

BAB II

KAJIAN PUSTAKA

2.1 Landasan Teori

Kinerja

Kata “kinerja” merupakan singkatan dari “kinetika energi kerja”, yang dalam bahasa Inggris dikenal sebagai “performance”, dan sering diterjemahkan ke dalam bahasa Indonesia sebagai “performa”(Wirawan dalam Nursam, 2017). Lebih lanjut juga dijelaskan menurut Schermerhorn (1991), Kinerja sebagai kualitas dan kuantitas pencapaian tugas-tugas, baik yang dilakukan oleh individu, kelompok maupun perusahaan (sebagaimana dikutip dalam Nursam, 2017).

Survei

Survei adalah metode pengumpulan informasi dari sebagian kecil individu atau populasi yang sedang diteliti (Brenner, 2014). Sampel yang diambil biasanya mewakili populasi yang lebih besar, dan data yang diperoleh dari survei digunakan untuk membuat estimasi tentang karakteristik populasi tersebut. Secara etimologi kata survei berasal dari Bahasa Latin yang terdiri dari dua suku kata yakni *sur* yang berasal dari kata super yang berarti di atas atau melampaui. Sedangkan suku kata *vey* berasal dari kata videre yang berarti melihat. Jadi survei berarti melihat di atas atau melampaui (Leedy, 1980, dalam Irawan Soeharto, 2000:53).

Survei Kepuasan

Menurut Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Nomor 14 Tahun 2017 tentang Pedoman

Penyusunan Survei Kepuasan Masyarakat Unit Penyelenggara Pelayanan Publik, survei kepuasan masyarakat adalah kegiatan pengukuran secara komprehensif tentang tingkat kepuasan masyarakat terhadap kualitas layanan yang diberikan oleh penyelenggara pelayanan publik (APARATUR, 2017). Survei kepuasan masyarakat pada umumnya menggunakan pendekatan metode kualitatif dengan dengan pengukuran skala *likert*. Skala *likert* adalah alat yang digunakan untuk mengukur sikap, opini, dan persepsi individu atau kelompok terhadap suatu layanan publik. Dalam skala ini, responden diminta untuk menunjukkan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu opsi yang tersedia. Tingkat kualitas pelayanan di mulai dari sangat baik/puas sampai dengan tidak baik/puas. Pembagian jawaban dibagi dalam 4 (empat) kategori, yaitu tidak baik, kurang baik, baik dan sangat baik.

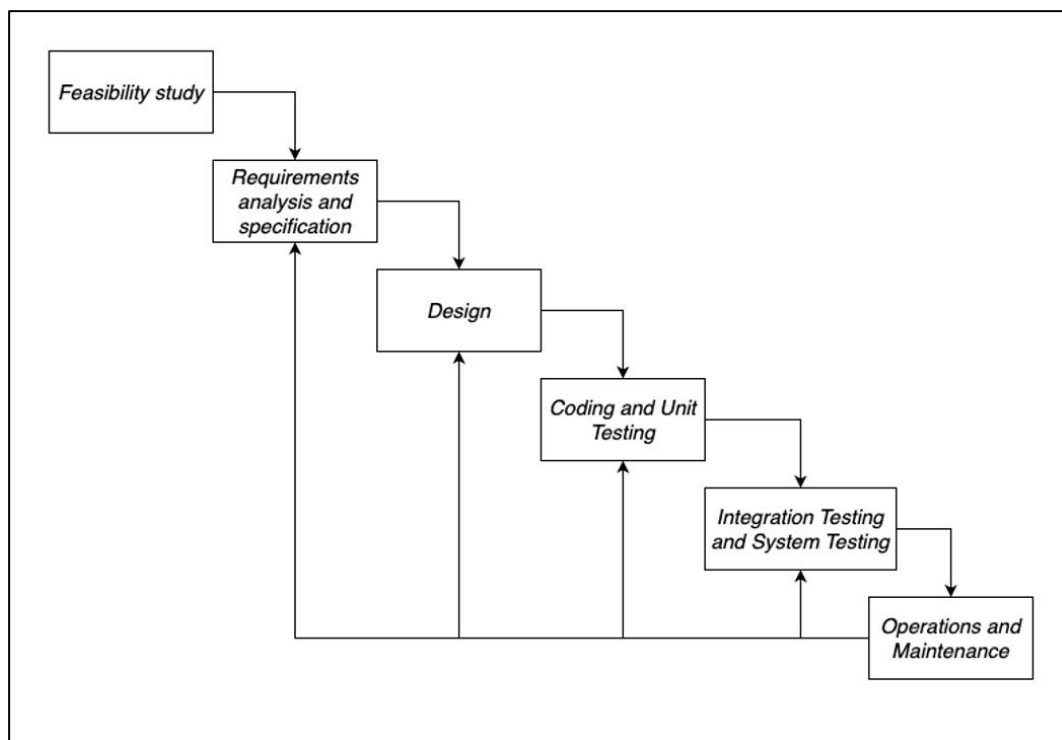
Metode Pengembangan Sistem (*System Development Life Cycle*)

