

APLIKASI MANEJEMEN TRANSAKSI PADA *EVENT ORGANIZER* MENGGUNAKAN *EXTREME PROGRAMMING*

Davina Valerian Cong, Johannes Fernandes Andry
Sistem Informasi, Universitas Bunda Mulia
Jl. Lodan Raya No.2, Ancol, Jakarta Utara, Indonesia
Davinavc18@gmail.com

ABSTRAK

Dalam dunia bisnis modern, sistem informasi memainkan peran penting dalam meningkatkan efisiensi dan efektivitas operasional. *Event Organizer* (EO) merupakan salah satu industri yang mengalami peningkatan signifikan pasca pandemi COVID-19. Namun, banyak perusahaan EO masih mengandalkan metode manual dalam pengelolaan transaksi dan informasi, yang berpotensi menyebabkan kesalahan *input* dan memperlambat proses bisnis. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan aplikasi manajemen transaksi berbasis web menggunakan metode *Extreme programming* (XP). Metode XP dipilih karena memiliki fleksibilitas tinggi dalam menghadapi perubahan kebutuhan sistem. Proses pengembangan dilakukan melalui tahapan perencanaan, desain, pengkodean, dan pengujian dengan metode *blackbox* untuk memastikan sistem berfungsi dengan baik. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan dapat meningkatkan efisiensi dalam pencatatan transaksi, pengelolaan komisi staf, serta distribusi informasi acara kepada *tenant* dan staf. Dengan demikian, penerapan sistem ini diharapkan dapat meningkatkan kualitas layanan EO dan mengurangi potensi kesalahan administratif.

Kata kunci : *Event Organizer, Manajemen Transaksi, Extreme programming, Sistem Informasi*

1. PENDAHULUAN

Peran sistem informasi saat ini dapat mampu membantu perekonomian dan berdampak pada kegiatan bisnis. Banyak sektor bisnis yang sudah mengalami perubahan mengikuti perkembangan jaman menjadi digital dengan memanfaatkan penerapan dari teknologi [1].

Kemajuan teknologi saat ini mendorong perusahaan untuk segera menggali potensi mereka guna meningkatkan kinerja operasional, sehingga dapat menghasilkan pendapatan bagi perusahaan [2].

Dengan adanya teknologi informasi, perusahaan dapat memperoleh keunggulan kompetitif dan bersaing secara lebih efektif di pasar global [3].

Karena itu hampir semua perusahaan telah mengadopsi TI/SI, tetapi masih banyak ditemukan kesenjangan dalam penerapannya. Hal ini disebabkan oleh perencanaan strategis yang kurang memadai, sehingga TI/SI yang diterapkan belum optimal dan selaras dengan kebutuhan bisnis [4].

Setelah pandemi COVID-19, bisnis jasa menjadi meningkat, dan semakin dibutuhkannya *Event Organizer* (EO) atau *Event Manager* untuk mengatur sebuah acara. Tugas utamanya adalah membantu klien mereka agar dapat menjalankan acara yang sesuai dengan keinginan klien berjalan dengan lancar [5].

Dengan peningkatan bisnis EO yang beriringan dengan banyaknya acara dan bangkitnya perekonomian Indonesia terlebih lagi setelah pandemic COVID-19. Fungsi dasarnya adalah manajemen acara. Secara umum, bertugas untuk membantu klien mulai dari perencanaan, persiapan, pelaksanaan, dan evaluasi dari acara yang telah dilaksanakan [6].

Tanggung jawab EO biasanya dimulai dari sebelum acara yaitu, tahap perancangan konsep, kemudian dilanjutkan dengan perencanaan dan persiapan hingga acara selesai, dan tugas klien adalah memantau keberlangsungan acara untuk memastikan bahwa acaranya berjalan dengan lancar [7].

Dengan penerapan sistem informasi dalam perusahaan dapat memberikan beberapa keuntungan. Pertama, meningkatkan efisiensi dalam pekerjaan dengan mengotomatisasikan beberapa proses yang mengelola informasi, yang kedua dapat memperbaiki efektivitas manajemen dengan menyediakan informasi yang diperlukan untuk pengambilan keputusan, dan yang ketiga dapat meningkatkan daya saing organisasi dengan mengubah cara gaya berbisnis [8].

Proses transaksi yang dilakukan oleh perusahaan EO ini masih melalui *sales* lalu dikumpulkan bukti-bukti transaksi oleh *admin* lalu mengkonfirmasi dan memberikan *invoice* kepada *sales* dan *sales* akan meneruskan hal tersebut kepada *tenant*. Proses pengolahan data yang dilakukan juga masih menggunakan aplikasi Microsoft Excel untuk pencatatan transaksi, jam kerja *staff*, dan jumlah komisi yang didapatkan *staff*. Meskipun Whatsapp telah digunakan untuk mendistribusikan informasi *event* kepada anggota *staff* sudah berjalan dengan cukup baik. Namun, metode ini kurang efektif karena Whatsapp berfungsi hanya sebagai alat komunikasi sehingga tidak terhubung dengan *database* dan membuat *admin* harus melakukan pencatatan secara manual, yang dimana hal ini dapat meningkatkan potensi kesalahan *input* dan memperlambat proses kerja. Salah satu solusi yang bisa dipertimbangkan adalah dengan mengembangkan sistem yang berbasis web untuk memudahkan *admin* dalam pembuatan

administrasi kepada klien, membagikan informasi kepada *staff* dan *tenant*, dan pengontrolan *event*.

Website InfiniteBreeze yang akan dirancang menggunakan metode *extreme programming*. Metode *Extreme programming* dipilih karena memiliki tahapan yang simpel dan mampu mengatasi perubahan proses dengan cepat [9].

Berdasarkan permasalahan yang dihadapi oleh perusahaan EO dalam pengolahan data yang dapat meningkatkan risiko kesalahan dan memperlambat *workflow*. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem informasi berbasis *website* menggunakan metode *extreme programming*.

