

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa penelitian yang dilakukan sebelumnya. Beberapa uraian penelitian sebelumnya yang menjadi acuan dalam penelitian ini, yaitu:

1. Dwi Fikri Haikal, Ahmad Bagus Setiawandan Danar Putra Pamungkas (2021) Dengan judul “Sistem Pemesanan *E-Cafe* Berbasis Android Dengan Menggunakan Metode FIFO’ Penelitian ini menggunakan metode fifo karena dapat kelebihan seperti nilai biaya persediaan disajikan secara *actual* dan *relevan* dan memungkinkan perusahaan untuk melakukan perbandingan yang lebih baik dalam beberapa waktu sehingga perusahaan dapat menghindari beberapa factor kerugian seperti menghindari barang agar tidak kadaluarsa dan tertimbun lama , dan untuk menyajikan persediaan akhir di dalam neraca berdasarkan harga yang paling akhir.
2. Sarumpaet, Fransisca Agustina OK (2020) Dengan judul “Penerapan Metode *First In First Out* (FIFO) Dalam Pembangunan Aplikasi Pemesanan Pakaian Pada CV. Rantau Bayur Permai Berbasis *Web*’ Penelitian ini menghasilkan bahwa menggunakan aplikasi dengan menggunakan metode Fifo dapat mempermudah pemesanan menjadi instan tidak manual seperti biasa dan

segala produk yang ada mempunyai data secara lengkap di dalam suatu aplikasi.

3. Ehtur Enjelita Gultom dan Dwi Oktarina (2019) Dengan judul “Rancang Bangun Sistem Informasi Pemesanan Antrian Service Mobil Berbasis Android’ Penelitian ini menggunakan metode Fifo dalam penyelesaian segala masalah di dalam rancangan aplikasi nya sebagai contoh seperti dalam sistem informasi antrian pada aplikasi gamma ban menggunakan metode fifo yang dimana mobil yang antri terlebih dahulu akan diproses terlebih dahulu. Dan dengan membangun sistem informasi bengkel gamma ban terkomputerisasi terbukti dapat meningkatkan pelayanan dan kegiatan transaksi dan membuat semuanya mejadi cepat.
4. Dede kurniawan (2017) Dengan judul “Aplikasi Pemesanan Mobil & Sparepart Pada PT.Surya Batara Mahkota (SBM) Berbasis Web” Berdasarkan hasil penelitian maka dapat disimpulkan bahwa Dengan dibuatnya aplikasi ini, PT. Surya Batara Mahkota mudah mempromosikan dan memasarkan secara online untuk bisa memasarkan produknya dan juga Masyarakat dapat melihat terlebih dahulu jenis Mobil ataupun Spare Part yang dbutuhkan konsumen tanpa harus melihat datang langsung ke lokasi perusahaan.
5. Arif Rahmad Hidayat, Volvo Sihombing, Deci Irmayani, (2021) Dengan judul “Sistem Informasi Pemesanan Tiket Bus Pada Chandra Bagan Batu Berbasis Web” Berdasarkan hasil dari penelitian ini maka dapat disimpulkan bahwa dengan aplikasi yang dibuat ini pemesanan tiket bus lebih efektif dan lebih

mudah sehingga dapat menciptakan efektifitas dan efisisensi dalam hal pemasaran dan pelayanan kepada pelanggan.

II.2. Landasan Teoritis

II.2.1. Metode Frist In First Out (Fifo)

Metode Fifo ini adalah persediaan barang masuk pertama, maka akan keluar barang pertama, sehingga untuk mendapatkan persediaan diakhir dinilai dengan perolehan yang terakhir dibeli atau masuk. (Nur Rohma,2018)

FIFO merupakan singkatan dari bahasa Inggris, *First In First Out*, yang secara harfiah dapat diartikan sebagai barang yang masuk pertama kali adalah barang yang akan pertama kali dikeluarkan. Dalam bisnis, istilah ini dapat diterapkan sesuai dengan arti harafiahnya, tetapi diberlakukan dengan konteks bisnis. Dengan kata lain, setiap barang atau komoditas bisnis yang masuk maupun keluar akan dikelola sesuai dengan urutan waktunya.

