

ROOMWISE: SISTEM INFORMASI PENGELOLAAN RUANG RAPAT BERBASIS ASP.NET (STUDI KASUS: PT XYZ BATAM)

Muhammad Ihsan Nabawi¹, Evaliata Br Sembiring²

¹ Politeknik Negeri Batam

Program Studi Teknik Informatika

Jl Ahmad Yani, Batam Centre, Batam 29461, Indonesia

E-mail: ihsannab213@gmail.com, eva@polibatam.ac.id

Abstrak

Pengelolaan ruang rapat secara manual sangat berpotensi menimbulkan berbagai permasalahan, yaitu bentrok jadwal pemakaian ruangan, informasi ketersediaan ruang yang tidak akurat, serta rendahnya efisiensi operasional, seperti halnya di PT XYZ Batam dengan tingkat penggunaan ruang rapat yang tinggi. Penelitian ini bertujuan untuk membuat sistem informasi pengelolaan ruang rapat berbasis *web*. Aplikasi *RoomWise* ini diharapkan dapat digunakan sebagai salah satu alternatif pengelolaan ruang rapat. Sistem dibuat menggunakan *framework ASP.NET Core MVC*, bahasa pemrograman *C#*, serta basis data *SQL Server*, dengan metode pengembangan *Prototype*. Hasil penelitian mencakup fitur utama sistem meliputi pemesanan ruang rapat secara *real-time*, manajemen data ruangan dan pengguna, pencarian dan pemfilteran ruang berdasarkan kriteria tertentu, notifikasi *email* otomatis, pencatatan aktivitas sistem, serta pengelolaan *feedback* pasca-rapat. Hasil pengujian sistem menunjukkan seluruh fungsional berjalan sesuai rancangan.

Kata kunci: Sistem Informasi Berbasis *Web*, Manajemen Ruang Rapat, ASP.NET Core MVC, *SDLC Prototype*, *SQL Server*.

Abstract

Manual meeting room management has the potential to cause various problems, such as scheduling conflicts in room usage, inaccurate room availability information, and low operational efficiency, as seen in PT XYZ Batam, where meeting room usage is relatively high. This study aims to develop a web-based meeting room management information system. The *RoomWise* application is expected to serve as an alternative solution for managing meeting rooms. The system was developed using the *ASP.NET Core MVC* framework, the *C#* programming language, and a *SQL Server* database, with the *Prototype* development method. The results of this study include the main system features, such as real-time meeting room booking, room and user data management, room searching and filtering based on specific criteria, automatic email notifications, system activity logging, and post-meeting feedback management. The system testing results show that all functionalities run according to the designed specifications.

Keywords : Web-Based Information System, Meeting Room Management, ASP.NET Core MVC, *SDLC Prototype*, *SQL Server*.

1. Pendahuluan

1.1 Latar Belakang

Ruang rapat digunakan untuk menyelenggarakan berbagai kegiatan, mulai dari diskusi internal, presentasi proyek, hingga pertemuan dengan pihak eksternal. Keberadaan ruang rapat yang memadai tidak hanya mencerminkan profesionalisme perusahaan, tetapi juga menjadi indikator kemampuan organisasi dalam mengelola sumber daya secara efektif. Di perusahaan berskala besar seperti PT XYZ Batam, tingkat kebutuhan akan ruang rapat sangat tinggi dan memerlukan sistem pengelolaan yang efisien dan terstruktur. Kompleksitas operasional yang melibatkan berbagai departemen dengan jadwal yang padat menuntut adanya mekanisme koordinasi yang solid untuk memastikan setiap unit kerja dapat mengakses fasilitas rapat sesuai kebutuhan tanpa menimbulkan hambatan operasional.

Namun, pengelolaan ruang rapat secara manual atau menggunakan sistem sederhana seperti pencatatan melalui *spreadsheet* sering kali menimbulkan berbagai permasalahan, seperti bentrok jadwal peminjaman, informasi yang tidak akurat mengenai ketersediaan ruangan, serta kurangnya transparansi dalam pemanfaatan ruang. Dampaknya, tidak jarang terjadi situasi di mana beberapa tim memesan ruangan yang sama pada waktu yang bersamaan, atau ruangan yang sebenarnya tersedia tidak termanfaatkan karena informasi yang tidak terupdate dengan baik.

Permasalahan ini dibuktikan oleh tidak adanya sistem terintegrasi untuk memesan ruang rapat yang menyebabkan pemesanan dilakukan secara manual melalui komunikasi verbal atau dokumen sederhana, yang sering kali menyebabkan konflik jadwal antar pengguna [1]. Peminjaman ruang *meeting* yang masih dilakukan secara konvensional mempunyai tingkat akurasi yang sangat minim dan penanganan yang memakan waktu dalam pencarian informasi [2]. Proses manual juga menimbulkan beban administratif yang signifikan bagi petugas pengelola ruangan, di mana mereka harus terus-menerus memperbarui catatan, merespons pertanyaan ketersediaan, dan menangani perubahan jadwal secara manual. Lebih lanjut, metode pencatatan manual rentan terhadap kesalahan manusia, di mana pembaruan jadwal dapat terabaikan atau tidak ada pengingat notifikasi, bahkan untuk pengguna yang telah memesan ruangan [3]. Ketergantungan pada memori dan catatan fisik membuat sistem ini sangat tidak andal, terutama ketika volume pemesanan meningkat pada periode-periode sibuk atau ketika terjadi pergantian petugas pengelola.

Metode manual meningkatkan resiko kesalahan dalam masukan data dan pencatatan, sehingga perusahaan tidak dapat memanfaatkan sumber daya secara optimal [4]. Kendala-kendala tersebut tidak

hanya menghambat operasional sehari-hari, tetapi juga dapat menyebabkan ketidakpuasan pengguna internal, baik dari sisi administrasi maupun para staf yang membutuhkan ruang. Ketidakefisienan ini pada akhirnya berdampak pada citra profesionalisme perusahaan, terutama ketika terjadi kekacauan jadwal yang melibatkan tamu atau klien eksternal. Hilangnya waktu produktif akibat mencari ruangan pengganti atau menunggu konfirmasi ketersediaan juga menimbulkan biaya tersembunyi yang cukup signifikan bagi organisasi dalam jangka panjang.

Sistem informasi merupakan kombinasi dari teknologi, orang dan proses yang dirancang untuk mengumpulkan data, memproses, menyimpan, dan mendistribusikan informasi [1]. Tujuan utama dari hadirnya sistem informasi adalah untuk meningkatkan efisiensi, mendukung pengambilan keputusan, dan mengoptimalkan operasi dalam suatu organisasi dengan mengubah data menjadi informasi yang berguna. Dalam konteks modern, sistem informasi tidak lagi sekadar alat bantu administratif, melainkan telah menjadi *backbone* yang menentukan daya saing dan kelancaran operasional perusahaan. *Management Information Systems* (MIS) adalah sistem informasi yang berada pada level manajemen, yang menyediakan fungsi pengambilan keputusan, pemantauan, pengendalian, dan administrasi [1].

Seiring berkembangnya teknologi informasi, penerapan sistem informasi berbasis *web* menjadi solusi yang tepat untuk mengatasi permasalahan tersebut. *Framework* ASP.NET MVC merupakan salah satu metode yang dikembangkan oleh Microsoft yang digunakan untuk pembangunan *website* dinamis, dengan kelebihan yaitu *SEO* (*Search Engine Optimizer*) *friendly*, artinya arsitektur ini dapat ter-*index* dengan mudah oleh *search engine* seperti Google, Yahoo, dan Bing [5]. Kemampuan ASP.NET MVC dalam memisahkan logika bisnis, tampilan, dan kontrol data membuat pengembangan dan pemeliharaan sistem menjadi lebih terstruktur dan mudah dikembangkan di masa mendatang. ASP.NET, sebagai *framework* pengembangan *web* yang dikembangkan oleh Microsoft, menyediakan berbagai fitur dan keunggulan dalam membangun aplikasi *web* yang dinamis, aman, dan terintegrasi [6]. Keamanan yang menjadi prioritas dalam ASP.NET sangat penting untuk melindungi data pemesanan dan informasi pengguna dari akses yang tidak berwenang.

Solusi sistem berbasis *web* memungkinkan proses manajemen ruang rapat menjadi lebih terorganisasi dan transparan. Implementasi aplikasi berbasis *web* dapat meningkatkan pemanfaatan ruang rapat berkat fitur seperti pencatatan notula, pemantauan *real-time*, dan pengingat jadwal [3]. Transparansi yang tercipta melalui sistem berbasis *web* memungkinkan seluruh karyawan memiliki akses yang sama terhadap informasi ketersediaan ruangan, sehingga prinsip

keadilan dalam pemanfaatan fasilitas dapat terwujud. Sementara itu, penggunaan sistem berbasis *web* secara signifikan mengurangi konflik jadwal yang sering terjadi akibat metode pencatatan manual [7].

Sistem informasi peminjaman ruangan berbasis *web* yang dirancang dengan mengotomatisasi proses peminjaman dapat menyediakan akses informasi langsung terhadap ketersediaan ruang, memudahkan peminjaman, serta meminimalisir resiko kesalahan dalam pengelolaan jadwal [5]. Dengan fitur pencarian berdasarkan kriteria tertentu, sistem ini dapat meningkatkan efisiensi kerja dalam pengelolaan ruang. Karyawan dapat dengan mudah mencari ruangan berdasarkan kapasitas, lokasi, atau fasilitas yang tersedia, sehingga proses pemilihan ruangan menjadi lebih cepat dan tepat sasaran. Sistem ini meningkatkan efisiensi dalam pengelolaan ruang rapat, memungkinkan pengguna untuk melakukan reservasi secara *real-time*, serta mengurangi konflik jadwal akibat pencatatan manual [1]. Fitur notifikasi otomatis juga memastikan bahwa pengguna mendapatkan pengingat sebelum waktu pertemuan, mengurangi tingkat ketidakhadiran atau kelupaan yang dapat mengganggu jadwal perusahaan.

Oleh karena itu, pemanfaatan ASP.NET dalam merancang sistem informasi ruang rapat dengan pendekatan metode *Prototype* diharapkan dapat membantu perusahaan dalam mengelola jadwal penggunaan ruangan secara *real-time*, menampilkan status ruangan (tersedia, digunakan, dalam perbaikan), serta membantu/memudahkan proses *booking* oleh karyawan secara mandiri dan transparan. Kemandirian karyawan dalam melakukan pemesanan mengurangi ketergantungan pada petugas administrasi, sehingga beban kerja dapat terdistribusi dengan lebih baik.

1.2 Rumusan Masalah

Berikut beberapa rumusan masalah yang dapat disimpulkan dari latar dan pendahuluan proyek ini.

1. Bagaimana merancang dan membangun sistem informasi pengelolaan ruang rapat berbasis *web* di PT XYZ Batam yang dapat mengatasi permasalahan bentrok jadwal, ketidakakuratan informasi ketersediaan ruang, serta kesulitan operasional yang selama ini terjadi akibat pengelolaan secara manual?
2. Bagaimana merancang fitur-fitur pendukung seperti pencarian dan pemfilteran ruang berdasarkan kriteria tertentu, notifikasi *email* otomatis, pencatatan aktivitas sistem, serta pengelolaan *feedback* pasca-rapat agar proses pemesanan ruang rapat menjadi lebih transparan dan sesuai dengan kebutuhan perusahaan?

1.3 Tujuan

Dari rumusan-rumusan masalah tersebut maka dapat diambil beberapa tujuan dari proyek penelitian ini, berikut beberapa tujuan itu.

1. Merancang dan membangun sistem informasi pengelolaan ruang rapat berbasis *web* (*RoomWise*) di PT XYZ Batam menggunakan *framework* ASP.NET Core MVC, bahasa pemrograman C#, dan basis data SQL Server dengan metode pengembangan *Prototype*, guna mengatasi permasalahan bentrok jadwal, ketidakakuratan informasi ketersediaan ruang, serta kesulitan operasional akibat pengelolaan manual?
2. Merancang dan mengimplementasikan fitur pencarian dan pemfilteran ruang berdasarkan kriteria tertentu, notifikasi *email* otomatis, pencatatan aktivitas sistem, serta pengelolaan *feedback* pasca-rapat, sehingga proses pemesanan ruang rapat menjadi lebih transparan dan sesuai dengan kebutuhan perusahaan?

1.4 Batasan Masalah

Agar penelitian ini lebih terarah, maka batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Sistem *RoomWise* dikembangkan khusus untuk kebutuhan pengelolaan ruang rapat di PT XYZ Batam dan tidak mencakup pengelolaan aset atau fasilitas kantor lainnya.
2. Sistem dibangun menggunakan *framework* ASP.NET Core MVC, bahasa pemrograman C#, dan basis data SQL Server, serta tidak membahas perbandingan dengan *framework* atau teknologi lain.
3. Pengujian sistem dilakukan menggunakan metode *black-box testing* dan tidak mencakup pengujian keamanan (*security testing*) secara mendalam.
4. Aplikasi ini di-deploy secara lokal.
5. Data yang ditampilkan demi kepentingan diluar perusahaan merupakan data *dummy*.

