

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Dasar**

##### **2.1.1 Konsep Dasar Sistem**

Secara garis besar sistem merupakan kumpulan dari komponen dan unsur-unsur yang saling terintegrasi, komponen-komponen itu terorganisasi dan bekerja sama dalam mewujudkan tujuan tertentu. Menurut Abdul Kadir (2018), Sistem adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai. Adapun suatu sistem mempunyai ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam mengidentifikasi pembuatan sistem. Adapun karakteristik sistem (Hutahaean, 2015:3) yang dimaksud adalah sebagai berikut:

1) **Komponen**

Suatu sistem terdiri dari beberapa komponen berinteraksi, yang berarti bekerja sama untuk membentuk persatuan. Komponen sistem terdiri dari komponen-komponen yang subsistem atau bagian dari sistem.

2) **Batasan sistem (*boundaries*)**

Batas sistem adalah area yang membatasi antara suatu sistem dengan sistem lain atau dengan lingkungannya di luar. Batas sistem ini memungkinkan sistem untuk dilihat sebagai satu kesatuan. Batas-batas suatu sistem menunjukkan ruang ruang lingkup sistem.

3) **Lingkungan luar sistem (*environment*)**

Lingkungan luar sistem (*environment*) berada di luar batas sistem yang mempengaruhi operasi sistem. Lingkungan bisa khasiat yang bermanfaat yang harus dipertahankan dan yang mana merugikan yang harus dijaga dan dikendalikan, jika tidak maka akan terjadi mengganggu kelangsungan sistem.

4) **Sistem penghubung (*interface*)**

Sistem penghubung adalah media penghubung antar satu *sub* sistem dengan subsistem lainnya. Melalui tautan ini memungkinkan sumber daya mengalir dari subsistem ke *sub* sistem lain. *Output* dari *sub* sistem akan menjadi masukan (*input*) ke subsistem lain melalui konektor.

5) Masuk ke sistem (*input*)

*Input* adalah energi yang dimasukkan ke dalam sistem yang dapat berupa pemeliharaan (*maintenace input*), dan input sinyal (sinyal masukan). *Input* pemeliharaan adalah energi dimasukkan agar sistem dapat beroperasi. Sinyal masukan adalah energi yang diproses untuk mendapatkan *output*. Contoh batin sistem program komputer adalah perawatan *input* sedangkan data merupakan sinyal masukan yang akan diolah menjadi informasi.

6) Keluaran sistem (*output*)

*Output* sistem adalah hasil dari energi yang diproses dan diklasifikasikan menjadi keluaran yang bermanfaat dan residu pembuangan. Contoh komputer yang menghasilkan panas adalah sisa pembuangan, sedangkan informasi adalah keluarannya diperlukan.

7) Prosesor sistem

Suatu sistem menjadi bagian dari prosesor yang akan berubah masukan menjadi keluaran. Sistem produksi akan memproses bahan bahan mentah menjadi bahan jadi, sistem akuntansi akan mengolah data tersebut ke dalam laporan keuangan.

8) Tujuan sistem

Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (objektif). Tujuan dari sistem sangat menentukan input dibutuhkan oleh sistem dan keluaran yang akan dihasilkan oleh sistem.

### 2.1.2 Konsep Dasar Informasi

Pengertian informasi secara singkat adalah bahwa informasi adalah hasil pengolahan data yang dibentuk menjadi lebih berguna dan berarti bagi penerimanya. Proses pengumpulan dan pengelolaan data dilakukan untuk memberikan pengetahuan atau informasi tertentu. Oleh karena itu, data merupakan

sumber informasi yang penting. Data itu sendiri merupakan gambaran dari suatu peristiwa atau entitas yang nyata.

Menurut Agus Mulyanto (2009:12) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata. Sedangkan menurut Sutarman (2009:14), informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima. Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Berdasarkan pendapat beberapa ahli, dapat disimpulkan informasi merupakan pengumpulan dan pengolahan data yang diolah agar menjadi sebuah informasi yang bermanfaat ataupun sebagai salah satu pertimbangan dalam pengambilan keputusan.

### **2.1.3 Konsep Dasar Sistem Informasi**

Menurut Kertahadi (2007), sistem informasi merupakan sebuah alat penyajian informasi yang telah dikelola sedemikian rupa agar bermanfaat bagi penerima. Sistem informasi juga bertujuan untuk memberikan informasi perencanaan, awalan, organisasi, dan operasional sebuah perusahaan pelayanan dalam sinergi organisasi.

