



Tersedia Online : <http://e-journals.unmul.ac.id/>

ADOPSI TEKNOLOGI DAN SISTEM INFORMASI (ATASI)

Alamat Jurnal : <http://e-journals2.unmul.ac.id/index.php/atasi/index>



Perancangan Aplikasi *Event Organizer* Berbasis Web dengan Implementasi Model *Software as a Service* (SaaS)

Widya Putri Billqis ^{1)*}, Devira Oktaviani ²⁾, Yopi Nugraha ³⁾

^{1,2,3)} Sistem Informasi, Fakultas Ilmu Terapan dan Sains, Institut Pendidikan Indonesia

E-Mail : widyapuqis21@gmail.com ¹⁾; deviraoktavia18@gmail.com ²⁾; yopi@institutpendidikan.ac.id ³⁾;

ARTICLE INFO

Article history:

Received : 02/05/2026

Revised : 01/06/2026

Accepted : 19/06/2026

Available online : 31/08/2026

Keywords:

Event Organizer,
Multi-tenancy,
Prototype,
SaaS,
Progress Transparency.

ABSTRACT

The Event Organizer (EO) industry still faces various operational challenges, particularly project data fragmentation, low transparency of progress updates to clients, and unintegrated work coordination. This study aims to design a web-based EO management application by implementing the Software as a Service (SaaS) model to provide a centralized system that can be used simultaneously by multiple organizations. The development method used is Rapid Application Development (RAD) because it is able to accelerate the design process through iterative and prototyping approaches. The system is designed using a multi-tenant architecture to ensure data isolation between organizations and is equipped with project management, task management, vendor management, financial, reporting, calendar, and client portal modules. The results of testing using the Black Box Testing method indicate that all main features operate according to functional requirements. The implementation of this system is able to improve project management efficiency, facilitate real-time event progress monitoring, and enhance service transparency for clients. This study shows that the implementation of the SaaS model with the RAD approach is effective for developing information systems in the EO industry.

ABSTRAK

Industri *Event Organizer* (EO) masih menghadapi berbagai kendala operasional, terutama fragmentasi data proyek, rendahnya transparansi progres kepada client, serta koordinasi kerja yang belum terintegrasi. Penelitian ini bertujuan merancang aplikasi manajemen EO berbasis web dengan mengimplementasikan model *Software as a Service* (SaaS) guna menyediakan sistem terpusat yang dapat digunakan oleh banyak organisasi secara bersamaan. Metode pengembangan yang digunakan adalah *Rapid Application Development* (RAD) karena mampu mempercepat proses perancangan melalui pendekatan iteratif dan prototyping. Sistem dirancang menggunakan arsitektur *multi-tenant* untuk menjamin isolasi data antar organisasi, serta dilengkapi modul *project management*, *task management*, vendor, keuangan, *reporting*, kalender, dan portal client. Hasil pengujian menggunakan metode *Black Box Testing* menunjukkan seluruh fitur utama berjalan sesuai kebutuhan fungsional. Implementasi sistem ini mampu meningkatkan efisiensi pengelolaan proyek, mempermudah monitoring progres acara secara *real-time*, serta meningkatkan transparansi layanan kepada client. Penelitian ini menunjukkan bahwa penerapan model SaaS dengan pendekatan RAD efektif digunakan dalam pengembangan sistem informasi pada industri EO.

2026 Adopsi Teknologi dan Sistem Informasi (ATASI) with CC BY SA license.

1. PENDAHULUAN

Perusahaan *Event Organizer* (EO) memiliki peran yang sangat penting dalam penyelenggaraan berbagai jenis acara (Dewidianto et al., 2024). Namun demikian, pengelolaan acara yang bersifat kompleks kerap menghadapi berbagai hambatan operasional, khususnya dalam aspek pencatatan, pemantauan, serta koordinasi antara tim internal, klien, dan vendor. Pada praktiknya, sebagian besar EO masih bergantung pada sistem komunikasi manual, penggunaan *spreadsheet*, maupun perangkat terpisah yang belum terintegrasi secara optimal, sehingga berpotensi

*) Corresponding Author

<https://doi.org/10.30872/atasi.v5i2.4610>

2026 Adopsi Teknologi dan Sistem Informasi (ATASI) with CC BY SA license.

menimbulkan miskomunikasi, keterlambatan, serta ketidaktepatan data (Rizky et al., 2021). Meskipun beberapa upaya digitalisasi telah dilakukan pada penelitian terdahulu seperti oleh (Premitasari et al., 2023) yang berfokus pada pembuatan aplikasi perancangan EO berbasis website yang dikhususkan untuk mengelola data pelanggan dan sistem *ticketing* pada satu mitra manajemen acara dan penelitian oleh (Alramdani & Saputri, 2025) yang mengembangkan sistem informasi manajemen internal menggunakan metode *Agile* untuk mengotomatisasi konversi data order dari *spreadsheet* menjadi dokumen proposal penawaran pada sebuah perusahaan, sistem yang dikembangkan umumnya masih bersifat konvensional (*single-tenant*) yang terisolasi untuk satu mitra bisnis saja. Oleh karena itu, diperlukan pengembangan aplikasi web manajemen proyek yang terintegrasi dan multi-pengguna untuk mendukung efektivitas operasional banyak EO secara simultan.