2. TINJAUAN PUSTAKA

Penelitian lainnya yang dilakukan oleh Muhammad Fachrul Rizky, Maria Cleopatra, dan Diyan Parwantiningtyas dengan judul Sistem Informasi Event Organizer pada Wahana Entertainment Berbasis Java. Bertujuan untuk mengatasi kendala dalam pengolahan data pemesanan yang masih dilakukan secara manual. Hasil dari penelitian ini menunjukkan bahwa penerapan sistem komputerisasi dapat mempercepat dan meningkatkan efisiensi proses pemesanan. Penelitian ini berfokus pada pengelolaan data pemesanan, paket layanan, dan serta manajemen karyawan [10].

Pada penelitian sebelumnya yang telah dilakukan oleh Aikal Alfrian Mucjal dengan judul Pengembangan E-Commerce Vendor dan Event Organizer Berbasis Website dengan Metode Waterfall. Bertujuan untuk menciptakan platform berbasis website yang dapat menghubungkan konsumen dengan event organizer serta vendor dalam satu platform yang terintegrasi. Website ini dikembangkan menggunakan metode waterfall dengan PHP Laravel dan menggunakan MySQL sebagai sistem manajemen basis data. Hasil dari penelitian ini adalah aplikasi berbasis website bernama Ivent, yang mempermudah konsumen dalam menemukan layanan event organizer secara lebih efisien pengujian dilakukan menggunakan blackbox testing untuk memastikan kualitas sistem yang dikembangkan [1].

2.1. Sistem Informasi

Sistem informasi adalah penerapan teknologi informasi dan komunikasi yang dijalankan oleh perusahaan bisnis [11].

Tujuan utama sistem komputer adalah memproses data untuk memperoleh informasi yang memerlukan dukungan dari elemen hardware, software, dan brainware [12].

Sistem informasi membantu manusia dan organisasi atau perusahaan dalam penciptaan, modifikasi, penyimpanan, komunikasi, dan penyebaran informasi dalam berbagai bidang kehidupan [13].

2.2. Event Organizer

Menurut [14], *event* atau acara merupakan suatu kegiatan yang diadakan untuk merayakan peristiwa penting bagi individu atau kelompok, dan biasanya terikat dengan adat, budaya, agama, serta tradisi, serta melibatkan banyak orang dan diadakan pada waktu tertentu. *Event organizer* adalah sebuah usaha yang secara konsisten dan berkesinambungan menerapkan konsep manajemen di dunia hiburan. Dalam pengertian sederhana, *Event Organizer* adalah pihak yang mengelola suatu acara [15].

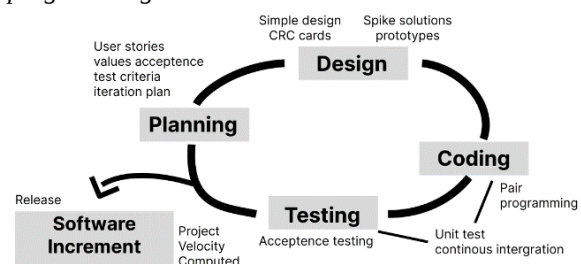
2.3. Extreme programming

Extreme programming (XP) adalah metode pengembangan sistem yang berfokus pada orientasi objek dan memiliki fleksibilitas tinggi dalam mengantisipasi perubahan *requirement* yang cepat [16].

Proses *Extreme programming* dimulai dengan tahap perencanaan, kemudian diikuti oleh empat langkah yang diulang setiap iterasi: *planning*, *design*, *coding*, dan *testing* [17].

3. METODE PENELITIAN

Penggunaan metode *Extreme programming* dalam penelitian ini karena mendukung pengembangan perangkat lunak berkualitas tinggi dengan praktik. Berdasarkan gambar 1 menjelaskan beberapa tahapan yang ada di dalam metode *Extreme programming*.



Gambar 1. Metode XP [16]

Pada gambar 1 menjelaskan tahapan-tahapan yang ada dalam metode XP:

- Planning**, tahap ini, peneliti melaksanakan perencanaan dalam pengembangan sistem, di mana dilakukan identifikasi masalah, analisis kebutuhan, serta penetapan jadwal pelaksanaan pembangunan sistem [18].
- Design**, pada tahap desain, kegiatan pemodelan dilakukan mulai dari pembuatan UML untuk pemodelan sistem, diagram untuk pemodelan arsitektur, hingga pemodelan basis data [19]. Pembuatan UML menggunakan *website* draw.io dan pembuatan desain tampilan menggunakan Figma.
- Coding**, tahapan ini melibatkan implementasi desain yang telah dirancang menjadi user interface menggunakan bahasa pemrograman [20]. Sistem manajemen transaksi dibangun menggunakan Framework Laravel dan database MySQL, serta

memanfaatkan Xampp untuk menjalankan web server dan mengelola database MySQL melalui phpMyAdmin.

- d. *Test*, pada tahap pengujian sistem dilakukan menggunakan metode blackbox untuk mengidentifikasi kesalahan yang terjadi selama aplikasi berjalan dan untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna [21].
- e. *Software increment*, pada tahap ini merupakan proses pengembangan sistem yang telah dibangun secara bertahap, dilakukan setelah sistem diterapkan dengan menambahkan fitur atau konten, sehingga meningkatkan kemampuan fungsionalitas sistem [22].

