

# Pengembangan Sistem Manajemen Peminjaman Ruang Moduler Berbasis *Internet Of Things* Dengan Pendekatan Arsitektur Web Dan Metode *Waterfall* (Studi Kasus: Fakultas Rekayasa Industri)

1<sup>st</sup> M Ikhsan Pratama Putra  
Fakultas Rekayasa Industri  
Universitas Telkom  
Bandung, Indonesia

ikhshanpratamaputra@student.telkomuni-  
versity.ac.id

2<sup>nd</sup> Ekky Novrizal Alam  
Fakultas Rekayasa Industri  
Universitas Telkom  
Bandung, Indonesia

ekkynovrizal@telkomuniversity.

3<sup>rd</sup> Tien Fabrianti Kusumasari  
Fakultas Rekayasa Industri  
Universitas Telkom  
Bandung, Indonesia

tienkusumasari@telkomuniversity.ac.id

**Abstrak** — Dalam era digital saat ini, kebutuhan akan teknologi informasi semakin meningkat, terutama untuk meningkatkan efisiensi kegiatan akademik di institusi pendidikan tinggi. FRI di Telkom University memiliki berbagai laboratorium yang mendukung kegiatan praktikum dan penelitian mahasiswa. Namun, proses peminjaman ruangan laboratorium masih mengalami beberapa kendala, seperti keterlambatan pengembalian kunci dan kesulitan dalam memantau penggunaan ruangan secara real-time. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan merancang dan mengembangkan sistem manajemen peminjaman ruangan berbasis Internet of Things (IoT) yang terintegrasi dengan smart lock TTLock. Sistem ini dikembangkan menggunakan metode Waterfall, yang dipilih karena cocok untuk proyek dengan persyaratan stabil dan tim pengembang yang berpengalaman. Hasil pengembangan menunjukkan bahwa sistem manajemen peminjaman ruangan berbasis IoT berhasil diimplementasikan dengan baik, menyediakan akses yang lebih mudah bagi pengguna serta kontrol dan pengelolaan ruangan yang lebih efisien. Pengujian sistem menunjukkan bahwa seluruh fungsionalitas, berfungsi dengan baik dan lulus uji sesuai dengan harapan. Dengan demikian, sistem ini berhasil mengatasi permasalahan peminjaman ruangan yang sebelumnya ada di modul laboratorium FRI. Pengembangan lebih lanjut disarankan untuk memperluas cakupan sistem ini ke seluruh ruangan di lingkungan FRI guna memberikan manfaat yang lebih besar dalam pengelolaan ruangan.

**Kata kunci** — Internet of Things (IoT), manajemen ruangan, peminjaman ruangan, smart lock, metode Waterfall

## I. PENDAHULUAN

Dalam era digital yang terus berkembang, kebutuhan akan informasi dalam kegiatan perkuliahan semakin meningkat. Teknologi informasi menjadi sangat penting dalam meningkatkan efisiensi kegiatan perkuliahan, memungkinkan

pengolahan informasi yang cepat dan akurat. Oleh karena itu, banyak institusi pendidikan tinggi telah beralih menggunakan sistem informasi untuk mendukung kegiatan perkuliahan.

Telkom University adalah sebuah perguruan tinggi swasta yang terletak di kota Bandung. Universitas ini memiliki tujuh fakultas, salah satunya adalah Fakultas Rekayasa Industri (FRI). Pada FRI terdapat berbagai laboratorium yang mendukung kegiatan praktikum dan penelitian mahasiswa. Pada lingkungan FRI terdapat pihak urusan laboratorium dan bengkel yang bertugas dalam mengelola dan memelihara ruangan laboratorium, serta membantu mahasiswa dalam pelaksanaan praktikum. Selain itu, laboran juga bertanggung jawab atas peminjaman ruang laboratorium untuk pelaksanaan praktikum dan penggunaan ruangan lainnya.

Berdasarkan wawancara yang telah dilakukan kepada kepala urusan laboratorium dan bengkel sebagai pihak yang mengatur urusan laboratorium, terdapat kendala yang terjadi ketika proses kegiatan dalam peminjaman ruangan, seperti keterlambatan dalam pengembalian kunci kepada urusan laboratorium dan bengkel yang mengakibatkan kesulitan dalam koordinasi ketika jadwal pemakaian saling berdekatan, tidak bisa melihat jadwal penggunaan ruangan yang sedang dipakai yang mengakibatkan kesulitan dalam perencanaan dan koordinasi penggunaan ruangan.

Pada kasus penelitian yang dilakukan oleh [1], Sistem pengelolaan kunci secara manual menggunakan buku catatan atau lemari penyimpanan sering kali tidak efisien dan rentan kesalahan sehingga dengan mengembangkan integrasi sistem *web-based* pemesanan dan pemantauan kunci dengan teknologi rak kunci cerdas dapat memberikan solusi yang komprehensif bagi pengelolaan sistem peminjaman kunci di lingkungan universitas. Pada kasus [2] juga mengembangkan sistem bernama Juno untuk pemesanan ruangan yang aman sehingga telah memberikan solusi yang efektif bagi perpustakaan dalam mengelola pemesanan ruang belajar yang efisien.

Berdasarkan permasalahan pada kegiatan peminjaman ruangan yang terjadi dan penelitian yang pernah dilakukan sebelumnya. Penelitian ini akan memberikan usulan solusi dengan mengembangkan sistem manajemen peminjaman ruangan berbasis *Internet of Things* (IoT) dengan harapan sistem yang dikembangkan akan membantu aktifitas peminjaman ruangan menjadi lebih efisien. Untuk mengembangkan sistem peminjaman ruangan tersebut diperlukannya metode pengembangan yang menjadi panduan dalam urutan pengerjaannya. Pada penelitian ini metode yang akan digunakan yaitu metode pengembangan *Waterfall*.

Metode *Waterfall* telah diterapkan oleh [3] dalam pengembangan sistem informasi dan dinilai sesuai serta direkomendasikan. Keberhasilan metode ini sangat bergantung pada definisi kebutuhan yang jelas di tahap awal proyek. Selain itu, menurut [4], model *Waterfall* memiliki tingkat keberhasilan yang cukup tinggi, terutama pada proyek perangkat lunak yang memiliki persyaratan stabil, melibatkan tim pengembangan berpengalaman, serta dengan resiko dan kendala yang dapat diantisipasi.

## II. KAJIAN TEORI

### A. Sistem Manajemen Ruangan

Sistem pengelolaan ruangan adalah solusi teknologi yang dirancang untuk memperlancar penjadwalan, reservasi, dan pemanfaatan ruangan di dalam organisasi. Dengan meningkatkan efisiensi dan mengoptimalkan penggunaan sumber daya, sistem ini bertujuan untuk menyederhanakan pengelolaan ruang, yang pada akhirnya meningkatkan produktivitas [5].

