

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka merupakan garis besar penelitian-penelitian sebelumnya untuk menguatkan penulis mengambil judul tugas akhir ini dan sebagai referensi atas kebaharuan topik yang diangkat.

Aplikasi PHP dan MySQL dalam Pembuatan Sistem Informasi SMAN 1 Dukuhwaru Kabupaten Tegal Berbasis *Web* menurut Sasriah (2007), pembuatan sistem informasi SMA Negeri 1 Dukuhwaru Berbasis *Web* yang didukung program *Dreamweaver MX*, *Fireworks MX*, PHP dan MySQL dalam menyajikan informasi sekolah yang meliputi penyajian data siswa, absensi siswa, guru, data mata pelajaran, dan nilai akhir siswa. Metode kegiatan yang dilakukan dengan mengumpulkan data baik dengan metode observasi, metode literatur, dan selanjutnya dilakukan perancangan program. Hasil kegiatan berupa *website* yang siap di *upload* ke *internet* sehingga hasilnya siswa dapat dengan mudah mengakses data absen dan data nilai selama proses belajar, tetapi siswa tidak dapat melakukan *update* data kecuali administrator.

Pembuatan *Website* SMA Negeri 1 Pracimantoro Menggunakan PHP dan MySQL menurut Pujantoko (2009). Tujuan tugas akhir ini adalah membuat *website* profil sekolah SMA Negeri 1 Pracimantoro. *Website* sekolah yang dibuat sebagai bahan pertimbangan dalam penyampaian informasi dan profil SMA Negeri 1 Pracimantoro agar lebih dikenal dan diakses dengan mudah oleh siswa, guru, dan pihak luar sekolah. *Website* SMA Negeri 1 Pracimantoro ini berisikan profil sekolah yang terdiri dari sejarah, visi misi, fasilitas sekolah, data sekolah, data pengajar, *gallery* dan buku tamu. Data tersebut oleh SMA Negeri 1 Pracimantoro dapat dipergunakan sebaik-baiknya untuk meningkatkan mutu sekolah dan penyampian informasi sekolah kepada semua pihak dengan cepat dan efisien. Sistem *website* ini dibuat dengan program PHP dan *database* MySQL. Dapat disimpulkan bahwa sistem ini sudah siap dikembangkan.

Membangun *website* SMK Muhammadiyah Watukelir Wiyanto (2014), SMK Muhammadiyah Watukelir dalam hal menyajikan sistem informasi sekolah dan kegiatan siswa-siswi masih sederhana, sistem yang digunakan belum bisa langsung sampai semua pihak sesuai dengan kebutuhan yang diperlukan. Oleh karena itu, dengan berkembangnya kemajuan teknologi, SMK Muhammadiyah Watukelir perlu media promosi melalui sebuah *website* yang *online* di *internet*. *Website* ini dibangun agar, masyarakat dengan mudah mengakses dan mengetahui segala sesuatu mengenai SMK Muhammadiyah Watukelir dengan cepat dan efisien. Tujuan penelitian dalam penyusunan tugas akhir ini adalah membangun *website* SMK Muhammadiyah Watukelir, agar profil sekolah, data siswa, dan informasi-informasi tentang SMK Muhammadiyah Watukelir dapat diakses dengan mudah oleh masyarakat. Metode yang diajukan dalam membangun *website* SMK Muhammadiyah Watukelir yaitu dengan metode *Object Oriented Analisis And Design* dengan menggunakan *Unified Modeling Language* (UML) sebagai bahasa pemodelannya dan menggunakan PHP sebagai programnya serta MySQL sebagai *database*. Hasil perhitungan menggunakan metode *webqual* menunjukkan *Usability Quality* mempunyai pengaruh sebesar 8,6% terhadap kualitas *website*, *Information Quality* mempunyai pengaruh sebesar 16,2% terhadap kualitas *website* dan *Interaction Quality* mempunyai pengaruh sebesar 18,6% terhadap kualitas *website*. Setelah proses pengujian *website* selesai kemudian *website* di *upload* pada alamat <http://muhammadiyah1watukelir.tk/>.