Barang yang pertama kali masuk gudang adalah barang yang diutamakan untuk dikeluarkan terlebih dahulu dari gudang, atau untuk skala yang lebih luas, stok barang yang pertama kali diterima adalah prioritas yang perlu dijual atau dipasarkan terlebih dahulu dibanding stok barang yang masuk belakangan.

Fifo (*First in First Out*) merupakan sebuah metode yang dimana sebuah barang pertama kali masuk harus juga pertama kali yang dikeluarkan atau dijual. Jadi, pencatatan persediaan yang terdapat di dalam laporan akan serupa dengan stok yang ada di dalam gudang.

Metode FIFO salah satu metode yang banyak digunakan oleh banyak pelaku usaha untuk memastikan pengelolaan stok opname maupun komoditas bisnis yang ada di gudang tetap baik secara kualitas dan juga tetap mematuhi hukum.

II.2.2. Pemesanan

Pemesanan memiliki arti dalam kelas nomina atau kata benda sehingga pemesanan dapat menyatakan nama dari seseorang, tempat, atau semua benda dan segala yang dibendakan.

Pemesanan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik. Tujuan pemesanan yaitu :

1. Memaksimalkan pelayanan bagi konsumen
2. Meminimumkan investasi pada persediaan
3. Perencanaan kapasitas
4. Pengesahan produksi dan pengendalian produksi
5. Persediaan dan kapasitas
6. Penyimpanan dan pergerakan material

Pemesanan adalah suatu perjanjian yang dilakukan oleh 2 (dua) pihak atau lebih yaitu pemberi dan pemakai jasa atau barang untuk memenuhi kebutuhannya dalam mengusahakan barang atau jasa tersebut sehingga dapat digunakan. (Susanti, 2017:2)

II.2.3. Sparepart

Sparepart adalah komponen yang dapat dirakit menjadi kesatuan dan memiliki fungsinya masing-masing. Satu sparepart dapat mempengaruhi kerja sparepart lainnya. Dalam satu mobil saja, ada ratusan sparepart penyusunnya. (Nafila Chaerunnisa, 2022)

Spare Part merupakan sebuah barang yang berisikan berbagai komponen dalam suatu kesatuan dan memiliki fungsi tertentu. Spare part banyak dipakai di berbagai jenis kendaraan, sehingga jenis-jenisnya juga sangat beragam.

Spare part pasti memiliki kegunaan tersendiri, namun ada yang sistem kerjanya saling terkait dengan spare part lain. Setidaknya ada beberapa jenis spare part yang sangat penting untuk diketahui dan dipelajari sistem kerjanya.

Ada beberapa komponen yang juga terdapat didalamnya beberapa komponen kecil, misalkan *engine* yang mempunyai komponen didalamnya yaitu *fuel injection pump*, *water pump*, *starting motor*, *alternator*, *oil pump*, *compressor*, *power steering pump*, *turbo charger*, dan lain-lain.

II.2.4. PHP

PHP adalah singkatan dari Hypertext Preprocessor, yaitu bahasa pemrograman yang sebenarnya mirip dengan Java Script dan Python. Perbedaannya adalah, PHP sering kali digunakan untuk komunikasi sisi server, sedangkan Java Script bisa digunakan untuk frontend dan backend. Sementara itu, Python hanya untuk sisi server.

Dalam pengembangan website, PHP menjadi salah satu bahasa pemrograman yang wajib dipelajari. Alasannya, bahasa pemrograman ini mampu untuk membuat website menjadi dinamis.

Nah, dalam artikel ini, Anda akan belajar tentang PHP secara lebih dalam. Kami akan membahas mengenai syntax dasar PHP beserta contoh-contohnya. Selain itu, kami juga akan membahas tentang hubungan PHP dengan MySQL.

Hypertext Preprocessor merupakan Bahasa scripting yang sangat diketahui sering dipakai dalam pengembangan web walaupun sebenarnya tidak hanya itu kegunaannya. Berdasarkan *w3techs.com* PHP merupakan bahasa scripting yang sering dipakai di internet sebesar 82%. (Das dan Saikia, 2016).

