

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### *2.1.1 Penerimaan Tamu*

Penerimaan tamu adalah suatu proses fundamental dari administrasi di berbagai jenis organisasi, termasuk institusi pendidikan seperti Politeknik Caltex Riau (PCR). Secara esensial, proses ini melibatkan serangkaian aktivitas mulai dari penyambutan, identifikasi, pendataan, hingga pengelolaan individu yang mengunjungi suatu entitas dengan tujuan tertentu. Dalam lingkup pendidikan tinggi, kategori tamu beragam, yang secara umum dapat dibedakan menjadi tamu dengan perjanjian sebelumnya dan tamu tanpa perjanjian (Sedianingsih dkk., 2014, dalam Purwanto dkk., 2024).

Pada era modern saat ini, pencatatan penerimaan tamu yang dilakukan secara manual menggunakan buku fisik mulai ditinggalkan karena dinilai tidak efisien dan rentan terhadap risiko kehilangan serta manipulasi informasi pengunjung. Digitalisasi melalui Sistem Informasi Buku Tamu kini menjadi solusi untuk meningkatkan pengawasan keamanan fisik sekaligus mutu layanan institusi. Transformasi menuju sistem presensi tamu digital ini tidak hanya mempercepat proses pendataan, tetapi juga mendukung tata kelola administrasi yang lebih terpusat sehingga pihak manajemen dapat melakukan evaluasi berdasarkan rekam jejak kunjungan yang valid (Bangsa dkk., 2025).

##### *2.1.2 Presensi Digital*

Presensi digital merupakan metode pencatatan kehadiran yang dilakukan secara elektronik dengan memanfaatkan perangkat digital seperti komputer, *smartphone*, atau tablet. Sistem ini dirancang untuk menggantikan metode presensi manual yang dinilai kurang efisien karena prosesnya lambat, tidak terpusat, dan rentan terhadap manipulasi data. Selain itu, pencatatan manual juga menyulitkan proses rekapitulasi, terutama ketika jumlah peserta atau tamu yang hadir dalam suatu kegiatan sangat banyak (Azizah dan Husni, 2025).

Melalui penerapan presensi digital, data kehadiran dapat dicatat secara *real-time* dan tersimpan ke dalam basis data. Penerapan ini juga dapat meningkatkan efisiensi administrasi dan meminimalkan *human error* (Heldiansyah dkk., 2023). Teknologi yang umum digunakan dalam sistem presensi digital adalah *QR Code* dan sidik jari *biometrik*.

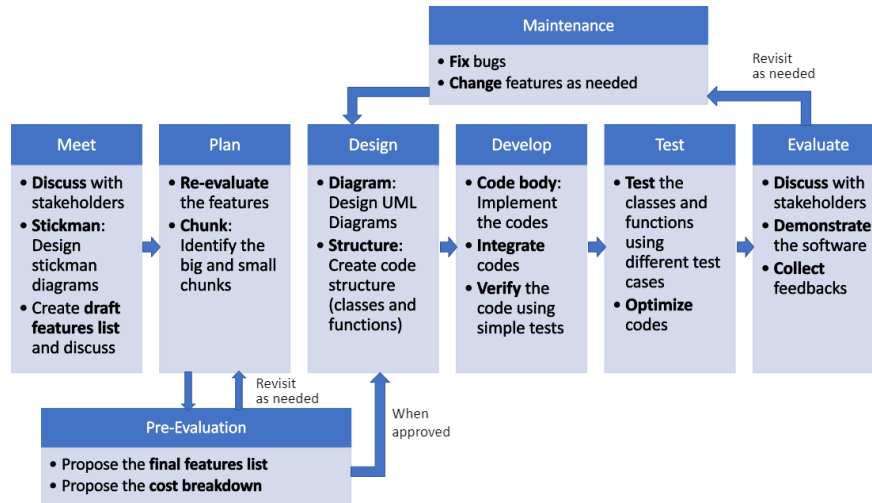
### 2.1.3 Rapid Solo Software Development (RSSD)

*Rapid Solo Software Development* (RSSD) adalah sebuah metodologi pengembangan perangkat lunak yang didasarkan pada Agile. Metodologi ini dirancang khusus untuk menjawab tantangan unik yang dihadapi oleh pengembang tunggal, seperti mahasiswa yang mengerjakan proyek akhir, *freelancer*, atau pengembang di perusahaan kecil. Meskipun prinsip-prinsip Agile sangat efektif, banyak kerangka kerja turunannya (seperti Scrum dan XP) sangat berorientasi pada kolaborasi tim, sehingga kurang praktis untuk diterapkan dalam konteks pengerjaan proyek secara mandiri.

