

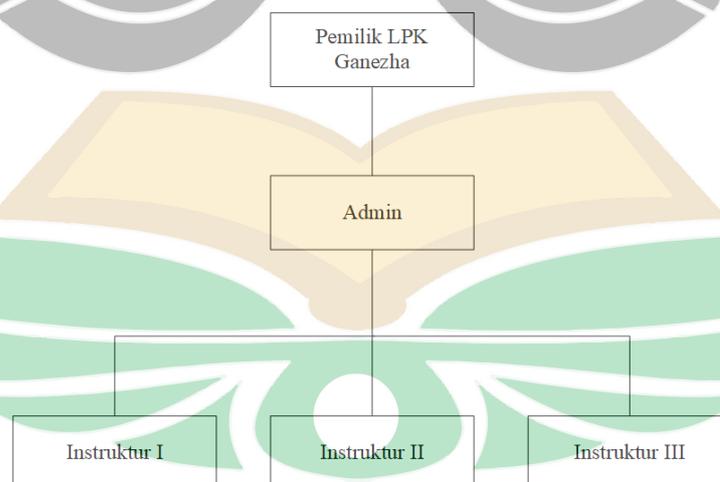
BAB 2 TINJAUAN PUSTAKA

www.itk.ac.id

Pada bab ini berisi kajian tinjauan pustaka yang membahas seputar dasar teori, metode yang digunakan dan penelitian terdahulu sebagai bahan pendukung selama penelitian sistem informasi manajemen yang akan dibuat.

2.1 LPK Ganezha

LPK Ganezha merupakan Lembaga Pelatihan Keterampilan mengemudi kendaraan roda empat yang berada di kota Balikpapan, Kalimantan Timur. LPK Ganezha terletak di Jl. A. Yani, No. 1a, Karang Bugis, Kelurahan Mekar Sari, Balikpapan Kota. LPK Ganezha memiliki metode pendidikan dan pelatihan yang dirancang dengan pola sistematis, mudah dipahami dan selalu berkembang mengikuti kebutuhan dunia kerja saat ini serta didukung oleh instruktur berpengalaman dengan jaminan bisa mengemudi sampai lancar. Awal mula LPK Ganezha bernama LPK Ganezha Manuntung yang dipimpin oleh Bapak Kartono dan didirikan sendiri oleh beliau pada tahun 2010. Berselang enam tahun, LPK Ganezha Manuntung berganti nama menjadi LPK Ganezha dan dipimpin oleh Ibu Yuli pada tahun 2016. Selama dibawah pimpinan Ibu Yuli, LPK Ganezha memiliki struktur organisasi seperti gambar 2.1.



Gambar 2.1 Struktur Organisasi LPK Ganezha

Gambar 2.1 merupakan struktur organisasi LPK Ganezha. LPK Ganezha memiliki struktur organisasi yang dipimpin oleh pemilik LPK Ganezha. Admin sebagai bawahan dari pemilik LPK Ganezha yang mengatur segala kegiatan administrasi di LPK Ganezha, sekaligus membawahi semua instruktur mengemudi yang ada di LPK Ganezha.

2.2 Sistem Informasi Manajemen

Koleksi dari berbagai macam interaksi sistem informasi untuk merencanakan dan mengendalikan data dalam menyediakan informasi untuk semua tingkatan dalam manajemen disebut Sistem Informasi Manajemen (SIM). SIM adalah sistem desain yang biasa digunakan dalam sebuah organisasi yang berkaitan dengan sumber daya, berkas, teknologi dan akuntansi manajemen dalam pengendalian strategi internal dalam bisnis (Hidayat *dkk.*, 2020).

SIM dirancang khusus untuk pemrosesan data analitis operasional dengan menganalisis informasi operasional yang disimpan dalam *database*. SIM sangat membantu dalam pengambilan keputusan suatu perusahaan karena dapat melihat dan menganalisis data dalam bentuk yang sesuai, akurat, dan rinci sehingga dapat memperoleh laporan berupa informasi yang diperlukan untuk manajemen perusahaan membuat keputusan manajemen (Aleksandrova, Vasiliev dan Aleksandrov, 2020).

2.3 Laravel

Laravel merupakan salah satu *framework* yang paling banyak digunakan dalam bahasa pemrograman PHP. *Framework* PHP ini menyediakan kode yang telah dirangkai, cukup melihat modul yang sudah disediakan Laravel dan menggunakannya berdasarkan kebutuhan. Merancang *web* dengan *framework* Laravel sangat mudah dan sederhana, tetapi satu yang harus diketahui yaitu dasar bahasa pemrograman PHP agar dapat memahami struktur *framework* Laravel (Yadav, Rajpoot dan Dhakad, 2019).