1.5 Manfaat

Manfaat dari pembuatan proyek penelitian yang menjadi tugas akhir ini antara lain.

1. Membantu PT XYZ Batam dalam mengelola jadwal penggunaan ruang rapat secara lebih terstruktur dan bebas dari bentrok jadwal.
2. Meningkatkan transparansi informasi ketersediaan ruang rapat bagi seluruh karyawan sehingga pemanfaatan ruang menjadi lebih optimal.
3. Memudahkan karyawan dalam melakukan pemesanan ruang rapat secara mandiri, cepat, dan transparan tanpa harus bergantung pada petugas administrasi.

4. Mengurangi beban administratif petugas pengelola ruangan melalui otomatisasi proses pemesanan, notifikasi, dan pencatatan aktivitas sistem.
5. Menambah wawasan dan pengalaman penulis dalam merancang serta membangun sistem informasi berbasis *web* menggunakan *framework* ASP.NET Core MVC dan basis data SQL Server sebagai penerapan ilmu yang telah dipelajari selama masa perkuliahan.

2. Tinjauan Pustaka

Beberapa penelitian telah dilakukan terkait pengembangan sistem informasi berbasis *web* untuk manajemen sistem ruang rapat atau *meeting room* dengan berbagai pendekatan dan metode pengembangan. Penelitian-penelitian tersebut menjadi acuan penting dalam pengembangan *website RoomWise* ini.

Pada [8] dilakukan pengembangan sistem informasi peminjaman ruangan berbasis *web* pada Fakultas Teknik. Penelitian tersebut menggunakan metode *Software Development Life Cycle (SDLC) Prototype* untuk menggantikan proses peminjaman yang masih manual dan berbasis kertas, karena saat itu peminjaman ruangan dilakukan melalui formulir fisik, menyebabkan birokrasi berbelit, antrean panjang, serta risiko dokumen hilang atau keliru. Penerapan Model *Prototype* dan teknologi yang ada beserta rancangannya maka dihasilkan platform yang lebih komprehensif dan mampu meminimalisir bentrok jadwal dan memberikan transparansi informasi ruang.

Pada [6] menghadapi permasalahan mengenai bagaimana teknologi *web* sebelumnya (PHP polos) menghadapi kendala dalam hal modularitas, pemisahan kode, dan keterbacaan menyulitkan tim dalam mengelola dan mengembangkan sistem secara kolaboratif. Serta kode yang tidak dikompilasi rentan terhadap kesalahan *runtime*, serta menyulitkan *debugging* maupun perbaikan *bug*. Penerapan ASP.NET dalam pengembangan sistem informasi berbasis *web* memberikan kemudahan dalam pengembangan aplikasi *web* yang aman dan terstruktur, serta mendukung integrasi dengan database SQL Server. Akhirnya sistem informasi yang telah dibangun mampu memenuhi kecukupan kebutuhan teknologi dan fungsionalitas yang memadai.

Selanjutnya pada [1] dilakukan rancang bangun sistem informasi manajemen ruang rapat berbasis *web* di PT XYZ untuk mengatasi masalah pengelolaan ruang rapat yang masih *manual* melalui komunikasi *verbal* atau dokumen sederhana, menyebabkan konflik jadwal antar karyawan, inefisiensi operasional, kurangnya transparansi ketersediaan ruang, serta sulitnya pemantauan *real-time* di berbagai lantai gedung. Penelitian tersebut mengidentifikasi permasalahan utama dari wawancara dan observasi,

seperti ketergantungan pada *admin* untuk pengecekan *manual* yang memicu antrean panjang dan kesalahan pencatatan, sehingga memerlukan solusi terintegrasi yang terstruktur. Pengembangan mengadopsi metode *Prototype* secara sekuensial, mulai analisis kebutuhan, desain (alur proses, *UI/UX* responsif, basis data), implementasi (*React.js front-end*, *Express.js back-end*, *Prisma ORM* dengan RDMBS), hingga pengujian *Black Box* untuk menghasilkan sistem yang mendukung reservasi daring, pengelolaan jadwal otomatis, dan notifikasi *email*.

Pada [4] dilakukan perancangan dan implementasi *sistem peminjaman ruang* berbasis *web* di Fakultas Teknologi Informasi UKSW Salatiga dengan menggunakan *framework Laravel* untuk menggantikan proses peminjaman manual yang masih bergantung pada formulir dan pencatatan konvensional. Sistem dirancang agar alur peminjaman, pengelolaan data ruangan, dan pengaturan hak akses antara *pengguna* dan *admin* dapat terintegrasi dalam satu platform yang lebih terstruktur, sehingga mengurangi kesalahan pencatatan lalu juga mempermudah pemantauan ketersediaan ruangan. Pendekatan *Prototype* ini memungkinkan adanya umpan balik berkelanjutan dari pengguna, sehingga desain antarmuka, alur peminjaman, dan pengelolaan data ruangan dapat disempurnakan sebelum diimplementasikan penuh di dalam aplikasi berbasis *Laravel*.

Berbagai penelitian tersebut menunjukkan bahwa pengembangan sistem informasi *RoomWise* berbasis *web* dengan menggunakan metode *SDLC Prototype* dan teknologi ASP.NET serta SQL Server dapat meningkatkan dapat kemudahan pengelolaan ruang rapat, mengurangi potensi bentrok jadwal, serta memberikan transparansi informasi ketersediaan ruang secara *real-time*. Selain itu, penggunaan metode *Prototype* memungkinkan pengembang untuk memperoleh umpan balik pengguna secara iteratif sehingga fitur, antarmuka, dan alur pemesanan pada *RoomWise* dapat disesuaikan dengan kebutuhan operasional secara lebih tepat. Integrasi ASP.NET dan SQL Server juga mendukung keamanan data, modularitas dan skalabilitas sistem, serta kemudahan pemeliharaan dan pengembangan lebih lanjut, sehingga *RoomWise* berpotensi menjadi solusi komprehensif dalam manajemen ruang rapat di lingkungan organisasi.

Selain tinjauan terhadap beberapa penelitian tentang pengembangan aplikasi *roomwise*, pada proyek ini juga dilakukan beberapa literasi tentang sistem informasi dan pengembangannya.

A. Sistem Informasi

Sistem informasi merupakan kombinasi teknologi, manusia, dan proses yang berfungsi untuk mengumpulkan, mengolah, menyimpan, serta mendistribusikan data menjadi informasi yang berguna. Tujuan utamanya adalah meningkatkan efisiensi, mendukung pengambilan keputusan, dan mengoptimalkan operasional organisasi [1].

B. Visual Studio Code

Visual Studio Code adalah *software* yang ringan namun kuat, dirancang untuk mendukung berbagai bahasa pemrograman, termasuk ASP.NET dan PHP. Berbasis pada teknologi lintas platform dari *GitHub's Electron*, editor ini menawarkan dukungan untuk Javascript, *Node.js*, dan berbagai ekstensi atau bahasa lain seperti *C#* dan *Python*. Dengan antarmuka pengguna yang intuitif, *Visual Studio Code* memudahkan pengembang untuk mengelola dokumen dan folder, serta menyediakan fitur canggih seperti *IntelliSense* dan pelengkapan otomatis, yang sangat berguna dalam pengembangan aplikasi *web* menggunakan PHP. *Visual Studio Code* dianggap sebagai lingkungan pengembangan terintegrasi (*IDE*) yang komprehensif, ideal bagi pengembang yang bekerja dengan berbagai teknologi dan bahasa pemrograman [9].

C. ASP.NET

ASP.NET Core MVC sebagai *framework* pengembangan web. ASP.Net MVC merupakan model pengembangan yang diadaptasi Microsoft untuk mengatasi kelemahan *web forms*, dengan mengadopsi pola *Model-View-Controller* (MVC) yang memungkinkan pemisahan perhatian (*separation of concern*). Pendekatan ini menjadikan pengembangan lebih modular, bersih, dan terstruktur, sehingga cocok untuk aplikasi web skala besar. Selain itu, MVC memudahkan pengujian unit (*unit testing*) dibandingkan model pengembangan lain [10].

D. C#

C# atau *C-Sharp* adalah bahasa pemrograman yang beroperasi dalam ekosistem .NET (*dotNET*). Meskipun tergolong bahasa yang relatif baru dengan keterbatasan kompatibilitas, *C#* menawarkan berbagai fitur yang menjanjikan dan potensial. Sebagai bahasa pemrograman berorientasi objek yang memiliki inti, *C#* menggabungkan elemen-elemen terbaik dari beberapa bahasa pemrograman, seperti efisiensi yang mirip dengan *C++*, kesederhanaan desain berorientasi objek seperti Java, dan kemudahan penggunaan bahasa seperti *Visual Basic* [11].

E. HTML dan CSS

HTML yaitu singkatan dari *Hypertext Markup Language*, merupakan bahasa standar untuk membangun struktur sebuah halaman *website*. Bahasa ini dikelola standarnya oleh *World Wide Web Consortium*. HTML menggunakan kumpulan tag untuk menyusun berbagai elemen yang ada dalam *website*. Perannya sangat penting dalam menentukan tata letak (*layout*) dari elemen-elemen tersebut sehingga setiap bagian halaman *web* dapat ditampilkan sesuai dengan desain yang diinginkan [12].

CSS yaitu singkatan dari *Cascading Style Sheets*, berfungsi untuk memperindah tampilan elemen-elemen dalam dokumen HTML serta menentukan bagaimana elemen-elemen tersebut ditampilkan di *browser*. Istilah *cascading* merujuk pada gaya berlapis atau hierarki, di mana format yang diterapkan pada elemen induk secara otomatis diwarisi oleh elemen anaknya. CSS memungkinkan pengaturan berbagai aspek visual, seperti jarak antar baris, warna, jenis font, format border, hingga tampilan file gambar. Dengan CSS, tampilan dokumen HTML dapat disesuaikan agar sesuai kebutuhan desain [12].

F. Javascript

Bahasa pemrograman ini terdiri dari serangkaian *script* yang dijalankan dalam sebuah dokumen. Javascript dapat meningkatkan tampilan dan fungsionalitas dalam aplikasi berbasis web yang sedang dikembangkan. Merupakan bahasa pemrograman tingkat tinggi, berbasis klien, berorientasi objek, dan bersifat longgar [13].

G. Bootstrap

Bootstrap adalah *framework* yang digunakan untuk membangun dan merancang tampilan halaman *website* dengan lebih cepat dan efisien. Alat ini sangat berguna bagi pengembang *web* dan desainer untuk mempercepat proses pembuatan tampilan *website* yang responsif, menarik, dan ramah pengguna [14].

H. SQL Server

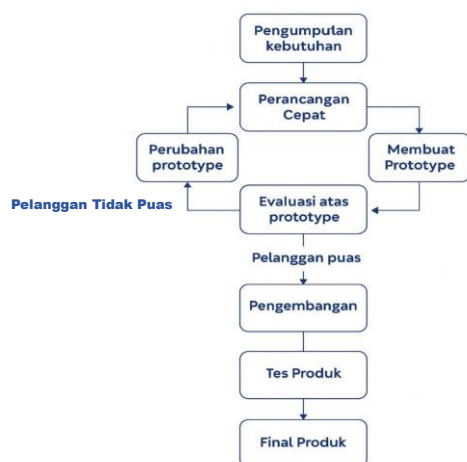
Microsoft SQL Server adalah sistem manajemen basis data relasional (*RDBMS*) yang dikembangkan oleh Microsoft untuk menangani penyimpanan dan pengelolaan data dalam format tabel yang saling terhubung. Sebagai salah satu *RDBMS* terkemuka, SQL Server memiliki kemampuan untuk mengatur data dengan efisien dan memungkinkan akses yang cepat serta akurat. Dengan dukungan berbagai fitur canggih, seperti pemrosesan transaksi, pengelolaan *query* yang kompleks. SQL Server menjadi pilihan utama bagi banyak organisasi yang memerlukan solusi pengelolaan data yang handal [15].

1. SSMS 20

SQL Server Management Studio (SSMS), merupakan sebuah *integrated environment* yang menjadi antarmuka utama bagi administrator basis data dalam mengelola SQL Server. SSMS menyediakan antarmuka manajemen yang lengkap, memungkinkan pengguna untuk mengonfigurasi dan berinteraksi dengan basis data melalui satu konsol. SSMS bisa digunakan dengan cara mengeluarkan perintah *Transact-SQL* ke basis data SQL Server maupun untuk membangun basis data menggunakan antarmuka grafis yang disediakan oleh SSMS. *SQL Server Management Studio (SSMS)* dirancang oleh Microsoft sebagai pengganti alat-alat terpisah (seperti *Query Analyzer* dan *Enterprise Manager*) yang ada pada versi SQL Server sebelumnya. SSMS memungkinkan pengguna untuk mengelola beberapa instance SQL Server dari satu platform Tunggal [16].