4. HASIL DAN PEMBAHASAN

Memuat hasil, Pengujian dan pembahasan tentang skripsi yang telah dilakukan

4.1. Planning

Pada tahap ini, berdasarkan hasil observasi dan wawancara yang dilakukan sehingga mendapatkan kebutuhan fungsional yang akan digunakan untuk mengembangkan sistem. Hasil analisis kebutuhan fungsional terdapat pada sebagai berikut:

a. Admin

- Menginput data Event
- Mengedit data Event
- Mendownload invoice transaksi tenant
- Membagikan informasi Event
- Mengupdate status staff
- Mengupdate status transaksi tenant
- Melihat ranking komisi staff
- Melihat history Event, Staff, Tenant
- Membuat report

b. Staff

- Menginput data diri
- Menerima informasi event
- Menginput jadwal kerja
- Melihat komisi
- Melihat history Event

c. Tenant

- Menginput data brand
- Melihat informasi event
- Melakukan transaksi
- Mendownload invoice transaksi
- Melihat history event

4.2. Design

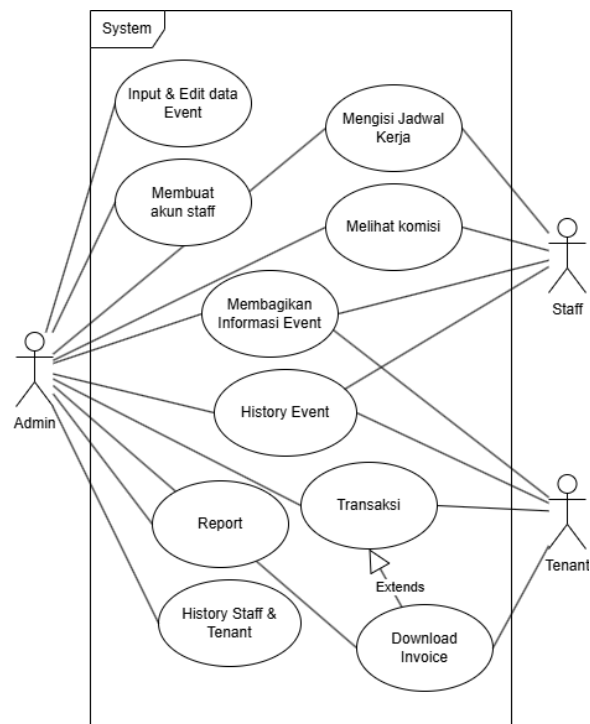
4.2.1. Use case Diagram

Pada tahap ini, dilakukan analisis terhadap sistem informasi yang dikembangkan untuk mengelola event. Diagram Use case yang ditampilkan menggambarkan interaksi antara aktor dan sistem dalam berbagai aktivitas yang dapat dilakukan oleh pengguna yang terdiri dari Admin, Staff, dan tenant. Pada gambar 2, dalam sistem ini menggambarkan fungsionalitas yang dapat diakses oleh masing-masing

aktor. Berikut adalah penjelasan dari masing-masing komponen yang terdapat dalam diagram:

a. Aktor

- *Admin*: Memiliki akses penuh terhadap sistem dan dapat menjalankan hampir semua fungsi yang tersedia.
- *Staff*: Bertanggung jawab dalam mengisi jadwal kerja dan melihat komisi.
- *Tenant*: Memiliki akses untuk melakukan transaksi dan mengunduh invoice.



Gambar 2. Use case diagram InfiniteBreeze

b. Use case

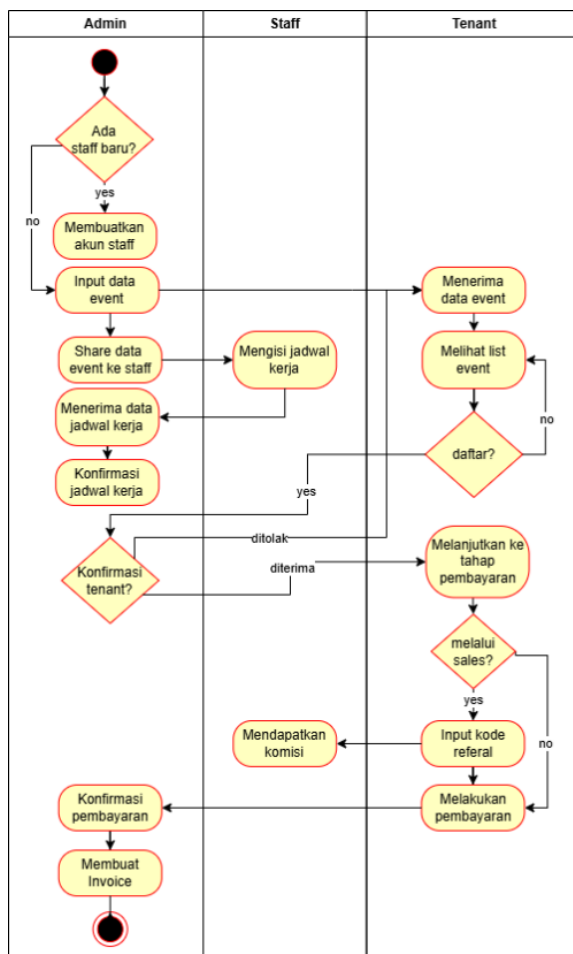
- *Input & Edit Data Event*: Admin dapat menambahkan dan memperbarui informasi event dalam sistem.
- *Membuat akun Staff*: Admin dapat membuatkan akun untuk staff.
- *Mengisi Jadwal Kerja*: Staff dapat memasukkan jadwal kerja mereka ke dalam sistem.
- *Melihat Komisi*: Staff dan admin dapat melihat komisi yang diperoleh berdasarkan jumlah tenant yang diperoleh.
- *Membagikan Informasi Event*: Admin dapat menyebarkan informasi mengenai event kepada pihak yang berkepentingan.
- *Transaksi*: Tenant dapat melakukan pembayaran atau transaksi terkait event.
- *History Event*: Admin, staff, dan tenant dapat melihat riwayat event yang telah berlangsung.
- *Download Invoice*: Admin dan tenant dapat mengunduh invoice dari transaksi yang telah dilakukan.
- *Report*: Admin dapat mengakses laporan mengenai berbagai aspek dalam sistem.

- *History Staff & Tenant*: Admin dapat melihat riwayat aktivitas *staff* dan *tenant* yang telah berinteraksi dengan sistem.

