

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian ini akan menggunakan beberapa tinjauan studi yang akan digunakan untuk mendukung penelitian yang dilakukan. Tinjauan studi yang digunakan adalah sebagai berikut :

- 1) Menurut M. Hery Setyawan (2013) Sistem Informasi Akademik Berbasis *Web* SMA NEGERI 1 Bandar Menggunakan PHP dan MySQL.

Sekolah sebagai suatu instansi pemerintah di bidang pendidikan banyak melakukan pengolahan data dalam pengadministrasian data baik data siswa, guru maupun staff. Sering kali data - data akademik tersebut dalam jumlah yang besar dan dapat berubah sewaktu - waktu sehingga penyimpanan dan pengadministrasian harus dilakukan dengan baik dan selalu di *update* secara *continue*.

Sasaran dari program ini adalah SMA Negeri I Bandar. Ditinjau dari sistem informasi akademiknya sekolah tersebut masih mempunyai banyak kendala dalam pengelolaan data akademik, kebanyakan sekolah menggunakan sistem pengelolaan secara manual atau sudah menggunakan komputer tetapi belum menggunakan internet. Sistem tersebut tentu saja memiliki banyak sekali kelemahan yang implikasinya adalah lemahnya sistem administrasi, pemborosan tenaga, pelayanan yang kurang optimal dan kualitas data yang rendah.

- 2) Menurut Taufik Ramadhan dan Victor G Utomo (2014) Rancang Bangun Aplikasi Mobile Untuk Notifikasi Jadwal Kuliah Berbasis Android (Studi kasus : STMIK ProVisi Semarang).

Wujud layanan STMIK ProVisi di bidang akademik salah satunya ialah tersedianya informasi jadwal kegiatan mahasiswa. Media yang digunakan untuk menyampaikan informasi jadwal saat ini ialah papan pengumuman dan situs *web* resmi kampus. Informasi yang tersedia pada situs web resmi kampus antara lain informasi jadwal kuliah, dan informasi kalender akademik.

Kesulitan mengakses informasi jadwal yang dialami oleh mahasiswa tingkat akhir terjadi saat mereka ingin mengakses informasi jadwal ujian proposal dan jadwal ujian skripsi Situs web resmi kampus dalam penyampaian informasi jadwal juga belum efektif dan efisien, karena kurang lengkap. Keharusan mengunduh dalam bentuk PDF juga membuat proses mengakses informasi jadwal tidak dapat dilakukan secara instan.

3) Menurut Abd Hamid (2015) Perancangan Sistem Informasi Akademik Berbasis *Website* di MA AN-NAJAH I Karduluk Pragaan Sumenep Madura.

Madrasah Aliyah (MA) An-najah I merupakan salah satu instansi pendidikan di kabupaten Sumenep. Sebagai salah satu sekolah dengan jumlah peserta didik yang relatif banyak, MA An-najah I selalu berusaha untuk meningkatkan mutu dan pelayanan yang ada dalam data akademik. Sehingga diperlukan sebuah fasilitas yang bisa mendukung peningkatan efektifitas kerja akademik dan pembelajaran yang maksimal.

Berdasarkan data yang diperoleh, sampai saat ini MA An-najah I masih menggunakan cara manual untuk pengolahan data akademik, yakni berupa tumpukan kertas yang disusun rapi dalam rak berkas. Proses manual dengan banyak kelemahan yang berimplikasi kepada lemahnya administrasi, pemborosan tenaga, penyajian yang buruk, pelayanan yang kurang optimal dan kualitas data yang rendah masih dinilai lamban untuk mengelola data dalam skala besar. Ditambah penggunaan komputer untuk membantu mengelola data-data tersebut belum dapat dilakukan secara optimal.