3. Metode Penelitian

Penelitian ini menggunakan menggunakan metode *SDLC Prototype* sebagai metode pengembangannya. Gambaran umum metode *prototype* dapat dilihat pada Gambar 1 dibawah yang saya referensikan dari penelitian sebelumnya [17].



Gambar 1. Metode SDLC Prototype (Sumber: [17]).

Dalam pengembangan sistem informasi, pemilihan metode pengembangan yang tepat sangat penting untuk memastikan sistem yang dihasilkan sesuai dengan kebutuhan pengguna. Metode *Prototype* merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat suatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai [2]. Keunggulan utama dari metode ini adalah kemampuannya untuk menghasilkan model awal yang dapat langsung diuji dan dievaluasi oleh calon pengguna, sehingga feedback dapat diperoleh sejak tahap awal pengembangan.

Metode *Prototype* dipilih karena memungkinkan pengembangan sistem dilakukan secara bertahap dengan partisipasi aktif pengguna dengan melakukan evaluasi berdasarkan umpan balik yang diterima, sehingga sistem dapat disesuaikan dengan cepat apabila ada perubahan kebutuhan, serta membantu dalam mengidentifikasi dan menangani kendala sebelum implementasi akhir [7].

3.1 Analisis atau Pengumpulan Kebutuhan

Pada tahap analisis atau pengumpulan kebutuhan, kebutuhan pengguna diidentifikasi untuk menghasilkan spesifikasi kebutuhan fungsional dan nonfungsional sebagai dasar dalam pembuatan prototype [2][7]. *Output* dari fase ini mencakup dokumen spesifikasi, kebutuhan fungsional, dan juga non-fungsional aplikasi, serta gambaran umum sistem.

3.1.1 Gambaran Umum Sistem



Gambar 2. Gambaran Umum Sistem.

Seperti pada Gambar 2, Sistem informasi yang dikembangkan dalam tugas akhir ini diberi nama *RoomWise*, yaitu sebuah aplikasi berbasis *web* yang dirancang khusus untuk mengelola proses pemesanan ruang rapat di PT XYZ Batam. Sistem ini bertujuan untuk menggantikan metode manual yang sebelumnya digunakan oleh perusahaan dalam pengelolaan ruang rapat, seperti pencatatan melalui *spreadsheet* atau komunikasi informal melalui *email* dan *chat*. Pengguna sistem terdiri dari dua peran utama, yaitu administrator yang bertanggung jawab atas pengelolaan data ruangan dan jadwal, serta karyawan yang dapat melakukan pemesanan dan melihat informasi ketersediaan ruang.

3.1.2 Kebutuhan Fungsional

Kebutuhan fungsional diberikan dengan bentuk kode fungsi, Kode KF untuk Kebutuhan Fungsional.

TABEL I

TABEL KEBUTUHAN FUNGSIONAL

Kode	Kebutuhan Fungsional
KF-01	<i>Semua pengguna</i> dapat mengakses halaman login
KF-02	<i>Semua pengguna</i> dapat melakukan <i>sign-in</i> sesuai peran
KF-03	<i>Semua Pengguna</i> dapat mengelola profil pribadi dan mengubah kata sandi
KF-04	<i>Admin</i> dapat melihat halaman dashboard
KF-05	<i>Admin</i> dapat mengelola data pengguna
KF-06	<i>Admin</i> dapat mengaktifkan dan menonaktifkan akun pengguna
KF-07	<i>Admin</i> dapat mengelola data <i>master</i> ruang rapat, meliputi data lokasi, ruang rapat, fasilitas ruang rapat, serta seluruh data pemesanan ruang rapat.
KF-08	<i>Admin</i> dapat mengirimkan notifikasi email otomatis
KF-09	<i>Admin</i> dapat melihat log aktivitas sistem
KF-10	<i>Admin</i> dapat mengekspor data dalam format <i>Spreadsheet</i>
KF-11	<i>Karyawan</i> dapat melihat ketersediaan ruang rapat
KF-12	<i>Karyawan</i> dapat mencari dan memfilter data ruang rapat berdasarkan data yang ada
KF-13	<i>Karyawan</i> dapat melakukan pemesanan ruang rapat
KF-14	<i>Karyawan</i> dapat mengubah atau membatalkan pemesanan yang masih aktif
KF-15	<i>Karyawan</i> dapat melihat riwayat pemesanan pribadi
KF-16	<i>Karyawan</i> dapat memberikan umpan balik setelah rapat
KF-17	<i>Karyawan</i> dapat melihat notifikasi email yang diterima
KF-18	<i>Semua pengguna</i> dapat mengakses halaman <i>logout</i>

3.1.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional diberikan dengan bentuk kode fungsi, Kode KNF untuk Kebutuhan Non Fungsional.

TABEL II

TABEL KEBUTUHAN NON FUNGSIONAL

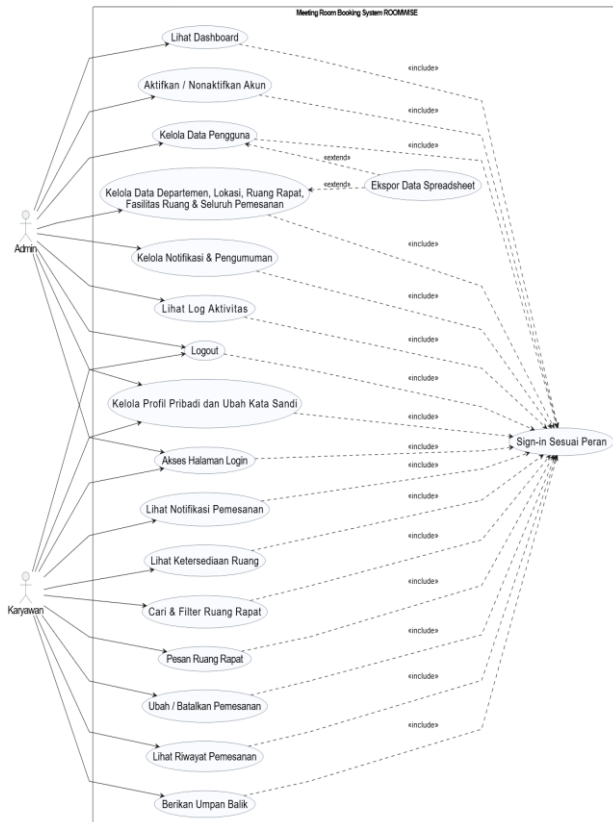
Kode	Kebutuhan Non Fungsional
KNF-01	Aplikasi dapat dijalankan pada <i>web browser</i> seperti <i>Google Chrome</i> , <i>Mozilla Firefox</i> , dan <i>Internet Explorer</i> .
KNF-02	Bahasa yang digunakan pada aplikasi Bahasa Indonesia.
KNF-03	Aplikasi memiliki tampilan (<i>UI/UX</i>) yang mudah dipahami dan <i>responsive</i> .
KNF-04	Aplikasi harus dapat diakses 24/7 dengan <i>downtime</i> minimal.
KNF-05	Aplikasi harus bisa <i>keep up-to-date</i> semua perubahan yang telah ditetapkan.
KNF-06	<i>Admin</i> diwajibkan melakukan autentikasi sebelum melakukan manajemen konten.

3.2 Perancangan Cepat

Pada tahap perancangan cepat, dibuat rancangan awal sistem yang dapat digunakan untuk memberikan gambaran kepada pengguna mengenai sistem yang akan dikembangkan [2][7]. Disini dihasilkan *wireframe* antarmuka, ERD dengan enam tabel utama termasuk *users*, *bookings*, *rooms*, dan *feedbacks*, serta *use case diagram* yang menggambarkan interaksi admin dan karyawan untuk memastikan desain awal mencakup fitur pencarian ruang berdasarkan lokasi, kapasitas, dan fasilitas

3.2.1 Use Case Diagram

Use Case Diagram merupakan salah satu diagram UML yang digunakan untuk memodelkan kebutuhan fungsional sistem berdasarkan interaksi antara aktor dan sistem. Gambar 3 menunjukkan bahwa sistem *RoomWise* melibatkan dua aktor utama, yaitu *Admin* dan *Karyawan*. *Admin* berperan dalam pengelolaan data master, meliputi data pengguna, ruang rapat, fasilitas ruang rapat, notifikasi, serta pengumuman. Selain itu, *Admin* juga dapat memantau aktivitas sistem melalui dashboard dan log aktivitas. Di sisi lain, *Karyawan* berinteraksi dengan sistem untuk melakukan pemesanan ruang rapat, melihat ketersediaan ruang, mengelola jadwal pemesanan, serta memberikan umpan balik terhadap layanan yang tersedia. Setiap aktor hanya dapat mengakses fitur sesuai dengan perannya setelah berhasil melakukan proses autentikasi.

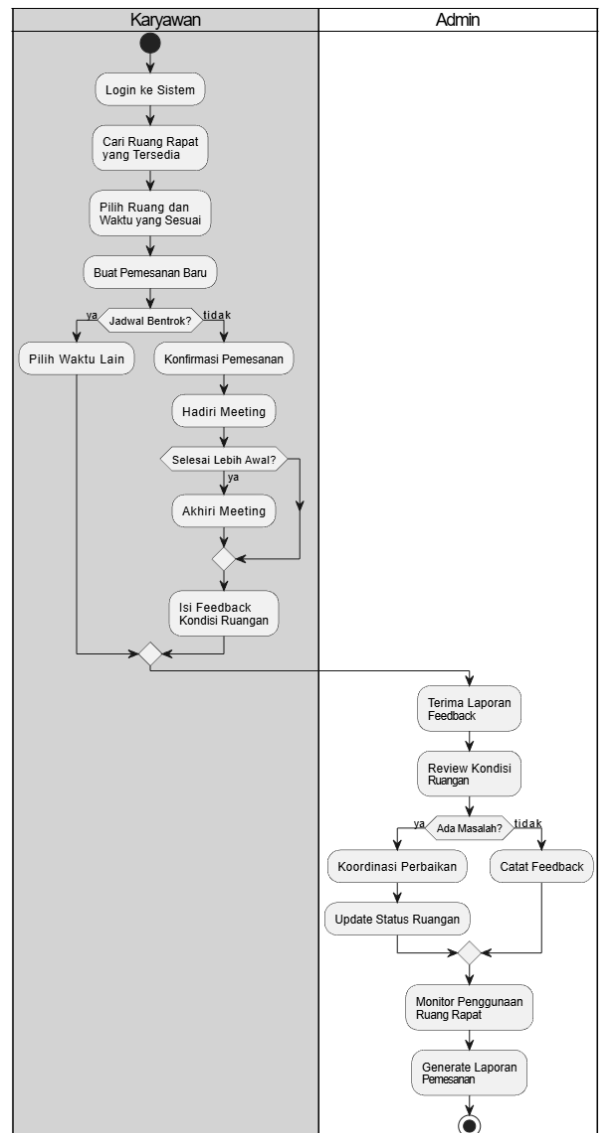


Gambar 3. Use Case Diagram Sistem

3.2.2 Activity Diagram

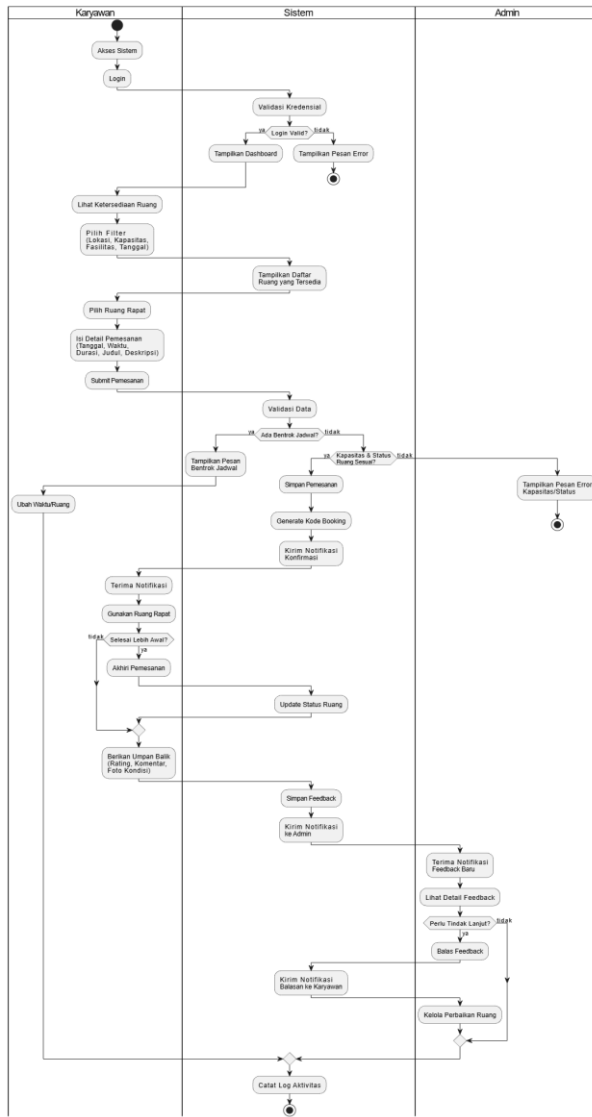
Activity diagram adalah diagram yang menggambarkan runtutan proses yang terjadi pada sebuah sistem. *Activity diagram* bisnis berfokus kepada alur ataupun proses bisnis atau interaksi antara admin dan karyawan.

