

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Pengembangan Sistem Informasi Fasilitas untuk Civitas Academica di Politeknik Caltex Riau memerlukan pemahaman mengenai konsep-konsep dasar teknologi informasi, metodologi pengembangan perangkat lunak, dan instrumen pengujian yang digunakan.

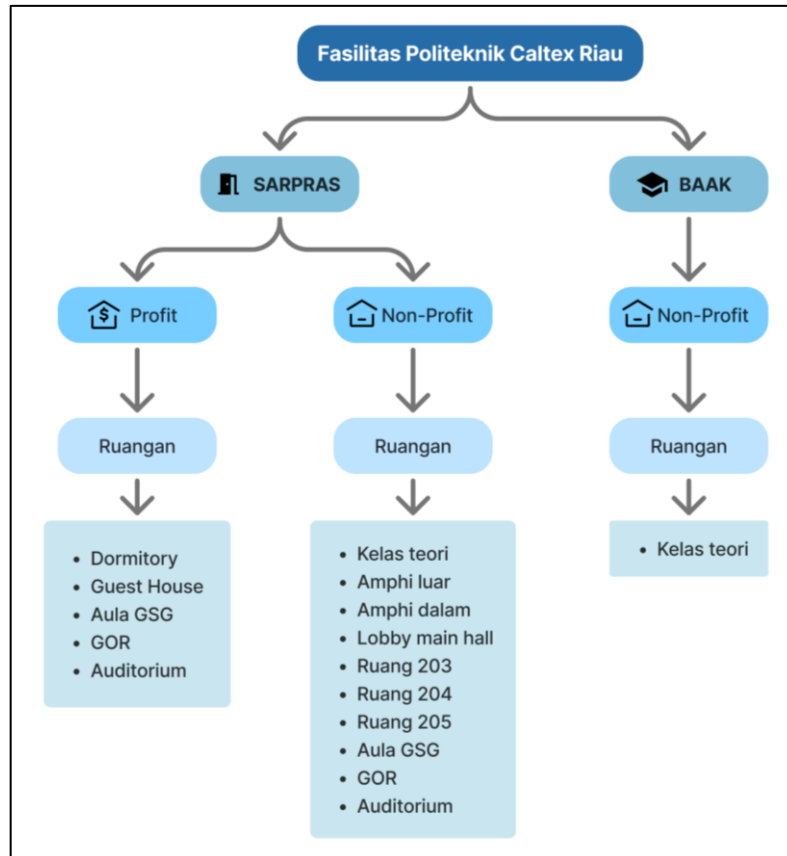
2.1.1 Sistem Informasi

Sistem informasi adalah gabungan dari teknologi dan aktivitas manusia yang dirancang untuk mengumpulkan, menyimpan, memproses, dan menyebarkan informasi serta membantu pengambilan keputusan bagi pengguna. Menurut Afrilliai & Ramadani (2021), sistem informasi adalah serangkaian langkah-langkah yang dilakukan untuk mengolah data mentah menjadi informasi yang berguna. Dalam penelitian ini, sistem informasi dirancang untuk mengelola proses peminjaman fasilitas agar lebih efisien dan transparan.

2.1.2 Peminjaman Fasilitas

Peminjaman adalah kegiatan memberikan atau mengambil sesuatu untuk digunakan sementara waktu. Peminjaman merupakan memberikan izin kepada orang lain untuk menggunakan barang atau aset milik kita untuk sementara waktu, sesuai aturan yang berlaku tanpa perlu membayar. Sedangkan fasilitas adalah sumber daya atau layanan yang disediakan untuk digunakan oleh banyak orang secara gratis (Ramadan dkk., 2024).

Jadi dapat disimpulkan bahwa peminjaman fasilitas adalah proses di mana seseorang meminjam atau memberikan izin penggunaan suatu barang, peralatan, atau ruang tertentu kepada pihak lain untuk jangka waktu tertentu tanpa adanya biaya atau imbalan, dengan mengikuti ketentuan yang telah ditetapkan.



Gambar 2.1 Bagan Fasilitas Politeknik Caltex Riau

Pada Gambar 2.2 (*Politeknik Caltex Riau*, 2026) menunjukkan fasilitas profit yang dikelola oleh SARPRAS mencakup ruangan seperti dormitory, guest house, dan ruangan besar seperti aula Gedung Serba Guna (GSG), Gedung Olah Raga (GOR), dan auditorium yang biasanya disewakan untuk kegiatan eksternal kampus dengan biaya tertentu.