4.2.2. Activity Diagram

Pada Gambar 3 dapat dilihat Activity Diagram yang digunakan dalam sistem ini menggambarkan alur proses utama dalam pengelolaan *event* oleh *Admin*, *Staff*, dan *Tenant*. Diagram ini memperlihatkan bagaimana interaksi antar pengguna dalam sistem berjalan dari awal hingga akhir. Proses dimulai dengan *admin* melakukan *input data event* ke dalam sistem, setelah berhasil *menginput* sistem akan menerima dan membaca data dan menyimpan ke dalam *database*. Setelah sistem menyimpan ke dalam *database* *tenant* dapat melihat *event* yang akan diadakan selanjutnya melalui dashboardnya. *Admin* juga melakukan *share data*nya kepada *staff* sebagai informasi untuk mengisi jadwal kehadiran mereka untuk mengisi acara.

Jika *tenant* ingin mendaftar *event*, *tenant* dapat melakukan daftar melalui *website* namun belum melakukan pembayaran karena perlu dilakukan pengecekan apakah sesuai dengan tema *event* nya.



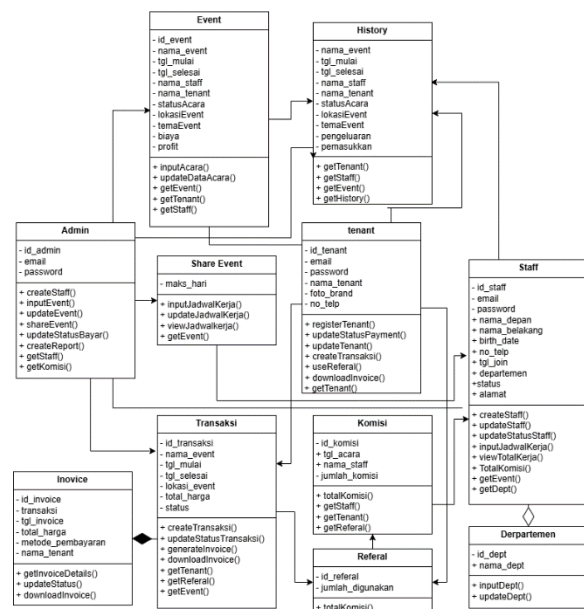
Gambar 3. Activity Diagram InfiniteBreeze

Jika tidak sesuai dengan tema yang di daftarkannya maka akan ditolak dan akan memunculkan notifikasi tidak dapat mendaftar

kembali di *event* tersebut dan tidak dapat mendaftar kembali. Jika sesuai dengan tema maka *admin* akan menerima pendaftaran dan akan memberikan notifikasi bahwa pendaftaran mereka diterima dan akan diarahkan untuk melanjutkan ke tahap pembayaran.

Pada tahap pembayaran jika *tenant* mengetahui melalui *sales* maka akan diarahkan register ke *website* dan melakukan pembayaran melalui *website*. Lalu perlu memasukkan kode referral *sales* tersebut pada saat akan melakukan pembayaran. Jika mendaftar karena melihat informasi pada *website* maka tidak perlu memasukkan kode referral dan melanjutkan pembayaran. Setelah itu *admin* akan mengkonfirmasi pembayaran tersebut. Setelah pembayaran dikonfirmasi sistem akan membuat *invoice* yang dapat di download oleh *tenant* maupun *admin*. Lalu jika *tenant* mengisi kode referral saat ingin melakukan pembayaran maka kode referral tersebut akan terbaca oleh sistem dan menambahkan komisi *sales* tersebut.

4.2.3. Class Diagram



Gambar 4. Class Diagram InfiniteBreeze

Pada gambar 4 menunjukkan class diagram, didalamnya memiliki 11 kelas dengan detail sebagai berikut: Admin memiliki peran dalam mengelola sistem, termasuk membuat akun untuk staff, memperbarui informasi event, dan menerima transaksi yang dilakukan oleh tenant. Event memiliki nama_event, tgl_mulai, tgl_selesai, lokasi, status, dan profit. History menyimpan Riwayat event yang telah selesai, tenant yang telah mengikuti event, dan staff yang telah resign. Share Event memiliki maks_hari dan metode untuk mengatur serta membagikan event kepada staff dan dapat diisi oleh staff mengenai hari kerja yang dapat dilakukan oleh staff berdasarkan maksimal hari yang telah ditentukan. Tenant memiliki id_tenant, nama, password, foto_tenant, dan job_tipe

serta bisa melakukan transaksi dan pembayaran. Staff memiliki nama, tanggal lahir, jenis kelamin, dan jabatan dalam sistem. Transaksi memiliki id_transaksi, tgl_transaksi, jumlah, status, dan metode pembayaran. Invoice memiliki id_invoice, nomor invoice, metode pembayaran, dan nama tenant serta dapat mendownload invoice. Komisi memiliki id_komisi, jumlah komisi, id_acara, dan id_staff, digunakan untuk mencatat komisi yang diperoleh staff. Referral memiliki id_referral dan jumlah_digunakan yang memungkinkan tenant menggunakan kode referral

untuk memberi komisi kepada staff. Departemen memiliki id_dept dan nama departemen serta dapat memperbarui informasi departemen.

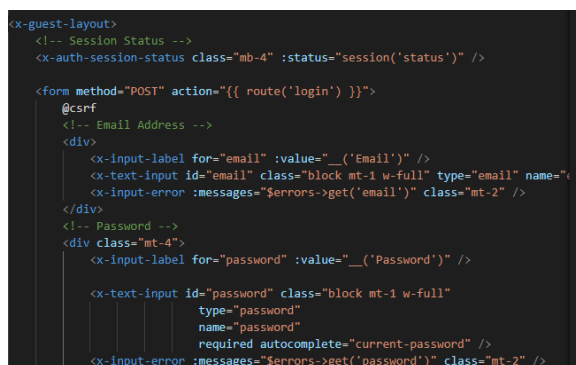
4.3. Coding

Tahap ini merupakan bagian yang krusial dalam pengembangan sistem, di mana *design* yang telah dirancang sebelumnya mulai diimplementasikan. Proses implementasi sistem ini menggunakan Laravel sebagai framework dan MySQL sebagai sistem manajemen basis data.

Tabel 1. Jadwal Iterasi Aplikasi InfiniteBreeze

No	Aktivitas	November				Desember				Januari				Februari	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2
1	Planning Iterasi 1														
	Menentukan studi kasus														
	Melakukan wawancara dan mengumpulkan data														
	Menganalisis masalah														
2	Design Iterasi 1														
	Design UML														
	Design ERD														
	Design Wireframe														
3	Coding Iterasi 1														
	Pengkodean														
4	Testing Iterasi 1														
	Testing Blackbox														
5	Planning Iterasi 2														
6	Design Iterasi 2														
7	Coding Iterasi 2														
8	Testing Iterasi 2														
9	Software Increment														

Berdasarkan tabel 1, penelitian ini dilakukan iterasi sebanyak dua kali. Iterasi pertama dilakukan pada bulan November hingga Desember lalu iterasi kedua dilakukan dari bulan Januari hingga pertengahan Februari. Pada gambar 5, memperlihatkan halaman dashboard untuk user *admin*. Terdapat side bar yang berisi fitur, *Transaction*, *Commission*, *Data*, *History*, dan *Setting*.

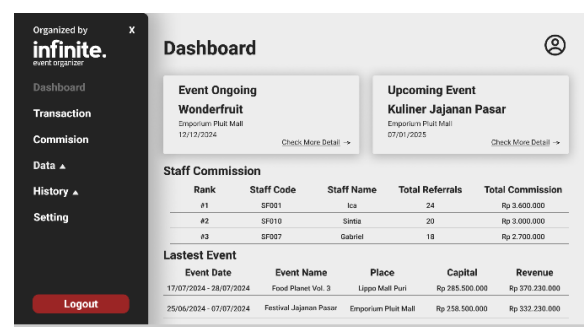


Gambar 5. Pengkodean aplikasi InfiniteBreeze

Pada tampilan dashboard terdapat 2 card yaitu, card untuk *event* yang sedang berjalan (Ongoing) dan untuk selanjutnya (Upcoming). *Admin* juga dapat melihat komisi *staff* untuk top 3 komisi terbanyak.

Terdapat juga informasi mengenai top 3 *event* yang baru saja dijalankan.

Selanjutnya pada fitur *Transaction admin* dapat melihat transaksi pembayaran yang telah dilakukan *tenant* dan *admin* dapat melakukan approval atau reject pendaftaran, setelah melakukan salah satunya maka status akan terupdate juga, jika *admin* mengapprove pendaftaran maka selanjutnya statusnya akan berubah menjadi menunggu pembayaran setelah *tenant* menyelesaikan pembayaran dan *admin* mengapprove nya maka status akan berubah menjadi confirm



Gambar 6. Halaman Dashboard Admin

Namun jika pada saat pendaftaran *tenant admin* mereject tersebut maka status akan diupdate menjadi reject dan tidak dapat mendaftar lagi di *event* tersebut

karena tidak sesuai dengan temanya. Namun jika pada saat pendaftaran diterima dan pembayaran ditolak maka status akan diupdate menjadi reject dan akan muncul di notifikasi *tenant* mengenai alasan pembayaran tersebut di reject. Jika pembayaran telah berhasil dilakukan akan muncul juga tombol button untuk mendownload *invoice* transaksi.

Selanjutnya fitur *Commision*, yang dimana pada fitur ini *admin* dan *staff* dapat melihat komisi yang didapatkan. Di *website admin*, dapat melihat komisi dari seluruh *staff* dan *tenant* apa saja yang mereka dapatkan, namun di *website staff* mereka tidak dapat melihat punya orang lain dan yang akan di tampilkan dalam fitur *Commision* adalah jumlah *tenant* mereka pada bulan yang sedang dijalankan. Selanjutnya fitur dropdown Data yang dimana bagian ini terdapat tiga bagian *Event*, *Staff*, dan *Tenant*. Fitur ini untuk melihat informasi-informasi mengenai *Event*, *Staff*, dan *Tenant*.

Dalam bagian *Event admin* dapat melihat informasi mengenai list *event* yang akan datang dan yang sedang berjalan. Terdapat juga fitur untuk melakukan add dan *edit*. Add digunakan untuk menambahkan *event* baru dan *Edit* digunakan untuk *admin* melakukan update status dan mengedit informasi. Terdapat tombol untuk melihat detail yang dimana berisi mengenai informasi *staff* yang bekerja, *tenant* yang berpartisipasi, dan jumlah pendapatan per hari nya. Selanjutnya bagian *Staff*, pada bagian ini berisi list *staff* yang ada di dalam perusahaan dan dapat mengupdate status kerja dan terdapat tombol New untuk menambahkan akun baru. Terdapat pula detail yang berisi informasi-informasi mengenai *staff* tersebut seperti data diri, profile, tanggal bergabung perusahaan, jabatan, dan list *event* yang pernah diikutinya. Pada bagian *Tenant*, bagian ini berisi list

tenant yang teregister di *website* InfiniteBreeze dan diurutkan sesuai abjad. Terdapat juga detail yang berisi mengenai data-data *tenant*, foto brand, dan list *event* yang pernah diikuti oleh *tenant* tersebut.

Selanjutnya pada fitur dropdown *History*, bagian ini memiliki fitur yang sama dengan dropdown Data, yang dimana fitur ini memiliki tiga bagian yaitu, *Event*, *Staff*, dan *Tenant*. Pada bagian *Event* digunakan untuk melihat list *event* yang sudah selesai dijalankan dan yang dibatalkan. Untuk bagian *staff* digunakan untuk melihat list *Staff* yang sudah tidak bekerja lagi. Lalu untuk bagian *Tenant* terdapat list *tenant* yang sudah tidak aktif lebih dari satu tahun, namun bukan berarti jika masuk ke *history* akun *tenant* tersebut akan hilang dan tidak dapat mengikuti *event* lagi.

Selanjutnya pada fitur setting, pada *admin* terdapat fitur untuk melihat atau mengupdate seperti profile, visi misi, alamat, email, dan juga bisa mengatur role dari user-user yang teregistrasi seperti *staff* yang sudah tidak bekerja dan dapat menonaktifkan akunnya agar tidak dapat diakses lagi oleh *staff* tersebut. Sedangkan fitur setting pada *staff* dapat digunakan untuk melihat atau mengupdate profile, email, nama, password, nomor telepon, dan alamat. Fitur setting pada *tenant* digunakan untuk melihat atau mengganti profile, deskripsi brand, nomor telepon, email, password, dan alamat.