Alur dimulai ketika *karyawan* melakukan login, mencari ketersediaan ruang, memilih ruang dan waktu, lalu membuat pemesanan. Jika terjadi bentrok jadwal, *karyawan* diarahkan untuk memilih waktu lain, sedangkan jika tidak ada konflik maka pemesanan dikonfirmasi dan rapat dilaksanakan. Setelah rapat selesai, baik sesuai jadwal maupun lebih awal, *karyawan* memberikan *feedback* terkait kondisi ruangan. *Admin* kemudian menerima laporan *feedback*, melakukan review, menentukan apakah ada masalah, serta menindaklanjuti melalui koordinasi perbaikan dan pembaruan status ruang. Seluruh proses ini berakhir pada aktivitas monitoring penggunaan ruang dan pembuatan laporan pemesanan sebagai bahan evaluasi manajemen.



Gambar 4. Activity Diagram Bisnis.

Setelah membahas mengenai alur bisnis, berikutnya ialah cara kerja sistem. Pada *activity diagram* cara kerja ini, Proses dimulai dari akses dan login *karyawan* yang divalidasi oleh *sistem*, kemudian dilanjutkan dengan penampilan *dashboard* atau pesan kesalahan jika kredensial tidak valid. *Karyawan* melakukan pencarian dan pemesanan ruang dengan mengisi detail, sementara *sistem* memvalidasi data, mengecek bentrok jadwal, kapasitas, serta status ruang sebelum menyimpan pemesanan, menghasilkan kode *booking*, dan mengirim notifikasi konfirmasi. Setelah ruang digunakan, *sistem* memperbarui status ruangan dan menyimpan *feedback* yang dikirim *karyawan*, lalu meneruskan notifikasi ke *admin*. *Admin* meninjau *feedback*, menentukan perlu tidaknya tindak lanjut, mengelola perbaikan ruang, serta memberikan balasan yang dikirim kembali ke *karyawan*, dan seluruh aktivitas akhirnya dicatat dalam *log* sebagai jejak operasional sistem.

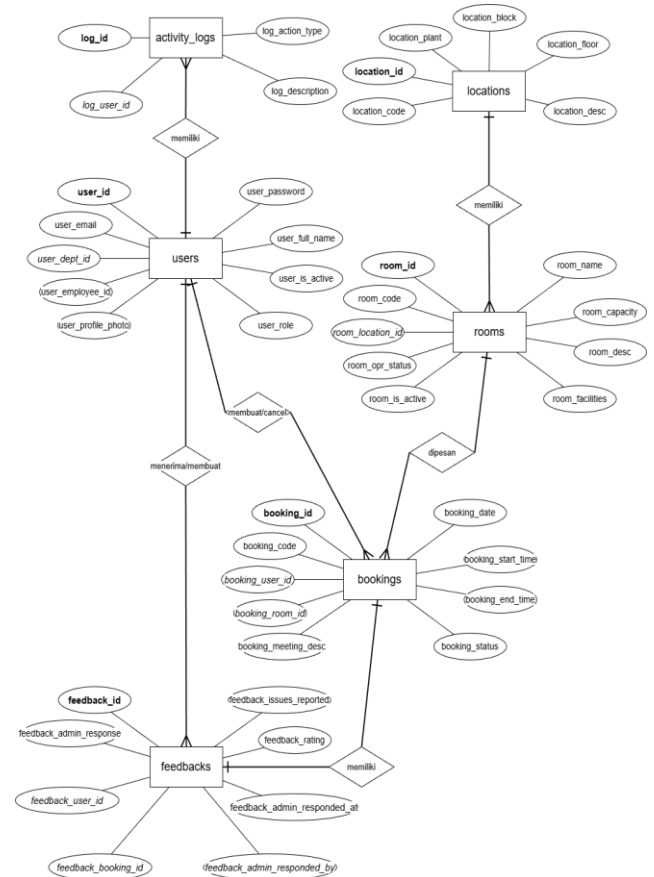


Gambar 5. Activity Diagram Cara Kerja.

3.2.3 Entity Relationship Diagram

Diagram ERD (*Entity Relationship Diagram*) menggambarkan sistem manajemen CDC yang terstruktur pada Gambar 8. Database RoomWise dirancang dengan enam tabel utama yang saling berelasi untuk mendukung sistem manajemen ruang rapat. Tabel *users* yang menyimpan data pengguna (*Admin* dan *Employee*) dengan *password* ter-hash MD5, sedangkan tabel *locations* berelasi dengan *rooms* yang menyimpan informasi ruang rapat beserta fasilitas dan status operasional. Tabel *bookings* menjadi inti sistem reservasi yang berelasi dengan *users* sebagai pembuat reservasi dan *rooms* sebagai ruangan yang dipesan, dengan status mencakup *Pending*, *Confirmed*, *In Progress*, *Completed*, dan *Cancelled*. Tabel *feedbacks* memiliki relasi *one-to-one* dengan *bookings* dan berelasi dengan *users* untuk mencatat pemberi *feedback* serta *admin* yang merespons. Tabel *activity_logs* berfungsi sebagai *audit trail* yang mencatat seluruh aktivitas pengguna dalam

sistem termasuk operasi *CRUD* dengan menyimpan nilai lama dan baru dalam format JSON, sehingga keseluruhan struktur *database* membentuk sistem terintegrasi untuk manajemen ruang rapat yang efisien dan teraudit.



Gambar 6. ERD sistem.

3.3 Membuat Prototype

Fase pembuatan *prototype* dilakukan dengan mengimplementasikan hasil dari perancangan cepat ke dalam bentuk aplikasi awal (*working prototype*) menggunakan teknologi **ASP.NET berbasis MVC**. *Prototype* ini dikembangkan dengan fokus pada fitur inti sistem *RoomWise*, yaitu halaman *login*, dashboard pengguna, manajemen data ruang rapat, serta fitur pemesanan (*booking*) ruang. Pada tahap ini, antarmuka pengguna (UI) dibuat berdasarkan *wireframe* yang telah dirancang sebelumnya. *Backend* sederhana juga mulai dibangun untuk mendukung proses input dan penyimpanan data menggunakan database **SQL Server Express 2025**.

Prototype yang dihasilkan belum mencakup seluruh fitur secara lengkap, namun sudah mampu menggambarkan alur utama sistem seperti proses *login*, melihat daftar ruang, melakukan pemesanan, serta menampilkan data *booking*. Tujuan utama dari *prototype* ini adalah memberikan gambaran nyata kepada pengguna agar dapat dilakukan evaluasi awal terhadap kesesuaian sistem dengan kebutuhan.

3.4 Evaluasi Atas Prototype

Fase evaluasi *prototype* dilakukan dengan melibatkan calon pengguna, yaitu admin dan karyawan, untuk menguji secara langsung *prototype* yang telah dibuat. Pengujian dilakukan dengan cara mencoba fitur-fitur utama seperti *login*, pencarian ruang, proses *booking*, serta tampilan *dashboard*.

Evaluasi atas *prototype* dilakukan secara bertahap melalui empat iterasi pengujian. Setiap iterasi bertujuan untuk memperoleh masukan dari pengguna dan pihak terkait agar *prototype* yang dibuat dapat disesuaikan dengan kebutuhan sistem. Pihak yang terlibat dalam evaluasi *prototype* adalah Pak Agung selaku mentor, Pak Fikry selaku pemesan di sistem, dan Ihsan selaku penulis sekaligus *admin* sistem.

Pada iterasi pertama, pengujian dilakukan oleh Ihsan sebagai admin untuk memastikan fungsi dasar sistem dapat berjalan, seperti proses *login*, pengelolaan data, dan alur penggunaan awal. Dari hasil pengujian ini, ditemukan beberapa bagian tampilan dan alur penggunaan yang masih perlu disesuaikan agar lebih mudah digunakan.

Pada iterasi kedua, pengujian dilakukan bersama Pak Agung selaku mentor. Evaluasi pada tahap ini berfokus pada kesesuaian alur sistem dengan metode pengembangan *prototype* serta kelengkapan fitur yang dibutuhkan. Masukan yang diberikan digunakan untuk memperbaiki struktur menu, tampilan halaman, dan kejelasan proses pada sistem.

Pada iterasi ketiga, pengujian dilakukan bersama Pak Fikry selaku pemesan sistem. Pengujian ini berfokus pada kesesuaian *prototype* dengan kebutuhan pengguna akhir. Dari evaluasi tersebut, terdapat beberapa masukan terkait kemudahan penggunaan, penyesuaian fitur, serta informasi yang perlu ditampilkan agar sistem lebih sesuai dengan kebutuhan operasional.

Pada iterasi keempat, pengujian dilakukan kembali oleh Ihsan sebagai admin setelah perbaikan dari iterasi sebelumnya diterapkan. Pengujian ini bertujuan memastikan bahwa fitur-fitur yang telah diperbaiki dapat berjalan dengan baik dan tidak menimbulkan kendala baru pada sistem.

Pengujian akhir dilakukan bersama Pak Agung selaku mentor, Pak Fikry selaku pemesan, dan Ihsan selaku *admin*. Berdasarkan hasil pengujian akhir, *prototype* dinilai telah sesuai dengan kebutuhan yang telah ditentukan. Oleh karena itu, *prototype* dapat dilanjutkan ke tahap pengembangan, pengujian produk, dan penyelesaian final produk.

Berdasarkan hasil evaluasi, diperoleh beberapa umpan balik dari pengguna terkait aspek fungsional maupun non-fungsional. Dari sisi fungsional, pengguna memberikan masukan terkait perlunya

penambahan fitur seperti filter pencarian ruang berdasarkan kapasitas dan fasilitas, serta notifikasi status pemesanan. Dari sisi non-fungsional, pengguna menginginkan tampilan antarmuka yang lebih intuitif, navigasi yang lebih sederhana, serta peningkatan kecepatan akses sistem. Evaluasi ini bertujuan untuk mengidentifikasi kekurangan dan ketidaksesuaian antara *prototype* dengan kebutuhan pengguna, sehingga dapat dilakukan perbaikan sebelum masuk ke tahap pengembangan lebih lanjut.

3.5 Perubahan Prototype

Berdasarkan hasil evaluasi yang telah dilakukan, dilakukan perbaikan dan penyempurnaan terhadap *prototype*. Perubahan yang dilakukan meliputi pengembangan fitur tambahan seperti filter pencarian ruang berdasarkan kapasitas, lokasi, dan fasilitas, serta penambahan sistem notifikasi untuk memberikan informasi status pemesanan kepada pengguna. Selain itu, dilakukan perbaikan pada tampilan antarmuka dengan menyederhanakan navigasi menu, memperbaiki layout halaman, serta meningkatkan konsistensi desain agar lebih *user-friendly*. Dari sisi sistem, dilakukan optimalisasi proses pengolahan data untuk meningkatkan performa dan responsivitas aplikasi.

Prototype yang telah diperbaiki ini kemudian menjadi dasar untuk masuk ke tahap pengembangan sistem secara penuh. Dengan adanya iterasi ini, diharapkan sistem *RoomWise* yang dikembangkan dapat lebih sesuai dengan kebutuhan pengguna dan meminimalkan kesalahan pada tahap implementasi terakhir.

3.6 Pengembangan

Fase pengembangan membangun sistem lengkap *RoomWise* dengan implementasi penuh *backend* ASP.NET terintegrasi SQL Server Express 2025 via SSMS, pengkodean fitur lengkap seperti manajemen lokasi, *feedback* pasca-rapat, dan *export* data, serta pengujian internal untuk memastikan modularitas MVC dan *SEO-friendly* struktur, menghasilkan aplikasi web yang siap produksi dengan *black-box* testing awal. Sepenuhnya bisa dilihat pada bagian Bab IV.