Laravel merupakan salah satu *web framework* yang banyak digunakan pengembang *web* karena memiliki aspek unik yang berarti kodenya tertata rapi dan ekspresif. Laravel dibangun dengan menggunakan ulang komponen dari *framework*

PHP sebelumnya, seperti *CodeIgniter*, *Yii* dan bahasa pemrograman lain seperti *Ruby on Rails*. Laravel terkenal dengan ketahanannya dan mengikuti pola desain MVC (*Model View Controller*) serta memiliki serangkaian fitur yang dapat meningkatkan kecepatan *web*. MVC dibagi menjadi tiga komponen yang disebut *Model* yaitu mengungkapkan area logika, *View* yaitu menyajikan antarmuka pengguna, dan *Controller* yang mengelola perubahan pada *View*. Desain MVC adalah cara yang efektif dan terbukti untuk menghasilkan modular, aplikasi terorganisir. MVC membantu mengurangi kompleksitas desain arsitektur dan meningkatkan fleksibilitas kode dan mudah digunakan kembali. Ketika merancang sebuah *website*, Laravel menawarkan beberapa keuntungan, seperti pembuatan aplikasi *web* yang menggunakan Laravel lebih terukur, kebutuhan waktu untuk membuat aplikasi lebih singkat karena komponen dapat digunakan kembali, termasuk *namespace* dan antarmuka yang membantu untuk mengatur dan mengelola sumber daya. (Sendiang dkk., 2018; Sunardi, 2019; Zanin and Wernke, 2019).

2.4 MySQL

Salah satu Sistem Manajemen Data Relasional (RDBMS) yang paling umum digunakan yaitu MySQL. MySQL adalah server *database* relasional yang mendukung bahasa *database* SQL (*Structured Query Language*). Tujuan utama dikembangkannya MySQL yaitu agar dapat mengelola data secara efektif dan aman sebagai pilihan untuk pengguna *database* rumahan dan professional.

MySQL tercatat sebagai penyimpanan data yang paling banyak digunakan untuk aplikasi web karena cepat dalam membaca data. Popularitas dari MySQL juga didukung oleh biaya implementasi yang rendah dan sebuah proyek sumber terbuka. MySQL umumnya untuk aplikasi web karena memiliki fitur yang mengoptimasi *web* seperti HTML, tipe data, dan tersedia secara gratis (Ongo dan Kusuma, 2018; Altıntaş, 2019; Ohyver dkk., 2019).

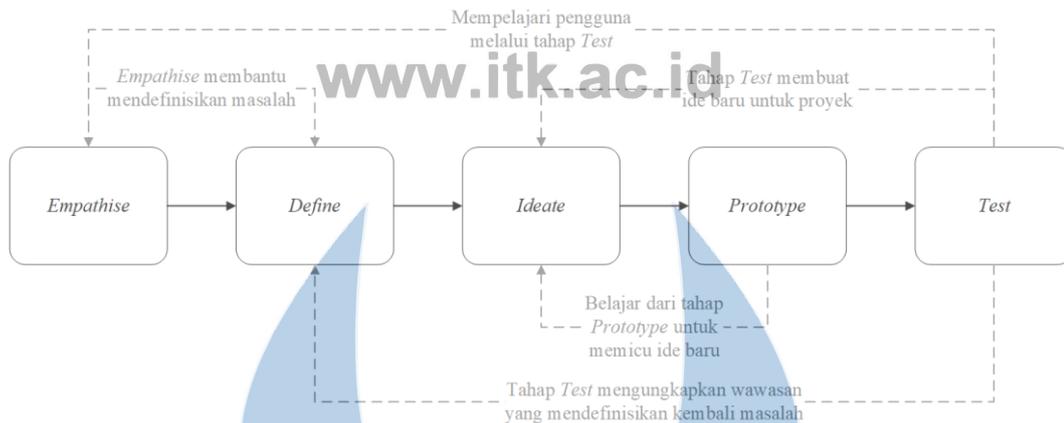
2.5 Design Thinking

Selama kurang lebih lima belas tahun terakhir, *Design Thinking* telah menarik banyak perhatian para akademisi dan banyak dianut serta dikembangkan oleh

banyak praktisi. Pada intinya *Design Thinking* berpusat pada manusia atau pengguna. *Design Thinking* adalah cara yang dapat mendorong suatu pemikiran praktisi agar terbantu menemukan sebuah solusi untuk banyak tantangan. Banyak perusahaan dan perancang *web* menggunakan *Design Thinking* sebagai cara untuk memahami pelanggan dan harapan mereka yang sangat membantu meningkatkan sudut pandang untuk pengalaman pengguna (Goeva, 2019; Marion dkk., 2021a).

Design Thinking juga bisa disebut sebagai kerangka pemikiran, suatu alat, dan metodologi yang didedikasikan untuk menemukan masalah yang ada, mengeksplorasi akar penyebab masalah tersebut, dan menghasilkan solusi inovatif. Metode *Design Thinking* memiliki proses berulang dengan tujuan agar dapat memahami pengguna, memperkuat asumsi, dan mendefinisikan kembali permasalahan dalam upaya untuk mengidentifikasi strategi dan solusi alternatif. Pada saat yang bersamaan, *Design Thinking* menyediakan yang berbasis solusi untuk memecahkan masalah yang dapat berpikir dan bekerja sebanding dengan kumpulan metode *hands-on* (keterlibatan pribadi secara langsung). Metode ini beredar pada orang-orang yang memiliki minat secara mendalam untuk mengembangkan pemahaman ketika merancang produk atau jasa (Guo dan Wang, 2020; Interaction design foundation, 2020).

