

BAB II LANDASAN TEORI

2.1. Sebelumnya/Kajian Pustaka

Penelitian-penelitian sebelumnya yang digunakan pada penelitian ini dapat dilihat pada tabel 2.1.

Tabel 2.1 Kajian Pustaka

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critize</i>	<i>Synthesize</i>	<i>Summarize</i>
1	Aplikasi Informasi Hasil Tangkap Ikan Berbasis Web (Studi Kasus: Pelabuhan Perikanan Nusantara Kejawanan DKP3 Kota Cirebon) [3].	Penelitian ini sebelumnya digunakan sebagai acuan dalam rancang bangun sistem informasi hasil tangkap ikan berbasis <i>website</i> . Pada penelitian ini objek yang digunakan yaitu Pelabuhan Perikanan Nusantara Kejawanan DKP3 Kota Cirebon.	Penelitian ini sebelumnya digunakan sebagai acuan dalam rancang bangun sistem informasi hasil tangkap ikan berbasis <i>website</i> . Pada penelitian ini objek yang digunakan yaitu Pelabuhan Perikanan Nusantara Kejawanan DKP3 Kota Cirebon.	Dalam penelitian desain aplikasi masih terlihat kaku dan juga jenis data yang dimasukan sangat terbatas.	Pada penelitian ini memiliki tujuan untuk membuat aplikasi informasi hasil tangkap ikan berbasis <i>website</i> untuk Pelabuhan Perikanan Kejawanan DKP3 Kota Cirebon.	Penelitian tersebut menjelaskan tentang rancang bangun aplikasi informasi berbasis <i>website</i> hasil tangkap ikan pada Pelabuhan Perikanan Nusantara Kejawanan DKP3 Kota Cirebon. Aplikasi tersebut digunakan untuk mengelola atau memberikan informasi produksi hasil tangkapan ikan, data armada perikanan, data nelayan, dan data alat tangkap.
2	Aplikasi <i>E-Log Book</i> Tangkap	Penelitian ini digunakan sebagai	Penelitian ini menggunakan metode	Penelitian ini belum terintegrasi dengan	Penelitian terkait memiliki tujuan untuk	Penelitian tersebut tentang

No	Judul	Comparing	Contrasting	Critize	Synthesize	Summarize
	Ikan Menggunakan Progressive Web App [4].	acuan dalam rancang bangun aplikasi <i>E-log Book</i> Menggunakan Progressive Web APP tangkap ikan berbasis <i>website</i> . Objek pada penelitian ini yaitu Kementerian Kelautan dan Perikanan Indonesia.	SDLC <i>Waterfall</i> dan menggunakan <i>library workbox</i> dalam penerapan <i>Service Worker</i> . Tujuannya adalah dapat melakukan pendekatan kepada perangkat lunak secara sistematis dan sekuensial dimulai dari analisis, desain, kode, pengujian, dan juga pemeliharaan sehingga mendapatkan aplikasi <i>e-log book</i> sesuai dengan harapan.	sistem basis data Kementerian Kelautan dan Perikanan Republik Indonesia.	merancang dan membangun <i>e-log book</i> tangkap ikan berbasis <i>website</i> dikembangkan dengan Progressive Web APP dengan keluaran berupa sebuah aplikasi web yang dapat mengumpulkan data <i>log book</i> hasil tangkapan ikan yang dapat diakses sekalipun dalam keadaan jaringan <i>internet</i> tidak stabil.	rancang bangun aplikasi <i>e-log book</i> tangkap ikan yang dapat membantu dalam pengumpulan data <i>log book</i> hasil tangkapan di seluruh Indonesia.
3	Rancang bangun Sistem Informasi Elektronik Log Book Tangkap Ikan Berbasis Web [5].	Penelitian ini sebelumnya digunakan sebagai acuan dalam rancang bangun sistem informasi elektronik <i>log book</i> tangkap ikan berbasis <i>website</i> . Objek pada penelitian ini yaitu Pusat Pengkajian dan	Penelitian terkait menggunakan metode deskriptif. Tujuan metode ini adalah untuk menjelaskan suatu masalah secara aktual, sistematis dan akurat dengan temuan fakta yang ada di lapangan.	Pada penelitian ini tidak menampilkan fitur regulasi tangkap ikan di laut serta fasilitas komunikasi dua arah antara nelayan dengan petugas Pelabuhan.	Pada penelitian ini memiliki tujuan untuk merancang dan mengembangkan sistem informasi elektronik <i>log book</i> tangkap ikan berbasis <i>website</i> . Dengan harapan dapat memudahkan pihak berkepentingan	Pada penelitian terkait menjelaskan tentang perancangan sistem informasi elektronik <i>e-log book</i> menggunakan metode deskriptif sehingga dapat memperoleh data tangkap secara efektif, aman dan cepat.

