

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam pembuatan Aplikasi Bimbingan Skripsi *Online* Mahasiswa Program Studi Teknik Informatika Fakultas Sains, Teknologi, Kesehatan Universitas Sahid Surakartaada beberapa penelitian yang terkait sebagai dasar dari penelitian ini sebagai berikut :

Penelitian dari Paramartha, dkk(2017) dengan judul Sistem Informasi Pembimbingan Skripsi*Online* Berbasis Web (Studi Kasus: FTK, UNDIKSHA) Penelitian ini bertujuan untuk membuat rancangan serta mengimplementasikan rancangan sistem skripsi online berbasis web. Penelitian ini dilakukan dengan model pengembangan SDLC (*System Development Life Cycle*) yaitu melalui tahapan requirement analysis, system design, implementation, testing, maintenance. Pengembangan sistem skripsi online berbasis web ini menggunakan bahasa pemrograman PHP dan basis data MySQL. Berdasarkan hasil pengujian blackbox dan whitebox, sistem pembimbingan skripsi online berbasis web telah berjalan dengan baik dan dapat digunakan untuk mengorganisasi dan memanajemen proses skripsi mahasiswa.

Penelitian dari Putra dan Arkan (2017) dengan judul Implementasi Sistem Bimbingan Tugas Akhir Mahasiswa Berbasis *Web Server* Dan Android membahas tentang proses bimbingan Tugas Akhir yang sering kali menyulitkan mahasiswa dikarenakan dosen yang sering tidak berada di tempat dan sedikitnya waktu untuk konsultasi. Dalam penelitian ini diimplementasikan Sistem Bimbingan Tugas Akhir Mahasiswa Universitas Bangka Belitung Berbasis Internet dan Android agar mahasiswa dapat melakukan proses bimbingan dimana saja selama ada koneksi Internet. Sistem bimbingan tugas akhir ini dibuat dengan sistem terintegrasi berbasis *web* menggunakan PHP, android dan database MySQL. *Web server* ini menyimpan data seperti identitas dosen pembimbing, mahasiswa bimbingan, tugas akhir mahasiswa, bimbingan tugas akhir secara *online*, sampai dengan informasi proses bimbingan. Mahasiswa dapat melakukan

bimbingan secara online dengan mengupload tugas akhir mereka serta dosen pembimbing dapat memeriksa tugas akhir mahasiswa secara online. Sehingga tanpa harus tatap muka mahasiswa dapat melakukan proses bimbingan tugas akhir mereka dan tanpa harus mencetak tugas akhir mereka selama proses pembimbingan. Selain itu, sistem ini juga akan menggunakan *notifikasi* berbasis android untuk mendapatkan informasi tentang Tugas Akhir.

Penelitian dari Sari, dkk (2017) dengan judul Sistem Informasi Bimbingan Tugas Akhir Pada Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Mulawarman membahas tentang sistem informasi yang digunakan untuk memfasilitasi mahasiswa dalam melakukan bimbingan dengan dosen pembimbing di FKTI Unmul. Mahasiswa dan dosen tidak perlu untuk melakukan tatap muka setiap kali melakukan bimbingan sehingga permasalahan yang dihadapi dalam pengerjaan tugas akhir tersebut dapat diselesaikan dengan lebih cepat. Penelitian ini menggunakan metode observasi dan studi literature dalam pengumpulan data. Pengembangan sistemnya menggunakan metode *waterfall*. Pemodelan analisis dan desain menggunakan bahasa pemrograman PHP dan *database* server MySQL. Metode pengujian menggunakan pengujian *white box*. Hasil penelitian ini adalah sebuah sistem informasi berbasis website yang dapat memudahkan dosen dan mahasiswa dalam melakukan bimbingan tugas akhir sehingga dalam pengerjaan tugas akhir tersebut dapat lebih efektif dan efisien.