4.4. Testing

Berdasarkan hasil implementasi pada tahap *Coding*, maka dilakukan pengujian yang dirincikan pada tabel 2 3, dan 4. Pengujian sistem ini menggunakan pengujian blackbox yang digunakan untuk mengidentifikasi fungsi fitur yang berada di sistem sudah beroperasi dengan sesuai.

Tabel 2. Penujian Blackbox User Admin

No	Halaman	Skenario Tes	Hasil yang diharapkan	Hasil Pengujian
1	Halaman Login	Mengisi form login <i>admin</i> menggunakan email dan password	<i>Admin</i> dapat masuk ke halaman Dashboard	Sesuai
2	Halaman Dashboard	<i>Admin</i> memilih menu Dashboard	Menampilkan halaman Dashboard	Sesuai
		<i>Admin</i> memilih menu <i>Transaction</i>	Menampilkan halaman <i>Transaction</i>	Sesuai
		<i>Admin</i> memilih menu <i>Commision</i>	Menampilkan halaman <i>Commision</i>	Sesuai
		<i>Admin</i> memilih menu <i>Event</i> dari Data	Menampilkan halaman <i>Event</i>	Sesuai
		<i>Admin</i> memilih menu <i>Staff</i> dari Data	Menampilkan halaman <i>Staff</i>	Sesuai
		<i>Admin</i> memilih menu <i>Tenant</i> dari Data	Menampilkan halaman <i>Tenant</i>	Sesuai
		<i>Admin</i> memilih menu <i>Event</i> dari <i>History</i>	Menampilkan halaman <i>History Event</i>	Sesuai
		<i>Admin</i> memilih menu <i>Staff</i> dari <i>History</i>	Menampilkan halaman <i>History Staff</i>	Sesuai
		<i>Admin</i> memilih menu <i>Tenant</i> dari <i>History</i>	Menampilkan halaman <i>History Tenant</i>	Sesuai
		<i>Admin</i> memilih menu Setting	Menampilkan halaman Setting	Sesuai
		<i>Admin</i> dapat menekan tombol Check More Detail	Menampilkan halaman Detail <i>Event</i>	Sesuai
		<i>Admin</i> dapat menekan logo profile	Menampilkan Setting Profile	Sesuai

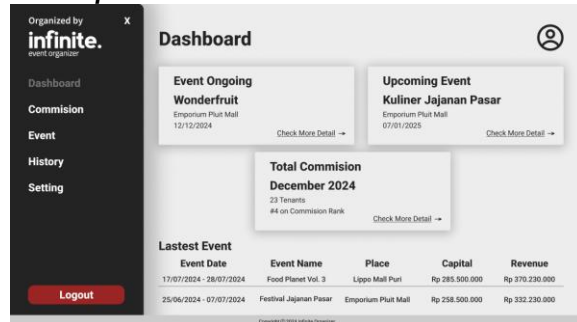
No	Halaman	Skenario Tes	Hasil yang diharapkan	Hasil Pengujian
3	Halaman Transaction	Admin dapat mengupdate status pendaftaran	Status pendaftaran berubah dan mengirimkan notifikasi kepada <i>tenant</i>	Sesuai
		Admin dapat mengupdate status pembayaran	Status pembayaran berubah dan mengirimkan notifikasi kepada <i>tenant</i>	Sesuai
		Admin dapat mendownload invoice	Admin dapat mengunduh <i>invoice</i> pembayaran	Sesuai
4	Halaman Commision	Admin dapat menekan Detail Commision setiap <i>staff</i>	Menampilkan halaman Detail <i>Staff Commision</i>	Sesuai
5	Halaman Data bagian Event	Admin dapat menekan tombol Add Event	Admin dapat menambahkan event baru	Sesuai
		Admin dapat menekan tombol Edit Event	Admin dapat mengedit event	Sesuai
		Admin dapat melihat Detail Event	Menampilkan halaman Detail Event	Sesuai
		Admin dapat menekan tombol Share	Admin dapat membagikan event baru kepada <i>staff</i>	Sesuai
		Admin dapat menekan tombol Add Schedule	Admin dapat menambahkan tabel jadwal kerja event	Sesuai
		Admin dapat menekan tombol Edit Schedule	Admin dapat mengedit tabel jadwal kerja event	Sesuai
6	Halaman Data bagian Staff	Admin dapat mengupdate status <i>staff</i>	Status <i>staff</i> berubah sesuai yang dipilih <i>admin</i>	Sesuai
		Admin dapat melihat Detail Staff	Menampilkan halaman Detail Staff	Sesuai
		Admin dapat membuat akun untuk Staff	Berhasil membuat akun <i>staff</i> dan <i>staff</i> bisa login	Sesuai
7	Halaman Data bagian Tenant	Admin dapat melihat Detail <i>tenant</i>	Menampilkan halaman Detail Tenant	Sesuai
8	Halaman History bagian Event	Admin dapat melihat Detail Event	Menampilkan halaman Detail Event	Sesuai
9	Halaman History bagian Staff	Admin dapat mengecek Detail Staff	Menampilkan halaman Detail Staff	Sesuai
10	Halaman History bagian Tenant	Admin dapat mengecek Detail <i>tenant</i>	Menampilkan halaman Detail Tenant	Sesuai
11	Halaman Setting	Admin dapat mengedit profile	Profile berubah	Sesuai

