

BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Bimbingan

Bimbingan memiliki arti untuk menunjukkan sebuah petunjuk, menuntut, mengatur, mengarahkan serta memberikan nasehat yang berarti bimbingan bermakna sebuah proses dalam memberikan bantuan atau pertolongan kepada seseorang dimana dengan tujuan agar orang tersebut mendapat pencerahan terkait tentang dirinya, lingkungan dan dapat memilih jalan yang akan diambilnya dalam mengatasi masalah yang dihadapinya [13].

Bimbingan memiliki metode yang berupa konseling dimana konseling berarti sebuah nasihat, anjuran atau pembicaraan. Konseling memiliki makna yang berarti sebagai sebuah bantuan yang diberikan kepada individu untuk memecahkan masalah yang dihadapinya dengan cara wawancara atau berkomunikasi *face to face* yang disesuaikan dengan kondisi yang terjadi dengan tujuan agar individu dapat mengambil keputusan dan memiliki tanggung jawab terhadap apa yang dipilih sebagai pemecah masalah tersebut [13].

Dalam sebuah universitas, bimbingan adalah sebuah usaha yang dilakukan untuk membantu mahasiswa agar mendapat sebuah pemahaman, pengarahan terkait dirinya dalam mencapai tingkat perkembangan yang optimal dan dapat menyesuaikan diri dengan lingkungan atau permasalahan yang dihadapinya [13].

2.1.2 Database

Database merupakan salah satu media yang dapat digunakan untuk menyimpan kumpulan data maupun file yang dapat diakses oleh pemilik maupun pihak dengan privilege atau otoritas atas akses dan/atau modifikasi berbagai data tersebut. Untuk memenuhi kebutuhan pengguna, saat ini

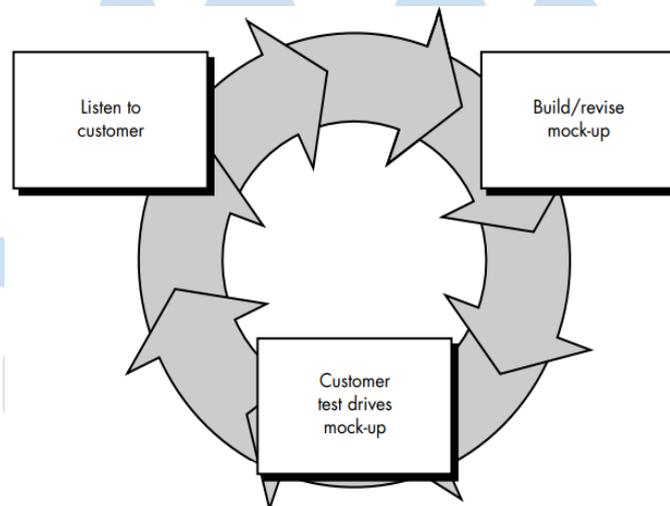
semakin berkembang jenis dari database yang mampu menyimpan jenis dan format data yang lebih kompleks dan beragam untuk fungsi-fungsi tertentu [14].

Pengguna dapat mengakses, mengatur, dan memodifikasi *database* dengan menggunakan sistem yaitu Database Management System (DBMS). DBMS juga dapat mengatur kebutuhan terkait pengaksesan, keamanan dan penyimpanan serta juga dapat menjadi *host* sebagai fungsi lain dari sistem *database* yang digunakan [15].

2.1.3 Metode Prototyping

Metode *prototyping* adalah teknik yang digunakan dalam pengembangan sistem secara terstruktur yang dibentuk sebagai perantara *developer* dan pengguna terkait gambaran dari interaksi sistem yang sudah dirancang sesuai kebutuhan proses bisnis dengan pelaku usaha (pengguna) [16].

Metode *prototyping* ini juga digunakan untuk tempat evaluasi pada sistem apabila terjadinya kekurangan atau belum sesuai dengan kebutuhan pengguna sebelum akhirnya dilakukan tahap *finishing* pada sistem [17]. Gambar 2.1 merupakan tahap-tahap dengan metode *prototyping* dalam membangun dan merancang sistem aplikasi berbasis *web*.