2.2. Bimbingan Skripsi Online

Menurut Kamus Besar Bahasa Indonesia, bimbingan adalah petunjuk cara mengerjakan sesuatu. Sedangkan definisi skripsi menurut Kamus Besar Bahasa Indonesia adalah karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademisnya. Tugas akhir merupakan tugas final yang harus diselesaikan oleh seorang mahasiswa yang akan lulus dari jenjang pendidikan strata satu atau diploma di perguruan tinggi. Dalam pengerjaan tugas akhir, mahasiswa perlu melakukan konsultasi atau bimbingan kepada dosen pembimbing yang telah ditunjuk oleh koordinator tugas akhir. Hal ini bertujuan untuk membantu mahasiswa dalam mengerjakan tugas akhir lebih

terarah dan lebih fokus agar hasil yang diharapkan dapat tercapai (Kandaga, dkk, 2011).

Bimbingan memiliki arti sebagai petunjuk (penjelasan) cara mengerjakan sesuatu, tuntunan atau pimpinan. Sedangkan kata *online* berasal dari bahasa inggris yang dapat diartikan sebagai suatu keadaan yang sedang menggunakan jaringan, terhubung dalam jaringan, satu perangkat dengan perangkat lainnya yang terhubung sehingga bisa saling berkomunikasi (Dwiyani, 2013).

2.3. Aplikasi

Perangkat lunak aplikasi(*application software*), yaitu program yang ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan suatu aplikasi tertentu (Jogiyanto, 2005).

2.4. Basis Data

Basis data adalah kumpulan terorganisasi dari data-data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi, serta dipanggil oleh pengguna (Nugroho, 2011).

2.5. PHP

PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang akan dikirimkan ke klien, tempat pemakai menggunakan *browser* (Kadir, 2008).

PHP mempunyai kemampuan untuk menciptakan file gambar yang dapat digunakan untuk berbagai kepentingan pengguna internet, antara lain penyediaan fasilitas web counter untuk menghitung jumlah pengunjung, grafik (*chart*) untuk polling atau statistik, modifikasi gambar (memperkecil, memperbesar, atau memotong gambar), dan kepentingan rekayasa grafis lainnya. (Sanjaya, 2006).

2.5.1. Keunggulan PHP

Menurut Simarmata (2006), PHP begitu cepat populer dan berkembang begitu cepat karena PHP mempunyai beberapa keunggulan, yaitu:

- a. Cepat, karena ditempelkan (*embedded*) didalam kode HTML, sehingga waktu tanggap menjadi pendek.
- b. Tidak mahal – gratis. Pada kenyataannya PHP adalah gratis dan anda bisa mendapatkannya tanpa harus membayarnya.
- c. Mudah untuk digunakan. PHP berisi beberapa fitur khusus dan fungsi yang dibutuhkan untuk membuat halaman web dinamis. Bahasa PHP dirancang untuk dimasukkan dengan mudah didalam file HTML.
- d. Berjalan pada beberapa sistem operasi. Dia berjalan pada sistem operasi yang beragam, Windows, Linux, Mac OS dan kebanyakan variasi dari Unix.
- e. Dukungan teknis tersedia secara luas, karena PHP menyediakan dukungan gratis via daftar diskusi e-mail.
- f. Aman. Pengguna tidak melihat kode PHP, karena kode yang ditampilkan pada browser adalah kode HTML.
- g. Dirancang untuk mendukung database. PHP meliputi kemampuan yang dirancang untuk berinteraksi dengan database tertentu.
- h. Customizable. Lisensi open source sehingga mengizinkan para pemrogram untuk memodifikasi software PHP, Menambahkan atau memodifikasi fitur-fitur yang dibutuhkan untuk lingkungan mereka sendiri.

2.6. MySQL

Menurut Aditya (2011) MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database managementsystem*) atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *GeneralPublic License* (GPL).