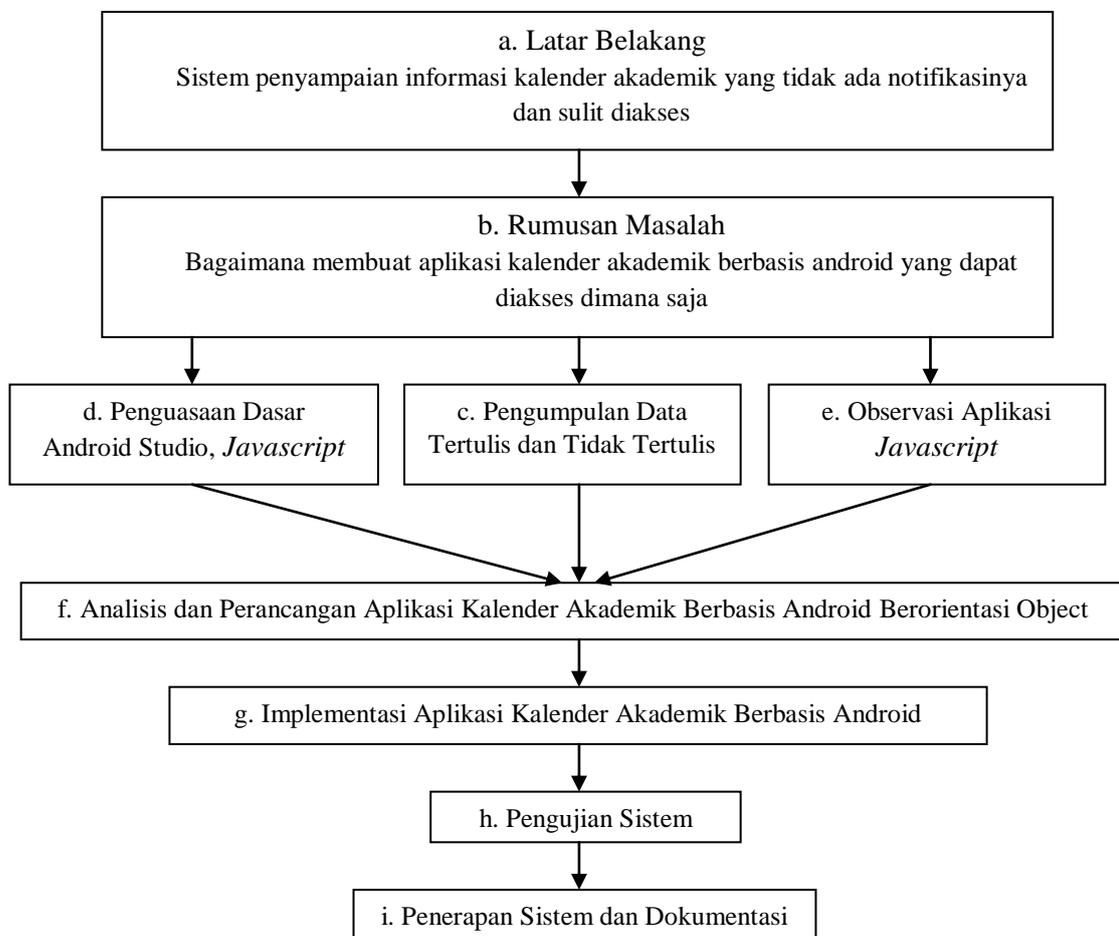
Dari ketiga jurnal diatas maka dapat disimpulkan bahwa rata – rata masih menggunakan metode penyampaian informasi secara manual yakni berupa papan pengumuman, tumpukan kertas yang disusun dalam rak berkas, dan penggunaan situs *web* resmi yang dinilai kurang efektif dan efisien. Untuk itu saran yang diajukan dalam pembuatan aplikasi kalender akademik di SMK Kasatrian Solo Sukoharjo adalah, aplikasi yang digunakan dapat dikembangkan lebih lanjut dengan menambahkan fitur seperti notifikasi untuk menghemat penggunaan memori *smartphone* penggunanya serta adanya basis data utama aplikasi yang

dapat digabungkan dengan basis data yang sudah terdapat di *website* resmi SMK Kasatrian Solo Sukoharjo sehingga memudahkan petugas memasukkan informasi kegiatan akademik.

2.2. Kerangka Berpikir

Menurut Sugiyono (2010), kerangka berfikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting.

Sistem andorid untuk membangun aplikasi kalender akademik di SMK Kasatrian Solo Sukoharjo ini dirancang untuk mempermudah pengguna dalam mendapatkan informasi seputar kegiatan di SMK Kasatrian Solo Sukoharjo. Data kegiatan ini didapatkan dari data real yang disusun sedemikian rupa, agar memudahkan pengguna untuk mengetahui informasi yang diinginkan.



Gambar 2.1 Kerangka Berpikir

2.2.1 Keterangan Kerangka Berpikir

a. Latar Belakang Masalah

Berisi mengenai kelemahan sistem penyampaian informasi kalender akademik yang tidak ada notifikasinya dan harus diakses melalui web yang membutuhkan komputer, membuat informasi sulit diakses secara cepat.

b. Rumusan Masalah

Membahas tentang bagaimana membuat aplikasi kalender akademik berbasis android yang dapat diakses dimana saja.

c. Pengumpulan Data

Dalam pembuatan aplikasi ini langkah - langkah pengumpulan data yang digunakan adalah secara tertulis dan tidak tertulis.

d. Penguasaan Dasar

Penguasaan dasar meliputi *software* yang digunakan dalam pembuatan aplikasi yaitu Android Studio menggunakan bahasa pemrograman *Javascript*.

e. Observasi Aplikasi

Bahasa pemrograman yang digunakan dalam pembuatan aplikasi kalender akademik ini adalah *Javascript*.

f. Analisis dan Perancangan Aplikasi Kalender Akademik Berbasis Android

Pada analisis dan perancangan aplikasi kalender akademik menggunakan metode berorientasi object.

g. Implementasi Aplikasi Kalender Akademik Berbasis Android

Setelah menganalisa dan merancang aplikasi kalender akademik maka selanjutnya akan dilaksanakan implementasi aplikasi kalender akademik menggunakan emulator android atau *smartphone*.

h. Pengujian Sistem

Pengujian sitem ini meliputi pengujian nilai – nilai aplikasi, fungsi tidak benar, kesalahan antarmuka atau *interface*, kesalahan akses, dan kesalahan dalam melakukan instalasi.

i. Penerapan Sistem dan Dokumentasi

Setelah dilakukan pengujian sistem tahap terakhir adalah penerapan hasil aplikasi kalender akademik yang akan dilakukan di SMK Kasatrian Solo Sukoharjo.

2.3 Teori Pendukung

2.3.1 Kalender Akademik

Kalender akademik adalah struktur kegiatan yang berfungsi memberikan informasi penjadwalan akademik secara tepat dan akurat untuk kegiatan pembelajaran siswa selama satu tahun pendidikan. Kalender akademik mencakup permulaan tahun ajaran pembelajaran, waktu efektif dalam pembelajaran dan hari libur. Kalender Akademik ditetapkan setiap permulaan awal tahun ajaran oleh setiap sekolah. Satuan pendidikan dapat menyusun kalender akademik sesuai dengan kebutuhan daerah, karakteristik sekolah, kebutuhan siswa dan guru, dengan memperhatikan kalender akademik sebagaimana yang dimuat dalam standar isi (Mulyasa, 2006: 184).

2.3.2. Android

Menurut Arifianto Teguh (2011 : 1) android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Menurut Stephanus Hermawan Susanto (2011 : 1), Android merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, *i-Phone OS*, *Symbian*, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

2.3.3 Versi Android

Android menurut Nazarudin Safaat H (2012) merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk

digunakan oleh bermacam peranti bergerak. Android umum digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS.

Android tidak terikat ke satu merek Handphone saja, beberapa vendor terkenal yang sudah memakai Android antara lain Samsung , Sony Ericsson, HTC, Nexus, Motorola, dan lain - lain. Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat.

Para pendiri Android Inc. bekerja pada Google, diantaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. Hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler.

Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi android terbaru yaitu versi 4.0. (Ice Cream Sandwich). 14

Android juga sudah bergabung dengan beberapa smart mobile seperti LG, Samsung, Sony Ericsson, dan lainnya. Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010). Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.

Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (Mobile) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Tabel 2.1 Tabel Versi Android

Versi	Nama kode	Tanggal rilis	Level API	Distribusi
6.0	Marshmallow	19 Agustus 2015	23	
5.x	<i>Lollipop</i>	15 Oktober 2014	21	
4.4.x	<i>KitKat</i>	31 Oktober 2013	19	24,5%
4.3.x	<i>Jelly Bean</i>	24 Juli 2013	18	8%
4.2.x	<i>Jelly Bean</i>	13 November 2012	17	20,7%
4.1.x	<i>Jelly Bean</i>	9 Juli 2012	16	25,1%
4.0.3–4.0.4	<i>Ice Cream Sandwich</i>	16 Desember 2011	15	9,6%
3.2	<i>Honeycomb</i>	15 Juli 2011	13	
3.1	<i>Honeycomb</i>	10 Mei 2011	12	
2.3.3–2.3.7	<i>Gingerbread</i>	9 Februari 2011	10	11,7%
2.3–2.3.2	<i>Gingerbread</i>	6 Desember 2010	9	
2.2	<i>Froyo</i>	20 Mei 2010	8	0,7%
2.0–2.1	<i>Eclair</i>	26 Oktober 2009	7	
1.6	<i>Donut</i>	15 September 2009	4	
1.5	<i>Cupcake</i>	30 April 2009	3	

2.3.4 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android (Yuniar Supardi, 2015).

2.3.5 Android Software Development Kit (SDK)

Android SDK adalah tool API (*application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat

lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang *release* oleh Google. Saat ini di sediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman java (Nazarudin Safaat H, 2012).

2.3.6 Aplikasi

Menurut Nazrudin Safaat H (2012 : 9) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Adapun karakteristik aplikasi adalah sebagai berikut:

- 1) Aplikasi merupakan elemen sistem logik dan bukan elemen sistem fisik seperti perangkat keras.
- 2) Elemen aplikasi bisa rusak.
- 3) Elemen aplikasi bisa direkayasa atau dikembangkan dan bukan dibuat di pabrik, seperti halnya perangkat keras.
- 4) Aplikasi tidak bisa dirakit atau disusun.

Pada sistem android ini akan menampilkan pilihan menu utama sebagai berikut :

- 1) Sejarah SMK Kasatrian Solo Sukoharjo
- 2) Kalender menu
- 3) List kalender
- 4) Tentang aplikasi

2.3.7 Struktur Navigasi

Struktur navigasi adalah alur yang digunakan dalam aplikasi yang dibuat. Sebelum menyusun aplikasi multimedia kedalam sebuah *software*, kita harus menentukan terlebih dahulu alur apa yang akan digunakan dalam aplikasi yang dibuat (Fauzi Fivi Syukriah : 2008). struktur navigasi terdiri dari 4, yaitu:

1. Struktur Navigasi Linier

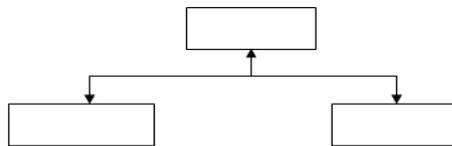
Struktur navigasi linier merupakan struktur yang mempunyai satu rangkaian cerita berurutan. Struktur ini menampilkan satu demi satu tampilan layer secara berurutan menurut aturannya. Struktur navigasi linier dapat dilihat pada Gambar 2.2.



Gambar 2.2 Struktur navigasi linier

2. Struktur Navigasi Hirarki

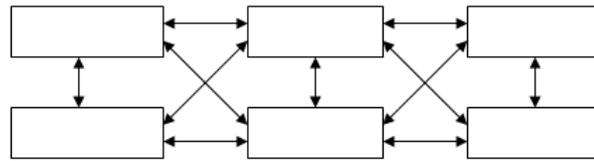
Struktur navigasi hirarki sering disebut dengan navigasi bercabang, yaitu merupakan struktur yang mengandalkan percabangan untuk menampilkan data atau gambar pada layer dengan kriteria tertentu. Tampilan pada menu utama disebut master page (halaman utama satu), halaman tersebut mempunyai halaman percabangan yang disebut slave page (halaman pendukung) dan jika dipilih akan menjadi halaman kedua, begitu seterusnya. Struktur navigasi Hirarki dapat dilihat pada Gambar 2.3.



Gambar 2.3 Struktur navigasi hirarki

3. Struktur Navigasi Non Linier (tidak berurut)

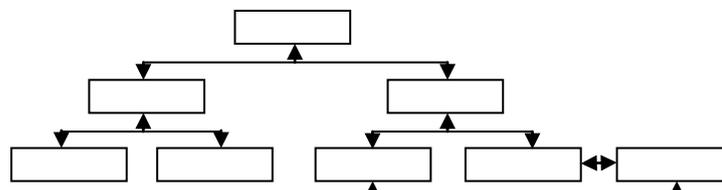
Struktur navigasi non linier merupakan pengembangan dari struktur navigasi linier, hanya saja pada struktur ini diperkenankan untuk membuat percabangan. Percabangan pada struktur non linier berbeda dengan percabangan pada struktur hirarki, pada struktur ini semua kedudukan page sama, sehingga tidak kenal adanya master atau slave page. begitu seterusnya. Struktur navigasi non linier dapat dilihat pada Gambar 2.4.



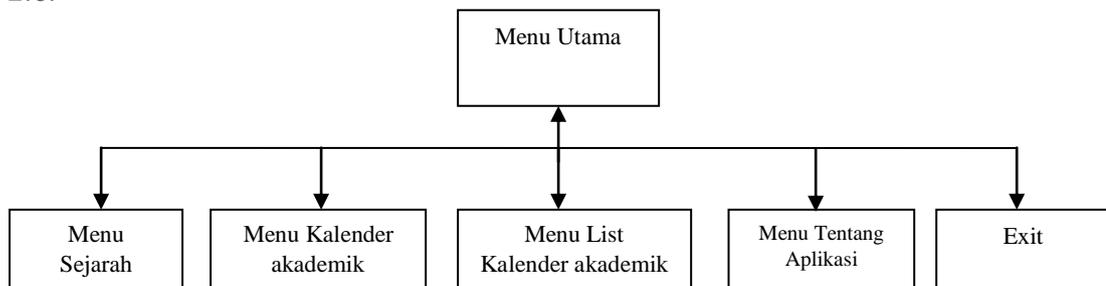
Gambar 2.4 Struktur navigasi non linier

4. Struktur Navigasi Campuran (*Composite*)

Struktur navigasi campuran struktur ini merupakan gabungan dari struktur sebelumnya dan disebut juga struktur navigasi bebas, maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam pembuatan aplikasi multimedia. Struktur navigasi composite dapat dilihat pada Gambar 2.5.

Gambar 2.5 Struktur navigasi *composite*

Struktur yang akan diterapkan pada aplikasi ini adalah navigasi Hirarki karena pada aplikasi ini terdapat halaman utama dengan halaman-halaman pendukung yang merupakan percabangan dari halaman utama. Struktur interaktif yang sesuai dengan kalender akademik ini adalah struktur navigasi model hirarki, karena model ini baik bagi aplikasi untuk menemukan lokasi halaman dengan mudah. Struktur navigasi Hirarki kalender akademik dapat dilihat pada Gambar 2.6.



Gambar 2.6 Struktur navigasi hirarki kalender akademik

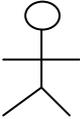
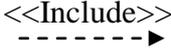
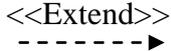
2.3.8 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek (Adi Nugroho, 2012).

2.3.8.1 Use Case Diagram

Use case diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. Simbol – simbol *Use Case Diagram* dapat dilihat pada Tabel 2.2

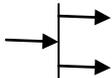
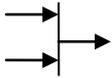
Tabel 2.2 Simbol – simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Use Case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2		<i>Actor</i>	Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
3		<i>Include</i>	Menspesifikasikan bahwa pemanggilan <i>use case</i> oleh <i>use case</i> lain
4		<i>Extend</i>	Menspesifikasikan bahwa perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi
5		<i>Generalisasi/ Generalization</i>	Hubungan generalisasi dan spesialis antara dua buah <i>use case</i> .

2.3.8.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan aliran kerja atau aktifitas dari suatu sistem. Perlu diperhatikan bahwa *activity diagram* menggambarkan aktifitas sistem bukan apa yang dilakukan aktor. Simbol – simbol *Activity Diagram* dapat dilihat pada Tabel 2.3.

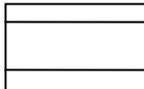
Tabel 2.3 Simbol – simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Activity Initial Node</i>	Bagaimana object dibentuk atau diawali.
3		<i>Activity Final Node</i>	Bagaimana object diakhiri atau dihancurkan
4		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
5		<i>Join node</i>	Beberapa aliran yang pada tahap tertentu menjadi satu aliran.
6		<i>Decision node</i>	Suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi

2.3.8.3 Class Diagram

Class Diagram pada Tabel 2.4 menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti *contaiment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem yang merupakan tipe diagram yang paling banyak dipakai. Simbol – simbol *Class Diagram* dapat dilihat pada Tabel 2.4.

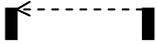
Tabel 2.4 Simbol – simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana object anak (<i>descendent</i>) berbagi perilaku dan struktur data dari object yang ada di atasnya object induk (<i>ancestor</i>).
2		<i>Class</i>	Himpunan dari object-object yang berbagi atribut serta operasi yang sama.
3		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu object.
4		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
5		<i>Association</i>	Apa yang menghubungkan antara object satu dengan object lainnya

2.3.8.4 Sequence Diagram

Sequence Diagram pada Tabel 2.5 mendeskripsikan bagaimana entitas dalam sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Diagram ini juga menunjukkan serangkaian pesan yang diperlukan oleh object-object yang melakukan suatu tugas atau aksi tertentu. Simbol – simbol *Sequence Diagram* dapat dilihat pada Tabel 2.5.

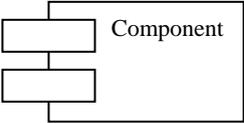
Tabel 2.5 Simbol – simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Object <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar object yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Actor</i>	Pengguna diluar sistem.
4		<i>Boundary</i>	Boundary biasanya berupa tepi dari system, seperti user interface, atau suatu alat yang berinteraksi dengan system lain.
5		<i>Control element</i>	Control element mengatur aliran dari informasi untuk sebuah scenario.
6		<i>Entity</i>	Entity biasanya elemen yang bertanggung jawab menyimpan data atau informasi.
7		<i>Return</i>	Spesifikasi dari komunikasi antar object yang memuat informasi-informasi tentang aktifitas yang terjadi

2.3.8.5 Deployment Diagram

Deployment Diagram pada Tabel 2.6 menggambarkan detail bagaimana komponen di-sebar (*di-deploy*) kedalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *node*, *server*, atau piranti kertas apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Simbol – simbol *Deployment Diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6 Simbol – simbol *Deployment Diagram*

No	Nama	Gambar	Keterangan
1	Component		Pada <i>deployment diagram</i> , <i>component - component</i> yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.
2	Node Name		<i>Node</i> menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
3			Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen <i>hardware</i> .

2.4 Metode Pengujian *Blackbox* (*Blackbox Testing*)

Black Box Testing atau Pengujian Kotak Hitam atau juga disebut Behavioral Testing, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black Box Testing* memungkinkan untuk mendapatkan *set* kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black Box Testing* bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black Box Testing* adalah pendekatan *komplementer* yang

mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*. (Roger S. Pressman : 2010).

Black Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut :

1. Fungsi tidak benar atau hilang.
2. Kesalahan *interface* atau antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja atau perilaku.
5. Kesalahan inisialisasi dan terminasi.

2.5 Kuesioner

Menurut Sugiono, (2010). Kuesioner merupakan teknik pengumpulan data dengan cara memberikan daftar pertanyaan tertulis yang ditujukan kepada responden yang jumlahnya banyak sehingga tidak memungkinkan untuk dilakukan pengumpulan data melalui wawancara. Terdapat lima buah pilihan jawaban yang disediakan untuk pertanyaan pada kuesioner, yaitu:

- a. Sangat Setuju (SS)
- b. Setuju (S)
- c. Biasa Saja (BS)
- d. Kurang Setuju (KS)
- e. Tidak setuju (TS)

Berdasarkan data yang dihasilkan dari kuesioner, dapat dihitung nilainya berdasarkan persentase masing-masing jawaban dengan menggunakan rumus:

$$Y = P/Q * 100\% \quad (I)$$

Keterangan: Y = Nilai presentase

P = Banyak jawaban responden dari tiap soal

Q = Jumlah responden

Sedangkan untuk menarik kesimpulan dari setiap pertanyaan, dilakukan perhitungan menggunakan skala likert. Skala likert adalah metode perhitungan yang digunakan untuk keperluan riset atas jawaban setuju atau tidaknya seorang responden terhadap suatu pernyataan. Untuk menghitung skor maksimum tiap jawaban, dengan mengalikan skor dengan jumlah keseluruhan responden, yaitu

skor responden. Skor jawaban untuk responden kuesioner dapat dilihat pada tabel 2.7.

Tabel 2.7 Tabel Skor Jawaban Kuesioner

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Biasa Saja	3
Kurang Setuju	2
Tidak Setuju	1