Metode Pengembangan Sistem adalah sebuah standar proses yang mencakup berbagai aktivitas, metode, praktik terbaik, hasil akhir, dan alat otomatisasi yang digunakan dalam pengembangan sistem (Whitten & Bentley, 2007). Metode ini berfokus pada prosedur optimal untuk memperbarui sistem informasi yang sudah usang. Setiap metode pengembangan sistem dilakukan secara bertahap, memastikan bahwa sistem informasi tetap konsisten, mengurangi kesalahan, serta mudah untuk direplikasi dan dipahami. Oleh karena itu, dokumentasi sistem informasi yang lengkap dan jelas sangat membantu dalam mempermudah pembuatan dan pelaksanaan metode pengembangan sistem.

Metode Pengembangan Sistem Model *Iterative Waterfall*

Salah satu model metode pengembangan sistem adalah model *waterfall*. Model *waterfall* sangat populer pada tahun 1970-an dan masih banyak digunakan pada saat ini. *Waterfall* merupakan metode pengembangan sistem dengan pendekatan analisis sistem dan desain yang setiap tahap diselesaikan satu demi satu dan hanya sekali penyelesaian. Namun, model *classic waterfall* memiliki beberapa kekurangan diantaranya tidak bisa kembali ke tahap sebelumnya dan sulit untuk mengakomodasi permintaan perubahan (Mall, 2014). Oleh karena itu, perlu modifikasi pada model *classic waterfall* agar bisa mengatasi kekurangan tersebut.

Salah satu modifikasi dari model *classic waterfall* adalah *iterative waterfall*. Model *iterative waterfall* adalah pendekatan pengembangan perangkat lunak yang menggabungkan antara model *waterfall* tradisional dengan fleksibilitas model *iterative*. Model *iterative waterfall* memberikan akses dari setiap fase ke fase sebelumnya. Sehingga, ketika ada permintaan perubahan, pengembang dapat memenuhi permintaan tersebut dengan kembali ke fase sebelumnya.






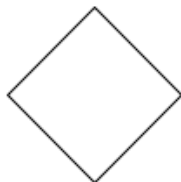
Sumber : R. Mall (2014)



Gambar 1. SDLC Model *Iterative Waterfall*

Diagram alir (*flowchart*)

Secara formal, *flowchart* adalah representasi diagramatis dari langkah-langkah sebuah algoritma (A. B. Chaudhuri, 2020). Dalam sebuah *flowchart*, berbagai bentuk digunakan untuk menunjukkan berbagai jenis operasi. Bentuk-bentuk ini kemudian dihubungkan oleh garis-garis dengan panah yang menunjukkan aliran atau arah yang harus diikuti untuk mengetahui langkah selanjutnya. Bentuk-bentuk ini kemudian dihubungkan oleh garis-garis dengan panah yang menunjukkan aliran atau arah yang harus diikuti untuk mengetahui langkah berikutnya. Garis-garis penghubung ini dikenal sebagai *flow lines*. Berikut ini adalah notasi yang biasa digunakan dalam *flow chart*.