2.6.1. Keunggulan MySQL

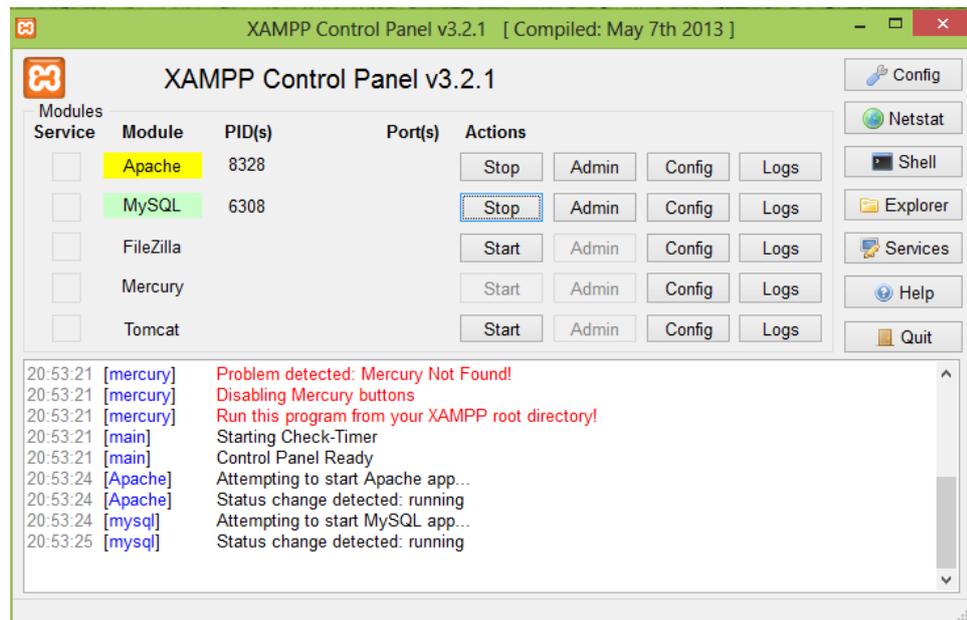
Menurut Simarmata (2006), keunggulan MySQL yaitu :

- a. Cepat. Tujuan utama dari pengembangan MySQL adalah kecepatan, sebagai konsekuensi software yang dirancang dari awal untuk kecepatan.
- b. Tidak mahal. MySQL adalah Cuma-Cuma dibawah lisensi GPI, open source, sementara pembiayaan untuk lisensi komersialnya sangatlah pantas.
- c. Mudah digunakan. Anda dapat membangun dan berinteraksi dengan database MySQL hanya dengan menggunakan sedikit pernyataan (statement) sederhana di dalam bahasa SQL, yang menjadi bahasa standar untuk komunikasi dengan RDBMS.
- d. Dukungan teknis secara luas tersedia. MySQL menyediakan dukungan Cuma-cuma untuk pengguna via mailing list. Pengembang MySQL juga berpartisipasi didalam e-mail list. Anda juga dapat membeli dukung teknis dari MySQL AB.
- e. Aman. MySQL adalah sistem otorisasi fleksibel yang mengizinkan beberapa atau semua privilege database (sebagai contoh, privilege untuk menciptakan suatu database atau menghapus data) untuk pengguna khusus atau kelompok pengguna.
- f. Mendukung databases yang besar. MySQL menangani database sampai 50 juta baris atau lebih. Batasan ukuran file secara default untuk tabel adalah 4 GB, tetapi Anda dapat menaikkannya (jika sistem operasi dapat menanganinya) hingga 8 juta terabytes (TB).
- g. Customizable. Lisensi GPI, open source mengizinkan pemrogram untuk memodifikasi software MySQL untuk mencocokkannya dengan lingkungan tertentu.

2.7. XAMPP

XAMPP adalah merupakan paket PHP dan MySQL berbasis *open source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP. XAMPP mengombinasikan beberapa paket perangkat lunak berbeda kedalam satu paket (Riyanto, 2011).

Berikut adalah tampilan panel XAMPP :



Gambar 2.1. XAMPP Control Panel

2.8. HTML

HTML merupakan dasar atau tulang punggung dalam pembuatan sebuah *web*. Dalam setiap file HTML akan selalu didahului dengan tag `<HTML>`. Tag ini akan membuat *web browser* mencoba untuk membaca file yang ada di dalam tag tersebut untuk ditampilkan dalam *webbrowser* (Aditya, 2011).

2.9. Sublime Text 3

Menurut Faridl (2015), “*Sublime text* adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer. Para

programmer biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang ia kerjakan. Sampai saat ini *sublime text* sudah mencapai versi 3”.