Sebagai solusi strategis, perancangan ini difokuskan pada pengembangan aplikasi yang mengimplementasikan model *Software as a Service* (SaaS). Pemilihan model SaaS ini didasarkan pada pertimbangan bahwa platform dapat beroperasi secara terukur (*scalable*) dan multi-pengguna (Kumar, 2020) melalui mekanisme berlangganan, yang sekaligus menjadi keterbaruan strategis (inovasi) dalam desain sistem informasi ini. Inovasi tersebut mengharuskan adanya perancangan arsitektur yang kuat serta fungsionalitas yang komprehensif, mencakup modul manajemen proyek, manajemen klien, manajemen vendor, hingga *cash flow*.

Menyikapi urgensi kebutuhan yang telah diuraikan, penelitian ini bertujuan untuk melakukan analisis menyeluruh terhadap kebutuhan fungsional dan non-fungsional dari aplikasi *Event Organizer* (EO) berbasis *Software as a Service* (SaaS). Selain itu, penelitian ini juga berfokus pada perancangan arsitektur sistem yang kokoh, meliputi desain basis data *multi-tenant*, pengembangan model fungsional menggunakan *Unified Modeling Language* (UML), serta perancangan prototipe antarmuka pengguna (UI/UX). Metodologi yang digunakan untuk memandu proses perancangan ini adalah *Rapid Application Development* (RAD).

2. TINJAUAN PUSAKA

A. *Software as a Service* (SaaS)

Software as a Service (SaaS) merupakan model distribusi perangkat lunak di mana aplikasi disediakan dan dikelola oleh penyedia layanan, kemudian diakses pengguna melalui internet dengan sistem berlangganan (Krishna et al., 2021). Model ini memungkinkan pengguna memanfaatkan perangkat lunak tanpa harus melakukan instalasi dan pemeliharaan infrastruktur secara mandiri. Salah satu karakteristik utama SaaS adalah *multi-tenancy*, yaitu satu aplikasi dapat melayani banyak pengguna atau organisasi secara bersamaan dengan data yang tetap terisolasi. Pendekatan ini dinilai efisien karena mampu menekan biaya operasional dan meningkatkan skalabilitas sistem (Liliana et al., 2024).

B. *Event Organizer* (EO)

Event Organizer (EO) merupakan pihak profesional yang bertanggung jawab dalam merencanakan, mengatur, mengoordinasikan, dan melaksanakan suatu acara sesuai kebutuhan klien. Dalam operasionalnya, EO membutuhkan pengelolaan data pelanggan, koordinasi vendor, penyusunan jadwal kegiatan, serta penyampaian laporan Administrasi yang terintegrasi. Penggunaan sistem informasi terpusat dapat membantu meningkatkan efisiensi kerja, meminimalisir kesalahan komunikasi, serta mempercepat proses pelayanan kepada pelanggan (Afifah et al., 2022). Selain itu, keberhasilan penyelenggaraan acara oleh *Event Organizer* sangat dipengaruhi oleh kemampuan manajemen, koordinasi tim, serta pengelolaan sumber daya secara efektif (Muarif et al., 2024).

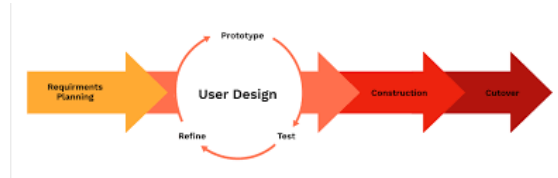
C. *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah metode pengembangan perangkat lunak yang menitikberatkan pada proses pembangunan sistem secara cepat melalui pendekatan *prototyping* dan iterasi berulang. Metode ini melibatkan pengguna secara aktif pada tahap perancangan sehingga kebutuhan sistem dapat divalidasi sejak awal. RAD sangat sesuai digunakan pada pengembangan aplikasi berbasis web karena mampu mempercepat waktu implementasi serta menghasilkan sistem yang sesuai kebutuhan pengguna (Irnawati, 2021). Selain itu, penerapan RAD juga dinilai efektif dalam meningkatkan fleksibilitas pengembangan sistem dan mempermudah proses penyempurnaan aplikasi secara bertahap (Kosidin et al., 2020).

3. METODE PENELITIAN

Penelitian ini diimplementasikan dengan menggunakan pendekatan metodologi *Rapid Application Development* (RAD). Pemilihan metode RAD ini didasarkan pada keunggulan fundamentalnya, yaitu kemampuan untuk mempercepat proses pengembangan sistem melalui siklus iteratif dan partisipatif (Nurman Hidayat & Kusuma Hati, 2021).

RAD secara spesifik sangat ideal karena menekankan pada teknik *prototyping* serta keterlibatan aktif pengguna dalam setiap tahapan (Santoso & Amanullah, 2022). Pendekatan ini dianggap paling tepat untuk menghasilkan prototipe UI/UX, karena keunggulannya dalam mendukung proses validasi dan evaluasi rancangan secara cepat dan responsif melalui siklus iteratif. Dengan demikian, pendekatan ini memastikan keselarasan optimal antara kebutuhan industri *Event Organizer* (EO) secara umum dengan hasil prototipe yang dikembangkan, sekaligus menjamin bahwa aspek arsitektur *multi-tenant* telah dipertimbangkan sejak tahap awal perancangan sistem.