Dengan meningkatnya model kerja hibrida, pentingnya mengelola ruang rapat dengan efektif semakin meningkat. Laporan menunjukkan bahwa persentase ruang rapat yang telah dipesan namun tetap kosong karena ketidakhadiran meningkat, yang mengakibatkan penggunaan sumber daya yang tidak efisien. [6]. Menerapkan sistem pemesanan yang kuat dapat meminimalkan konflik penjadwalan, memberikan pembaruan dan wawasan secara real-time tentang pemanfaatan ruang, dan pada akhirnya berkontribusi pada tempat kerja yang lebih produktif [5].

### B. Aplikasi Web, Mobile, Desktop

Dalam era digital saat ini, aplikasi telah menjadi bagian integral dari kehidupan kita sehari-hari. Aplikasi ini bisa berbentuk web, mobile, atau desktop, dan masing-masing memiliki kelebihan serta kekurangannya sendiri.

#### 1. Aplikasi Berbasis Web

Aplikasi web merupakan jenis aplikasi yang dijalankan di dalam peramban (*browser*) dan dapat diakses melalui *URL* tanpa perlu diunduh atau diinstal. Beberapa contoh populer dari aplikasi web antara lain *Google Docs*, *Gmail*, dan *YouTube*. Karakteristik utama dari aplikasi berbasis web adalah kemampuannya untuk diakses melalui peramban (seperti *Chrome*, *Firefox*, atau *Safari*) tanpa perlu proses pengunduhan atau instalasi, serta dapat digunakan pada berbagai perangkat yang memiliki *peramban* dan koneksi internet.

#### 2. Aplikasi Berbasis Mobile

Aplikasi *mobile* adalah aplikasi yang dirancang khusus untuk perangkat seluler seperti *smartphone* dan *tablet*.

Aplikasi ini bisa diunduh dan diinstal melalui toko aplikasi seperti *Google Play Store* atau *Apple App Store*.

#### 3. Aplikasi Berbasis Desktop

Aplikasi *desktop* adalah aplikasi yang diinstal langsung pada komputer dan dijalankan tanpa memerlukan *browser*. Aplikasi berbasis *desktop* digunakan pada perangkat komputer dengan sumber daya yang lebih besar dibandingkan perangkat *mobile*. Contoh dari aplikasi *desktop* termasuk *Microsoft Word* dan *Adobe Photoshop*.

### C. TTLock Smart Lock

The *TTLock smart lock* adalah solusi penguncian yang serbaguna dan canggih yang telah menjadi populer karena berbagai fiturnya, kemudahan penggunaan, dan langkah-langkah keamanan yang kuat. Kunci ini mendukung berbagai metode akses, termasuk *Bluetooth*, sidik jari, kode sandi, dan kartu *IC*, membuatnya sangat dapat beradaptasi untuk berbagai kebutuhan pengguna. Beberapa fitur yang terdapat pada *TTLock*.

- Multi-Method Access:** *TTLock* memungkinkan pengguna membuka pintu menggunakan beberapa metode seperti aplikasi seluler (melalui *bluetooth*), sidik jari, kode sandi, atau kartu *IC*. Fleksibilitas ini memastikan bahwa pengguna dapat memilih metode yang paling nyaman sesuai dengan situasi mereka [7], [8].
- Remote Management:** Melalui aplikasi *TTLock*, pengguna dapat mengelola kunci mereka dari jarak jauh. Ini termasuk mengunci/membuka pintu, membuat dan membagikan *e-key*, serta memantau log akses. Antarmuka aplikasi ini dirancang agar ramah pengguna, memungkinkan bahkan pengguna yang tidak teknis sekalipun untuk mengelola keamanan rumah atau bisnis mereka dengan mudah [7], [9].
- Enhanced Security:** *TTLock* menggunakan enkripsi canggih untuk melindungi dari akses yang tidak sah. Selain itu, fitur seperti peringatan kerusakan dan diagnosis kesalahan semakin meningkatkan profil keamanan kunci pintar ini, menjadikannya pilihan yang andal untuk aplikasi perumahan maupun komersial [9].
- Gateway Integration:** Dengan mengintegrasikan gateway *Wi-Fi*, *TTLock* dapat melampaui keterbatasan *bluetooth* dengan memungkinkan akses dan kontrol jarak jauh dari mana saja di dunia. Ini sangat berguna untuk properti sewaan liburan, di mana akses dapat dikelola tanpa memerlukan kunci fisik [7], [8].
- Versatile Application:** *TTLock* cocok untuk berbagai lingkungan, termasuk rumah, kantor, hotel, dan sewaan liburan. Kemampuannya untuk mengelola beberapa titik akses dari satu aplikasi membuatnya ideal bagi manajer properti dan pemilik bisnis yang perlu mempertahankan kontrol atas beberapa kunci secara bersamaan [7], [10].

### D. Metodologi Pengembangan Perangkat Lunak

Metodologi pengembangan perangkat lunak adalah kerangka kerja atau pendekatan yang terstruktur yang menentukan perencanaan, implementasi, dan kontrol praktik yang digunakan dalam proyek pengembangan perangkat lunak. Metodologi ini terdiri dari beberapa fase, termasuk pengumpulan kebutuhan, analisis, desain, pengkodean, pengujian, dan pemeliharaan, yang bertujuan untuk memastikan pengembangan perangkat lunak yang berkualitas tinggi [4]. Beberapa metodologi umum yang diterapkan

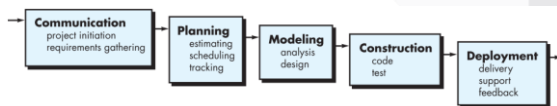
termasuk *Agile*, *Waterfall*, *Scrum*, dan *Iterative Incremental*, yang masing-masing memiliki berbagai proses berbeda dan fokus pada beberapa faktor, seperti fleksibilitas, kolaborasi dengan pelanggan, dan pengembangan iteratif [11]. Pemilihan metodologi yang tepat sangat penting dan harus disesuaikan dengan spesifikasi proyek, ukuran tim, dan risiko yang terlibat untuk mencapai keberhasilan proyek [4].

### 1. Waterfall

Model *Waterfall* adalah metode pengembangan perangkat lunak klasik yang ditandai oleh pendekatan yang linier dan berurutan. Metode ini melibatkan beberapa fase yang berbeda, yaitu pengumpulan kebutuhan, perancangan sistem, implementasi, pengujian, penerapan, dan pemeliharaan. Setiap fase harus diselesaikan sepenuhnya sebelum beralih ke fase berikutnya. Model ini paling efektif untuk proyek yang memiliki kebutuhan yang telah jelas dan stabil, dengan kemungkinan perubahan yang minim selama tahap pengembangan [12].

Berikut penjelasan mengenai tahapan yang ada pada model *Waterfall* [13], [14].