3.7 Testing

Pada tahap *testing*, *prototype* diuji untuk memastikan setiap fungsi sistem telah berjalan sesuai dengan kebutuhan yang ditetapkan. Pengujian dilakukan menggunakan metode *black-box testing*, yaitu pengujian yang berfokus pada kesesuaian keluaran sistem berdasarkan masukan yang diberikan tanpa memeriksa kode program secara langsung [1]. Pada *RoomWise*, pengujian dilakukan terhadap 18 KF dan 6 KNF, semua bertotal 24.

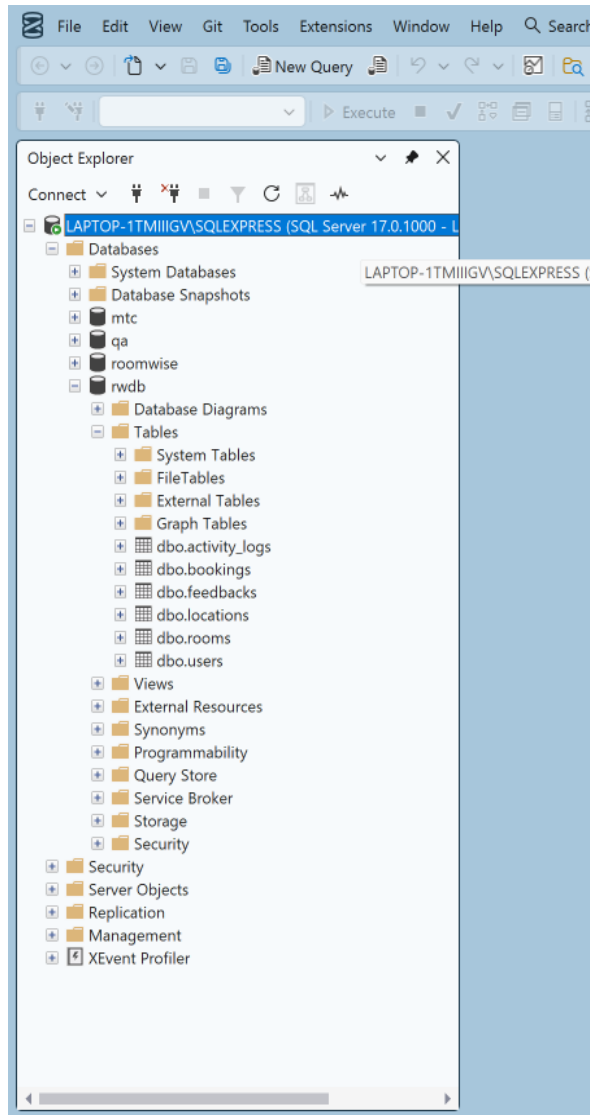
3.8 Final Produk

Pada tahap *final product*, *prototype* yang telah diuji, dievaluasi, dan diperbaiki ditetapkan menjadi produk akhir yang siap digunakan [2][7]. Pada penelitian ini, hasil akhir berupa aplikasi *RoomWise* yang telah memenuhi kebutuhan fungsional dan nonfungsional serta dapat digunakan untuk mendukung pengelolaan ruang rapat di PT XYZ Batam.

4. Hasil dan Pembahasan

4.1 Hasil Implementasi Basis Data

Hasil implementasi untuk struktur *database* *Microsoft SQL Server* sistem yang telah dibuat, seperti ditunjukkan pada Gambar 7, terdapat 6 tabel dan sistem terintegrasi dengan baik sehingga semua fungsional dapat berhasil sesuai rancangan.



Gambar 7. Implementasi Basis Data di SQL Server.

4.2 Hasil Implementasi Fungsional

4.2.1 Hasil Pengujian KF-01 (Semua pengguna dapat mengakses halaman login).

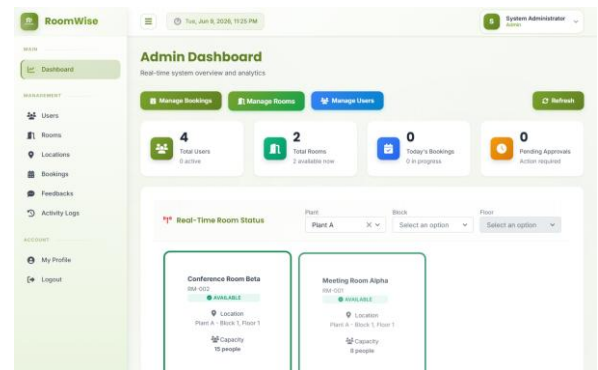
Halaman *Login* guna pengguna melakukan autentikasi sebelum memasuki *website* dan mengakses berbagai macam fitur yang tersedia.



Gambar 8. Halaman Login.

4.2.2 Hasil Pengujian KF-02 (Semua pengguna dapat melakukan sign-in sesuai peran).

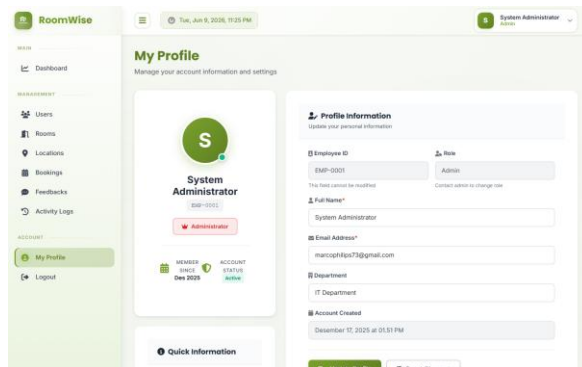
Hasil implementasi proses *sign-in* pengguna sesuai dengan perannya. Pengguna memasukkan alamat *email* dan kata sandi pada halaman *login*. Sistem kemudian melakukan validasi terhadap data akun yang dimasukkan. Apabila data akun valid, sistem mengarahkan pengguna ke halaman utama sesuai dengan perannya, yaitu admin atau karyawan.



Gambar 9. Hasil Sign-In Pengguna Sesuai Peran.

4.2.3 Hasil Pengujian KF-03 (Semua Pengguna dapat mengelola profil pribadi dan mengubah kata sandi).

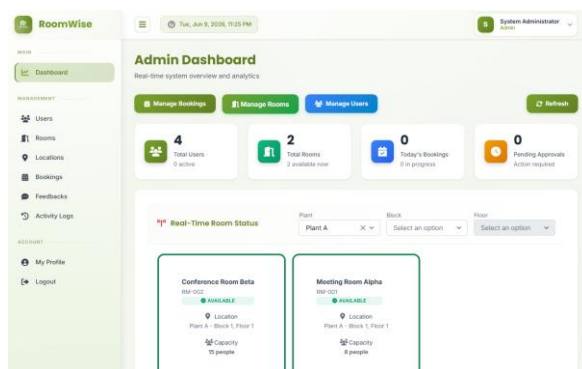
Halaman pengelolaan profil pengguna. Pada halaman ini, pengguna dapat melihat dan memperbarui informasi pribadi serta melakukan perubahan kata sandi. Sistem melakukan validasi terhadap data yang dimasukkan sebelum menyimpan perubahan ke dalam basis data.



Gambar 10. Halaman Pengelolaan Profil Pengguna.

4.2.4 Hasil Pengujian KF-04 (Admin dapat melihat halaman dashboard).

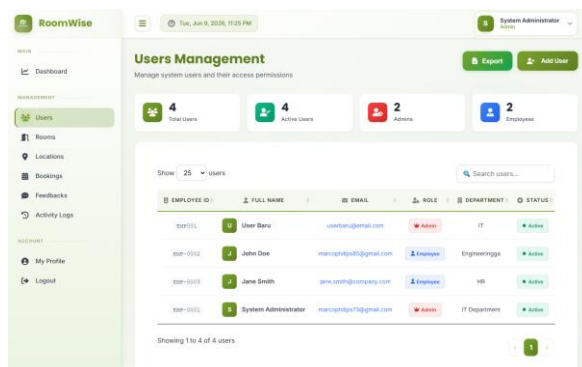
Halaman *dashboard* admin yang ditampilkan setelah admin berhasil melakukan sign-in. Halaman tersebut menyajikan ringkasan informasi sistem, seperti jumlah pengguna, ruang rapat, pemesanan, dan aktivitas terbaru. Informasi tersebut membantu admin dalam memantau penggunaan sistem *RoomWise*.



Gambar 11. Halaman Dashboard Admin.

4.2.5 Hasil Pengujian KF-05 (Admin dapat mengelola data pengguna).

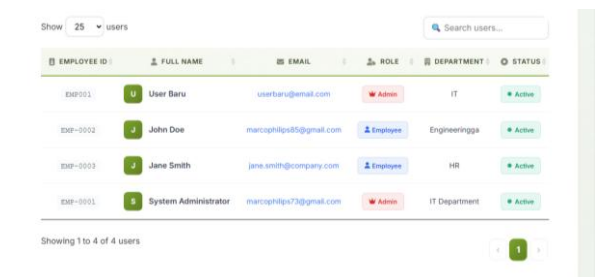
Halaman pengelolaan data pengguna. Admin dapat menambahkan pengguna baru, melihat informasi pengguna, serta memperbarui data pengguna yang telah tersimpan. Setiap perubahan yang dilakukan oleh admin akan disimpan ke dalam basis data sistem.



Gambar 12. Halaman Pengelolaan Data Pengguna.

4.2.6 Hasil Pengujian KF-06 (Admin dapat mengaktifkan dan menonaktifkan akun pengguna).

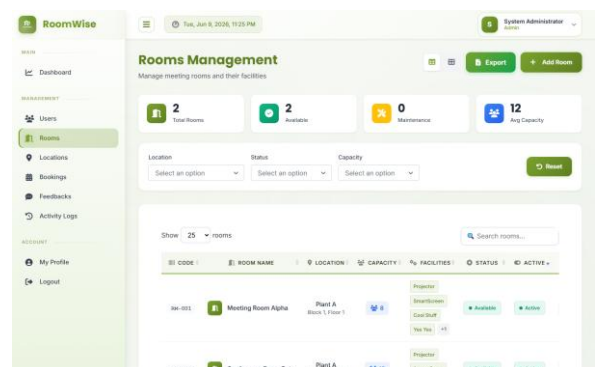
Fitur aktivasi dan nonaktivasi akun pengguna. Admin dapat mengubah status akun pengguna menjadi aktif atau tidak aktif. Pengguna dengan akun yang tidak aktif tidak dapat melakukan sign-in ke dalam sistem *RoomWise*.



Gambar 13. Halaman Aktivasi dan Nonaktivasi Akun Pengguna.

4.2.7 Hasil Pengujian KF-07 (Admin dapat mengelola data master ruang rapat, meliputi data lokasi, ruang rapat, fasilitas ruang rapat, serta seluruh data pemesanan ruang rapat).

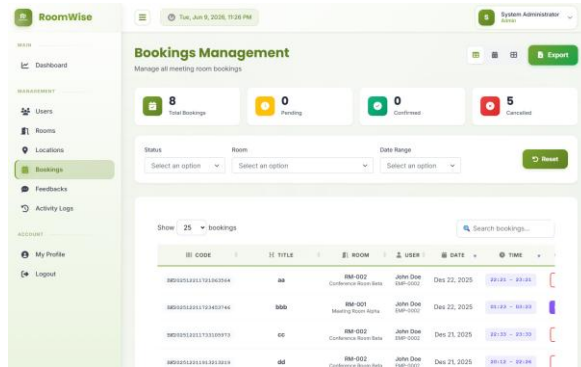
Halaman pengelolaan data master dan pemesanan. Admin dapat mengelola data lokasi, ruang rapat, fasilitas ruang rapat, serta seluruh data pemesanan. Admin dapat menambahkan, memperbarui, mengaktifkan, atau menonaktifkan data sesuai dengan kebutuhan operasional perusahaan.



Gambar 14. Halaman Pengelolaan Rooms Master.

4.2.8 Hasil Pengujian KF-08 (Admin dapat mengirimkan notifikasi email otomatis).

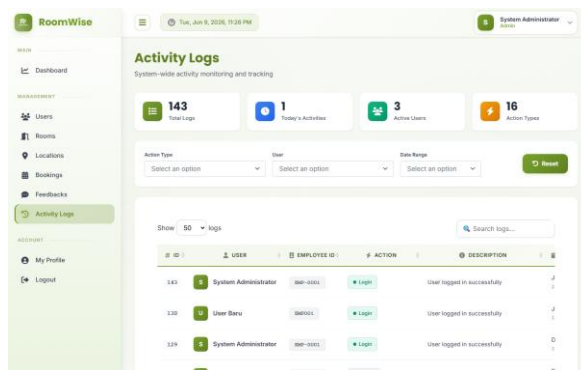
Hasil implementasi notifikasi *email* otomatis. Sistem mengirimkan *email* kepada pengguna ketika terjadi perubahan pada pemesanan, seperti pemesanan berhasil dibuat, disetujui, diperbarui, atau dibatalkan. Notifikasi tersebut membantu pengguna memperoleh informasi terbaru mengenai pemesanannya.