Model terbaru dari *Design Thinking* dikembangkan di Universitas *Stanford*, lalu disempurnakan oleh IDEO, dan didukung oleh *Hasso Plattner*, salah satu pendiri SAP. Ada banyak variasi dari proses *Design Thinking* yang digunakan saat ini, dari proses yang berisi tiga hingga tujuh fase, tahapan, atau mode. Namun, semua variasi *Design Thinking* tersebut memiliki prinsip sangat mirip. Pada penelitian ini akan fokus pada model lima fase dari *Hasso Plattner* dari Institut Desain (juga dikenal sebagai *d.school*) yang ditemukan pada di Universitas *Stanford* pada tahun 2005. Model tersebut dipilih karena sudah melewati pengujian pada lingkungan *engineering* dan telah berada di garis terdepan dalam menerapkan dan mengajarkan pemikiran desain. Pendekatan *Design Thinking* didefinisikan oleh *d.school* sebagai *framework* “proses kerja secara konstan” dalam mode pengerjaan. Pada gambar dibawah ini adalah bentuk proses dari fase *Design Thinking* yang terdiri dari lima langkah. (Pham, Fucci dan Maalej, 2018; Alhazmi dan Huang, 2020; Interaction design foundation, 2020).



Gambar 2.2 Metode *Design Thinking* (Interaction design foundation, 2020)

Gambar 2.2 merupakan metode *Design Thinking* yang berisi lima tahap yaitu *empathise*, *define*, *ideate*, *prototype*, dan *test*. Berikut ini penjelasan dari lima tahap yang ada dalam metode *Design Thinking*.

2.5.1 *Empathise*

Empathise merupakan dasar dari *Design Thinking* untuk menetapkan permasalahan yang berguna di beberapa tahap berikutnya. Permasalahan diperoleh desainer dengan cara memahami pengguna melalui proses empati kepada pengguna agar dapat merancang solusi yang tepat dan sesuai dengan kebutuhan pengguna. *User research* atau meneliti pengguna merupakan inti dari tahap *empathise* karena menghasilkan hal-hal penting berupa informasi langsung dari pengguna. Informasi dapat diperoleh dengan cara melakukan wawancara, diskusi grup, menyebar survey, dan studi lapangan. Setelah itu, informasi dikumpulkan sebagai data yang berguna untuk memenuhi kebutuhan pengguna (Pham, Fucci dan Maalej, 2018; Shan, Neo dan Yang, 2021).

2.5.2 *Define*

Pada tahap ini, desainer memerlukan pemahaman secara mendalam dari pengguna untuk mengidentifikasi permasalahannya. Pemahaman dilakukan dengan cara menyatukan temuan atau wawasan yang ada pada tahap sebelumnya. Saat yang bersamaan, perancang juga mendefinisikan permasalahan berdasarkan sudut pandang pengguna. Setelah itu, perancang bisa memperoleh sebuah tantangan yang

spesifik dan bermakna agar menjadi acuan di tahap selanjutnya (Pham, Fucci and Maalej, 2018; Shan, Neo dan Yang, 2021).

2.5.3 *Ideate*

Tantangan yang ditetapkan pada tahap sebelumnya merupakan sebuah titik awal bagi desainer untuk mencari solusi. Pada tahap ini desainer akan melakukan *brainstorming* guna menghasilkan ide-ide baru. Keberagaman ide yang didapat dapat menjadi solusi yang sesungguhnya untuk memecahkan masalah secara tepat dan tahu harus mulai darimana. Sementara mencari solusi, desainer akan menemukan ide-ide baru. Ide-ide tersebut akan menjadi *high-fidelity mockup* untuk dilanjutkan pada tahap selanjutnya (Pham, Fucci dan Maalej, 2018; Nasution and Nusa, 2021).

2.5.4 *Prototype*

Ide solusi yang telah dipilih akan direalisasikan. Realisasi dilakukan dengan cara mengembangkan ide solusi menjadi prototipe. Setelah prototipe dibuat, akan ditambahkan interaksi sehingga menghasilkan produk nyata dalam bentuk prototipe. Tahap ini bertujuan untuk mempelajari interaksi pengguna ketika menguji prototipe di tahap selanjutnya dan mendorong pemahaman lebih dalam agar membentuk solusi yang berhasil (Nasution dan Nusa, 2021; Shan, Neo dan Yang, 2021).

2.5.5 *Test*

Tahap terakhir yaitu menguji prototipe yang telah dibuat pada tahap sebelumnya kepada target pengguna. Pengujian *prototype* dilakukan kepada pengguna agar dapat memperoleh umpan balik berupa saran dan masukan agar produk bisa dikembangkan lebih baik lagi sesuai kebutuhan pengguna (Nasution dan Nusa, 2021; Shan, Neo dan Yang, 2021).