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critize</i>	<i>Synthesize</i>	<i>Summarize</i>
		Rekayasa Teknologi Kelautan dan Perikanan			memperoleh data tangkap untuk mengelola sumber daya perikanan.	
4	Rancang Bangun Sistem Informasi Hasil Perikanan (SIHasper) di Kabupaten Cilacap [6].	Penelitian ini sebelumnya digunakan sebagai acuan dalam rancang bangun sistem informasi hasil perikanan berbasis <i>website</i> . Objek dari penelitian ini yaitu TPI (tempat pelelangan ikan) di Kabupaten Cilacap.	Penelitian terkait menggunakan metode <i>waterfall</i> pada pengerjaannya. Tujuan dari metode tersebut adalah menghasilkan suatu sistem yang efektif berdasarkan kebutuhan informasi para pengguna dengan membuat <i>database</i> .	Pada penelitian ini menu yang disajikan masih berupa menu yang sederhana dan membutuhkan adanya perbaikan setiap menunya.	Pada penelitian terkait memiliki tujuan untuk membuat sistem informasi hasil perikanan yang dapat menyampaikan suatu informasi menjadi lebih baik dan dapat diakses dengan mudah oleh seluruh pelaku sistem perikanan.	Penelitian terkait menjelaskan tentang rancang bangun sistem informasi hasil perikanan pada TPI Cilacap. Aplikasi tersebut digunakan untuk mengelola atau memberikan informasi dari hasil tangkapan ikan kepada TPI Cilacap.

2.2. Dasar Teori

2.2.1. Rancang Bangun

Rancang bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dari suatu sistem[7]. Rancang bangun adalah menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian

2.2.2. Sistem Informasi

Istilah Sistem Informasi (SIM) pertama kali diciptakan oleh Raymond McLeod Jr. pada tahun 1996. Raymond mengusulkan bahwa sistem informasi adalah sistem komputer yang dirancang untuk menyediakan berbagai informasi melalui persyaratan tertentu[8]. Sistem informasi menghubungkan antar pengguna pada suatu lingkup perusahaan, yang artinya setiap pengguna dapat mengetahui informasi satu sama lain guna mencapai tujuan bisnis yang telah ditentukan.

Terdapat lima komponen pada sistem informasi berdasarkan sisifungsionalnya, sebagai berikut:

1. Sistem Administrasi dan Operasional

Sistem administrasi dan operasional berperan dalam mengelola kegiatan rutin yang dilakukan berdasarkan prosedur yang telah disetujui.

Komponen sistem administrasi dan operasional perlu diawasi dengan teliti, karena selain memuat sebagian besar kegiatan, komponen ini juga bersifat kompleks.

2. Sistem Pelaporan

Sistem pelaporan memiliki peran sebagai penyusun laporan kinerja pada bisnis yang dijalankan. Laporan kinerja yang disusun dapat dilakukan secara rutin maupun periodik.