Gambar 15. Halaman Booking Letak Fitur Email Otomatis.

4.2.9 Hasil Pengujian KF-09 (Admin dapat melihat log aktivitas sistem).

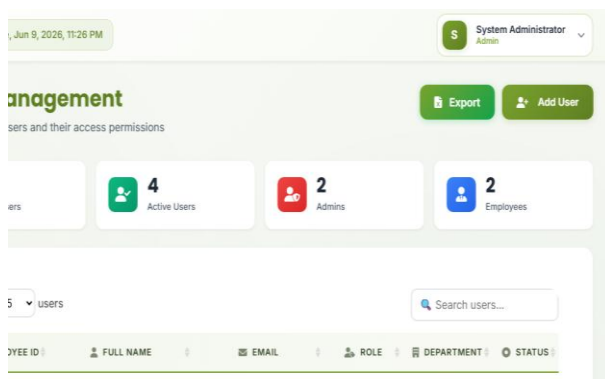
Halaman log aktivitas sistem. Halaman tersebut mencatat berbagai tindakan pengguna, seperti melakukan *sign-in*, membuat pemesanan, memperbarui data, dan melakukan ekspor data. Admin dapat menggunakan informasi tersebut untuk memantau aktivitas yang terjadi di dalam sistem.



Gambar 16. Halaman Log Aktivitas Sistem.

4.2.10 Hasil Pengujian KF-10 (Admin dapat mengekspor data dalam format Spreadsheet).

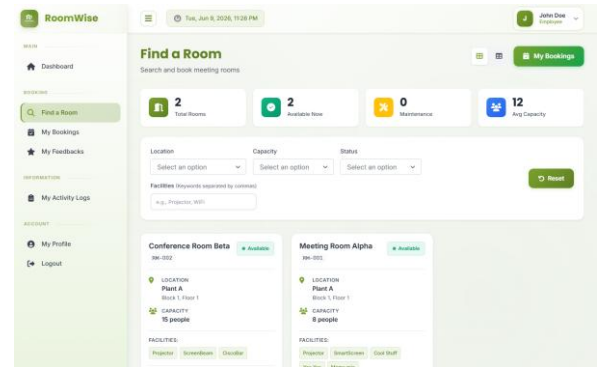
Hasil implementasi fitur ekspor data. Admin dapat mengekspor data pengguna, ruang rapat, atau pemesanan ke dalam format *spreadsheet*. File yang dihasilkan dapat digunakan untuk kebutuhan dokumentasi dan pelaporan.



Gambar 17. Halaman Users Letak Salah Satu Fitur Ekspor.

4.2.11 Hasil Pengujian KF-11 (Karyawan dapat melihat ketersediaan ruang rapat).

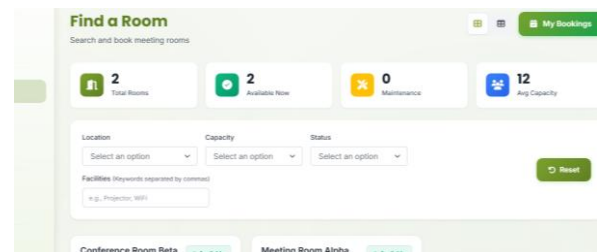
Halaman ketersediaan ruang rapat. Karyawan dapat melihat informasi ruang rapat, seperti nama ruang, lokasi, kapasitas, fasilitas, serta status ketersediaannya. Informasi tersebut membantu karyawan menentukan ruang yang dapat digunakan.



Gambar 18. Halaman Ketersediaan Ruang Rapat.

4.2.12 Hasil Pengujian KF-12 (Karyawan dapat mencari dan memfilter data ruang rapat berdasarkan data yang ada).

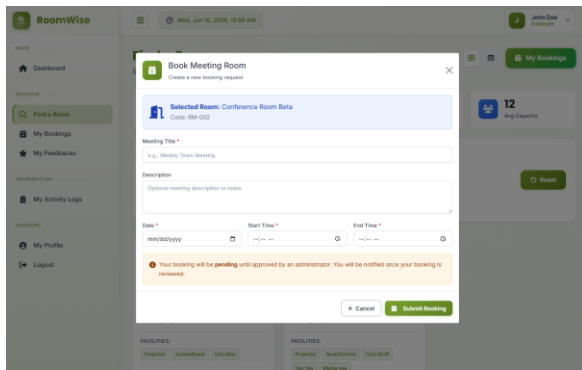
Fitur pencarian dan filter ruang rapat. Karyawan dapat mencari ruang berdasarkan lokasi, kapasitas, fasilitas, tanggal, dan waktu penggunaan. Sistem kemudian menampilkan daftar ruang yang sesuai dengan kriteria pencarian pengguna.



Gambar 19. Find a Room Juga Memiliki Filter.

4.2.13 Hasil Pengujian KF-13 (Karyawan dapat melakukan pemesanan ruang rapat).

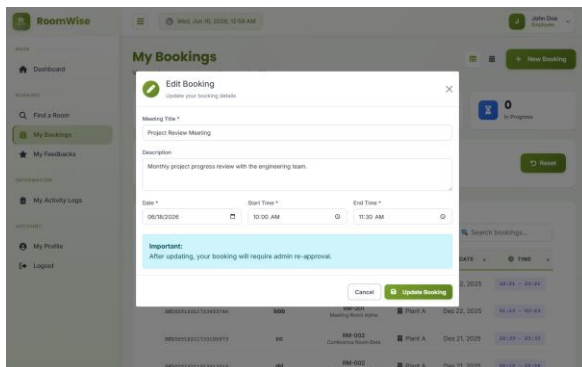
Halaman pemesanan ruang rapat. Karyawan dapat memilih ruang dan memasukkan informasi rapat, seperti judul, deskripsi, tanggal, waktu mulai, dan waktu selesai. Sebelum menyimpan pemesanan, sistem memeriksa ketersediaan ruang untuk mencegah terjadinya bentrok jadwal.



Gambar 20. Halaman Pemesanan Ruang Rapat.

4.2.14 Hasil Pengujian KF-14 (Karyawan dapat mengubah atau membatalkan pemesanan yang masih aktif).

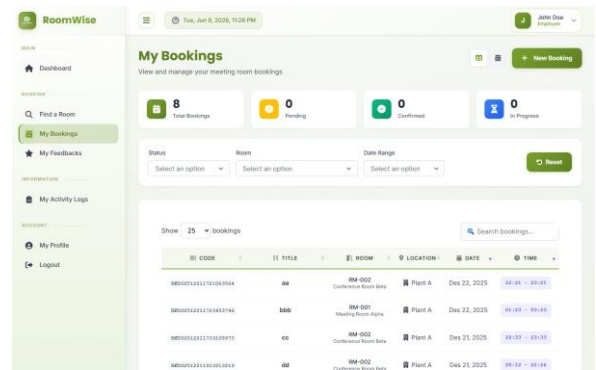
Fitur perubahan dan pembatalan pemesanan. Karyawan dapat memperbarui informasi pemesanan yang masih aktif atau membatalkan pemesanan dengan memberikan alasan pembatalan. Sistem akan memperbarui status dan informasi pemesanan sesuai dengan tindakan pengguna.



Gambar 21. Halaman Perubahan dan Pembatalan Pemesanan.

4.2.15 Hasil Pengujian KF-15 (Karyawan dapat melihat riwayat pemesanan pribadi).

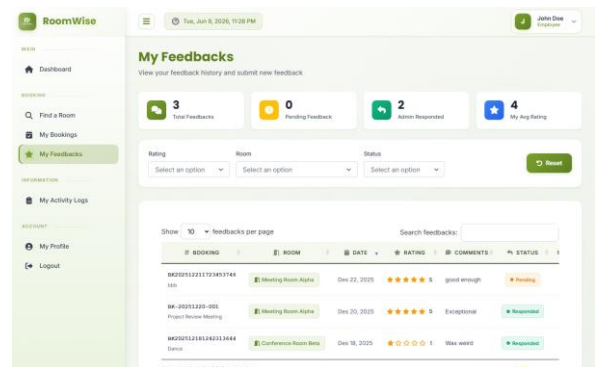
Halaman riwayat pemesanan pribadi. Halaman tersebut menampilkan seluruh pemesanan yang dibuat oleh pengguna beserta statusnya, seperti menunggu persetujuan, disetujui, selesai, atau dibatalkan. Data yang ditampilkan hanya berasal dari akun pengguna yang sedang aktif.



Gambar 22. Halaman Riwayat Pemesanan Pribadi.

4.2.16 Hasil Pengujian KF-16 (Karyawan dapat memberikan umpan balik setelah rapat).

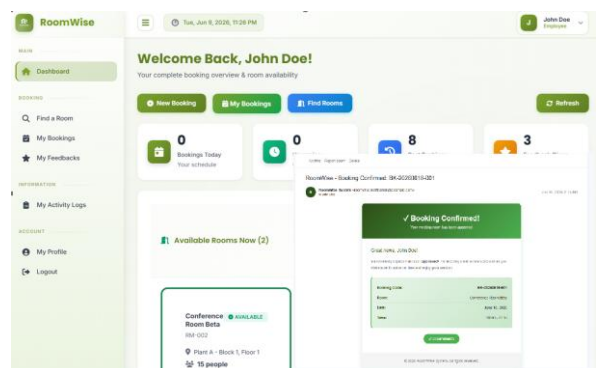
Halaman pemberian umpan balik setelah rapat. Karyawan dapat memberikan penilaian dan komentar mengenai kondisi ruang rapat yang telah digunakan. Sistem menyimpan umpan balik tersebut sebagai bahan evaluasi bagi pengelola ruang rapat.



Gambar 23. Halaman Pemberian Umpan Balik.

4.2.17 Hasil Pengujian KF-17 (Karyawan dapat melihat notifikasi email yang diterima).

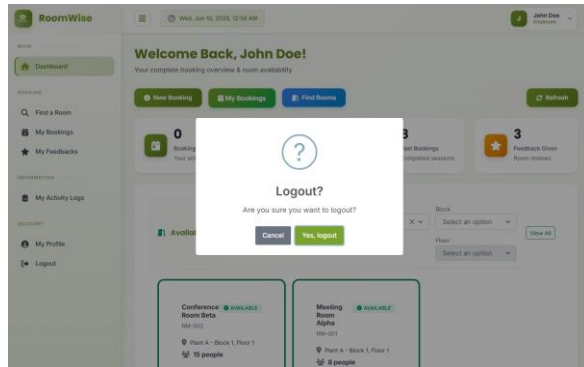
Notifikasi email yang diterima oleh karyawan. Email tersebut berisi informasi mengenai aktivitas pemesanan, seperti pemesanan berhasil dibuat, disetujui, diperbarui, atau dibatalkan. Notifikasi membantu karyawan memantau perkembangan pemesanan tanpa harus membuka sistem secara berulang.



Gambar 24. Notifikasi Email yang Diterima Karyawan.

4.2.18 Hasil Pengujian KF-18 (Semua pengguna dapat mengakses halaman logout).

Hasil implementasi proses *logout*. Ketika pengguna memilih menu *logout*, sistem menghapus sesi pengguna yang sedang aktif dan mengarahkan pengguna kembali ke halaman *login*.



Gambar 25. Hasil Proses Logout Pengguna.

4.3 Hasil Pengujian Sistem

Pengujian sistem menggu metode *black box*. Uji dilakukan oleh mentor atau atasan semasa bekerja. Pengujian yang dilakukan adalah menguji seluruh fungsional dan non fungsional pada Tabel I dan Tabel II telah sesuai dengan rancangan atau tidaknya akan ditentukan dengan keberhasilan uji. Lihat Tabel III.