Gambar 2.1 Metode Pengembangan *Prototyping*

Sumber : [40]

Seperti pada gambar 2.1, model *prototyping* diawali dengan berdiskusi dengan klien terkait apa saja yang dibutuhkan dan dibutuhkan dalam pengembangan aplikasi, kemudian dimulainya untuk membangun *prototype* sebagai gambaran dari alur sistem dan tampilan dari aplikasi yang dikembangkan. Setelah itu, klien menguji coba *prototype* yang diberikan agar dapat mengetahui apakah tampilan, fungsi dan alur kerja sistem dari aplikasi yang dibangun sudah sesuai dengan kebutuhan atau belum. Hal ini berulang sampai alur kerja sistem dan tampilan sudah sesuai dengan kebutuhan klien.

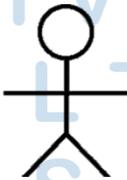
2.1.4 Unified Modelling Language (UML)

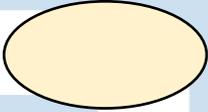
Unified Modelling Language (UML) merupakan teknik pemodelan untuk membangun, merancang, dan mengembangkan sistem yang dapat melakukan interaksi dengan manusia dimulai dari cara kerja sistem berdasarkan kegiatan bisnis dan penggunaan sistem melalui fitur-fitur yang dibangun [18].

2.1.4.1 Use Case Diagram

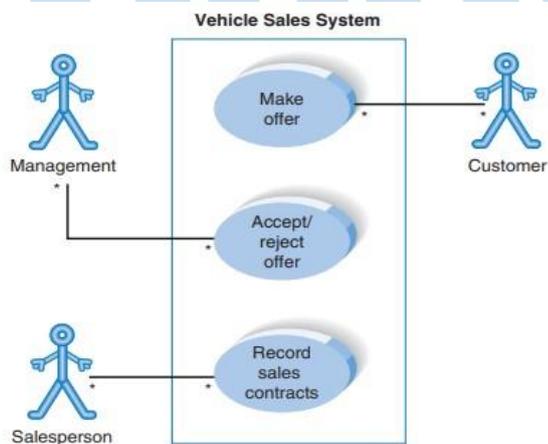
Use Case Diagram adalah teknik dari pemodelan UML berpandangan objek yang digunakan untuk mengembangkan sistem informasi untuk kebutuhan sistem yang dibangun. *Use case* memberikan gambaran interaksi yang terjadi antara pengguna dengan sistem yang dibuat [18]. Berikut pada Tabel 2.1 adalah simbol yang terdapat dalam *Use Case Diagram* [19]:

Tabel 2.1 *Use Case Diagram*

Simbol	Keterangan
	<p>Aktor :</p> <p>Kegiatan sistem atau orang yang berinteraksi dengan sistem informasi yang pembuatannya berada di luar sistem. Aktor digambarkan dalam bentuk <i>stick figure</i> secara <i>defaultnya</i> namun apabila aktor bukan manusia dapat digambarkan dengan persegi</p>

	panjang dengan <<actor>> di dalam sebagai alternatifnya.
Simbol	Keterangan
	<i>Use Case</i> : Menjadikan sistem sebagai unit-unit yang digunakan untuk bertukar pesan antar aktor atau unit. Selain itu diletakkan dalam batas sistem yang dapat menyertakan kasus penggunaan lain dengan pemberian label kata benda atau deskriptif.
	<i>Subject Boundary</i> : Merupakan nama subjek yang berada di atas atau di bawah yang dapat mengartikan sebagai ruang lingkup subjek seperti contoh adanya sistem atau individu proses bisnis.
	<i>Association Relationship</i> : Komunikasi yang menghasilkan interaksi yang terjadi antara aktor dengan <i>use case</i> yang berhubungan dengannya.