2.6 *Personal Extreme Programming*

Salah satu metode yang ada dalam pengembangan perangkat lunak yaitu metode *Agile*. Kata *Agile* yaitu lebih cepat dan lebih tepat, mudah, bebas bergerak,

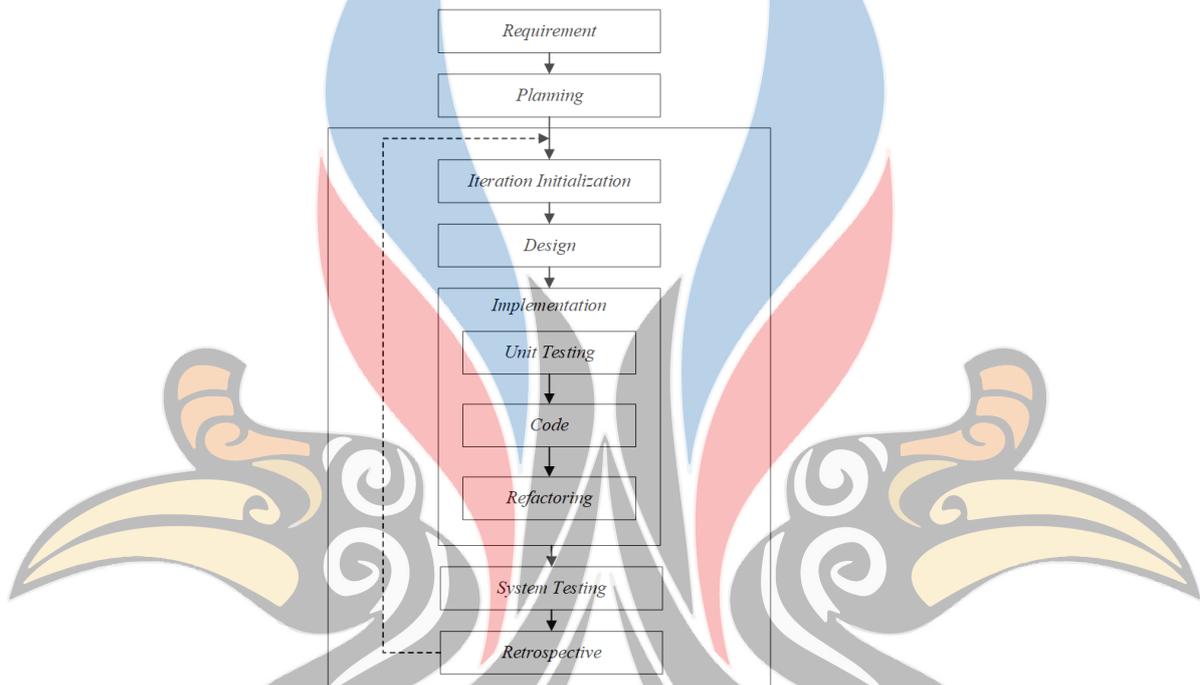
dan perlu hati-hati. Istilah *Agile* digunakan untuk menggambarkan sebuah konsep model proses yang berbeda dari yang lain. Belakangan ini, metode *Agile* secara luas diterapkan dalam industri teknik perangkat lunak. Metode ini juga mengandung sekelompok metode pengembangan perangkat lunak secara iterative dan bertahap (Legowo, Indiarto dan Prayitno, 2019).

Metode *Agile* yang terkenal dan tidak asing di kalangan pengembang yaitu *Scrum*, *ScrumBan*, *Kanban*, dan *Extreme Programming*. Salah satu dari keempat metode tersebut akan digunakan pada penelitian ini yaitu *Extreme Programming*. *Extreme Programming* lebih menghemat waktu karena faktanya akan berfokus pada pengiriman produk yang sudah jadi secara tepat waktu. Selain itu, metode ini juga menghemat banyak uang karena tidak menggunakan terlalu banyak dokumentasi (Šimícková, Bugarová dan Mošková, 2021).

Extreme Programming (XP) telah secara luas digunakan dan cukup terkenal untuk metode pengembangan perangkat lunak. Pengembangan metode XP juga dilakukan untuk tujuan tertentu dengan cara melakukan kombinasi dengan metode lain, contohnya *Personal Software Process* (PSP) adalah metode yang menyediakan kerangka kerja yang tertata secara pribadi untuk melakukan pekerjaan perangkat lunak. Penelitian terkait XP dan PSP telah dilakukan oleh beberapa peneliti dan mengembangkannya sebagai metode baru berdasarkan XP yang dikombinasikan dengan PSP, sehingga fokus pada praktik XP untuk satu orang pemrogram bukan beberapa pemrogram. PXP mengenalkan subset dari pengembang praktisi XP yang disesuaikan untuk dilakukan oleh pemrogram atau pengembang tunggal. PXP menawarkan kerangka kerja baru untuk mengembangkan perangkat lunak atau sistem informasi. Metode ini dipilih karena praktik pengembangan yang praktis dari *Extreme Programming* dalam mendukung rencana proyek yang lebih baik dan kontrol kualitas produk serta bisa digunakan oleh pengembang otonom. (S. A. Asri dkk., 2018).

