

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Venus, dkk (2020) dengan judul Rancang Bangun Aplikasi Client – Server Control Slide Presentation Berbasis Desktop Dan Android, menyimpulkan Desain sistem secara global menggunakan bahasa pemodelan UML yang digunakan dalam pembuatan aplikasi Client - Server Control Slide Presentation Berbasis Desktop dan Android terdiri dari diagram use case, diagram activity, dan diagram sequence.

Berdasarkan penelitian yang dilakukan oleh Muzawi Rometdo, dkk (2017) dengan judul Perancangan Aplikasi Berbasis *Client Server* dalam Meng-upload *File-File* Ujian pada Laboratorium Komputer STMIK Amik Riau, menyimpulkan bahwa Keefisienan dari segi waktu karena mahasiswa dapat mengirimkan secara langsung *file* ujian yang diberikan oleh Dosen dan dapat mengetahui proses peng-upload dan yang telah dilakukan berhasil atau tidaknya.

Berdasarkan penelitian yang dilakukan oleh Heryanto Ahmad, dkk (2017) dengan judul Sistem *Monitoring Server* dan Perangkat jaringan pada *Enterprise*

Resource Planning Fasilkom UNSRI Menggunakan Protokol ICMP dan SNMP, menyimpulkan bahwa sistem Aplikasi *monitoring* mampu membuat *report* terhadap kondisi perangkat jaringan Fakultas Ilmu Komputer diharapkan yang dapat memberikan kemudahan bagi *admin* dalam mengelola data.

Berdasarkan penelitian yang dilakukan oleh Putra, dkk (2016) dengan judul Komputerisasi Sistem Analisis Data Mahasiswa Berbasis *Client - Server*, menyimpulkan bahwa Pemodelan Sistem Analisis Data Mahasiswa dalam meningkatkan kualitas pelayanan dapat dilakukan perancangannya dengan berbasis *Client-Server* dan juga analisis data dapat dilakukan dengan lebih cepat dan fleksibel kemudian data dapat disimpan dengan lebih aman dan terstruktur.

Berdasarkan penelitian yang dilakukan oleh Junaid, dkk (2019) dengan Judul Sistem Informasi Penggajian Karyawan Berbasis *Client - Server* Pada PT. CMPP (Citra Mandiri Persada Pratama), menyimpulkan bahwa Dengan Adanya sistem informasi penggajian yang dirancang berbasis *Client- Server* telah berjalan dengan sempurna, bisa mempercepat dalam proses pendataan karyawan, jabatan, dan juga absensi karyawan yang dilakukan oleh *admin*.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang di ambil dari beberapa jurnal :

II.2.1. Perancangan

Perancangan adalah proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik. Tujuan perancangan yaitu untuk memenuhi kebutuhan pemakaian sistem (*user*) dan memberikan gambaran yang jelas dan menghasilkan rancangan bangun yang lengkap kepada pemograman komputer dan ahli-ahli teknik lainnya yang terlibat dalam pengembangan atau pembuatan sistem. (Adiyanti, dkk, 2021 : 11).

II.2.2. Aplikasi

Aplikasi merupakan suatu program komputer yang dapat digunakan pada berbagai perangkat elektronik, terutama pada komputer/laptop atau telepon seluler yang berbasis *android*. (Efendi, dkk, 2020 : 341).

Komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. (Nugraha, dkk, 2020 : 138).

II.2.3. Monitoring

Arti dari *monitoring* sendiri merupakan kegiatan yang terdiri dari proses mengamati atau meninjau dan mempelajari secara berkala yang dilakukan oleh pengelola proyek di tiap pelaksanaan kegiatan aktivitas proyek untuk memastikan bahwa hasil yang diperoleh pada hari itu sesuai dengan keinginan. Tidak hanya itu saja, proses *monitoring* juga terkadang perlu menelaah pengadaan dan penggunaan *input*, jadwal kerja, laporan perkembangan *progres*, pengaturan jam kerja SDM untuk menyelesaikan proyek, dan lain-lain. (Pratama, 2017 : 17).

II.2.4. PT. Pelayaran Laksita Aditya Parama

Kantor Cabang PT. Pelayaran Laksita Aditya Parama (PLAP) yang beralamat di Jln. Sumatera No. 28 Kel. Belawan II, Kec. Medan Belawan, 20412 berdiri dan ditetapkan di Jakarta pada tanggal 20 Juni 2017 oleh Menteri Hukum dan Hak Asasi Manusia Republik Indonesia dalam Surat Keputusan (SK) Nomor : AHU-076.AH.02.02-Tahun 2012 tanggal 12 September 2012. Dalam Surat Keputusan ini tercantum Perusahaan Terbatas ini bernama PT. Pelayaran Laksita

Aditya Parama (cukup disingkat dengan Persero) berkedudukan di Jakarta Utara. Persero ini pun membuka kantor cabang atau kantor perwakilan, baik itu didalam maupun diluar wilayah Republik Indonesia sebagaimana ditetapkan oleh Direksi dengan persetujuan Komisaris Utama atau salah seorang Komisaris. Dan jangka waktu didirikannya Perseroan ini tidak terbatas.