Komponen dari *usecase* diagram yaitu: *system boundary* yang menggambarkan *scope* sistem dan mengenkapsulasikan bagian komplet dan fungsionalitas sistem; *actor* yaitu pengguna yang berinteraksi dengan sistem dan sebuah *use case* yang dapat berupa manusia, organisasi, atau sistem dan dapat memberikan pengaruh pada fungsionalitas sistem; dan *use case* yang merupakan gambaran visual dari fungsionalitas yang berbeda-beda dalam sistem. Pada gambar 2.2 adalah contoh dari *use case diagram*:



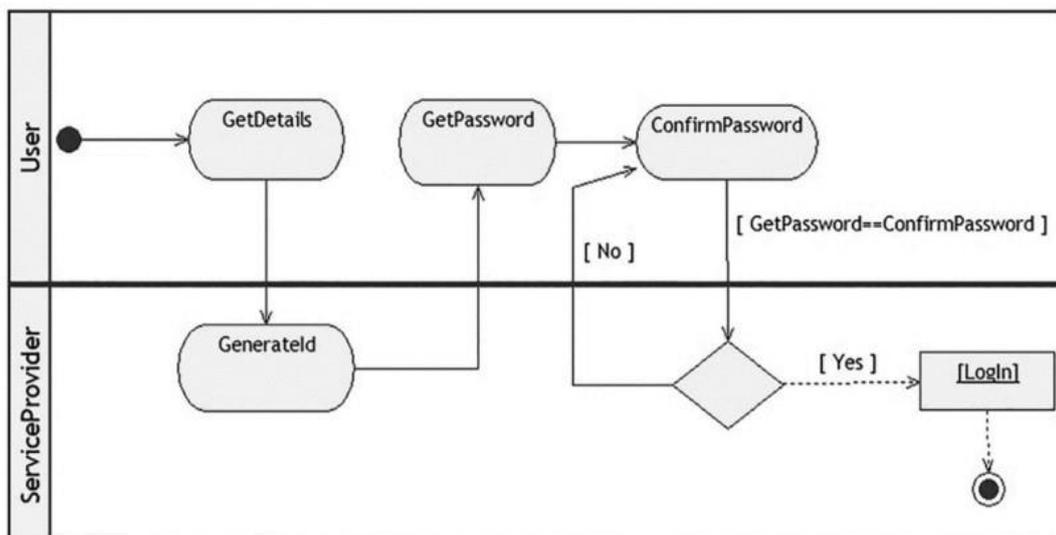
Gambar 2. 2 Contoh *Use Case Diagram*

Sumber: [19]

2.1.4.2 Activity Diagram

Activity Diagram adalah teknik yang digunakan untuk menggambarkan aktivitas-aktivitas yang terjadi berdasarkan proses bisnis yang dirancang dalam sistem. *Activity diagram* ini dirancang berdasarkan dari *use case* atau beberapa *use case* yang dibuat di *use case diagram* [20].

Komponen *activity diagram* antara lain: *initial state* yang menggambarkan kondisi awal sistem yang dipertimbangkan; *final state* yaitu kondisi yang mengakhiri sistem yang dipertimbangkan; *swimlanes* yaitu partisi yang terdiri dari *entity* dan kumpulan aktivitas yang terkait; *action state* yaitu sebuah operasi, aktivitas bisnis, atau proses; *object* yaitu *entity* yang membawa data antar dua *action state*; *decision* merupakan *node* dengan 1 atau 2 atau input dan banyak *output* yang bergantung pada kondisi yang dirancang; dan *transition* yaitu panah yang menggambarkan pergerakan dari *activity state* sumber ke *activity state target* yang dipicu oleh penyelesaian aktivitas dari *activity state* sumber. Berikut adalah contoh dari *activity diagram* [21]:



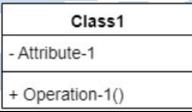
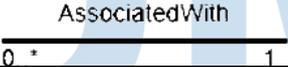
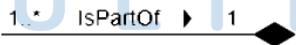
Gambar 2. 3 Contoh Activity Diagram

Sumber: [21]

2.1.4.3 Class Diagram

Class Diagram adalah teknik yang menggambarkan struktur dan hubungan antar *class* dalam sebuah sistem dan bagaimana *class* tersebut saling berhubungan satu dengan yang lain dalam berjalannya sistem. *Class diagram* ini setiap kelasnya memberikan gambaran berupa perilaku dan keadaan [22]. Berikut pada Tabel 2.3 adalah simbol yang terdapat dalam *Class diagram* [23]:

Tabel 2.2 *Class Diagram*

Simbol	Keterangan
	<p><i>Class:</i> Didefinisikan sebagai orang, tempat atau hal yang berkaitan dengan sistem</p>
	<p><i>Operation:</i> fungsionalitas yang dilakukan <i>class</i> dengan klasifikasi sebagai <i>query</i>, <i>constructor</i> atau <i>update operation</i></p>
	<p><i>Attribute:</i> Digambarkan sebagai kondisi dari suatu objek serta objek ini diturunkan dari atribut lain</p>
	<p><i>Generalization:</i> Didefinisikan sebagai hubungan beberapa kelas</p>
	<p><i>Association:</i> Didefinisikan sebagai penggabungan antara dua kelas atau di kelas itu sendiri dengan ditandai adanya peran yang menggambarkan hubungan antar <i>class</i></p>
	<p><i>Aggregation:</i> Mendefinisikan sebuah hubungan yang terjadi antara sebuah kelas atau beberapa kelas dengan dirinya sendiri secara logis</p>
	<p><i>Composition:</i> Mendefinisikan hubungan antara sebuah kelas atau beberapa kelas dengan dirinya sendiri secara langsung</p>

2.1.5 MySQL

Database yang bersifat *open source* yang memiliki tingkat fleksibilitas dalam teknologi saat ini sehingga lebih mudah mengintegrasikan dengan berbagai bahasa pemrograman seperti C++, Java, PHP dan lainnya. MySQL ini mendukung adanya penggunaan *multiuser* yang menjadikan dapat digunakan dalam waktu yang bersamaan tanpa adanya efek samping terjadinya *crash*. Disisi lain, MySQL ini memberikan keamanan dimana dalam pengaturan hak akses *user* untuk mengelola *database* sudah dienkripsi dengan *password* tingkat tinggi yang tentu mencegah terjadinya adanya pengaksesan *database* milik *user* terakses orang lain [24].

2.1.6 Hypertext Markup Language (HTML)

HTML merupakan bahasa yang digunakan untuk membuat struktur dari sebuah halaman *website*. Dimana penggunaan HTML ini memberikan kemudahan dalam membuat atau menyusun *heading*, paragraf, gambar, *link* dan lain-lain dalam halaman *website*. Dalam membuat *file* HTML dapat menggunakan aplikasi editor HTML dan menyimpannya dalam format *.html* atau *.htm* [25].

2.1.7 Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman yang bersifat *open source* yang digunakan untuk pembuatan aplikasi berbasis *web* secara dinamis dan dapat diintegrasikan dengan *database* [26].

2.1.8 Cascading Style Sheets (CSS)

CSS adalah bahasa dalam pemrograman untuk menata dan menampilkan elemen-elemen yang terdapat dalam bahasa *markup*, salah satunya yaitu HTML. CSS juga memiliki fungsi dalam mengubah dan mendesain tampilan halaman *website* [27].

2.1.9 Bootstrap

Bootstrap adalah salah satu jenis *library framework* CSS yang bersifat *open-source* dalam membuat dan merancang sebuah halaman *website*. dimana

terdapat HTML, CSS dan komponen *interface* lainnya yang menghasilkan tampilan *front-end website* yang responsif, *modern* dan *friendly user*. Hal ini memberikan kemudahan untuk para pengembang untuk merancang tampilan *front-end website* lebih mudah, praktis, menghemat waktu dan tentu sesuai dengan kebutuhan dari pembuatan *website* itu sendiri dengan *bootstrap* ini [28].

2.2 Penelitian Terdahulu

Pada tabel berikut adalah penelitian-penelitian yang berkaitan dengan sistem pencatatan dan penyimpanan yang dibangun berbasiskan *web*:

Tabel 2.3 Penelitian Terdahulu

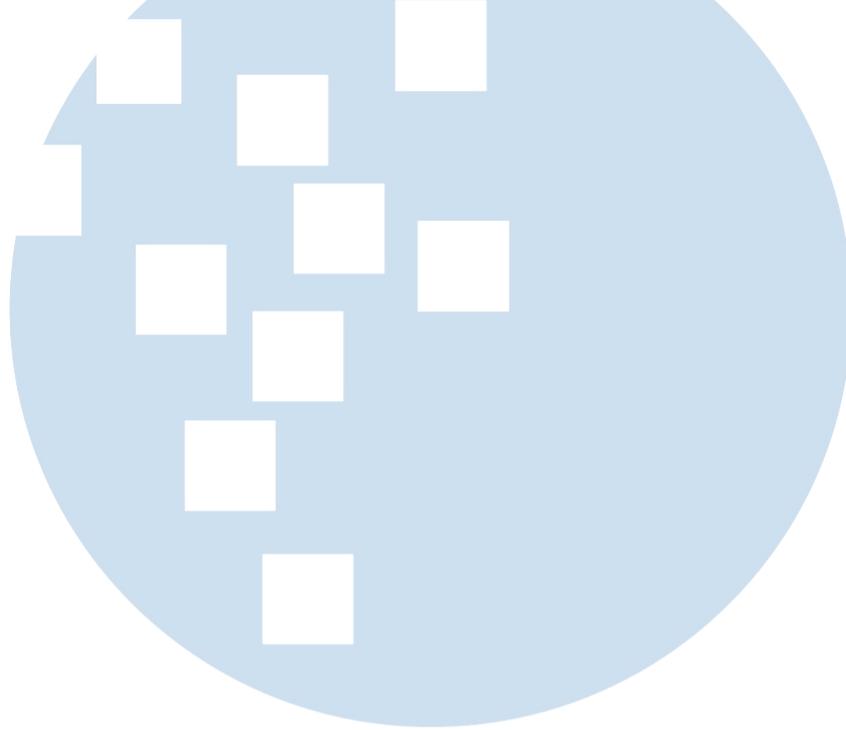
1	Nama Jurnal :	Jurnal Sains dan Informatika, Volume 4, Nomor 2
	Judul Artikel :	Perancangan Sistem Informasi Absensi Dosen Dengan Validasi Mahasiswa Berbasis Web
	Penulis Artikel (Tahun) :	Rina Firliana, Fatkur Rhohman, dan Rolistian Willy Purwinanto (2018)
	Metode pengembangan yang digunakan :	Metode pengembangan dalam pembangunan dan perancangan sistem ini yaitu metode <i>waterfall</i> .
	Kesimpulan :	Pembangunan aplikasi berbasis <i>web</i> untuk pemantauan absensi mahasiswa dan dosen yang mengajar dalam suatu mata kuliah dengan validasi dari Penanggung Jawab Mata Kuliah (PJM) yaitu mahasiswa yang diajar oleh dosen pada hari tersebut dan sistem dapat melakukan pembuatan laporan rekap terhadap absensi dosen serta rekap laporan mahasiswa dapat menjadi acuan dosen untuk menilai kehadiran dari setiap mahasiswa.
2	Nama Jurnal :	Jurnal Sigmata Volume : 9 Nomor : 1 Edisi : Oktober 2020 - Maret 2021
	Judul Artikel :	Sistem Informasi Absensi Pegawai Berbasis Web Pada Kantor Kelurahan Sako Palembang
	Penulis Artikel (Tahun) :	Ruslan (2021)
	Metode pengembangan yang digunakan :	Metode pengembangan dalam pembangunan dan perancangan sistem ini yaitu metode <i>waterfall</i> .