3. Data

data pada sebuah sistem informasi memiliki fungsi untuk memberikan kepastian terkait kualitas data dari berbagai sisi. Selain itu, data memiliki fungsi sebagai jembatan antara komponen sistem informasi lainnya dengan data yang dimiliki oleh perusahaan.

4. Sistem *Database*

Sistem *database* memiliki fungsi menyimpan seluruh data-data yang terdapat pada bagian perusahaan. Seluruh data yang dimiliki oleh perusahaan terdapat di dalamnya.

5. Sistem Pencarian

Sistem pencarian berperan dalam menyediakan sebuah informasi yang dibutuhkan oleh proses bisnis dalam suatu perusahaan. Informasi yang terkandung di dalamnya dapat digunakan sebagai acuan pengambilan keputusan bisnis melalui analisis.

2.2.3. Website

Website adalah kumpulan halaman (*page*) memuat beragam informasi yang membentuk sebuah rangkaian dengan jaringan-jaringan halaman lainnya (*hyperlink*). Sebuah *website* dapat dikatakan bersifat statis apabila informasi yang ditampilkan terfokus hanya pada satu hal saja, contohnya seperti website yang memuat informasi terkait teknologi, *fashion*, kuliner, dan sebagainya. Sedangkan apabila sebuah *website* dikatakan bersifat dinamis apabila *website* tersebut memunculkan halaman informasi yang selalu berubah-ubah, tak hanya terfokus pada satu hal saja [9].

Sebuah *website* terdiri dari 3 elemen. Tanpa elemen-elemen

ini, sebuah *website* tidak dapat diakses oleh siapapun. Unsur elemen pada *website* tersebut adalah sebagai berikut:

1. Domain

Domain adalah nama unik yang digunakan pada sebuah *website* untuk mengidentifikasi alamat *email* secara umum. Sederhananya, domain adalah halaman yang diketik ke kolom *URL* sebuah *browser*. Ilustrasi :

Domain diibaratkan adalah nama yang akan dikunjungi contohnya seperti facebook.com, sedangkan hosting/server adalah bangunan fisik tempat disimpannya berbagai hal atau dalam konteks server untuk menyimpan *file* berupa foto, video dan teks yang diunggah untuk keperluan online.

2. Hosting

Hosting adalah tempat semua informasi di sebuah *website* yang telah disimpan di server. Data yang disimpan pada *hosting* di antaranya datateks, gambar, *script*, dan sebagainya. Ilustrasi :

Hosting digunakan untuk menampung sebuah *website* alias dengan alamat *IP* diganti dengan nama domain atau pada contoh adalah “Purwokerto”. Kemudian tuan rumah juga menyimpan informasi terkait “ruangan” pada sebuah *website*, yang disebut “subdomain”.

3. Content

Content (konten) merupakan isi informasi yang ditampilkan pada sebuah *website*. Tanpa sebuah konten, sebuah *website* tidak akan memiliki tujuan yang jelas. Konten pada sebuah *website* biasanya memuat susunan teks, gambar, video, dan sebagainya. Ilustrasi :

Sebuah *website* bisnis/toko online sangat membutuhkan konten karena merupakan cara utama dalam menyampaikan suatu informasi kepada konsumen. Seperti halnya membuat konten yang bagus, Deskripsi produk yang tidak dangkal, mudah

dipahami, dan lain-lainnya.

2.2.4. Laravel

Laravel merupakan sebuah *framework* yang berfungsi sebagai kerangka pemrograman *website* yang bersifat *open source*. Saat ini, *Laravel* telah banyak digunakan oleh banyak *developer* di seluruh dunia. *Laravel* banyak digunakan karena penggunaan pada *Laravel* yang cukup mudah. Selain itu, *Laravel* memiliki fitur dokumentasi yang cukup lengkap [10]. Pola arsitektur pada *Laravel* merupakan metode *Model View Controller* (MVC). Pada metode tersebut, komponen akan saling terbagi sesuai dengan fungsinya, seperti *user interface*, pengelolaan data, serta bagian perintah (*command*) pada *website*. *Framework Laravel* yang akan digunakan pada implementasi pembuatan *website* sistem informasi perikanan tangkap yaitu versi *Laravel* 8.0.