TABEL III

TABEL PENGUJIAN BLACK-BOX TESTING

No. Uji	Aktor	Deskripsi Skenario	Hasil yang Diharapkan	Hasil Uji
KU-01	Semua Pengguna	Pengguna mengakses halaman <i>login</i> sistem	Berhasil melakukan <i>login</i>	Berhasil
		Pengguna memasukkan <i>email</i> yang tidak terdaftar pada system atau <i>password</i> salah	Sistem menolak <i>login</i> dan menampilkan pesan “Invalid email or password”	Berhasil
KU-02	Semua Pengguna	Pengguna melakukan <i>sign-in</i> sesuai peran masing-masing.	Berhasil masuk dengan peran yang sesuai	Berhasil
		Pengguna <i>Employee</i> mencoba mengakses fungsi <i>Admin</i> secara langsung melalui <i>URL/action</i>	Sistem menolak akses dan mengarahkan pengguna ke halaman <i>login</i> atau memberi pesan <i>Unauthorized</i>	Berhasil
KU-03	Semua Pengguna	Pengguna mengelola profil pribadi dan mengubah kata sandi	Berhasil memperbarui profil dan kata sandi	Berhasil

No. Uji	Aktor	Deskripsi Skenario	Hasil yang Diharapkan	Hasil Uji
KU-04	Admin	<i>Admin</i> mengakses halaman dashboard	Berhasil mengakses <i>dashboard</i>	Berhasil
KU-05	Admin	<i>Admin</i> mengelola data pengguna (tambah, ubah, hapus)	Berhasil menambah, mengubah, atau menghapus data pengguna	Berhasil
KU-06	Admin	<i>Admin</i> mengaktifkan dan menonaktifkan akun pengguna	Berhasil mengaktifkan atau menonaktifkan akun pengguna	Berhasil
KU-07	Admin	<i>Admin</i> dapat mengelola data <i>master</i> , ruang rapat, meliputi data lokasi, ruang rapat, fasilitas ruang rapat, serta seluruh data pemesanan ruang rapat	Admin berhasil mengelola (menambah, mengubah, atau menghapus) <i>master</i> data	Berhasil
KU-08	Admin	<i>Admin</i> mengirimkan notifikasi <i>email</i> otomatis	Berhasil mengirimkan notifikasi <i>email</i> otomatis	Berhasil
KU-09	Admin	<i>Admin</i> melihat log aktivitas sistem	Berhasil melihat log	Berhasil
KU-10	Admin	<i>Admin</i> mengekspor data ke format <i>spreadsheet</i>	Berhasil mengekspor data	Berhasil
KU-11	Karyawan	Karyawan melihat ketersediaan ruang rapat	Berhasil melihat ketersediaan ruang rapat	Berhasil
KU-12	Karyawan	Karyawan mencari dan memfilter data ruang rapat	Berhasil memfilter data ruang rapat	Berhasil
KU-13	Karyawan	Karyawan melakukan pemesanan ruang rapat	Berhasil melakukan pemesanan	Berhasil
		Karyawan melakukan <i>booking</i> pada jadwal yang sudah digunakan oleh <i>booking</i> lain	Sistem menolak pemesanan karena ruangan tidak tersedia pada slot waktu tersebut	Berhasil
		Karyawan mengisi jam selesai lebih awal atau sama dengan jam mulai	Sistem menolak input waktu karena tidak valid	Berhasil
		Karyawan mencoba melakukan booking untuk tanggal yang sudah lewat	Sistem menolak pemesanan untuk tanggal lampau	Berhasil

No. Uji	Aktor	Deskripsi Skenario	Hasil yang Diharapkan	Hasil Uji
		Karyawan mencoba memesan ruangan yang status operasionalnya tidak tersedia	Sistem menolak pemesanan karena status ruang tidak <i>Available</i>	Berhasil
KU-14	Karyawan	Karyawan mengubah atau membatalkan pemesanan aktif	Berhasil mengubah atau membatalkan pesanan	Berhasil
		Karyawan mencoba mengubah pemesanan milik pengguna lain	Sistem menolak perubahan karena pengguna hanya boleh mengubah booking miliknya sendiri	Berhasil
		Karyawan mencoba mengubah <i>booking</i> yang sudah <i>Completed</i> atau <i>Cancelled</i>	Sistem menolak perubahan karena <i>booking</i> sudah tidak aktif	Berhasil
		Karyawan mencoba membatalkan <i>booking</i> milik pengguna lain	Sistem menolak pembatalan	Berhasil
KU-15	Karyawan	Karyawan melihat riwayat pemesanan pribadi	Berhasil melihat riwayat pemesanan	Berhasil
KU-16	Karyawan	Karyawan memberikan umpan balik setelah rapat.	Berhasil memberikan <i>feedback</i>	Berhasil
		Karyawan mencoba memberi <i>feedback</i> pada <i>booking</i> yang belum selesai	Sistem menolak <i>feedback</i> karena <i>feedback</i> hanya untuk <i>booking Completed</i>	Berhasil
		Karyawan mengirim <i>feedback</i> lebih dari satu kali untuk <i>booking</i> yang sama	Sistem menolak <i>feedback</i> ganda	Berhasil
KU-17	Karyawan	Karyawan melihat notifikasi <i>email</i> yang diterima	Berhasil melihat notifikasi <i>email</i>	Berhasil
KU-18	Semua Pengguna	Pengguna melakukan <i>logout</i> dari sistem	Berhasil melakukan <i>logout</i>	Berhasil
KU-19	Semua Pengguna (KNF)	Membuka aplikasi <i>RoomWise</i> melalui <i>web browser</i> dan mengakses semua halaman	Aplikasi dapat dibuka dan fitur utama dapat digunakan melalui <i>browser</i>	Berhasil

No. Uji	Aktor	Deskripsi Skenario	Hasil yang Diharapkan	Hasil Uji
KU-20	Semua Pengguna (KNF)	Memeriksa teks pada halaman <i>login</i> , <i>dashboard</i> , menu, tombol, <i>form</i> , dan pesan sistem	Seluruh informasi utama pada aplikasi menggunakan Bahasa Indonesia yang mudah dipahami pengguna	Berhasil
KU-21	Semua Pengguna (KNF)	Mengakses aplikasi melalui tampilan <i>desktop</i> dan ukuran layar lebih kecil, lalu mencoba navigasi menu dan <i>form booking</i>	Tampilan tetap rapi, menu mudah digunakan, dan form dapat diisi tanpa kebingungan pengguna	Berhasil
KU-22	Semua Pengguna (KNF)	Menjalankan aplikasi melalui <i>server</i> lokal dan mengakses beberapa halaman utama secara berulang	Aplikasi tetap dapat diakses selama <i>server</i> dan database berjalan	Berhasil
KU-23	Semua Pengguna (KNF)	Melakukan perubahan data seperti menambah atau mengubah data ruangan, mengubah status <i>booking</i> , dan membatalkan pemesanan	Perubahan data tersimpan di database dan langsung ditampilkan pada halaman terkait	Berhasil
KU-24	Semua Pengguna (KNF)	Mencoba mengakses halaman manajemen admin tanpa <i>login</i> sebagai Admin	Sistem menolak akses dan mengarahkan pengguna ke halaman <i>login</i>	Berhasil

Pada bagian ini dijelaskan pembahasan hasil pengujian sistem yang telah dilakukan menggunakan metode *black-box testing*. Pengujian dilakukan untuk memastikan bahwa setiap fungsi pada sistem RoomWise berjalan sesuai dengan kebutuhan fungsional dan non fungsional yang telah dirancang. Berdasarkan hasil pengujian, sistem mampu menjalankan proses utama seperti autentikasi pengguna, pengelolaan data oleh admin, pemesanan ruang rapat oleh karyawan, pengelolaan riwayat pemesanan, pemberian umpan balik, notifikasi, serta proses keluar dari sistem. Selain itu, non fungsional seperti bahasa sistem ataupun responsifitas juga berjalan sesuai kebutuhan perusahaan.

Pada bagian ini dijelaskan pembahasan hasil pengujian sistem yang telah dilakukan menggunakan metode *black-box testing*. Pengujian dilakukan untuk memastikan bahwa setiap fungsi pada sistem RoomWise berjalan sesuai dengan kebutuhan fungsional dan non fungsional yang telah dirancang. Berdasarkan hasil pengujian, sistem mampu menjalankan proses utama seperti autentikasi pengguna, pengelolaan data oleh admin, pemesanan ruang rapat oleh karyawan, pengelolaan riwayat pemesanan, pemberian umpan balik, notifikasi, serta proses keluar dari sistem. Selain itu, non fungsional seperti bahasa sistem ataupun responsifitas juga berjalan sesuai kebutuhan perusahaan.

1) Implementasi akses halaman login

Proses akses halaman *login* dilakukan ketika pengguna membuka sistem RoomWise melalui peramban. Sistem akan menampilkan halaman *login* sebagai halaman awal sebelum pengguna dapat masuk ke menu utama. Berdasarkan hasil pengujian, halaman *login* dapat ditampilkan dengan baik dan pengguna dapat melihat formulir untuk memasukkan data akun. Hal ini menunjukkan bahwa sistem telah mampu menyediakan pintu masuk awal bagi seluruh pengguna sebelum proses autentikasi dilakukan.

2) Implementasi sign-in sesuai peran pengguna

Proses *sign-in* dilakukan ketika pengguna memasukkan data akun berupa alamat *email* dan kata sandi pada halaman *login*, kemudian menekan tombol masuk. Sistem akan membaca dan mencocokkan data tersebut dengan data pengguna yang tersimpan pada *database*. Jika data yang dimasukkan sesuai, maka sistem akan mengarahkan pengguna ke halaman utama sesuai dengan perannya, yaitu admin atau karyawan. Berdasarkan hasil pengujian, proses *sign-in* berjalan sesuai dengan hak akses pengguna sehingga penerapan *role-based access* pada sistem dapat berjalan dengan baik.

3) Implementasi pengelolaan profil pribadi dan perubahan kata sandi

Proses pengelolaan profil dilakukan setelah pengguna berhasil masuk ke dalam sistem. Pengguna dapat membuka halaman profil, melihat data pribadi, kemudian melakukan perubahan pada informasi tertentu seperti nama, atau alamat *email*. Setelah pengguna menekan tombol simpan, sistem akan memperbarui data tersebut ke dalam *database*. Pada proses perubahan kata sandi, pengguna diminta memasukkan kata sandi lama dan kata sandi baru. Sistem akan melakukan validasi terhadap kata sandi lama sebelum menyimpan perubahan. Berdasarkan hasil pengujian, fungsi profil dan perubahan kata sandi dapat berjalan sesuai kebutuhan.

4) Implementasi halaman dashboard admin

Halaman *dashboard* admin ditampilkan setelah pengguna dengan peran admin berhasil masuk ke sistem. Pada halaman ini, sistem menampilkan ringkasan data seperti jumlah pengguna, jumlah ruang rapat, jumlah pemesanan, serta informasi aktivitas terbaru. Berdasarkan hasil pengujian, data pada *dashboard* dapat ditampilkan dengan baik dan membantu admin dalam memantau kondisi sistem secara umum.

5) Implementasi pengelolaan data pengguna oleh admin

Proses pengelolaan data pengguna dilakukan oleh admin melalui halaman manajemen pengguna. Admin dapat menambahkan pengguna baru dengan mengisi data seperti identitas karyawan, nama lengkap, *email*, peran, dan kata sandi. Setelah tombol simpan ditekan, sistem akan menyimpan data pengguna ke dalam *database*. Admin juga dapat mengubah data pengguna yang sudah ada, kemudian sistem akan memperbarui data tersebut. Berdasarkan hasil pengujian, proses tambah dan ubah data pengguna dapat berjalan sesuai dengan skenario yang diharapkan.

6) Implementasi aktivasi dan nonaktivasi akun pengguna

Proses aktivasi dan nonaktivasi akun dilakukan ketika admin memilih data pengguna tertentu, kemudian mengubah status akun menjadi aktif atau tidak aktif. Sistem akan memperbarui status akun tersebut pada *database*. Jika akun dinonaktifkan, pengguna tidak dapat menggunakan akun tersebut untuk masuk ke sistem. Berdasarkan hasil pengujian, fungsi ini berjalan sesuai kebutuhan dan dapat membantu admin dalam mengontrol akses pengguna.

7) Implementasi pengelolaan data-data master lokasi, ruang rapat, fasilitas ruang rapat, dan seluruh data pemesanan oleh admin

Implementasi pengelolaan data master oleh admin meliputi pengelolaan data lokasi, ruang rapat, fasilitas ruang rapat, serta seluruh data pemesanan. Admin dapat menambahkan, mengubah, mengaktifkan, atau menonaktifkan data sesuai kebutuhan melalui halaman yang tersedia, kemudian sistem akan menyimpan data memperbarui data tersebut ke dalam *database*. Data lokasi digunakan sebagai referensi ruang rapat, sedangkan data ruang dan fasilitas digunakan untuk mendukung proses pencarian serta pemesanan oleh karyawan. Selain itu, admin juga dapat melihat, menyetujui, atau membatalkan pemesanan ruang rapat dengan alasan tertentu. Berdasarkan hasil pengujian, seluruh fungsi pengelolaan tersebut berjalan dengan baik dan dapat mendukung proses manajemen ruang rapat secara teratur.

8) Implementasi pengelolaan notifikasi dan pengumuman dan pengiriman email otomatis

Proses pengelolaan notifikasi dan pengumuman dilakukan oleh admin untuk menyampaikan informasi kepada pengguna. Admin dapat membuat informasi baru yang kemudian ditampilkan kepada pengguna sesuai kebutuhan. Berdasarkan hasil pengujian, sistem dapat menampilkan informasi notifikasi dan pengumuman dan *email* yang terkirim otomatis.

