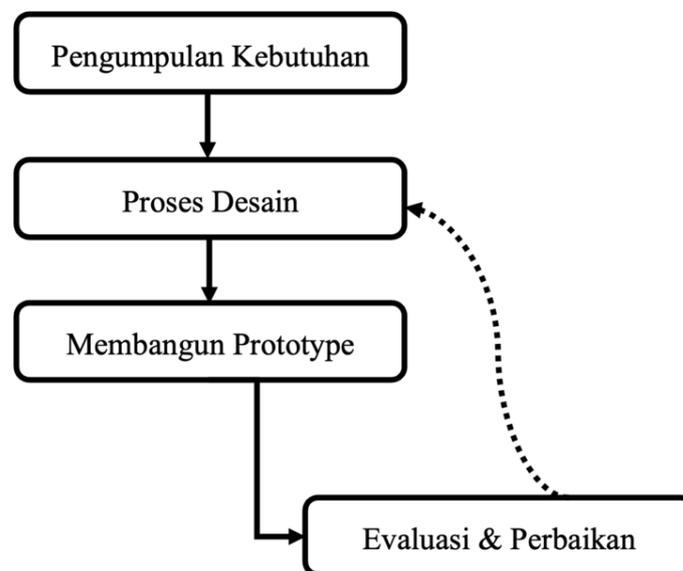


BAB II

TINJAUAN PUSTAKA

2.1 Metode Pengembangan *Prototype*

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



Gambar 2.1 Langkah-Langkah *Prototype*

Menurut (Purnomo 2017), prototyping dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi dan kebutuhan operasional sistem. Langkah- langkah dalam prototyping adalah sebagai berikut :

1. Pengumpulan Kebutuhan.
2. Proses desain yang cepat.
3. Membangun prototipe.
4. Evaluasi dan perbaikan.

Mengumpulkan persyaratan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan tujuan keseluruhan perangkat lunak, mengenali kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat. Desain berfokus pada representasi aspek perangkat lunak dari sudut pandang pengguna ini meliputi format input, proses, dan output.

Desain yang cepat mengarah pada pengembangan prototipe, prototipe dievaluasi oleh pengguna dan analis desain dan digunakan untuk menyesuaikan dengan kebutuhan perangkat lunak yang akan dikembangkan. Prototipe diatur untuk memenuhi kebutuhan pengguna, dan pada saat yang sama developer lebih mengerti dan dengan apa yang perlu dilakukan. Setelah keempat langkah prototyping dijalankan, langkah selanjutnya adalah pembuatan atau desain produk yang sebenarnya.

2.2 Algoritma *Priority Scheduling*

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing. Prioritas tersebut dapat ditentukan melalui beberapa karakteristik antara lain:

1. *Time limit*
2. *Memory requirement*
3. *Akses file*
4. Tingkat kepentingan proses

Priority scheduling juga dapat dijalankan secara *preemptive* maupun *nonpreemptive*. Pada *preemptive*, jika ada suatu proses yang baru datang memiliki prioritas yang lebih tinggi daripada proses yang sedang dijalankan, maka proses yang sedang berjalan tersebut dihentikan, lalu CPU dialihkan untuk proses yang baru datang tersebut.

Sementara itu, pada *non-preemptive*, proses yang baru datang tidak dapat mengganggu proses yang sedang berjalan, tetapi hanya diletakkan di dalam *queue*.

Berikut contoh analisis Algoritma *Priority Scheduling* :

Tabel 2.1. Contoh Priority Scheduling

Process	Duration	Priority	Arrival Time
---------	----------	----------	--------------

P1	6	4	0
P2	8	1	0
P3	7	3	0
P4	3	2	0

Kelemahan pada *priority scheduling* adalah dapat terjadinya *indefinite blocking (starvation)*. Yaitu proses dengan prioritas rendah berkemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi darinya.

