

BAB II

LANDASAN TEORI

2.1. Kajian Literatur

2.1.1. Sistem

Sitohang (2018) mengatakan Sistem merupakan suatu jaringan kerja dari prosedur-prosedurnya yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem adalah himpunan suatu “benda” nyata (*set of* atau *think*) yang abstrak terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, dan saling mendukung, yang secara keseluruhan bersatu dalam kesatuan (*unity*) untuk mencapai tujuan tertentu secara efisien dan efektif (Febrianto *et al.*, 2023). Menurut Rusdiana dan Irfan (2014), pendekatan sistem adalah transaksi dari suatu urutan logis dari operasi untuk tujuan mengubah satu atau lebih faktor dalam suatu sistem

2.1.2. Informasi

Sitohang (2018) dalam jurnalnya mendefinisikan informasi sebagai data yang telah diolah menjadi bentuk yang lebih berguna bagi yang menerima. Informasi juga didefinisikan sebagai data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya (Febrianto *et al.*, 2023). Menurut Rusdiana dan Irfan (2014) Informasi adalah suatu data yang sudah diolah atau diproses sehingga menjadi suatu bentuk yang memiliki arti bagi penerima informasi yang memiliki nilai bermanfaat.

2.1.3. Sistem Informasi

Chiasson *et al.* (2007) mengatakan “*information systems (IS) field is one area that has developed and elaborated theories and methods for studying IT in social and organizational settings*”. Menurut Sitohang (2018) Sistem Informasi adalah elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses, dan menyimpan serta mendistribusikan informasi. Sapto Aji *et al.* (2014) mengatakan bahwa Sistem Informasi memiliki komponen-komponen yang terdapat di dalamnya yaitu terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, blok kendali. Penerapan sistem informasi di suatu organisasi atau perusahaan sangat dibutuhkan untuk membantu dalam melakukan fungsi pengolahan data atau informasi yang bertujuan untuk meningkatkan efektivitas penggunaan teknologi yang mendukung kegiatan perusahaan, serta membantu pengambilan keputusan manajemen. Menurut Rusdiana dan Irfan (2014) Sistem informasi adalah sekumpulan *hardware, software, brainware*, prosedur, dan/atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan. Bin Ladjamudin, A. B. (2005, dikutip dalam Muslih & Purnama, 2013) mengatakan

bahwa Sistem informasi adalah satu kesatuan data olahan yang terintegrasi dan saling melengkapi yang menghasilkan data olahan, baik dalam bentuk gambar, suara maupun tulisan.

2.1.4. Pengembangan Sistem Informasi

Pengembangan sistem informasi adalah aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk menyelesaikan permasalahan organisasi atau memanfaatkan kesempatan yang timbul (Bahar, 2021). Pengembangan sistem terdiri dari beberapa macam berdasarkan sudut pandang yang berbeda yaitu pandangan metodologi, pandangan sasaran, penentuan kebutuhan, teknik pengembangan, teknologi, siklus hidup pengembangan sistem (Prabowo, 2020).

2.1.5. Siklus Hidup Pengembangan Sistem

Siklus Hidup Pengembangan Sistem atau *System Development Life Cycle* (SDLC) adalah proses lima langkah yang digunakan untuk merancang dan mengimplementasikan sistem baru (Romney & Steinbart, 2018). Bahar (2021) mendefinisikan SDLC sebagai siklus yang dilalui dalam pengembangan sistem informasi yang dimulai dari sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan, dipelihara, dan dapat kembali ke tahap awal perencanaan apabila terdapat permasalahan kritis yang mengharuskan sistem tersebut perlu ditinjau kembali.

Prabowo (2020) menyebutkan terdapat 5 tahap dalam SDLC yaitu perencanaan, analisis, *design*, *implementation*, dan *activities*. Berikut adalah penjelasan masing-masing tahapan yang dijelaskan oleh Prabowo:

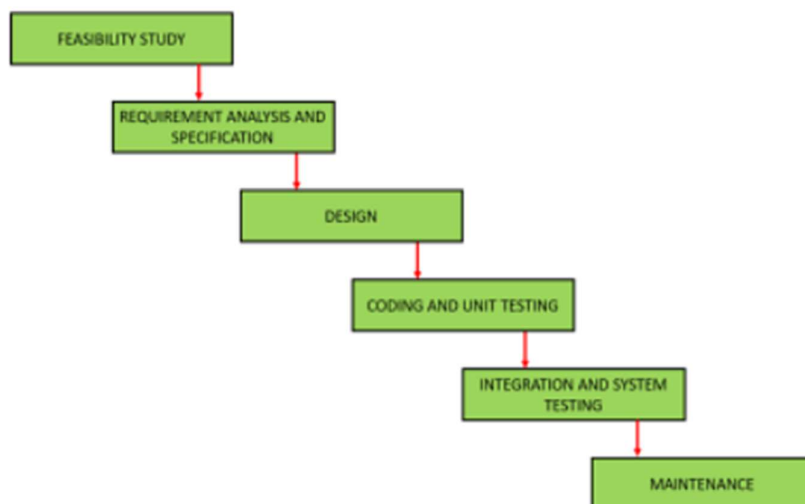
- 1) Tahap perencanaan pada dasarnya bertujuan untuk mengetahui dan mengedepankan sistem yang akan dikembangkan, target apa yang akan dicapai, lama pengerjaannya, serta mempertimbangkan kebutuhan dana yang digunakan dalam pengembangan sistem. Tahap perencanaan juga mendefinisikan masalah bisnis yang dihadapi oleh organisasi serta perihal apa saja yang kiranya menghambat laju kembangnya bisnis.
- 2) Tahap analisis merupakan tahapan mengumpulkan kebutuhan-kebutuhan organisasi dari berbagai *stakeholder* yang berkepentingan. Kebutuhan-kebutuhan bisnis ini nantinya akan ditindaklanjuti guna diakomodir oleh sistem yang akan dibangun.
- 3) Tahap *design* pada dasarnya melibatkan proses perencanaan yang melibatkan integrasi ke dalam jaringan, perancangan arsitektur aplikasi, pengembangan tampilan pengguna, desain antarmuka, penyelarasan sistem dengan *database*, pembuatan *prototype* untuk rincian perancangan, dan integrasi sistem dengan kontrol sistem.
- 4) Tahap *implementation* adalah langkah dalam pembangunan elemen-elemen perangkat lunak. Ini melibatkan validasi dan pengujian sistem yang telah dibuat, migrasi data, pelatihan pengguna untuk menggunakan sistem, penyusunan dokumentasi sistem (termasuk manual), dan instalasi sistem di terminal komputer yang memerlukannya.
- 5) Tahap *activities* pada dasarnya tahap melakukan pemeliharaan sistem dengan pengecekan berkala. Pengembang juga dapat melakukan penambahan fitur-

fitur baru yang dapat meningkatkan kinerja pengguna untuk mendukung kinerja bisnis.

2.1.6. Metode *Waterfall* Klasik

Classical waterfall atau *Linear Sequential Model* adalah metode pengembangan perangkat lunak tertua yang bersifat sistematis dan berurutan. Model ini dapat digunakan pada saat kebutuhan untuk sebuah masalah telah dipahami dengan baik, dan pekerjaan dapat mengalir secara linear dari proses komunikasi hingga penyebaran (Prabowo, 2020). Fase sekuensial atau tahapan pengembangan dengan menggunakan metode *waterfall* klasik penulis sajikan pada Gambar II.1.

Gambar II.1 Tahapan Metode *Waterfall* Klasik



Sumber : Prabowo, 2020

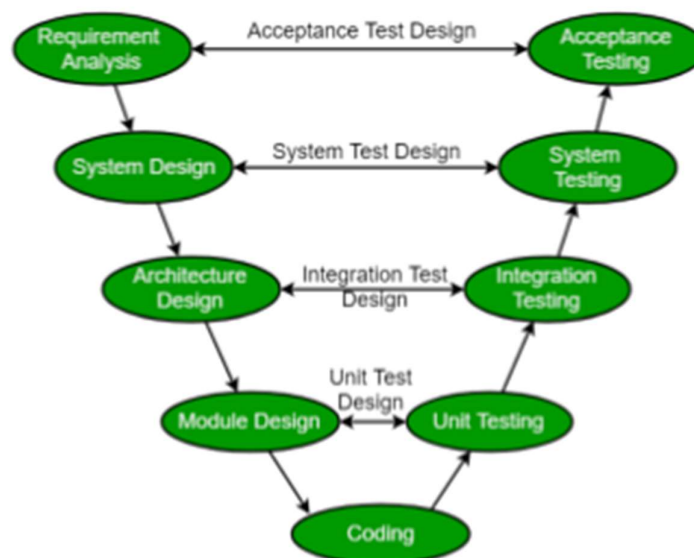
Maulana dan Ikasari (2023) mengatakan bahwa kelebihan dari metode *waterfall* diantaranya adalah adanya kerangka kerja yang terstruktur dan berurutan. Selain itu metode ini juga memungkinkan tahapan analisis kebutuhan yang jelas,