Sistem Informasi Sekolah SMA Swasta Mulia Berbasis *Web* Menggunakan PHP dan MySQL menurut Kurniawan (2012), perkembangan teknologi khususnya komputer pada saat ini mempunyai peranan yang sangat penting dalam kehidupan di masyarakat. Teknologi yang berkembang sesuai tuntutan zaman diharapkan menjadi sarana penunjang untuk menangani permasalahan yang timbul dalam mengelola dan menyelesaikan permasalahan yang ada di perusahaan atau organisasi. Begitu juga dengan SMA Swasta Mulia Medan, diharapkan dapat menyediakan informasi yang tepat dan akurat dalam data siswa secara *online*. Sistem yang sedang berjalan pada SMA Swasta Mulia Medan saat ini belum sepenuhnya menggunakan sistem komputerisasi. Oleh sebab itu mencoba

merancang sebuah *database* pada SMA Swasta Mulia Medan. Dengan adanya portal ini diharapkan sekolah ini dapat menyediakan informasi yang lebih cepat dan akurat pada siswa maupun masyarakat yang juga dapat membantu dalam mempromosikan sekolah ini. *Web* ini dirancang dengan menggunakan bahasa pemrograman *web*, yaitu PHP 5.0 dan MySQL 5.0 sebagai *server database* serta *Apache* 2.0 sebagai *web server*. *Database* ini dirancang sedemikian rupa agar dapat menarik minat siswa dalam memperoleh informasi yang lebih cepat dan akurat serta menarik. *Database* ini bersifat *dinamis*, yaitu berarti bahwa informasi yang diberikan tersebut dapat berubah sewaktu-waktu berdasarkan keadaan yang sedang terjadi. *Database* ini meliputi banyak informasi seperti profil sekolah, data pegawai dan data siswa. Objektif utama portal ini adalah untuk menyediakan jendela bagi masing-masing siswa dalam mendapatkan informasi secara lengkap dan jelas.

Membangun *Website* Program Studi Psikologi Universitas Sahid Surakarta menurut Sutopo (2015). Program Studi Psikologi Universitas Sahid Surakarta saat ini dibawah Fakultas Ilmu Kesehatan di mana program studi Psikologi Universitas Sahid Surakarta masih manual dan kurang efektif. Sesuai dengan perkembangan teknologi informasi berbasis internet, *website* adalah solusi tepat sebagai media penyampaian informasi yang cepat dan efektif. Tujuan dari penelitian ini adalah membangun *website* Program Studi Psikologi Universitas Sahid Surakarta supaya memudahkan pengelolaan dan penyampaian informasi kepada mahasiswa dan masyarakat. Metode yang digunakan untuk pengumpulan data adalah observasi, wawancara, *literatur*, dan metode kuesioner untuk pengujian *webqual*. Perbandingan penelitian sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1. Perbandingan Penelitian Sebelumnya.

| No | Judul | Penulis | Tahun | Informasi Penting |
|----|--|-------------|-------|---|
| 1 | Aplikasi PHP dan MySQL dalam Pembuatan Sistem Informasi SMAN 1 Dukuhwaru Kabupaten Tegal Berbasis <i>Web</i> | Hery Sasria | 2007 | Penerapan sistem informasi SMA Negeri 1 Dukuhwaru berbasis <i>Web</i> yang didukung program <i>Dreamweaver</i> MX, <i>Fireworks</i> MX, PHP dan MySQL |

