuk mendukung aktivitas *Event Organizer*. Namun demikian, penelitian ini menawarkan pengembangan yang lebih komprehensif melalui penerapan arsitektur SaaS *multi-tenant* dan penyediaan *Client Portal* yang memungkinkan client memantau progres proyek secara langsung. Berbeda dengan penelitian terdahulu yang lebih berfokus pada pengelolaan data internal perusahaan, sistem yang dikembangkan pada penelitian ini memberikan transparansi informasi kepada client sehingga komunikasi antara penyelenggara acara dan pelanggan menjadi lebih efektif. Temuan tersebut juga mendukung penelitian [14] yang menyatakan bahwa sistem informasi berbasis *web* dapat meningkatkan transparansi serta kualitas komunikasi antara perusahaan dan pelanggan.

Kontribusi penelitian ini terletak pada pengembangan sistem manajemen proyek *Event Organizer* yang mengintegrasikan pengelolaan operasional internal dengan layanan monitoring proyek bagi client dalam satu platform berbasis SaaS. Integrasi tersebut mendukung konsep integrasi sistem informasi yang dikemukakan oleh [15] yaitu bahwa penyatuan berbagai proses bisnis ke dalam satu sistem dapat meningkatkan konsistensi data, mempercepat proses pengambilan keputusan, dan meningkatkan efisiensi operasional organisasi. Dengan demikian, penelitian ini tidak hanya memberikan solusi praktis bagi perusahaan *Event Organizer*, tetapi juga memberikan kontribusi terhadap pengembangan implementasi SaaS pada bidang manajemen proyek yang masih relatif terbatas.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Implementasi dan pengujian sistem dilakukan pada satu objek penelitian sehingga hasilnya belum sepenuhnya merepresentasikan kebutuhan seluruh perusahaan *Event Organizer* dengan karakteristik yang berbeda. Selain itu, evaluasi sistem masih berfokus pada aspek fungsionalitas dan penerimaan pengguna tanpa melakukan pengujian performa pada skala pengguna yang lebih besar. Oleh karena itu, penelitian selanjutnya dapat mengembangkan fitur *Business Intelligence*, *Artificial Intelligence* (AI) untuk analisis dan prediksi proyek, serta melakukan pengujian pada lebih banyak perusahaan agar diperoleh hasil yang lebih komprehensif dan memiliki tingkat generalisasi yang lebih tinggi.

5. Kesimpulan

Penelitian ini berhasil mengembangkan Sistem Informasi Manajemen Proyek *Event Organizer* berbasis *Software as a Service* (SaaS) menggunakan *framework* Laravel dengan arsitektur *multi-tenant*. Sistem yang dibangun mampu mengintegrasikan pengelolaan proyek, *vendor*, produk, laporan keuangan, serta menyediakan fitur *Client Portal* untuk memantau progres proyek secara *real-time*. Hasil pengujian Black Box menunjukkan seluruh fitur sistem berjalan sesuai dengan kebutuhan pengguna, sedangkan hasil *User Acceptance Test* (UAT)

memperoleh nilai rata-rata 4,4 dari skala 5,0 yang menunjukkan tingkat penerimaan pengguna yang sangat baik.

Berdasarkan hasil penelitian, sistem yang dikembangkan terbukti mampu meningkatkan efisiensi pengelolaan proyek dan mengurangi kesenjangan informasi (*information gap*) antara admin dan client melalui penyajian informasi yang lebih transparan dan terintegrasi. Kontribusi penelitian ini terletak pada penerapan konsep SaaS dan fitur monitoring proyek dalam satu platform terpusat yang dapat digunakan oleh banyak *Event Organizer*. Meskipun demikian, penelitian ini masih terbatas pada satu objek penelitian, sehingga penelitian selanjutnya dapat mengembangkan fitur analitik, integrasi kecerdasan buatan (*Artificial Intelligence*), serta melakukan pengujian pada lingkungan pengguna yang lebih luas.

