

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Kajian Pustaka

Kajian pustaka merupakan langkah kritis dalam penelitian yang melibatkan analisis terhadap penelitian terdahulu yang relevan dengan topik penelitian. Kajian pustaka memberikan landasan teoretis dan kontekstual untuk memahami ruang lingkup penelitian serta mendukung identifikasi pengetahuan pada penelitian yang sedang dilakukan [12]. Jurnal atau riset terdahulu pada penelitian ini dirangkum dalam tabel tinjauan pustaka dengan cara 3C2S, yang terdiri dari Judul, *Comparing*, *Contrasting*, *Criticize*, *Synthesize*, dan *Summarize*. *Comparing* berisikan persamaan penelitian, *Contrasting* berisikan perbedaan penelitian, *Criticize* berisikan kritik, *Synthesize* berisikan tujuan, dan *Summarize* berisikan ringkasan atau kesimpulan dari penelitian terdahulu.

Penelitian dengan judul "Sistem Informasi Penjualan Paket Wisata Berbasis Web Menggunakan Metode *Rapid Application Development* Pada PO. Amura Tours" yang dilakukan oleh "Fransisca Monica Sari, Daniel Alfa Puryono, Moh Rofi" merupakan jurnal acuan pada penelitian ini, yang terbit pada tahun 2022 [13].

Kajian sebelumnya telah diperiksa untuk mempelajari riset yang telah dilakukan sebelumnya dengan tujuan membandingkannya dengan riset yang akan datang. Kajian sebelumnya juga penting untuk mengidentifikasi perbedaan dalam isi keseluruhan karya ilmiah. Terdapat beberapa studi sebelumnya yang terkait dengan riset yang sedang dilakukan ini, ditunjukkan dalam tabel 2.1 berikut:

Tabel 2.1 Kajian Pustaka

No	Judul	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesis</i>	<i>Summarize</i>
1	Rancang Bangun Aplikasi Pariwisata di Kabupaten Parigi Moutong Berbasis <i>Android</i> [14].	Menggunakan metode <i>Rapid Application Development</i> (RAD).	Sistem diuji menggunakan <i>Blackbox testing</i> .	Pada penelitian ini dapat ditambahkan sistem pemesanan paket wisata secara <i>online</i> .	Penelitian ini terdapat pemesanan wisata yang dapat dilakukan secara <i>online</i> .	Penelitian ini menghasilkan aplikasi pariwisata yang dapat menampilkan informasi tentang lokasi wisata, kegiatan pariwisata, dan video pariwisata.
2	Rancang Bangun <i>Website</i> Sebagai Sarana Promosi Wisata Pulau Lembeh [15].	Sistem dikembangkan dengan menggunakan metode <i>Rapid Application Development</i> (RAD) dan diuji menggunakan metode	Pengembangan <i>website</i> menggunakan <i>CMS (Content Managenet System)</i> .	Penelitian ini dapat ditambahkan sistem pemesanan paket wisata berbasis <i>online</i> .	Penelitian ini mencakup kemampuan untuk melakukan pemesanan wisata secara <i>online</i> .	Sistem informasi yang dibuat untuk wisata Pulau Lembeh berfungsi sebagai sarana promosi dan pemasaran.

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
		<i>Blackbox testing.</i>				
3	Sistem Informasi Penjualan Paket Wisata Berbasis Web Menggunakan Metode <i>Rapid Application Development</i> Pada PO. Amura Tours [13].	Metode pengembangan <i>Rapid Application Development</i> (RAD) dan sistem diuji menggunakan pendekatan <i>Blackbox testing.</i>	Pemesanan paket wisata secara <i>online</i> yang dilakukan oleh PO. Amura Tours.	<i>Payment gateway</i> dapat ditambahkan untuk memfasilitasi proses pembayaran secara <i>online.</i>	Menambahkan <i>payment gateway</i> agar memudahkan <i>customer</i> dalam melakukan pembayaran.	Sistem informasi penjualan paket yang telah dibuat ini dapat membantu dalam hal menghemat biaya cetak tiket, brosur, atau katalog wisata.
4	Rancang Bangun Sistem Informasi <i>Booking Order Online Barbershop</i> Berbasis Website [16].	Pengembangan sistem menggunakan <i>Rapid Application Development</i> (RAD) sebagai pendekatan utama.	Sistem diuji dilakukan dengan metode UAT (<i>User Acceptance Testing</i>).	Menambahkan <i>payment gateway</i> untuk memfasilitasi proses pembayaran secara <i>online.</i>	Mengintegrasikan <i>payment gateway</i> untuk mempermudah <i>customer</i> dalam proses pembayaran.	Pengembangan sistem informasi ini adalah untuk memberikan kenyamanan kepada pelanggan dalam mengakses informasi mengenai

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
						layanan dan fasilitas yang disediakan di <i>barbershop</i> , serta melakukan pemesanan secara <i>online</i> tanpa harus datang ke tempat dan menghemat waktu. Sistem ini dikembangkan menggunakan <i>framework</i> <i>Laravel</i> .
5	Rancang Bangun Aplikasi Pemesanan <i>Online</i> Tiket Bioskop Berbasis <i>Mobile</i> [17].	Metode pengembangan sistem <i>Rapid Application Development</i> (RAD).	Rancangan bangun menggunakan dua media multiplatform (<i>Android</i> dan <i>Website</i>).	Melakukan penambahan <i>payment gateway</i> untuk mempermudah proses pembayaran.	Penelitian ini melakukan penambahan <i>payment gateway</i> untuk memberikan kemudahan kepada <i>customer</i> dalam	Penelitian ini merancang aplikasi pemesanan <i>online</i> tiket bioskop yang dapat memudahkan <i>customer</i> dapat