Tabel 3. Pengujian Blackbox User Staff

No	Halaman	Skenario Tes	Hasil yang diharapkan	Hasil Pengujian
1	Halaman Login	Mengisi form login <i>staff</i> menggunakan email dan password	<i>Staff</i> dapat masuk ke halaman Dashboard	Sesuai
2	Halaman Dashboard	<i>Staff</i> memilih menu Dashboard	Menampilkan halaman Dashboard	Sesuai
		<i>Staff</i> memilih menu Commision	Menampilkan halaman Commision	Sesuai
		<i>Staff</i> memilih menu Event	Menampilkan halaman Event	Sesuai
		<i>Staff</i> memilih menu History	Menampilkan halaman History	Sesuai
		<i>Staff</i> memilih menu Setting	Menampilkan halaman Setting	Sesuai
		<i>Staff</i> dapat menekan tombol Check More Detail	Menampilkan halaman Detail Event	Sesuai
		<i>Staff</i> dapat menekan logo profile	Menampilkan Setting Profile	Sesuai
3	Halaman Commision	<i>Staff</i> dapat melihat seluruh list <i>tenant</i> yang didapatkannya per bulan	Menampilkan list <i>tenant</i> yang didapatkan oleh <i>staff</i> per bulan	Sesuai
4	Halaman Event	<i>Staff</i> dapat melihat Detail Event	Menampilkan halaman Detail Event	Sesuai
		<i>Staff</i> dapat mengisi Jadwal Kerja	Menampilkan daftar tanggal yang masih tersedia untuk diisi sebagai tanggal event jika belum terisi oleh <i>staff</i> lain.	Sesuai
5	Halaman History	<i>Staff</i> dapat melihat list event yang sudah dikerjakan	Menampilkan list event yang sudah dikerjakan	Sesuai
		<i>Staff</i> dapat melihat Detail Event	Menampilkan halaman Detail Event	Sesuai
6		<i>Staff</i> dapat mengubah password	Password berubah	Sesuai

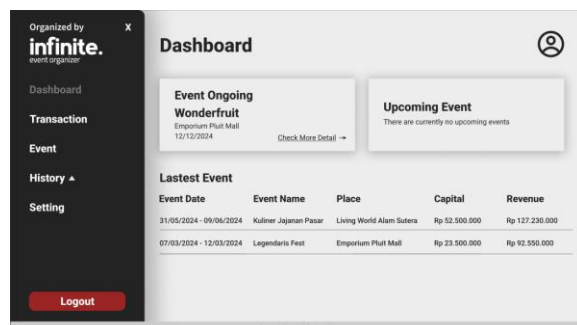
No	Halaman	Skenario Tes	Hasil yang diharapkan	Hasil Pengujian
	Halaman	Staff dapat mengganti profile	Profile berubah	Sesuai
	Setting	Staff dapat mengganti data diri	Data diri terupdate	Sesuai

4.5. Software Increment



Gambar 7. Dashboard Staff pada InfiniteBreeze

Staff memiliki dashboard yang dimana pada halaman dashboard ini berisi event yang sedang berjalan, yang akan datang, dan total komisi staff tersebut dalam bulan itu. Di dalam staff terdapat menu commission, event, history, dan setting. Pada commission dapat digunakan oleh staff untuk melihat list tenant yang sudah di dapatkannya. Lalu untuk bagian event, staff dapat melihat list event yang upcoming dan bisa mengisi jadwal pada page tersebut. Pada History dapat melihat event yang sebelumnya sudah pernah diikuti. Dan setting untuk mengganti profile, email, password, dan alamat.



Gambar 8. Dashboard Tenant pada InfiniteBreeze

Tenant juga memiliki dashboard seperti admin dan staff. Di dalam tenant ini terdapat menu transaction, event, history, dan setting. Pada transaction dapat digunakan oleh tenant untuk melakukan pembayaran mengikuti event. Lalu untuk bagian event tenant dapat melihat list event yang upcoming dan bisa melakukan registrasi. Pada History dapat melihat transaksi yang sebelumnya sudah pernah dilakukan. Dan setting untuk mengganti profile, email, password, logo, dan alamat.

5. KESIMPULAN DAN SARAN

Penelitian ini menghasilkan aplikasi manajemen transaksi berbasis web yang meningkatkan efisiensi operasional *Event Organizer* dengan mengotomatiskan pencatatan transaksi, distribusi informasi, dan

pengelolaan komisi staf. Pengujian menunjukkan bahwa sistem berjalan sesuai spesifikasi dan lebih unggul dibandingkan metode manual sebelumnya. Untuk pengembangan lebih lanjut, disarankan integrasi dengan pembayaran digital, pengembangan aplikasi mobile, penerapan kecerdasan buatan (AI) untuk analisis tema *event*, serta evaluasi berkala guna memastikan keandalan sistem. Dengan pengembangan berkelanjutan, sistem ini diharapkan memberikan manfaat lebih besar bagi industri EO dalam menghadapi digitalisasi.

DAFTAR PUSTAKA

- [1] A. A. Mucjal, "Pengembangan E-Commerce Vendor Dan Event Organizer Berbasis Website Dengan Metode Waterfall," Universitas Islam Indonesia, Jakarta, 2021. Accessed: Dec. 11, 2024. [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/34251>
- [2] J. F. Andry, B. G. Sudarsono, F. S. Lee, V. Christy, and L. Liliana, "Pengembangan Enterprise Resource Planning (ERP) Untuk Infrastruktur Industri SDGS 2030," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 4, pp. 7212–7218, Jun. 2024, doi: 10.36040/jati.v8i4.10204.
- [3] R. Doharma, A. A. Prawoto, and J. F. Andry, "Audit Sistem Informasi Menggunakan Framework Cobit 5 (Studi Kasus: PT Media Cetak)," *JBASE - Journal of Business and Audit Information Systems*, vol. 4, no. 1, pp. 22–28, Apr. 2021, doi: 10.30813/jbase.v4i1.2730.
- [4] J. F. Andry, L. Liliana, and A. Chakir, "Enterprise Architecture Landscape using Zachman Framework and Ward Peppard Analysis for Electrical Equipment Export Import Company," *Trends in Sciences*, vol. 18, no. 19, p. 23, Oct. 2021, doi: 10.48048/tis.2021.23.
- [5] I. D. Hasanti, "Hambatan Komunikasi Event Project Team dengan Account Executive pada Perusahaan Jasa Event Organizer Twisbless," *Journal of Servite*, vol. 1, no. 2, p. 37, Jan. 2021, doi: 10.37535/102001220195.
- [6] A. Muarif, A. Syarifudin, and M. R. H. Hamandia, "Analisis Faktor Keberhasilan Event Organizer Dalam Menyelenggarakan Acara (Studi Kasus Cv Soundtrack Indonesia)," *Journal of Business and Halal Industry*, vol. 1, no. 4, pp. 1–8, Jul. 2024, doi: 10.47134/jbhi.v1i4.414.
- [7] I. S. Pangkey, H. Mulyono, and J. S. Kalangi, "Strategi Komunikasi Bisnis Dalam Meningkatkan Pelanggan Event Organizer Onet Twenty One Organizer Desa Suwaan Minahasa Utara Provinsi Sulawesi Utara," *Acta Diurna*