perancangan terencana, implementasi yang sistematis, pengujian yang menyeluruh, dan pemeliharaan yang terorganisir. Sedangkan kekurangan metode *waterfall* klasik menurut Fadillah dan Ikasari (2023) yaitu sulitnya pengujian secara iteratif selama proses pengembangan dan bersifat tidak fleksibel jika ada penyesuaian yang dibutuhkan selama proses pengembangan.

2.1.7. Metode V-Model

Model V adalah salah satu model proses pengembangan perangkat lunak yang merupakan variasi dari metode *waterfall*. Apabila model *waterfall* dilakukan secara linear, maka dalam V-model prosesnya dilakukan secara bercabang (Prabowo, 2020). Model ini juga dikenal sebagai model verifikasi dan validasi. Hal ini didasari dengan fase pengujian untuk setiap tahap pengembangan yang sesuai. Tahapan pengembangan dengan V-model penulis sajikan pada Gambar II.2.

Gambar II.2 Tahapan Pengembangan V-Model



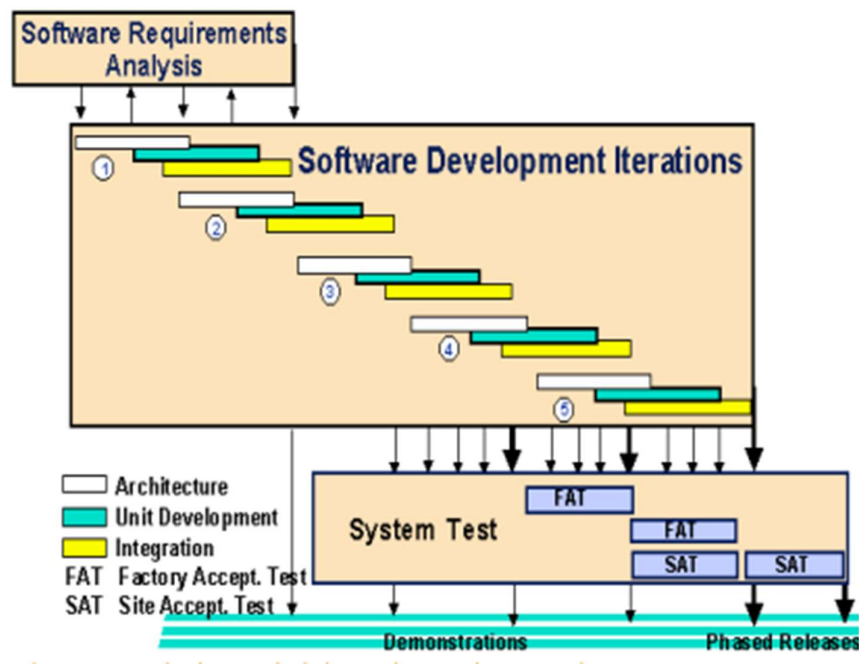
Sumber : Prabowo, 2020

Munassar dan Govardhan (2010) mengatakan bahwa terdapat beberapa kelebihan dan kekurangan dari penggunaan V-model. Kelebihan dari v-model ini diantaranya adalah sederhana dan mudah digunakan, setiap fase memiliki hasil yang spesifik, peluang sukses yang lebih tinggi dibandingkan dengan metode *waterfall*, dan akan bekerja lebih baik untuk proyek kecil yang persyaratannya mudah dipahami. Sedangkan kekurangannya adalah model ini sangat kaku seperti model *waterfall*, fleksibilitas dan penyesuaian ruang lingkup yang sulit dan mahal, perangkat lunak dikembangkan pada tahap implementasi sehingga tidak ada prototipe awal yang diproduksi, dan model ini tidak memberikan jalur yang jelas terhadap permasalahan yang ditemukan pada saat fase pengujian.

2.1.8. Metode *Iterative*

Munassar dan Govardhan (2010) mengatakan bahwa Model Waterfall sering menimbulkan masalah yang memicu permintaan untuk pendekatan pengembangan baru yang mampu memberikan hasil lebih cepat, memerlukan informasi awal yang lebih sedikit, dan memberikan fleksibilitas yang lebih besar. Pendekatan iteratif dalam pengembangan memungkinkan proyek dibagi menjadi bagian-bagian kecil, memungkinkan tim untuk menunjukkan hasil di awal proses dan mendapatkan umpan balik berharga dari pengguna. Setiap iterasi sering merupakan proses mini-Waterfall, dengan umpan balik dari satu fase yang memberikan informasi penting untuk desain fase berikutnya. Dalam variasi model ini, produk perangkat lunak yang dihasilkan pada setiap langkah dapat langsung dirilis sebagai tambahan. Ilustrasi tahapan dari model *iterative* penulis sajikan pada Gambar II.3.

Gambar II.3 Tahapan Metode Incremental



Sumber : Munassar dan Govardhan, 2010

Prabowo (2020) menjelaskan bahwa metode *iterative* memiliki kelebihan dan kelemahan. Kelebihan dari metode ini diantaranya adalah model dengan manajemen yang sederhana, pengguna tidak perlu menunggu sampai seluruh sistem dikirim untuk mendapatkan keuntungan dari sistem tersebut, risiko kegagalan proyek secara keseluruhan lebih rendah, nilai penggunaan dapat ditentukan pada setiap *increment* sehingga fungsionalitas sistem dapat disediakan lebih awal, dan prioritas tertinggi pada pelayanan sistem adalah yang paling diuji. Sedangkan kelemahan dari metode ini diantaranya adalah memungkinkan tiap bagian tidak dapat diintegrasikan, dapat menjadi *build* dan *fix* model, harus *open architecture*, mungkin terjadi kesulitan untuk memetakan kebutuhan pengguna ke dalam rencana

spesifikasi masing-masing hasil *increment*, dan membutuhkan waktu yang relatif lama untuk menghasilkan produk yang lengkap.

2.1.9. Metode *Prototyping*

Prototyping didefinisikan sebagai metode pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pengguna (Santi & Fitriyah, 2016). Romney dan Steinbart (2018) menjelaskan *prototype* masih merupakan bagian dari SDLC namun memungkinkan bagi pengembang untuk menyingkat dan mempercepat beberapa tugas analisis dan desain. Romney dan Steinbart lebih lanjut menjelaskan bahwa *prototyping* dikembangkan dengan menggunakan empat langkah, yaitu berkomunikasi dengan pengguna, mengembangkan *prototype* awal, uji coba dan modifikasi, dan penggunaan sistem. *Flowchart* tahapan pengembangan *prototype* menurut Romney dan Steinbart penulis sajikan pada Gambar II.1.

1) Berkomunikasi dengan pengguna

Komunikasi dengan pengguna bertujuan untuk menyepakati ruang lingkup sistem dan memutuskan apa saja yang akan dimasukkan ke dalam sistem. Penekanannya ada pada *output* yang harus dihasilkan, pemrosesan transaksi, serta masukan dan data yang diperlukan dalam menghasilkan *output* tersebut. Pengembang juga harus memastikan bahwa harapan pengguna realistis dan kebutuhan informasi dasar mereka dapat terpenuhi.

2) Mengembangkan *prototype* awal.

Pengembangan *prototype* awal berfokus pada biaya yang rendah dan perkembangan pesat. Fungsi-fungsi yang tidak penting, kontrol, penanganan

pengecualian, validasi input, dan kecepatan pemrosesan dapat diabaikan demi kesederhanaan, fleksibilitas, dan kemudahan pengguna. Pengguna perlu melihat dan menggunakan layar entri data tentatif, menu, dan dokumen sumber; menanggapi petunjuknya; menanyakan sistem; waktu respons hakim; dan mengeluarkan perintah. Apabila *prototype* awal telah selesai maka pengembang harus mendemonstrasikannya kepada pengguna. Hal ini bertujuan untuk mendapatkan umpan balik terkait hal-hal yang mereka suka dan tidak suka atas *prototype* tersebut.

3) Uji coba dan modifikasi

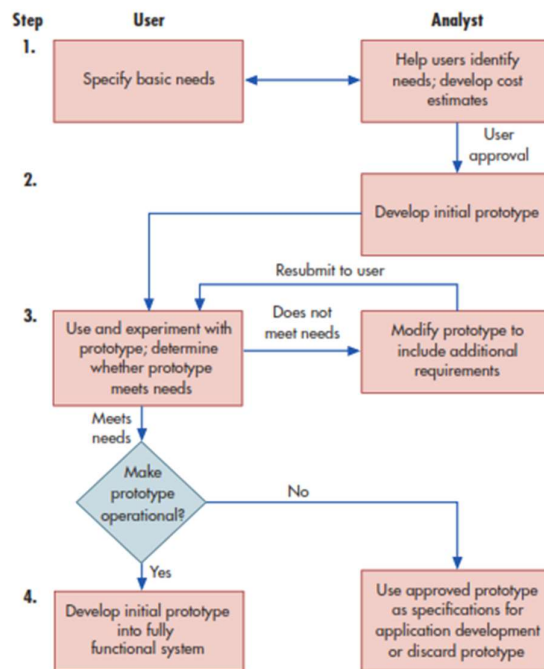
Pengembang akan menggunakan umpan balik atas hasil demonstrasi pada tahap sebelumnya untuk melakukan modifikasi sistem. Sistem yang telah selesai dimodifikasi akan dikembalikan ke pengguna. Uji coba penggunaan dan modifikasi ini akan terus berjalan hingga pengguna puas dan merasa bahwa sistem telah memenuhi kebutuhan mereka. *Prototyping* biasanya melewati empat hingga enam siklus iterasi modifikasi.

4) Penggunaan sistem

Prototype yang disetujui biasanya digunakan dalam salah satu dari dua cara. Cara pertama yaitu separuh dari *prototype* diubah menjadi sistem yang berfungsi penuh, *prototype* ini disebut dengan *prototype* operasional. Pengembang menggabungkan hal-hal yang diabaikan pada langkah pertama, menyediakan cadangan dan pemulihan, dan mengintegrasikan *prototype* dengan sistem lain untuk membuatnya dapat beroperasi. Sedangkan pada cara *prototype* non-operasional,

prototype pada dasarnya akan dibuang, namun persyaratan sistem akan diidentifikasi untuk digunakan dalam pengembangan sistem baru lainnya.

Gambar II.4 *Flowchart* Tahapan Pembangunan *Prototype*



Sumber : Romney & Steinbart, 2018

Purnomo (2017) mengatakan bahwa adanya *prototyping* bagi pengembang sistem bertujuan untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model *prototype* yang dikembangkan, sebab *prototype* menggambarkan versi awal dari sistem untuk kelanjutan sistem sesungguhnya yang lebih besar. Purnomo juga menjelaskan bahwa terdapat beberapa manfaat lain dari penggunaan *prototyping* antara lain:

- 1) Mewujudkan sistem sesungguhnya dalam sebuah replika sistem yang akan berjalan, menampung masukan dari pengguna untuk kesempurnaan sistem;

- 2) Pengguna akan lebih siap menerima setiap perubahan sistem yang berkembang sesuai dengan berjalannya *prototype* sampai dengan hasil akhir pengembangan yang akan berjalan nantinya;
- 3) *Prototype* dapat ditambah maupun dikurangi sesuai berjalannya proses pengembangan. Kemajuan tahap demi tahap dapat diikuti langsung oleh pengguna; dan
- 4) Penghematan sumber daya dan waktu dalam menghasilkan produk yang lebih baik dan tepat guna bagi pengguna.

2.1.10. Aplikasi PHPRad

Aplikasi PHPRad merupakan lingkungan pengembangan aplikasi yang mampu menghasilkan aplikasi php dalam database (Ritoyan, 2021). Ritoyan menjelaskan lebih lanjut bahwa PHPRad *Classic* menghasilkan tampilan klasik aplikasi *website* yang ringan serta menggunakan teknologi:

- 1) *Laravel*, yang membuat bahasa pemrograman PHP menjadi dinamis, *powerful*, cepat, aman, dan sederhana;
- 2) *Jquery*, yang merupakan *library JavaScript* antar *platform* yang dirancang untuk menyederhanakan *client side scripting* pada bahasa HTML; dan
- 3) *Bootstrap 4*, CSS yang digunakan untuk merancang situs web.

PHPRad versi *Classic* memiliki beberapa keunggulan seperti dapat mengurangi waktu pengembangan, integrasi developer dan sumber daya, cocok digunakan untuk semua level pengguna, dan portabel (Afiana, 2023).

