

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Rancang Bangun

Menurut Maulani, G., Septiani, D., & Sahara, P. N. (dalam Girsang, 2018), “Rancang Bangun” diartikan sebagai proses pembuatan dan pengembangan aplikasi atau sistem yang sebelumnya belum ada di suatu instansi atau objek. secara khusus. Sementara itu, pandangan lain menjelaskan bahwa desain dan bangun adalah suatu proses pengembangan sistem, baik untuk membuat sistem baru maupun mengganti atau memperbaiki sistem yang sudah ada, baik secara keseluruhan maupun sebagian.

Maka dari itu, rancang bangun dapat diartikan sebagai proses perencanaan dan pembuatan suatu produk, sistem, atau solusi. Dalam konteks pengembangan perangkat lunak, rancang bangun mengacu pada tahapan di mana spesifikasi yang telah ditetapkan dalam tahap analisis diubah menjadi desain yang lebih rinci dan terperinci. Ini melibatkan pembuatan struktur sistem, antarmuka pengguna, algoritma, serta pemodelan data yang diperlukan untuk membangun aplikasi atau sistem yang diinginkan. Rancang bangun merupakan langkah kunci dalam siklus pengembangan perangkat lunak, yang kemudian akan menjadi dasar untuk proses implementasi dan pengujian selanjutnya.

2.1.2 Aplikasi

Aplikasi adalah perangkat lunak yang berisi kode atau instruksi yang dapat disesuaikan sesuai kebutuhan. Aplikasi dirancang untuk mendukung berbagai aktivitas komputerisasi yang dilakukan oleh pengguna. Menurut Rahayu, Fajri, dan Hambali (2019), aplikasi juga didefinisikan sebagai program komputer yang dibuat untuk menyelesaikan masalah tertentu.

Aplikasi berfokus pada perancangan sistem untuk memproses data dengan menggunakan aturan atau bahasa pemrograman tertentu. Tujuan utamanya adalah untuk melaksanakan tugas spesifik yang dibutuhkan oleh pengguna.

Berdasarkan pengertian tersebut, dapat diartikan bahwa aplikasi adalah perangkat lunak yang terdiri dari kode atau perintah yang dapat disesuaikan sesuai dengan kebutuhan atau keinginan pengguna. Aplikasi dirancang untuk mendukung berbagai aktivitas komputerisasi yang dilakukan oleh pengguna, dengan tujuan untuk memecahkan masalah tertentu.

2.1.3 Website

Menurut (Abdullah et al., 2016), website atau disingkat web, dapat diartikan sebagai suatu kumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi berupa data digital, baik berupa teks, gambar, video, audio dan animasi lainnya disediakan melalui koneksi Internet. Menurut pendapat lain, website merupakan media online yang memiliki nilai tambah yang signifikan bagi pemilik dan penggunanya. Website memberikan berbagai fungsi dan kenyamanan bagi penggunanya. Dengan adanya website, tujuan perusahaan atau pebisnis individu dapat tercapai dengan cara yang efektif dan efisien. (Yeni Susilowati, 2019).

Adapun beberapa jenis – jenis dari website menurut (Abdulloh, 2018) yaitu website statis, website dinamis, dan website interaktif, dapat diperjelas seperti sebagai berikut :

a. Website Dinamis

Jenis website ini menyediakan konten atau isi yang selalu berubah setiap saat. Contohnya adalah media berita daring.

b. Website Statis

Website statis memiliki konten yang jarang diubah. Misalnya, profil organisasi atau halaman informasi tetap.

c. Website Interaktif

Jenis website ini memungkinkan pengguna untuk berinteraksi secara aktif dengan kontennya. Contoh website interaktif adalah forum diskusi, platform sosial media, dan aplikasi e-learning.