Menurut Lippeveld, Sauerborn, dan Bodart (2000), sistem informasi merupakan seperangkat komponen berhubungan untuk pengumpulan, penyimpanan, dan pengolahan data hingga menjadi informasi yang siap untuk digunakan.

### **2.1.4 Konsep Dasar LCDP (*Low Code Development Platform*)**

*Low Code Development Platform (LCDP)* merupakan sebuah pendekatan pengembangan dengan cara *drag-and-drop* untuk menyederhanakan pembuatan sebuah aplikasi dengan cara mengurangi penggunaan kode program secara manual sebanyak mungkin. Platform ini menyediakan user interface yang ramah bagi penggunanya untuk mempermudah menggabungkan komponen-komponen dengan

API untuk membuat aplikasi fungsional. Menurut penelitian yang dilakukan oleh Vaibhav Vijay Rajgor dan Sarla Mary dari Mumbai University, India. Dibuatlah sebuah uji coba perbandingan membuat sebuah program dengan menggunakan metode pemrograman biasa dan LCDP. Dapat disimpulkan untuk membuat sebuah aplikasi biasa menggunakan metode pemrograman konvensional membutuhkan setidaknya 50 – 80 *line* kode pemrograman dan menggunakan paling sedikit 3 bahasa pemrograman dan waktu yang diperlukan untuk membuatnya aplikasi ini sekitar 1 hingga 2 jam. Sedangkan, ketika membuat sebuah aplikasi menggunakan LCDP, hanya membutuhkan paling sedikit 3 *line* kode dengan 1 bahasa pemrograman saja dan membutuhkan waktu sekitar 305 – 10 menit.

Selain itu, Menurut *Forester Study*, hari ini sekitar 84% perusahaan telah beralih ke platform kode rendah karena kemampuannya untuk mengurangi jumlah stress pada sumber daya TI, menurunkan biaya, meningkatkan kecepatan ke pasar dan keterlibatan pemangku kepentingan bisnis yang lebih baik. Dengan demikian, dengan tujuan mengatasi kekurangan profesional dan terampil tinggi pengembang itu memungkinkan para profesional tanpa khusus latar belakang pemrograman untuk menambah pengembangan perangkat lunak proses tanpa mengorbankan produktivitas pengembang yang berpengalaman dan terampil.

### **2.1.5 Konsep Dasar Power Apps**

Microsoft PowerPlatform merupakan produk dari *Microsoft Corporation* yang terdiri dari empat aplikasi *Power Automate*, *Power Apps*, *Power BI*, dan *Power Virtual Agents*. Dalam hal ini ini, penggunaan aplikasi dari *Microsoft Power Platform* dapat digabungkan dengan masing-masing lain, memungkinkan perusahaan untuk meningkatkan produktivitas dan efisiensi proses mereka melalui penggunaan yang ditargetkan. Salah satu produk dari *Microsoft PowerPlatform* adalah *Power Apps*. *Power Apps* adalah *platform* yang memungkinkan pengguna untuk mengembangkan aplikasi kecil dengan usaha yang relatif sedikit. Hal ini memungkinkan orkestrasi dari aplikasi sesuai dengan jenis kit konstruksi. Aplikasi dikembangkan dengan cara yang disederhanakan seperti desain aplikasi visual, dan tanpa klasik, pemrograman manual. Aplikasi ini dapat tersambung dengan berbagai

jenis data *source online* maupun *local* seperti *Cloud, Share Point, Microsoft 365*, SQL bahkan bisa terkoneksi dengan API dan lain sebagainya.

### **2.1.6 Konsep Dasar SDLC (*System Development Life-Cycle*)**

SDLC (*Software Development Life Cycle*) adalah kerangka kerja atau model manajemen proyek terstruktur yang menguraikan fase-fase yang diperlukan untuk membangun sistem TI, dari awal hingga hasil akhir. SDLC adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem rekayasa perangkat lunak. SDLC terdiri dari beberapa tahapan, yaitu perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Tahapan-tahapan ini membantu pengembang untuk memahami kebutuhan pengguna, merancang solusi, mengimplementasikan solusi, menguji solusi, dan memelihara solusi. SDLC membantu pengembang untuk memastikan bahwa solusi yang dibangun memenuhi kebutuhan pengguna, berfungsi dengan baik, dan mudah dipelihara. SDLC juga membantu pengembang untuk mengelola risiko dan memastikan bahwa proyek selesai tepat waktu dan sesuai anggaran. Menurut Prof. Dr. Sri Mulyani, AK., CA. (2017) SDLC adalah proses logika yang digunakan oleh seorang analis sistem untuk mengembangkan sebuah sistem informasi yang melibatkan *requirements, validation, training* dan pemilik sistem. Terdapat beberapa tahapan pada SDLC adalah sebagai berikut:

1) Perencanaan (*Planning*)

Tahap di mana project leaders mengevaluasi persyaratan yang ada dalam sebuah proyek, membuat jadwal dengan tujuan target, membentuk tim serta struktur kepemimpinan untuk proyek tersebut.

2) Analisis (*Analysis*)

Tahap di mana sistem akan dianalisis bagaimana akan dijalankan nantinya.

3) Desain (*Design*)

Tahap di mana pengembang merancang solusi untuk memenuhi kebutuhan pengguna.

4) Implementasi (*Implementation*)

Tahap di mana pengembang mengimplementasikan solusi yang telah dirancang.

5) Pengujian (*Testing*)

Tahap di mana pengembang menguji solusi untuk memastikan bahwa solusi berfungsi dengan baik.

6) Pemeliharaan (*Maintenance*)

Tahap di mana pengembang memelihara solusi untuk memastikan bahwa solusi tetap berfungsi dengan baik dan memenuhi kebutuhan pengguna

### 2.1.7 Konsep Dasar OOAD (*Object Oriented Analysis and Design*)

Pengertian OOAD oleh Mathiassen, *Object Oriented Analysis Diagram* (OOAD) ialah sebuah pendekatan untuk melaksanakan pengembangan sistem berdasarkan objek-objek (Purwaningtias, 2018).

Konsep OOAD mencakup analisis dan perancangan sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA merupakan metode analisis yang mengkaji kebutuhan (requirements/persyaratan) yang harus dipenuhi oleh suatu sistem) dari sudut pandang kelas dan objek yang dihadapi dalam lingkup perusahaan. Sedangkan OOD adalah metode untuk mengarahkan arsitektur perangkat lunak berdasarkan manipulasi objek sistem atau subsistem.

Hal tersebut diimplementasikan di seluruh siklus pengembangan sistem, yang meningkatkan kualitas sistem dan bahkan mendorong keterlibatan dan komunikasi pengguna. Metodologi Pengembangan OOAD memiliki 3 karakteristik yaitu :

1) *Encapsulation* (Pengkapsulan)

- *Encapsulation* merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
- Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.
- Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

2) *Inheritance* (Pewarisan)

- *Inheritance* adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung.

- Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya.
- *Inheritance* mempunyai arti bahwa atribut dan operasi yang dimiliki bersama diantara kelas yang mempunyai hubungan secara hirarki.
- Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.
- Kelas Objek dapat didefinisikan atribut dan service dari kelas Objek lainnya.
- *Inheritance* menggambarkan generalisasi sebuah kelas.

### 3) *Polymorphism* (Polimorfisme)

- *Polimorfisme* yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.
- *Polimorfisme* mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.
- Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama.
- Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Objek.

## 2.1.8 *Unified Modelling Language* (UML)

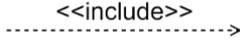
*Unified Modelling Language* (UML) desain mata yang mengatakan dan melukiskan serta mencatat perancangan sistem informasi yang menggandrungi paradig objek. Dalam UML, didalamnya ada diagram yang bisa digunakan untuk menyederhanakan pihak dalam pemodelan sistem untuk dikembangkan. (Waruwud dan Nasution, 2018). Berikut adalah diagram-diagram UML yang akan dibuat menggunakan pendekatan OOAD:

### 1) *Use Case Diagram*

Menurut Munawar (2018:89), Salah satu jenis diagram pada UML yang menjelaskan hubungan sistem dengan aktor berperan dalam sistem. Bekerja

dengan cara menjelaskan tipikal hubungan melalui sebuah cerita dengan simbol. Simbol terdapat pada pendekatan OOAD *Use Case Diagram*.