2.2. Penelitian Terdahulu

Akhmad Syarifudin dan Nur Ani (2019)

Akhmad Syarifudin dan Nur Ani melakukan perancangan Sistem Informasi untuk pengajuan dan pelaporan Tunjangan Kinerja Kementerian Keuangan dengan menggunakan metode *prototyping*. Tahapan pengembangan Sistem Informasi yang dilakukan oleh Syarifudin dan Ani adalah komunikasi, perencanaan secara tepat, pemodelan perancangan secara tepat, pembentukan *prototype*, dan penyerahan sistem ke pelanggan, pengiriman, serta umpan balik.

Hal yang dapat penulis pelajari dari penelitian ini adalah terdapat sepuluh prinsip yang dapat digunakan dalam melakukan evaluasi rancangan Sistem Informasi, yaitu visibilitas dari suatu sistem; kesesuaian antara sistem dengan dunia nyata; kendali dan kebebasan pengguna; standar dan konsistensi; pencegahan kesalahan; bantu pengguna untuk mengenali, mendiagnosa, dan mengatasi masalah; fleksibilitas dan efisiensi; estetika dan desain minimalis; pertolongan pengguna mengenal, berdialog, dan memperbaiki kesalahan; dan fitur bantuan dan dokumentasi.

Rina Candara Noor Santi dan Anida Fitriyah (2016)

Noor Santi dan Anida melakukan pengembangan sistem dengan menggunakan sudut pandang UCD (*User Centered Design*) untuk menghindari adanya bias dalam penyajian informasi yang pengguna ingin dapatkan pada suatu sistem informasi atau aplikasi. Tahapan dalam UCD mencakup memahami kebutuhan *user*; mendeskripsikan kebutuhan *user*; merancang *prototype* sebagai alternatif; dan mengevaluasi perancangan.

Hal yang dapat penulis pelajari dari penelitian ini adalah dalam melakukan *prototyping* adalah bagaimana pengembang menentukan sudut pandang pengguna. Hal ini dilakukan agar sistem informasi yang dibangun dapat mengoptimalkan cara penyajian informasi agar berguna bagi *user*.

Maryadi, Miftahul Hadi, dan Syanni Yustiani (2022)

Maryadi, *et al.* melakukan pengembangan sistem informasi *e-learning* dengan menggunakan model *prototyping*. Hal ini disebabkan oleh pengguna *e-learning* akuntansi sektor publik adalah seluruh lapisan masyarakat Indonesia yang sulit untuk diketahui kebutuhannya. Tahapan pengembangan yang dilakukan diantaranya perencanaan, analisis, desain, dan implementasi. Setelah prototipe awal selesai dibangun, peneliti melakukan survey untuk mendapatkan tingkat kepuasan pengguna yang diukur dengan menggunakan metode *Customer Satisfaction Index* untuk menggali kebutuhan pengguna.

Hal yang dapat penulis dapat dari penelitian ini adalah penting bagi pengembang untuk melakukan pengujian kepada *user* agar pengembang mendapatkan umpan balik pengguna. Umpan balik ini nantinya dapat digunakan untuk perbaikan Sistem Informasi kedepannya agar menjadi lebih baik.

Dwi Purnomo (2017)

Penelitian yang dilakukan oleh Purnomo memberikan gambaran penggunaan model *prototyping* pada kegiatan pengembangan sistem informasi dengan harapan dapat menghasilkan *prototype* sebagai salah satu langkah awal sebuah kegiatan pengembangan sistem informasi. Penelitian ini menghasilkan 3

rekomendasi, yaitu rekomendasi dalam perencanaan sistem, rekomendasi atas personel yang menangani sistem, dan rekomendasi terkait arsitektur jaringan.

Hal yang dapat penulis pelajari dari penelitian ini adalah diperlukan perencanaan yang matang dalam perencanaan dan pengembangan sistem, perencanaan dimulai dari sistem yang baik, pemilihan personel, penentuan perangkat lunak dan perangkat keras, serta arsitektur jaringan. Selain itu, komitmen yang jelas antara pengembang, pengguna, serta pimpinan organisasi diperlukan agar Sistem Informasi yang dibuat akan menjadi lebih optimal.

Arfianto (2019)

Arfianto melakukan penelitian untuk mendapatkan informasi terkait kesesuaian *interface* pada aplikasi parkir yang tersedia untuk *smartphone* dengan prinsip *user interface* yang baik. Hal yang dapat dipelajari adalah bahwa menurut Arfianto *user interface* yang baik terdiri dari beberapa aspek yaitu kejelasan, ringkas, mudah dikenali, responsif, konsisten, dan memiliki muatan estektika.