II.2.5. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Sukanto dan Shalahuddin, 2018:13).

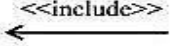
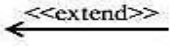
Mengetahui pengertian atau definisi UML sesungguhnya tidak terlalu penting, barangkali lebih penting mengenal jenis-jenis UML, sejarah dan cara membuatnya. Meskipun demikian, tidak ada salahnya kita untuk mencari tahu mengenai definisi atau pengertian UML, mungkin saja suatu saat ada yang bertanya akan hal itu kepada kita.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

a. *Use Case Diagram*

Use case atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang digunakan di dalam *Use Case* diagram yaitu :

Tabel II.1. *Use Case Diagram*

	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

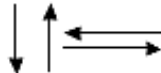
Sumber : Heriyanto Yunahar, (2018)

b. *Activity diagram*

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Activity diagram menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang digunakan di dalam *Activity diagram* yaitu :

Tabel II.2. Activity Diagram

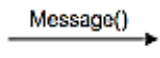
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Sumber : Heriyanto Yunahar, (2018)

c. *Sequence* diagram

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan di dalam *Sequence* diagram yaitu :

Tabel II.3. *Sequence Diagram*

1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
		<i>Actor</i>	Digunakan untuk menggambarkan user / pengguna.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form.
4		<i>Control Class</i>	Digunakan untuk menghubungkan boundary dengan tabel.
5		<i>Entity Clas</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

Sumber : Heriyanto Yunahar, (2018)

d. *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 oprasi. Atribut merupakan variable-variable yang dimiliki oleh suatu kelas. Oprasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Simbol-simbol yang digunakan di dalam *Class diagram* yaitu :

Tabel II.4. *Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih

1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

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>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

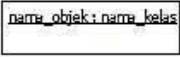
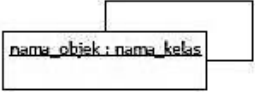
Sumber : Heriyanto Yunahar, (2018)

e. *Diagram Collaboration*

Diagram Collaboration juga merupakan diagram *Interaction* berfungsi membawa informasi yang sama dengan diagram *Sequence*, tetapi lebih memuaskan atau memfokuskan pada kegiatan obyek dari waktu informasi itu dikirimkan.

Simbol yang digunakan dalam *Diagram Collaboration* adalah sebagai berikut :

Tabel II.5. Collaboration Diagram

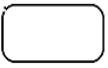
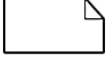
Objek 	objek yang melakukan interaksi pesan
Link 	relasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri 

Sumber : Heriyanto Yunahar, (2018)

f. *Diagram State Chart*

Menurut *State chart diagram* adalah diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem objek. Simbol yang digunakan dalam *state chart diagram* adalah sebagai berikut :

Tabel II.6. State Chart Diagram

1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

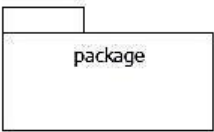
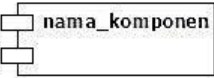
Sumber : Heriyanto Yunahar, (2018)

g. *Diagram Component*

Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem.

Simbol yang digunakan dalam *Diagram Component* adalah sebagai berikut :

Tabel II.7. *Component Diagram*

Package 	package merupakan sebuah bungkusan dari satu atau lebih komponen
Komponen 	Komponen sistem
Kebergantungan / dependency 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antarmuka / interface	sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen

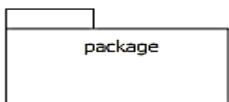
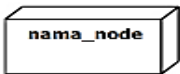
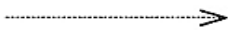
Sumber : Heriyanto Yunahar, (2018)

h. *Deployment Diagram*

Diagram deployment atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

Simbol yang digunakan dalam *Deployment Diagram* adalah sebagai berikut :

Tabel II.8. *Deployment Diagram*

Package 	package merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
Node 	biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan / <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
Link 	relasi antar <i>node</i>

Sumber : Heriyanto Yunahar, (2018)