Gambar 1. Tahapan RAD (Iqbal, 2023)

Penelitian perancangan ini menerapkan empat fase utama dalam metodologi *Rapid Application Development* (RAD) guna memastikan bahwa prototipe yang dihasilkan bersifat valid dan relevan (Sitio, 2023). Fase pertama, *Requirement Planning*, melibatkan studi literatur dan analisis pasar untuk merumuskan kebutuhan fungsional dan non-fungsional aplikasi, dengan fokus pada pengembangan arsitektur *multi-tenant* serta modul-modul manajemen *Event Organizer* (EO). Fase kedua, *User Design*, menitikberatkan pada perancangan arsitektur sistem, penyusunan skema basis data *multi-tenant* menggunakan Entity Relationship Diagram (ERD), dan pemodelan fungsionalitas melalui Unified Modeling Language (UML). Fase ketiga, *Construction*, mencakup proses pembangunan prototipe antarmuka pengguna (UI/UX *prototyping*) dengan memanfaatkan perangkat desain yang relevan. Terakhir, fase *Cutover* dilakukan untuk memvalidasi hasil perancangan melalui pengujian prototipe, sehingga dapat dipastikan kesesuaiannya dengan kebutuhan yang telah ditetapkan pada tahap awal (Bahari & Pramudwiatmoko, 2025).

Karakteristik utama model pengembangan berbasis Software as a Service (SaaS) yang menjadi sorotan dalam penelitian ini diintegrasikan secara mendalam pada setiap proses perancangan melalui konfigurasi arsitektur *multi-tenancy*. Implementasi ini diwujudkan dengan merancang satu infrastruktur

Karakteristik utama model pengembangan berbasis *Software as a Service* (SaaS) yang menjadi sorotan dalam penelitian ini diintegrasikan secara mendalam pada setiap proses perancangan melalui konfigurasi arsitektur *multi-tenancy*. Implementasi ini diwujudkan dengan merancang satu infrastruktur aplikasi dan satu basis data terpusat yang dapat diakses oleh banyak institusi EO (*tenant*) yang berbeda secara simultan, di mana pemisahan data antar-organisasi dikunci secara logis (*logical separation isolation*) melalui kolom relasi khusus pada level basis data. Melalui pendekatan layanan *on-demand* ini, pengguna dari pihak institusi EO dapat memanfaatkan fungsionalitas manajemen operasional langsung melalui *web browser* tanpa melakukan instalasi lokal. Selain itu, sistem juga memfasilitasi sub-ruang kerja terintegrasi yang memungkinkan klien dari masing-masing *tenant* EO untuk memantau status persiapan acara secara *real-time*. Seluruh spesifikasi tersebut didasarkan pada teknik pengumpulan data yang dilakukan melalui pendekatan triangulasi untuk mendapatkan perspektif industri yang komprehensif (Nurfajriani et al., 2024), meliputi: studi literatur terkait SaaS dan EO, *benchmarking* platform kompetitor, serta wawancara terstruktur dengan praktisi EO anonim. Hasilnya menjadi fondasi dalam menentukan spesifikasi sistem pada tahap *Requirement Planning*.

4. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan Sistem

Berdasarkan fase *Requirement Planning* yang telah dilakukan, ditemukan bahwa permasalahan utama pada industri *Event Organizer* (EO) adalah fragmentasi data operasional, rendahnya transparansi progres kepada *Client*, serta koordinasi kerja yang masih bergantung pada banyak media terpisah. Penggunaan *spreadsheet*, aplikasi pesan instan, dan pencatatan manual menyebabkan informasi proyek sulit dipantau secara menyeluruh dan berpotensi menimbulkan keterlambatan pekerjaan.

Sebagai solusi, sistem dirancang dalam bentuk aplikasi berbasis web dengan model SaaS agar seluruh data proyek, pengguna, vendor, layanan, dan keuangan dapat dikelola dalam satu platform terintegrasi. Pendekatan ini dinilai mampu meningkatkan efisiensi operasional karena proses kerja tidak lagi tersebar di berbagai media yang berbeda. Selain itu, penerapan arsitektur *multi-tenant* memungkinkan banyak organisasi EO menggunakan platform yang sama dengan data yang tetap terisolasi antar tenant. Model ini memberikan keunggulan dari sisi efisiensi biaya implementasi, pemeliharaan sistem, dan skalabilitas layanan.

a. Analisis Kebutuhan Fungsional

Kebutuhan fungsional dirancang untuk menjawab proses bisnis utama pada operasional EO, mulai dari pengelolaan tenant, proyek, vendor, hingga transparansi layanan kepada *Client*. Setiap modul dikembangkan agar aktivitas kerja yang sebelumnya dilakukan secara manual dapat dilakukan secara terpusat dan terdokumentasi.

Tabel 1. Kebutuhan Fungsional Sistem Berdasarkan Peran Pengguna

Aktor	Modul Kebutuhan	Deskripsi Fitur
Super Admin	Manajemen Tenant	Mengelola siklus hidup akun organisasi EO, mulai dari pendaftaran, aktivasi, hingga penghentian akses.
	Manajemen Paket	Menentukan limitasi fitur (kapasitas data, jumlah user) berdasarkan tingkatan paket SaaS yang dipilih.
	Monitoring & Backup	Memantau aktivitas sistem secara global dan melakukan pencadangan data rutin untuk ketersediaan layanan.
Admin EO	Project & Task Management	Merancang alur kerja, menetapkan tugas (<i>task assignment</i>), dan memantau penyelesaian <i>timeline</i> proyek secara mendetail.
	Financial & Reporting	Melakukan pencatatan arus kas, mengelola invoice, dan menghasilkan laporan (<i>Report</i>) finansial proyek secara otomatis.
	Vendor & Product Catalog	Mengelola basis data pihak ketiga serta katalog produk/layanan (<i>Services</i>) yang ditawarkan organisasi.