PXP merupakan suatu metodologi yang meningkatkan kualitas dari pengembangan perangkat lunak. PXP cocok untuk pengembang otonom karena PXP menyediakan pendekatan untuk meningkatkan rencana pribadi dan di saat yang bersamaan juga meningkatkan kualitas dari pengembangan perangkat lunak. Metode PXP menyajikan proses untuk mengurangi risiko kegagalan proyek

perangkat lunak saat dikerjakan oleh pengembang otonom (Iyawa, 2020). Penerapan metode PXP umumnya diterapkan pada pembuatan sistem informasi yang proses pengembangannya sering kali mengalami perubahan. Metode ini juga dirancang untuk pengembang tunggal yang memiliki waktu proses pengembangan yang terbatas. Pada gambar dibawah ini adalah bentuk proses dari fase *Personal Extreme Programming* terdiri dari tujuh langkah (Suharto, 2020).



Gambar 2.3 Metode *Personal Extreme Programming* (PXP) (Suharto, 2020)

Gambar 2.3 merupakan metode *Personal Extreme Programming* (PXP) yang berisi tujuh tahap yaitu *requirement*, *planning*, *iteration initializaton*, *design*, *implementation*, *system testing*, dan *retrospective*. Berikut ini penjelasan dari tujuh tahap yang ada dalam metode *Personal Extreme Programming*.

2.6.1 *Requirement*

Pada tahap pertama metode PXP yaitu *Requirement*, dilakukan analisis kebutuhan yang diperlukan pengguna sistem. Kebutuhan tersebut dianalisis berdasarkan data-data yang telah dikumpulkan. Pengumpulan data bisa dilakukan dengan cara observasi, wawancara, atau menyebar kuesioner. Data informasi yang diperoleh akan direpresentasikan menjadi *user story* karena berisi deskripsi fitur perangkat lunak dari sudut pandang pengguna (Sri Andriati Asri dkk., 2018).

2.6.2 *Planning*

Tahap *Planning* menjelaskan fungsionalitas sistem yang akan berguna bagi pengguna. Setiap *user story* memiliki skor atau nilai yang mewakili estimasi usaha dan skala prioritas untuk mengembangkan fitur yang ada pada sistem. Fitur yang diwakilkan pada *user story* mengandung deskripsi kebutuhan pengguna dan kriteria penerimaan yang mendefinisikan perilaku yang diharapkan dari fitur yang dibuat (Rachmaniah, Krismanti and Darissalam, 2020).

2.6.3 *Iteration Initialization*

Pada tahap *Iteration Initialization*, merupakan proses tahap awal dari tiap iterasi. Tahap ini sangat membentuk proses iterasi menjadi lebih tertata, efisien dan mudah dimengerti. Mulai dari membagi pengembangan sistem menjadi beberapa unit. Jangka waktu pengerjaan tiap unit iterasi yaitu satu sampai tiga minggu, dan terus diulang sampai selesai dan tidak ditemukan permasalahan (Sri Andriati Asri dkk., 2018)

2.6.4 *Design*

Pada tahap *Design*, pengembang perlu merancang *database* yang disajikan dalam bentuk *class diagram*. Tujuan dibuatnya *class diagram* untuk mengetahui letak modul, kelas dan fungsi dari sistem. Pengembang juga perlu merancang tampilan antarmuka pengguna sesuai *database* yang sudah dirancang. Tahap ini juga membantu menentukan fungsionalitas saat dilakukan pengembangan (Sri Andriati Asri dkk., 2018).

2.6.5 *Implementation*

Tahap *Implementation* merupakan proses implementasi pembuatan kode program yang sebenarnya terjadi. Ada 3 sub-proses pada tahap ini, *unit testing*, *code*, dan *refactoring*. Pertama, *Unit Testing* atau pengujian unit perlu dibuat sebelum membuat kode program agar membantu pengembang mendeteksi error yang terjadi pada kode saat pengembangan sistem. Kedua, *Code* atau kode yang dibuat berdasarkan *user story* yang didefinisikan pada tahap sebelumnya. Ketiga, *Refactor* atau koreksi ulang kode program dilakukan jika terdapat permasalahan,

jika tidak ada maka dilanjutkan ke tahap selanjutnya (Sri Andriati Asri *dkk.*, 2018; Suharto, 2020)

www.itk.ac.id

2.6.6 System Testing

Sistem yang diujinya pada sisi fungsionalitas sistem merupakan *system testing*. Pengujian sistem berupa validasi menggunakan kriteria tes kesepakatan pengguna atau *User Acceptance Test (UAT)*. UAT merupakan tahap terakhir dari pengujian sistem dari sisi pengguna yang telah menyepakati format yang telah disediakan. Tujuan UAT yaitu untuk mengetahui jawaban dari pengguna apakah sudah paham dengan sistem atau belum. Pengujian dengan UAT merupakan tahap terakhir dari pengembangan sistem (Pratama, 2018).

Kriteria UAT pada penelitian ini diterapkan dalam metode *black box testing*. *Black box* adalah salah satu pengujian yang ada pada *system testing*, dimana modul atau unit dijalankan, lalu hasil dari unit diamati sampai menghasilkan jawaban telah sesuai atau tidak sesuai dengan proses bisnis yang ditentukan (Aulia dan Suharto, 2019).