Metodologi ini dibangun di atas tiga nilai utama (Purba dan Ramli, 2022):

1. *Efficient*, yaitu berfokus pada penyederhanaan proses, terutama pada tahap perencanaan, untuk mengurangi potensi pengerjaan ulang yang tidak perlu.
2. Modular, yaitu menekankan pada pemecahan kode menjadi fungsi-fungsi kecil dengan dependensi yang minim untuk mengurangi kompleksitas dan meningkatkan fleksibilitas.
3. *Revisitable*, yaitu berfokus pada peningkatan kejelasan diagram, dokumentasi, dan struktur kode agar mudah dipahami dan dimodifikasi kembali di masa depan.

Alur kerja metodologi RSSD divisualisasikan pada Gambar 2.1.



Gambar 2.1 Alur Metodologi *Rapid Solo Software Development* (RSSD)

(Sumber: Purba dan Ramli, 2022)

Berikut adalah penjelasan rinci untuk setiap fase dalam metodologi RSSD (Purba dan Ramli, 2022):

#### 1. Fase *Meet*

Ini adalah tahap pertemuan awal antara pengembang dengan pemangku kepentingan untuk membahas kebutuhan, membuat sketsa sederhana untuk memvisualisasikan fitur, dan menyusun draf daftar fitur yang disetujui bersama.

#### 2. Fase *Plan and Pre-Evaluation*

Pada tahap ini, pengembang mengevaluasi kembali kelayakan teknis fitur, lalu memecah proyek menjadi bagian-bagian kecil (*chunks*).

#### 3. Fase *Design*

Fase ini berfokus pada perancangan cetak biru teknis. Menurut RSSD, pengembang hanya perlu membuat diagram yang esensial saja (seperti *use case* diagram atau ERD).

#### 4. Fase *Develop*

Ini adalah fase implementasi dimana pengembang menulis kode program berdasarkan struktur yang telah dirancang. Aktivitasnya meliputi penulisan badan kode, mengintegrasikan berbagai fungsi, serta melakukan verifikasi sederhana untuk memastikan setiap fungsi berjalan secara individual.

#### 5. Fase *Test*

Fase ini bertujuan untuk memastikan kualitas dan stabilitas perangkat lunak. Pengujian dilakukan dengan menggunakan berbagai kasus uji untuk setiap

fungsi, dengan metode *black-box* menjadi pilihan utama. Fase ini juga mencakup aktivitas optimisasi kode, seperti menyempurnakan *query* SQL untuk memastikan performa aplikasi tetap baik.

#### 6. Fase *Evaluate and Maintenance*

Ini adalah fase akhir yang berjalan secara bersamaan. Pada tahap evaluasi, pengembang mendemonstrasikan produk akhir kepada pemangku kepentingan dan mengumpulkan umpan balik. Berdasarkan umpan balik tersebut, tahap pemeliharaan dilakukan untuk memperbaiki *bug* atau menangani permintaan perubahan fitur minor.

#### 2.1.4 Kanban

Kanban merupakan sebuah kerangka kerja manajemen proyek visual yang populer, yang sering digunakan untuk mengimplementasikan prinsip-prinsip Agile. Dalam konteks penelitian ini, Kanban digunakan sebagai alat untuk mengelola dan memvisualisasikan alur kerja proyek secara sistematis (Dien dkk., 2021; Hartono dan Ramadhan, 2022).