9) *Implementasi penampilan log aktivitas sistem*

Log aktivitas digunakan untuk mencatat tindakan yang dilakukan oleh pengguna di dalam sistem, seperti masuk ke sistem, membuat pemesanan, mengubah data, menyetujui pemesanan, atau melakukan ekspor data. Admin dapat membuka halaman *log* aktivitas untuk melihat riwayat aktivitas tersebut. Berdasarkan hasil pengujian, sistem mampu menampilkan data aktivitas sehingga admin dapat melakukan pemantauan terhadap penggunaan sistem.

10) *Implementasi tombol ekspor data dalam format spreadsheet*

Proses ekspor data dilakukan ketika admin memilih fitur ekspor pada data tertentu, seperti data pengguna, data ruang, atau data pemesanan. Sistem akan menghasilkan file dalam format *spreadsheet* yang dapat digunakan untuk dokumentasi atau pelaporan. Berdasarkan hasil pengujian, fungsi ekspor dapat berjalan sesuai kebutuhan dan file yang dihasilkan berisi data yang sesuai dengan data pada sistem.

11) *Implementasi penampilan ketersediaan ruang rapat oleh karyawan*

Karyawan dapat melihat daftar ruang rapat yang tersedia melalui halaman pencarian ruang. Sistem menampilkan informasi ruang seperti nama ruang, lokasi, kapasitas, fasilitas, dan status ketersediaan. Berdasarkan hasil pengujian, informasi ruang dapat ditampilkan dengan baik sehingga karyawan dapat mengetahui ruang mana yang dapat digunakan untuk kegiatan rapat.

12) *Implementasi pencarian dan filter ruang rapat*

Proses pencarian dan *filter* ruang rapat dilakukan ketika karyawan memasukkan kriteria tertentu, seperti lokasi, kapasitas, fasilitas, status ruang, tanggal, dan waktu penggunaan. Sistem akan memproses kriteria tersebut dan menampilkan ruang yang sesuai. Berdasarkan hasil pengujian, fitur pencarian dan *filter* berjalan sesuai kebutuhan sehingga pengguna dapat menemukan ruang rapat secara lebih cepat dan efisien.

13) *Implementasi pemesanan ruang rapat oleh karyawan*

Proses pemesanan dilakukan ketika karyawan memilih ruang rapat, kemudian mengisi data pemesanan seperti judul rapat, deskripsi, tanggal, waktu mulai, dan waktu selesai. Setelah tombol *simpan* ditekan, sistem akan memeriksa ketersediaan ruang pada waktu tersebut. Jika tidak terdapat bentrok jadwal dan ruang dalam kondisi tersedia, sistem akan menyimpan data pemesanan ke dalam *database*. Berdasarkan hasil pengujian, fungsi pemesanan dapat berjalan dengan baik dan sistem mampu mencegah pemesanan pada jadwal yang tidak tersedia.

14) *Implementasi perubahan dan pembatalan pemesanan aktif*

Karyawan dapat mengubah data pemesanan yang masih aktif, seperti judul rapat, waktu, tanggal, dan deskripsi. Setelah perubahan disimpan, sistem akan melakukan validasi ulang terhadap ketersediaan ruang. Karyawan juga dapat membatalkan pemesanan dengan memasukkan alasan pembatalan. Berdasarkan hasil pengujian, sistem dapat memperbarui dan membatalkan data pemesanan sesuai skenario, sehingga pengguna tetap memiliki kendali terhadap pemesanan yang dibuat.

15) *Implementasi penampilan riwayat pemesanan pribadi*

Riwayat pemesanan pribadi ditampilkan pada halaman pemesanan milik karyawan. Sistem menampilkan daftar pemesanan yang pernah dibuat oleh pengguna, baik yang masih menunggu persetujuan, disetujui, selesai, maupun dibatalkan. Berdasarkan hasil pengujian, data riwayat pemesanan bersifat personal dan sesuai hak akses.

16) *Implementasi pemberian feedback setelah rapat*

Proses pemberian *feedback* dilakukan setelah karyawan menyelesaikan rapat. Pengguna dapat memilih pemesanan yang telah selesai, kemudian mengisi penilaian dan komentar. Setelah tombol *kirim* ditekan, sistem akan menyimpan data *feedback* ke dalam *database*. Sistem juga melakukan pengecekan agar satu pemesanan tidak diberikan *feedback* lebih dari satu kali. Berdasarkan hasil pengujian, fungsi *feedback* berjalan dengan baik dan dapat digunakan sebagai bahan evaluasi kualitas ruang rapat.

17) *Implementasi penampilan notifikasi pemesanan untuk karyawan*

Karyawan dapat melihat notifikasi yang berkaitan dengan aktivitas pemesanan, seperti status pemesanan yang menunggu persetujuan, disetujui, atau dibatalkan. Sistem menampilkan informasi tersebut agar pengguna mengetahui perkembangan pemesanan yang telah dibuat. Berdasarkan hasil pengujian, notifikasi dapat membantu pengguna dalam memantau status pemesanan tanpa harus melakukan pengecekan manual secara berulang.

18) *Implementasi proses logout*

Proses *logout* dilakukan ketika pengguna menekan tombol keluar dari sistem. Sistem akan menghapus sesi pengguna yang sedang aktif dan mengarahkan pengguna kembali ke halaman *login*. Berdasarkan hasil pengujian, fungsi *logout* berjalan dengan baik sehingga keamanan akses sistem dapat tetap terjaga setelah pengguna selesai menggunakan aplikasi.

5. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dilakukan, dapat disimpulkan bahwa sistem informasi pengelolaan ruang rapat berbasis web *RoomWise* berhasil dibangun dan mampu menjawab permasalahan pengelolaan ruang rapat di PT XYZ Batam. Sistem ini menggantikan proses manual yang sebelumnya rentan terhadap kesalahan pencatatan dan konflik jadwal dengan mekanisme pemesanan ruang rapat yang terstruktur, dan transparan. Penerapan metode *SDLC Prototype* memungkinkan keterlibatan pengguna secara aktif dalam proses pengembangan, sehingga sistem yang dihasilkan sesuai dengan kebutuhan operasional perusahaan. Penggunaan teknologi ASP.NET Core MVC dan SQL Server memberikan struktur sistem yang modular, aman, serta mudah dikembangkan di masa mendatang. Hasil pengujian menggunakan metode *black-box testing* menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik dan sesuai dengan spesifikasi kebutuhan fungsional maupun non-fungsional yang telah ditetapkan.

6. Saran dan Pengembangan

Berdasarkan hasil pengembangan dan pengujian sistem yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan pertimbangan untuk pengembangan selanjutnya. Sistem yang telah dibuat diharapkan dapat terus dikembangkan dengan menambahkan fitur-fitur pendukung yang lebih lengkap sesuai kebutuhan pengguna, seperti peningkatan tampilan antarmuka, penyempurnaan alur penggunaan, serta pengelolaan data yang lebih terstruktur. Selain itu, sistem juga perlu dilakukan pemeliharaan secara berkala agar tetap berjalan dengan baik dan dapat menyesuaikan dengan perubahan kebutuhan operasional di masa mendatang. Pengujian lebih lanjut juga disarankan dengan melibatkan lebih banyak pengguna agar sistem dapat dievaluasi dari berbagai sudut pandang dan menghasilkan aplikasi yang lebih optimal. Selain itu, untuk pengembangan selanjutnya disarankan agar sistem melakukan peningkatan keamanan pada proses autentikasi, khususnya dengan mengganti metode penyimpanan kata sandi yang masih menggunakan MD5 menjadi metode yang lebih aman seperti *bcrypt*. Penggunaan *bcrypt* lebih disarankan karena memiliki mekanisme hashing yang lebih kuat dan dapat membantu melindungi data pengguna dari risiko pencurian atau penyalahgunaan. Pengembangan lain yang dapat dilakukan adalah penambahan fitur pencatatan aktivitas pengguna, peningkatan validasi input, serta penyempurnaan fitur backup data agar sistem menjadi lebih aman, stabil, dan mudah dikelola.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada PT XYZ dan mentor terkait yang telah memberikan dukungan fasilitas dan bimbingan selama penelitian ini berlangsung. Ucapan terima kasih juga disampaikan kepada Polibatam dan dosen pembimbing yang telah menyempatkan bimbingan, serta kesempatan untuk mengangkatnya untuk menjadi tugas akhir.

Referensi

- [1] Izzuddin, M., Wibowo, N. C., & Wahyuni, E. D. (2025). Rancang Bangun Sistem Informasi Manajemen Ruang Rapat pada PT XYZ Menggunakan JavaScript. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(2).
- [2] Sudradjat, B. (2021). Penerapan Metode Prototype Sistem Informasi Peminjaman Ruang Meeting. *REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer*, 5(2), 222-228.
- [3] Falya, A. A., Ramadhan, K. A., Handayani, P. S., & Rahayu, T. (2024). MANAJEMEN PROYEK SISTEM INFORMASI PEMBUATAN APLIKASI RESERVASI RUANG MEETING BERBASIS WEB. In *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya* (Vol. 5, No. 2, pp. 105-116).
- [4] Juyuspan, F. A., & Oktivasari, P. (2017). Pengembangan website dinamis menggunakan ASP.NET MVC dan SQL server dengan metode RAD (studi kasus: PT X). *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringannya)*, 2(1), 16-21.
- [5] Raharjo, D. (2018). Penerapan ASP.NET dalam Pengembangan Sistem Informasi Berbasis Web. *Jurnal Teknologi dan Sistem Komputer*, 6(1), 67-73.
- [6] Susanto, R., & Andriana, A. D. (2016). Perbandingan Model Waterfall dan Prototyping untuk Pengembangan Sistem Informasi. *Majalah Ilmiah UNIKOM*, 14(1), 41-46.
- [7] Aulia, L., & Tambotoh, J. J. (2025). Perancangan Sistem Informasi Peminjaman Ruangan Berbasis Web Menggunakan Metode Prototype. *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 14(1).
- [8] Putri, R. A., & Nugroho, A. (2020). Sistem Informasi Peminjaman Ruangan Berbasis Web pada Fakultas Teknik. *Jurnal Sistem Informasi*, 12(1), 25-32.

- [9] Fatimah, Q. I., Marselino, R., & Asnil, A. (2021). Web-Based DC Motor Speed Design and Control. *MOTIVECTON: Journal of Mechanical, Electrical and Industrial Engineering*, 3(3), 101-112.
- [10] Ardilla, Y., Pramesti, S., dan Sunaryono, D. (2020). Rancang Bangun Sistem Informasi Manajemen Proyek Departemen IT PT. Pertamina UPMS V Surabaya. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 5(3), 136-145.
- [11] Wijaya, A. T., Bernard, B., Edwin, E., Erick, E., Jeslyn, J., dan Melinda, R. (2023). Pengaruh Penerapan Enterprise Resource Planning dan Supply Chain Management Pada PT. TDK Electronics Indonesia. *Jurnal Pendidikan Tambusai*, 7(1), 1909-1917.
- [12] Rudjiono, D., & Saputro, H. (2020). PENGEMBANGAN DESAIN WEBSITE SEBAGAI MEDIA INFORMAS DAN PROMOSI (Studi Kasus: PT. Nada Surya Tunggal Kecamatan Pringapus): PENGEMBANGAN DESAIN WEBSITE SEBAGAI MEDIA INFORMAS DAN PROMOSI (Studi Kasus: PT. Nada Surya Tunggal Kecamatan Pringapus). *Pixel: Jurnal Ilmiah Komputer Grafis*, 13(2), 56-66.
- [13] Mariko, S. (2019). Aplikasi website berbasis HTML dan Javascript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus. *Jurnal Inovasi Teknologi Pendidikan*, 6(1), 80-91.
- [14] Wardana, M. I., & Jazman, M. (2017, May). Rancang bangun sistem informasi geogafis pemetaan ruang ujian menggunakan Bootstrap dan leaflet. js (studi kasus: Fakultas sains dan teknologi uin suska riau). In *Seminar Nasional Teknologi Informasi Komunikasi dan Industri* (pp. 257-264).
- [15] Nurhadi, N. (2022). Buku Ajar: Bahasa Pemrograman Pemrograman Database Menggunakan VISUAL BASIC. NET (VB. NET) DAN MICROSOFT SQL SERVER CONTOH PROGRAM: PEMBUATAN PROGRAM APLIKASI INVENTORI.
- [16] Chapple, M. (2009). Microsoft SQL Server 2008 for dummies. John Wiley dan Sons.
- [17] Sutabri, T. (2012). Analisis Sistem Informasi. Yogyakarta: Andi. Wicaksono, Y. (2008). *Membangun bisnis online dg mambo+ cd*. Elex Media Komputindo.
- [18] Mulyani, S., "Metode Analisis dan perancangan sistem." Abdi Sistematika, Bandung, Indonesia, 2017.