- Komunikasi, Tahap awal proyek di mana dilakukan inisiasi proyek dan pengumpulan persyaratan. Di sini, tujuan dan kebutuhan proyek didiskusikan dan dipahami dengan baik melalui komunikasi dengan pemangku kepentingan.
- Perencanaan, Setelah persyaratan dikumpulkan, dilakukan perencanaan proyek. Ini mencakup estimasi waktu, biaya, dan sumber daya, penjadwalan aktivitas proyek, serta pelacakan perkembangan agar proyek tetap sesuai rencana.
- Pemodelan, Pada tahap ini, dilakukan analisis dan perancangan sistem berdasarkan persyaratan yang telah dikumpulkan. Pemodelan ini bertujuan untuk memastikan solusi yang dikembangkan akan memenuhi kebutuhan pengguna.
- Konstruksi, Tahap ini melibatkan pengkodean dan pengujian. Sistem dirancang dalam bentuk kode program dan diuji untuk memastikan fungsionalitasnya berjalan sesuai spesifikasi.
- Penerapan, Sistem yang sudah selesai dikirimkan kepada pengguna akhir. Setelah diterapkan, tim pengembang memberikan dukungan dan mengumpulkan umpan balik untuk perbaikan di masa mendatang.



GAMBAR 1  
Fase *Waterfall* [14]

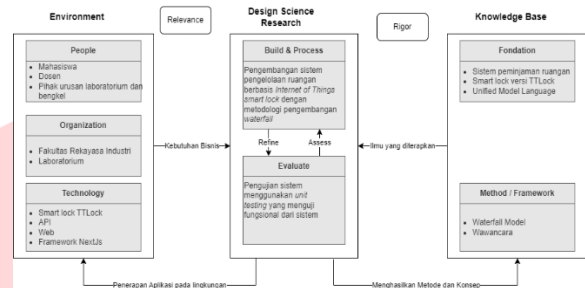
Berikut menguraikan kelebihan dan kekurangan metodologi *Waterfall*

## III. METODE

### A. Model Konseptual

Model konseptual adalah gambaran abstrak dari suatu sistem atau konsep yang digunakan untuk mendeskripsikan, menjelaskan, dan menganalisis aspek-aspek penting dari sistem tersebut. Model ini biasanya tidak bersifat teknis dan lebih menekankan pada interaksi komponen-komponen

sistem secara konseptual, tanpa terlalu memperhatikan detail teknis dari implementasinya [17]. Model konseptual yang digunakan dalam penelitian ini adalah model *design research science* [18] yang berfungsi sebagai kerangka untuk pemecahan masalah dengan cara menggambarkan lingkungan dan dasar ilmu yang terkait. Pemilihan model *design research science* sebagai kerangka pemecahan masalah didasarkan pada relevansinya serta kemampuannya dalam memberikan solusi terhadap penelitian dalam pengembangan sistem. Gambar III.1 merupakan gambaran dari model konsep pada penelitian ini.



GAMBAR 2  
Model Konseptual

Gambar 2 memperlihatkan diagram yang menjelaskan kerangka kerja Penelitian *Design Science Research* dalam konteks pengembangan sistem manajemen ruangan berbasis *Internet of Things* (IoT) menggunakan *smart lock* TTLock. Diagram ini dibagi menjadi tiga kolom utama: Lingkungan (*Environment*), Penelitian Desain Sains (*Design Science Research*), dan Basis Pengetahuan (*Knowledge Base*). Setiap kolom memiliki komponen-komponen yang menjelaskan elemen penting dalam penelitian dan pengembangan sistem tersebut.

#### 1. Environment

*Environment* menyoroti tiga aspek utama: *people*, *organization*, dan *technology*. Pada bagian *people* disebutkan bahwa mahasiswa, dosen, dan pihak yang bertanggung jawab atas laboratorium dan bengkel adalah pemangku kepentingan utama dalam penelitian ini. Ini menunjukkan bahwa penelitian dan pengembangan sistem ini sangat berfokus pada kebutuhan dan partisipasi aktif dari berbagai pihak di lingkungan akademik.

#### 2. Design Science Research

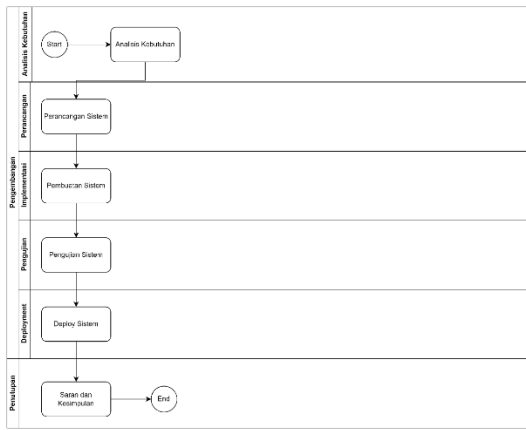
*Design Science Research* dibagi menjadi dua bagian utama: *build & process* dan *evaluate*. Pada bagian *build & process* dijelaskan bahwa sistem manajemen ruangan yang dikembangkan berbasis *Internet of Things* menggunakan *smart lock* TTLock dengan metodologi pengembangan *Waterfall*.

#### 3. Knowledge Base

*Knowledge Base* terdiri dari dua bagian: *foundation* dan *method / framework*.

### B. Sistematika Penelitian

Sistematika penelitian merupakan rangkaian tahap yang akan dilakukan dan digunakan dalam melakukan penelitian untuk mendapatkan solusi dari rumusan masalah dari studi kasus yang ada. Dengan menggunakan metode *Waterfall* setiap pengerjaan dilakukan secara bertahap dengan memastikan sebuah tahapan harus selesai sebelum memulai tahapan selanjutnya.



GAMBAR 3  
Sistematisa penelitian

1. Tahap Pengembangan
  - a. Analisa Kebutuhan
  - b. Perancangan
  - c. Implementasi
  - d. Pengujian
  - e. Deployment.

#### IV. HASIL DAN PEMBAHASAN

##### A. Tahap Komunikasi

Tahap ini merupakan langkah awal untuk memahami kebutuhan dari para pemangku kepentingan (*stakeholder*) yang akan menggunakan atau terlibat dalam pengembangan sistem. Berdasarkan hasil komunikasi dengan para *stakeholder*, kebutuhan fungsional sistem dapat diidentifikasi dengan lebih jelas. Adapun *stakeholder* yang terlibat dan kebutuhan fungsional yang terkait adalah sebagai berikut:

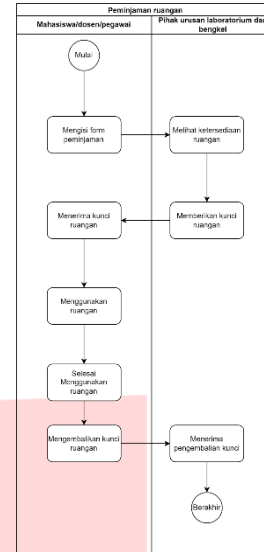
1. Mahasiswa, Dosen, dan Pegawai: Sebagai pengguna utama, mereka membutuhkan sistem yang memungkinkan pemesanan dan peminjaman ruangan laboratorium untuk berbagai keperluan seperti kegiatan akademik, penelitian, atau kegiatan lainnya.
2. Pihak Urusan Laboratorium: Sebagai admin, mereka bertugas mengelola data ruangan, menyetujui atau menolak permohonan peminjaman ruangan, serta memastikan ketersediaan fasilitas laboratorium.
3. Kepala Urusan Laboratorium: Sebagai super admin, bertanggung jawab untuk memantau keseluruhan sistem, manajemen *user* dan manajemen akun TTLock.