2.6.7 Retrospective

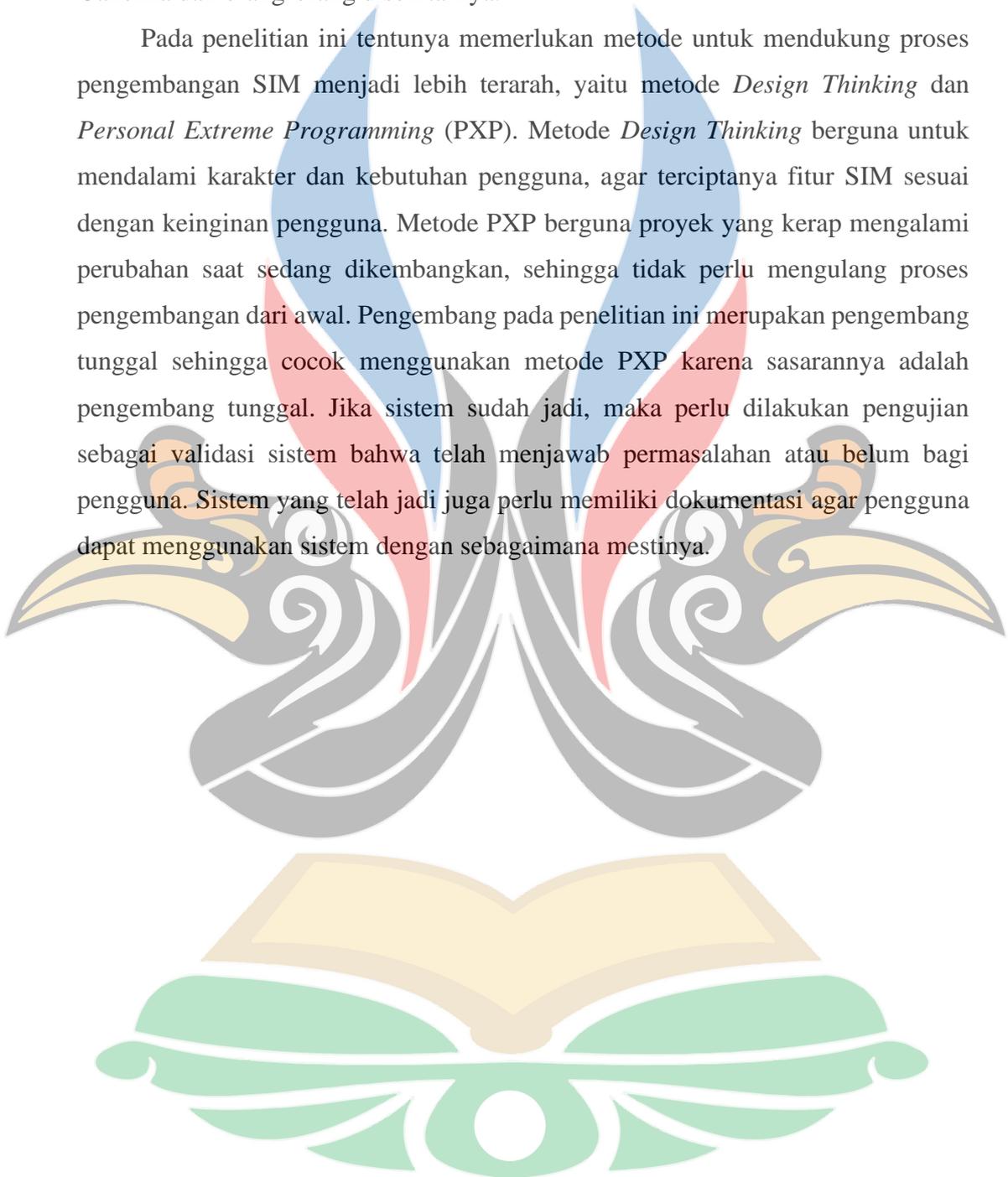
Tahap terakhir dari metode PXP yaitu *Retrospective*, berarti telah berakhirnya iterasi. Pengembang melakukan verifikasi waktu penyelesaian sudah sesuai atau tidak. Jika lebih dari rencana estimasi awal, maka dicari permasalahannya dan harus ditemukan solusi untuk pengembangan selanjutnya. Proyek dapat dinyatakan selesai jika semua kebutuhan sistem telah terpenuhi (Suprpto, Marthasari dan Nuryasin, 2020).

2.7 Penelitian Terdahulu

Penelitian terdahulu yang telah dirangkum pada Tabel 2.1 merupakan penelitian yang telah dilakukan sebelumnya, berisi topik dan metode yang berkaitan dengan pengembangan SIM. Berdasarkan penelitian sebelumnya, ditarik kesimpulan bahwa penerapan teknologi informasi dapat memberikan dampak positif bagi pengguna, yaitu kegiatan manajemen data informasi jadi lebih mudah karena dapat memanfaatkan fitur-fitur yang tersedia, sehingga dapat lebih praktis

dari segi waktu dan pengambilan keputusan. Data informasi tersebut dikelola dengan cara disimpan di *database* agar lebih aman dan terstruktur, sehingga diharapkan penerapan SIM di LPK Ganezha dapat membawa manfaat bagi LPK Ganezha dan orang-orang disekitarnya.