2.1.4 *Software Development Life Cycle (SDLC)*

Menurut (Dennis, 2015), SDLC memiliki empat tahap utama yang serupa, yaitu perencanaan, analisis, desain, dan implementasi. Meskipun proyek yang berbeda mungkin menekankan aspek tertentu dari SDLC atau menggunakan pendekatan yang berbeda untuk tiap tahap, setiap proyek tetap melibatkan elemen dari keempat tahap tersebut. Setiap tahap terdiri dari rangkaian langkah yang memanfaatkan teknik tertentu untuk mencapai hasil yang diinginkan. Berikut adalah beberapa tahapan dalam SDLC menurut Dennis :

a. Perencanaan (*Planning*)

Tahap pertama dalam pengembangan dan pembuatan sistem adalah perencanaan. Tujuan utama dari tahap ini adalah untuk memahami alasan perusahaan membangun sistem informasi. Analisis kelayakan dan pengendalian arah proyek melalui fase SDLC merupakan langkah-langkah yang perlu dilakukan pada tahap perencanaan. Pada tahap ini, analisis kelayakan mencakup aspek kelayakan ekonomi, teknis, dan organisasi.

b. Analisis (*Analysis*)

Tahap analisis dilakukan dengan tujuan untuk mengevaluasi sistem yang sedang berjalan. Hasil dari analisis ini digunakan sebagai dasar untuk perencanaan dan pengambilan keputusan dalam pengembangan sistem. Evaluasi terhadap sistem yang ada bertujuan untuk memahami kondisi sistem saat ini dan sebagai acuan dalam pembuatan sistem baru. Survei atau kuesioner digunakan untuk mengumpulkan informasi mengenai kebutuhan sistem baru dan untuk mengetahui operasi bisnis yang terkait dengan sistem yang akan dikembangkan. Perancangan sistem baru juga memerlukan persetujuan dari pengguna serta pemenuhan persyaratan

bisnis yang harus dipenuhi oleh sistem baru. Persyaratan tersebut kemudian dijabarkan dalam proposal sistem.

c. Desain (*Design*)

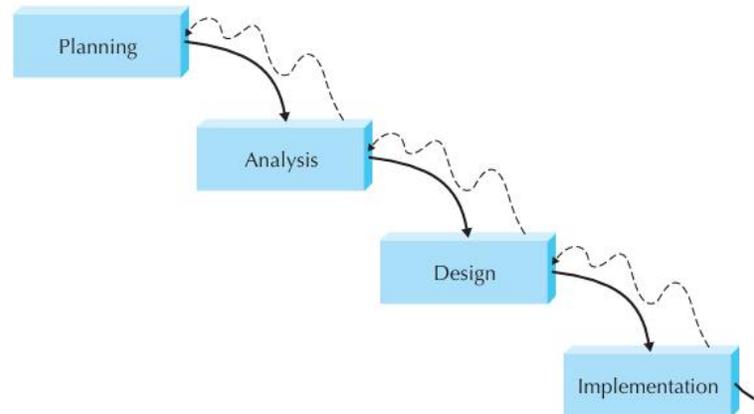
Tahap desain merupakan tahap perancangan sistem yang berfokus pada spesifikasi rancangan. Hasil akhir dari tahap ini mencakup rancangan kontrol, sistem, basis data, masukan, dan keluaran. Verifikasi sistem yang akan dibangun merupakan langkah pertama dalam proses desain. Langkah kedua melibatkan pembuatan desain arsitektur dasar, yang menggambarkan infrastruktur dan perangkat yang akan digunakan, menentukan spesifikasi sistem yang akan dikembangkan, serta lokasi basis data. Pada tahap terakhir, pengembangan sistem dilakukan dengan menentukan program yang akan digunakan selama tahap implementasi.

d. Implementasi (*Implementation*)

Tahap implementasi dimulai dengan pengkodean sistem informasi yang dapat dijalankan oleh komputer menggunakan bahasa pemrograman. Tahap ini terdiri dari tiga langkah, yaitu pembangunan sistem menggunakan bahasa pemrograman dan pengujian sistem yang dilakukan secara bersamaan untuk menghindari kesalahan atau bug. Langkah kedua mencakup konversi dan pelatihan pengguna, dengan konversi yang dapat dilakukan melalui salah satu dari tiga strategi, yaitu konversi bertahap, konversi paralel, dan cutover langsung. Langkah terakhir adalah menyusun rencana dukungan sistem, yang mencakup tujuan pasca-implementasi dan penyediaan buku panduan pengguna untuk memudahkan penggunaan sistem.

Melalui SDLC, tim pengembang dapat memahami kebutuhan pengguna, merancang solusi yang tepat, dan memastikan bahwa sistem yang dihasilkan memenuhi standar kualitas yang diinginkan. SDLC juga memungkinkan untuk pengelolaan proyek yang efisien dan pemantauan kemajuan pengembangan perangkat lunak dari awal hingga akhir. Dengan menerapkan tahapan-tahapan SDLC dengan cermat, organisasi dapat

mengurangi risiko, mengoptimalkan penggunaan sumber daya, dan menghasilkan produk atau sistem yang berkualitas tinggi sesuai dengan kebutuhan pengguna dan tujuan bisnis. Pada **Gambar 2.1** dapat dilihat empat tahapan tersebut.