| | | | | |
|---|---|------------------|------|--|
| 2 | Pembuatan <i>Website</i> SMA Negeri 1 Pracimantoro Menggunakan PHP dan MySQL | Yoga Pujantoko | 2009 | <i>Website</i> sekolah yang dibuat sebagai bahan pertimbangan dalam penyampaian informasi dan profil SMA Negeri 1 Pracimantoro agar lebih dikenal dan diakses dengan mudah oleh siswa, guru, dan pihak luar sekolah. |
| 3 | Membangun <i>website</i> SMK Muhammadiyah Watukelir | Andi Wiyanto | 2014 | SMK Muhammadiyah Watukelir perlu media promosi melalui sebuah <i>website</i> yang <i>online</i> di <i>internet</i> . <i>Website</i> ini dibangun agar, masyarakat dengan mudah mengakses dan mengetahui segala sesuatu mengenai SMK Muhammadiyah Watukelir dengan cepat tanpa memerlukan banyak waktu |
| 4 | Sistem Informasi Sekolah SMA Swasta Mulia Berbasis <i>Web</i> Menggunakan PHP dan MySQL | Yofie K. Tarigan | 2012 | Teknologi yang berkembang sesuai tuntutan zaman diharapkan menjadi sarana penunjang untuk menangani permasalahan yang timbul dalam mengelola dan menyelesaikan permasalahan yang ada di perusahaan atau organisasi |
| 5 | Membangun <i>Website</i> Program Studi Psikologi Universitas Sahid Surakarta | Agus Sutopo | 2015 | <i>Website</i> adalah solusi tepat sebagai media penyampaian informasi yang cepat dan efektif. Tujuan dari penelitian ini adalah membangun <i>website</i> Program Studi Psikologi Universitas Sahid Surakarta supaya memudahkan pengelolaan dan penyampaian informasi kepada mahasiswa dan masyarakat. |

2.2 Kerangka Pemikiran

Kerangka pemikiran dapat dilihat pada Tabel 1.1. dan urutannya sebagai berikut:

1. Latar Belakang Masalah

Latar belakang masalah pada tugas akhir ini adalah Yayasan Winayabhakti Solo belum memiliki *website* sendiri.

2. Rumusan Masalah

Membangun *website* pada Yayasan Winayabhakti Solo.

3. Judul Tugas Akhir

Membangun *website* pada Yayasan Winayabhakti Solo dengan menggunakan PHP dan MySQL.

4. Pengumpulan Data Tertulis Dan Tidak Tertulis

Mengumpulkan semua data yang dibutuhkan baik melalui *interview* dengan staff IT Yayasan Winayabhakti Solo maupun dengan kepala yayasan.

5. Penguasaan dasar (PHP dan MySQL)

Penulis mencoba melakukan beberapa percobaan membuat aplikasi sederhana dengan tujuan agar dapat lebih menguasai bahasa pemrograman PHP dan MySQL sehingga hasilnya menjadi lebih maksimal.

6. Observasi Aplikasi

Mengamati beberapa aplikasi yang sudah ada, baik dari karya ilmiah, buku atau *internet* yang dapat dijadikan referensi untuk *website* Yayasan Winayabhakti Solo.

7. Analisis dan Perancangan Sistem

Menganalisa sistem yang sekarang digunakan dan merancang sistem yang akan dibangun seperti apa, bagaimana desainya, dan apa saja isinya, sehingga sistem ini dapat membantu memecahkan permasalahan dalam proses membangun *website* Yayasan Winayabhakti Solo.

8. Implementasi dan Pengembangan Sistem

- a. Perancangan *Database* MySQL

Membuat *database* dari data-data yang telah didapatkan sesuai dengan kebutuhan sistem.

- b. Perancangan Aplikasi Pengolahan Data

Membuat aplikasi dengan dasar *database* yang telah selesai dibuat.

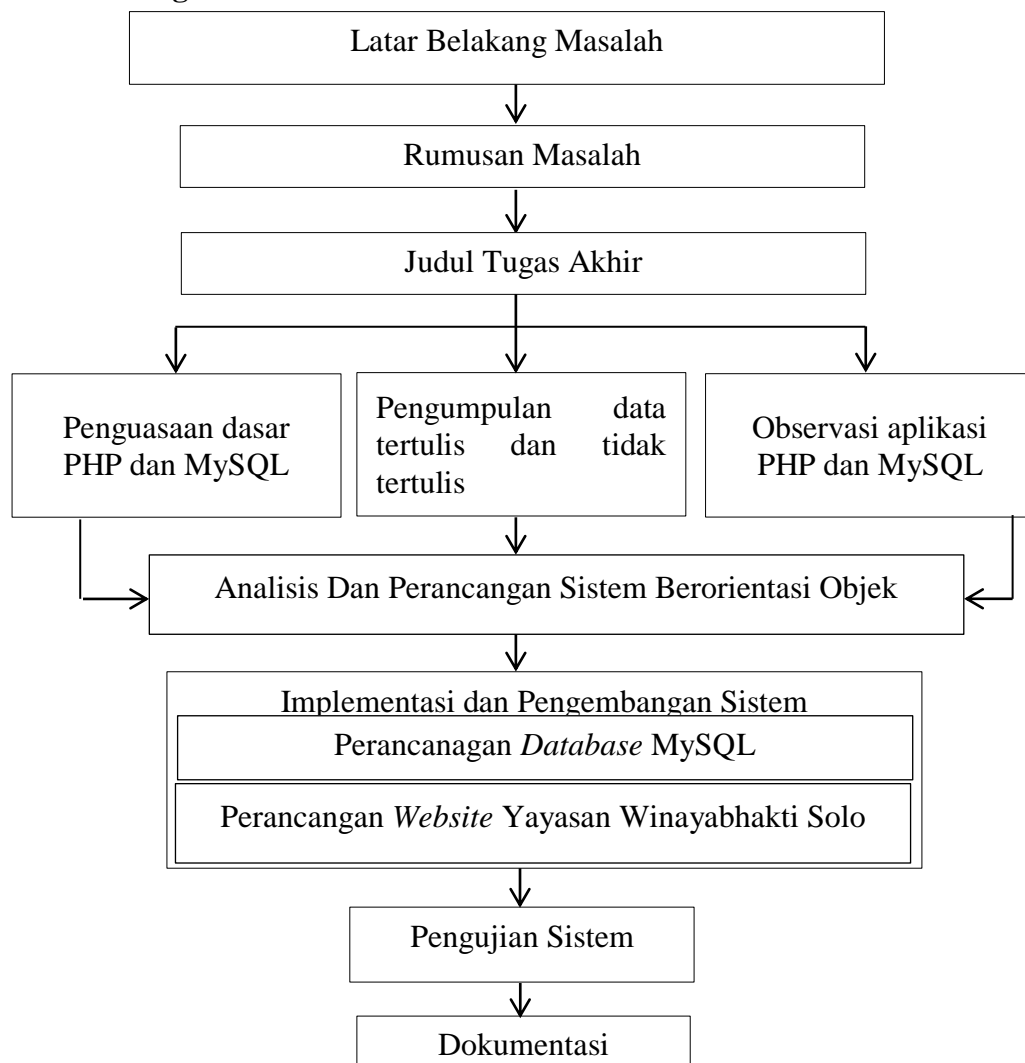
9. Pengujian Sistem

Pengujian sistem ini dilakukan bertujuan untuk mengetahui jika masih ada kesalahan atau kekurangan pada sistem yang dibuat.

10. Penerapan Sistem dan Dokumentasi

Pada tahap akhir dimana sistem telah siap digunakan pada *website* Yayasan Winayabhakti Solo setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan kegiatan penyusunan Tugas Akhir.

2.2.1 Kerangka Pemikiran



Gambar 1.1. Diagram Kerangka Pemikiran

2.3. Teori Pendukung

2.3.1. Pengertian *Website*

Sistem pengaksesan informasi dalam internet yang paling terkenal adalah *World Wide Web* (WWW) atau biasa dikenal dengan istilah *web*. Pertama kali diciptakan pada tahun 1991 di *Cern*, laboratorium fisika partikel Eropa, Jenewa Swiss. Tujuan awalnya adalah untuk menciptakan media yang mudah untuk berbagi informasi di antara para fisikawan dan ilmuwan Kadir (2013).

Informasi yang terdapat pada *web* disebut halaman *web*. Pengaksesan sebuah halaman *web*, pemakai perlu menyebutkan URL (*Uniform Resource Allocator*). URL tersusun atas format transfer, nama *host*, *path* berkas dokumen menurut Kadir (2013).

Web sendiri merupakan kumpulan dokumen-dokumen multimedia yang saling terhubung satu sama lain yang menggunakan protokol HTTP dan untuk mengaksesnya menggunakan *browser*. *Browser* merupakan perangkat lunak untuk menampilkan halaman-halaman *web* dalam format HTML, contoh *web browser* diantaranya adalah *internet explorer*, *mozilla firefox*, *opera* dan lain-lain. Secara umum *web* dibagi menjadi dua kategori yaitu *web dinamis* dan *web statis*. Perbedaan dari keduanya dapat dilihat dari isi atau konten jenis *web* tersebut. *Web dinamis* merupakan jenis *web* yang isinya selalu berubah-ubah, contohnya *web berita*, *web perdagangan (e-commerce)*, sedangkan *web statis* adalah jenis *web* yang tidak berubah-ubah kontennya biasanya *web* jenis ini dibuat menggunakan metode HTML yang hanya dapat diubah langsung pada kodingnya dan tidak memiliki halaman *admin*.

2.3.2. PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan diproses di dalam *server*. Hasilnya dikirim ke klien, tempat pemakai menggunakan *browser*. Konsep kerja PHP diawali dengan permintaan (*request*) suatu halaman *web* oleh *browser* berdasarkan URL (*Uniform Resource Locator*) atau yang biasa dikenal dengan alamat *internet*, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki

dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari *file* yang diminta dan memberikan isinya ke *web browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode dan menampilkan ke layar pemakai Kadir (2008). *Software* digunakan adalah *Notepad++*.

Notepad++ adalah program aplikasi pengembang yang berguna untuk mengedit teks dan skrip kode pemrograman. Versi terbaru program ini adalah *notepad++ v5.9*, yang dirilis pada tanggal 06 April 2012. *Software Notepad++* dibuat dan dikembangkan oleh tim *notepad++*. Perangkat lunak komputer ini memiliki kelebihan pada peningkatan kemampuan sebuah program *text editor*, lebih dari sekedar program *notepad* bawaan *windows*. *Notepad++* bisa mengenal *tag* dan kode dalam berbagai bahasa pemrograman. Fitur pencarian tingkat lanjut dan pengeditan teks yang tersedia juga cukup ampuh, sangat membantu tugas seorang programmer atau *developer* dalam menyelesaikan skrip kode programnya. Program *notepad++* banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman aplikasi *desktop* dan *web*. *Notepad++* merupakan *software* gratis (*opensource*). *Notepad++* dapat dijalankan di sistem operasi *Windows XP*, *Vista*, *Windows 7*, dan lain-lain.

2.3.3. Database

Database merupakan suatu koleksi dari data yang memiliki relasi atau hubungan. *Database* didesain untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi Connolly (2005). *Software* yang digunakan dalam membuat *database* adalah MySQL.

MySQL merupakan salah satu jenis *database* yang terkenal dan termasuk jenis RDBMS (*Relational Database Management System*). Kepopuleran MySQL disebabkan karena MySQL menggunakan bahasa SQL sebagai bahasa dasar untuk *query* dan *bersifat open sources* di berbagai *platform*. Ketika aplikasi yang dibuat membutuhkan informasi yang cukup banyak dan kompleks maka perlu adanya suatu tempat untuk menyimpan berbagai informasi atau data yang dibutuhkan secara terstruktur menggunakan *database*. Penggunaan *database* dimaksudkan

agar informasi yang ditampilkan dapat lebih fleksibel. Data terbaru dapat diakses oleh pengunjung dan terdokumentasi dengan baik. Oleh karena itu, peran MySQL sangat diperlukan sebagai sistem manajemen *database* Kadir (2008).

2.3.4. Web Server

Web server merupakan sebuah bentuk *server* yang khusus digunakan untuk menyimpan halaman *website* atau *homepage*. *Apache* merupakan turunan dari *web server* yang dikeluarkan oleh NSCA yaitu NSCA HTTPd sekitar tahun 1995-an. Pada dasarnya, *Apache* adalah “*APatCHy*” (*patch*) dan pengganti dari NCSA HTTPd. *Apache web server* merupakan tulang punggung permintaan dari *client* yang menggunakan *browser*, seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla*, *lynx* dan lain-lain. *Web Server* dalam berkomunikasi dengan kliennya menggunakan protokol HTTP (*Hyper Text Transfer Protocol*) Firdaus (2007).

XAMPP *Control Panel* merupakan sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai *server web* pada komputer lokal. XAMPP juga dapat disebut sebagai sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan *internet* Wicaksono (2008).