Fasilitas non-profit yang dikelola oleh SARPRAS mencakup ruangan dan fasilitas yang biasanya digunakan untuk kegiatan internal kampus, seperti kelas teori pada saat diluar hari dan jam kerja, amfiteater luar dan dalam, ruangan pertemuan (ruang 203, 204 dan 205), serta ruang serbaguna seperti aula GSG, GOR, dan auditorium. Ruangan ini dapat digunakan tanpa biaya tambahan oleh mahasiswa dan staf untuk kegiatan akademik maupun organisasi. Sedangkan ruangan yang dikelola oleh BAAK fokus pada kebutuhan akademik dasar, yaitu kelas teori. Fasilitas ini mendukung kegiatan pembelajaran reguler yang berlangsung setiap hari tanpa biaya tambahan untuk mendukung proses belajar-mengajar di kampus.

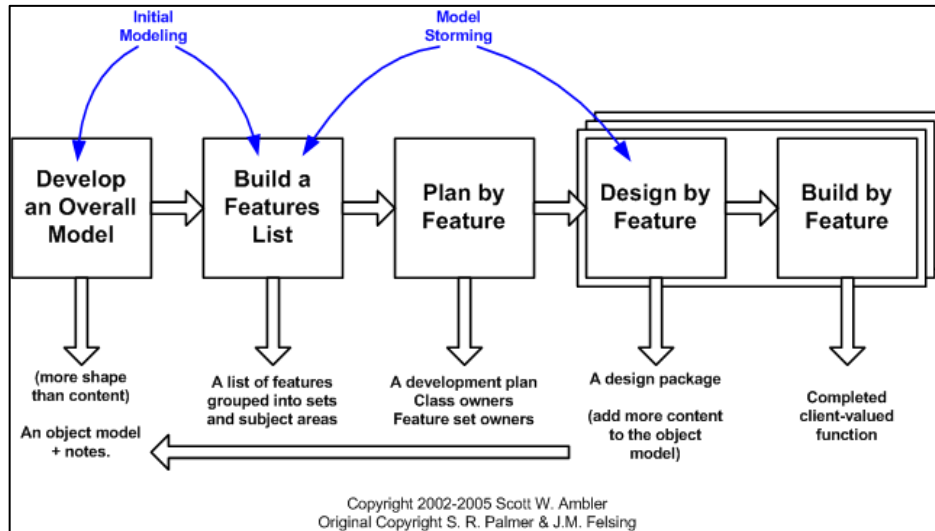
2.1.3 Metode Feature Driven Development (FDD)

Metode *Feature Driven Development* (FDD) adalah metode pengembangan perangkat lunak yang berfokus pada pengembangan fitur-fitur kecil yang bernilai nyata bagi pengguna (*client-valued features*) secara bertahap dan terstruktur (Palmer & Felsing, 2002). FDD menggabungkan praktik terbaik industri dengan proses kerja yang cepat dan fleksibel. Prinsip utamanya adalah selalu melibatkan pengguna (pelanggan) sejak awal pembangunan sistem, sehingga perangkat lunak yang dihasilkan sesuai dengan kebutuhan mereka (Wahyudi dkk., 2022). Selain itu, metode ini juga menekankan pada kerja berdasarkan fitur dan pengiriman hasil kerja secara bertahap.

Istilah model dalam *Develop an Overall Model* merujuk pada representasi struktural dan perilaku sistem secara menyeluruh, yang menurut Palmer dan Felsing (2002) diwujudkan dalam bentuk domain object model, yaitu model yang menggambarkan objek-objek domain bisnis beserta atribut, relasi dan interaksinya, tanpa masih terikat pada detail implementasi teknis. Model ini bersifat *walkthrough-based*, artinya dibangun secara kolaboratif melalui sesi pemodelan bersama domain expert, kemudian dituangkan ke dalam notasi visual (use case, arsitektur dan class diagram) sebagai alat komunikasi antara tim pengembang dan pemangku kepentingan.

Sebagai bagian dari metode *Agile*, FDD memiliki beberapa keunggulan utama (Palmer & Felsing, 2002; Wahyudi dkk., 2022).