Tabel 1. Notasi pada *flowchart*

Notasi (1)	Nama Notasi (2)	Keterangan (3)
	Terminal	Menunjukkan titik awal atau akhir dari sistem dengan memuat kata start atau mulai inapun end atau selesai.
	Masukan/Keluaran (Input/Output)	Menunjukkan masukan atau keluaran dari suatu proses.
	Proses (Processing)	Menunjukkan suatu proses atau aktivitas dalam sistem.
	Keputusan (decision)	Jajar genjang dengan satu panah alur yang masuk dan dua atau lebih alur yang keluar. Alur yang keluar ditandai untuk menunjukkan alternatif keputusan yang harus diambil




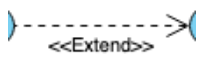
		berdasarkan suatu kondisi yang terpenuhi.
	Panah (arrow)	Penghubung yang menunjukkan hubungan antar objek.
	Basis data (database)	Merepresentasikan penyimpanan data dalam basis data

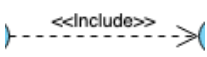
Sumber : R. Mall (2014)

Use-case Diagram

Use-case adalah cara untuk menggambarkan bagaimana pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu (Whitten & Bentley, 2007: 246-251). *Use-case* mendokumentasikan langkah-langkah yang dilakukan oleh aktor untuk berinteraksi dengan sistem, serta respons sistem terhadap tindakan tersebut. Notasi yang sering digunakan dalam *us-case diagram* adalah sebagai berikut.

Tabel 2. Notasi pada *use-case diagram*

Notasi (1)	Nama Notasi (2)	Keterangan (3)
 Actor	Aktor	Entitas eksternal yang berinteraksi dengan sistem. Aktor dapat berupa manusia, perangkat keras lain, ataupun perangkat lunak lain.
 UseCase	Use-case	Urutan langkah-langkah yang berhubungan secara perilaku (suatu skenario), baik yang otomatis maupun manual, dengan tujuan menyelesaikan satu tugas bisnis tunggal
	Asosiasi	Hubungan antara seorang aktor dan use case di mana interaksi terjadi di antara keduanya
 <<Extend>>	Extend	Hubungan antar use-case yang mana use-case arah panah adalah hasil ekstraksi use-case utama dengan memperluas fungsionalitasnya.


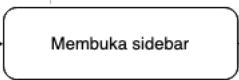


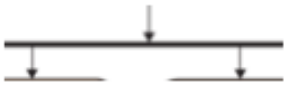
	Include	Hubungan antara use-case dengan abstract use-case untuk mengurangi duplikasi.
---	---------	---

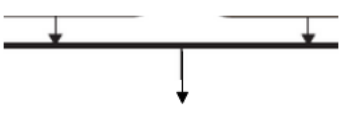

Sumber : Whitten dan Bentley (2007)

Activity Diagram

Activity Diagram adalah salah satu jenis diagram yang digunakan dalam UML (Unified Modeling Language) untuk menggambarkan aliran kerja atau proses bisnis dari suatu sistem (Whitten & Bentley, 2007:390). Secara detail, *activity diagram* menjelaskan bagaimana *workflow* dimulai dan selesai. Selain itu juga menjelaskan apa aktifitas yang terjadi dan urutan kejadian. Simbol umum pada *activity diagram* diilustrasikan pada tabel berikut.

Tabel 3. Notasi pada *activity diagram*

Notasi	Nama Notasi	Keterangan
(1)	(2)	(3)
	Titik awal (Initial node)	Lingkaran padat yang mengawali dari suatu proses
	Aksi (actions)	Persegi panjang dengan sudut tumpul yang menunjukkan tahapan tunggal. Rangkaian dari aksi membuat keseluruhan aktivitas dalam diagram.
	Alur (flow)	Panah pada diagram yang mengindikasikan progres yang melewati aksi. Biasanya alur tidak memerlukan kata-kata untuk mengidentifikasinya, kecuali alur yang keluar dari keputusan.
	Keputusan (decision)	Jajar genjang dengan satu panah alur yang masuk dan dua atau lebih alur yang keluar. Alur yang keluar ditandai untuk menunjukkan suatu kondisi.
	Percabangan (Fork Node)	Fork berupa garis hitam dengan satu alur masuk dan dua atau lebih alur yang keluar. Aktivitas pada aliran paralel setelah fork dapat terjadi secara berurutan ataupun bersamaan.

	Penyatuan (Join Node)	Join berupa garis hitam dengan dua tau lebih alur masuk dan satu alur keluar yang menandakan akhir dari proses yang terjadi S bersamaan. Semua aktivitas yang masuk harus sudah selesai sebelum proses dilanjutkan
	Akhir aktivitas (Activity Final)	Suatu lingkaran padat dalam lingkaran aktivitas berongga yang menunjukkan akhir dari suatu proses.