Aktor	Modul Kebutuhan	Deskripsi Fitur
Client	Visual Scheduling	Memantau jadwal seluruh proyek secara visual melalui modul Kalender (<i>Calendar</i>) untuk optimasi sumber daya.
	Portal Progress	Mengakses halaman khusus untuk melihat sejauh mana perkembangan acara secara <i>real-time</i> melalui visualisasi progres.
	Pusat Unduhan	Melakukan pengunduhan mandiri terhadap dokumen kontrak, draf proposal, serta <i>invoice</i> tagihan.
	Sistem Feedback	Memberikan persetujuan atau tanggapan langsung terkait konsep acara yang diajukan oleh tim EO.

Integrasi modul *Project Management*, *Task Management*, dan *Financial Reporting* dalam model SaaS ini memotong rantai asimetri informasi yang sering terjadi pada koordinasi manual. Penyatuan ini memastikan bahwa setiap perubahan status tugas operasional di lapangan secara otomatis memperbarui arus kas proyek secara *real-time*. Dampaknya, risiko defisit anggaran saat acara berlangsung dapat diminimalisasi tanpa memerlukan rekonsiliasi data manual yang memakan waktu."

b. Analisis Kebutuhan Non-Fungsional

Selain fitur utama, sistem juga dirancang memenuhi kebutuhan non-fungsional agar mampu beroperasi secara optimal pada model SaaS.

Tabel 2. Kebutuhan Non-Fungsional Sistem

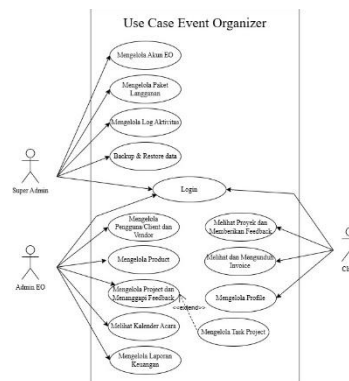
Aspek	Deskripsi Kebutuhan
Keamanan	Sistem wajib mengimplementasikan isolasi data tingkat basis data (<i>multi-tenant architecture</i>) menggunakan atribut <i>tenant_id</i> agar data antar organisasi EO tidak saling bercampur.
Ketersediaan	Aplikasi harus berbasis <i>web (cloud-based)</i> agar dapat diakses oleh EO dan klien dari berbagai perangkat kapan saja secara <i>real-time</i> .
Skalabilitas	Infrastruktur sistem harus mampu menangani penambahan jumlah pengguna (<i>tenant</i>) baru serta volume data proyek yang besar tanpa menurunkan performa aplikasi secara signifikan.
Autentikasi & Otorisasi	Setiap pengguna wajib melalui proses <i>Login</i> yang aman dengan pembagian hak akses berdasarkan peran (<i>Role-based access control</i>), seperti <i>Super Admin</i> , <i>Admin EO</i> , atau <i>Client</i> .
Integritas Data	Menjamin keakuratan data yang disajikan pada modul <i>Report</i> dan <i>Calendar</i> sehingga informasi yang diterima pengguna konsisten dengan data operasional di basis data.

Aspek keamanan menjadi perhatian utama melalui penerapan isolasi data tenant menggunakan atribut *tenant_id*. Selain itu, kebutuhan skalabilitas dan ketersediaan sistem penting agar platform mampu melayani banyak organisasi EO secara bersamaan tanpa menurunkan performa layanan.

B. Pemodelan Sistem dengan UML

Perancangan fungsional sistem digambarkan melalui Unified Modeling Language (UML) untuk memberikan gambaran logis interaksi pengguna (Azis, 2022).

a. Use Case Diagram.

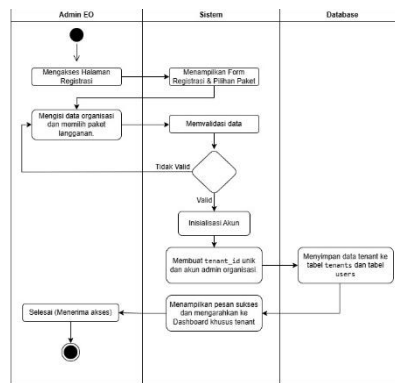


Gambar 2. Usecase Event Organizer

Use Case Diagram menunjukkan tiga aktor utama, yaitu *Super Admin*, *Admin EO*, dan *Client*. Pembagian aktor dilakukan untuk memastikan setiap pengguna memperoleh hak akses sesuai kewenangannya. *Super Admin* berfokus pada pengelolaan tenant dan pemeliharaan sistem. *Admin EO* menangani operasional internal seperti proyek, vendor, dan keuangan. Sementara *Client* diberikan akses terbatas pada progres proyek, dokumen, dan *feedback*. Struktur ini penting untuk meningkatkan transparansi tanpa mengorbankan keamanan data internal organisasi.

b. Activity Diagram yang digunakan untuk menggambarkan alur kerja sistem secara lebih mendetail. Diagram ini berfokus pada proses bisnis utama yang mencerminkan karakteristik model SaaS dan operasional *Event Organizer*, yaitu prosedur pendaftaran tenant baru, manajemen proyek, serta interaksi klien pada portal progress.

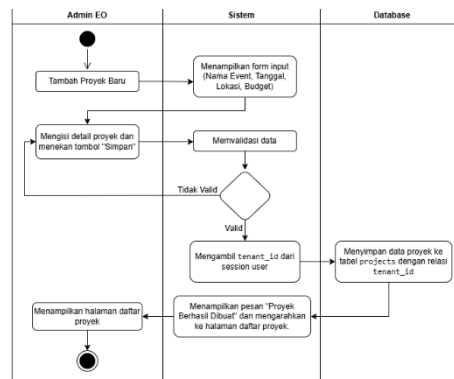
1) Activity Diagram Pendaftaran Tenant Baru, proses ini merupakan alur awal bagi organisasi EO untuk mendapatkan akses ke dalam sistem sebagai penyewa layanan (*tenant*).