2.4. Model WebQual

WebQual yang merupakan salah satu teknik pengukuran kualitas *website* berdasarkan persepsi pengguna. Vidgen & B. (2003, 2005) sebagai pengembang metode *WebQual* telah melakukan beberapa penelitian menggunakan metode *WebQual* 4.0 untuk mengevaluasi beberapa *website* baik *website* non pemerintah (*ecommerce*) maupun *website* pemerintahan (*e-government*) yang mengacu pada tiga dimensi kualitas, yaitu *usability quality*, *information quality*, dan *service interaction*.

Beberapa penelitian selanjutnya telah memperluas dan mengembangkan metode *WebQual* 4.0 menjadi beberapa variabel *WebQual* untuk menilai kualitas sebuah *website*. Mudjahidin & N. (2013) melakukan penelitian untuk mengukur

kualitas *website* pemerintahan dari sisi kepuasan pengguna akhir dan keinginan pengunjung situs untuk menggunakan kembali layanan *website*, yang menitikberatkan pada sejauh mana persepsi tentang mutu layanan *website* yang dirasakan (*actual*) dengan tingkat harapan (*ideal*). Penelitian Mudjahidinn (2013) menggunakan tiga dimensi utama *WebQual* 4.0 (*usability quality, information quality, dan service interaction*) ditambahkan dengan dimensi keempat yaitu *design*.

Mengembangkan metode *WebQual* dan memperkenalkan *WebQual*TM, sebuah ukuran kualitas website dengan 12 dimensi didasarkan pada tinjauan literatur yang luas dan wawancara dengan desainer dan pengunjung *website*. Lima kategori umum kualitas *website* yang didapatkan dari kajian literatur dan eksplorasi yaitu: *ease of use, usefulness, entertainment, complementary relationship, dan customer service*.

Khawaja & Bokhari (2010) mengidentifikasi sembilan dimensi *University WebQual* (*reliability, navigability, responsiveness, efficiency, functionality, usefulness, ease of use, accuracy, dan web appearance*) untuk mengukur layanan kualitas website universitas ditambah efeknya pada kepuasan siswa. Dalam penelitian ini, metode *WebQual* yang digunakan adalah *WebQual* versi 4.0 yang telah dimodifikasi dengan menambahkan dimensi kualitas antarmuka pengguna (*user interface quality*) dengan tiga dimensi utama *WebQual* 4.0, yaitu kualitas kegunaan (*usability quality*), kualitas informasi (*information quality*) dan layanan interaksi (*service interaction*). Penambahan dimensi kualitas antarmuka pengguna (*user interface quality*) dalam penelitian ini adalah untuk mengukur kualitas antarmuka *website* yang berkaitan dengan daya tarik visual *website* karena pengguna *website* akan melihat tampilan *website* sebagai interaksi pertama.

2.5. Unified Modeling Language (UML)

UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. *Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh

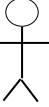
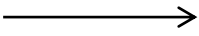

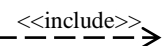
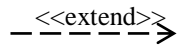
meta-model tunggal, yang membantu mendeskripsikan dan desain sistem perangkat lunak.

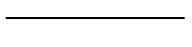

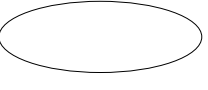

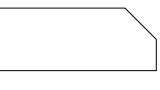
Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang diberbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang Shalahuddin, R. A. (2016) contoh dari diagram tersebut, antara lain:

2.5.1. Use Case Diagram

Diagram ini memperlihatkan himpunan *use case* dan aktor (sesuatu jenis dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. *Use case diagram* terlihat pada Tabel 2.2.

Tabel 2.2. Simbol *Use Case Diagram*

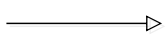

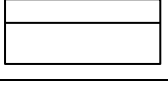

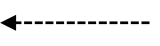
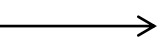
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|--|
| 1 |  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> |
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri (<i>independent</i>) |
| 3 |  | <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>) |
| 4 |  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> |
| 5 |  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan |


| | | | |
|----|---|----------------------|--|
| 6 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek yang lainnya |
| 7 |  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas |
| 8 |  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang di tampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi) |
| 10 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumberdaya komputasi |

2.5.2. Class Diagram

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka, kolaborasi dan relasi-relasi antar objek. Simbol *class diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3. Simbol *Class Diagram*





