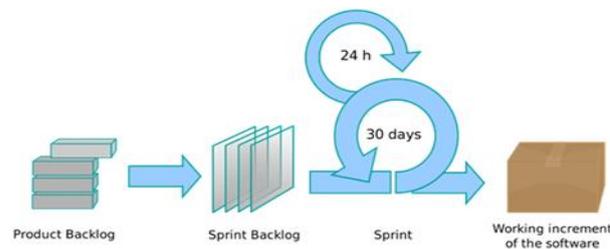


BAB II

TINJAUAN PUSTAKA

2.1 Metode Pengembangan *Scrum*

Metode pengembangan yang digunakan dalam pembuatan aplikasi ini yaitu dengan menggunakan metode *scrum*.



Gambar 2.1 Metode Scrum

Model *scrum* merupakan metode pengembangan perangkat lunak secara cepat (*agile*). Model *scrum* memiliki beberapa kelebihan seperti, mampu mentransformasikan proses bisnis yang sulit menjadi mudah dikembangkan, dengan model *scrum* mampu memonitoring dan mengontrol aktivitas pada proses pengembangan sistem (Kuswinanti et al., 2021) Metode pengembangan sistem mengacu pada metode *Scrum* yang merupakan sebuah metode yang mudah dikontrol, fleksibel, memuat strategi pengembangan menyeluruh dimana seluruh tim bekerja sebagai satu unit untuk mencapai goal yang sama (Warkim et al., 2020)

Proses pengembangan menggunakan metode *scrum* terdapat empat tahapan pengembangan yaitu : ((1) *product backlog*, (2) *sprint Backlog*, (3) *sprint log*, (4) *Increment*. Adapun kerangka kerja dalam model *scrum* terdiri dari:

a. *Product Backlog*

Tahap *Product Backlog* lebih mengarah kepada pengumpulan kebutuhan, pembaruan, pemeliharaan, dan deskripsi singkat tentang fungsi-fungsi yang diinginkan pada saat aplikasi akan dibangun.

b. Sprint Backlog

Tahap *Sprint backlog* dilakukan untuk sebuah proses pemenuhan kebutuhan sesuai dengan yang diinginkan pada proses *Product Backlog* sebelumnya.

c. Sprint Log

Tahap *Sprint Log* merupakan proses dimana paparan aplikasi dalam bentuk sebuah *prototype* dan pemaparan dalam bentuk hal teknis baik berupa *tools* yang dibutuhkan untuk mengembangkan aplikasi.

d. Increment

Tahap Increment merupakan hasil akhir dari tahap *Product Backlog* yang telah selesai dikembangkan pada saat tahapan *Sprint Log*. Pada tahap ini, diharapkan tahapan *Increment* telah selesai dilakukan sehingga mampu untuk digunakan sesuai dengan yang diinginkan.

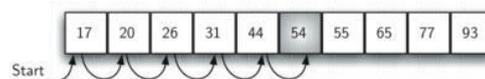
2.2 Algoritma Sequential Search

Sequential Search adalah proses membandingkan setiap elemen array satu persatu secara beruntun dimulai dari elemen pertama hingga elemen yang dicari ditemukan atau hingga elemen terakhir dari array. Metode *Sequential Search* atau disebut pencarian beruntun dapat digunakan untuk melakukan pencarian data baik pada array yang sudah terurut maupun yang belum terurut. Proses yang terjadi pada metode pencarian ini adalah sebagai berikut :

1. Membaca array data.
2. Menentukan data yang dicari.
3. Mulai dari data pertama sampai dengan data terakhir,

Data yang dicari dibandingkan dengan masing-masing data di dalam array. Jika data yang dicari tidak ditemukan maka semua data atau elemen array dibandingkan sampai selesai. Jika data yang dicari ditemukan maka perbandingan akan dihentikan.