Pada penelitian ini tentunya memerlukan metode untuk mendukung proses pengembangan SIM menjadi lebih terarah, yaitu metode *Design Thinking* dan *Personal Extreme Programming* (PXP). Metode *Design Thinking* berguna untuk mendalami karakter dan kebutuhan pengguna, agar terciptanya fitur SIM sesuai dengan keinginan pengguna. Metode PXP berguna proyek yang kerap mengalami perubahan saat sedang dikembangkan, sehingga tidak perlu mengulang proses pengembangan dari awal. Pengembang pada penelitian ini merupakan pengembang tunggal sehingga cocok menggunakan metode PXP karena sasarannya adalah pengembang tunggal. Jika sistem sudah jadi, maka perlu dilakukan pengujian sebagai validasi sistem bahwa telah menjawab permasalahan atau belum bagi pengguna. Sistem yang telah jadi juga perlu memiliki dokumentasi agar pengguna dapat menggunakan sistem dengan sebagaimana mestinya.



www.itk.ac.id
Tabel 2.1 Penelitian Terdahulu

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian	Kelebihan	Kekurangan
1.	<i>Integrating Design Thinking into Scrum Framework in the Context of Requirement Engineering Management</i>	(Alhazmi dan Huang, 2020)	Kebutuhan rekayasa sistem kurang memadai, sering berubah, dan tidak ada kejelasan	<i>Design Thinking</i> dan <i>Scrum</i>	Menyajikan penentuan yang jelas pada tahap kebutuhan rekayasa, menaikkan kualitas produk, meningkatkan produktivitas dan kreativitas tim.	Kebutuhan rekayasa lebih pasti karena dibantu oleh metode <i>Design Thinking</i>	Perlu dilakukan secara tim dan tidak individual
2.	<i>Design Thinking and Agile Practices for Software Engineering: An</i>	(Corral dan Fronza, 2018)	Kelas rekayasa perangkat lunak di universitas memerlukan model pengembangan rekayasa perangkat	<i>Design Thinking</i> dan <i>Agile</i>	Kedua metode diuji pada sekitar 20 mahasiswa di universitas yang berbeda, untuk	Metode <i>Agile</i> : berpusat pada perangkat lunak (kualitas yang baik bagi	Perlu diuji ke banyak pengguna (siswa). Pengujian metode <i>Design</i>

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian	Kelebihan	Kekurangan
	<i>Opportunity for Innovation</i>		lunak yang mudah dipahami		mengetahui metode yang dapat melatih kreativitas dan teknik disiplin untuk mengembangkan perangkat lunak	perangkat lunak) Metode <i>Design Thinking</i> : menyampaikan solusi yang berpusat pada manusia (menyelesaikan masalah, memiliki <i>prototype</i>)	<i>Thinking</i> : tidak ada dokumentasi. Metode Agile : hanya ada skenario produk tanpa visualisasi
3.	<i>A Novel Perspective to Threat Modelling using Design</i>	(De, 2020)	Melihat dasar kepribadian manusia untuk mengidentifikasi kelompok pengguna	<i>Design Thinking, Threat Modelling, dan Scrum</i>	Memetakan 3 metode yaitu <i>Threat Modelling, Design</i>	Memberi pemahaman lebih secara keseluruhan ketika	Proses metode <i>Scrum</i> yang digunakan tidak detail

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian	Kelebihan	Kekurangan
	<i>Thinking and Agile Principles</i>		yang bisa menjadi ancaman suatu sistem		<i>Thinking</i> , dan <i>Scrum</i> yang menjadi proses baru	menghadapi suatu ancaman	
4.	Pengaruh Metode <i>Design Thinking</i> yang digabungkan dengan metode <i>Extreme Programming</i> Dalam Membangun Inovasi pada Website “Kerjayuk” Untuk Mahasiswa	(Fachrian dkk., 2021)	Proses metode <i>Extreme Programming</i> memiliki pengetahuan yang kurang dalam memahami <i>user</i> , sehingga dibutuhkan metode <i>Design Thinking</i> untuk mendukung inovasi pengembangan website Kerjayuk	<i>Extreme Programming</i> dan <i>Design Thinking</i>	Anggota tim positif untuk menggabungkan metode <i>Design Thinking</i> dan <i>Extreme Programming</i> agar dapat memperoleh kebutuhan <i>user</i> secara jelas melalui tahapan yang ada dalam	Mengatasi kekurangan metode XP yaitu siap menerima perubahan <i>requirement</i>	Kesulitan saat mencari solusi di tahap <i>Define</i> dan jika ada perubahan pada desain <i>website</i> , tidak diketahui alur proses selanjutnya.