**Tabel 2.1 Use Case Diagram Symbol**

Simbol	Nama	Keterangan
	Aktor	Aktor merupakan pelaku utama yang menggambarkan siapa yang berinteraksi dengan sistem.
	Use Case	Bagian dari fungsionalitas sistem yang menggambarkan abstraksi dan interaksi antara aktor dan sistem
	Boundaries	Berisi nama sistem yang ditempatkan pada atau di atas batas. Mewakili jangkauan sistem dan aktor berada di luar cakupan sistem
	Association	Menghubungkan aktor dengan use case dan menunjukkan komunikasi dua arah
	Generalitation	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur dari data objek yang ada di atasnya objek induk ( <i>ancestor</i> )
	Include	Behavior atau perilaku yang harus terpenuhi agar event dapat terjadi

---

<code>&lt;&lt;extend&gt;&gt;</code> ----->	<i>Extend</i>	Tambahkan fungsionalitas dari <i>use case</i> lain jika suatu kondisi terpenuhi
---	---------------	---

---

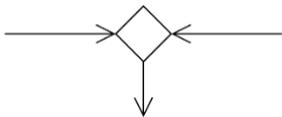
Sumber : Dokumentasi Pribadi

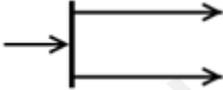
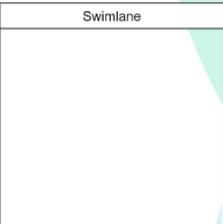


## 2) Activity Diagram

Alur kerja diagram menjelaskan aktivitas komponen dan pengguna secara berurutan. *Activity diagram* bermanfaat guna menunjukkan aksi yang terjadi dalam suatu langkah sistem dengan menampilkan aktivitas pengguna dengan sistem secara berurutan. Berikut adalah penggunaan simbol dalam pendekatan OOAD *Activity Diagram*.

**Tabel 2.2 Activity Diagram Symbol**

Simbol	Nama	Keterangan
	<i>Initial State</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	<i>Final State</i>	Status akhir aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Activity</i>	<i>Activity</i> yang dilakukan oleh sistem, biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Merge</i>	<i>Merge</i> berperan untuk menggabungkan kembali aliran kerja yang terpisah.

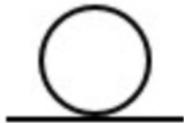
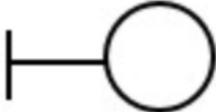
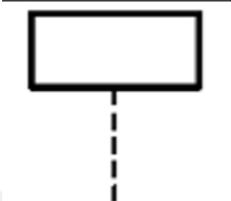
Simbol	Nama	Keterangan
	<i>Association</i>	Menghubungkan aliran <i>activity</i> satu dengan <i>activity</i> lainnya
	<i>Synchronization Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel
	<i>Synchronization Join</i>	Digunakan untuk menunjukkan kegiatan yang digabungkan
	<i>Swimlane</i>	Memisahkan antara aktor satu dengan yang lain

Sumber : Dokumentasi Pribadi

### 3) *Sequence Diagram*

Deskripsi dan tampilan interaksi objek pada sistem secara detail dapat terlihat pada diagram ini. *Sequence diagram* bermanfaat untuk memvisualisasikan dengan detail keterkaitan antar objek dalam suatu sistem. Objek yang berinteraksi pada proses pengoperasian sistem secara umum disusun dari kiri ke kanan. Simbol pendekatan OOAD *Sequence Diagram*.

**Tabel 2.3 Sequence Diagram Symbol**

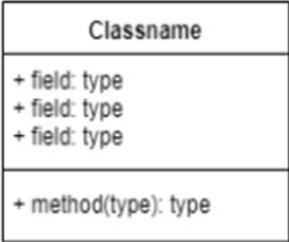
Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Entity</i> <i>Class</i>	Menggambarkan hubungan yang akan dilakukan.
	<i>Boundary</i> <i>Class</i>	Menggambarkan sebuah gambaran dari antarmuka
	<i>Control</i> <i>Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel
	<i>Life Line</i>	Menggambarkan <i>entitas</i> tunggal pada sebuah <i>sequence</i> diagram

Sumber : Dokumentasi Pribadi

#### 4) *Class Diagram*

Menurut Munawar (2018:101), Tampilan kelas atau paket sebagaimana ada pada sistem yang dikembangkan dapat dilihat pada *Class diagram*. *Class diagram* menjelaskan atribut, operasi dan juga *constraint* ada pada sistem. Berikut adalah simbol yang digunakan dalam perancangan *class diagram* dapat terlihat pada **Tabel 2.4**.