Sumber : Whitten dan Bentley (2007)

PHP

PHP adalah singkatan dari *Hypertext Preprocessor*. PHP adalah bahasa pemrograman untuk membangun situs web yang interaktif (Gousset et al., 2010). Secara aturan umum, program PHP berjalan di server web dan melayani halaman web untuk pengunjung berdasarkan permintaan. Salah satu fitur utama PHP adalah Anda dapat mencantumkan kode PHP di dalam halaman web HTML, sehingga sangat mudah untuk membuat konten dinamis dengan cepat. Dinamis disini artinya halaman yang isinya dapat berubah secara dinamis setiap kali halaman dilihat. Proses berjalannya skrip PHP pada web server dimulai dari pengunjung yang meminta halaman web dengan mengetik alamat tautan ke *adress bar* peramban. Selanjutnya server web mengenali bahwa alamat tautan adalah skrip PHP, dan menginstruksikan mesin PHP untuk memproses dan menjalankan skrip. Setelah itu skrip berjalan dan ketika selesai akan mengirimkan halaman HTML ke peramban web yang kemudian dilihat pengguna di layar.

Laravel

Laravel adalah kerangka kerja (*framework*) berbasis *open source* untuk bahasa pemrograman PHP (Fauzi, 2020). *Framework* ini menggunakan prinsip

MVC (*Model View Controller*). Laravel bertujuan untuk menyediakan pengalaman bekerja dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. Per tanggal 2 Juni 2024, versi terbaru dari laravel adalah versi 11.0.

Sebagai salah satu *framework* bahasa PHP terbaik, laravel memiliki banyak fitur yang memudahkan dalam pembuatan aplikasi. Salah satu fitur laravel yaitu *command line tool* yang bernama “Artisan” yang dapat digunakan untuk beberapa perintah. Perintah yang dapat dilakukan dengan artisan diantaranya adalah menghasilkan berkas-berkas seperti *view*, *model*, dan *controller*. Selain itu, dengan artisan juga dapat melakukan tugas rutin seperti migrasi basis data, membersihkan *cache*, dan mengoptimalkan aplikasi. Contoh perintah artisan yang sering digunakan adalah `php artisan migrate` dan `php artisan make:controller`.

Selain artisan, laravel juga memiliki fitur Eloquent. Eloquent adalah sebuah pemetaan objek relasional atau biasanya disebut ORM (*Object Relational Mapping*) yang memudahkan pemrogram berinteraksi dengan basis data. Dalam menggunakan eloquent, setiap tabel basis data memiliki model yang berinteraksi dengan tabel tersebut. selain itu eloquent dapat digunakan untuk melakukan serangkaian aksi CRUD (*Create, Read, Update, Delete*) pada basis data.

Livewire

Livewire adalah *library* untuk laravel yang memungkinkan pemrogram untuk membangun antarmuka yang reaktif dan dinamis dengan hanya sedikit bahasa pemrograman javascript (James, 2021). Secara *default*, komponen livewire *dirender* di server dan ditampilkan menggunakan template blade. Keuntungan livewire adalah menghasilkan waktu yang cepat untuk memuat konten karena server tidak perlu menunggu semua bagian yang diperlukan diunduh dan

ditampilkan, sehingga pengguna bisa melihat hasilnya dengan cepat. Livewire dapat digunakan untuk fungsionalitas seperti paginasi halaman, validasi formulir dan unggah fail tanpa harus memuat ulang halaman.

Javascript

Javascript adalah bahasa pemrograman yang bersifat *interpreted*. Javascript tidak langsung dipahami oleh komputer dalam bentuk aslinya. Komputer hanya memahami kode mesin, yang terdiri dari angka biner (nol dan satu) (Wilton & Mcpeak, n.d.). Oleh karena itu, Javascript memerlukan sebuah program khusus yang disebut interpreter untuk mengubah kode Javascript menjadi kode mesin yang dapat dipahami oleh komputer. Javascript merupakan scripting language, dimana tidak seperti program yang berdiri sendiri untuk menjalankan kodenya. Javascript memberikan perintah pada web browser untuk menjalankan apa yang diminta. Web browser kemudian menafsirkan skrip dan melakukan semua pekerjaan yang diperintahkan.

Javascript dan PHP sering digunakan bersama dalam pengembangan web untuk membuat website yang dinamis dan interaktif. Masing-masing memiliki peran yang berbeda. Peran Javascript dalam pemrograman web adalah mengontrol halaman web, membuatnya dinamis dan responsif sedangkan PHP digunakan untuk menghasilkan konten ke halaman web sebelum dikirimkan ke browser pengguna.