Gambar 3. Activity Diagram Pendaftaran Tenant Baru

Mekanisme penggunaan *tenant_id* unik secara otomatis saat registrasi merupakan hulu dari pengondisian keamanan platform SaaS ini. Struktur tersebut mengunci seluruh proses kueri data pada sistem sejak awal. Hasilnya, meskipun terdapat banyak organisasi EO yang mengakses tabel basis data yang sama secara bersamaan, batas-batas privasi data antar-*tenant* tetap terjaga secara ketat pada level *back-end*.

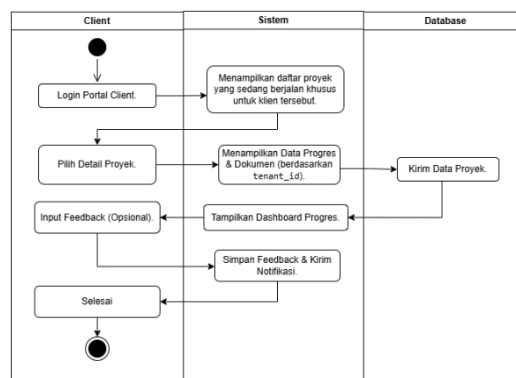
- 2) Activity Diagram Manajemen Proyek, proses ini menggambarkan bagaimana *Admin* EO mengelola data operasional acara yang sedang ditangani.



Gambar 4. Activity Diagram Manajemen Proyek

Diagram ini menjelaskan alur pengelolaan proyek mulai dari input data acara, penjadwalan, pembagian tugas, hingga monitoring progres. Fitur ini menjawab masalah koordinasi manual yang umum terjadi pada EO, karena seluruh aktivitas proyek dicatat dalam satu sistem dan dapat dipantau secara langsung.

- 3) Activity Diagram Portal Progress *Client*, Diagram ini menggambarkan alur transparansi bagi klien dalam memantau perkembangan acara secara mandiri.



Gambar 5. Activity Diagram Portal Progress *Client*

Portal klien dirancang agar pelanggan dapat memantau progres acara secara mandiri, mengakses dokumen penting, dan memberikan tanggapan. Keberadaan fitur ini menjadi pembeda utama dibanding sistem EO konvensional yang umumnya masih bergantung pada komunikasi manual melalui pesan pribadi.

c. Entity Relationship Diagram (ERD)



Gambar 6. Entity Relationship Diagram

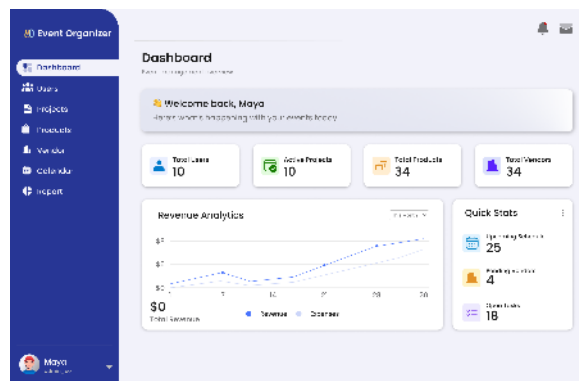
Struktur basis data menggunakan pendekatan *shared schema multi-tenant*, di mana seluruh data disimpan dalam satu basis data dengan atribut *tenant id* sebagai pembeda. Model ini meningkatkan efisiensi infrastruktur tanpa mengurangi keamanan data antar organisasi pengguna.

C. Perancangan Antarmuka Pengguna (UI/UX)

Tahap perancangan antarmuka merupakan implementasi visual dari kebutuhan fungsional dan pemodelan sistem yang telah dirancang sebelumnya. Antarmuka ini dibangun dengan prinsip *user-centered design* untuk memastikan kemudahan operasional bagi *Admin EO* maupun transparansi informasi bagi klien. Berikut adalah beberapa tampilan utama dari sistem yang dikembangkan:

a. Dashboard Utama Admin EO

Dashboard utama dirancang sebagai pusat kendali operasional agar *Admin EO* dapat memantau kondisi organisasi secara cepat melalui satu halaman terintegrasi. Penyajian indikator seperti jumlah proyek aktif, vendor, dan pendapatan membantu pengambilan keputusan lebih efisien dibanding metode manual yang mengharuskan pengecekan data pada beberapa file terpisah.



Gambar 7. Dashboard Admin EO

b. Manajemen Pengguna (User List)

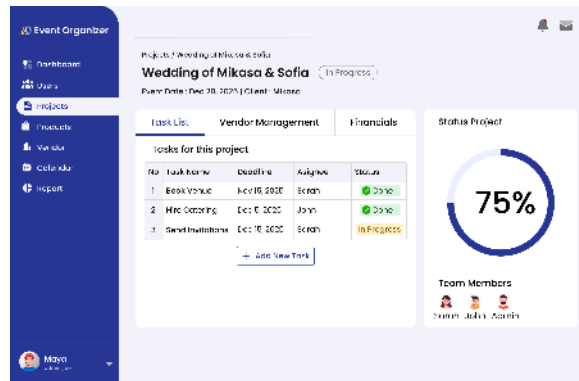
Fitur manajemen pengguna penting untuk mengatur akses staf internal maupun *Client* sesuai kewenangannya. Dengan penerapan *role-based access control*, sistem mampu menjaga keamanan data serta mengurangi risiko penyalahgunaan akses yang sering terjadi pada pengelolaan akun secara manual.