- Komunikasi, vol. 3, no. 2, 2021, Accessed: Dec. 21, 2024. [Online]. Available: <https://ejournal.unsrat.ac.id/index.php/actadiurnakomunikasi/article/view/33628>
- [8] Sumaryanto, Purwati, and S. Prihatmoko, "Analisa Perancangan Sistem Informasi Akuntansi dalam pengelolaan Laporan Keuangan pada Perusahaan," *Bridge: Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 4, pp. 364–378, 2024, doi: <https://doi.org/10.62951/bridge.v3i1.331>.
- [9] A. L. Kalua, "Penerapan Extreme Programming Pada Sistem Informasi Keuangan Sekolah Berbasis Website," *Jurnal Ilmiah Informatika dan Ilmu Komputer (JIMA-ILKOM)*, vol. 1, no. 2, pp. 69–76, Sep. 2022, doi: 10.58602/jima-ilkom.v1i2.10.
- [10] M. F. Rizky, M. Cleopatra, and D. Parwatiningtyas, "Sistem Informasi Event Organizer Pada Wahana Entertainment Berbasis Java," *Journal of Information System, Applied, Management, Accounting and Research*, vol. 5, no. 1, p. 117, Feb. 2021, doi: 10.52362/jisamar.v5i1.346.
- [11] E. P. Primawanti and H. Ali, "Sistem Informasi Berbasis Web Dan Knowledge management terhadap Kinerja Karyawan (Literature Review Executive Support Sistem (Ess) For Business)," *Jurnal Ekonomi Manajemen Sistem Informasi*, vol. 3, no. 3, pp. 267–285, Jan. 2022, doi: <https://doi.org/10.31933/jemsi.v3i3.818>.
- [12] M. Aswiputri, "Literature Review Determinasi Sistem Informasi Manajemen: Database, Cctvdan Brainware," *Jurnal Ekonomi Manajemen Sistem Informasi*, vol. 3, no. 3, pp. 312–322, Jan. 2022, doi: <https://doi.org/10.31933/jemsi.v3i3.821>.
- [13] W. N. M. Sukma, A. Reynata, D. A. Yasmine, and D. M. P. Pratama, "Systematic Literature Review (Slr): Keamanan Dalam Sistem Informasi," *Journal of Comprehensive Science (JCS)*, vol. 2, no. 6, pp. 1738–1754, Jun. 2023, doi: 10.59188/jcs.v2i6.395.
- [14] A. Noor, *Management Event*. Bandung: Alfabeta, 2017.
- [15] I. N. Hafidz, *CEO: Chief Event Organizer*. Yogyakarta: Gava Media, 2017.
- [16] Z. Juniardi, A. Ariansyah, and N. Nurmayanti, "Rancang Bangun Aplikasi Top Up Voucher Game Online Berbasis Website Menggunakan Metode Extreme Programming," *Jurnal Minfo Polgan*, vol. 12, no. 2, pp. 1724–1733, Sep. 2023, doi: 10.33395/jmp.v12i2.12964.
- [17] G. Taufik, Y. Handrianto, and Suharjanti, "Model Extreme Programming Untuk Rancang Bangun Sistem Informasi Penjadwalan Kuliah," *Jurnal Infortech*, vol. 4, no. 2, pp. 190–199, 2022, doi: <https://doi.org/10.31294/infortech.v4i2.14065>.
- [18] C. D. Rukminingtyas, "Rancang Bangun Sistem Informasi Berbasis Website Lembaga Pelatihan Kewirausahaan (LPK) Ayostart dengan Metode Extreme Programming (XP)," *Jurnal Pilar Teknologi Jurnal Ilmiah Ilmu Teknik*, vol. 7, no. 2, pp. 43–51, Nov. 2022, doi: 10.33319/piltek.v7i2.136.
- [19] A. Hidayatulloh, "Rancang Bangun Aplikasi Penjualan Berbasis Website Pada UD. Riko Jaya," Universitas Dinamika, Surabaya, 2022. Accessed: Dec. 26, 2024. [Online]. Available: <https://repository.dinamika.ac.id/id/eprint/6503/>
- [20] T. Ardiansah, Y. Rahmanto, and Z. Amir, "Penerapan Extreme Programming Dalam Sistem Informasi Akademik SDN Kuala Teladas," *Journal of Information Technology, Software Engineering and Computer Science*, vol. 1, no. 2, pp. 44–51, 2023, doi: <https://doi.org/10.58602/itsecs.v1i1.25>.
- [21] M. D. Arfian and S. Wahyuni, "Implementasi Metode Extreme Programming Pada Website Marketplace Menggunakan API Indoregion dan Integrasi Midtrans (Studi Kasus: Toko Forbidden Merch)," *Prosiding SEINTEK Universitas Pamulang*, vol. 1, no. 1, pp. 162–173, 2022, Accessed: Dec. 30, 2024. [Online]. Available: <https://repository.unpam.ac.id/10537/>
- [22] A. Anharudin, S. Siswanto, and R. M. Syakira, "Rancang Bangun Data Storage System berbasis Web Dengan Metode Extreme Programming," *Jurnal Tekno Kompak*, vol. 16, no. 1, p. 123, Feb. 2022, doi: 10.33365/jtk.v16i1.1454.