	Kesimpulan :	<p>Sistem absensi pegawai berbasis <i>web</i> pada Kantor Kelurahan Sako Palembang memberikan fitur dapat memasukkan data secara terintegrasi dan meminimalisir terjadinya kehilangan atau rusaknya data.</p> <p>Di sisi lain, pimpinan dapat melakukan pemantauan dan pengontrolan terhadap kehadiran setiap pegawai serta sistem absensi ini hanya dapat diakses oleh pimpinan dan admin yang mengelola sistem sehingga tidak menimbulkan terjadinya kecurangan atau pemasukkan data oleh pihak yang tidak berwenang.</p>
3	Nama Jurnal :	JTIS, Volume 2 Nomor 2
	Judul Artikel :	Sistem Informasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website Menggunakan Metode Waterfall
	Penulis Artikel (Tahun) :	Dedi Saputra, Haryani, Artika Surniandari, Martias, dan Fajar Akbar (2022)
	Metode pengembangan yang digunakan :	Metode pengembangan dalam pembangunan dan perancangan sistem ini yaitu metode <i>waterfall</i>
	Kesimpulan :	Aplikasi sistem berbasis <i>web</i> ini membantu proses dalam kegiatan bimbingan antara mahasiswa dengan dosen pembimbing dari awal sampai akhir. Dalam aplikasi sistem ini dirancang dan dibangun dengan memberikan pelayanan dalam pengajuan judul dan pengajuan laporan skripsi mahasiswa ke dosen pembimbing
4	Nama Jurnal :	<i>Journal of Electrical, Electronic, Mechanical, Informatic and Social Applied Science (EEMISAS)</i> Vol. 2, No. 2, Hal. 09-16
	Judul Jurnal :	Sistem Informasi Monitoring Bimbingan Proyek Akhir Berbasis Website
	Penulis Artikel (Tahun) :	Wandha Wahyu Asmetha, Bagus Julianto, dan Dwi Ariani Finda Yuniarti (2022)
	Metode pengembangan yang digunakan :	Metode pengembangan dalam pembangunan dan perancangan sistem ini yaitu metode <i>waterfall</i> .
	Kesimpulan :	Sistem yang dirancang dan dibangun berbasis <i>web</i> memberikan kemudahan pada mahasiswa dan dosen dalam melakukan bimbingan proyek akhir dengan membantu

		dalam pengarsipan riwayat bimbingan yang dilaksanakan oleh dosen dan mahasiswa bimbingannya.
	Nama Jurnal :	<i>Journal of Student Development Information System (JoSDIS)</i>
	Judul Artikel :	Sistem Informasi Bimbingan Tugas Akhir Pada Prodi Manajemen Informatika Fakultas Sains dan Teknologi Universitas Labuhanbatu
	Penulis Artikel (Tahun) :	Tri Nizar Aulia (2022)
	Metode pengembangan yang digunakan :	Metode pengembangan dalam pembangunan dan perancangan sistem ini yaitu metode <i>waterfall</i> .
5	Kesimpulan :	Perancangan sistem berbasis <i>web</i> pada bimbingan tugas akhir pada program studi manajemen informatika dirancang menggunakan bahasa pemrograman PHP dan <i>software</i> Mozilla Firefox, notepad++ dan XAMPP. Dengan sistem ini mahasiswa dan dosen pembimbing diberikan kemudahan untuk melakukan bimbingan sampai dalam melakukan penyerahan tugas akhir dalam bentuk <i>soft copy</i> .

Berdasarkan hasil dari penelitian terdahulu pada dari [8], [9], [10], [11], dan [12], dapat disimpulkan bahwa dalam perancangan dan pembangunan sistem *website* dapat dilakukan menggunakan metode pengembangan *waterfall*. Metode ini membantu dalam pembangunan *website* pengolahan sistem informasi bimbingan mahasiswa tugas akhir dimana dalam alur kerja dan tahap perancangannya dilakukan secara terstruktur [11]. Namun pada metode *waterfall* ini memiliki kekurangan dimana dalam pengerjaan sistemnya sendiri membutuhkan waktu yang cukup lama, kurang fleksibel apabila adanya perubahan pada saat pembangunan *website*, hal ini dapat memakan waktu dan menghambat proses dalam pembuatannya [12]. Pembaharuan metode pengembangan pada penelitian yang dilakukan kali ini dari kelima penelitian pada tabel 2.3 yaitu metode pengembangan sistem untuk membangun sistem informasi *monitoring* bimbingan skripsi ini dilakukan menggunakan metode *prototyping*. Metode *prototyping* ini membutuhkan tingkat keahlian yang tidak terlalu tinggi namun dapat menghasilkan sistem *website* dengan tingkat fleksibilitas yang baik dan kompleks. Kemudian

waktu yang dibutuhkan untuk pengerjaan sistem *website* dengan metode pengembangan ini cukup singkat dan dapat membuat sistem *website* sesuai dengan kebutuhan dari pengguna dengan baik dan seksama.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA