

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Sistem Informasi Unit Kegiatan Mahasiswa (UKM) Berbasis Web Pada Universitas Batanghari (Sadewa dan Siahaan, 2016)

Tampilan *website* Unit Kegiatan Mahasiswa pada Universitas Batanghari tercantum pada Gambar 2.1



Gambar 2.1 Tampilan *website* UKM pada Universitas Batanghari

Menurut Sadewa dan Siahaan (2016), dalam penelitiannya menjelaskan *website* yang telah dibuat tentang unit kegiatan mahasiswa (UKM) pada Universitas Batanghari me-restrukturisasi sistem informasi yang berisikan tampilan tentang kepengurusan, anggota serta kegiatan-kegiatan UKM, informasi pelaksanaan kegiatan, dokumentasi kegiatan serta info/berita terkini yang berhubungan dengan UKM. Informasi kepengurusan serta keanggotaan pada masing-masing UKM belum terekam secara *digital* yakni masih pada lembaran kertas Surat keputusan kepengurusan dan anggota, serta data kepengurusan hanya berupa nama bukan data lengkap identitas mahasiswa. Hal ini tentu menyulitkan bagian kemahasiswaan ketika mencari biodata lengkap pengurus UKM. Kemudian sering terdapatnya rangkap jabatan kepengurusan karena belum terdapatnya sistem informasi daftar kepengurusan. Pelaksanaan kegiatan UKM

terkadang memiliki kendala, yakni dalam hal waktu pelaksanaan dihari dan jam yang sama terdapat beberapa kegiatan, sementara banyak juga organisasi kemahasiswaan lainnya dilingkungan Universitas Batanghari, hal tersebut berdampak terhadap keterbatasan aula serta fasilitas seperti *sound system* dan parkir kendaraan.

Website UKM Batanghari ini dirancang untuk mempermudah pengguna internet mengakses informasi tentang UKM yang ada di Universitas tersebut. Dalam perancangan website tersebut menggunakan PHP sebagai bahasa pemrograman, Xampp (Apache, MySQL) digunakan sebagai *database*, dan untuk *interface website* pada penelitian ini menggunakan CMS Jomla 3.0. Sistem informasi tersebut dikatakan lebih efisien dari yang sebelumnya.

2.1.2 Sistem Informasi Unit Kegiatan Mahasiswa Stmik Amikom Yogyakarta Berbasis Web (Budyanto, 2014)

Tampilan *website* Unit Kegiatan Mahasiswa pada Universitas STMIK Amikom Yogyakarta tercantum pada Gambar 2.2



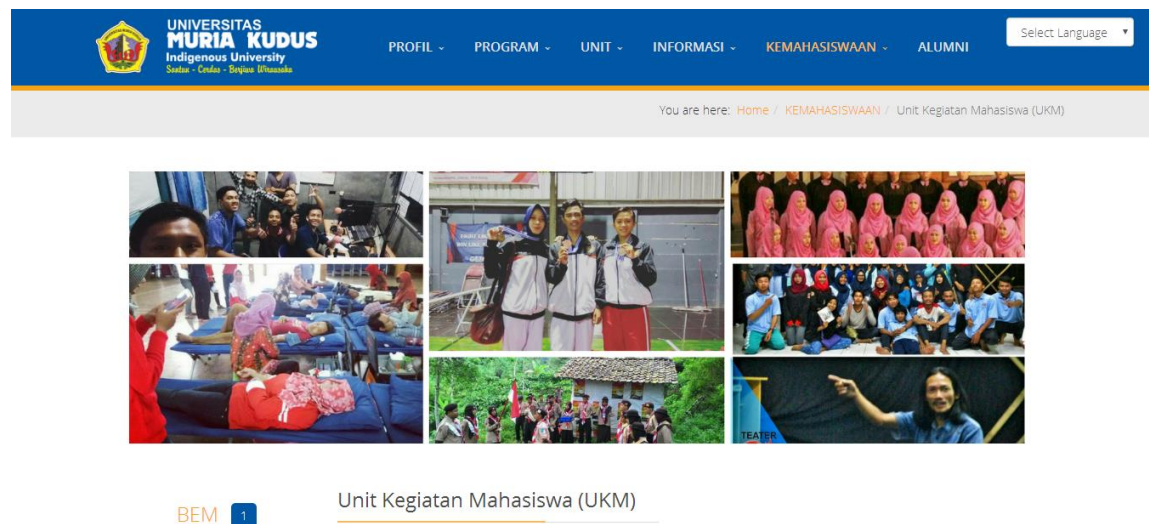
Gambar 2.2 Tampilan *website* UKM pada Universitas STMIK Amikom Yogyakarta

Menurut Budyanto (2014), dalam penelitiannya yang berjudul Sistem Informasi Unit Kegiatan Mahasiswa STMIK Amikom Yogyakarta Berbasis Web, yang dapat diakses oleh para mahasiswa yang mengikuti UKM untuk dapat membantu mereka untuk mempermudah urusan UKM dengan pihak Universitas.

Dengan adanya sistem informasi UKM dalam penelitian tersebut diharapkan dapat membantu agar mempermudah UKM dalam menyelesaikan urusan dengan bagian kemahasiswaan. *Website* UKM AMIKOM Yogyakarta ini dirancang untuk mempermudah pengguna *internet* mengakses informasi tentang UKM yang bisa dikatakan lebih akurat, relevan dan lebih mempersingkat waktu dari sistem sebelumnya. Dalam perancangan *website* tersebut menggunakan PHP sebagai bahasa pemrograman, menggunakan MySQL digunakan sebagai *database*, namun untuk *interface website* pada penelitian ini menggunakan Dreamweaver CS3. Dalam *website* tersebut ada beberapa halaman yaitu halaman *list* UKM yang berisikan *list* UKM yang ada di Universitas tersebut, dan halaman *list* agenda UKM yang berisikan agenda atau kegiatan yang dilaksanakan oleh UKM tersebut.

2.1.3 Aplikasi Manajemen Unit Kegiatan Mahasiswa Pada Universitas Muria Kudus Berbasis Web (Hidayat, dkk, 2015)

Tampilan *website* Unit Kegiatan Mahasiswa pada Universitas Muria Kudus tercantum pada Gambar 2.3.



Gambar 2.3 Tampilan *website* UKM pada Univesirtas Muria Kudus

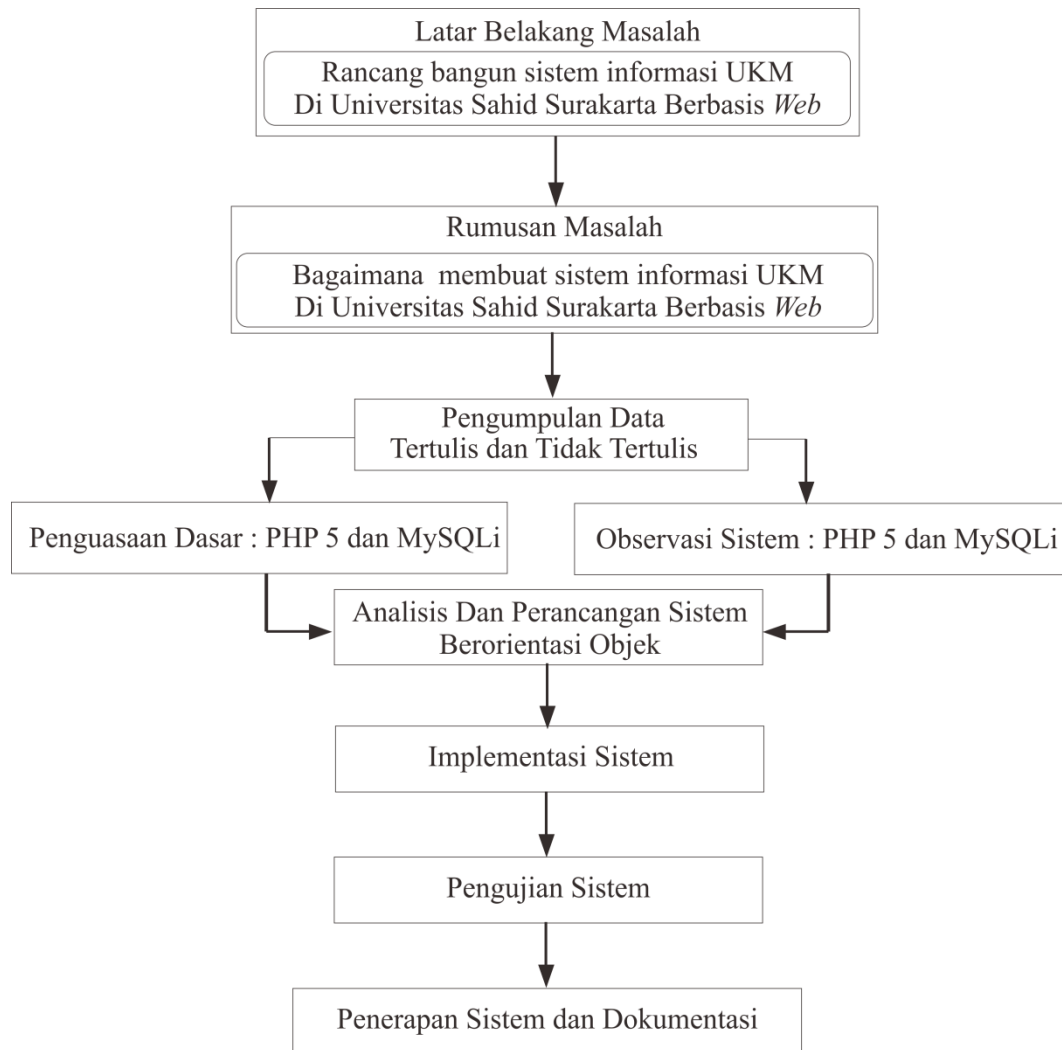
Menurut Hidayat, dkk (2015), hal dasar yang melatarbelakangi melakukan penelitian tersebut adalah kurangnya manajemen pada kegiatan-kegiatan yang mereka lakukan yang mengakibatkan banyak data yang hilang serta laporan-laporan yang tidak tertata rapi sehingga pada waktu laporan pertanggungjawaban sering kesusahan dalam mencari data kegiatan yang sudah terlaksana. Untuk itu perlu adanya aplikasi yang dapat mengolah data kegiatan dan laporan di Unit Kegiatan Mahasiswa (UKM). Dalam Aplikasi Manajemen Unit Kegiatan Mahasiswa yang telah dibuat dapat memberikan informasi yang cepat dan tepat sehingga dapat mempermudah kinerja anggota ukm dalam pengelolaan data yang ada. Aplikasi ini dapat digunakan sebagai alat bantu dalam memberikan informasi kegiatan, pengurus, rapat, sponsor, proposal, surat, dokumentasi, dan keuangan. Serta mendukung pembuatan laporan pertanggungjawaban. Selain memberikan informasi aplikasi ini juga dapat *upload* dan *download* data rapat, data surat dan data proposal. Aplikasi yang dibangun dapat menghasilkan laporan data kegiatan, data pengurus, data rapat, data sponsor, data proposal, data surat, dan data keuangan.

Dalam merancang aplikasi manajemen UKM berbasis *website* ini menggunakan bahasa pemrograman PHP dan MySQL digunakan untuk menghubungkan *database*. Tampilan atau *layout* dari *website* tersebut menggunakan Dreamweaver CS 3.

Dari tinjauan pustaka di atas dapat disimpulkan dengan adanya *website* tersebut pengguna *internet* dapat mengakses informasi UKM lebih efektif dari sebelumnya, dan rata-rata *website* pada UKM di setiap Universitas sudah dikatakan cukup baik, akan tetapi dari segi keamanan *website* perlu ditambah, karena perancangan dari tiga *website* tersebut menggunakan PHP dan MySQL. Sehingga dengan adanya referensi di atas pada penelitian kali ini dapat membangun sistem informasi UKM pada Universitas Sahid Surakarta dengan menambah beberapa fitur seperti pendaftaran mahasiswa menjadi anggota UKM dan menambahkan keamanan pada *website* agar data yang diakses lebih optimal dan keamanan *database* lebih terjamin.

2.2 Kerangka Pemikiran

Kerangka pemikiran dari penelitian tugas akhir ini telah tercantup pada Gambar 2.4.



Gambar 2.4 Kerangka Pemikiran

Uraian dari kerangka pemikiran penelitian tugas akhir sebagai berikut :

1. Latar Belakang Masalah

Inti permasalahan yang melatar belakangi perlu dibangun Sistem Informasi UKM di Universitas Sahid Surakarta berbasis *Website*.

2. Rumusan Masalah

Rumusan masalah yaitu penyelesaian dari permasalahan yang tertera di latar belakang.

3. Pengumpulan data

Pengumpulan penelitian ini menggunakan metode observasi, literatur, dan wawancara.

4. Penguasaan Dasar

Dalam penelitian tersebut sebelumnya harus melakukan percobaan membuat sistem supaya lebih menguasai terlebihnya untuk bahasa pemrograman

5. Observasi Sistem

Pada penelitian ini dilakukan observasi supaya di penelitian sebelumnya menjadi referensi dalam penelitian ini.

6. Analisis Dan Perancangan Sistem

Pada penelitian ini dilakukan analisis dan perancangan sistem bertujuan untuk mempermudah dalam pembuatan sistem.

7. Implementasi Sistem

Perlunya implementasi sistem ke dalam *website* yang telah dirancang dan dibuat.

8. Pengujian Sistem

Perlu adanya pengujian sistem supaya mengetahui apakah sistem tersebut ada yang kurang.

9. Penerapan Sistem dan Dokumentasi

Perlu adanya penerapan sistem dan dokumentasi agar sistem dapat di jalankan sesuai keinginan bisa berjalan seperti yang diharapkan.

2.3 Teori Pendukung

2.3.1 Sistem Informasi

Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi (Kadir, 2014).

Sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang mengolah data menggunakan perangkat memegang peranan yang penting dalam sistem informasi. Data akan

dimaksudkan dalam sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya (Subhan, 2012).

2.3.2 Pengertian Internet

Internet adalah solusi jaringan yang dapat menghubungkan beberapa jaringan lokal yang ada pada suatu daerah, kota bahkan negara. *Internet* dapat menghubungkan beberapa jaringan lokal yang ada pada setiap tempat (Kurniawan, 2004).

2.3.3 PHP 5.0

PHP kependekan dari *Hypertext Preprocessor* yang dibangun yang dibangun oleh Rasmus Lerdorf pada awal pengembangan PHP disebut sebagai *Personal Home Page*. *Php* merupakan produk *open source* sehingga kita dapat mengakses *source code*, menggunakan dan mengubahnya tanpa harus membayar sepeserpun.

PHP adalah bahasas *cripting* yang menyatu dengan HTML dan dilajankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada server sedangkan yang dikirim ke *browser* hanya hasil saja. PHP mampu berjalan melalui *platform* seperti *Windows*, *Unix* serta varian *Linux*. Beberapa kelebihan PHP antara lain dapat membuat situs yang interaktif dengan forum diskusi, *guestbok* dan sebagainya, koneksitas yang baik dengan bermacam-macam database seperti *Oracle*, *PostgreSQL*, dan lain-lain (Pratama, 2010).

PHP 5 saat ini memiliki fitur yang cukup banyak. Menjadikan bahas pemrograman PHP mendapatkan penghargaan dari masyarakat dunia, melalui *Zend Technologies (Zend Engine)* sebagai pengembangnya. Keluarga PHP 5.x mendukung sepenuhnya teknik *Object Oriented Programming* atau OOP, menjadikan bahasa ini sebagai salah satu yang terbaik didalam mengembangkan aplikasi *web* yang besar. Sekalipun demikian penggunaan teknik modulasi atau fungsi masih tetap digunakan sampai pada dekade terakhir ini karena merupakan unsur utama dalam penggunaan teknik OOP (Sakur, 2010).

Pada Juni 2004 Zend merilis kembali versi PHP 5.0 perubahan sangat besar sekali terjadi karena paradigmanya telah berubah menjadi suatu bahasa pemrograman yang berorientasi pada objek (OOP) (Saputra, 2013).

2.3.4 MySQLi

MySQL *Improved* atau MySQLi, merupakan peningkatan pengaksesan terhadap database MySQL, yang merupakan perkembangan dari PHP 5.x atau yang terbaru. Dengan menggunakan MySQLi, akan dapat mengakses seluruh fungsi-fungsi dari MySQL versi 4.1 ke atas, termasuk di dalamnya dapat mengakses *Stored Procedure*, *Stored Function* atau *Prepare Statement*. *Stored Procedure* dan *Function* serta *Trigger* merupakan komponen dasar dari SQL yang sudah didukung Oleh MySQL pada versi 5.0.2 atau yang terbaru.

Penggunaan MySQLi, merupakan opsional bagi pemrograman PHP dalam arti bahwa setiap pemrogram dapat saja tidak menggunakan APIs ini, akan tetapi tidak akan dapat memanfaatkan kemampuan MySQL yang lebih baru.

MySQLi dikembangkan dengan tujuan utama agar PHP dapat mengakses seluruh fasilitas-fasilitas terbaru yang ada pada MySQL 4.1 atau lebih baru. Sehingga secara tidak langsung fitur APIs ini menjadi kewajiban bagi pemrogram PHP ketika menggunakan database MySQL 5.0 atau lebih tinggi. MySQLi extension memiliki sejumlah manfaat, yang merupakan peningkatan dari MySQL extension, di antaranya adalah :

- a. Menggunakan *Interface Object Oriented*.
- b. Mendukung penggunaan *Prepared Statement*.
- c. Mendukung penggunaan *Multiple Statement*.
- d. Mendukung penggunaan *Transaction*.
- e. Peningkatan terhadap kemampuan pencarian kesalahan program (Debugging)
- f. Mendukung penggabungan dengan server.

Sekalipun menggunakan interface object-Oriented, MySQLi *extension* juga menyediakan pengaksesan dengan cara prosedural (Sakur, 2010).

2.3.5 Basis Data (*Database*)

Basis Data (*Database*) terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Fathansyah, 2007).

2.3.6 *Unified Modeling Language (UML)*

Menurut Sukamto dan Shalahuddin (2013: 133) dijelaskan *Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman *Smalltalk* pada awal 1980-an. Pada 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *Unified Modeling Language (UML)* diakomodasi oleh *Object Management Group (OMG)* sehingga sampai saat ini *Unified Modeling Language (UML)* telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

2.3.7 XAMPP Control Panel

XAMPP adalah *software* yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai server web pada komputer lokal. Xampp juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview*

sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan *internet* (Wicaksono, 2008)

2.3.8 Adobe Photoshop CS5

Adobe Photoshop CS5 merupakan *software* yang dapat digunakan untuk editing dan manipulasi foto. Kelengkapan fasilitas dan kemampuan yang luar biasa dalam editing foto atau gambar, menjadikan *software* ini paling banyak dipakai fotografer dan desainer grafis (Madcoms, 2011)

2.3.9 Unit Kegiatan Mahasiswa (UKM)

Unit Kegiatan Mahasiswa (UKM) adalah lembaga kemahasiswaan tempat berhimpunnya para mahasiswa yang memiliki kesamaan minat, kegemaran, kreativitas, dan orientasi aktivitas penyaluran kegiatan ekstrakurikuler di dalam kampus. Unit Kegiatan Mahasiswa pada perguruan tinggi memiliki andil yang cukup besar bagi perguruan tinggi, hal ini dikarenakan kegiatan ekstrakurikuler dapat membentuk pribadi mahasiswa yang berwawasan, bersosialisasi, beradaptasi dengan orang sekitar dan lingkungan, kreatif dan melatih diri menjadi pemimpin dalam organisasi (Sadewa dan Siahaan, 2016).

2.3.10 Bootstrap

Bootstrap sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat oleh alat bantu ini memiliki tampilan halaman yang sama atau mirip dengan tampilan halaman *twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *desktop*, tablet ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun dinon-aktifkan sesuai keinginan. Sehingga, membuat web untuk tampilan *desktop* saja dan apabila di-*reader* oleh *mobile browser* maka tampilan dari *web* yang dibuat sesuai layar. Dengan

bootstrap kita juga bisa membangun *web* dinamis ataupun statis. (Widyantoro, 2014).

2.3.11 Analisis Sistem

Menurut Mardi (2011: 124), analisis sistem adalah proses kerja untuk menguji sistem informasi yang sudah ada dengan lingkungannya sehingga diperoleh petunjuk berbagai kemungkinan perbaikan yang dapat dilakukan dalam meningkatkan kemampuan sistem. Jadi dapat disimpulkan bahwa analisis sistem adalah penelitian atas sistem yang telah ada supaya dapat mencapai tujuan dan spesifikasi untuk merancang sistem yang baru.