Proses pencarian data dengan metode ini cukup sederhana dan mudah. Proses pencarian data dilakukan dengan mencocokkan data yang dilakukan secara berurut satu demi satu dimulai dari data ke-1 hingga data pada urutan terakhir. Jika data yang dicari mempunyai nilai yang sama dengan data yang ada dalam kelompok data, berarti data telah ditemukan. Jika data yang dicari tidak ada yang cocok dengan data dalam sekelompok data, data tersebut tidak ada dalam sekelompok data. Selanjutnya kita tinggal menampilkan hasil yang diperoleh tersebut.



Gambar 2.2 Sequential Search

- a. Pertama data melakukan perbandingan satu per satu secara berurutan dalam kumpulan data dengan data yang di cari sampai data tersebut ditemukan atau tidak ditemukan.
- b. Pada dasarnya, pencarian ini hanya melakukan pengulangan data dari 1 sampai dengan jumlah data (n).
- c. Setiap pengulangan, dibandingkan data ke-i dengan data yang sedang dicari.
- d. Apabila data sama dengan yang dicari, berarti data telah berhasil di temukan. Sebaliknya apabila sampai akhir melakukan pengulangan tidak ada data yang sama dengan yang dicari, berarti data tidak ada yang ditemukan.

Urutan Algoritma Sequential Searching:

1. $i \leftarrow 0$
2. $Ketemu \leftarrow \text{false}$
3. Selama (tidak ketemu) dan ($i < N$) kerjakan baris 4
4. Jika ($\text{Data}[i] = \text{key}$) maka $\text{ketemu} \leftarrow \text{true}$

Jika tidak $i \leftarrow i + 1$

5. Jika (Ketemu) maka i adalah indeks dari data yang dicari

2.3 Angkutan Barang

Kata “pengangkut” berasal dari kata dasar “angkut” yang memiliki arti mengangkat dan membawa. Dalam kamus hukum tertulis bahwa, pengangkutan adalah timbal balik antara pengangkut dan pengirim, dimana pengangkut mengikatkan diri untuk melakukan pengangkutan barang atau orang dari suatu tempat ke tempat tujuan tertentu dengan selamat, sedangkan pengirim mengikatkan diri untuk membayar ongkos angkutan.

2.3.1 Klasifikasi Angkutan Barang

Adapun klasifikasi dari angkutan barang yaitu :

1. Barang Kering
 - a. Biasanya barang belum jadi atau bahan baku
 - b. Pada umumnya tidak dikemas dan dapat langsung di bongkar muat.
2. Barang Cair
 - a. Berupa cairan atau kemasan atau curah
 - b. Perlu penanganan khusus
 - c. Pada volume yang besar, dimungkinkan melalui pipa.
3. Barang umum
 - a. Barang kiriman berupa barang jadi atau setengah jadi
 - b. Dikemas dalam satu unit (misalnya melalui kontainerisasi)
 - c. Moda angkutan tergantung pada kemasan.

2.4 Unified Modeling (UML)

Menurut (Rosadi et al., 2018) UML (Unified Modeling Language) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek. UML tidak hanya merupakan sebuah bahasa

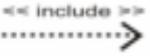
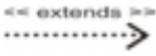
pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah objek – oriented database.

Menurut Rosa dan Shalahuddin (2016) *Unified Modelling Language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu Unified Modeling Language (UML) dalam UML terdiri dari 13 macam diagram yang dikelompokkan dalam 4 kategori sebagai berikut :

2.4.1 Use Case Diagram

Merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. secara kasar usecase digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol dan keterangan use case diagram seperti pada tabel 2.1.

Tabel 2.1 Simbol dan Keterangan *Use case Diagram*

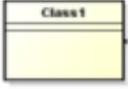
SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun
	<i>Use case</i>	Menggambarkan bagaimana seseorang menggunakan system
	<i>Relasi asosiasi</i>	Relasi yang dipakai untuk menunjukkan hubungan antara <i>actor</i> dan <i>use case</i>
	<i>Relasi include</i>	Memungkinkan satu use case menggunakan fungsionalitas yang disediakan oleh usecase lainnya
	<i>Relasi extend</i>	Memungkinkan suatu use case secara optional menggunakan fungsionalitas yang disediakan oleh usecase lainnya

Sumber : (Nugroho, 2010)

2.4.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar pembuatan program membuat kelas-kelas sesuai rancangan didalam diagram kelas antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut ini adalah simbol-simbol yang ada pada *Class Diagram*. Simbol dan keterangan class diagram seperti pada tabel 2.2.