Solusi dari permasalahan ini adalah *aging*, yaitu meningkatkan prioritas dari setiap *task* yang menunggu dalam queue secara bertahap. Contoh: Setiap 10 menit, prioritas dari masing-masing *task* yang menunggu dalam *queue* dinaikkan satu tingkat. Maka, suatu *task* yang memiliki prioritas 127, setidaknya dalam 21 jam 20 menit, *task* tersebut akan memiliki prioritas 0, yaitu prioritas yang tertinggi (semakin kecil angka menunjukkan bahwa prioritasnya semakin tinggi).

2.3 Unified Modeling (UML)

Menurut (Fadallah and S Rosyida 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.

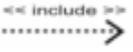
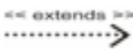
Menurut (Dwinda and Henderi 2018) *Unified Modelling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

Pada perkembangan teknik pemograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang di bangun dengan menggunakan teknik pemograman berorientasi objek yaitu Unified Modeling Language (UML) dalam UML terdiri dari 13 macam diagram yang dikelompokan dalam 4 kategori sebagai berikut :

2.3.1. 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 2.2 Simbol dan Keterangan Use case Diagram

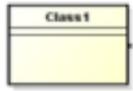
SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun
	<i>Use case</i>	Menggambarkan bagaimana seseorang menggunakan system
	<i>Relasi asosiasi</i>	Relasi yang dipakai untuk menunjukkan hubungan antara <i>actor</i> dan <i>use case</i>
	<i>Relasi 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)

2.3.2. 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 2.3 Simbol dan Keterangan Class Diagram

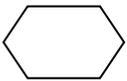
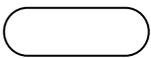
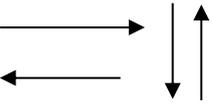
SIMBOL	NAMA	KETERANGAN
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Generalization</i>	Garis yang melambangkan konsep pewarisan dari satu kelas ke satu atau lebih sub kelas
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

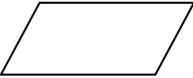
Sumber : (Nugroho, 2010)

2.3.3. 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 2.4 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

2.4 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 2.5 Penelitian Terkait

No	Judul, Penulis, Tahun	Jumlah dan Atribut	Algoritma/Metode
1.	Pengembangan Algoritma Penjadwalan Dan Maintenance Mesin (Dewi and Endah 2020)	Ada 2 atribut : <i>Penjadwalan job, slack time</i>	Penjadwalan Produksi
2.	Penerapan Algoritma Dynamic Priority Scheduling Pada Aplikasi Antrian Pencucian Mobil Berbasis Mobile (Setyawati and Maulachel 2020)	Ada 3 atribut : <i>Dynamic Priority Scheduling Algorithm, Queue, Mobile Application</i>	Penjadwalan Dinamis Berprioritas
3.	Rancang Bangun Sistem Informasi Inventory Fasilitas Maintenance Pada Pt. Pln (Persero) Tangerang	Ada 3 Atribut : <i>Information Systems, Inventory, Maintenance facility.</i>	Priority Schedule

No	Judul, Penulis, Tahun	Jumlah dan Atribut	Algoritma/Metode
	(Maulani, Septiani, and Noer Fauziyah Sahara 2018)		
4.	Sistem Penjadwalan Iklan Menggunakan Metode Priority Scheduling pada PT. Kidung Indah Selaras Suara (Radio Kiss FM) untuk Efektivitas dan Efisiensi Produksi Siaran (Simarmata and Harahap 2019)	Ada 3 Atribut : <i>scheduling,</i> <i>algorithm,</i> <i>priority scheduling</i>	Priority Schedule
5.	Penjadwalan Mesin Menggunakan Algoritma Non Delay untuk Mereduksi Mean Tardiness pada Lingkungan Batch Production (Nur'aeni, Muhammad, and Amaranti 2022)	Ada 3 atribut : <i>Scheduling,</i> <i>Non Delay</i> <i>Algorithm,</i> <i>Mean Tardiness.</i>	Non Delay