Adapun metode yang digunakan dalam analisis pemanfaatan multimedia interaktif sebagai media pengenalan UKM di Universitas Sahid Surakarta adalah analisis PIECES. Pengertian analisis PIECES adalah analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan harus dilakukan untuk mengidentifikasi masalah. Hal tersebut disebut dengan analisis PIECES yang berarti singkatan dari *Performance, Information, Economic, Control, Efficiency, Service*.

2.3.11.1 Analisis Kinerja (*Performance*)

Menilai proses atau prosedur yang ada apakah masih perlu ditingkatkan kinerjanya. Dalam hal ini kinerja diukur dari *throughput*, yaitu jumlah pekerjaan yang dapat dihasilkan pada saat tertentu dan *response time*, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output* tertentu. Kinerja adalah suatu kemampuan sistem dalam menyelesaikan tugas dengan cepat sehingga sasaran dapat segera tercapai. Kinerja diukur dengan jumlah produksi dan waktu yang digunakan untuk menyesuaikan perpindahan pekerjaan.

2.3.11.2 Analisis Informasi (*Information*)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Yang dimaksud kualitas informasi yang semakin baik adalah yang semakin relevan, akurat, andal, dan lengkap serta disajikan secara tepat waktu. Informasi

merupakan hal penting karena dengan informasi tersebut pihak manajemen dan user dapat melakukan langkah selanjutnya. Apabila kemampuan sistem informasi baik, maka *user* akan mendapatkan informasi yang akurat, tepat waktu dan relevan sesuai yang diharapkan.

2.3.11.3 Analisis Ekonomi (*Economy*)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya atau diturunkan biaya penyelenggaraannya. Pemanfaatan biaya yang digunakan dari pemanfaatan informasi. Peningkatan terhadap kebutuhan ekonomis mempengaruhi pengendalian biaya dan peningkatan manfaat.

2.3.11.4 Analisis Pengendalian (*Control*)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan atau kecurangan menjadi semakin baik pula. Analisis ini digunakan untuk membandingkan sistem yang dianalisa berdasarkan pada segi ketepatan waktu, kemudahan akses, dan ketelitian data yang diproses.

2.3.11.5 Analisis Efisiensi (*Efficiency*)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi. Efisiensi berhubungan dengan bagaimana sumber tersebut dapat digunakan secara optimal.

2.3.11.6 Analisis Pelayanan (*Service*)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Pelayanan berhubungan dengan tingkat kenyamanan dari seorang *user*.

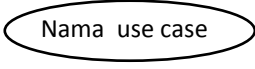

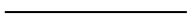
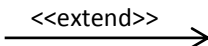
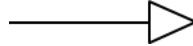
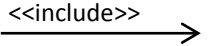
2.3.12 Perancangan Sistem.

2.3.12.1 Use Case Diagram

Use case diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case diagram* mendefinisikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use case diagram* berhubungan erat dengan kejadian-kejadian. Kejadian (skenario) merupakan contoh yang terjadi ketika

seseorang berinteraksi dengan sistem. *Use case diagram* dibuat untuk menggambarkan hubungan antara *Actor* dan *Use case*. Simbol-simbol dari *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*.

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>aktor</i> , biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi akan dibuat itu sendiri.
3.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
4.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalani fungsinya atau sebagai syarat

Sumber : Sukamto dan Salahudin (2016)

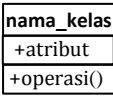



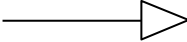

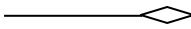
2.3.12.2 *Class Diagram*

Class diagram adalah himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute* atau

propoerties. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak atau beraksi dan memberikan reaksi.

Simbol-simbol dari *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol *Class Diagram*.



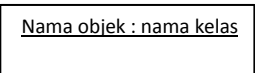

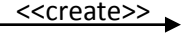
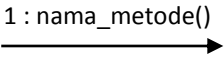
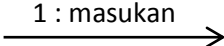
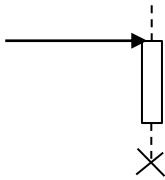
NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class</i>	Kelas pada struktur sistem
2.		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi object.
3.		<i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>Directed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

Sumber : Sukamto dan Salahudin (2016).

2.3.12.3 *Sequence Diagram*

Sequence diagram adalah grafik dua dimensi dimana objek ditunjukkan dalam dua dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal. Simbol-simbol dari *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Sequence Diagram*.


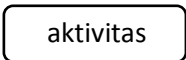
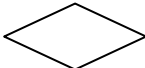


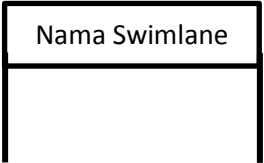
NO.	SIMBOL	NAMA	KETERANGAN
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang,
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
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 onjek lainnya, arah panah mengarah pada objek yang dikirim
8.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : Sukamto dan Salahudin (2016).

2.3.12.4 Activity Diagram

Activity diagram adalah diagram yang menggambarkan logika prosedural, proses dan jalur kerja. Dalam beberapa hal, *activity diagram* memainkan peranan yang mirip dengan diagram alir tetapi perbedaan prinsip antara notasi diagram alir adalah *activity diagram* mendukung *behavior paralel*. Simbol-simbol dari *activity diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol *Activity Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

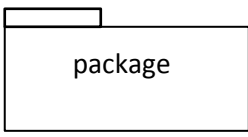
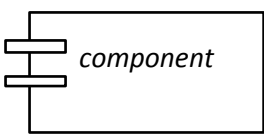


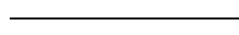
Sumber : Sukamto dan Salahudin (2016)

2.3.12.5 Component Diagram

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

Hubungan antara *component* dan *class*, *Component* adalah implementasi *software* dan sebuah *class*. *Class* mewakili abstraksi dari serangkaian *attribute* dan *operation*. Hal terpenting yang perlu diingat tentang *class* dan *component* adalah sebuah *component* bisa jadi merupakan implementasi dari lebih dari sebuah *class*. Simbol *Component Diagram* disajikan pada Tabel 2.5.

Tabel 2.5 Simbol *Component Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
2.		<i>Component</i>	Komponen sistem
3.		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4.		<i>interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5.		<i>Link</i>	Relasi antar komponen

Sumber : Sukamto dan Salahudin (2016).

2.3.12.6 *Deployment Diagram*

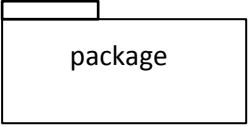
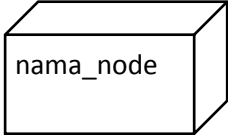


Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

- a. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node* dan *hardware*.
- b. Sistem *client/server*.
- c. Sistem berdistribusi murni.

d. Rekayasa ulang aplikasi.

Simbol-simbol *deployment diagram* dapat dilihat pada Tabel 2.6

Tabel 2.6 Simbol-Simbol *Deployment Diagram*.

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

Sumber : Sukamto dan Salahudin (2016).

2.3.13 Pengujian Sistem

2.3.13.1 Model WebQual

Webqual merupakan metode pengukuran kualitas *website* berdasarkan persepsi pengguna akhir (masyarakat). Metode ini merupakan pengembangan dari *servqual* Zeithaml et al. 1990 yang banyak digunakan sebelumnya pada pengukuran kualitas jasa.

WebQual adalah pengukuran berdasarkan *Quality Function Deployment* (QFD). Pengertian dari *WebQual* yaitu “*Structured and disciplined process that provides a means to identify and carry the voice of the customer through each stage of product and or service development and implementation*” (Slabey, 1990) dalam Barnes dan Vidgen (2003).

Dari pengertian di atas *WebQual* adalah suatu pengukuran untuk mengukur kualitas dari sebuah *website* berdasarkan instrumen-instrumen penelitian yang dapat dikategorikan ke dalam empat *variable* yaitu : *usability*, kualitas informasi, *services interaction* dan *overall*. Kesemuanya adalah pengukuran kepuasan konsumen atau *user* terhadap kualitas dari *website* tersebut. *WebQual* biasanya digunakan untuk menganalisa beberapa *website* di antaranya *website* pembelian, sekolah, pendidikan dan *website internet banking*. Instrumen-instrumen pada *webqual 4.0* adalah pengembangan dari versi-versi pendahulunya, yaitu *webqual 1.0*, *webqual 1.0*, *webqual 2.0*, dan *servqual* (Barnes dan Vidgen, 2003).