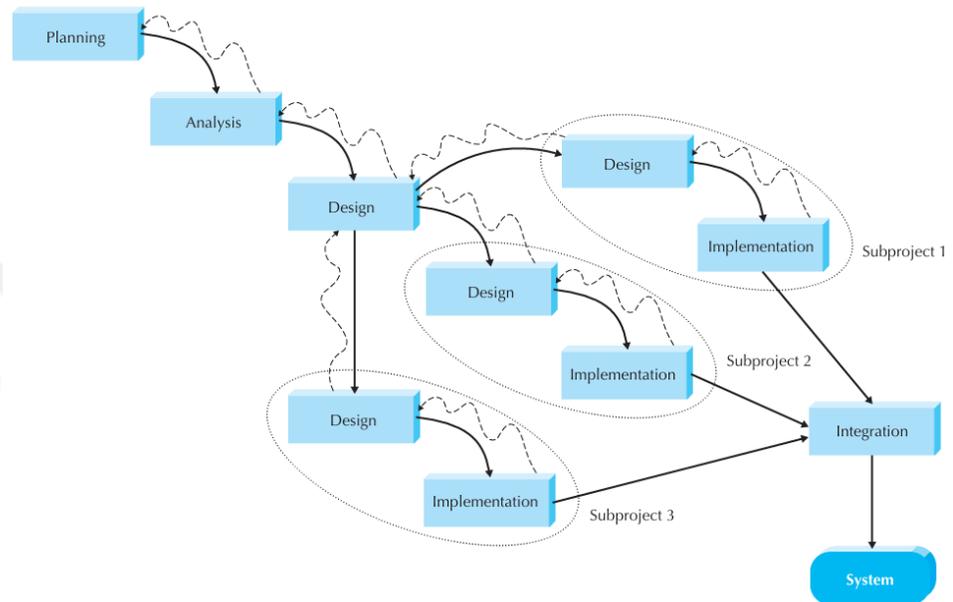
Gambar 2. 1 Tahapan - Tahapan SDLC (Dennis, 2015).

2.1.5 *Rapid Application Development (RAD)*

Menurut (Dennis, 2015), Rapid Application Development (RAD) adalah pendekatan pengembangan sistem yang menekankan kecepatan dalam proses pengembangan melalui penggunaan teknik-teknik khusus dan alat bantu tertentu. Metodologi ini dirancang untuk mempercepat analisis, desain, dan implementasi, sehingga sistem dapat segera digunakan oleh pengguna.

RAD memperpendek waktu pengembangan dengan mengadaptasi fase-fase dalam Software Development Life Cycle (SDLC) dan memperkenalkan teknik seperti Computer-Aided Software Engineering (CASE), sesi desain bersama (Joint Application Design atau JAD), penggunaan bahasa pemrograman generasi keempat, hingga generator kode otomatis. Pendekatan ini memungkinkan pengembangan sistem yang lebih cepat dengan kualitas yang tetap terjaga. Namun, RAD juga memiliki tantangan, seperti mengelola ekspektasi pengguna yang dapat meningkat seiring dengan pemahaman mereka terhadap teknologi..

Menurut (Denis, dkk, 2015), dalam suatu proses pengembangan dengan metode *Rapid Application Development* melibatkan beberapa fase utama yaitu :



Gambar 2. 4 Model Rapid Application Development (Dennis, 2015).

1. *Planning*

Proses dimulai dari tahap perencanaan, di mana kebutuhan proyek dan lingkup pengembangan sistem ditentukan. Tahap ini bertujuan untuk menyusun rencana proyek yang mencakup jadwal, sumber daya, dan ruang lingkup sistem secara keseluruhan, sehingga memberikan fondasi yang jelas untuk tahap berikutnya.

2. *Analysis*

Setelah perencanaan selesai, proyek berlanjut ke tahap analisis. Pada tahap ini, kebutuhan sistem dikumpulkan dan dianalisis secara rinci, mencakup identifikasi kebutuhan fungsional dan non-fungsional. Informasi ini menjadi dasar untuk perancangan sistem di tahap berikutnya.