##### B. Tahap Analisis

Pada tahap analisis, peneliti mengidentifikasi proses peminjaman ruangan yang berlangsung di urusan laboratorium dengan menggambarannya menggunakan *Business Process Model and Notation (BPMN)*. Peneliti juga menggunakan analisis *GAP* untuk menentukan kebutuhan sistem yang akan dikembangkan. Analisis *GAP* ini berfungsi untuk mengidentifikasi perbedaan antara sistem atau proses yang ada saat ini (*as-is*) dan sistem atau proses yang diinginkan (*to-be*). Tujuan dari analisis ini adalah untuk menemukan kesenjangan yang ada serta memberikan pemahaman yang lebih mendalam mengenai aspek-aspek yang perlu diperbaiki atau ditingkatkan.

##### 1. Proses Bisnis Eksisting

Pada gambar 4 menggambarkan proses yang terjadi pada kegiatan peminjaman ruangan laboratorium.



GAMBAR 4  
Proses bisnis eksisting peminjaman ruangan

Diagram *BPMN* di atas menggambarkan proses bisnis peminjaman ruangan laboratorium yang melibatkan dua entitas utama: mahasiswa/dosen/pegawai dan pihak urusan laboratorium dan bengkel. Berikut adalah penjelasan alur prosesnya:

- a. Proses dimulai ketika mahasiswa, dosen, atau pegawai ingin meminjam ruangan laboratorium.
- b. Pengguna (mahasiswa/dosen/pegawai) mengisi formulir peminjaman ruangan sebagai langkah pertama dalam proses.
- c. Setelah formulir diterima, pihak urusan laboratorium mengecek ketersediaan ruangan yang diminta.
- d. Jika ruangan tersedia, pihak urusan laboratorium memberikan kunci ruangan kepada peminjam.
- e. Pengguna kemudian menggunakan ruangan sesuai dengan kebutuhan.
- f. Selesai Menggunakan Ruangan: Setelah selesai menggunakan ruangan, pengguna mengembalikan kunci ke pihak urusan laboratorium.
- g. Pihak urusan laboratorium menerima kunci yang dikembalikan dan memverifikasi proses peminjaman selesai.
- h. Setelah semua langkah selesai, proses peminjaman dinyatakan berakhir.

Diagram ini menunjukkan alur sederhana dan terstruktur, dengan interaksi antara pengguna dan pihak urusan laboratorium pada setiap tahapan kunci, mulai dari pengisian formulir hingga pengembalian kunci.

##### 2. GAP Analysis

Pada tabel IV.1 merupakan analisis *GAP* berdasarkan kegiatan peminjaman ruangan yang terjadi.

TABEL II.1  
Analisis GAP

No	As-Is (Sistem Saat Ini)	To-Be (Sistem yang Diusulkan)	GAP (Kesenjangan)
1	Pengelolaan Kunci Manual:	Pengelolaan Kunci Digital:	Manual vs otomatisasi;



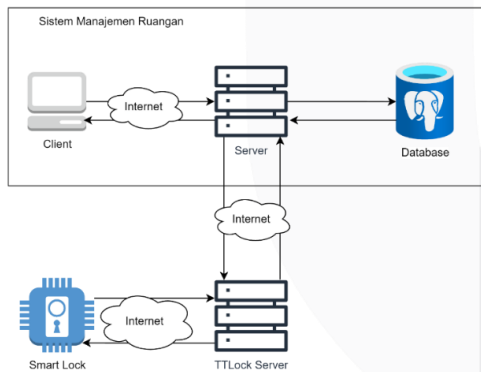
No	As-Is (Sistem Saat Ini)	To-Be (Sistem yang Diusulkan)	GAP (Kesenjangan)
	Keterlambatan dalam pengembalian kunci menghambat operasional.	Integrasi dengan TTLock untuk kunci pintar	memperbaiki waktu pengembalian kunci secara signifikan
2	Monitoring Sulit: Karena ruangan tersebar di tiga gedung, monitoring ruangan menjadi tidak efisien.	Monitoring <i>Real-time</i> : <i>Dashboard</i> untuk melihat status penggunaan ruangan.	Tidak adanya monitoring terpusat, sehingga menghambat kontrol yang efektif.
3	Koordinasi Kompleks: Perencanaan penggunaan ruangan sulit karena tidak ada sistem terintegrasi.	Koordinasi Terpusat: Booking dan jadwal terintegrasi dalam sistem digital	Koordinasi manual vs sistem otomatis, membantu perencanaan dan alokasi ruangan.

### C. Tahap Perancangan

Pada tahap perancangan, peneliti memvisualisasikan kebutuhan sistem yang telah diidentifikasi sebelumnya pada tahap analisis. Peneliti menggunakan diagram UML untuk membantu memvisualisasikan kebutuhan yang ada pada sistem.

#### 1. Diagram Arsitektur Sistem

Diagram arsitektur digunakan untuk merepresentasikan visual dari komponen sistem yang digunakan pada pengembangan sistem manajemen ruangan. Gambar 4 merupakan representasi dari diagram arsitektur pada penelitian ini.



GAMBAR 5  
Architecture diagram

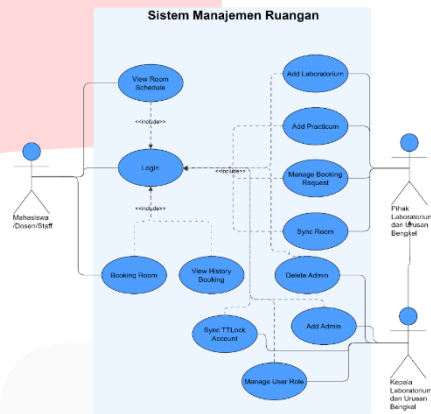
Gambar tersebut menggambarkan arsitektur sistem manajemen ruangan yang terintegrasi dengan smart lock dan server TTLock. Sistem ini terdiri dari beberapa komponen utama:

- Client: Pengguna berinteraksi dengan sistem melalui perangkat client, yang terhubung dengan server untuk mengakses informasi terkait manajemen ruangan.
- Server: Server bertugas untuk memproses permintaan dari client dan berkomunikasi dengan database. Server ini juga bertanggung jawab untuk meneruskan informasi yang terkait dengan *smart lock* ke TTLock Server.
- Database: Database digunakan untuk menyimpan data.