Tailwind CSS

Berdasarkan situs resmi dari Tailwind CSS, Tailwind CSS adalah *framework* CSS yang mengutamakan utilitas (*utility-first*) (Tailwind CSS, n.d.). Pendekatan *utility-first* dalam CSS berarti bahwa *framework* ini menyediakan kelas-kelas CSS yang sangat spesifik dan kecil yang dirancang untuk melakukan

satu tugas tertentu, seperti mengatur margin, *padding*, warna, ukuran teks, dan lain sebagainya. Dengan kata lain, pemrogram dapat menggunakan banyak kelas utilitas kecil ini langsung di dalam markup HTML daripada harus menulis kelas CSS yang kompleks untuk elemen yang berbeda.

MySQL

MySQL adalah sebuah sistem manajemen basis data (DBMS) *open-source* yang dikembangkan oleh MySQL AB. Basis data adalah kumpulan data yang terstruktur dan terorganisir dengan cara tertentu sehingga mudah diakses, dikelola, dan diperbarui. Data dalam sebuah basis data dapat mencakup berbagai jenis informasi, seperti teks, angka, gambar, atau video dan biasanya disimpan dalam tabel yang terdiri dari baris dan kolom. DBMS adalah program komputer yang memiliki kemampuan untuk membuat, memproses, dan mengelola basis data. Kata “SQL” pada MySQL adalah singkatan dari “Structured Query Language”, yaitu bahasa standar yang digunakan untuk mengakses basis data. MySQL unggul dalam aplikasi web dibandingkan dengan DBMS lain karena kemampuannya untuk menangani banyak fitur baca dan sedikit fitur tulis. Keunggulan MySQL terletak pada kecepatannya, yang dapat memenuhi tuntutan kecepatan internet (Suehring, 2005).

White-box testing

White-box testing juga biasa disebut dengan *structural testing* atau *glass-box testing*. Teknik ini merupakan teknik pengujian struktural yang merancang kasus pengujian berdasarkan informasi yang diperoleh dari *source code* (Nidhra, 2012). *White-box testing* biasanya dilakukan oleh *developer*. Penguji mengetahui seperti apa *source code* dan menulis skenario pengujian dengan menjalankan fungsi

dengan parameter tertentu. *White-box* bekerja dengan memasukkan sebuah input ke dalam fungsi, dan membuat skenario valid dan tidak valid, lalu mengecek apakah output yang dihasilkan sesuai dengan yang seharusnya fungsi hasilkan atau tidak. *White-box testing* berkaitan dengan mekanisme internal suatu sistem, terutama berfokus kepada aliran kontrol atau aliran data suatu program.

Black-box testing

Black-box testing, yang juga dikenal sebagai pengujian fungsional, sering digunakan untuk validasi perangkat lunak. Dalam pengujian ini, penguji tidak memiliki akses ke *source code*, sehingga tidak perlu memperhatikan detail implementasi internal dari aplikasi. Fokus utama pengujian adalah pada *output* yang dihasilkan dari input yang diberikan serta kondisi eksekusi fungsi. Dalam konteks ini, pengujian berlangsung seolah-olah pengguna hanya dapat memasukkan informasi ke dalam "kotak hitam" dan menerima respons kembali. Menurut Nidhra (2012), pengujian kotak hitam dilakukan sesuai dengan panduan kebutuhan, sehingga penguji mengetahui apa yang seharusnya dihasilkan oleh "kotak hitam" dan memastikan bahwa output sesuai dengan yang diharapkan (Nidhra, 2012).

Black-box testing sangat berguna untuk mengidentifikasi kekurangan atau kebutuhan yang tidak terduga dari perspektif pengguna. Dengan berfokus pada perspektif pengguna akhir, pengujian ini dapat membantu memastikan bahwa perangkat lunak dapat beroperasi sesuai dengan harapan pengguna. Selain itu, keuntungan utama dari *black-box testing* adalah bahwa penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman atau detail teknis lainnya. Hal ini mempermudah proses pengujian dan memungkinkan penguji untuk fokus sepenuhnya pada validasi fungsionalitas dan respons dari aplikasi. Dengan

demikian, pengujian kotak hitam memainkan peran penting dalam memastikan kualitas dan kelayakan penggunaan perangkat lunak.