Tabel 2.2 Simbol dan Keterangan *Class Diagram*

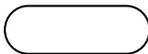
SIMBOL	NAMA	KETERANGAN
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Generalization</i>	Garis yang melambangkan konsep pewarisan dari suatu kelas ke satu atau lebih sub kelas
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

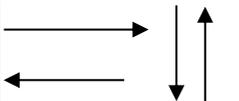
Sumber : (Nugroho, 2010)

2.4.3 Flowchart

Flowchart adalah unit simbol gambar (chart) yang menunjukkan aliran (flow) dari proses terhadap data. Flowchart adalah Suatu Bagan yang menggambarkan arus logika dari data dalam suatu program dari awal sampai akhir. Bagan alir program merupakan alat yang berguna bagi programmer untuk mempersiapkan program yang rumit.

Tabel 2.3 Simbol dan Keterangan *Flowchart*

Simbol	Nama Simbol	Keterangan
	<i>Predefined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage
	<i>Terminal</i>	Simbol untuk permulaan atau akhir dari suatu program
	<i>Manual Input</i>	Simbol untuk pemasukan data secara manual on-line keyboard

	<i>Arus / Flow</i>	Penghubung antara prosedur / proses
	<i>Connector</i>	Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang sama
	<i>Off-line Connector</i>	Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang lain
	<i>Input-Output</i>	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
	<i>Document</i>	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output di cetak dikertas

2.5 Penelitian Sebelumnya

Dalam penyusunan skripsi ini, peneliti terinspirasi dan mereferensi dari penelitian penelitian sebelumnya yang berkaitan dengan skripsi ini. Daftar penelitian terkait ialah sebagai berikut :

Tabel 2.4 Penelitian Terkait

No	Judul, Penulis, Tahun	Jumlah dan Atribut	Algoritma/Metode
1.	Sistem Informasi Pembuatan Manifest Muatan Kapal Berbasis Dekstop Dan Android (Utama & Nuryana, 2020)	Ada 4 atribut : <i>Manifest,</i> <i>Ship, SDLC,</i> <i>Android</i>	<i>System</i> <i>Development</i> <i>Life Cycle (SDLC)</i>
2.	Perancangan Aplikasi Pencatatan Data Manifes Penumpang Kapal Penyeberangan Danau Toba Menggunakan Web (Wahit Ambo et al., 2021)	Ada 2 atribut : <i>Manifest and</i> <i>crossing ship</i>	Kualitatif
3.	Pelaksanaan Pengawasan Terhadap Kendaraan Yang Muatannya Melebihi Daya Angkut Dan Dimensi Berdasarkan Undang-Undang No.22 Tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan Di Pulau Lombok (Mursityo, 2020)	Ada 4 Atribut : <i>supervision,</i> <i>vehicle,</i> <i>haulage,</i> <i>dimensions</i>	Empiris
4.	Perancangan Aplikasi Pencatatan Data Manifest Penumpang Kapal Penyeberangan Di Minahasa Selatan Berbasis Web (Rizan et al., 2019)	Ada 2 Atribut : <i>manifes and</i> <i>crossing ship</i>	<i>Waterfall</i>

No	Judul, Penulis, Tahun	Jumlah dan Atribut	Algoritma/Metode
5.	Implementasi Kebijakan Odol Dalam Upaya Meningkatkan Sistem Pengawasan Dan Pengendalian Muatan Angkutan Barang (Lambang Antono, 2022)	Ada 4 atribut : <i>transport,</i> <i>supervision,</i> <i>control,</i> <i>cargo</i>	Kualitatif