Daftar Referensi

- [1] R. Priyardana, "Sistem Informasi Event Organizer Berbasis Website Pada Seven Organizer Dengan Metode Waterfall," *Jurnal Ilmu Komputer Dan Informatika* vol. 1, no. 3, pp. 72–78, 2025.
- [2] T. Murianto, "Rancang Bangun Sistem Informasi Event Organizer Berbasis Web Pada PT . Danadipa Nusa Arta Jakarta," *JURNAL LENTERA ICT* vol. 8, no. 2, pp. 247–254, 2023.
- [3] J. M. Inovatif *et al.*, "E-Marketplace Event Organizer Berbasis Progressive Web Application Dengan Metode Agile Scrum," *Jurnal Multidisiplin Inovatif* vol. 9, no. 1, pp. 324–331, 2025.
- [4] R. Setiawan, "Perancangan Elektronik Marketplace Event Organizer," *Jurnal Algoritma* vol. 16, no. 2, pp. 247–254, 2018.
- [5] Ahmadi, and A. Surachman, "Pengembangan Aplikasi Penjualan Dengan Metode Extreme Programming Dan Penerapan Model Multi-Tenancy," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)* vol. 8, no. 3, pp. 303–309, 2024.
- [6] E. B. Agustha, S. Adhy, D. Mutiara, and K. Nugraheni, "Pengembangan Aplikasi Monitoring Informasi Proyek Berbasis Web Menggunakan Metode Iconix Process (Studi Kasus : PT Telkom Akses Witel Semarang)," *Jurnal Masyarakat Informatika* vol. 15, no.1,2024
- [7] Ramdoni and D. S. Bahri, "Implementasi Private Cloud Storage Berbasis Debian Server Menggunakan Nextcloud Sebagai Sentralisasi Penyimpanan Data Pada CV Unelma Pictura," *Biner : Jurnal Ilmu Komputer, Teknik dan Multimedia* vol. 2, no. 4, pp. 576–591, 2024.
- [8] L. M. Harahap, C. Y. Napitu, D. A. Butarbutar, and D. Rachel, "Strategi Manajemen Penyelenggara Event Dalam Menjaga Kepercayaan Stakeholder : Studi Kualitatif Pada Krisis Pestapora 2025 Event Organizer Management Strategy In Maintaining Stakeholder Trust : A Qualitative Study On The Pestapora 2025 Crisis," *JICN: Jurnal Intelpek dan Cendekiawan Nusantara* vol. 2, no. 5, pp. 7766–7774, 2025.
- [9] Halimah, Daniel Arsa, and Benedika Ferdian Hutabarat, "Penerapan Model Incremental Pada Implementasi Dan Pengujian Sistem Informasi Pendanaan Usaha," *Jurnal Sistem Informasi Dan Bisnis Cerdas* vol. 17, no. 2, pp. 11–20, 2024.
- [10] A. Gunawan, M. A. Tihami, E. Masdariah, E. Kurniawati, and R. Suwenti, "Analisis Strategi Manajemen Pendidikan Islam dalam Menghadapi Era Society 5 . 0," *Didaktika: Jurnal Kependidikan* vol. 14, no. 2, pp. 2309–2318, 2025.
- [11] N. V. Amanda and Vicky Brama Kumbara, "Meningkatkan Efisiensi Operasional," *Journal of Science Education and Management Business (JOSEAMB)* vol. 5, no. 1, pp. 412–418, 2026.
- [12] M. Lende, H. H. Ratu, and D. R. Sabawaly, " Perancangan Sistem Pengolahan Data Penduduk di Desa Kadi Pada berbasis web," *Modem: Jurnal Informatika dan Sains Teknologi* vol. 3 no. 4, pp. 91–103, 2025.
- [13] M. Arfan, and Maman Somantri, "Optimalisasi Cloud Software as a Service pada Enterprise Architecture," *Jurnal PROtek* vol. 5, no. 1, pp. 58–61, 2018.
- [14] S. Rahmatullah, A. Rojak, and D. Erica, "Sistem Informasi Smart Project Management (SPM) Berbasis Web Pada PT . Andromeda Data Nusantara Bogor," *Indonesian Journal Computer Science* vol. 4, no. 1, pp. 61–68, 2025.
- [15] M. A. Putri, M. Irwan, and P. Nasution, "Integrasi sistem informasi manajemen dengan teknologi big data dalam bisnis," *Jurnal Ilmiah Ekonomi Dan Manajemen* vol. 3, no. 2, pp. 27–33, 2025.