System Usability Scale (SUS)

Menurut Brooke, *System Usability Scale (SUS)* adalah sebuah skala pengukuran kegunaan sederhana yang terdiri dari sepuluh item (Brooke, 1995). Skala ini memberikan pandangan global tentang penilaian subjektif terhadap kegunaan suatu sistem. SUS dirancang sebagai metode yang cepat dan mudah untuk melakukan penilaian kegunaan pada evaluasi sistem industri dengan biaya yang rendah. Kuesioner SUS disajikan dalam skala *likert*. Skala *likert* adalah skala yang digunakan dalam mengukur sikap, pendapat, dan persepsi seseorang atau orang tentang fenomena sosial (Bahrin et al., 2017). Setiap pertanyaan pada kuesioner SUS memiliki opsi jawaban lima skala *likert* dengan rentang sangat tidak setuju hingga sangat setuju. Menurut penelitian Tullis dkk., untuk mendapatkan hasil yang cukup andal, sampel yang dibutuhkan pada SUS adalah 12-14 responden (Tullis & Stetson, 2004).

Butir pertanyaan dalam SUS disajikan sebagai berikut (Brooke, 1995).

1. I think that I would like to use this system frequently;
2. I found the system unnecessarily complex;
3. I thought the system was easy to use;
4. I think that I would need the support of a technical person to be able to use this system;
5. I found the various functions in this system were well integrated;
6. I thought there was too much inconsistency in this system;
7. I would imagine that most people would learn to use this system very quickly;

8. I found the system very cumbersome to use;
9. I felt very confident using the system;
10. I needed to learn a lot of things before I could get going with this system.

SUS terdiri dari 10 pertanyaan, masing-masing dengan skala *likert* lima poin, yaitu sangat tidak setuju (1), tidak setuju (2), ragu-ragu (3), setuju (4), sangat setuju (5). Setiap pertanyaan memiliki kontribusi skor. Pertanyaan bernomor ganjil (1, 3, 5, 7, 9), kontribusi skornya adalah skor jawaban – 1 (Brooke, 1995). Misalnya, jika seorang responden memilih skor 4 untuk pertanyaan nomor 1, maka kontribusi skornya adalah 4 - 1 = 3. Sedangkan pertanyaan bernomor genap (2, 4, 6, 8, 10) kontribusi skornya adalah 5 – skor jawaban (Brooke, 1995). Misalnya, jika seorang responden memilih skor 4 untuk pertanyaan nomor 2, maka kontribusi skornya adalah 5 - 4 = 1. Skor SUS dari setiap responden adalah total skor kontribusi dari seluruh pertanyaan dikali 2,5. Skor SUS adalah rata-rata dari skor SUS semua responden.

Nilai rata rata=
$$\sum_{i=1}^n x_i / N$$

dimana: x_i : nilai score responden
 N : Jumlah Responden

Penilaian berdasarkan 3 (tiga) kategori:

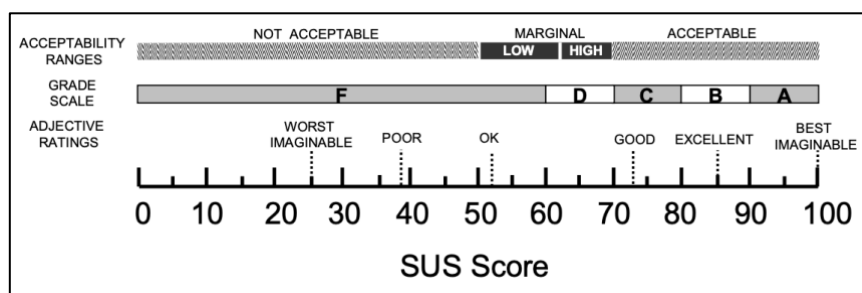
- a. Not Acceptable = skor 0-50,9
- b. Marginal = skor 51-70,9
- c. Acceptable = skor 71-100

Sumber : Sri Handayani & Adelin (2019)

Gambar 2. Rumus Skor SUS

Dalam penelitian yang dilakukan oleh Bangor, Kortum, dan Miller, skor System Usability Scale (SUS) dapat diinterpretasikan dengan menggunakan skala

kata sifat (*adjective rating scale*) dan skala penilaian huruf (*letter grade scale*) sebagai tambahan untuk memberikan pemahaman yang lebih mendalam tentang tingkat kegunaan suatu antarmuka. Skor SUS yang diperoleh dari responden dapat dihubungkan dengan skala kata sifat yang memberikan label subjektif untuk menilai kegunaan produk secara relatif. Selain itu, dengan penambahan skala penilaian huruf, seperti skala penilaian sekolah tradisional (A, B, C, D, F), skor SUS dapat diartikan dalam konteks yang lebih familiar dan mudah dipahami oleh berbagai pihak yang terlibat dalam pengembangan produk. Dengan pendekatan ini, interpretasi skor SUS memberikan gambaran yang lebih komprehensif tentang tingkat kegunaan suatu antarmuka, sehingga membantu dalam pengambilan keputusan terkait perbaikan dan pengembangan produk secara efektif .



Sumber : Bangor et al. (2009)

Gambar 3. Perbandingan Tingkatan Skor SUS

2.2 Penelitian Terkait

Terdapat sejumlah penelitian terdahulu yang berkaitan dengan topik penelitian ini. Penelitian yang dilakukan oleh Prawardana & Prasetyo (2022) menjelaskan tentang pembangunan sistem informasi survei kepuasan internal kinerja perangkat daerah berbasis web di lingkungan Kabupaten Tuban. Sistem informasi ini dikembangkan guna membantu Badan Pembangunan Daerah,

Penelitian dan Pengembangan (Bappeda Litbang) Kabupaten Tuban dalam pelaksanaan Survei Kepuasan Internal Kinerja Perangkat Daerah di Lingkungan Pemerintah Kabupaten Tuban (Prawardana & Prasetyo, 2022). Kesamaan penelitian penulis dengan penelitian ini adalah aplikasi yang dibuat yaitu sistem informasi survei kepuasan berbasis web. Penelitian ini juga memiliki bahasa pemrograman yang sama dengan penelitian penulis yaitu bahasa pemrograman PHP.

Perbedaan penelitian ini dengan penelitian penulis adalah ruang lingkup penelitiannya. Pada penelitian ini sistem digunakan di lingkungan Kabupaten Tuban, sedangkan penulis akan membangun sistem di lingkungan Politeknik Statistika STIS. Metode pengembangan sistem yang digunakan juga berbeda. Penelitian ini menggunakan metode pengembangan sistem SDLC pendekatan waterfall, sedangkan penelitian penulis menggunakan metode pengembangan sistem SDLC pendekatan iterative waterfall.

Penelitian oleh Putri et al (2022) dengan judul Perancangan Sistem Informasi Survei Kepuasan Pengguna Layanan Berbasis Website untuk Penjaminan Mutu Internal di FMIPA UNJ. Penelitian ini dilakukan karena adanya permasalahan dari penyelenggaraan survei instrumen kepuasan di SPM UNJ yang sedang berjalan tidak langsung menampilkan visualisasi secara *real time* (Putri et al., 2022). Penelitian ini menghasilkan sistem yang dapat menampilkan visualisasi data secara *real time* dan adanya peraturan hak akses bagi responden.