- 1) Sederhana dan praktis karena menggunakan proses baku yang ringkas sehingga mudah dipahami dan diimplementasikan oleh seluruh *stakeholder*.
- 2) Berorientasi kepada klien (*client-centric*) karena berfokus pada prioritas fitur-fitur utama yang bernilai bisnis terlebih dahulu, sehingga hasil kerja fungsional dapat diserahkan dan ditinjau lebih cepat.
- 3) Adaptif terhadap perubahan yang memungkinkan penyesuaian kebutuhan selama proses iterasi pengembangan, menjaga relevansi sistem dengan perubahan kebutuhan bisnis.



Gambar 2.2 Tahapan Metode *Feature Driven Development*

Pada Gambar 2.2 menunjukkan 5 tahapan metode FDD, menurut Palmer dan Felsing dalam bukunya yang berjudul *Practical Guide to Feature-Driven Development*, yaitu:

1. *Develop an Overall Model* (Pengembangan Model Objek Menyeluruh)
Tim pengembang (*Chief Architect* dan *Developers*) berkolaborasi dengan pakar bisnis (*Domain Expert*) untuk menganalisis cakupan dan proses bisnis sistem. Pada tahap ini, dibangun *Domain Object Model* (model objek domain makro) yang menggambarkan struktur kelas utama, atribut, dan relasi antaraksi objek secara keseluruhan sebagai fondasi arsitektur sistem.
2. *Build a Feature List* (Penyusunan Daftar Fitur)
Berdasarkan model objek yang telah disepakati, tim mengidentifikasi seluruh fungsi kecil bernilai bisnis yang dibutuhkan oleh pengguna (*client-valued features*). Fitur-fitur ini dikelompokkan ke dalam struktur hierarki tiga tingkat (*Subject Area* → *Feature Set* → *Feature*) dengan syarat estimasi waktu pengerjaan setiap fitur tidak boleh melebihi dua minggu.
3. *Plan by Features*
Development Manager bersama *Chief Programmer* merencanakan urutan eksekusi pengembangan berdasarkan tingkat ketergantungan dan prioritas fitur. Pada tahap ini, dilakukan penetapan jadwal rilis serta alokasi tanggung jawab

kepemilikan kelas (*Individual Class Ownership*) kepada para pengembang (*Class Owner*).

4. *Design by Feature*

Merupakan tahap perancangan teknis terperinci yang bersifat iteratif untuk setiap kelompok fitur. *Chief Programmer* membentuk *Feature Team* dinamis untuk memperbarui spesifikasi kelas dan method (*Class & Method Signatures*) serta melakukan pengecekan desain tampilan (*Design Inspection*) sebelum kode ditulis.

5. *Build by Feature*

Tahap implementasi teknis di mana masing-masing pengembang (*Class Owner*) menuliskan kode program untuk kelas yang menjadi tanggung jawabnya, melakukan pengujian (*testing*), dilanjutkan dengan pemeriksaan kode bersama tim (*Code Inspection*) sampai akhirnya diintegrasikan ke dalam sistem utama (*promote to build*).

2.1.4 Website

Website adalah kumpulan halaman yang berisi berbagai macam informasi, seperti tulisan, gambar, video, dan suara (Istiqomah & Hidayat, 2021). *Website* merupakan sebuah sistem informasi yang terhubung melalui internet. Sistem ini terdiri dari halaman-halaman *web* yang saling terkait. Setiap halaman dapat berisi berbagai jenis data, baik statis maupun dinamis. Desain dan tata letak yang baik pada sebuah *website* sangat penting untuk memberikan pengalaman pengguna yang optimal.

2.1.5 Civitas Academica

Civitas berasal dari bahasa latin yang artinya kewarganegaraan atau komunitas warga negara. Sedangkan *academica* berasal dari bahasa Yunani kuno, yaitu *Akadēmia* yang memiliki arti taman atau hutan kecil yang berada di luar kota Athena tempat filsuf Plato mengajar murid-muridnya. Menurut *Oxford Learner's Dictionaries* (1950), *Academica* berarti dunia pembelajaran, pengajaran, penelitian, dan lainnya di universitas, dan orang-orang yang terlibat di dalamnya. Menurut Kamus Besar Bahasa Indonesia (KBBI) (2016), *civitas academica* adalah kelompok (warga) masyarakat akademik yg terdiri atas dosen dan mahasiswa dengan

perwakilannya yang terbentuk melalui senat masing-masing.

Dari beberapa pengertian di atas, dapat disimpulkan bahwa pengertian *civitas academica* merujuk pada seluruh komunitas akademik di institusi pendidikan, termasuk mahasiswa, dosen, dan staf serta pihak lain yang terlibat dalam kegiatan akademik dan kemahasiswaan.

