

BAB II TINJAUAN PUSTAKA

2.1 Teori Dasar

1. Sistem Informasi

Buku yang berjudul “SISTEM INFORMASI MANAJEMEN” (Wijoyo 2021) yang diterbitkan oleh Penerbit Insan Cendekia Mandiri dijelaskan mengenai manajemen sistem informasi merupakan sejalur sub-sistem informasi yang terstruktur sebagai suatu sistem berbasis komputer. Buku ini menjelaskan prinsip-prinsip dasar sistem informasi manajemen dan komponen manajemen dalam sebuah sistem. Peran sistem informasi sangat penting dalam mengelola data menggunakan bukti dari flow diagram untuk meningkatkan perkembangan data dalam sebuah sistem memudahkan pemahaman logika dan struktur sistem serta menyimpan data historis dari masa lalu dan sekarang untuk digunakan sebagai informasi di masa depan seperti dalam sistem informasi manajemen persediaan.

2. Konsep Dasar Sistem

Menurut R.Djoht (2017) Sistem adalah kumpulan objek untuk saling tergabung melalui jaringan yang berkaitan atau saling tergantung. Objek-objek ini bisa berupa kumpulan unit unit berbedaan Dan bergerak dalam satu kesatuan yang terintegrasi. Sistem minimal memiliki tiga komponen utama: input proses dan output menurut Mujilan (2015).

Menurut B. Romney (2015) Bagian sistem adalah kumpulan dari dua mungkin lebih elemen yang berinteraksi agar mencapai sebuah tujuan. Dan karena itu sistem terdiri dari komponen-komponen seperti masukan metode dan suatu proses yang saling berinteraksi dalam sebuah kesatuan agar termencapainya tujuan tertentu.

Karakteristik sistem menurut Jeperson Hutahea (2015) adalah sebagai berikut :

a. Komponen

Komposisi yang terdiri dari unsur yang berinterkoneksi menjadi bagian kesatuan.

b. Batasan Sistem (boundary)

Memisahkan sebuah sistem dari bagian sistem lain atau dari bagian lingkungan luarnya.

c. Lingkungan Luar Sistem (environment)

Area di luar batasan sistem yang mempengaruhi bagian operasi sistem.

d. Penghubung Sistem (interface)

Sebuah Media yang memungkinkan interaksi antara bagian subsistem.

e. Masukan Sistem (input)

Daya yang disisipkan ke dalam bagian sistem.

f. Keluaran Sistem (output)

Efek dari bagian energi yang dikelola oleh sistem.

g. Pengolah Sistem

Suatu sistem menjadi bagian pengolah yang dapat mengubah masukan menjadi keluaran.

h. Sasaran Sistem

Memastikan sebuah masukan yang diperlukan dan buatan yang diharapkan.

3. Konsep Dasar Informasi

a. Definisi Data

Menurut Marshall B. Romney (2015) data adalah fakta-fakta yang dikumpul, direkam, disimpan, dan diproses oleh sistem informasi.

Gordon B. Davis (dalam Hutapean Jeperson 2015) menjelaskan data sebagai materi dasar dalam pembentukan informasi terdiri dari simbol -simbol yang tidak teratur yang mewakili nilai tindakan dan lain -lain. Metode pengumpulan data menurut Hutapean (2015) meliputi pengamatan langsung wawancara perkiraan korespondensi dan daftar pertanyaan.

b. Definisi Informasi

Ungkapan B. Romney (2015) informasi adalah bukti yang diproses dan terorganisir menghasilkan suatu pengertian serta dapat memajukan proses dalam pengutipan keputusan.

Davis (dalam Jeperson 2015) menerangkan bahwasanya informasi adalah sebuah data yang diolah sehingga membentuk bagian hal penting bagi penerima hingga mendapatkan kualitas yang nyata dalam pengambilan keputusan. I

Informasi memainkan peran penting dalam meningkatkan pemahaman dan mengurangi ketidakpastian bagi pengguna serta bermanfaat sebagai indikator bagi pengambil keputusan. Kegunaan informasi dipengaruhi oleh tujuan penerima ketelitian penyampaian dan pengolahan data waktu ruang atau tempat bentuk dan semantik. Karakter informasi yang berguna tergantung pada relevansi kehandalan kelengkapan ketepatan waktu kemudahan pemahaman keabsahan dan kemudahan akses

4. Pengertian *Point Of Sales*

Point Of Sale (POS) adalah aspek dari kegiatan bisnis yang terfokus pada penjualan dan memainkan peran penting dalam membantu proses transaksi. Perangkat lunak POS menjadi komponen kunci dari sistem POS karena menentukan langkah-langkah proses mengatur apa yang perlu dilakukan dan bagaimana melakukannya. Sementara perangkat keras POS sangat diperlukan untuk memfasilitasi proses pembayaran serta menghasilkan tanda terima pelanggan.

i. Bisnis Kuliner

Jurnal penelitian “Perancangan aplikasi POS berbasis web untuk efisiensi antrean pada restoran serba sambal” yang diteliti oleh (Fitriana , Lsitiyoko 2020) dibahas tentang perancangan sistem POS berbasis web pada restoran serba sambal dengan tujuan meningkatkan efisiensi antrean menggunakan program PHP dan Maria DB. Penelitian ini bertujuan untuk mendapatkan kinerja yang optimal dengan kombinasi kualitas rasa dari bahan baku makanan yang fresh penyajian yang menarik sederhana dan elegan serta pelayanan yang luar biasa. Ini menunjukkan bahwa penggabungan sistem informasi dengan strategi bisnis dapat menghasilkan ide yang inovatif.

b. Tinjauan Studi

i. Perancangan POS

Berbagai hasil dari penelitian terkait yang akan dijadikan sebagai referensi. Jurnal penelitian mengenai analisis dan perancangan berjudul “Perancangan Sistem Informasi *Point Of Sales* (POS) berbasis web pada Rumah Makan Kokobop Chicken” yang diteliti oleh (Sugumonrong,Ray,Victori 2019). membahas perancangan sistem POS pada rumah makan menggunakan PHP dan MySQL untuk pengelolaan dan penyimpanan data dengan kesimpulan bahwa pemilik usaha mendapatkan manfaat maksimal.

Jurnal penelitian tentang rancang bangun berjudul “Rancang bangun sistem informasi software *Point Of Sale* (POS) dengan metode waterfall berbasis web” yang diteliti oleh (Putu Gede Surya Cipta Nugraha, Ni Wayan Wardani, I Wayan Sukarmayasa, 2021) menjelaskan rancangan bangun sistem informasi POS untuk digunakan kepada bisnis minimarket yang berfokus pada inventory dan POS dengan kesimpulan dapat membantu beberapa pelaku wirausaha dalam mengelola arus bisnis usahanya terkhusus dalam bidang administrasi dan kasir.

ii. Rancang Bangun POS

Jurnal penelitian tentang rancang bangun berjudul “Rancang bangun sistem informasi software point of sale (POS) pada Toko Azam Grosir Berbasis Website dengan Metode Waterfall oleh Herdiansyah Pratama Octavia Baehaqi Saifudin dan Desyan (2021) menggunakan metode waterfall pada toko grosir mempermudah proses transaksi penjualan. Jurnal lainnya “Rancang Bangun mengenai Sistem Point of Sale Berbasis Web menggunakan Metode Pengembangan Waterfall untuk Kasir Syams Boutique” oleh Sari dan Marsa (2022) menunjukkan penerapan sistem informasi menggunakan metode waterfall pada toko butik. Kesimpulannya aplikasi Point of Sale (POS) pada kasir mempunyai manfaat signifikan dalam memajukan kinerja serta pelayanan karena efisiensi dalam pendataan transaksi dan pengelolaan data. Dengan merujuk pada penelitian ini dapat diharapkan bahwa perancangan dan implementasi sistem POS berbasis web akan memberikan kontribusi positif dalam meningkatkan efisiensi dan kualitas layanan pada berbagai jenis bisnis termasuk rumah makan minimarket dan butik.

iii. Pengertian Aplikasi

Aplikasi adalah program atau metode perangkat lunak yang memiliki sistematis dalam mengelola data berfungsi untuk mendukung berbagai aktivitas manusia dan menghasilkan informasi yang bermanfaat bagi pengguna (Huda & Priyatna).

Secara etimologis aplikasi berasal dari kata *application* yang mengacu pada program atau penerapan yang membantu dalam menjalankan tugas atau aktivitas yang dilakukan oleh pengguna (Azis).

iv. Aplikasi Web

Aplikasi web mengacu pada aplikasi situs web yang dioptimalkan khusus untuk digunakan di berbagai platform perangkat. Program ini dibangun dengan metode menggunakan standar teknologi web diantaranya adalah JavaScriptCSS dan HTML dan dapat diakses melalui browser di berbagai jenis platform perangkat (Prasetya). Namun penting untuk dicatat bahwa aplikasi web hanya dapat beroperasi secara online dan memerlukan koneksi internet yang tersedia pada perangkat yang digunakan. secara *online* dan bergantung pada koneksi yang terhubung dengan *device*.

v. Progressive Web Application (PWA)

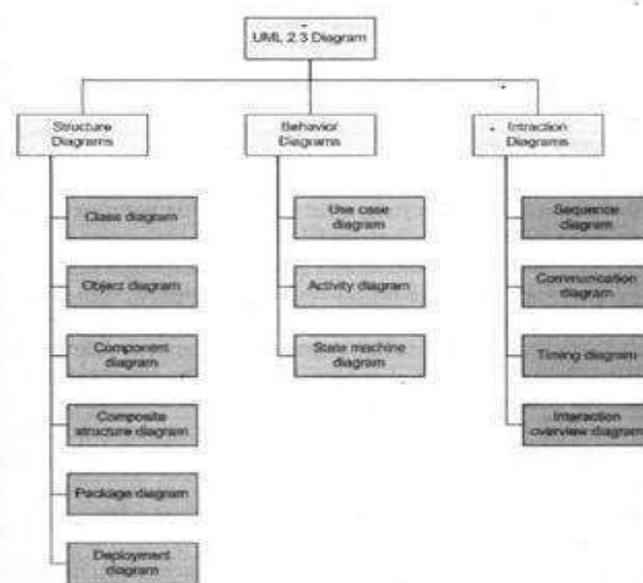
Progressive Web Application (PWA) adalah hasil pengembangan dari teknologi HTML5 yang menggabungkan karakteristik aplikasi native web dan desktop. PWA memiliki kemampuan fungsionalitas untuk bekerja dengan cara offline hingga dapat diakses dengan berbagai perangkat mobile web (Angga Dewi). Ciri-ciri PWA termasuk kemampuan progresif responsif independen konektivitas dan keamanan (Tandel & Jamadar).

vi. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa permodelan yang digunakan dengan visualisasi merancang juga mendokumentasikan sistem piranti lunak. UML menyediakan standar untuk merancang model sebuah sistem dan terdiri dari berbagai macam diagram salah satunya adalah Class Diagram Activity Diagram Use Case Diagram (Yasin Nugroho Rosa & Shalahuddin).

vii. System Development Life Cycle (SDLC)

Untuk Siklus Pengembangan Perangkat Lunak atau Software Development Life Cycle (SDLC) ialah rangkaian proses yang digunakan dalam pengembangan perangkat lunak yang mencakup tahap pengembangan pemeliharaan dan penggunaan sistem informasi (Rosa & Shalahuddin Kadir). Untuk Jenis yang sering digunakan dalam SDLC adalah jenis Waterfall yang menjadi pendekatan sekuensial linier dalam pengembangan perangkat lunak (Saxena & Upadhyay Yasin). Penjelasan mengenai tahapan model Waterfall meliputi bagian dari analisis kebutuhan Implimentasi desain sistem dan perangkat lunak integrasi dan juga pengujian serta operasi dan pemeliharaan (Rosa & Shalahuddin Yasin). Bagian Model Waterfall merupakan salah satu kumpulan pengembangan dari metode SDLC yang mengalir dengan menyelesaikan satu tahap selanjutnya terintegrasi ke tahap berikutnya seperti air terjun. Dengan demikian pengembangan aplikasi baik berbasis web maupun progresif sering melibatkan penggunaan UML untuk merancang model sistem serta mempertimbangkan tahapan-tahapan dalam SDLC seperti model Waterfall untuk memastikan pengembangan perangkat lunak yang efektif dan efisien.



Gambar 2. 1 Diagram Pengelompokan UML

Dari 13 macam diagram UML tersebut, penulis mengambil 3 diagram UML sebagai bahan referensi pustaka yaitu *Use Case Diagram*, *Activity Diagram* dan *Class Diagram* :

1. Use Case Diagram

Use case atau *use case diagram* merupakan permodelan pada kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* menunjukkan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* juga digunakan agar mengetahui fungsi yang terdapat didalam sebuah sistem informasi dan dapat diakses oleh semua orang fungsi- fungsi itu (S & Shalahudin, 2015).

Berikut adalah simbol-simbol yang digunakan dalam *use case diagram* (S & Shalahudin, 2015):

Tabel 2. 1 Simbol *Use case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> <p>nama use case</p>	<p>Fungsionalitas disediakan oleh sistem sebagai unit yang mampu saling bertukar pesan antar unit atau aktor. Umumnya dinyatakan menggunakan suatu kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor / actor</p> <p>nama aktor</p>	<p>Orang, proses, dan sistem lain yang memiliki interaksi pada sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor merupakan gambar orang, tetapi aktor belum tentu merupakan orang; umumnya dinyatakan dengan kata benda di awal frase nama aktor</p>

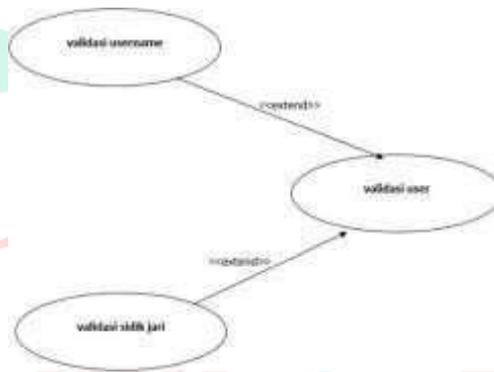
<p>Asosiasi / Association</p> <p>—————</p>	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
--	---

Ekstensi / Extend

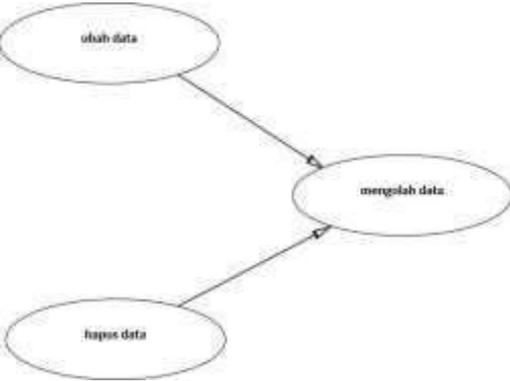
<<extend>>



Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, ditambahkan, misal:



Arah panah mengarah pada *use case* yang ditambahkan, biasanya *use case* yang menjadi extend-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.

<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Contoh:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / include / uses</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas memberikan gambaran aktivitas sistem bukan apa yang dilakukan aktor sehingga aktivitas dapat dilakukan oleh sistem (S & Shalahudin, 2015).

Berikut tabel simbol-simbol yang digunakan dalam *activity diagram* (S & Shalahudin, 2015) :

Tabel 2. 2 Simbol Activity Diagram

Simbol	Deskripsi
 Status awal / start	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas / activity	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Penggabungan / join	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Status akhir / Final	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimline	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

3. Class Diagram

Diagram kelas atau *Class Diagram* memberikan gambaran struktur sistem berdasarkan pendefinisian kelas-kelas yang akan dibuat agar membangun sistem. Kelas terdapat atribut dan metode atau operasi (S & Shalahudin, 2015).

Berikut adalah simbol-simbol yang ada pada *class diagram* (S & Shalahudin, 2015) :

Tabel 2. 3 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas / Class</p> 	<p>Kelas pada struktu sistem</p>
<p>Antarmuka / <i>user interface</i></p>	<p>Sama dengan konsep <i>user interface</i> dalam pemrograman berorientasi objek</p>
<p>Asosiasi / <i>association</i></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i></p>
<p>Asosiasi berarah / <i>directed association</i></p>	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i></p>

<p>Generalisasi / generalization</p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)</p>
<p>Ketergantungan / dependency</p> 	<p>Relasi antar kelas dengan makna kbergantungan antar kelas</p>
<p>Agregasi / Aggregation</p> 	<p>Relasi antar kelas dengan makna semua-bagian (whole-part)</p>

Unified Modelling Language (UML) merupakan bahasa permodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ yang digunakan pada visualisasi, peancangan dan dokumentasi sistem piranti lunak yang salah satunya terdiri dari *use case diagram*, *activity diagram* dan *class diagram*.

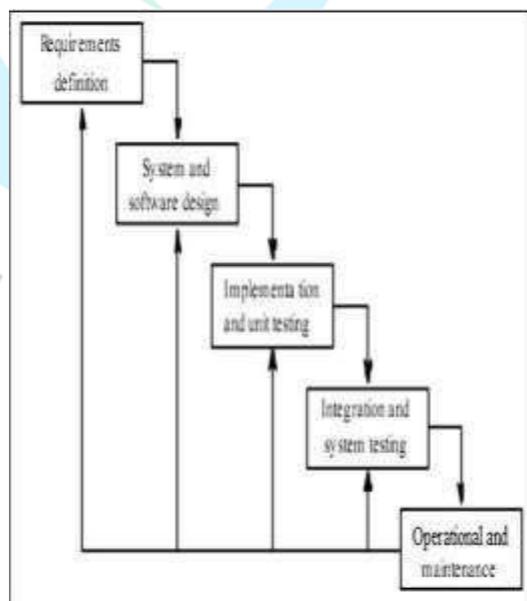
1. System Development Life Cycle (SDLC)

Menurut Rosa dan Shalahuddin (2015, hal. 26), SDLC (*software development life cycle*) atau dikenal dengan *system development life cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model yang umum digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik). Kadir (2014), SDLC atau *software development life cycle* merupakan sebuah alur sistem yang digunakan dalam proses pengembangan yang meliputi tahap pengembangan, pemeliharaan, serta penggunaan sistem informasi.



Gambar 2.5 System Development Life Cycle

Salah satu model yang terdapat dalam SDLC adalah model *Waterfall*. Menurut Saxena dan Upadhyay (2016:1012), model *Waterfall* merupakan model siklus pengembangan perangkat lunak sekuensial linier (SDLC)”. Menurut Rosa dan Shalahuddin (2015, hal. 28), model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung. Menurut Yasin (2012, hal. 18), sekuensial linier atau yang sering disebut siklus air terjun (*waterfall*) dikenal dengan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, desain, kode dan pengujian.



Gambar 2.6 Ilustrasi alur model waterfall (Sommerville 2011)

Penjelasan fase berurutan dalam model *waterfall*, yaitu sebagai berikut:

1. *Requirement Analysis and Definition*

Proses pengumpulan kebutuhan yang dilakukan secara intensif selanjutnya dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh perangkat lunak yang akan. Tahap ini dilakukan proses identifikasi spesifikasi kebutuhan perangkat lunak apa yang dibutuhkan karena perangkat lunak nantinya dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan lain sebagainya.

2. *System and Software Design*

Desain perangkat lunak merupakan suatu proses multi-langkah berfokus dengan desain pembuatan program perangkat lunak, khususnya struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mengubah bagaimana perangkat lunak diperlukan untuk mengubah kebutuhan analisis menjadi representasi desain yang dapat diimplementasikan sebagai program di tahap selanjutnya.

3. *Implementation and Unit Testing*

Desain harus diterjemahkan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. *Integration and System Testing*

Pengujian berfokus pada perangkat lunak dalam segi logik dan fungsional. Memastikan semua bagian telah diuji. Hal tersebut dilakukan agar meminimalisir kesalahan (*error*) serta memastikan hasil yang diperoleh sesuai dengan yang diinginkan.

5. *Operation and Maintenance*

Sebuah perangkat lunak dapat mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan dapat terjadi karena terdapat kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan dimulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak membuat perangkat lunak baru.

Berdasarkan uraian di atas dapat disimpulkan bahwa pengembangan *Waterfall* merupakan salah satu siklus pengembangan dari metode SDLC. Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier atau alur hidup klasik. Tahapan *waterfall* mengalir dengan menyelesaikan satu tahap selanjutnya pindah ke tahap berikutnya seperti air terjun. Terdapat beberapa tahap, seperti analisis kebutuhan perangkat lunak, desain, pengkodean, dan pengujian.