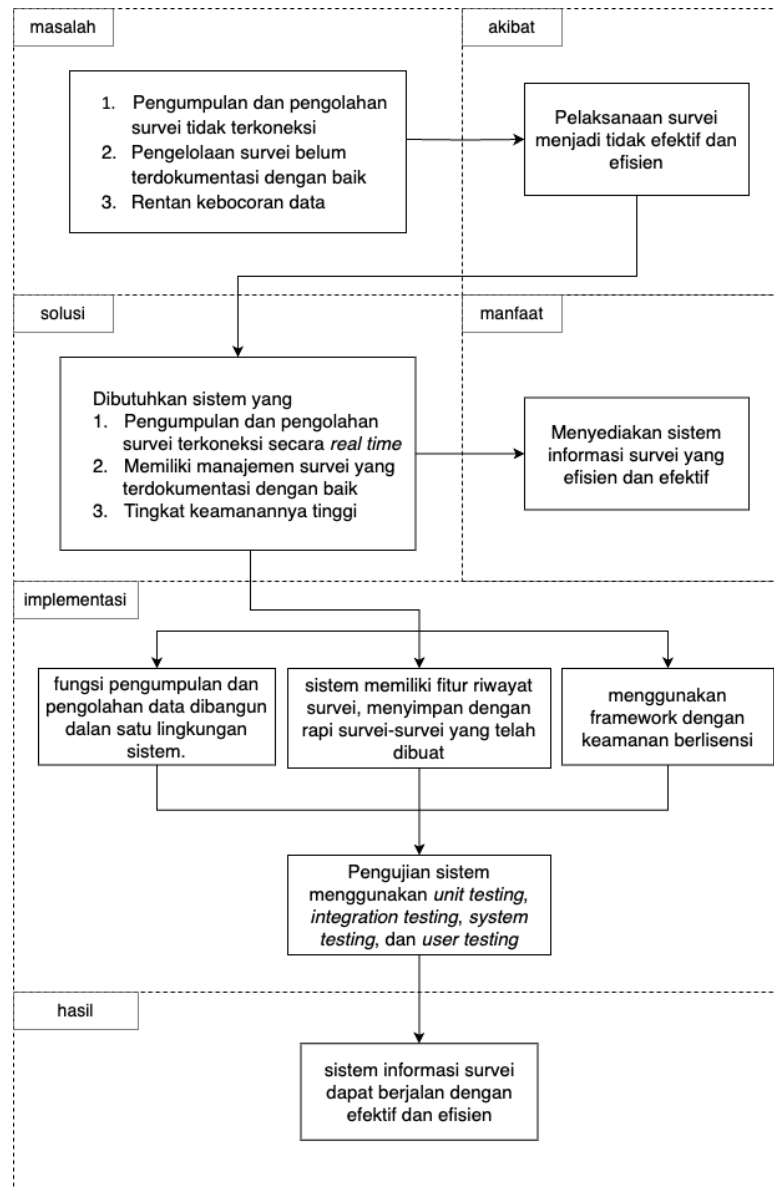
Kesamaan penelitian ini dengan penelitian penulis adalah aplikasi yang dibangun, yaitu sistem informasi survei kepuasan. Selain itu, objek penelitian juga analog, yaitu sebuah institusi perguruan tinggi. Perbedaan penelitian ini dengan

penelitian penulis adalah metode pengembangan sistem dan *framework* yang digunakan. Penelitian ini menggunakan SDLC pendekatan spiral dan *framework* Codeigniter 4, sedangkan penulis menggunakan SDLC pendekatan *iterative waterfall* dan *framework* Laravel 10.

Kamsurya (2021) dalam penelitiannya membahas mengenai kualitas data pada web survei. Penelitian ini berfokus pada kendala yang muncul dalam mendapatkan data kualitas tinggi melalui survei web. Proses ini sering kali menimbulkan berbagai masalah yang mengakibatkan kesalahan pada pengukuran sistematis dan menghasilkan pengukuran yang bias. Beberapa faktor yang dapat memengaruhi mutu pengumpulan data melalui survei web mencakup pemilihan sampel, desain instrumen pertanyaan, perangkat yang digunakan untuk mengisi survei, tampilan antarmuka, dan penggunaan pengingat dalam survei (Kamsurya et al., 2021).

Kalantari D. et al. (2021) menjelaskan dalam penelitiannya yang berjudul “*E-survey (surveys based on e-mail & web)*” membahas tentang jenis-jenis e-survei, metode pelaksanaannya, serta manfaat dan kekurangannya. Menurut penelitian ini, perkembangan e-survei terletak pada metode surveinya yang lebih ekonomis dan efisien dibandingkan dengan survei dengan kertas (Kalantari D. et al., 2011). Manfaat dari penggunaan e-survei adalah akses sampel yang luas, kemudahan dalam pelaksanaan, kecepatan dalam pengolahan data, dan efisiensi biaya.

2.3 Kerangka Pikir



Gambar 4. Kerangka Pikir

Seperti pada Gambar 4, penelitian ini dimulai dari penemuan berbagai kendala yang terjadi selama pelaksanaan sistem yang berjalan. Permasalahan yang terjadi diantaranya, survei tidak terkoneksi secara *real time*, dokumentasi yang belum baik, dan isu *human error*. Hal itu berakibat pada pelaksanaan survei yang tidak efektif dan efisien. Dari permasalahan tersebut diperlukan solusi, yaitu

dibutuhkan sistem yang dapat terkoneksi secara *real time*, dokumentasi yang baik, dan keamanan tinggi. Solusi ini diimplementasikan melalui pembangunan proses survei secara menyeluruh, penyimpanan riwayat survei, dan penerapan framework dengan tingkat keamanan yang terjamin. Dengan itu terciptanya sistem informasi survei kepuasan yang efektif dan efisien.