Prinsip utama Kanban adalah visualisasi alur kerja melalui penggunaan papan Kanban. Papan ini adalah representasi visual yang digunakan untuk mengelola alur kerja proyek dari awal hingga akhir (Hammarberg dan Sunden, 2014). Papan ini umumnya dibagi menjadi beberapa kolom yang merepresentasikan tahapan proses, seperti tugas yang akan dilakukan (*to do*), tugas yang sedang dikerjakan (*in progress*), dan tugas yang sudah selesai (*done*). Dengan mengetahui status dari setiap tugas, pengembang dapat lebih fokus dan efisien dalam bekerja, serta memungkinkan kemajuan proyek dapat dipantau secara transparan dan terstruktur. Menurut Hammarberg dan Sunden (2014), selain visualisasi, Kanban didasarkan pada beberapa prinsip inti lainnya untuk mengoptimalkan alur kerja:

##### 1. *Limit Work in Progress*.

Prinsip ini mengharuskan pengembang untuk menetapkan batas maksimum jumlah tugas yang boleh berada di kolom *in progress* pada satu waktu. Tujuannya adalah untuk mengurangi pemborosan akibat *multitasking* dan membantu pengembang tetap fokus dalam menyelesaikan pekerjaan yang sedang berlangsung sebelum memulai tugas baru.

## 2. *Focus on Flow of Work.*

Untuk mencapai hasil yang efektif, penting bagi pengembang untuk berfokus pada kelancaran aliran pengerjaan proyek dari awal hingga selesai. Dengan begitu, pengembang dapat mengidentifikasi dan menangani hambatan yang mungkin timbul, sehingga alur proyek dapat berjalan secara lancar dan efisien. Penggunaan Kanban dalam penelitian ini secara spesifik mendukung implementasi metodologi RSSD dengan cara menerjemahkan item-item *product backlog* yang telah disusun pada tahap *plan* menjadi kartu-kartu tugas visual pada papan Kanban. Hal ini memungkinkan pelacakan kemajuan yang jelas selama fase *develop* dan *test*.

### 2.1.5 *Aplikasi Web*

Aplikasi web adalah aplikasi perangkat lunak yang diakses menggunakan peramban (*browser*) melalui jaringan internet atau intranet tanpa perlu instalasi pada perangkat pengguna. Aplikasi ini dibangun menggunakan teknologi berbasis web seperti HTML, CSS, JavaScript, serta bahasa pemrograman *server-side* seperti PHP atau Python untuk memproses data di server (Puspita dkk., 2024). Keunggulan aplikasi web meliputi kemudahan akses dari berbagai perangkat dan efisiensi biaya pemeliharaan.

Keunggulan utama dari implementasi aplikasi berbasis web terletak pada tingkat aksesibilitasnya yang tinggi serta sifatnya yang tidak terikat pada satu sistem operasi tertentu. Pengguna dapat mengoperasikan sistem dari berbagai jenis *gadget*, asalkan perangkat tersebut dilengkapi dengan peramban dan koneksi jaringan. Dalam konteks sistem presensi, pendekatan berbasis web dinilai efisien karena pengguna (seperti tamu) hanya perlu memindai kode *QR* untuk langsung membuka antarmuka formulir kehadiran tanpa membuang waktu mengunduh aplikasi tambahan, sehingga secara signifikan mempercepat alur kerja (Septiano dan Pramono, 2024).

### 2.1.6 *Data Warehouse*

*Data warehouse* adalah sebuah sistem basis data yang bersifat terpusat, terintegrasi, dan dirancang khusus untuk mendukung proses pengambilan keputusan. Sistem ini berfungsi sebagai gudang data komprehensif yang

menyediakan ringkasan dan rincian informasi untuk analisis (Wahono dan Ali, 2021). Secara konseptual, informasi yang disimpan di dalam infrastruktur ini memiliki empat karakteristik utama, yaitu berorientasi pada subjek (*subject-oriented*), terintegrasi (*integrated*), tidak mudah berubah (*nonvolatile*), dan memiliki rentang waktu tertentu (*time-variant*). Keempat karakteristik pokok tersebut membedakannya secara tegas dari basis data biasa, menjadikannya sangat ideal untuk tujuan pelaporan analitis daripada pemrosesan transaksi harian.

Dalam tahap perancangannya, arsitektur *data warehouse* umumnya mengadopsi pendekatan pemodelan dimensional yang memecah struktur informasi ke dalam entitas tabel fakta dan tabel dimensi. Tabel fakta bertugas untuk menyimpan metrik atau nilai numerik yang dapat diukur dari suatu kejadian bisnis, sedangkan tabel dimensi mendeskripsikan konteks atribut dari fakta tersebut, seperti waktu, lokasi, ataupun entitas profil pengguna (Kimball dan Ross, 2013). Untuk menyuplai data operasional ke dalam struktur dimensional ini, diperlukan tahapan integrasi otomatis yang dikenal dengan proses *Extract, Transform, Load* (ETL) maupun variasinya yaitu *Extract, Load, Transform* (ELT). Melalui kombinasi skema yang terstruktur dan aliran pembaruan data yang konsisten, aliran informasi mentah dapat divisualisasikan menjadi sistem *business intelligence* yang akurat guna mengevaluasi kinerja institusi (Wahono dan Ali, 2021).