ID	Username	Email	Role	Status	Action
1001	admin	admin@eventorganizer.com	Admin	Active	Edit / Delete
1002	user1	user1@eventorganizer.com	User	Active	Edit / Delete
1003	user2	user2@eventorganizer.com	User	Active	Edit / Delete
1004	user3	user3@eventorganizer.com	User	Active	Edit / Delete
1005	user4	user4@eventorganizer.com	User	Active	Edit / Delete
1006	user5	user5@eventorganizer.com	User	Active	Edit / Delete
1007	user6	user6@eventorganizer.com	User	Active	Edit / Delete
1008	user7	user7@eventorganizer.com	User	Active	Edit / Delete
1009	user8	user8@eventorganizer.com	User	Active	Edit / Delete
1010	user9	user9@eventorganizer.com	User	Active	Edit / Delete

Gambar 8. Manajemen Pengguna

c. Detail Proyek dan Manajemen Tugas (Project Management Detail)

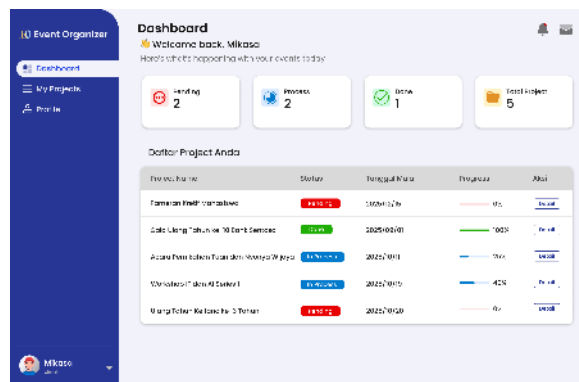
Sebagai inti dari fungsionalitas sistem, halaman detail proyek memungkinkan *Admin* untuk mengelola *Task List*, *Vendor Management*, dan *Financials* dalam satu tampilan terpadu. Fitur ini memudahkan pembagian tugas (*task assignment*) kepada tim dan memberikan visualisasi progres proyek melalui grafik lingkaran (*Progress Circle*) yang dihitung secara otomatis berdasarkan status penyelesaian tugas.



Gambar 9. Proyek dan Manajemen Tugas

d. Portal Dashboard Klien (*Client Monitoring Portal*)

Penyajian *Client Monitoring Portal* mengubah pola komunikasi industri EO yang semula bersifat reaktif di mana klien harus terus-menerus menghubungi pihak EO via aplikasi pesan instan menjadi lebih transparan dan mandiri. Dengan menyajikan status *Pending*, *In Process*, atau *Done* yang terhubung langsung dengan pembaruan data tim lapangan, interaksi visual ini berhasil memangkas frekuensi rapat koordinasi formal sekaligus meningkatkan indeks kepercayaan klien secara signifikan.



Gambar 10. *Dashboard Client*.

D. Pengujian Sistem

Metode pengujian yang digunakan adalah *Black box testing*, yaitu teknik pengujian perangkat lunak yang berfokus pada fungsionalitas sistem. Pengujian ini bertujuan untuk memverifikasi input dan output pada fitur-fitur utama seperti manajemen proyek, isolasi data *multi-tenant*, hingga portal transparansi bagi klien. Hasil pengujian terhadap skenario utama disajikan dalam Tabel 3.

Tabel 3. Pengujian Sistem

ID	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
TP1	<i>Login & Auth</i>	Input kredensial valid (email & password).	Akses diberikan sesuai <i>role</i> user.	Berhasil
TP2	<i>User Management</i>	Registrasi akun <i>Client</i> baru oleh <i>Admin</i> .	Data tersimpan dan tampil di <i>User List</i> .	Berhasil
TP3	<i>Project Creation</i>	Pembuatan proyek (input tanggal & budget).	Proyek tampil di <i>Dashboard</i> & daftar aktif.	Berhasil
TP4	<i>Task Management</i>	Perubahan status tugas ke <i>Done</i> .	Persentase <i>Progress Circle</i> meningkat.	Berhasil
TP5	<i>Product Catalog</i>	Penambahan jasa layanan baru.	Jasa muncul di katalog dan opsi proyek.	Berhasil
TP6	<i>Client Portal</i>	Akses portal oleh akun <i>Client</i> .	Hanya menampilkan data proyek milik klien.	Berhasil
TP7	<i>Reporting</i>	Akses menu grafik pendapatan.	Visualisasi <i>Revenue Analytics</i> sesuai data.	Berhasil
TP8	<i>Data Isolation</i>	Akses data antar <i>tenant</i> (EO A ke EO B).	Sistem menolak akses (keamanan <i>tenant</i>).	Berhasil

ID	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
TP9	<i>Calendar Sync</i>	Penambahan proyek pada tanggal tertentu.	Label proyek otomatis tampil di kalender.	Berhasil
TP10	<i>Quick Stats</i>	Penambahan 5 tugas berstatus <i>Open</i> .	Angka kartu <i>Open Tasks</i> terupdate otomatis.	Berhasil
TP11	<i>Vendor Management</i>	Input data kontak dan kategori vendor.	Data tersimpan di tabel <i>Vendor List</i> .	Berhasil
TP12	<i>Progress Calculation</i>	Penyelesaian 3 dari 4 tugas proyek.	Visualisasi progres akurat di angka 75%.	Berhasil
TP13	<i>Responsiveness</i>	Akses sistem via mobile/tablet.	<i>Layout</i> adaptif dan informasi tidak terpotong.	Berhasil
TP14	<i>Revenue Graph</i>	Input data pembayaran masuk.	Grafik garis pendapatan terupdate otomatis.	Berhasil
TP15	<i>Profile Management</i>	Perubahan foto dan kontak profil.	Data terupdate di database dan header aplikasi.	Berhasil