3. *Parallel Design*

Tahap desain dimulai setelah analisis selesai dan sistem dipecah menjadi beberapa subproyek yang lebih kecil. Setiap subproyek bekerja secara paralel untuk merancang bagian spesifik dari sistem, mempercepat proses pengembangan dan memungkinkan tim untuk fokus pada modul masing-masing.

4. *Parallel Implementation*

Setelah desain selesai, setiap subproyek memasuki fase implementasi secara paralel. Fase ini memungkinkan pengembangan dilakukan secara simultan untuk setiap modul, sehingga mempercepat proses pengembangan sistem secara keseluruhan.

5. *Integration*

Setelah implementasi setiap subproyek selesai, hasil dari subproyek tersebut digabungkan dalam tahap integrasi. Pada tahap ini, semua bagian sistem diuji bersama untuk memastikan mereka berfungsi sebagai satu kesatuan yang utuh dan sesuai dengan kebutuhan awal.

6. *System*

Tahap akhir adalah menghasilkan sistem yang lengkap dan siap digunakan. Sistem yang telah terintegrasi dan diuji ini kemudian diserahkan kepada pengguna untuk digunakan dalam lingkungan operasional yang sesungguhnya.

2.1.6 ***Object Oriented Analysis and Design (OOAD)***

Menurut (Dennis, 2015), prinsip dasar Analisis dan Desain Berorientasi Objek (OOAD) adalah bahwa setiap elemen dianggap sebagai objek yang memiliki data (atribut) dan fungsi (perilaku). Suatu objek hanya dapat mengakses atau mengubah atributnya melalui perilakunya sendiri. Objek-objek ini dapat berkomunikasi satu sama lain untuk bertukar informasi atau melakukan tindakan tertentu. Perubahan pada satu objek tidak akan memengaruhi objek lain, karena atribut dan perilakunya terkapsulasi dalam objek itu dan bersifat independen. Dengan enkapsulasi ini, objek dapat

digunakan kembali di beberapa sistem, yang memungkinkan objek ditambahkan atau dihapus dari aplikasi dengan dampak minimal pada komponen lain.

2.1.7 *Unified Modelling Language (UML)*

Menurut Dennis (2015), UML mempunyai tujuan yang berkaitan dengan penyelesaian integral umum dan teknik diagram untuk memudahkan pemodelan proyek untuk merancang sistem, setelah tahap analisis implementasi, terutama dalam konteks yang berorientasi pada enfoque keberatan. Berbagai diagram UML, seperti diagram penggunaan, kelas, dan keamanan, digunakan untuk menjelaskan berbagai aspek sistem yang sedang dipelajari. UML juga menyediakan bentuk visual yang mendokumentasikan persyaratan sistem, hubungan antara komponen dan aliran pekerjaan yang dilakukan. Ini tidak hanya memfasilitasi identifikasi dan penyelesaian sementara kemungkinan masalah, tetapi Anda juga harus melakukan proses berulang dalam pencarian sistem yang fleksibel dan berkelanjutan.

Tabel 2. 1 Struktur Diagram pada UML

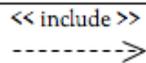
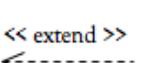
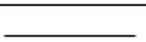
Nama Diagram	Fungsi	Fase Primer
<i>Structure Diagrams</i>		
Class	Mengilustrasikan hubungan antar kelas yang dimodelkan di dalam sistem	Analysis, Design
<i>Behavioral Diagrams</i>		
Activity	Mengilustrasikan alur kerja bisnis yang tidak bergantung pada <i>class</i> , alur aktivitas dalam <i>use case</i> , atau desain metode secara mendetail	Analysis, Design
Sequence	Menggambarkan model perilaku objek dalam <i>use case</i> berfokus pada pengurutan aktivitas berdasarkan waktu	Analysis, Design
Use-Case	Menangkap kebutuhan bisnis untuk sistem dan menggambarkan interaksi antara sistem dengan <u>lingkungannya</u>	Analysis

2.1.7.1 Use Case Diagram

Menurut (Dennis, 2015), Use Case Diagram menggambarkan bagaimana suatu sistem berfungsi atau berperilaku. Diagram ini mencakup serangkaian kemungkinan urutan interaksi antara sistem dan pengguna dalam konteks tertentu, yang terkait dengan tujuan tertentu. Dalam Use Case Diagram, sistem menunjukkan perilakunya dalam berbagai kondisi sambil merespons permintaan dari aktor utama.