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
					melakukan pembayaran.	booking tiket tanpa datang langsung ke tempat bioskop.
6	Impelementasi Metode <i>Rapid Application Development</i> Pada Rancang Bangun Sistem Informasi Berbasis <i>Web</i> dengan <i>Framework W3.CSS</i> (Studi Kasus : POUK Lenteng Agung) [18].	Metode pengembangan pada penelitian ini menggunakan <i>Rapid Application Development</i> (RAD) sebagai pendekatan utama dan <i>Blackbox testing</i> digunakan untuk menguji sistem yang telah dikembangkan.	Rancangan bangun menggunakan <i>framework W3.CSS</i> .	Adanya pelatihan kepada pengguna sistem ini agar membantu mengurangi kesalahan saat menggunakan sistem.	Penelitian ini mencakup pelatihan dan panduan penggunaan sistem untuk mengurangi kesalahan untuk mengurangi adanya kesalahan penggunaan sistem.	Sistem informasi berbasis <i>web</i> ini dirancang untuk membantu POUK Lenteng Agung dalam engoptimalkan mefektivitas dan efisiensi dalam mengelola data-data anggota serta transaksi pembayaran.
7	Rancang Bangun Sistem Pemesanan <i>Meeting Room</i> Berbasis <i>Web</i>	Metode <i>Rapid Application Development</i> (RAD) diadopsi	UAT (<i>User Acceptance Test</i>) digunakan untuk menguji sistem	Tampilan <i>website</i> kurang menarik mengakibatkan	Penelitian ini berfokus pada upaya meningkatkan	Rancang bangun sistem ini digunakan untuk

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
	Menggunakan <i>Framework</i> Laravel [19].	sebagai pendekatan utama, dan <i>framework</i> Laravel digunakan sebagai alat bantu dalam pengembangan sistem.	yang dikembangkan.	penurunan minat pengguna.	tampilan agar lebih menarik dan menarik minat pengguna.	mempermudah dalam memesan ruang diskusi dengan metode RAD untuk pengembangannya.
8	<i>Design of a Rapid Development Model for Mobile Internet</i> [20].	Metode <i>Rapid Application Development</i> (RAD) diadopsi sebagai pendekatan utama	Metode RAD digunakan untuk pengembangan Web dan antarmuka API.	Penelitian ini tidak ada informasi tentang pengujian atau validasi sehingga hasil dari penelitian tidak konsisten dan belum bisa diandalkan.	Fokus penelitian lebih kepada pembuatan model mobile internet sebagai rekomendasi untuk memodifikasi logika bisnis sesuai kebutuhan.	Pengembangan aplikasi ini tidak diperlukan penulisan kode secara manual untuk membangun berbagai sistem manajemen sehingga mempersingkat prosedur pengembangan. Model ini juga mendukung

No	Judul	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesis</i>	<i>Summarize</i>
						pengembangan aplikasi yang bersifat dinamis dan adaptif yang terdiri dari beberapa subsistem atau modul.
9	<i>Design and Development Meeting Schedule Managemen Application using the RAD Method [21].</i>	Metode yang digunakan dalam penelitian ini <i>Rapid Application Development (RAD)</i>	Metode <i>Rapid Application Development (RAD)</i> pada penelitian ini digunakan untuk pengembangan aplikasi manajemen yang berguna untuk membantu penjadwalan pertemuan dan meminimalkan risiko kehilangan dokumen.	Penelitian ini belum ada penilaian yang mendalam mengenai dampak dan efektivitasnya untuk mengukur keberhasilan sistem terhadap kinerja dan efisiensi instansi pemerintah.	Fokus penelitian lebih kepada pengembangan sistem penjadwalan pertemuan yang efisien dan aman untuk instansi pemerintah.	Penggunaan sistem penjadwalan pertemuan elektronik dapat mengatasi permasalahan pencatatan jadwal rapat yang masih menggunakan buku dan rawan kerugian. Sistem ini membantu dalam

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
						penjadwalan pertemuan yang efisien.
10	<i>Islamic Calendar: Prototype of Hijri Calendar Application using Rapid Application Development Method</i> [22].	Mengadopsi metode <i>Rapid Application Development</i> (RAD).	Metode <i>Rapid Application Development</i> (RAD) pada penelitian ini digunakan untuk membuat aplikasi kalender Hijriah yang dapat memberikan informasi tentang hari besar Islam dengan akurasi tinggi.	Penelitian ini, tidak menjelaskan keterlibatan pengguna aplikasi dalam proses pengujian.	Fokus penelitian lebih kepada pengembangan sebuah <i>prototype</i> aplikasi kalender Hijriah.	Penelitian ini memberikan kontribusi dalam pengembangan aplikasi kalender Hijriah yang akurat dan efisien.

Hasil analisis pada Tabel 2.1, dapat disimpulkan bahwa terdapat kesamaan dan perbedaan antara penelitian ini dengan penelitian sebelumnya. Secara khusus, kedua penelitian memiliki fokus yang serupa, yakni pengembangan sistem informasi berbasis *website* dan memiliki tujuan yang sama, yaitu meningkatkan ketersediaan serta aksesibilitas informasi. Meskipun demikian, perbedaan muncul dalam objek dan subjek penelitian, di mana penelitian ini difokuskan pada *Basecamp* Gunung Slamet jalur Bambangan dengan subjek penelitian yang terutama mencakup calon pendaki, khususnya yang berminat mendaki Gunung Slamet jalur Bambangan.