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian	Kelebihan	Kekurangan
	Universitas Telkom				metode <i>Design Thinking</i>		
5.	<i>The Development of Website on Management Information System for E-commerce and Services</i>	(Hidayat, Dewantara dan Saifullah, 2020)	Perlu pengembangan sistem informasi manajemen <i>e-commerce</i> dan <i>services</i>	Waterfall	<i>Prototype</i> telah dikembangkan menjadi platform <i>e-commerce</i> dan fiturnya telah diuji dengan <i>black box</i> dan <i>white box testing</i> serta diterapkan dalam sistem informasi manajemen	Semua fitur telah diuji menggunakan <i>blackbox</i> dan <i>whitebox</i>	Tidak dapat mengulang kembali proses tahapan metode

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian	Kelebihan	Kekurangan
6.	<i>A CONCEPTUAL MODEL FOR INTEGRATING DESIGN THINKING AND LEAN STARTUP METHODS INTO THE INNOVATION PROCESS</i>	(Marion dkk., 2021b)	Masih belum tau efektivitas dari pengimplementasian metode yang fokus pada pengguna dalam proses inovasi organisasi	<i>Design Thinking dan Lean Startup dan Agile</i>	Proses pengembangan konsep, prototipe, dan interaksi dengan pengguna dapat diintegrasikan secara lancar selama proses inovasi	Kebutuhan pelanggan, ide, dan pengembangan terintegrasi dengan baik dan fokus pada interaksi pengguna	Penelitian ini terbatas pada perbandingan teoritis dan pemikiran sistem dan pemikiran desain
7.	<i>Combining Design Thinking and Agile Development to Master Highly Innovative IT Projects</i>	(Przybilla dkk., 2018)	Mengintegrasikan model Design Thinking, dalam konteks manajemen kebutuhan rekayasa, ke dalam kerangka kerja Scrum	<i>Design Thinking dan Agile</i>	Meningkatkan kebutuhan rekayasa Scrum yang ada dengan mengatasi tantangannya dalam beberapa kasus	Memberikan definisi yang jelas untuk tahap kebutuhan rekayasa, meningkatkan kualitas	Belum dilakukan validasi kerangka kerja, tidak adanya poin prioritas terkait hasil <i>user story</i>

No.	Judul Penelitian	Peneliti dan Tahun	Permasalahan	Metode	Hasil Penelitian	Kelebihan	Kekurangan
						produk, meningkatkan produktivitas tim, dan meningkatkan kreativitas tim	sebagai acuan pengembangan software.
8.	<i>Integrating Design Thinking into Extreme Programming</i>	(Sohaib dkk., 2019)	Meningkatnya permintaan untuk sistem informasi bisnis dan melakukan inovasi untuk bersaing di pasar dalam konteks pengembangan perangkat lunak	<i>Design Thinking</i> dan XP	Kerangka kerja terintegrasi berdasarkan konsep DT dan XP yang dapat meningkatkan kualitas pengembangan perangkat lunak.	Berkontribusi terus-menerus dan konstan kepada pelanggan dengan lebih cepat dibanding pendekatan tradisional.	Perlu pengevaluasian dalam proyek perangkat lunak

2.8 Komparasi Metode

Metode *Design Thinking* dipilih karena mengedepankan permasalahan pengguna. Tahapan yang ada dalam *Design Thinking* memiliki ruang untuk menampung permasalahan pengguna, yaitu dapat mengerti kebutuhan dan mendapat celah kesempatan melalui empati kepada pengguna. Pengembangan konsep dan struktur ide menjadi tahap yang dapat menghasilkan apa yang sebenarnya dibutuhkan oleh pengguna. Beda dengan metode yang masih satu rumpun dengan *Design Thinking* yaitu *Lean Startup* yang hanya fokus pada *functional prototype* yang akan diujikan pada pengguna. *Design Thinking* juga memiliki tahapan *prototype* sehingga lebih lengkap dari pada *Lean Startup*. (Marion dkk., 2021a). Berbeda dengan metode *Waterfall* dan *Kanban*, kedua metode itu tidak berpusat pada permasalahan pengguna, karena metode *Waterfall* adalah metode yang linier dan hanya dapat mencari kebutuhan pada saat tahap awal dan perlu disetujui sehingga pada tahap tahapan selanjutnya tidak dapat merubah kebutuhan tersebut (Herawati dkk., 2021). Metode *Kanban* tidak detail untuk mencari permasalahan pengguna, karena metode *Kanban* lebih fleksibel yaitu menyerahkan kepada tim untuk melakukan kegiatan apa yang ingin dikerjakan terlebih dahulu lalu dikategorikan berdasarkan tipe kegiatan masuk di tahap *do*, *process*, atau *done* (Šimícková, Bugarová dan Mošková, 2021).

Penerapan *Design Thinking* dan *Personal Extreme Programming* pada penelitian ini bertujuan untuk saling melengkapi kedua metode. Secara teknis *Design Thinking* lebih berpusat pada pengguna, mulai dari mencari permasalahan sampai memecahkan masalah dan mengujikan produk berupa *prototype* kepada pengguna. Pembuatan produk yang sebenarnya memerlukan metode pendukung khusus untuk pembuatan sistem, yaitu *Personal Extreme Programming* (XP) karena terdapat beberapa celah yang dapat menghasilkan informasi kurang detail dan rinci, sehingga kedua metode tersebut digabung untuk melihat proses dan hasil dari tiap tahapan yang dilakukan pada saat mencari kebutuhan pengguna sampai selesainya pembuatan sistem.