Use Case Diagram membantu mengidentifikasi dan menggambarkan fungsi-fungsi utama yang harus disediakan oleh sistem untuk memenuhi kebutuhan pengguna. Aktor dalam diagram ini, baik berupa manusia maupun sistem eksternal lainnya, memiliki peran penting dalam menggambarkan bagaimana mereka berinteraksi dengan sistem untuk mencapai tujuan tertentu.

Tabel 2. 2 Notasi Use Case Diagram

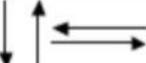
NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

2.1.7.2 Activity Diagram

Menurut Dennis (2015), Activity Diagram menggambarkan logika kondisional untuk urutan aktivitas dalam sistem yang diperlukan untuk mencapai proses bisnis. Aktivitas-aktivitas tersebut bisa bersifat manual atau otomatis, dan setiap kegiatan biasanya menjadi tanggung jawab unit organisasi tertentu.

Activity Diagram membantu menggambarkan alur kerja proses bisnis atau sistem secara rinci, termasuk langkah-langkah, keputusan, dan hasil yang mungkin terjadi. Diagram ini sering digunakan untuk memvisualisasikan proses yang kompleks, memastikan setiap pemangku kepentingan memiliki pemahaman yang sama tentang bagaimana aktivitas berjalan. Dengan menggunakan simbol seperti swimlanes, diagram ini dapat memperjelas peran dan tanggung jawab setiap unit atau individu dalam melaksanakan proses bisnis tertentu, sehingga meningkatkan efisiensi dan kolaborasi.

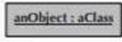
Tabel 2. 3 Notasi Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

2.1.7.3 Sequence Diagram

Menurut (Dennis, 2015), Sequence Diagram menggambarkan objek-objek yang terlibat dalam suatu Use Case dan pesan-pesan yang saling dikirim di antara objek-objek tersebut selama jalannya suatu skenario. Diagram urutan ini adalah model dinamis yang menunjukkan urutan pesan eksplisit yang dikirim antar objek dalam suatu interaksi. Sequence Diagram memberikan gambaran mendetail tentang interaksi antar objek, mulai dari inisiasi tindakan hingga pencapaian tujuan akhir. Setiap pesan dalam diagram ini tidak hanya menunjukkan alur komunikasi, tetapi juga mengungkapkan urutan kronologis serta konteks dari setiap interaksi. Dengan demikian, Sequence Diagram membantu pengembang memahami alur kerja sistem dengan lebih terstruktur, memastikan bahwa semua elemen berkolaborasi dengan tepat untuk mendukung fungsionalitas sistem.

Tabel 2. 4 Notasi Sequence Diagram

Term and Definition	Symbol
An actor: <ul style="list-style-type: none"> Is a person or system that derives benefit from and is external to the system. Participates in a sequence by sending and/or receiving messages. Is placed across the top of the diagram. Is depicted either as a stick figure (default) or, if a nonhuman actor is involved, as a rectangle with <<actor>> in it (alternative). 	 anActor 
An object: <ul style="list-style-type: none"> Participates in a sequence by sending and/or receiving messages. Is placed across the top of the diagram. 	
A lifeline: <ul style="list-style-type: none"> Denotes the life of an object during a sequence. Contains an X at the point at which the class no longer interacts. 	
An execution occurrence: <ul style="list-style-type: none"> Is a long narrow rectangle placed atop a lifeline. Denotes when an object is sending or receiving messages. 	
A message: <ul style="list-style-type: none"> Conveys information from one object to another one. A operation call is labeled with the message being sent and a solid arrow, whereas a return is labeled with the value being returned and shown as a dashed arrow. 	
A guard condition: <ul style="list-style-type: none"> Represents a test that must be met for the message to be sent. 	
For object destruction: <ul style="list-style-type: none"> An X is placed at the end of an object's lifeline to show that it is going out of existence. 	
A frame: <ul style="list-style-type: none"> Indicates the context of the sequence diagram. 	

2.1.7.4 Class Diagram

Menurut (Dennis, 2015) Class Diagram adalah diagram yang menggambarkan struktur statis dari sistem dengan menunjukkan kelas-kelas, atribut, metode, dan hubungan antar kelas dalam sistem tersebut. Diagram ini membantu dalam memodelkan struktur data dan hubungan antar objek dalam sistem, sehingga memudahkan pemahaman dan pengembangan sistem berbasis objek.