**Tabel 2.4 Class Diagram Symbol**

Simbol	Nama	Keterangan
	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	<i>Agregation</i>	Relasi antar kelas dengan makna semua-bagian(whole-part)
	<i>Direct Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	<i>Generalitation</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum – khusus)
	<i>Class</i>	Kumpulan objek yang memiliki attribute dan operation yang sama. Attribute, merupakan property dari sebuah class.

Sumber : Dokumentasi Pribadi

## **2.2 Literatur Review**

### **1. “Analisis Dan Perancangan Sistem Informasi Layanan Perizinan Dan Monitoring Berkas Pada Pelayanan Terpadu Satu Pintu Kanwil Kemenag Provinsi Jambi”**

Merupakan hasil studi yang dilakukan oleh Erica Alvionita dan Kondar Siahaan yang dipublikasikan oleh ejournal STIKOM pada tahun 2021. Penelitian ini membahas masalah yang terkait Proses pembuatan izin yang berjalan pada PTSP Kanwil Kemenag masih mengharuskan pemohon untuk datang langsung ke kantor Pelayanan Terpadu Satu Pintu Kanwil Kemenag untuk dapat memproses perizinannya, dimulai dari pengambilan nomor antrian, pengisian formulir serta penginputan berkas secara manual ke loket yang tersedia. Proses ini dirasa masi kurang memenuhi tujuan Layanan dari Pelayanan Terpadu Satu Pintu Kanwil Kemenag Provinsi Jambi yang bertujuan mensederhanakan prosedur layanan pada Kanwil Kementerian Agama Provinsi Jambi dan kurang efektif bagi pemohon yang memiliki kendala untuk hadir secara langsung karena tidak ada pilihan pendaftaran secara online. Berdasarkan identifikasi masalah yang sudah dijelaskan, dibuatlah penelitian untuk merancang prototype Sistem Informasi Layanan Perizinan pada PTSP Kanwil Kemenag Provinsi Jambi. Pada rancangan sistem ini dibuat alur bisnis proses beserta mockup agar memudahkan penggunaanya membaca flow sistem.

### **2. “Rancang Bangun Aplikasi Sistem Pelayanan Masyarakat Berbasis Web Pada Kecamatan Cadasari Kabupaten Pandeglang”**

Merupakan hasil studi yang dilakukan oleh Gabriel Ryfan K dan Drs.AtiZaidiah,S.Kom.,MTI yang dipublikasikan oleh SENAMIKA pada tahun 2020. Penelitian ini membahas masalah yang terjadi pada sistem pelayanan masyarakat di Kecamatan Cadasari Kabupaten Pandeglang. pelayanan administrasi terpadu oleh Kecamatan Cadasari masih belum memadai. Untuk saat ini pelayanan hanya dilakukan di kantor Kecamatan Cadasari karena segala jenis pelayanan yang ditawarkan belum terkomputerisasi. Hal ini berarti bahwa setiap masyarakat yang ingin menerima pelayanan harus mengunjungi kantor

kecamatan. Karena hal tersebut, banyak data penambahan penduduk yang masih belum tercatat dengan baik. Berdasarkan identifikasi masalah yang sudah dijelaskan, maka dibuatlah penelitian ini yang bertujuan untuk merancang aplikasi sistem pelayanan masyarakat berbasis web pada Kecamatan Cadasari . Dari penelitian juga di bahas proses analisis kebutuhan sistem, Use Case Diagram, dan Class Diagram yang nantinya akan menjadi acuan pembuatan sistem pelayanan masyarakat.

### **3. “Penggunaan Metode SDLC Waterfall Dalam Pembuatan Program Pengajuan Kartu Kredit”**

Merupakan hasil penelitian yang dilakukan oleh Redi Nurhidayat, Nani Agustina dan Entin Sutinah pada tahun 2020. Penelitian ini dipublikasi oleh Journal STMIK Jakarta. Penelitian ini membahas terkait permasalahan pada PT. Bank Tabungan Negara KK Galaxy Bekasi yang dimana proses pengajuan kartu kredit masih dilakukan secara manual dengan mengisi data berupa formulir dengan menggunakan kertas dan butuh waktu lama untuk prosesnya karena harus menunggu persetujuan dari kantor pusat. Hal tersebut Kelemahan pertama adalah human error terjadi jika karyawan atau nasabah mengisi formulir dan data yang di masukan salah maka karyawan atau nasabah harus mengulang dari awal lagi. Kelemahan kedua adalah waktu yang dihabiskan di pengisian kertas formulir akan memakan waktu lama dan apabila keadaan kantor sedang banyak nasabah, maka nasabah yang lain juga harus mengantri dan memakan waktu cukup lama. Dengan sistem yang masih dilakukan secara manual, sering terjadi beberapa masalah seperti karyawan kurang mengetahui informasi tentang proses pembuatan kartu kredit sudah sejauh mana serta pendataan yang dilakukan dengan manual dapat memicu timbulnya *human error* lebih besar. Berdasarkan identifikasi masalah yang sudah dijelaskan, dibuatlah penelitian untuk merancang sebuah Sistem Informasi untuk membantu mempermudah pengajuan pembuatan kartu kredit di PT. Bank Tabungan Negara KK Galaxy. Di dalam penelitian ini dibahas metode yang digunakan untuk proses pembuatan aplikasi dengan metode SDLC dan membuat rencana desain dengan menggunakan pemodelan *visual Unified Modeling Language* (UML). Harapan dari sistem informasi yang