Sublime text mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*.

Berdasarkan definisi di atas, dapat disimpulkan *sublime text* adalah suatu aplikasi yang digunakan untuk membangun sebuah *web*.

2.9.1. Keunggulan Sublime Text 3

Berikut keunggulan-keunggulan fitur yang dimiliki Sublime Text 3, yaitu:

a. *Multiple Selection*

Multiple Selection mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. *Multiple selection* ini juga merupakan salah satu fitur unggulan dari *Sublime Text 3*. Kita dapat meletakkan kursor pada kode yang akan di ubah/*edit*, lalu tekan Ctrl+klik atau blok kode yang akan diubah kemudian Ctrl+D setelah itu kita dapat merubah kode secara bersamaan.

b. *Command Pallete*

Command Pallete mempunyai fungsi yang berguna untuk mengakses file shortcut dengan mudah. Untuk mencari file tersebut kita dapat tekan Ctrl+Shift+P, kemudian cari perintah yang kita inginkan.

c. *Distraction Free Mode*

Fitur ini mempunyai fungsi untuk merubah tampilan layar menjadi penuh dengan menekan Shift+F11. Fitur ini sangat dibutuhkan ketika pengguna ingin fokus pada pekerjaan yang sedang dikerjakannya.

d. *Find in project*

Fitur ini kita dapat mencari dan membuka file di dalam sebuah project dengan cepat dan mudah. Hanya dengan menekan Ctrl+P anda dapat mencari file yang diinginkan.

e. *Plugin API Switch*

Sublime Text mempunyai keunggulan dengan plugin yang berbasis Python Plugin API. Teks editor ini juga mempunyai plugin yang sangat beragam, dan ini dapat memudahkan pengguna dalam mengembangkan softwaranya.

f. *Drag and Drop*

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dalam editor yang akan membuka tab baru secara otomatis.

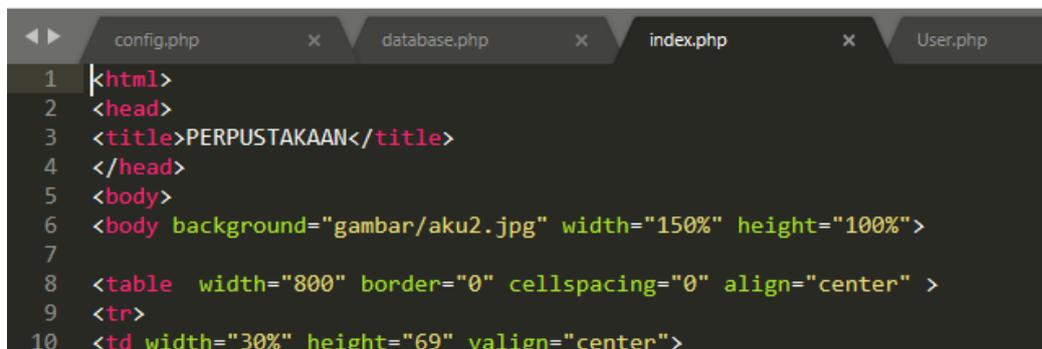
g. *Split Editing*

Di dalam fitur ini pengguna dapat mengedit file secara berdampingan dengan klik File->New menu into file.

h. *Multi Platform*

Sublime Text juga mempunyai keunggulan dalam berbagai platform. Sublime text sendiri sudah tersedia dalam berbagai platform sistem operasi, yaitu Windows, Linux, dan MacOS.

Berikut ini adalah contoh *script* menggunakan *software* sublime :



```

1 <html>
2 <head>
3 <title>PERPUSTAKAAN</title>
4 </head>
5 <body>
6 <body background="gambar/aku2.jpg" width="150%" height="100%">
7
8 <table width="800" border="0" cellspacing="0" align="center" >
9 <tr>
10 <td width="30%" height="69" valign="center">

```

Gambar 2.2. *Script* menggunakan *software* sublime

2.10. *Flowchart*

Menurut Suprpto, dkk(2008), Notasi algoritma yang paling banyak digunakan adalah *flow chart* karena bentuknya yang sederhana dan mudah dipahami. *Flow chart*(diagram alir) adalah penggambaran secara grafik dari langkah-langkah pemecahan masalah yang harus diikuti oleh pemroses. *Flow chart* terdiri atas sekumpulan simbol, dimana masing-masing simbol menggambarkan suatu kegiatan tertentu. *Flow chart* diawali dengan penerimaan

masukan (*input*), pemrosesan masukan, dan diakhiri dengan menampilkan hasilnya (*output*). Adapun simbol-simbol yang sering digunakan untuk menyusun *flow chart* adalah sebagai berikut :

a. Masukan

Masukan merupakan kegiatan penerimaan data yang disimbolkan dengan jajaran genjang. Kita dapat menuliskan masukan yang diperlukan pada suatu waktu secara satu per satu maupun secara keseluruhan, akan tetapi untuk alasan efisiensi ruang gambar biasanya masukan dituliskan bersamaan secara keseluruhan. Berikut merupakan gambar simbol masukan :



Gambar 2.3. Simbol Masukan

b. Masukan manual

Untuk masukan secara manual yang dimasukkan melalui *keyboard*, atau perangkat input lainnya seperti *barcode reader*, kita dapat menggunakan simbol masukan secara manual. Sama dengan simbol masukan, pada simbol masukan manual ini untuk alasan efisiensi ruang gambar biasanya masukan juga dituliskan bersamaan secara keseluruhan. Berikut merupakan gambar simbol masukan manual :



Gambar 2.4. Simbol Masukan Manual

c. Proses

Data yang dimasukkan kemudian diproses untuk menghasilkan jawaban atas persoalan yang ingin dipecahkan. Kegiatan memproses data ini disimbolkan dengan persegi panjang. Sama seperti simbol pada masukan, penulisan operasi-operasi pada data dapat dilakukan secara satu per satu maupun secara keseluruhan. Berikut merupakan gambar simbol proses :



Gambar 2.5. Simbol Proses

d. Keluaran

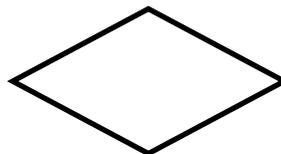
Keluaran adalah hasil dari pemrosesan data dan merupakan jawaban atas permasalahan yang ada. Keluaran ini harus ditampilkan pada layar monitor agar dapat dibaca oleh pengguna program. Sama seperti aturan pada simbol-simbol sebelumnya, penulisan hasil pemrosesan data dapat dilakukan secara satu per satu maupun secara keseluruhan. Berikut merupakan gambar simbol keluaran :



Gambar 2.6. Simbol Keluaran

e. Percabangan

Yang dimaksud dengan percabangan disini adalah suatu kegiatan untuk mengecek atau memeriksa suatu keadaan apakah memenuhi suatu kondisi tertentu atau tidak. Jadi dalam percabangan ini, kita harus menuliskan kondisi apa yang harus dipenuhi oleh suatu keadaan. Hasil dari pemeriksaan keadaan ini adalah YA atau TIDAK. Jika pemeriksaan keadaan menghasilkan kondisi yang benar, maka jalur yang dipilih adalah jalur yang berlabel YA, sedangkan jika pemeriksaan keadaan menghasilkan kondisi yang salah, maka jalur yang dipilih adalah jalur yang berlabel TIDAK. Berbeda dengan aturan pada simbol-simbol sebelumnya, penulisan kondisi harus dilakukan secara satu per satu (satu notasi percabangan untuk satu kondisi). Berikut merupakan gambar simbol percabangan :



Gambar 2.7. Simbol Percabangan

f. Sub rutin

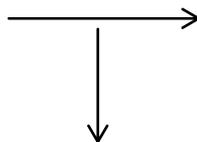
Sub rutin adalah suatu bagian dalam program yang dapat melakukan (atau diberi) tugas tertentu. Jadi sub rutin merupakan “program kecil” yang menjadi bagian dari suatu program yang besar. Sub rutin ada dua macam, yaitu prosedur (*procedure*) dan fungsi (*function*). Perbedaan antara keduanya adalah setelah dipanggil prosedur tidak mengembalikan suatu nilai sedangkan fungsi selalu mengembalikan suatu nilai. Dalam bahasa C++ kedua sub rutin tersebut dijadikan satu yaitu *function*, sedangkan untuk Java menggunakan *class* dimana keduanya bisa diatur untuk dapat mengembalikan nilai atau tidak dapat mengembalikan nilai. Sub rutin ini akan didiskusikan pada bab tersendiri. Sub rutin memiliki suatu *flow chart* yang berdiri sendiri diluar *flow chart* utama. Jadi dalam simbol sub rutin, kita cukup menuliskan nama sub rutusnya saja, sama seperti jika kita melakukan pemanggilan terhadap suatu sub rutin pada program utama (main program) yang akan anda pelajari pada bagian atau bab lain pada buku ini. Aturan penulisan simbol sub rutin sama dengan simbol percabangan, yaitu penulisan nama sub rutin dilakukan secara satu per satu. Berikut merupakan gambar simbol sub rutin :



Gambar 2.8. Simbol Sub Rutin

g. Arah aliran

Arah aliran merupakan jalur yang harus diikuti dan merupakan garis penghubung yang menghubungkan setiap langkah pemecahan masalah yang ada dalam *flow chart*. Arah aliran ini disimbolkan dengan anak panah. Berikut merupakan gambar simbol arah aliran :



Gambar 2.9. Simbol Arah Aliran

h. Terminator

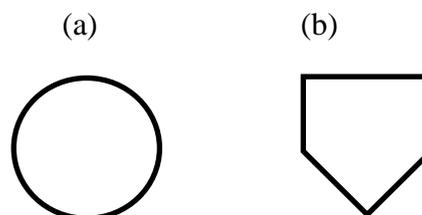
Terminator berfungsi untuk menandai titik awal dan titik akhir dari suatu *flow chart*. Simbol terminator ini diberi label MULAI untuk menandai titik awal dari *flow chart* dan label SELESAI untuk menandai titik akhir dari *flow chart*. Jadi dalam sebuah *flow chart* harus ada dua simbol terminator, yaitu simbol terminator untuk MULAI dan SELESAI. Berikut merupakan gambar simbol terminator :



Gambar 2.10. Simbol Terminator

i. Konektor

Konektor berfungsi untuk menghubungkan suatu langkah dengan langkah lain dalam sebuah *flow chart* dengan keadaan *on page* atau *off page*. Yang dimaksud dengan konektor *on page* adalah konektor yang digunakan untuk menghubungkan suatu langkah dengan langkah lain dalam satu halaman. Sedangkan konektor *off page* adalah konektor untuk menghubungkan suatu langkah dengan langkah lain dalam halaman yang berbeda. Konektor ini digunakan apabila ruang gambar yang kita gunakan untuk menggambar *flow chart* tidak cukup luas untuk memuat *flow chart* secara utuh, jadi perlu dipisahkan atau digambar di halaman yang berbeda. Berikut merupakan gambar simbol konektor *on page* dan *off page* :



Gambar 2.11. Simbol Konektor *On Page* (a) dan *Off Page* (b)

j. Dokumen

Dokumen merupakan tampilan data secara fisik yang dapat dibaca oleh manusia. Data ini biasanya merupakan hasil pemecahan masalah (informasi) yang telah dicetak (*print out*). Berikut merupakan simbol dokumen yang tertera pada Gambar 2.11.



Gambar 2.12. Simbol Dokumen

2.11. Unified Modelling Language

Unified Modelling Language adalah bahasa standar untuk penulisan cetak biru perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, mengonstruksi dan mendokumentasikan artifak-artifak suatu sistem *softwareintensive* (Pressman, 2010).

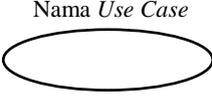
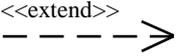
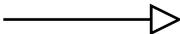
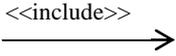
UML (*Unified Modeling Language*) merupakan bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma, berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010).

2.11.1. Tipe-Tipe Diagram UML

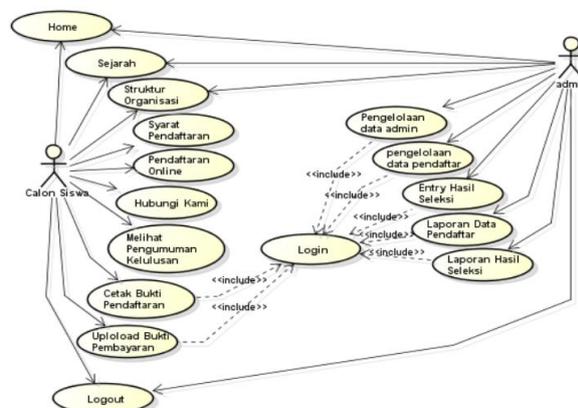
a. *Use Case Diagram*

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. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1. Simbol-simbol *Use Cass Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1	 Nama <i>Use Case</i>	<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frasa nama <i>use case</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat/titusendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frasa nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahkan sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun <i>use case</i> tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antar dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>Include</i>	Relasi <i>use case</i> tambahkan sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Gambar 2.13. adalah contoh *use case diagram* dari Eviani dkk. (2016) :



Gambar 2.13. Contoh *use case diagram* (Eviani dkk., 2016)

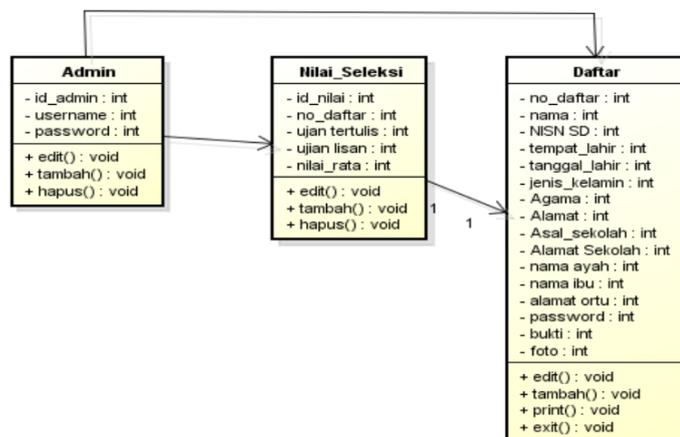
b. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi definisi kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. (Rosa & Shalahuddin, 2016). Simbol *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2. Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Class</i>	Kelas pada struktur sistem
2		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum - khusus).
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Gambar 2.14. adalah contoh *class diagram* dari Eviani dkk. (2016) :



Gambar 2.14.Contohclass diagram(Eviani dkk., 2016)

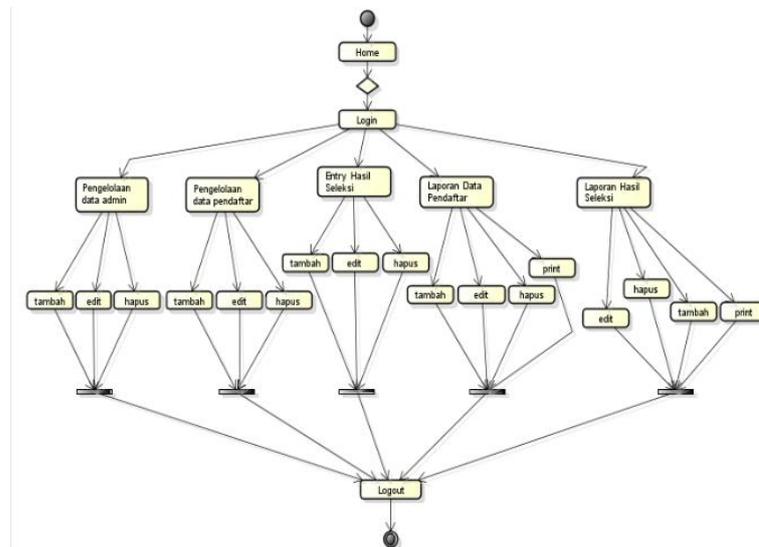
c. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa & Shalahuddin, 2016). Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.