#### 2.1.7 Business Intelligence

*Business Intelligence* (BI) adalah sekumpulan proses, teknologi, dan alat yang digunakan untuk mengubah data mentah menjadi informasi yang bermakna guna mendukung pengambilan keputusan bisnis yang efektif (Arifin dan Sugiharto, 2013). *BI* memungkinkan organisasi untuk mengumpulkan, mengintegrasikan, menganalisis, dan menyajikan data dalam bentuk yang mudah dipahami, seperti laporan dan *dashboard* interaktif. Dengan demikian, *BI* membantu manajemen dalam merumuskan strategi berdasarkan data yang akurat dan terkini.

Pada tingkat implementasinya, lapisan antarmuka dari sistem *business intelligence* umumnya direpresentasikan melalui dasbor analitik yang interaktif (Arifin dan Sugiharto, 2013). Dasbor ini bertugas merangkum kompleksitas data *warehouse* menjadi representasi visual, seperti diagram, grafik, dan metrik kinerja

utama, yang dapat dipahami dengan cepat oleh pengguna akhir. Dengan mengintegrasikan sistem informasi operasional seperti aplikasi pencatatan presensi tamu ke dalam arsitektur BI, institusi pendidikan dapat melakukan pemantauan aktivitas pengunjung secara komprehensif, mengidentifikasi pola mobilitas kunjungan, serta melakukan penyaringan data secara mandiri dalam satu pusat kendali visual.

#### 2.1.8 Feature Testing

*Feature testing* adalah pendekatan pengujian perangkat lunak yang dirancang untuk memvalidasi fungsionalitas spesifik dari sebuah aplikasi secara menyeluruh pada lapisan *backend*. Berbeda dengan metode *unit testing* yang berfokus ketat pada pengujian potongan kode yang sangat kecil dan terisolasi secara mandiri, *feature testing* menguji beberapa objek komponen sekaligus untuk mengevaluasi bagaimana entitas-entitas tersebut berperilaku dan berinteraksi satu sama lain. Melalui pendekatan holistik ini, pengembang dapat memastikan bahwa integrasi antarkomponen sistem perangkat lunak telah berjalan selaras untuk menghasilkan fungsionalitas bisnis yang utuh (Pratomo dkk., 2019).

Dalam ekosistem kerangka kerja (*framework*) pengembangan web modern, *feature testing* menjadi instrumen esensial dalam mengukur stabilitas logika aplikasi sebelum diuji secara visual pada antarmuka. Proses pengujian ini secara teknis dieksekusi dengan menyimulasikan permintaan HTTP (seperti *GET*, *POST*, *PUT*, atau *DELETE*) secara otomatis langsung menuju titik akhir (*endpoint*) atau rute aplikasi web. Setelah simulasi dikirimkan, kerangka pengujian akan melakukan validasi (*assertion*) terhadap kesesuaian respons dari *server*, kecocokan kode status HTTP yang dikembalikan, hingga mengevaluasi keberhasilan transaksi data aktual di dalam basis data (Pratomo dkk., 2019). Dengan mengeksekusi simulasi aliran data riil ini, segala bentuk anomali, kesalahan logika, maupun celah keamanan sistem dapat dideteksi secara dini guna menjamin keandalan aplikasi.

#### 2.1.9 Black-Box Testing

*Black-box testing* adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem tanpa mengetahui struktur internal atau cara

kerja perangkat lunak tersebut. Dalam pengujian ini, penguji hanya berinteraksi dengan antarmuka pengguna dan memeriksa *output* yang dihasilkan berdasarkan *input* yang diberikan. Metode ini berguna untuk memastikan bahwa sistem berfungsi sesuai dengan yang diharapkan sesuai dengan spesifikasi yang telah ditentukan (Myers dkk., 2012).