2.1.6 *Laravel*

Laravel sebuah kerangka kerja atau *framework* PHP *open-source* populer yang dikembangkan oleh Taylor Otwell. *Framework* ini mengadopsi pola arsitektur MVC (*Model-View-Controller*) yang memisahkan logika aplikasi dari antarmuka pengguna (Abyan, 2024). Fitur *routing* pada *Laravel* berperan penting dalam menghubungkan permintaan pengguna dengan bagian yang tepat dalam aplikasi. Hal ini membuat struktur kontroler *Laravel* tidak dapat menerima langsung permintaan pengguna (Yuniarti dkk., 2022).

2.1.7 *MySQL*

MySQL merupakan adalah sebuah sistem manajemen basis data relasional (RDBMS) populer yang menggunakan bahasa *Structured Query Language* (SQL) dibawah lisensi *General Public License* (GPL). *MySQL* sering digunakan dalam aplikasi berbasis *web* karena memungkinkan pengguna untuk mengelola data yang terorganisir dalam bentuk tabel, baris, dan kolom (Yuniarti dkk., 2022). *MySQL* bersifat *open source*, artinya kode sumbernya bisa diakses dan dimodifikasi oleh siapa saja secara gratis.

2.1.8 *Figma*

Figma adalah alat desain berbasis *web* yang digunakan untuk membuat prototipe dan desain antarmuka pengguna (UI). Figma memungkinkan kolaborasi *real-time* antara desainer dan pengembang, sehingga memudahkan proses desain dan pengembangan (Albert dkk., 2021). Figma mendukung berbagai fitur desain seperti pembuatan wireframe, desain antarmuka yang responsif, dan prototipe interaktif. Figma sangat berguna dalam pengembangan aplikasi karena kemampuannya dalam menerjemahkan desain ke dalam kode yang dapat diimplementasikan serta mempermudah validasi konsep sebelum masuk ke tahap implementasi.

2.1.9 *Black Box Testing*

Black box testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa melihat kode sumbernya. Pengujian ini dilakukan dengan memberikan input pada sistem dan memeriksa *output* yang dihasilkan untuk memastikan bahwa aplikasi berfungsi sesuai dengan yang diharapkan. Menurut Suryadi dkk. (2022), pengujian ini penting untuk memastikan bahwa semua fitur yang ada dalam sistem dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna.

Dalam pengujian ini, digunakan *Use Case Testing* karena metode ini memungkinkan pengujian sistem berdasarkan skenario nyata yang dialami oleh pengguna dalam proses peminjaman fasilitas di Politeknik Caltex Riau. Setiap alur fungsional dalam sistem akan diuji sesuai dengan skenario penggunaan yang telah dirancang, seperti proses pengajuan peminjaman, persetujuan oleh pihak terkait, serta pemantauan status peminjaman. Jenis pengujian ini membantu memastikan bahwa semua fitur utama dalam sistem informasi peminjaman fasilitas berbasis *web* berfungsi dengan baik dalam berbagai kondisi penggunaan. Selain itu, metode ini memungkinkan identifikasi dan perbaikan terhadap potensi kesalahan pada alur proses pengguna sebelum sistem diterapkan secara penuh. Keunggulan lain dari *Use Case Testing* adalah kemampuannya dalam memudahkan komunikasi antara pengembang, penguji, dan pemangku kepentingan lainnya, karena skenario pengujian didasarkan pada use case yang telah disepakati bersama.

2.1.10 *Usability Testing*

Usability testing adalah metode evaluasi untuk menilai kemudahan penggunaan suatu aplikasi oleh pengguna. Pengujian ini melibatkan pengguna nyata dalam menyelesaikan tugas-tugas tertentu pada aplikasi untuk mengidentifikasi masalah dalam desain antarmuka dan interaksi pengguna. Tujuan utama *usability testing* adalah mengidentifikasi masalah yang dihadapi pengguna dan meningkatkan pengalaman pengguna (Nuantra dkk., 2022). *Usability testing* dapat dilakukan dengan berbagai metode, termasuk pengujian langsung dengan pengguna, pengujian jarak jauh, dan pengujian berbasis kuesioner.

Dalam pengujian ini, digunakan *System Usability Scale* (SUS) yang

dikembangkan oleh John Brooke pada tahun 1986 dan menjadi salah satu metode yang paling banyak digunakan untuk mengevaluasi kebergunaan produk dan sistem berdasarkan persepsi pengguna. SUS terdiri dari sepuluh pernyataan yang mencakup aspek positif dan negatif dari pengalaman pengguna, sehingga mudah dan cepat untuk diisi oleh responden (Kosim dkk., 2022). Setiap pernyataan dinilai menggunakan skala *Likert*, di mana responden dapat menyatakan seberapa setuju mereka dengan masing-masing pernyataan. Adapun skor yang diberikan pengguna terdiri dari 5 poin yaitu:

Skor 1	: Sangat Tidak Setuju (STS)
Skor 2	: Tidak Setuju (TS)
Skor 3	: Netral
Skor 4	: Setuju (S)
Skor 5	: Sangat Setuju (SS)

Untuk melakukan perhitungan skor SUS dilakukan dengan cara khusus melalui langkah-langkah berikut ini:

1. Untuk pernyataan ganjil (1, 3, 5, 7, 9), nilai responden dikurangi 1.
2. Untuk pernyataan genap (2, 4, 6, 8, 10), nilai 5 dikurangi nilai responden.
3. Hasil dari semua perhitungan dijumlahkan, kemudian dikalikan dengan 2,5 untuk mendapatkan skor akhir dalam rentang 0–100.

Setelah didapatkan hasil SUS yang dihitung menjadi satu nilai tunggal yang berkisar antara 0 hingga 100, yang dapat digunakan untuk menilai kebergunaan sistem dengan ketentuan sebagai berikut.

1. Skor 80 – 100 : Sangat Baik (A) : Sistem sangat mudah digunakan
2. Skor 74 – 79 : Baik (B) : Sistem mudah digunakan
3. Skor 68 – 73 : Cukup (C) : Sistem perlu perbaikan
4. Skor 51 – 67 : Buruk (D) : Sistem tidak dapat diterima
5. Skor 0 – 50 : Sangat Buruk (F) : Sistem sangat tidak dapat diterima



Gambar 2.3 System Usability Score

2.1.11 User Acceptance Testing

Sistem informasi adalah gabungan dari teknologi dan aktivitas manusia yang dirancang untuk mengumpulkan, menyimpan, memproses, dan menyebarkan informasi serta membantu pengambilan keputusan bagi pengguna. Menurut Afrilliai & Ramadani (2021), sistem informasi adalah serangkaian langkah-langkah yang dilakukan untuk mengolah data mentah menjadi informasi yang berguna. Dalam penelitian ini, sistem informasi dirancang untuk mengelola proses peminjaman fasilitas agar lebih efisien dan transparan.

2.2 Penelitian Terdahulu

Penelitian terdahulu mengenai pengembangan sistem informasi berbasis *web* untuk peminjaman fasilitas telah banyak dilakukan oleh berbagai pihak dengan pendekatan yang beragam. Penelitian-penelitian terdahulu digunakan sebagai landasan, perbandingan dan pertimbangan dalam merancang penelitian ini. Berikut ini beberapa referensi dari penelitian terdahulu yang dijadikan rujukan dalam penulisan proyek akhir ini, yaitu:

Penelitian yang dilakukan oleh Melki Abyan (2024) berjudul "Rancang Bangun Sistem Informasi Penyewaan Gedung dan Fasilitas Politeknik Caltex Riau" mengembangkan sistem informasi yang bertujuan untuk mempermudah proses penyewaan gedung dan fasilitas untuk pihak eksternal dengan pendekatan berbasis profit. Penelitian ini menekankan pentingnya sistem yang terintegrasi untuk meningkatkan efisiensi dan transparansi dalam pengelolaan fasilitas di institusi pendidikan, namun penelitian ini belum mencakup aspek non-profit seperti

peminjaman fasilitas untuk kebutuhan internal civitas academica Politeknik Caltex Riau. Oleh karena itu, penelitian ini menjadi dasar untuk membuat sistem peminjaman fasilitas dengan pendekatan non-profit yang lebih sesuai dengan kebutuhan internal institusi.

Selanjutnya pada tahun yang sama, penelitian yang dilakukan oleh Bernadus dkk. (2024) berjudul "Perancangan Sistem Informasi Peminjaman Sarana dan Prasarana Universitas Dhyana Pura Berbasis *Web*" yang menyoroti pentingnya perancangan sistem yang *user-friendly* untuk meningkatkan pengalaman pengguna. Penelitian ini menunjukkan bahwa desain antarmuka yang baik dan navigasi yang intuitif sangat berpengaruh terhadap tingkat adopsi sistem oleh pengguna. Aspek ini menjadi referensi penting bagi peneliti dalam merancang *wireframe* antarmuka sistem agar tidak membingungkan kelompok pengguna saat diimplementasikan. Namun, penelitian ini masih berfokus pada visualisasi peminjaman standar dan belum mengakomodasi siklus pelaporan kondisi fisik fasilitas pasca-penggunaan.

Penelitian yang dilakukan oleh Wulandari dkk. (2023) berjudul "Pengembangan Aplikasi Peminjaman Ruang Berbasis *Web* dengan Metode *Agile Feature Driven Development* pada Universitas Harapan Bangsa" menggunakan pendekatan *Agile Feature Driven Development* dalam pengembangan aplikasi peminjaman ruang. Penelitian ini membuktikan bahwa metode FDD sangat adaptif dalam dinamika perubahan kebutuhan pemangku kepentingan melalui pendekatan berbasis fitur yang iteratif. Sistem yang dikembangkan berhasil mengatasi permasalahan seperti konflik jadwal penggunaan ruangan serta mempercepat proses verifikasi peminjaman. Walau demikian, model arsitektur sistem yang dibangun masih mengandalkan basis data lokal terisolasi tanpa adanya pengamanan otorisasi akses khusus berbasis *email domain* institusi resmi.

Penelitian yang dilakukan oleh Sukmana dkk. (2022) berjudul "Model Aplikasi Peminjaman Ruangan Berbasis *Web* Pada Tingkat Fakultas di Perguruan Tinggi" menekankan pada pengujian fungsional sistem menggunakan metode *Black Box Testing*. Fungsionalitas aplikasi yang dibangun mencakup manajemen jadwal ruangan serta pelaporan kondisi ruangan secara *real-time*. Penelitian ini memberikan dasar kuat bagi peneliti mengenai pentingnya standar validasi input dan pengujian sistem. Meski telah menyentuh aspek pelaporan kondisi ruangan,

sistem tersebut belum menyediakan fitur unggahan bukti visual (foto) sebagai validasi atas status kerusakan aset perkuliahan.

Tabel 2.1 Perbandingan variabel penelitian

Kategori	Peneliti 1	Peneliti 2	Peneliti 3	Peneliti 4
Pendekatan Operasional	Komersial (<i>Profit</i> untuk umum)	Non-Komersial (<i>Non-profit</i> untuk internal)	Non-Komersial (<i>Non-profit</i> untuk internal)	Non-Komersial (<i>Non-profit</i> untuk internal)
Metode Pengembangan	<i>Prototype</i>	<i>Waterfall</i>	<i>Feature Driven Development</i>	<i>Waterfall</i>
Mekanisme Autentikasi	Akun Lokal Standar (<i>Username & Password</i>)	Akun Lokal Standar (<i>Username & Password</i>)	Akun Lokal Standar (<i>Username & Password</i>)	Akun Lokal Standar (<i>Username & Password</i>)
Kelola Pasca-Penggunaan	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
Metode Pengujian	<i>Black Box, User Acceptance Testing, dan Usability Testing</i>	Uji Adopsi & Pengalaman Pengguna (<i>User Experience</i>)	<i>Black Box dan User Acceptance Testing</i>	<i>Black Box Testing</i>

Berdasarkan kajian terhadap beberapa penelitian terdahulu, diketahui bahwa penelitian-penelitian tersebut telah membahas topik yang relevan. Penelitian ini memiliki perbedaan dengan penelitian sebelumnya karena mengkaji permasalahan tata kelola fasilitas internal kampus secara *end-to-end* dengan menyesuaikan pendekatan penelitian terhadap kebutuhan operasional BAAK dan SARPRAS PCR. Kebaruan (*novelty*) dari penelitian ini pada implementasi gerbang login Google OAuth 2.0 untuk menyaring civitas, pemetaan jadwal visual untuk mencegah double-booking, verifikasi penanggung jawab organisasi, serta pelaporan pengembalian ruangan yang didukung bukti foto guna menjaga kualitas aset institusi secara transparan. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata pada digitalisasi manajemen aset perguruan tinggi dalam menjawab perumusan masalah serta melengkapi hasil-hasil penelitian yang telah ada sebelumnya.