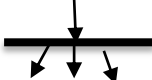
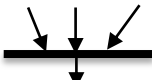
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------------------|--|
| 1 |  | <i>Generalization</i> | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum ke khusus) |
| 2 |  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari dua objek. |
| 3 |  | <i>Class</i> | Kelas pada struktur sistem |
| 4 |  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang di tampilkan 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 |

| | | | |
|---|---|--------------------|--|
| | | | mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri |
| 7 |  | <i>Association</i> | Hubungan antara objek satu dengan objek lainnya |

2.5.3. Activity Diagram

Diagram ini memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini penting terutama dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek. Simbol *activity diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4. Simbol *Activity Diagram*

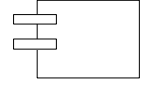


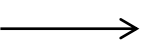
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|----------------------------|---|
| 1 |  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | <i>Action</i> | <i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | <i>Initial node</i> | Bagaimana objek dibentuk atau diawali |
| 4 |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | <i>Fork node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran. |
| 6 |  | <i>Join node</i> | Untuk menentukan gabungan dari beberapa bagian objek yang menjadi satu tujuan |

2.5.4. Deployment Diagram

Deployment diagram adalah salah satu jenis alat atau bahasa (*UML*) yang digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis *Object Oriented* yang akan dibangun. Tujuan atau fungsi dari *deployment diagram* yaitu

untuk menggambarkan/memvisualisasikan secara umum proses yang terjadi pada suatu sistem/software. Simbol *deployment diagram* dapat dilihat pada Tabel 2.5.





Tabel 2.5. Simbol *Deployment Diagram*



| NO | SIMBOL | NAMA | KETERANGAN |
|----|--|--------------------|--|
| 1 |  | <i>Component</i> | Pada <i>deployment diagram</i> , komponen-komponen yang ada diletakkan didalan <i>node</i> untuk memastikan keberadaan posisi mereka |
| 2 |  | <i>Node</i> | <i>Node</i> menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk <i>node</i> digambarkan sebagai sebuah kubus tiga dimensi. |
| 3 |  | <i>Association</i> | Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara komponen-komponen <i>hardware</i> . |
| 4 |  | <i>Depedency</i> | Kebergantungan antara <i>node</i> arah panah mengarah pada <i>node</i> yang dipakai. |

2.5.5. Statechart Diagram

Statechart diagram adalah suatu diagram yang menggambarkan daur hidup dari sebuah objek, dari awal objek tersebut diinisialisasi sampai di-*destroy*. Menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulasi yang diterima. Simbol *Statechart diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6. Simbol *Statechart Diagram*

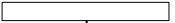
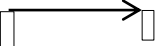
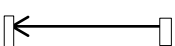
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|------------------------------|--|
| 1 |  | <i>State</i> | Nilai atribut dan nilai <i>link</i> pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | <i>Initial Peseudo State</i> | Bagaimana objek dibentuk atau diawali. |
| 3 |  | <i>Final State</i> | Bagaimana objek dibentuk dan dihancurkan. |
| 4 |  | <i>Transition</i> | Sebuah kejadian yang memicu sebuah <i>state</i> 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. |

2.5.6. Sequence Diagram

Sequence Diagram merupakan penjabaran *behavior* sebuah skenario tunggal. *Sequence Diagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *use case*. Simbol *Sequence Diagram* dapat dilihat pada Tabel 2.7.

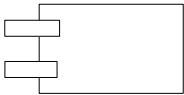
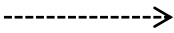
Tabel 2.7. Simbol *Sequence Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|--|
| 1 |  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi |
| 2 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | <i>Return Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

2.5.7. Component Diagram

Component merupakan implementasi *software* dari sebuah atau lebih *class*. *Component* dapat berupa *source code*, komponen biner atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Simbol *Component Diagram* dapat dilihat pada Tabel 2.8.

Tabel 2.8. Simbol *Component Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|------------------|--|
| 1 |  | <i>Component</i> | <i>Physical</i> dari sebuah sistem |
| 2 |  | <i>Usage</i> | Situasi di mana suatu elemen membutuhkan elemen lain untuk fungsi yang benar |