- TTLock Server: Server ini khusus digunakan untuk mengelola interaksi antara server utama sistem manajemen ruangan dan perangkat *smart lock*. Server TTLock mengatur kontrol akses ke *smart lock* yang ada di setiap ruangan.
- Smart Lock: Perangkat smart lock yang dipasang di pintu ruangan dapat dikendalikan oleh server melalui TTLock server, memungkinkan akses ruangan secara otomatis berdasarkan otorisasi yang diberikan oleh sistem.
- Internet: Digunakan agar semua komponen diatas saling terhubung dan dapat berkomunikasi satu sama lain.

#### 2. Use Case Diagram

*Use case diagram* digunakan untuk menggambarkan hubungan antara actor yang terlibat pada sistem dengan fitur yang bisa diaksesnya. Berikut merupakan *use case diagram* sistem manajemen ruangan untuk penelitian ini.

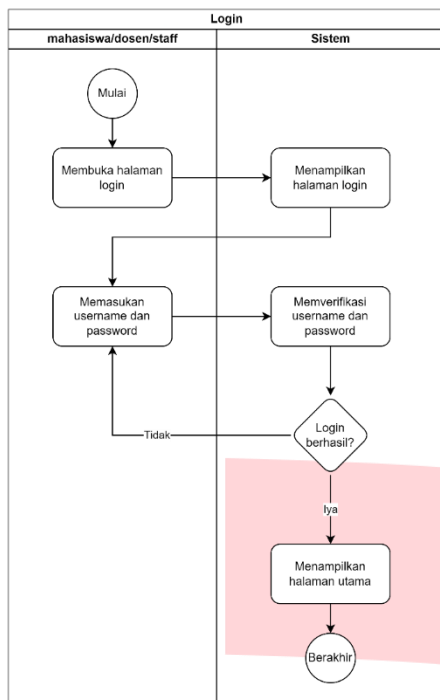


GAMBAR 6  
Use case diagram

#### 3. Diagram Aktifitas

Pada penelitian ini, peneliti menggunakan diagram aktifitas untuk menjelaskan alur aktifitas ketika pengguna berinteraksi dengan sistem. Berikut merupakan diagram aktifitas yang digunakan pada tahap perancangan ini.

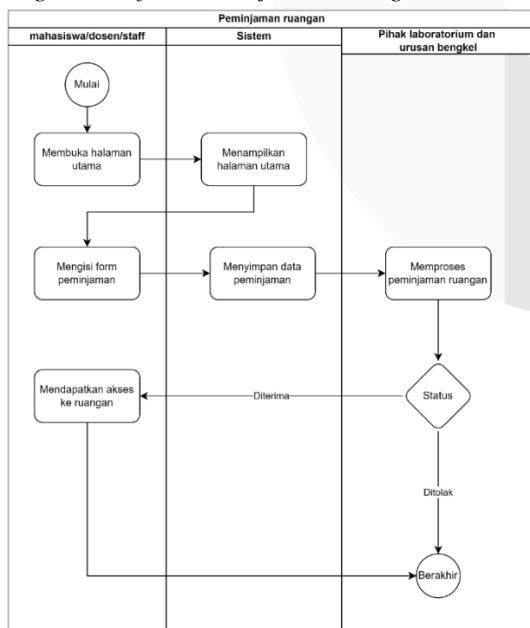
##### a. Diagram Aktifitas Login



GAMBAR 7  
Aktifitas login

Proses dimulai ketika pengguna membuka halaman login. Setelah itu, sistem akan menampilkan halaman login kepada pengguna. Pengguna kemudian memasukkan username dan password mereka. Sistem menerima input tersebut dan melakukan verifikasi untuk memastikan bahwa username dan password yang dimasukkan valid. Jika verifikasi berhasil, sistem akan menampilkan halaman utama kepada pengguna, menandakan bahwa proses login telah berhasil dan berakhir. Namun, jika verifikasi gagal, pengguna akan diminta untuk memasukkan kembali username dan password mereka sampai proses login berhasil.

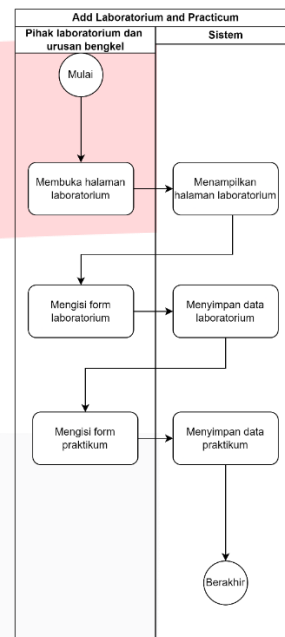
b. *Diagram Aktifitas Peminjaman Ruangan*



GAMBAR 8  
Aktifitas peminjaman ruangan

Proses pemesanan ruangan dimulai ketika pengguna membuka halaman utama. Sistem kemudian menampilkan halaman utama tersebut. Pengguna mengisi formulir peminjaman yang tersedia, dan sistem akan menyimpan data peminjaman tersebut. Setelah data peminjaman tersimpan, admin akan memproses permintaan peminjaman ruangan tersebut. Admin kemudian menentukan status dari permintaan peminjaman, apakah diterima atau ditolak. Jika permintaan diterima, pengguna akan mendapatkan akses ke ruangan yang dipesan. Jika permintaan ditolak, proses berakhir tanpa akses yang diberikan.

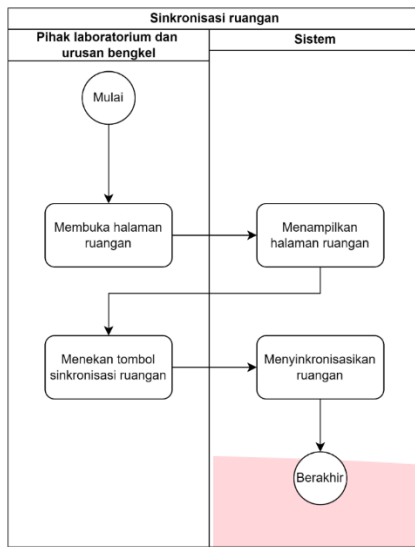
c. *Diagram Aktifitas Menambahkan Laboratorium dan Praktikum*



GAMBAR 9  
Aktifitas penambahan laboratorium dan praktikum

Proses dimulai ketika admin membuka halaman laboratorium. Sistem kemudian menampilkan halaman laboratorium kepada admin. Setelah itu, admin mengisi formulir laboratorium yang tersedia, dan sistem akan menyimpan data laboratorium tersebut. Setelah data laboratorium tersimpan, admin melanjutkan dengan mengisi formulir praktikum. Sistem kemudian menyimpan data praktikum yang diisi oleh admin. Setelah kedua data tersebut tersimpan, proses penambahan data laboratorium dan praktikum berakhir.

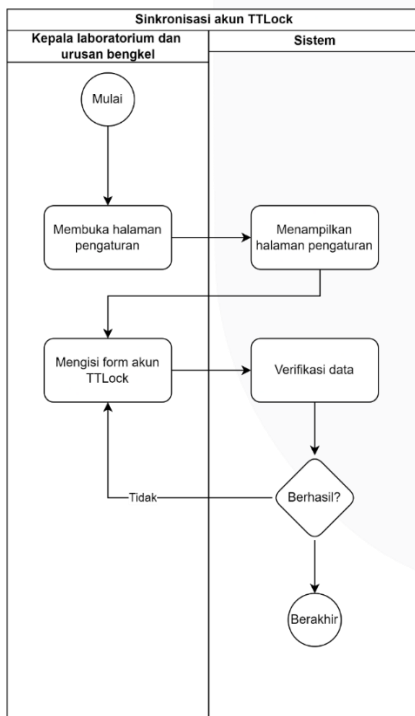
d. *Diagram Aktifitas Sinkronisasi Ruangan*



GAMBAR 10  
Aktifitas sinkronisasi ruangan

Proses dimulai ketika admin membuka halaman ruangan. Setelah itu, sistem merespons dengan menampilkan halaman ruangan kepada admin. Selanjutnya, admin menekan tombol sinkronisasi ruangan. Selanjutnya, sistem melakukan proses sinkronisasi ruangan. Setelah sinkronisasi selesai, proses ini berakhir.

e. Diagram Aktifitas Sinkronisasi akun TTLock

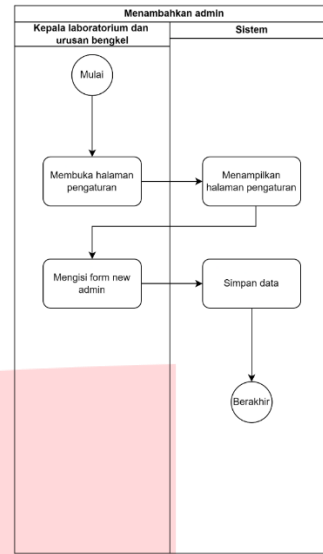


GAMBAR 11  
Aktifitas sinkronisasi akun TTLock

Proses dimulai ketika super admin membuka halaman pengaturan. Setelah halaman pengaturan dibuka, sistem akan menampilkan halaman tersebut kepada pengguna. Super admin kemudian mengisi form akun TTLock di halaman pengaturan. Setelah form diisi, sistem melakukan verifikasi data yang telah dimasukkan. Jika verifikasi data berhasil, proses berakhir dengan sukses. Namun, jika verifikasi gagal,

super admin harus mengisi ulang form akun TTLock untuk mencoba lagi, dan proses kembali ke langkah pengisian form.

f. Diagram Aktifitas Menambahkan Admin

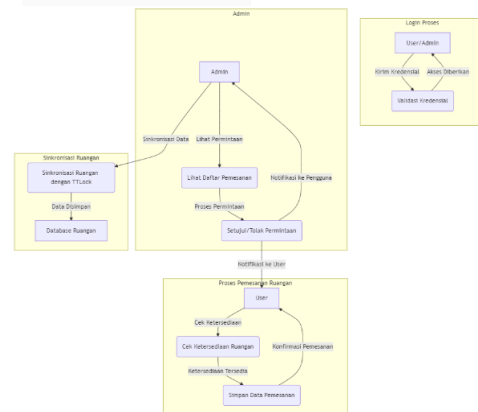


GAMBAR 12  
Aktifitas menambahkan admin

Proses dimulai ketika super admin membuka halaman pengaturan. Setelah itu, sistem menampilkan halaman pengaturan kepada super admin. Pada tahap berikutnya, super admin mengisi form untuk menambahkan admin baru. Setelah form diisi, sistem menyimpan data admin baru yang telah dimasukkan oleh super admin. Setelah data berhasil disimpan, proses ini berakhir.

4. Data Flow Diagram

Data flow diagram digunakan untuk menggambarkan proses alur data pada sistem manajemen ruangan. Berikut merupakan data flow diagram pada sistem manajemen ruangan.



GAMBAR 13  
Data flow diagram

Pertama, dalam proses login, baik User maupun Admin memulai interaksi dengan sistem melalui pengiriman kredensial. Kredensial yang dikirimkan oleh User atau Admin akan diverifikasi dalam proses validasi kredensial. Setelah validasi berhasil, akses ke sistem diberikan, memungkinkan mereka untuk melanjutkan ke aktivitas lainnya seperti sinkronisasi data atau pemesanan ruangan.

Bagian kedua dari diagram ini berfokus pada proses sinkronisasi ruangan. Admin bertanggung jawab untuk

melakukan sinkronisasi data dengan TTLock, sebuah sistem kunci pintar yang terintegrasi dengan manajemen ruangan. Setelah proses sinkronisasi selesai, data mengenai ruangan yang tersedia disimpan dalam database ruangan. Database ini kemudian digunakan oleh sistem untuk memeriksa ketersediaan ruangan ketika ada permintaan pemesanan dari User.

Selanjutnya, diagram ini menggambarkan proses pemesanan ruangan oleh User. User memulai dengan memeriksa ketersediaan ruangan yang ingin dipesan. Sistem akan melakukan pengecekan terhadap database ruangan yang telah disinkronisasi sebelumnya. Jika ruangan tersedia, informasi ini disampaikan kepada User, yang kemudian dapat melanjutkan proses dengan melakukan konfirmasi pemesanan. Setelah konfirmasi dilakukan, data pemesanan disimpan dalam sistem.

Di sisi lain, Admin memiliki kemampuan untuk melihat dan memproses permintaan pemesanan yang masuk. Admin dapat melihat daftar pemesanan yang telah dilakukan oleh User, dan kemudian mengambil keputusan untuk menyetujui atau menolak permintaan tersebut. Setelah keputusan dibuat, sistem akan mengirimkan notifikasi kepada User terkait status permintaan yang diajukan, apakah disetujui atau ditolak.

### 5. Component Diagram

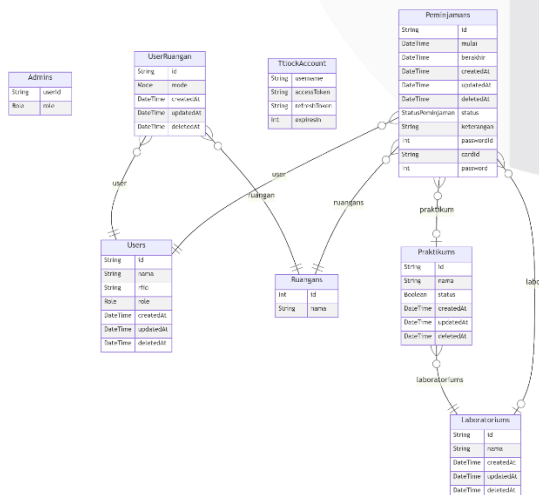
Komponen-komponen yang ditampilkan dalam diagram ini membantu dalam memahami struktur dan pembagian tanggung jawab di dalam sistem. Dengan memvisualisasikan hubungan antar komponen, kita bisa lebih mudah mengidentifikasi potensi masalah, mengoptimalkan arsitektur, serta memastikan bahwa semua bagian sistem terintegrasi dengan baik.

### 6. Sequence diagram

Sequence diagram menggambarkan alur proses pengiriman data antar *class/object* yang terjadi pada sistem manajemen ruangan modul laboratorium.

### 7. Entity-Relationship Diagram (ERD)

Entity-Relationship Diagram (ERD) digunakan untuk menggambarkan desain database yang akan digunakan dalam pengembangan sistem manajemen ruangan. ERD membantu dalam memvisualisasikan struktur database, termasuk entitas utama yang akan dikelola.



GAMBAR 14 ERD

### D. Implementasi

Pada tahap implementasi peneliti mengembangkan sistem manajemen ruangan modul laboratorium berdasarkan rancangan yang telah dibuat. Sistem yang dikembangkan menggunakan teknologi berupa *framework NextJs, database PostgreSQL, API TTLock*.

### E. Tahap Pengujian

Tahap pengujian dilakukan untuk memastikan bahwa sistem yang telah dikembangkan berfungsi sesuai dengan yang diharapkan serta memiliki performa yang optimal. Pengujian fungsionalitas dilakukan menggunakan metode *white box testing* untuk mengevaluasi logika dan struktur internal kode secara rinci. Selain itu, pengujian performa juga dilakukan untuk mengukur kecepatan, stabilitas, dan kemampuan sistem dalam menangani berbagai skenario penggunaan, termasuk beban yang tinggi dan kondisi ekstrem.

#### 1. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memastikan bahwa setiap fitur pada aplikasi berfungsi sesuai dengan yang diharapkan. Setiap *tes case* memiliki hasil yang diharapkan yang akan dibandingkan dengan hasil yang ditemukan saat pengujian. Tabel V. Berikut merupakan hasil pengujian.

TABEL II.2  
Pengujian fungsionalitas

No	Tes Class	Tes Case	Hasil yang diharapkan	Hasil yang ditemukan	Status
1	Login	Pengguna memasukkan username dan password yang valid	Pengguna berhasil masuk ke sistem dan diarahkan ke dashboard pengguna	Pengguna berhasil masuk ke sistem dan diarahkan ke dashboard pengguna	Lulus
2	Login	Pengguna memasukkan username dan/atau password yang salah	Pesan error ditampilkan, dan pengguna diminta untuk memasukkan username dan password yang benar	Pesan error ditampilkan, dan pengguna diminta untuk memasukkan username dan password yang benar	Lulus
3	User Booking Ruangan	Pengguna memilih ruangan dan waktu booking	Booking ruangan berhasil dilakukan dan muncul di histori booking	Booking ruangan berhasil dilakukan dan muncul di histori booking	Lulus
4	User Melihat Histori Booking Ruangan	Pengguna melihat histori booking ruangan yang telah dilakukan	Riwayat booking ruangan yang dilakukan pengguna ditampilkan dengan benar	Riwayat booking ruangan yang dilakukan pengguna ditampilkan dengan benar	Lulus
5	Admin Memproses Booking	Admin menerima atau menolak booking ruangan	Status booking berubah sesuai dengan keputusan Pihak urusan	Status booking berubah sesuai dengan keputusan	Lulus



No	Tes Class	Tes Case	Hasil yang diharapkan	Hasil yang ditemukan	Status
			laboratorium dan bengkel (diterima/ditolak) dan pemberitahuan dikirim ke pengguna	Pihak urusan laboratorium dan bengkel dan pemberitahuan dikirim ke pengguna	
6	Pihak urusan laboratorium dan bengkel Sinkronisasi Ruang dari TTLOCK	Pihak urusan laboratorium dan bengkel melakukan sinkronisasi data ruangan dengan TTLOCK	Data ruangan berhasil disinkronisasi dari TTLOCK ke sistem dengan status sinkronisasi yang sesuai	Data ruangan berhasil disinkronisasi dari TTLOCK ke sistem dengan status sinkronisasi yang sesuai	Lulus
7	Kepala urusan laboratorium dan bengkel Sinkronisasi Token Akses TTLOCK	Kepala Pihak urusan laboratorium dan bengkel melakukan sinkronisasi token akses TTLOCK	Token akses TTLOCK berhasil disinkronisasi dan diperbarui dalam sistem	Token akses TTLOCK berhasil disinkronisasi dan diperbarui dalam sistem	Lulus
8	Kepala urusan laboratorium dan bengkel Menambahkan Pihak urusan laboratorium dan bengkel Baru	Kepala urusan laboratorium dan bengkel menambahkan admin baru ke dalam sistem	Admin baru berhasil ditambahkan dengan kredensial yang valid	Admin baru berhasil ditambahkan dengan kredensial yang valid	Lulus
9	Pihak urusan laboratorium dan bengkel Menambahkan Laboratorium	Pihak urusan laboratorium dan bengkel menambahkan laboratorium baru ke dalam sistem	Laboratorium baru berhasil ditambahkan dan ditampilkan dalam daftar laboratorium	Laboratorium baru berhasil ditambahkan dan ditampilkan dalam daftar laboratorium	Lulus
10	Pihak urusan laboratorium dan bengkel Menambahkan Praktikum pada Laboratorium	Pihak urusan laboratorium dan bengkel menambahkan praktikum pada laboratorium tertentu	Praktikum baru berhasil ditambahkan pada laboratorium yang dipilih dan ditampilkan di daftar praktikum	Praktikum baru berhasil ditambahkan pada laboratorium yang dipilih dan ditampilkan di daftar praktikum	Lulus

## V. KESIMPULAN

### A. Kesimpulan

Pengembangan sistem manajemen peminjaman ruangan modul laboratorium berbasis IoT dengan metode Waterfall telah berhasil dilakukan, ditandai dengan setiap fitur utama seperti login, booking ruangan, sinkronisasi data TTLOCK, dan pengelolaan laboratorium oleh admin berfungsi dengan baik. Semua proses pengembangan mengikuti tahapan Waterfall, dan sistem yang dihasilkan mampu berjalan sesuai harapan.

Hasil pengujian sistem menunjukkan bahwa semua fungsionalitas yang diuji telah berfungsi sesuai dengan yang diharapkan, dengan semua skenario pengujian fungsionalitas, seperti proses login, peminjaman ruangan, sinkronisasi, dan penambahan admin serta laboratorium baru, berhasil lulus pengujian tanpa kendala. Ini membuktikan bahwa sistem memenuhi spesifikasi dan dapat diimplementasikan dengan baik di lingkungan operasional.

### B. Saran

Disarankan untuk melakukan evaluasi dan pemeliharaan sistem secara berkala guna memastikan sistem tetap berjalan dengan baik dan mampu memenuhi kebutuhan pengguna. Evaluasi ini dapat melibatkan pengumpulan umpan balik dari pengguna untuk mengetahui area-area yang memerlukan perbaikan atau pengembangan lebih lanjut. Disarankan untuk melakukan pengembangan lebih lanjut dengan memperluas cakupan sistem manajemen peminjaman ruangan ini ke seluruh ruangan di Fakultas Rekayasa Industri, tidak hanya terbatas pada modul laboratorium. Hal ini bertujuan untuk meningkatkan skala manfaat yang diperoleh dari sistem yang telah dikembangkan.

## REFERENSI

- [1] F. ' Aini Idris and N. Zainal, "Integration of Web-Based Key Booking and Monitoring System with Smart Key Rack for University Application," *Evolution in Electrical and Electronic Engineering*, vol. 3, no. 1, pp. 225–234, 2022, doi: 10.30880/eeee.2022.03.01.026.
- [2] W. Xuan, "Implementation of a secure room booking system at the University of Manitoba Libraries," *International Journal of Librarianship*, vol. 6, no. 2, pp. 63–72, Dec. 2021, doi: 10.23974/ijol.2021.vol6.2.194.
- [3] N. Raptama and Riki, "Analysis and Design of Information Systems for Make-Up Using the Waterfall Methodology," *bit-Tech*, vol. 5, no. 2, pp. 54–66, Dec. 2022, doi: 10.32877/bt.v5i2.553.
- [4] A. Angela Adanna and O. Francisca Nonyelum, "CRITERIA FOR CHOOSING THE RIGHT SOFTWARE DEVELOPMENT LIFE CYCLE METHOD FOR THE SUCCESS OF SOFTWARE PROJECT," 2020.
- [5] R. Bhagat, "An Ultimate Guide to Meeting Room Booking System." Accessed: Aug. 21, 2024. [Online]. Available: <https://www.vizitorapp.com/blog/ultimate-guide-to-meeting-room-management-system/>
- [6] M. Conquest, "HOME SHAREPOINT ONLINE MASTERING MEETING ROOM SETUP: TIPS FOR SUCCESS Mastering Meeting Room Setup: Tips for

- Success.” Accessed: Aug. 21, 2024. [Online]. Available: <https://blog.virtosoftware.com/meeting-room-management-guide/>
- [7] N. Askarov, K. Kozhakhmet, and L. Atymtayeva, “ANALYSIS OF WEB-BASED APPLICATIONS FOR EXPERT SYSTEM,” 2011.
- [8] N. Ketut Dewi Ari Jayanti, E. Triandini, G. Sastrawangsa, and N. Wayan Deriani, “Mobile Application Characteristics and User Perspective in Smart Healthcare Service Applications,” 2022.
- [9] D. Jane, “Building Desktop Apps for Education and Research: Bridging the digital divide.”
- [10] E. S. Soegoto and E. Mustafa, “Application of Web-based for E-business,” 2021.
- [11] I. Team, “TTLock 101: All You Need to Know.” Accessed: Aug. 11, 2024. [Online]. Available: <https://www.ilockey.com/ttlock-101-guide/>
- [12] V. Zhu, “TTlock: Everything About TTlock Usage And ‘How To’ Guide.” Accessed: Aug. 11, 2024. [Online]. Available: <https://www.acslocks.com/tt-lock/>
- [13] Lumive, “TTlock Smart Door Lock: What You Should Know.” Accessed: Aug. 11, 2024. [Online]. Available: <https://lumivestore.com/ttlock-smart-door-lock-what-you-should-know/>
- [14] M. Worth, “The BuZZ on TTLocks.” Accessed: Aug. 11, 2024. [Online]. Available: <https://remotelock.com/access-basics/the-buzz-on-ttlocks/>
- [15] Omg, “An OMG ® Unified Modeling Language ® (OMG UML ® ) OMG Document Number: Date,” 2009. [Online]. Available: <https://www.omg.org/spec/UML/20161101/PrimitiveTypes.xmi>
- [16] M. Mohanan, “Natural Language to Unified Modeling Diagrams; A Deep Learning Approach,” S. J. Nanda, R. P. Yadav, A. H. Gandomi, and M. Saraswat, Eds., Singapore: Springer Nature Singapore, Feb. 2024, pp. 495–509.
- [17] H. Koç, A. M. Erdoğan, Y. Barjakly, and S. Peker, “UML Diagrams in Software Engineering Research: A Systematic Literature Review,” MDPI AG, Mar. 2021, p. 13. doi: 10.3390/proceedings2021074013.
- [18] L. Letaw, “Handbook of Software Engineering Methods,” 2024.
- [19] C. D. Sang, N. Van He, N. Van Hien, and N. T. Khuyen, “An OOSEM-Based Design Pattern for the Development of AUV Controllers,” *J Mar Sci Eng*, vol. 12, no. 8, p. 1342, Aug. 2024, doi: 10.3390/jmse12081342.
- [20] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and W. Wang, “Software Testing with Large Language Models: Survey, Landscape, and Vision,” Feb. 2024, doi: <https://doi.org/10.1109/TSE.2024.3368208>.
- [21] Testilo, “Software Testing Trends to Look Out for in.”
- [22] L. Khoza and C. Marnewick, “Waterfall and agile information system project success rates-a South African perspective,” *South African Computer Journal*, vol. 32, no. 1, pp. 43–73, Jul. 2020, doi: 10.18489/sacj.v32i1.683.
- [23] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, “Agile software development: Methodologies and trends,” *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.
- [24] A. Mishra and Y. I. Alzoubi, “Structured software development versus agile software development: a comparative analysis,” *International Journal of System Assurance Engineering and Management*, vol. 14, no. 4, pp. 1504–1522, Aug. 2023, doi: 10.1007/s13198-023-01958-5.
- [25] R. S. Pressman and B. R. Maxim, *Software Engineering: A PRACTITIONER’S APPROACH, EIGHT EDITION*, 8th ed. McGraw Hill, 2014.
- [26] K. Schwaber and J. Sutherland, “The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game,” Nov. 2011.
- [27] C. Fagarasan, O. Popa, A. Pisla, and C. Cristea, “Agile, waterfall and iterative approach in information technology projects,” *IOP Conf Ser Mater Sci Eng*, vol. 1169, no. 1, p. 012025, Aug. 2021, doi: 10.1088/1757-899x/1169/1/012025.
- [28] N. Engelhardt, “COMPARISON OF AGILE AND TRADITIONAL PROJECT MANAGEMENT: SIMULATION OF PROCESS MODELS,” *Acta academica karviniensia*, vol. 19, no. 2, pp. 15–27, Jul. 2019, doi: 10.25142/aak.2019.011.
- [29] R. S. Pressman, *Software Engineering: A Practitioner’s Approach*, 6th ed. Elizabeth A. Jones, 2005.
- [30] J. Vom Brocke, A. Hevner, and A. Maedche, “Introduction to Design Science Research,” 2020.