2.2 Landasan Teori

2.2.1 Rancang Bangun

Rancang bangun merupakan sebuah tahapan yang akan dilakukan setelah menganalisis dalam pengembangan sebuah sistem. Pada tahap rancang bangun, dilakukan penentuan kebutuhan fungsional serta perencanaan untuk membentuk suatu sistem, termasuk penggambaran, perencanaan, dan pengaturan elemen-elemen yang berbeda menjadi kesatuan yang utuh dan fungsional. Selain itu, dalam tahap ini juga mencakup pengaturan komponen perangkat lunak serta perangkat keras pada sistem tersebut [23].

Rancang bangun adalah proses yang melibatkan pembangunan sistem, baik membuat sistem yang baru inovatif atau mengganti dan meningkatkan sistem yang sudah ada sebelumnya. Proses ini dapat mencakup perubahan secara keseluruhan maupun hanya pada bagian-bagian tertentu. Pembangunan sistem dapat diartikan sebagai sebuah kegiatan yang bertujuan untuk menciptakan sistem baru yang revolusioner atau melakukan perubahan serta perbaikan pada sistem yang sudah ada secara menyeluruh [24].

Berdasarkan pengertian di atas, dapat ditarik kesimpulan bahwa rancang bangun adalah sebuah tahapan penting dalam pengembangan sistem setelah melakukan analisis. Rancang bangun ini, dapat dilakukan penentuan kebutuhan perencanaan untuk membentuk suatu sistem yang utuh dan fungsional. Proses ini melibatkan penggambaran, perencanaan, dan konfigurasi komponen *hardware* dan *software*. Rancang bangun juga mencakup pembangunan suatu sistem yang baru atau perbaikan pada sistem yang sudah ada sebelumnya, baik secara keseluruhan maupun pada bagian tertentu. Fokus utamanya adalah mengembangkan sistem yang inovatif dan efektif sesuai dengan kebutuhan yang ada.

2.2.2 Sistem Informasi

Sistem informasi merupakan suatu sistem menggabungkan kebutuhan transaksi harian guna mendukung pelaksanaan fungsi manajemen dan aktivitas strategis. Tujuan utamanya adalah memberikan laporan sesuai dengan permintaan oleh pihak eksternal [25]. Sistem informasi adalah bagian integral dari serangkaian

proses teratur yang terkoordinasi dan saling mendukung yang meliputi *software*, *hardware*, dan pelaksana. Tujuannya adalah untuk mengubah data mentah menjadi sebuah informasi yang memiliki nilai dan manfaat bagi sebuah lembaga atau organisasi. Secara umum, sistem informasi dianggap sebagai sebuah perangkat lunak yang dapat membantu dalam sebuah pengelolaan dan analisis data. Informasi dari pemrosesan data tersebut, pengambilan keputusan dalam suatu organisasi dapat dilakukan dengan lebih efektif dan berdasarkan fakta [26].

Berdasarkan pengertian di atas, dapat ditarik kesimpulan bahwa sistem informasi adalah fondasi yang krusial pada organisasi, dengan mengintegrasikan pemrosesan transaksi yang terjadi setiap hari dan mendukung fungsi manajemen serta kegiatan strategis. Tujuannya adalah untuk menyediakan pelaporan yang dibutuhkan oleh pihak lembaga atau organisasi. Sistem meliputi *software*, *hardware*, dan pelaksana yang saling bekerja sama dalam mengolah data mentah menjadi informasi yang berharga. Adanya bantuan sistem informasi, pengambilan keputusan dalam organisasi dapat dilakukan secara efektif dan berdasarkan fakta, memastikan kelancaran operasi bisnis dan meningkatkan produktivitas.

2.2.3 Website

Website merupakan serangkaian halaman yang menyajikan informasi yang dapat dijangkau oleh pengguna di berbagai belahan dunia melalui koneksi internet. Komponen utama dari sebuah *website* meliputi Gambar, animasi, teks, dan suara yang dirancang untuk menarik perhatian pengunjung [27]. *Website* terdiri dari sejumlah halaman web yang saling terhubung membentuk sebuah kesatuan dan semua elemen di dalamnya saling berkaitan. Web ini terdiri dari halaman individual, sekumpulan halaman juga dikenal sebagai halaman *homepage* [28].

Berdasarkan pengertian di atas, dapat ditarik kesimpulan bahwa *website* merupakan sekumpulan halaman yang berisi informasi yang dapat dijangkau oleh pengguna dari seluruh dunia melalui internet.

2.2.4 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan suatu metode yang dapat digunakan dalam merancang sistem berbasis pemrograman berorientasi objek atau biasa disebut dengan *Object Oriented Programming* (OOP) [29]. *Unified Modeling*

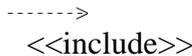
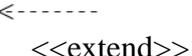
Language (UML) merupakan sebuah metode yang digunakan untuk memodelkan dan mengkomunikasikan sistem dengan menggunakan bahasa visual. *Unified Modeling Language* (UML) memungkinkan pemodelan terdiri dari berbagai jenis, meliputi *use case diagram*, *class diagram*, *activity diagram*, serta *Sequence Diagram*, melalui penggunaan diagram dan teks pendukung [30].