Pada implementasi pengembangan aplikasi berbasis web, pengujian *black-box* sangat esensial untuk memvalidasi alur kerja interaksi pengguna, seperti kelancaran proses pengisian formulir, mekanisme autentikasi *login*, hingga ketepatan navigasi antarhalaman. Salah satu teknik yang paling umum diterapkan dalam pengujian ini adalah *Equivalence Partitioning*, sebuah pendekatan yang membantu penguji menentukan skenario pengujian secara lebih terstruktur dengan membagi masukan data ke dalam kelompok valid dan tidak valid (Amalia dkk., 2021). Melalui pelaksanaan skenario pengujian ini secara komprehensif, pengembang dapat memastikan bahwa sistem mampu memberikan respons yang tepat, termasuk menampilkan pesan *error* saat menerima data yang salah, sehingga keandalan aplikasi dapat terjamin sebelum dioperasikan secara penuh oleh pengguna.

#### 2.1.10 Performance Testing

*Performance testing* adalah jenis pengujian yang bertujuan untuk menentukan dan memvalidasi karakteristik seperti kecepatan, skalabilitas, dan stabilitas suatu sistem atau aplikasi ketika berada di bawah beban kerja tertentu (Hegde dan S, 2014). Pengujian ini juga dapat didefinisikan sebagai proses untuk mengukur seberapa efektif dan cepat sebuah aplikasi, jaringan, atau perangkat beroperasi. Tujuan utamanya adalah untuk memastikan sistem siap digunakan di lingkungan produksi dan memenuhi kriteria yang telah ditetapkan, serta mengidentifikasi sumber masalah performa, mendukung penyetelan sistem, dan mengumpulkan data untuk pengambilan keputusan (Hegde dan S, 2014).

Dalam konteks pengembangan aplikasi berbasis web, *performance testing* umumnya direalisasikan melalui teknik *load testing* untuk menyimulasikan akses pengguna dalam jumlah besar secara bersamaan. Pendekatan ini bertujuan untuk mengukur batas kapasitas maksimum *server* dan memastikan bahwa waktu respons

halaman tetap berada pada ambang toleransi yang wajar meskipun terjadi lonjakan lalu lintas jaringan secara mendadak. Eksekusi pengujian ini biasanya dilakukan secara terotomatisasi menggunakan perangkat lunak khusus yang dikonfigurasi untuk mengirimkan ribuan permintaan *HTTP* virtual secara terus-menerus ke *server* target. Berdasarkan hasil metrik lalu lintas yang diperoleh dari skenario simulasi tersebut, pihak pengembang dapat memutuskan secara objektif apakah diperlukan eskalasi spesifikasi perangkat keras (*RAM/CPU*) atau sekadar optimalisasi logika *kueri* basis data untuk menjaga stabilitas sistem (Ismail dkk., 2023).

#### 2.1.11 Usability Testing

*Usability testing* adalah jenis pengujian yang bertujuan untuk mengevaluasi seberapa mudah dan efisien suatu sistem atau aplikasi digunakan oleh penggunanya. Dalam pengujian ini, pengguna atau partisipan diminta untuk menyelesaikan tugas-tugas tertentu menggunakan sistem, dan penguji mengamati serta mencatat kesulitan atau masalah yang mereka temui dalam proses tersebut (Nielsen, 1993). *Usability testing* biasanya digunakan pada tahap akhir pengembangan perangkat lunak untuk memastikan bahwa aplikasi dapat digunakan dengan nyaman oleh pengguna yang dituju, tanpa kebingungannya.

Untuk mengukur tingkat kebergunaan suatu sistem secara kuantitatif dan objektif, para peneliti umumnya menggunakan instrumen kuesioner standar, di mana salah satu yang paling umum adalah metode *System Usability Scale* (SUS). Instrumen pengujian SUS terdiri dari sepuluh butir pernyataan evaluasi yang diukur menggunakan skala *likert* lima poin, yang menilai persepsi pengguna terhadap aspek kompleksitas antarmuka, kecepatan pembelajaran, dan tingkat kepuasan interaksi (Pratama dkk., 2024). Melalui akumulasi dan perhitungan skor akhir dari kuesioner ini, pengembang dapat memetakan kualitas sistem ke dalam matriks penilaian standar industri untuk menentukan apakah aplikasi tersebut berada pada tingkat penerimaan yang memadai atau justru memerlukan penyempurnaan fungsionalitas lebih lanjut.