Dalam Class Diagram, setiap kelas diwakili sebagai sebuah kotak yang dibagi menjadi tiga bagian: nama *class*, *attribute*, dan *methode*. Hubungan antar kelas, seperti asosiasi, pewarisan, dan agregasi, digambarkan dengan berbagai jenis garis dan simbol, yang menunjukkan bagaimana kelas-kelas tersebut berinteraksi dan berhubungan satu sama lain dalam sistem. Dengan demikian, Class Diagram menjadi alat yang penting dalam analisis dan desain sistem berorientasi objek, karena memberikan gambaran yang jelas tentang struktur dan interaksi komponen-komponen dalam sistem.

Tabel 2. 5 Notasi Class Diagram

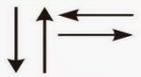
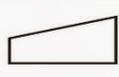
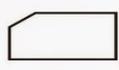
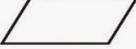
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	<u>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor</u>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	<u>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri</u>
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.1.8 Flowchart

Menurut (Rosaly, 2019), Flowchart, yang juga dikenal sebagai diagram alir, adalah jenis diagram yang menggambarkan algoritma atau urutan langkah-langkah instruksi dalam suatu sistem. Seorang analis sistem menggunakan flowchart sebagai dokumentasi untuk menjelaskan gambaran logis sistem yang akan dikembangkan kepada programmer. Dengan demikian, flowchart membantu memberikan solusi terhadap potensi masalah yang mungkin timbul dalam proses pembangunan sistem. Secara umum, flowchart digambarkan dengan simbol-simbol yang masing-masing mewakili suatu proses tertentu, sementara garis penghubung digunakan untuk menghubungkan satu proses dengan proses berikutnya.

Dengan menggunakan flowchart, urutan setiap proses dapat digambarkan dengan lebih jelas. Selain itu, penambahan proses baru dapat dilakukan dengan mudah dengan menggunakan flowchart ini.

Tabel 2. 6 Notasi Flowchart Diagram

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2.1.9 Basis Data

Menurut (Rahimi Fitri, 2020), basis data (atau database dalam bahasa Inggris) adalah kumpulan data yang terorganisir, yang biasanya disimpan dan diakses secara elektronik melalui sistem komputer. Seiring dengan semakin kompleksnya basis data, maka teknik perancangan dan pemodelan formal dikembangkan untuk mengelolanya. Perangkat lunak yang digunakan untuk mengelola basis data disebut sistem manajemen basis data (database management system), yang disingkat DBMS.

Dalam konteks ini, Sistem Manajemen Basis Data (DBMS) berperan sebagai perangkat lunak yang mengelola dan memfasilitasi interaksi antara pengguna dan basis data. DBMS menyediakan antarmuka untuk melakukan operasi seperti penyimpanan, pembaruan, pengambilan, dan penghapusan data, serta memastikan keamanan dan integritas data. Dengan menggunakan DBMS, organisasi dapat mengelola data secara terpusat, mengurangi redundansi, dan meningkatkan efisiensi operasional.

Selain itu, basis data dapat diklasifikasikan berdasarkan model data yang digunakan, seperti model relasional, hierarkis, atau jaringan. Model relasional, yang paling umum digunakan saat ini, mengorganisir data dalam bentuk tabel-tabel yang saling berhubungan. Setiap tabel terdiri dari baris dan kolom, di mana baris mewakili entitas data dan kolom mewakili atribut dari entitas tersebut. Hubungan antar tabel ditentukan melalui kunci primer dan kunci asing, yang memungkinkan integrasi data yang kompleks dan fleksibel.

Dengan demikian, basis data dan DBMS memainkan peran krusial dalam pengelolaan informasi modern, memungkinkan organisasi untuk menyimpan, mengakses, dan mengelola data secara efisien dan efektif.

2.2 Peneliti Terdahulu

Sebagai penunjang penelitian riset, peneliti melakukan tinjauan literatur yang berkaitan dengan topik penelitian yang dibahas. Berikut beberapa referensi literatur yang dikaji oleh peneliti, diantaranya :

RANCANG BANGUN APLIKASI DONASI BERBASIS WEB DI PANTI ASUHAN AL-IHSAN BANJARMASIN

Penelitian ini yang ditulis oleh Syahrianor dan M. Rizki Zulkarnain, serta diterbitkan di Jurnal Pendidikan Teknologi Informasi Volume 1 Nomor 1 (2022), menggambarkan perancangan dan pembangunan aplikasi donasi berbasis web di Panti Asuhan Al-Ihsan Banjarmasin. Dalam penelitian ini, para peneliti menggunakan pendekatan pengembangan sistem dengan metode Rapid Application Development (RAD).

Proses penelitian dimulai dengan analisis sistem yang sudah ada sebelumnya, untuk kemudian dilanjutkan dengan analisis kebutuhan sistem baru. Setelah itu, dilakukan penyusunan kerangka sistem sekaligus penyusunan beberapa diagram UML yang diperlukan untuk membangun sistem tersebut, termasuk use case diagram dan class diagram.

Pendekatan pengembangan sistem dengan metode RAD memungkinkan para peneliti untuk menghasilkan prototipe aplikasi donasi secara cepat dan efisien. Dengan analisis yang teliti terhadap sistem yang sudah ada dan kebutuhan yang diperlukan, serta dengan bantuan kerangka sistem dan diagram UML, pengembangan aplikasi dapat dilakukan dengan lebih terstruktur dan efektif. Selain itu, penggunaan metode RAD juga memungkinkan untuk adanya iterasi dan perbaikan yang cepat berdasarkan umpan balik dari pengguna selama proses pengembangan aplikasi berlangsung.

RANCANG BANGUN SISTEM INFORMASI RUMAH SEDEKAH SOLO

Penelitian yang dilakukan oleh Priyanto, yang diterbitkan di repository Universitas Muhammadiyah Surakarta pada tahun 2018, mengambil studi di Rumah Sedekah di Solo. Fokus penelitian ini adalah perancangan aplikasi website menggunakan metode Rapid Application Development (RAD). Tujuan utama dari penelitian ini adalah meningkatkan efektivitas dan efisiensi dalam penyebaran informasi dan pengelolaan dana sedekah di Rumah Sedekah tersebut.

Dengan menerapkan sistem informasi yang terintegrasi dan transparan, penelitian ini bertujuan untuk mempermudah proses pengelolaan dana sedekah, menarik minat donatur, serta meningkatkan produktivitas dan kualitas mutu dalam penyampaian informasi di Rumah Sedekah Solo. Dalam pengembangannya, peneliti melakukan pengujian sebelum program diaplikasikan secara luas pada organisasi untuk memastikan tidak adanya kesalahan yang signifikan. Pengujian juga melibatkan pengguna secara langsung untuk mendapatkan tanggapan dan umpan balik terkait dengan sistem yang telah dibuat.

Harapannya, sistem informasi yang dikembangkan dapat memberikan kemudahan dalam pekerjaan petugas Rumah Sedekah Solo, meningkatkan produktivitas, serta menarik minat donatur dengan menyajikan laporan yang informatif dan terpercaya. Dengan demikian, kontribusi sedekah dapat dikelola dengan lebih baik dan transparan, memperkuat kepercayaan masyarakat serta meningkatkan kualitas pelayanan yang diberikan oleh Rumah Sedekah tersebut.

PENERAPAN MODEL RAPID APPLICATION DEVELOPMENT (RAD) DALAM RANCANG BANGUN SISTEM INFORMASI WARGA

Penelitian yang ditulis oleh Suryanto dan diterbitkan dalam jurnal Infotek: Jurnal Informatika dan Teknologi pada tahun 2022, membahas penerapan Model Rapid Application Development (RAD) dalam rancang bangun Sistem Informasi Warga. Tujuan utama dari pembuatan aplikasi website ini adalah untuk memudahkan warga dalam pengajuan surat pengantar, memberikan informasi keuangan kas RT, serta mempermudah ketua RT dalam pembuatan laporan.

Metode pengembangan yang digunakan adalah RAD, dengan memanfaatkan framework Codeigniter 3 dan Bootstrap. Penelitian ini melibatkan tinjauan pustaka, landasan teori, dan pengertian sistem informasi serta rancang bangun aplikasi tersebut.

Sistem aplikasi yang berhasil dibangun telah menjalani serangkaian pengujian untuk memastikan keamanan dan keandalan fitur-fiturnya, termasuk fitur notifikasi kesalahan, keamanan sistem, dan implementasi pengujian Black Box Testing. Dengan demikian, penelitian ini memberikan kontribusi yang signifikan dalam pengembangan sistem informasi berbasis web yang efektif dan efisien untuk memenuhi kebutuhan masyarakat.