Adapun maksud dan tujuan Perseroan ini ialah menjalankan usaha dalam bidang pelayaran dan untuk mencapai maksud dan tujuan tersebut, Perseroan dapat melaksanakan kegiatan usaha sebagai berikut :

1. Berusaha dalam bidang pelayaran termasuk tetapi tidak terbatas pada pelayaran lokal, interinsuler, angkutan penyeberangan, maupun pelayaran internasional serta dapat mempunyai hubungan kerja dari agen per agen untuk perusahaan sejenis.
2. Menjalankan usaha dalam bidang angkutan laut untuk keperluan pengangkutan / pengiriman barang orang dan surat – surat / paket – paket dalam arti kata seluas – luasnya termasuk melakukan usaha gelangan kapal, reparasi dan perbaikan kapal pelayaran umum/khusus antar pulau atau antar samudera.
3. Menjalankan usaha sebagai agen kapal perusahaan pelayaran lain termasuk perantara dan mengusahakan atau mencari muatan kapal.
4. Menjalankan kegiatan usaha sewa-menyewa kapal, mencharter atau menyewakan kapal kepunyaan pihak lain.

II.2.5. Logo Perusahaan

Pada umumnya setiap perusahaan pasti memiliki logo, karena logo merupakan identitas dari perusahaan tersebut. Berikut ini adalah logo yang dimiliki oleh PT. Pelayaran Laksita Aditya Parama yang ditunjukkan pada gambar II.1.



PT. PELAYARAN LAKSITA ADITYA PARAMA

Gambar II.1. Logo PT. Pelayaran Laksita Aditya Parama
(Sumber: PT. KLAP)

II.2.6. Web

Web merupakan kumpulan halaman *web* yang saling berkaitan. *Web* adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video dan dapat diakses melalui perangkat yang biasa disebut *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019 : 151).

II.2.7. Database MySQL

Database MySQL merupakan salah satu DBMS yang familiar digunakan dengan total 6 juta instalasi di dunia. Kelebihan *MySQL* diantaranya bisa digunakan pada semua sistem operasi seperti Linux, Mac OS X, Open BSD dan Windows secara gratis. *MySQL* juga bisa digunakan pada beberapa bahasa pemrograman, diantaranya yaitu bahasa pemrograman C, C++, Java, Perl, PHP, dan Ruby. (Qitvirul,, dkk, 2020 : 17).

II.2.8. XAMPP

XAMPP merupakan alat bantu yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* dan *MySQL* secara manual. *XAMPP* akan menginstalasi dan mengkonfigurasikannya secara otomatis atau *auto* konfigurasi. *XAMPP* merupakan paket *PHP* yang berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan *XAMPP* tidak dibingungkan dengan penginstalan program-program lain, karena semua kebutuhan telah tersedia oleh *XAMPP*. Yang terdapat pada *XAMPP* di antaranya : *Apache*, *MySQL*, *PHP*, *FilZilla FTP Server*, *PHPmyAdmin* dll. (Santoso, 2017 : 86).

II.2.9. Hyper Text Markup Language (HTML)

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. HTML

bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word. (Sugijanto, dkk, 2020 : 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui tag-tag yang ditulis dalam HTML. (Rahmasari, 2019 : 415).

II.2.10. Hypertext Preprocessor (PHP)

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019 : 415).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan *script* yang

digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script*/program tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020 : 2).

II.2.11. Unified Modeling Language (UML)

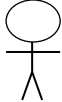
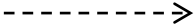



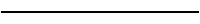




UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

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

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem *use case* diagram dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol Use Case

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<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>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi






(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang

mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II. 2. Simbol Activity Diagram

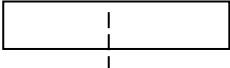
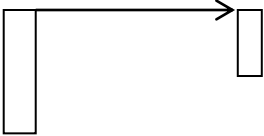
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran


(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.



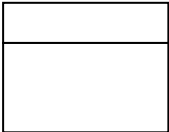
Gambar	Nama	Keterangan
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.


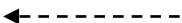
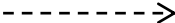

(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<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>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

Gambar	Nama	Keterangan
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Andikos, 2019 : 39)