Tabel 2.3. Simbol-simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1	●	Status awal	Status awal aktivitas sistem.
2	aktivitas	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	◇	<i>Decision</i>	Asosiasi percabang di mana jika ada pilihan aktivitas lebih dari satu.
4	—	<i>Join</i>	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.
5	●	Status akhir	Status akhir yang dilakukan sistem.
6	Nama Swimlane	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Gambar 2.15. adalah contoh *activity diagram* dari Eviani dkk. (2016) :



Gambar 2.15.Contoh *activity diagram*(Eviani dkk., 2016)

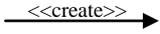
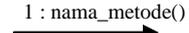
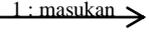
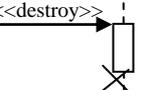
d. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. (Rosa & Shalahuddin, 2016) simbol *sequence* diagram dapat dilihat pada Tabel 2.4.

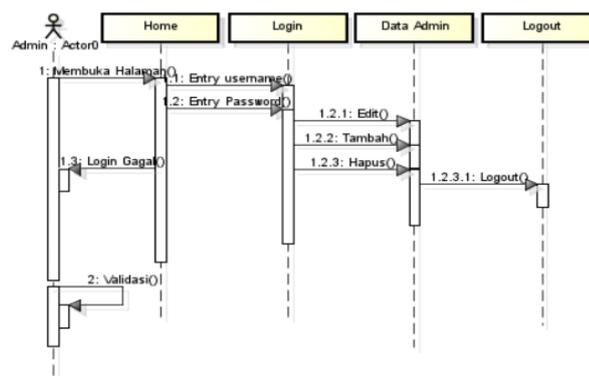
Tabel 2.4. Tabel *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frasa nama aktor.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.

Tabel 2.4. Tabel *Sequence Diagram*(Lanjutan)

NO	SIMBOL	NAMA	KETERANGAN
5		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8		Pesan tipe destroy	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Gambar 2.16. adalah contoh *sequence diagram* dari Eviani dkk. (2016) :



Gambar 2.16. Contoh *sequence diagram* (Eviani dkk., 2016)

2.12. Pengujian *Black Box*

Menurut Rosa dan Shalahuddin (2011) Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- Fungsi-fungsi yang tidak benar atau hilang.
- Kesalahan interface.

- c. Kesalahan dalam struktur data atau akses eksternal.
- d. Kesalahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.

2.13. Pengujian *Webqual*

Menurut Yudi (2011) Pengujian aplikasi bimbingan skripsi *online* terbagi menjadi 4 parameter yaitu kemudahan penggunaan, kualitas informasi, kualitas layanan interaksi, dan kepuasan pengguna. Adapun hasil dari pengujian disajikan berdasarkan parameter masing-masing. Interval penilaian dapat dilihat pada Tabel 2.5.

Tabel 2.5.Interval Penilaian

Jawaban	Skor	Interval	Interpretasi
STS	1	$1.00 \leq x < 1.75$	Sangat Tidak Setuju
TS	2	$1.75 \leq x < 2.50$	Tidak Setuju
S	3	$2.50 \leq x < 3.25$	Setuju
SS	4	$3.25 \leq x \leq 4.00$	Sangat Setuju

Perhitungan penilaian untuk menyatakan interpretasi dihitung dengan menentukan nilai rata-rata masing-masing pertanyaan di setiap parameter pengujian. Sebagai contoh perhitungan penilaian, dari 20 responden didapat data sebagai berikut :

STS = 2 responden dengan skor 1

TS = 3 responden dengan skor 2

S = 10 responden dengan skor 3

SS = 5 responden dengan skor 4

Jumlah responden = 20 responden

Nilai rata-rata = $\{(2 \times 1) + (3 \times 2) + (10 \times 3) + (5 \times 4)\} / 20$

= $\{2+6+30+20\} / 20$

= $58 / 20$

= 2.9

Dari hasil perhitungan, maka nilai 2.9 dapat diinterpretasikan setuju.