Unified Modeling Language (UML) meliputi beberapa jenis diagram yang sering digunakan, di antaranya adalah sebagai berikut

1) *Use Case Diagram*

Use case diagram adalah model yang dapat digunakan untuk mendeskripsikan tingkah laku atau perilaku dari suatu sistem informasi yang sedang direncanakan. Penjelasan mengenai simbol-simbol pada *use case diagram* tercantum di tabel 2.2 berikut ini.

Tabel 2.2 *Use Case Diagram* [4]

No	Simbol	Keterangan
1	<p><i>Use case</i></p> 	Simbol yang menggambarkan interaksi-interaksi antar aktor (pengguna) dan tujuannya untuk menjelaskan fungsionalitas dan perilaku sistem yang akan dikembangkan.
2	<p><i>Actor</i></p> 	Menggambarkan peran khusus yang dimainkan oleh pengguna saat terlibat dalam interaksi dengan <i>use case</i> .
3	<p><i>Dependency</i></p> 	Hubungan ini mengacu pada suatu perubahan dimana suatu unsur yang mandiri akan berpengaruh terhadap unsur yang tidak memiliki otonomi.
4	<p><i>Include</i></p> 	Simbol ini mewakili hubungan di mana <i>use case</i> tambahan terkandung di dalam <i>use case</i> lain yang mana <i>use case</i> tersebut ditambahkan dengan <i>use case</i> utama yang digunakan untuk melakukan beberapa fungsionalitas.
5	<p><i>Generalization</i></p> 	Menggambarkan objek anak mengadopsi perilaku dan struktur data yang diwarisi dari objek induk yang berada di atasnya.
6	<p><i>Extend</i></p> 	Menggambarkan <i>use case</i> target melibatkan perluasan fungsionalitas dari <i>use case</i> sumber pada titik yang telah ditentukan sebelumnya.

No	Simbol	Keterangan
7	<p><i>Collaboration</i></p> 	Menggambarkan interaksi antara aturan-aturan dan elemen-elemen lain dalam sebuah <i>use case</i> yang saling bekerja bersama untuk menciptakan tindakan yang lebih besar daripada jumlah dan kontribusi individu yang disebut sinergi
8	<p><i>Note</i></p> 	Mengacu entitas fisik yang ada saat aplikasi sedang berjalan, penggunaan tersebut mewakili sumber daya komputasi.
9	<p><i>System</i></p> 	Menggambarkan sebuah paket yang memberikan Gambaran sistem dengan cakupan yang terbatas.
10	<p><i>Association</i></p> 	Menggambarkan koneksi antara satu objek dengan objek lainnya.

2) *Class Diagram*

Class diagram adalah representasi visual yang digunakan untuk menggambarkan struktur sistem dengan cara mendefinisikan dan mengidentifikasi kelas-kelas yang selanjutnya digunakan dalam proses pembangunan sebuah sistem. Berikut ini merupakan simbol dari *class diagram* tercantum di tabel 2.3.

Tabel 2.3 *Class Diagram* [4]

No	Simbol	Keterangan
1	<p><i>Class</i></p> 	Mengacu pada sekumpulan objek dengan karakteristik dan fungsi yang sama.
2	<p><i>Collaboration</i></p> 	Menggambarkan rangkaian tindakan yang dilaksanakan oleh sistem sehingga hasil dapat diukur oleh aktor yang berkaitan.
3	<p><i>Realization</i></p> 	Menggambarkan tindakan atau operasi yang dilakukan oleh objek.
4	<p><i>Nary Association</i></p> 	Merujuk pada usaha untuk menghindari hubungan asosiasi yang melibatkan lebih dari dua objek.
5	<p><i>Association</i></p> 	Mengilustrasikan keterkaitan antara satu objek dengan objek-objek yang lainnya.

No	Simbol	Keterangan
6	<i>Dependency</i> ----->	Hubungan ini mengacu pada suatu perubahan dimana suatu unsur yang mandiri akan berpengaruh terhadap unsur yang tidak memiliki otonomi.
7	<i>Generalization</i> ———	Menggambarkan objek anak mengadopsi perilaku dan struktur data yang diwarisi dari objek induk yang berada di atasnya.

3) *Activity Diagram*

Activity diagram merupakan representasi visual yang menggambarkan urutan kegiatan atau proses kerja di dalam sistem, proses bisnis, maupun menu yang ada didalam perangkat lunak. Penjelasan mengenai simbol-simbol pada *activity diagram* diagram tercantum di tabel 2.4 berikut ini.

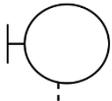
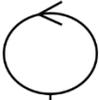
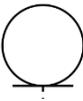
Tabel 2.4 *Activity Diagram* [4]

No	Simbol	Keterangan
1	<i>Initial Node</i> ●	Bagaimana objek suatu aktivitas pada sistem diawali.
2	<i>Activity Final Node</i> ●	Menggambarkan suatu aktivitas pada sistem dibentuk atau diakhiri.
3	<i>Activity</i> 	Menggambarkan interaksi antarmuka kelas yang saling terhubung.
4	<i>Action</i> 	Menunjukkan kondisi sistem yang mencerminkan implementasi suatu tindakan.
5	<i>Fork Node</i> ■	Perubahan suatu aliran menjadi beberapa aliran pada tahapan tertentu.
6	<i>Decision</i> ◇	Merupakan suatu percabangan asosiasi, di mana terdapat penggabungan lebih dari satu aktivitas menjadi satu kesatu.

4) *Sequence Diagram*

Sequence diagram adalah salah satu tipe diagram pada *Unified Modeling Language* (UML) yang menunjukkan bagaimana interaksi yang terjadi antar objek di dalam sistem, termasuk antarmuka, pengguna, dan lainnya, melalui pertukaran pesan. Penjelasan mengenai simbol-simbol pada *Sequence Diagram* tercantum di tabel 2.5 berikut ini.

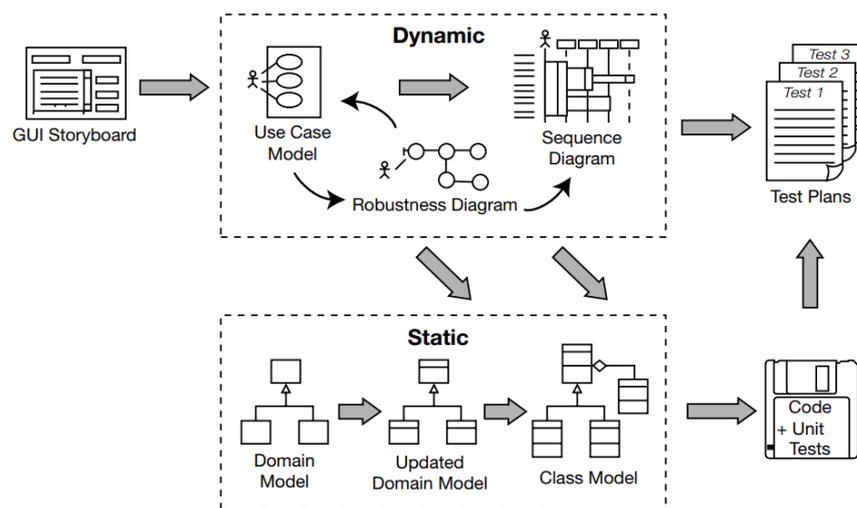
Tabel 2.5 *Sequence Diagram* [4]

No	Simbol	Keterangan
1	<i>LifeLine</i> 	Menggambarkan suatu entitas objek saling berhubungan satu sama lain.
2	<i>Message</i> 	Menjelaskan interaksi antar objek meliputi informasi mengenai kegiatan yang sedang terjadi.
3	<i>Message</i> 	Menjelaskan rincian mengenai komunikasi antar objek-objek, termasuk informasi terkait kegiatan yang terjadi.
4.		<i>Boundary Class</i> merupakan kumpulan kelas yang berfungsi sebagai antarmuka atau media interaksi antara satu atau lebih pemain dengan sistem. Contohnya mencakup tampilan formulir <i>input</i> dan formulir cetak.
5.		<i>Control Class</i> adalah suatu objek yang memuat logika aplikasi tanpa memiliki kewajiban kepada entitas. Sebagai contoh, termasuk perhitungan dan aturan bisnis yang melibatkan berbagai objek.
6.		<i>Entity Class</i> merupakan bagian integral dari sistem yang mengandung sekelompok kelas yang mewakili entitas-entitas yang membentuk representasi awal sistem. Kelas-kelas ini menjadi dasar untuk menyusun struktur basis data.

2.2.5 ICONIX Process

Berdasarkan buku "*Use Case Driven Object Modeling with UML: Theory and Practice*" disajikan suatu pendekatan analisis dan perancangan yang sangat bermanfaat dalam proses pembuatan kode program. Buku ini mengadopsi metodologi USDP (*Unified Software Development Process*) dan ditulis oleh

seorang analis yang memiliki pengalaman sebagai programmer [31]. Penggunaan metode yang tidak tepat dapat membuat peran analis terlihat kurang bermanfaat di mata pengembang perangkat lunak. Metode yang tidak tepat juga dapat mendorong tim untuk lebih memilih membuat kode program terlebih dahulu, kemudian melakukan reverse engineering untuk menghasilkan diagram *Unified Modeling Language* (UML) [31]. Terdapat kecenderungan untuk membuat sistem tanpa melibatkan analisis dan perancangan, sehingga diagram *Unified Modeling Language* (UML) hanya dianggap sebagai pelengkap penelitian tanpa memberikan fungsi yang signifikan. *ICONIX Process* dapat dilihat pada Gambar 2.1 [31].



Gambar 2.1 *ICONIX Process* [31]

Proses analisis dan perancangan sistem berdasarkan buku *Use Case Driven Object Modelling with UML: Theory and Practice*, adalah sebagai berikut [31]:

1. *Requirement Analysis*

- a. *Functional Requirements*

Tahapan ini dilakukan pengumpulan semua kebutuhan fungsional yang diperlukan dalam pengembangan perangkat lunak. Kebutuhan fungsional ini dianggap sebagai fondasi utama dalam proses pengembangan, dimana semua kebutuhan terkait perangkat lunak dikumpulkan menjadi satu kesatuan dan dilakukan analisis terhadap kebutuhan fungsional dan non-fungsional [31].

b. *Domain modelling*

Proses ini melibatkan ekstraksi kata benda dari kebutuhan fungsional, yang kemudian dihubungkan satu sama lain sesuai dengan kebutuhan perangkat lunak sebagai dasar untuk bagian statis dalam *Unified Modeling Language* (UML) [31].

c. *GUI Storyboard*

Storyboard GUI disusun sesuai dengan kebutuhan fungsional yang sudah diuraikan sebelumnya. Rancangan *storyboard* GUI dibentuk berdasarkan *functional requirement* [31].

d. *Use Case Modelling*

Use case memetakan kebutuhan perilaku berdasarkan kebutuhan fungsional. Kalimat dalam *use case* dirancang dalam bentuk aktif, yang berbeda dari ciri kalimat pasif pada *functional requirement*. Setiap *use case* harus mencakup nama yang sesuai dengan domain model. Selain menyajikan skenario ideal (*sunny-day scenario*), *use case* juga harus mencakup skenario alternatif atau *rainy-day scenario* yaitu skenario yang terjadi apabila terjadi kesalahan atau masalah [31].

2. *Analysis and Preliminary Design*

a. *Robustness Analysis*

Robustness merupakan metode untuk menganalisis teks *use case* dan memperbaiki teks tersebut serta model domain. *Robustness Analysis* digunakan untuk menjembatani kesenjangan antara analisis dan desain dengan memahami teks *use case* dan mengidentifikasi asumsi awal terkait objek pada setiap *use case* melalui pembuatan *Robustness Diagram*. Diagram ini untuk memastikan bahwa *use case* ditulis dalam konteks model domain, di mana semua syarat (kata benda dan frasa kata benda) dalam model domain juga diterapkan langsung dalam deskripsi *use case*. Analisis ini secara visual merepresentasikan perilaku *use case* dan menunjukkan kelas yang terlibat [31].

b. *Update Domain Model*

Domain model yang sudah disusun mengalami modifikasi berdasarkan hasil *Robustness Analysis*. Modifikasi ini mencakup penambahan kelas yang belum ada sebelumnya, menghilangkan ambiguitas, serta menambahkan atribut pada objek domain [31].

3. *Detailed Design*

a. *Sequence Diagram*

Sequence diagram adalah representasi grafis yang dibuat berdasarkan *Robustness Analysis*. Setiap diagram dibuat khusus untuk masing-masing *use case*. *Sequence diagram* dianggap sebagai yang paling cocok untuk menggali skenario atau alur dari suatu *use case*. Selain itu, diagram ini tidak hanya mudah untuk digambarkan tetapi juga mudah dipahami oleh tim pengembang dan klien [31].

b. *Update Domain Model*

Domain model mengalami penyesuaian sejalan dengan informasi yang dihasilkan dari *Sequence Diagram*. Penambahan operasi pada objek domain dilakukan dalam tahap penyesuaian [31].

4. *Implementation*

a. *Coding/Unit Testing*

Tahapan ini dimulai dengan proses penulisan kode berdasarkan model pengembangan yang telah dipersiapkan sebelumnya. Setelah proses penulisan kode selesai, langkah selanjutnya adalah melakukan pengujian [31].

b. *Integration and Scenario Testing*

Tahapan dilakukan pengujian secara integrasi dan sesuai dengan skenario yang telah ditetapkan. Proses pengujian mencakup metode *black box testing* untuk menguji fungsionalitas eksternal [31].

2.2.6 My Structured Query Language (MySQL)

My Structured Query Language (MySQL) merupakan salah satu server database yang terkenal dan banyak digunakan. Keberhasilan MySQL disebabkan oleh penggunaan MySQL sebagai bahasa utama untuk mengelola database [32]. MySQL menjadi salah satu *Database Management System* (DBMS) yang termasuk

dalam kategori sistem manajemen basis data yang meliputi Postgre SQL, Oracle, MS SQL, serta DBMS lainnya. Fungsi MySQL adalah untuk mengelola database dengan menggunakan bahasa SQL. Sebagai sebuah perangkat lunak open source, MySQL dapat digunakan tanpa dikenakan biaya [33].

2.2.7 Laragon

Laragon adalah sebuah lingkungan pengembangan universal yang mendukung banyak bahasa pemrograman seperti PHP, Python, Node.js, Java, Go, dan Ruby. Laragon memiliki kelebihan sebagai sebuah *platform* yang dapat dipindahkan, terisolasi, ringan, cepat, dan juga mudah digunakan. Selain itu, Laragon juga menyediakan fitur-fitur seperti Pretty URLs untuk akses proyek secara mudah, kemampuan portable yang memungkinkan pemindahan folder Laragon tanpa merusak sistem, isolasi sistem yang memisahkan Laragon dari sistem operasi yang ada, operasi yang mudah dengan banyak konfigurasi yang sudah terotomatisasi. Laragon memiliki kemampuan untuk menambahkan versi terbaru dari berbagai bahasa pemrograman seperti PHP, Java, Ruby, Python, Apache, Nginx, MySQL, PostgreSQL, MongoDB, dan lainnya dengan mudah tanpa mengganggu sistem yang sudah ada. Selain itu, Laragon juga memiliki arsitektur terbaru yang sesuai untuk pengembangan aplikasi *web* modern. Ini termasuk dukungan untuk Apache dan Nginx secara bersamaan, serta fitur *QR-Code* untuk evaluasi yang lebih mudah [34].

2.2.8 Laravel

Laravel merupakan sebuah kerangka kerja (*framework*) berbasis PHP yang gratis dan bersifat *open source*. *Framework* ini dikembangkan oleh Taylor Otwell untuk mengembangkan aplikasi *web* ini berpola MVC (*Model-View-Controller*). Walaupun mengadopsi pola MVC, struktur MVC dalam Laravel memiliki perbedaan khusus yang membedakannya dari struktur MVC yang umum digunakan[35]. Laravel merupakan kerangka kerja PHP yang menitikberatkan pada kemudahan dan kefleksibelan dalam desainnya. Laravel mengadopsi pola desain MVC (*Model-View-Controller*). Artisan adalah salah satu fitur unggulan dalam

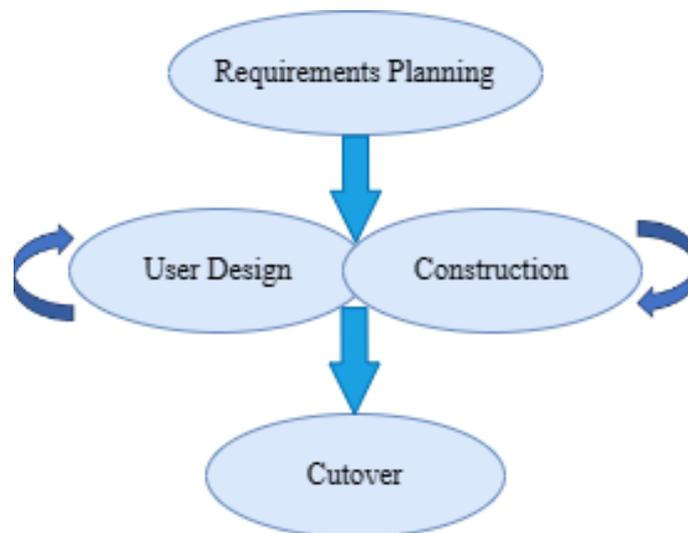
Laravel yang berupa alat baris perintah yang membantu dalam proses pengemasan dan pemasangan paket dalam pengembangan aplikasi [36].

2.2.9 Midtrans *Payment Gateway*

Midtrans merupakan suatu *payment gateway* yang menawarkan beragam fasilitas untuk berbagai cara pembayaran. Midtrans menyediakan metode pembayaran seperti pembayaran dengan kartu, transfer bank, debit langsung, dompet elektronik, dan pembayaran langsung di loket. Selain itu, API Midtrans juga dapat dikembangkan pada *framework* React Native dan NodeJS [37]. Terdapat delapan alur *backend* yang menjelaskan bagaimana integrasi Midtrans bekerja meliputi alur *checkout*, permintaan token *snap*, tanggapan token *snap*, halaman *checkout*, *snap pay*, proses *charge*, tanggapan *charge*, dan notifikasi *charge* [38].

2.2.10 *Rapid Application Development (RAD)*

Metode pengembangan sistem informasi yang dikenal sebagai *Rapid Application Development (RAD)* merujuk pada proses pengembangan yang dirancang untuk memberikan pengembangan yang lebih cepat dan hasil yang lebih baik dalam hal kualitas. *Rapid Application Development (RAD)* adalah sebuah metode pengembangan perangkat lunak yang terkenal karena kecepatan dan efektivitasnya. Metode ini pertama kali diperkenalkan oleh James Martin pada tahun 1991 [39].



Gambar 2.2 Siklus *Rapid Application Development* (RAD) [39]

Rapid Application Development (RAD) dibagi menjadi empat tahap terstruktur yang saling terkait satu sama lain. Keempat tahap ini saling bergantung dalam proses pengembangan sistem yang efektif, dan tahap-tahap tersebut sebagai berikut:

1. *Requirements Planning* (Perencanaan Kebutuhan)

Requirements Planning merupakan tahap analisis dengan pengguna bertemu untuk menetapkan tujuan pembuatan sistem dan memperoleh informasi mengenai hal-hal yang diperlukan untuk mencapai tujuan dan hasil yang diinginkan. Tahap ini melibatkan pengumpulan dan analisis kebutuhan pengguna serta perencanaan keseluruhan proyek. Tahap ini pengembang perangkat lunak perlu memahami kebutuhan pengguna dan memastikan bahwa proyek dapat dijalankan dengan efektif [39].

2. *User Design* (Desain Pengguna)

Tahapan *User Design* melibatkan pengguna dalam proses desain aplikasi untuk memastikan bahwa aplikasi yang dapat dibuat sesuai dengan kebutuhan. Pengembang perangkat lunak juga memastikan bahwa desain aplikasi memenuhi kebutuhan pengguna dan dapat digunakan dengan mudah. Tahap *User Design* mencakup proses iterasi yang melibatkan penggunaan *prototype* untuk menjelajahi dan memperkuat fungsi serta desain sistem yang belum sepenuhnya dipahami. *Prototype* digunakan untuk menguraikan fungsi sistem. Pendekatan ini

memungkinkan penyesuaian dan perbaikan yang terus-menerus, memungkinkan respon yang cepat terhadap perubahan kebutuhan atau perubahan selama pengembangan perangkat lunak dengan tujuan mencapai hasil proyek secara cepat dan fleksibel [39].

3. *Construction* (konstruksi)

Pada tahap *Construction*, pengembang aplikasi menggunakan perangkat lunak seperti generator kode dan alat pemodelan basis data. Selama tahap ini, pengembang perangkat lunak bertanggung jawab untuk memastikan bahwa aplikasi dapat dibangun dengan cepat dan efektif. *Tools* dapat membantu pengembang perangkat lunak dalam meningkatkan produktivitas dan mengoptimalkan proses pengembangan, sehingga menghasilkan aplikasi yang berkualitas dalam waktu yang lebih singkat [39].

4. *Cutover* (Pemotongan)

Tahapan *Cutover* dilakukan dengan pengujian dan peluncuran aplikasi ke dalam lingkungan organisasi. Pengembang perangkat lunak memastikan bahwa aplikasi berjalan dengan baik dan memenuhi kebutuhan pengguna. Pada tahapan ini, pengguna memiliki kesempatan untuk memberikan umpan balik dan menyetujui sistem yang telah dikembangkan [39].

2.2.11 *Blackbox Testing*

Metode pengujian *blackbox testing* merupakan pendekatan yang dapat digunakan dalam melakukan pengujian sistem berdasarkan fungsinya, tanpa memerhatikan *detail* kode program dan desain internal yang digunakan [40]. Pengujian *blackbox testing* memiliki fokus utama yaitu pada *input* dan *output* sistem tanpa mengetahui rincian internal programnya. Pendekatan ini dilakukan dari perspektif pengguna dan bertujuan untuk mengidentifikasi ambiguitas atau inkonsistensi dalam spesifikasi persyaratan [41].

Metode ini fokus pada pengujian spesifikasi kebutuhan fungsional aplikasi dengan tujuan memastikan bahwa fungsi, *input*, dan *output* dari aplikasi sesuai dengan spesifikasi awalnya. Kelebihan utama dari metode *Blackbox testing* adalah kemudahannya dalam implementasi, karena metode ini berfokus pada spesifikasi fungsional aplikasi dan tidak memerlukan pemahaman mendalam tentang bahasa

pemrograman yang digunakan. Pengujian dilakukan dengan sudut pandang seorang pengguna aplikasi, sehingga dapat dengan cepat mengidentifikasi celah-celah yang perlu diperbaiki, seperti ketidakjelasan dan inkonsistensi dalam spesifikasi fungsional aplikasi [10].

2.2.12 *User Acceptance Test (UAT)*

User Acceptance Testing (UAT) adalah proses pengujian yang melibatkan interaksi langsung antara pengguna akhir dan sistem, dengan tujuan memverifikasi bahwa fitur-fitur berjalan sesuai dengan kebutuhan pengguna tersebut. *User Acceptance Testing (UAT)* merupakan tahap pengujian terakhir dalam proses pengembangan sistem, di mana sistem telah selesai melalui tahap pengembangan [11]. *User Acceptance Testing (UAT)* menjadi bagian penting dalam rangkaian pengujian akhir perangkat lunak sebelum dikembangkan dan diluncurkan sebagai syarat agar sistem tersebut dapat diterima oleh pengguna [42]. *User Acceptance Testing (UAT)* merupakan pendekatan pemeriksaan yang melibatkan pengguna dalam menghasilkan suatu dokumen hasil uji untuk mengkonfirmasi bahwa sistem yang sudah dibuat dapat diterima baik oleh pengguna [43]. Proses dalam pengujian *User Acceptance Testing (UAT)* dilakukan dengan memberikan formulir penilaian sebagai bentuk penilaian langsung dari pengguna terhadap sistem yang telah dibuat. Data yang diperoleh dari formulir penilaian selanjutnya diolah menggunakan perhitungan presentase dengan mengacu pada hasil interpretasi skor penilaian pengguna [44].

Tabel 2. 6 Kriteria Skor Penilaian [44]

Presentase	Keterangan
0%-20%	Sangat Kurang Baik
21%-40%	Kurang Baik
41%-60%	Cukup Baik
61%-80%	Baik
81%-100%	Sangat Baik

Berikut ini adalah rumus perhitungan presentase berdasarkan skor dari skala yang sudah ditentukan [11] :

$$P = \frac{S}{Skor} \times 100\% \quad (2.1)$$

Keterangan:

P = Nilai presentase

S = Jumlah frekuensi dikalikan dengan skor jawaban

Skor = Skor tertinggi dikali dengan Jumlah ideal sampel

2.2.13 Visual Studio Code

Visual Studio Code merupakan teks editor yang lebih kuat serta ringan, kompatibel dengan Windows, macOS, dan Linux, serta dirancang untuk pengembangan perangkat lunak. Adanya dukungan dari bawaan untuk JavaScript, Node.js dan TypeScript, editor ini mempermudah pengembangan aplikasi *web*. Selain itu, Visual Studio Code memiliki beragam ekstensi yang mendukung berbagai bahasa program seperti PHP, Python, C++, Java, C#, Go, dan NET, sehingga memperluas fungsionalitasnya untuk kebutuhan pengguna yang berbeda [45].

Visual Studio Code adalah editor kode sumber yang diciptakan oleh Microsoft diperuntukan sistem operasi seperti Windows, Linux, dan macOS. Editor teks ini memiliki fungsi-fungsi seperti pemecahan masalah, integrasi kontrol git dengan GitHub, penyorotan sintaksis, penyelesaian kode cerdas serta *refactoring* kode. Editor kode ini juga sangat adaptif karena dapat dengan mudah disesuaikan oleh pengguna. Pengguna memiliki fleksibilitas untuk mengubah pintasan keyboard, tema, preferensi, dan juga dapat menginstal ekstensi sebagai fungsi tambahan [46].

2.2.14 Bootstrap

Bootstrap adalah sebuah kerangka kerja CSS (*Cascading Style Sheet*) yang digunakan dalam merancang sebuah tampilan situs *web*. *Framework* ini merupakan alat yang sangat berguna bagi *programmer* dalam membangun situs *web*.

Contohnya, dalam Bootstrap terdapat CSS yang menyediakan jenis-jenis, tombol-tombol, navigasi, dan elemen-elemen yang ketika digabungkan dengan JavaScript, memudahkan dan menperstabil proses pengembangan antarmuka [47].