dibuat adalah untuk mempermudah proses pengajuan pembuatan kartu kredit pada PT. Bank Tabungan Negara KK Galaxy.

#### **4. “Automasi Travel Spend Dalam Procurement Menggunakan *Microsoft Power Platform*”**

Merupakan hasil studi yang dilakukan oleh Feri Setiyo Aji pada tahun 2022 dan dipublikasikan oleh *e-prints* Universitas Muhammadiyah Surakarta. Dalam penelitian ini membahas permasalahan terkait dokumen kesepakatan atau *agreement letter*, yang diserahkan kepada vendor masih melalui proses manual dan memiliki volume permintaan yang sangat tinggi, sehingga kegiatan yang dilakukan tidak efektif. Berdasarkan masalah tersebut, dibuatlah penelitian pengembangan aplikasi Automasi Travel Spend Dalam Procurement Menggunakan *Microsoft Power Platform*. Penelitian ini dibuat untuk meminimalisir waktu pembuatan, serta proses yang ada dalam pembuatan dokumen *agreement letter* oleh *travel specialist*. Metode pengembangan aplikasi yang digunakan dalam penelitian ini yaitu metode *prototyping*, dimana dalam penelitian ini penulis menggunakan *Microsoft Power Platform* (*Microsoft Power Apps* dan *Microsoft Power Automate*) untuk membuat *user interface* dan *flow* untuk automasi proses yang berjalan. Hasil dari penelitian ini adalah, aplikasi yang dapat membuat dokumen kesepakatan atau *agreement letter* dalam satu klik, serta dapat secara otomatis melakukan *follow-up* kepada *approver* sehingga dapat memudahkan tugas dari *travel specialist* dalam pembuatan dokumen *agreement letter*.

#### **5. *Implementation of a mobile business application built in Microsoft Power Platform.***

Merupakan hasil tesis yang dibuat oleh Heejin Moon pada tahun 2020 dan dipublikasikan oleh Universitas of Applied Science. Tesis ini membahas tentang mengeksplorasi bidang pembuatan aplikasi bisnis yang relatif baru yaitu LCDP (*Low-Code Development Platform*). Tesis ini sebagian besar membahas sumber daya yang tersedia untuk mempraktikkan LCDP seperti *Microsoft Power Platform* yaitu *Power Apps* *Microsoft* dan *Power Automate*. Berdasarkan hasil penelitian yang diungkapkan oleh Heejin Moon pada jurnalnya yang berjudul “*Implementation of a mobile business application built in Microsoft Power Platform.*” Dapat disimpulkan bahwa penggunaan *Microsoft power platform* seperti *Power Apps* sangat membantu dalam pembuatan aplikasi dengan fitur-

fitur yang tersedia, pembuatan aplikasi menjadi lebih mudah dan efisien. Microsoft juga sudah menyiapkan template yang siap digunakan pada Power Apps, yang hanya perlu diisi dan disiapkan dengan benar dengan sumber data, dan siap digunakan sebagai aplikasi yang berfungsi penuh. Dengan menggunakan template ini, tidak jarang aplikasi dapat diselesaikan dengan cepat. Inti dari tesis ini adalah untuk memperkenalkan metode pengembangan aplikasi seluler yang relatif baru (dan terus dikembangkan). Ini adalah solusi yang sangat valid untuk orang yang kurang paham teknologi, pengembang baru/tidak terampil, atau perusahaan pemula dengan rendah anggaran, kerangka waktu yang singkat, tingkat persyaratan yang rendah, dll.