Keberhasilan 100% pada 15 skenario pengujian *Black Box* ini memberikan dua kesimpulan penting terkait keandalan sistem. Pertama, validitas pada skenario TP6 dan TP8 (*Data Isolation*) membuktikan bahwa logika penguncian kueri berbasis *tenant_id* bekerja dengan tepat dalam menolak intrusi data antar-organisasi EO. Kedua, korelasi sukses antara TP4 (*Task Management*) dan TP12 (*Progress Calculation*) mengonfirmasi bahwa algoritma kalkulasi pada sistem mampu menyajikan representasi data yang akurat antara kinerja riil di lapangan dengan visualisasi pada antarmuka pengguna.

E. Pembahasan

Penelitian ini diselesaikan menggunakan metode *Rapid Application Development* (RAD) melalui empat tahapan utama dengan total waktu pengerjaan selama 63 hari (9 minggu). Tahapan dimulai dari fase *Requirement Planning* selama 2 minggu, fase *User Design* selama 4 minggu, fase *Construction* selama 2 minggu, hingga fase *Cutover* selama 1 minggu. Dari seluruh rangkaian tersebut, fase *User Design* menjadi tahapan yang memakan waktu paling lama karena melalui 2 kali siklus iterasi balik (*feedback loop*) bersama praktisi EO. Iterasi pertama menghasilkan revisi berupa penyederhanaan relasi tabel basis data *multi-tenant* dan penyesuaian hak akses. Sementara itu, jika terjadi siklus balik pada iterasi kedua, hal tersebut memicu perbaikan desain antarmuka akibat adanya permintaan pengguna untuk mengoptimalkan visualisasi grafik pelacakan progres (*real-time progress tracking*) agar lebih mudah dipahami oleh klien pemilik acara. Setelah prototipe fungsional disepakati, proses dilanjutkan ke fase *Construction* untuk perwujudan komponen desain, dan diakhiri dengan fase *Cutover* untuk validasi akhir hasil rancangan.

Hasil perancangan arsitektur *Software as a Service* (SaaS) *multi-tenant* pada penelitian ini dibuktikan melalui penguncian pemisahan data antar-organisasi secara logis (*logical separation isolation*) menggunakan parameter ID penyewa (*tenant*) dalam satu basis data terpusat. Pendekatan ini memungkinkan banyak institusi EO beroperasi secara mandiri dan aman dalam satu infrastruktur platform terpadu tanpa risiko kebocoran data (*data leakage*), sekaligus memberikan transparansi bagi klien untuk memantau grafik persiapan acara secara *real-time*.

Karakteristik arsitektur ini memberikan diferensiasi yang nyata jika disandingkan dengan fokus pemodelan pada beberapa penelitian terdahulu di bidang sistem informasi EO. Penelitian oleh (Premitasari et al., 2023) membuat aplikasi perencanaan EO berbasis *website* konvensional yang bersifat *single-tenant* khusus untuk mengelola data pelanggan dan sistem *ticketing* pada satu mitra tertentu. Di sisi lain, sistem manajemen EO yang dirancang oleh (Alramdani & Saputri, 2025) juga bertumpu pada aplikasi web *single-tenant* yang berfokus pada digitalisasi alur kerja internal staf untuk mengotomatisasi pembuatan proposal penawaran. Kedua penelitian terdahulu tersebut mengadopsi model arsitektur *single-tenant*, di mana karakteristik platformnya dirancang optimal untuk melayani satu entitas organisasi secara terdedikasi.

Berbeda dengan fokus batasan ruang lingkup tersebut, perancangan SaaS dalam penelitian ini diarahkan untuk menjawab tantangan skalabilitas platform berskala luas. Melalui pemanfaatan arsitektur *multi-tenant*, sistem yang diusulkan mampu mengeliminasi kebutuhan replikasi server maupun instalasi ulang basis data untuk setiap organisasi baru. Pendekatan ini menawarkan model solusi yang adaptif dan efisien dari segi biaya infrastruktur IT bagi industri jasa acara skala UMKM, sekaligus memotong hambatan investasi teknologi bagi pelaku bisnis EO semula melalui pemanfaatan ekosistem berbagi pakai (*shared infrastructure*). Dari aspek pengembangan ilmu, perancangan ini mempertegas metodologi integrasi arsitektur *multi-tenancy* pada level perancangan basis data logis, yang membuktikan bahwa kompleksitas sistem latar belakang (*back-end*) dapat dikemas dalam antarmuka pengguna yang sederhana dan transparan.

5. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dipaparkan, dapat disimpulkan bahwa **perancangan** prototipe sistem manajemen *Event Organizer* (EO) berbasis *Software as a Service* (SaaS) ini telah berhasil menjawab permasalahan utama terkait fragmentasi data operasional dan kurangnya transparansi progres bagi klien. Penerapan arsitektur *multi-tenant* terbukti mampu memberikan isolasi data yang aman bagi setiap organisasi EO, sementara integrasi fitur *Task Management*, *Calendar*, dan *Reporting* dalam satu dasbor terpusat berhasil menyederhanakan kompleksitas manajemen proyek. Kesesuaian antara kebutuhan fungsional, perancangan UML, hingga implementasi antarmuka pengguna memastikan bahwa sistem ini mampu meningkatkan akuntabilitas melalui visualisasi progres pengerjaan acara secara *real-time*. Luaran perancangan ini memberikan nilai praktis

dalam menekan biaya investasi infrastruktur IT bagi pelaku bisnis EO skala UMKM melalui konsep berbagi pakai (*shared infrastructure*), sekaligus memberikan kontribusi keilmuan mengenai metodologi pengondisian keamanan data logis berskala jamak yang tetap mempertahankan kesederhanaan interaksi antarmuka pengguna.

6. DAFTAR PUSTAKA

- Afifah, Z., Fedrina, R., Darmawan, R., & Abidin, J. (2022). Client Perception of Groovy Event Organizer in Handling Event. *Proceeding The 2nd ICHELSS*, 943–951.
- Alramdani, N. H. A., & Saputri, D. U. E. (2025). *Perancangan Aplikasi Event Organizer Pada*. 9(3), 4756–4763.
- Azis, N. (2022). *Analisis Perancangan Sistem Informasi* (N. S. Wahyuni (ed.)). Widina Bhakti Persada Bandung.
- Bahari, A. F., & Pramudwiatmoko, A. (2025). Implementation of Rapid Application Development (RAD) Method for Mobile-Based Ice Cream Ordering Application. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 5(January), 283–291.
- Dewidianto, R., Manalullaili, M., & Ayu Ningsih, C. P. (2024). Peran Komunikasi Organisasi dalam Event Organizer (Studi pada Event Organizer Soundtrack Indonesia Kota Palembang). *Jurnal Bisnis dan Komunikasi Digital*, 1(4), 10. <https://doi.org/10.47134/jbkd.v1i4.2771>
- Iqbal, D. M. (2023). *Perancangan Sistem Informasi Manajemen Administrasi Badan Usaha Milik Desa Berbasis Web (Studi Kasus: Bumdes Citeko Kaler Purwakarta)*.
- Irnawati, O. (2021). Sistem Informasi Pengerjaan Proyek Berbasis Web Menggunakan Metode Rapid Application Development. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, 3(02), 119–129. <https://doi.org/10.46772/intech.v3i02.501>
- Kosidin, Firdaus, E., & Rusmana, A. (2020). Penerapan Metode Rapid Application Development (RAD) Dalam Pengembangan Aplikasi Simpan Pinjam Koperasi Syariah. *Jurnal Informatika dan Komputer (INFOKOM)*, 8(No. 1), 24–36.
- Krishna, A., Rath, D. S., & Mishra, A. (2021). Systematic Literature Review of Software as a Service [SaaS] in view of Security and Multitenancy. *International Journal for Research in Applied Science and Engineering Technology*, 9(9), 1742–1751. <https://doi.org/10.22214/ijraset.2021.38205>
- Kumar, R. (2020). Multi-Tenant SaaS Architectures : Design Principles and Security Considerations. *Journal of Architecture and Civil Engineering*, 5(2), 49–62. <https://doi.org/10.35629/8193-05024962>
- Liliana, Soesanto, D., Prijambodo, B., & Ohanna, J. (2024). Exploration of Software as a Service (SaaS) as a Project Management Tools. *Teknika*, 13(2), 309–317. <https://doi.org/10.34148/teknika.v13i2.933>
- Muarif, A., Syarifudin, A., Muhammad Randicha, & Hamandia. (2024). Analisis Faktor Keberhasilan Event Organizer Dalam Menyelenggarakan Acara(Studi Kasus Cv Soundtrack Indonesia). *Journal of Business and Halal Industry*, 1(4), 8. <https://doi.org/10.47134/jbhi.v1i4.414>
- Nurfajriani, W. V., Wahyu, M., Arivan, I., Sirodj, R. A., & Afgani, M. W. (2024). Triangulasi Data Dalam Analisis Data Kualitatif Wiyanda. *Jurnal Ilmiah Wahana Pendidikan*, 10(September), 826–833.
- Nurman Hidayat, & Kusuma Hati. (2021). Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE). *Jurnal Sistem Informasi*, 10(1), 8–17. <https://doi.org/10.51998/jsi.v10i1.352>
- Premitasari, M., Riandy, F. D., Ahmad, F., & Gumelar, R. (2023). Aplikasi Perencanaan Event Organizer berbasis Website Untuk Meningkatkan Proses Pengelolaan Pelanggan. *REKA KARYA: Jurnal Pengabdian Kepada Masyarakat*, 2(2), 167–174.
- Rizky, M. F., Cleopatra, M., & Parwatiningsy, D. (2021). Sistem Informasi Event Organizer Pada Wahana Entertainment Berbasis Java. *Journal of Information System, Applied, Management, Accounting and Research*, 5(1), 117. <https://doi.org/10.52362/jisamar.v5i1.346>
- Santoso, L., & Amanullah, J. (2022). Pengembangan Sistem Informasi Akademik Berbasis Website Menggunakan Metode Rapid Application Development (Rad). *Elkom: Jurnal Elektronika dan Komputer*, 15(2), 250–259. <https://doi.org/10.51998/jsi.v10i1.352>
- Sitio, S. L. M. (2023). *Penerapan Metode Rapid Application Development (Rad) Untuk Aplikasi E Learning Berbasis Web*. Eureka Media Aksara.