## 2.2 Penelitian Terdahulu

Penelitian terdahulu dilakukan agar mendapatkan wawasan sebagai salah satu sumber ide dan referensi dalam penelitian yang akan dilakukan. Dengan meninjau penelitian sebelumnya, dapat dipahami pendekatan, metode, serta teknologi yang telah digunakan untuk memecahkan permasalahan yang serupa. Selain itu, bermanfaat juga untuk mengidentifikasi kelebihan dan kekurangan dari sistem yang telah ada, sehingga dapat menjadi dasar pengembangan sistem yang lebih optimal.

Penelitian dilakukan oleh Gunawan dkk. (2023). Penelitian ini bertujuan untuk membangun aplikasi presensi digital berbasis web menggunakan *QR Code* untuk menggantikan presensi secara manual yang dinilai kurang efektif dan rawan kecurangan. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan berhasil meningkatkan efisiensi pencatatan kehadiran serta mengurangi kemungkinan manipulasi data presensi.

Penelitian dilakukan oleh Fahrezi dkk. (2024). Penelitian ini bertujuan untuk membangun sistem buku tamu berbasis web sebagai solusi atas pencatatan tamu yang masih dilakukan secara manual. Hasil dari penelitian menunjukkan bahwa sistem yang dibangun mampu mempercepat proses entri data, memberikan akses secara *real-time* dari berbagai perangkat, serta mengurangi kesalahan yang sering terjadi dalam pencatatan manual.

Penelitian dilakukan oleh Yanti dkk. (2024). Penelitian ini bertujuan untuk merancang sistem buku tamu berbasis web untuk menggantikan pencatatan manual. Hasil penelitian menunjukkan bahwa sistem yang dibangun dapat mempermudah dan mempercepat pencatatan identitas tamu serta tujuan kunjungan, sehingga proses administrasi menjadi lebih efisien.

Penelitian dilakukan oleh Setiawan dkk. (2020). Penelitian ini bertujuan mengatasi permasalahan dalam proses administrasi kegiatan yang masih dilakukan secara manual. Sistem yang dibangun mampu mengelola berbagai data seperti data petugas, data *event*, hingga data keuangan kegiatan. Hasil penelitian menunjukkan bahwa aplikasi yang dibangun dapat memudahkan pengelolaan administrasi *event*.

Penelitian dilakukan oleh Hariyati dkk. (2021). Penelitian ini bertujuan untuk menggantikan sistem manual dalam pelayanan administrasi dan manajemen kegiatan dengan sistem berbasis web. Hasil dari penelitian ini menunjukkan bahwa

aplikasi yang dikembangkan mampu mempercepat proses pengolahan data serta memberikan tingkat keamanan yang lebih baik dalam penyimpanan data.

Penelitian dilakukan oleh Mauludin dan Ghazi (2021). Tujuan dari penelitian ini adalah mempermudah penyampaian informasi mengenai jadwal kegiatan yang sebelumnya dilakukan secara konvensional. Hasil dari penelitian menunjukkan bahwa sistem informasi jadwal berbasis web yang dibangun dapat meningkatkan efisiensi dalam penyampaian informasi terkait acara.

Penelitian dilakukan oleh Jayanti dan Khadija (2024). Penelitian ini bertujuan membangun *dashboard* operasional menggunakan Google Looker Studio untuk menyajikan data pengunjung secara *real-time* dan mudah dipahami oleh pemangku kepentingan. Hasilnya, *dashboard* ini mampu memberikan informasi statistik kunjungan yang lebih transparan, serta mendukung evaluasi efektivitas program.

Penelitian dilakukan oleh Perdana (2020). Tujuan dari penelitian ini adalah membangun *dashboard* berbasis web untuk mempermudah proses pelaporan data pengunjung yang sebelumnya dilakukan secara manual menggunakan Microsoft Excel. Hasil dari penelitian ini menunjukkan bahwa sistem yang dibangun mampu menyajikan informasi pengunjung dalam bentuk tabel dan grafik yang lebih praktis untuk *monitoring*.