Melalui penerapan RAD dan penggunaan teknologi seperti framework Codeigniter 3 dan Bootstrap, sistem informasi yang dikembangkan dapat memberikan manfaat yang besar bagi warga dan ketua RT dalam mengelola informasi dan proses administratif yang berkaitan dengan kehidupan sehari-hari di lingkungan mereka. Diharapkan, implementasi aplikasi ini dapat meningkatkan efisiensi, transparansi, dan partisipasi masyarakat dalam proses administrasi dan pelaporan di tingkat RT.

RANCANG BANGUN SEDEKAH ONLINE PADA PANTI ASUHAN AL MARHAMAH MEDAN BERBASIS WEB

Jurnal yang ditulis oleh Ningsih dan diterbitkan di repository Panca Budi pada tahun 2020 membahas pembuatan sistem informasi sedekah online untuk Panti Asuhan Al-Marhamah di Medan. Penelitian ini merupakan respons terhadap pandemi COVID-19 yang membatasi interaksi fisik dan menghambat proses penggalangan dana secara konvensional. Sistem informasi ini dirancang untuk memudahkan donatur dalam memberikan sedekah secara online, sehingga memfasilitasi kontinuitas dukungan kepada panti asuhan tersebut.

Dengan memanfaatkan bahasa pemrograman PHP dan basis data MySQL, sistem ini diharapkan dapat memberikan kemudahan bagi panti asuhan dalam mengelola donasi dan memperluas jangkauan kepada donatur potensial. Selain fungsi utama sebagai platform untuk menerima donasi secara online, sistem ini juga bertujuan untuk mempermudah penyimpanan data donasi dan mengajarkan kemandirian kepada anak-anak asuh melalui kewirausahaan.

Dokumen penelitian ini juga mencakup penjelasan tentang teori-teori pendukung, metodologi penelitian yang digunakan, dan rancangan sistem yang diusulkan. Dengan demikian, penelitian ini tidak hanya memberikan solusi praktis dalam mengatasi tantangan yang dihadapi oleh panti asuhan selama pandemi, tetapi juga mengintegrasikan nilai-nilai pendidikan dan kemandirian kepada anak-anak asuh melalui implementasi sistem informasi sedekah online.

RANCANG BANGUN SISTEM PENDATAAN JUAL BELI TANAH MENGUNAKAN METODE RAPID APPLICATION DEVELOPMENT

Jurnal yang ditulis oleh Alam, dkk. dan diterbitkan dalam Jurnal Testing dan Implementasi Sistem Informasi Volume 1 Nomor 1 tahun 2023, mengangkat topik tentang Sistem Pengumpulan Data Jual Beli Tanah berbasis web yang dirancang menggunakan metode Rapid Application Development (RAD). Tujuan utama dari pengembangan sistem ini adalah untuk memudahkan dan mengoptimalkan kinerja pegawai di Kecamatan Binawidya dalam mengelola data transaksi jual beli tanah.

Sistem ini tidak hanya bertujuan untuk membantu institusi dalam mengelola dan menyimpan data jual beli tanah secara aman dalam basis data komputer untuk jangka waktu yang lama, tetapi juga untuk meningkatkan kinerja di kantor Kecamatan Binawidya secara keseluruhan. Melalui penerapan teknologi informasi, artikel ini menekankan pentingnya efisiensi dan keamanan dalam pengelolaan data transaksi tanah.

Dengan menerapkan metode RAD, pengembangan sistem dapat dilakukan dengan cepat dan responsif terhadap kebutuhan pengguna. Hal ini membantu mempercepat proses pengumpulan data jual beli tanah dan meningkatkan efisiensi kerja pegawai. Selain itu, penggunaan sistem berbasis web juga memungkinkan akses data yang lebih mudah dan cepat, baik dari internal maupun eksternal institusi.

Artikel ini memberikan kontribusi penting dalam bidang pengembangan sistem informasi di sektor pemerintahan, khususnya dalam hal pengelolaan data transaksi tanah. Diharapkan, implementasi sistem ini dapat memberikan manfaat yang signifikan dalam meningkatkan efisiensi, akurasi, dan keamanan dalam pengelolaan data transaksi tanah di Kecamatan Binawidya maupun di lingkungan sekitarnya.