2.2.5. Bootstrap

Bootstrap adalah salah satu jenis *framework* gabungan dari *CSS* dan *Javascript* yang ditawarkan sebagai alternatif diantara *framework* lainnya yang dimana awal *framework* ini dikembangkan oleh Mark Otto dan Jacob Thornton dikantor Twitter dengan maksud untuk menghadirkan konsistensi terhadap tahap *interface development* dalam membangun sebuah *website*. Saat ini *bootstrap* itu sendiri sudah menjadi aplikasi yang *open-source* dan mendukung platform seperti *HTML5* dan *CSS3*. Dalam *Bootstrap* itu sendiri seperti fungsinya yaitu memudahkan *devloper* dalam membangun *interface* dalam *websitenya*, terdapat template untuk font atau tombol, navigasi, dan *interface* lainnya yang *responsive* ketika diakses melalui aplikasi yang berbasis *desktop*[11]. *Framework Bootstrap* yang akan digunakan pada implementasi pembuatan *website* sistem informasi perikanan tangkap menggunakan versi *Bootstrap* 5.0.

2.2.6. Database

Database atau basis data adalah kumpulan informasi yang

disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut [12]. Kegunaan utama sistem basis data adalah agar pemakai mampu menyusun suatu pandangan (*view*) abstraksi data. Hal ini bertujuan untuk menyederhanakan interaksi antara pengguna dengan sistemnya dan basis data dapat mempresentasikan pandangan yang berbeda kepada para pengguna, *programmer*, dan administratornya.

2.2.7. MySQL

MySQL merupakan sebuah perangkat lunak atau *software* sistem basis data *SQL* atau *DBMS Multithread* dan *multi user*. *MySQL* sebenarnya merupakan turunan dari salah satu konsep utama dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis. *MySQL* diciptakan oleh Michael "Monty" Widenius pada tahun 1979, seorang programmer komputer asal Swedia yang mengembangkan sebuah sistem *database* sederhana yang dinamakan *UNIREG* yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing* [13]. *MySQL* bersifat *open source* dengan ketentuan *General Public License (GPL)*, yang dapat diartikan bahwa *MySQL* dapat digunakan untuk kepentingan pribadi maupun komersil secara gratis. Kelebihan yang dimiliki oleh *MySQL* sebagai berikut:

1. Dapat melakukan integrasi berbagai macam bahasa pemrograman.
2. Tidak membutuhkan spesifikasi komputer yang terlalu besar.
3. Dapat dijalankan oleh *user* dalam waktu yang bersamaan (*multi-user*).
4. Bersifat *open source* dan memiliki ketentuan lisensi GPL.
5. Memiliki struktur dan operasi tabel yang lebih fleksibel.
6. Memiliki tipe data yang sangat bervariasi.

MySQL yang akan digunakan pada implementasi pembuatan *website* sistem informasi perikanan tangkap menggunakan versi *MySQL* 6.0.0.

2.2.8. XAMPP

XAMPP adalah kumpulan kelompok perangkat lunak yang terdiri dari *Apache*, *PhpMyAdmin*, *MySQL*, *PHP*, *Filezilla*, *Perl*, dan lainnya. Selain itu *XAMPP* memiliki fungsi dalam memberikan kemudahan instalasi *PHP*. Kemudian *xampp* diperlukan sebagai wadah lingkungan pengembangan web [14]. *XAMPP* yang akan digunakan pada implementasi pembuatan *website* sistem informasi perikanan tangkap menggunakan versi *XAMPP* 8.0.6/PHP.

2.2.9. Visual Studio Code

Visual Studio Code adalah perangkat lunak editor teks yang dikembangkan oleh Microsoft untuk berbagai *platform* seperti *Windows*, *Linux*, dan *Mac*. *Visual Studio Code* mendukung banyak bahasa pemrograman seperti *C#*, *Python*, *Java*, dan lainnya[15]. Fungsi dari *Visual Studio Code* adalah sebagai teks editor yang digunakan untuk mengembangkan sebuah program atau sistem. Terdapat banyak fitur yang terdapat di dalamnya, contohnya adalah fitur *cross platform*, *lightweighth*, *powerful editor*, *code debugging*, dan masih banyak lagi. *Visual Studio Code* yang akan digunakan pada implementasi pembuatan *website* sistem informasi perikanan tangkap menggunakan versi *Visual Studio Code* 1.56.

2.2.10. Javascript

Javascript adalah bahasa pemrograman yang memiliki karakteristik bahasa pemrograman *client-side*. Di mana property ini adalah jenis bahasa yang dikelola oleh *clint*. *Javascript* berbeda dari *Java*, dalam bentuk tertulis, itu telah tertanam dalam dokumen *HTML* atau dapat digunakan sebagai dokumen terpisah yang terkait dengan dokumen target[16]. *Javascript* akan digunakan pada implementasi pembuatan *website* sistem informasi perikanan

tangkap menggunakan versi *Javascript* 4.2.



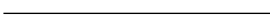
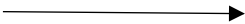
2.2.11. StarUML

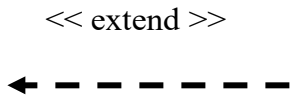
StarUML adalah alat *Unified Modeling Language* (UML) bertujuan untuk menulis suatu denah atau sketsa perangkat lunak yang menentukan serta melakukan dokumentasi artefak pada sistem perangkat lunak [17]. Dalam melakukan rancangan sistem penulis memakai komponen *diagram StarUML* sebagai berikut.

2.2.12. Use Case Diagram

Menurut Mamed Rofendy Manalu, *Use Case Diagram* Deskripsi proses yang mewakili tindakan yang dilakukan oleh aktor dalam menyelesaikan tugas [17]. Berikut adalah simbol-simbol yang terdapat pada *use case diagram* dapat dilihat pada tabel 2.2 sebagai berikut:

Tabel 2.2 Simbol *Use case Diagram*



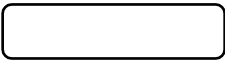
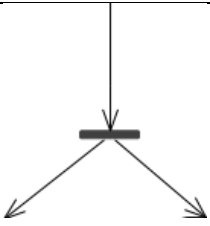
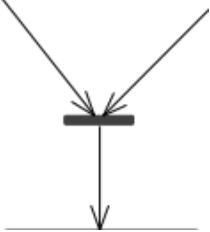
Simbol	Nama	Keterangan
	<i>Use Case</i>	Penggambaran interaksi antara aktor dan sistem.
	<i>Actor</i>	Representasi seseorang atau sistem yang berinteraksi bersama <i>use case</i> .
	<i>Association</i>	Sebagai penghubung antara tujuan (objek) satu dengan tujuan lainnya.
	<i>Generalization</i>	Hubungan berbagi perilaku objek induk terhadap objek anak
<< include >>	<i>Include</i>	Untuk menunjukkan bahwa use case adalah bagian fungsional dari use case lain

Simbol	Nama	Keterangan
	<i>Extend</i>	Untuk menunjukkan bahwa <i>use case</i> adalah perpanjangan fungsional dari <i>use case</i> lain ketika kondisi lain terpenuhi.

2.2.13. Activity Diagram

Activity Diagram ini menggambarkan alur kerja dari suatu sistem atau proses bisnis [17]. Berikut adalah simbol-simbol yang terdapat pada *activity diagram* dapat dilihat pada tabel 2.3 sebagai berikut:

Tabel 2.3 Simbol Activity Diagram



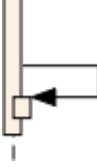


Simbol	Keterangan
	<i>Start Point</i> , terletak di pojok kiri atas dan merupakan awal kegiatan
	<i>End Point</i> , akhir kegiatan
	<i>Activities</i> , menggambar kan pekerjaan atau proses aktivitas
	<i>Fork</i> atau percabangan, digunakan untuk menunjukkan aktivitas paralel atau menggabungkan dua aktivitas paralel menjadi satu.
	<i>Join</i> atau penggabungan, digunakan untuk menunjukkan adanya suatu dekomposisi.

2.2.14. Sequence Diagram

Sequence Diagram digunakan untuk mendefinisikan aliran interaksi antara objek. Isi dari *diagram* ini harus sesuai dengan *use*

case dan *diagram* kelas [17]. Berikut adalah simbol-simbol yang terdapat pada *sequence diagram* dapat dilihat pada tabel 2.4 sebagai berikut:

Tabel 2.4 Simbol *Sequence Diagram*






Simbol	Keterangan
	<i>Entity Class</i> , adalah bagian dari sistem yang berisi sekumpulan kelas berupa unit-unit yang membentuk model awal sistem dan menjadi dasar pembuatan <i>database</i> .
	<i>Message</i> , interaksi antar objek dengan objek lain. Objek dapat mengirimkan pesan ke objek lainnya.
	<i>Recursive</i> , menggambarkan kirim pesan ke diri sendiri
	<i>Activation</i> , menunjukkan masa hidup dari objek
	<i>Return</i> , komunikasi antar objek.

2.2.15. Class Diagram

Pada *Class Diagram* ini menjelaskan hubungan terjadi pada suatu objek dengan objek lain yang membentuk sistem aplikasi [17]. Berikut adalah simbol-simbol yang terdapat pada *class diagram* dapat dilihat pada tabel 2.5 sebagai berikut:

Tabel 2.5 Simbol *Class Diagram*


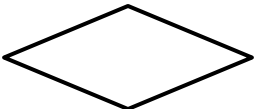
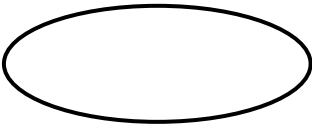

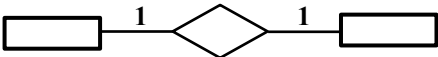
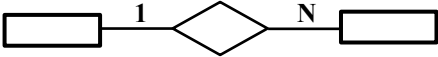
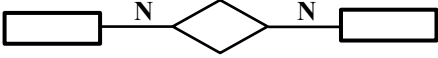
Simbol	Deskripsi
--------	-----------

<p>Kelas</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">Nama Kelas</td> </tr> <tr> <td style="text-align: center;">Atribut</td> </tr> <tr> <td style="text-align: center;">Operasi</td> </tr> </table>	Nama Kelas	Atribut	Operasi	<p>Kelas yang ada pada struktur sistem.</p>
Nama Kelas				
Atribut				
Operasi				
<p>Antarmuka</p> 	<p>Seperti konsep antarmuka pada pemrograman berorientasi objek.</p>			
<p>Asosiasi</p> 	<p>Relasi kelas dengan makna umum.</p>			
<p>Generalisasi</p> 	<p>Relasi kelas dengan makna generalisasi spesialisasi.</p>			
<p>Kebergantungan</p> 	<p>Relasi kelas dengan makna ketergantungan antar kelas.</p>			
<p>Agresiasi</p> 	<p>Relasi antar kelas dengan makna semua bagian.</p>			

2.2.16. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah alat rancang bangun untuk memodelkan suatu basis data. Tujuan dibuatnya ERD dalam suatu organisasi adalah untuk pemodelan yang menunjukkan hubungan antar data yang mempunyai relasi, juga mendokumentasikan data yang ada dengan cara menjelaskan tiap masing-masing data dan relasinya [18]. Berikut adalah simbol-simbol yang terdapat pada *entity relationship diagram* dapat dilihat pada tabel 2.6 sebagai berikut:

Tabel 2.6 Entity Relationship Diagram

Simbol	Deskripsi
Entitas 	Kumpulan dari objek yang dapat diidentifikasi secara unik
Relasi 	Hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain yaitu <i>one to one</i> , <i>one to many</i> , dan <i>many to many</i> .
Atribut 	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas
Hubungan 	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.
<i>One to One</i> 	Satu anggota entitas dapat berelasi dengan satu anggota entitas lainnya.
<i>One to Many</i> 	Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lainnya.
<i>Many to Many</i> 	Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

2.2.17. System Development Life Cycle (SDLC)

System development life cycle (SDLC) merupakan suatu bentuk penggambaran tahapan proses pengembangan sistem. Siklus hidup pengembangan sistem menyajikan metodologi atau proses yang diorganisasikan untuk membangun suatu sistem. *System Development Life Cycle* (SDLC) merupakan model klasik yang bersifat sistematis dan berurutan dalam membangun perangkat lunak [19]. Fungsi utama dari SDLC adalah mengakomodasi kebutuhan pengguna berkaitan dengan sistem yang akan dikembangkan. Kebutuhan pengembangan sistem dapat berupa perubahan atau penciptaan aplikasi baru apakah secara modular maupun dengan proses instalasi baru[20]. SDLC memiliki tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi. Ada 6 tahapan secara umum didalam SDLC yaitu:

1. Analisis sistem, yaitu membuat analisis aliran kerja yang sedang berjalan.
2. Desain Spesifikasi kebutuhan sistem, yaitu melakukan perincian mengenai apa saja yang dibutuhkan dalam pengembangan sistem dan membuat perencanaan yang berkaitan dengan proyek sistem.
3. Kontruksi sistem, yaitu membuat desain aliran kerja dan desain pemrograman yang diperlukan untuk pengembangan sistem informasi.
4. Implementasi sistem, yaitu tahap menjalankan sistem yang sesuai dengan fungsi masing-masing.
5. Pengujian sistem, yaitu melakukan pengujian terhadap sistem yang telah dibuat.
6. Pemeliharaan sistem, yaitu menerapkan dan memelihara sistem yang telah dibuat.

SDCL memiliki beberapa pendekatan metode diantaranya adalah:

1. Metode Prototype

Metode *Prototype* menurut Roger S. Pressman dengan metode *Prototype* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem[21]. *Prototype* merupakan salah satu metode dari SDLC yang merupakan metode pengembangan sistem berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Pada model ini pengembang dan pelanggan menentukan tujuan keseluruhan untuk perangkat lunak yang akan dikembangkan dan mengidentifikasi persyaratan-persyaratan apa saja yang diketahui[22].

2. Metode Agile Scrum

Metode agile adalah salah satu metodologi SDLC yang berbasis pada incremental model dan iteratif. Agile merupakan sebuah metode pengembangan software dengan berbagai proses kecil yang berulang. Hal ini menjadikan metode agile cukup fleksibel akan perubahan ditengah proses pengembangan. Dalam penerapannya agile membutuhkan kerangka kerja yang mendukung konsep agile, salah satunya adalah scrum[23].

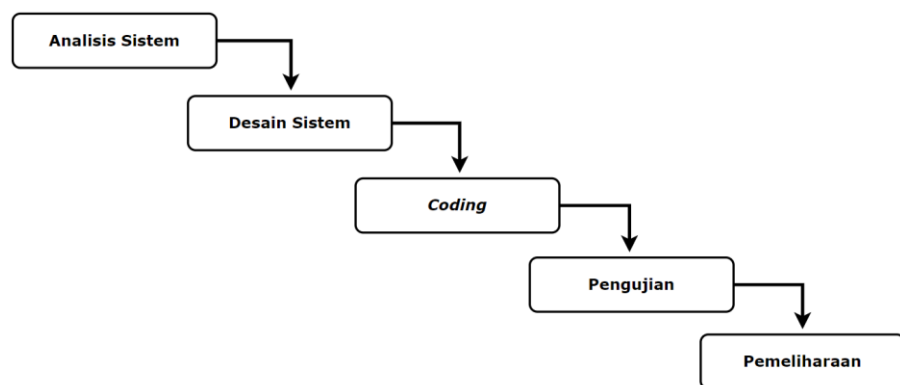
Menurut Schwaber and Sutherland, scrum merupakan sebuah kerangka kerja yang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah dan pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif. scrum dibagi menjadi beberapa aktivitas, di antaranya adalah sebagai berikut[24]:

- *Backlog* adalah daftar kebutuhan yang menjadi prioritas klien dan daftar yang dibuat dapat bertambah.
- *Sprints* merupakan unit pekerjaan yang diperlukan untuk memenuhi kebutuhan yang ditetapkan dalam *backlog* sesuai dengan waktu yang ditetapkan dalam *time-box*. Selama proses ini berlangsung, *backlog* tidak ada tambahan.

- *Scrum Meeting* merupakan pertemuan rutin yang dilakukan per hari untuk evaluasi pekerjaan, hambatan yang ada, dan target penyelesaian untuk bahan pertemuan selanjutnya.
- *Demo* adalah penyerahan *software increment* ke klien yang didemonstrasikan dan dievaluasi oleh klien

2.2.18. Metode *Waterfall*

Menurut Moenir dan Yuliyanto, model *waterfall* adalah model klasik yang bersifat sistematis dan berurutan dalam membangun *software*. Nama model *waterfall* sebenarnya adalah “*Linear Sequential Model*”. Model *waterfall* sering disebut juga dengan “*Classic Life Cycle*” atau metode *waterfall*. Metode *waterfall* termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970an [25]. Urutan dalam merancang metode *waterfall* dapat dilihat pada gambar 2.1, sebagai berikut:



Gambar 2.1 Metode *Waterfall*

Dari gambar 2.1 menjelaskan mengenai tahapan yang dilakukan pada metode *Waterfall*[17].

1. Analisis Sistem

Pada tahap ini, menjelaskan pada saat dilakukanlah analisis kebutuhan sistem yang digunakan dalam proses rancang bangun.

2. Desain Sistem

Pada tahap ini, menjelaskan pada saat dilakukan rancang bangun suatu sistem yaitu dengan membuat jenis atau tampilan beserta alur dari perancangan program.

3. Pengkodean (*coding*) Sistem

Pada tahap ini, menjelaskan kode pemrograman yang digunakan untuk membuat sebuah aplikasi seperti halnya dalam proses pembuatan program aplikasi dengan menggunakan bahasa pemrograman *javascript*, *php*, dan lainnya.

4. Pengujian Sistem

Pada tahap ini, diadakan pengujian pada program aplikasi yang telah dibuat oleh pengembang.

5. Pemeliharaan Sistem

Pada tahap akhir dalam metode ini, program aplikasi yang telah selesai dan dapat dijalankan akan dilakukan sebuah pemeliharaan aplikasi. Pemeliharaan melibatkan koreksi kesalahan yang tidak ditemukan pada langkah sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, implementasi unit sistem, dan penyesuaian sistem sesuai dengan kebutuhan pengguna.

2.2.19. Black Box Testing

Pengujian *Black Box* adalah metode pengujian dimana pengujian perangkat lunak didasarkan pada spesifikasi fungsional. Pengujian dalam *Black Box* tidak melihat pada desain dan *source code* dari aplikasi, melainkan lebih menitikberatkan pada *input* dan *output* yang dihasilkan oleh sistem yang diuji. Pengujian *Black Box* juga merupakan cara mudah untuk meningkatkan akurasi dalam memperbaiki kesalahan yang ada [26]. Pengujian *Black Box* dilakukan berdasarkan kebutuhan pengguna sehingga kebutuhan yang hilang atau tidak terduga dapat dengan mudah diidentifikasi.