Tabel 2.1 Perbandingan Penelitian Terdahulu

| Penulis             | Judul                                                                                                                                            | Masalah                              | Metode                                     | Hasil                                                                               |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|--------------------------------------------|-------------------------------------------------------------------------------------|
| Gunawan dkk. (2023) | Rancang Bangun Aplikasi E-Presensi Berbasis Web Menggunakan <i>QR Code</i>                                                                       | Presensi manual dan rawan kecurangan | <i>Design Science Research Methodology</i> | Efisiensi meningkat, mengurangi kecurangan presensi                                 |
| Fahrezi dkk. (2024) | Rancang Bangun Aplikasi Sistem Informasi Buku Tamu Berbasis Web Menggunakan Metode Native (Studi Kasus: Kantor Badan Pengelola Keuangan dan Aset | Pencatatan tamu masih manual         | Native                                     | Proses entri data lebih cepat, akses <i>real-time</i> , mengurangi kesalahan manual |

| Penulis                    | Judul                                                                                                                     | Masalah                                                  | Metode                          | Hasil                                                                            |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|---------------------------------|----------------------------------------------------------------------------------|
|                            | Daerah Provinsi Sulawesi Tenggara)                                                                                        |                                                          |                                 |                                                                                  |
| Yanti dkk. (2024)          | Perancangan Sistem Informasi Buku Tamu Pada <i>Website</i> Kantor Pencarian Dan Pertolongan Kelas A Medan                 | Pencatatan tamu masih manual                             | Waterfall                       | Mempermudah dan mempercepat pencatatan tamu                                      |
| Setiawan dkk. (2020)       | Sistem Informasi Administrasi Kegiatan ( <i>Event</i> ) Di Perusahaan CRPG Menggunakan Java                               | Administrasi kegiatan masih manual dan data tercecer     | <i>Grounded Research</i>        | Data kegiatan terkelola lebih baik                                               |
| Hariyati dkk. (2021)       | Sistem Informasi Pelayanan Administrasi dan Manajemen Kegiatan Pada Kantor Kelurahan Kuala Jelai                          | Sistem administrasi dan manajemen kegiatan masih manual  | Observasi, Wawancara, Literatur | Pengolahan data lebih cepat dan aman                                             |
| Mauludin dan Ghozi (2021)  | Rancang Bangun Sistem Informasi Jadwal Kegiatan Acara di Dinas Komunikasi Dan Informatika (Dinkominfo) Demak Berbasis Web | Informasi kegiatan masih konvensional dan kurang dinamis | Waterfall                       | Mempermudah penyampaian informasi jadwal kegiatan                                |
| Jayanti dan Khadija (2024) | <i>Business Intelligence Dashboard</i> Data Pengunjung Kearsipan di Dinas Perpustakaan dan Kearsipan Kota Surakarta       | Membutuhkan laporan <i>real-time</i> yang mudah dipahami | Deskriptif                      | Memahami statistik pengunjung, evaluasi program, dan transparansi data meningkat |

| Penulis        | Judul                                                                                                                          | Masalah                                                                                 | Metode                                        | Hasil                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------------------------|----------------------------------------------------------|
| Perdana (2020) | Rancang Bangun Aplikasi <i>Dashboard</i> Pengunjung Museum Teknoform Universitas Dinamika                                      | Laporan pengunjung manual dengan Excel                                                  | Waterfall                                     | Menyajikan info pengunjung dalam bentuk tabel dan grafik |
| Usulan         | Rancang Bangun Sistem Digital Presensi Tamu dengan Integrasi <i>Business Intelligence</i> (Studi Kasus Politeknik Caltex Riau) | Presensi tamu masih manual, data belum terintegrasi, dan belum dianalisis secara visual | <i>Rapid Solo Software Development</i> (RSSD) | Mempermudah pencatatan dan analisis data presensi tamu   |

Dari beberapa penelitian terdahulu yang telah diulas, dapat disimpulkan bahwa pengembangan sistem presensi tamu digital berbasis web terbukti efektif untuk menggantikan proses manual dan memusatkan data. Namun, dari tinjauan yang dilakukan, belum ada penelitian yang mengintegrasikan sistem presensi tamu dengan *Business Intelligence* (BI) dalam penyajian *dashboard* interaktif untuk mendukung analisis tren dan pengambilan keputusan berbasis data. Oleh karena itu, penelitian ini diusulkan untuk mengisi celah tersebut dengan merancang sistem yang tidak hanya berfungsi sebagai alat pencatatan operasional, tetapi juga sebagai alat analisis data yang andal melalui penerapan BI.