

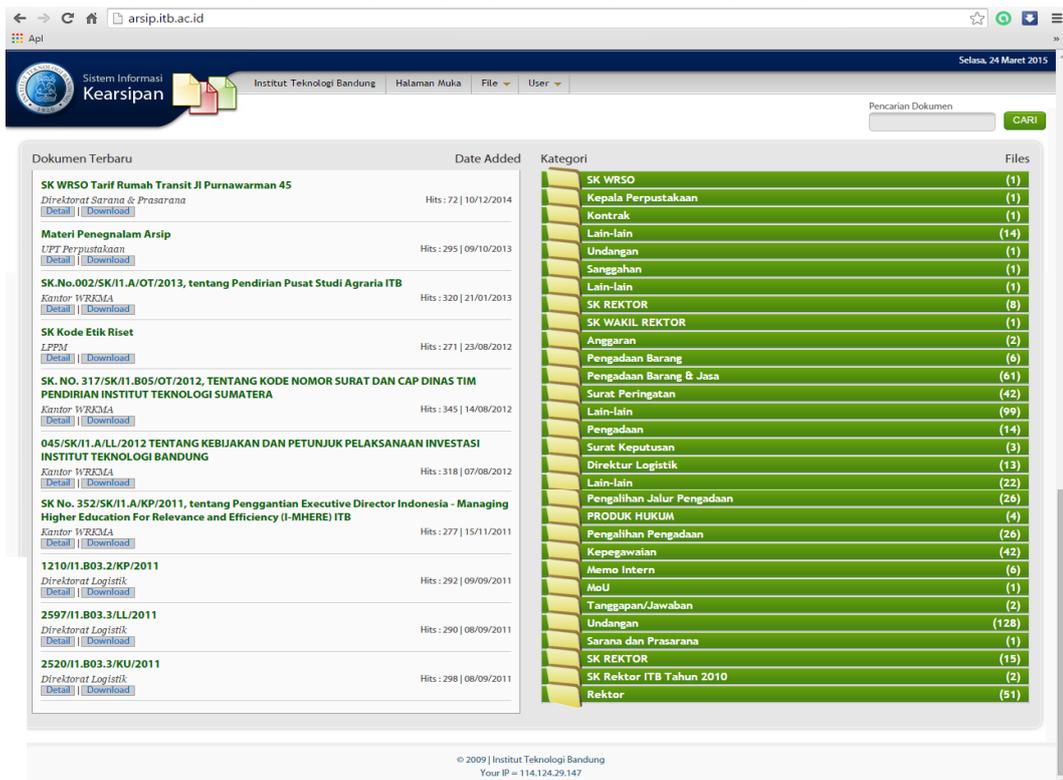
## BAB II

### LANDASAN TEORI

#### 1.7. Tinjauan Pustaka

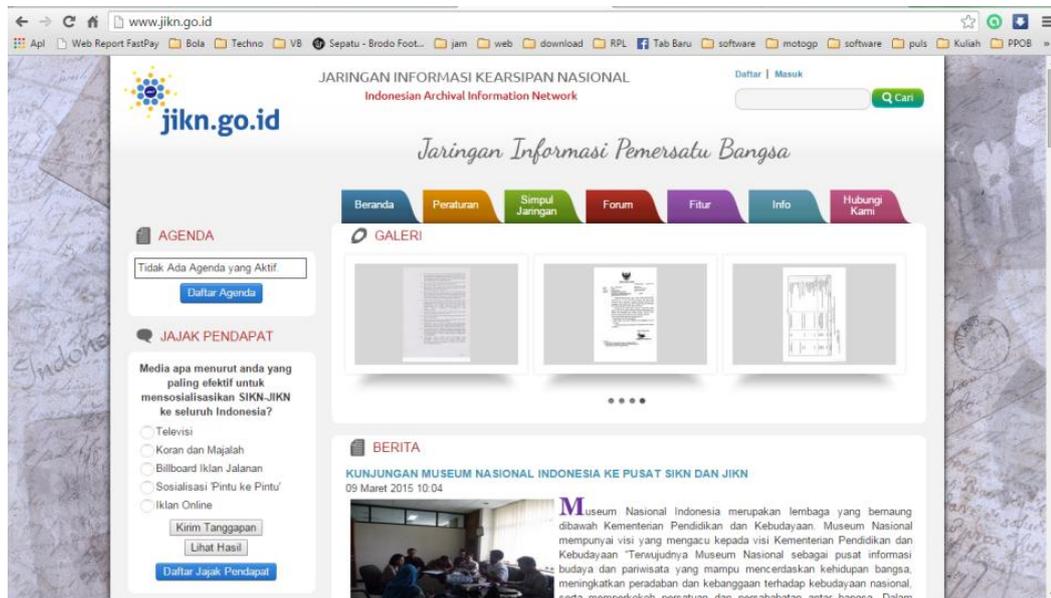
##### 2.1.1. Sistem Pengarsipan

Sistem Informasi Kearsipan ITB (Institut Teknologi Bandung) dengan alamat *website* <http://arsip.itb.ac.id> terakhir di akses pada tanggal 24 maret 2015 pukul 15.56 wib, *website* ini memiliki menu pengguna antara lain Halaman Muka, Dokumen Terbaru, Kategori, File yang terdiri dari Arsip, Disposisi, Surat Keluar, Memo dan Menu *User* untuk *login*. Sistem Informasi Kearsipan ITB disajikan pada Gambar 2.1.



Gambar 2.1. Sistem Informasi Kearsipan ITB

JIKN (Jaringan Informasi Kearsipan Nasional) dengan alamat *website* <http://www.jikn.go.id> terakhir di akses pada tanggal 24 maret 2015 pukul 16.20 wib, *website* ini memiliki menu pengguna antara lain Beranda, Peraturan, Simpul Jaringan, Forum, Fitur yang terdiri dari Info Diklat, Pencarian Arsip, Arsip Tematik, Pameran Virtual, Info dan Hubungi Kami. Jaringan Informasi Kearsipan Nasional disajikan pada Gambar 2.2.



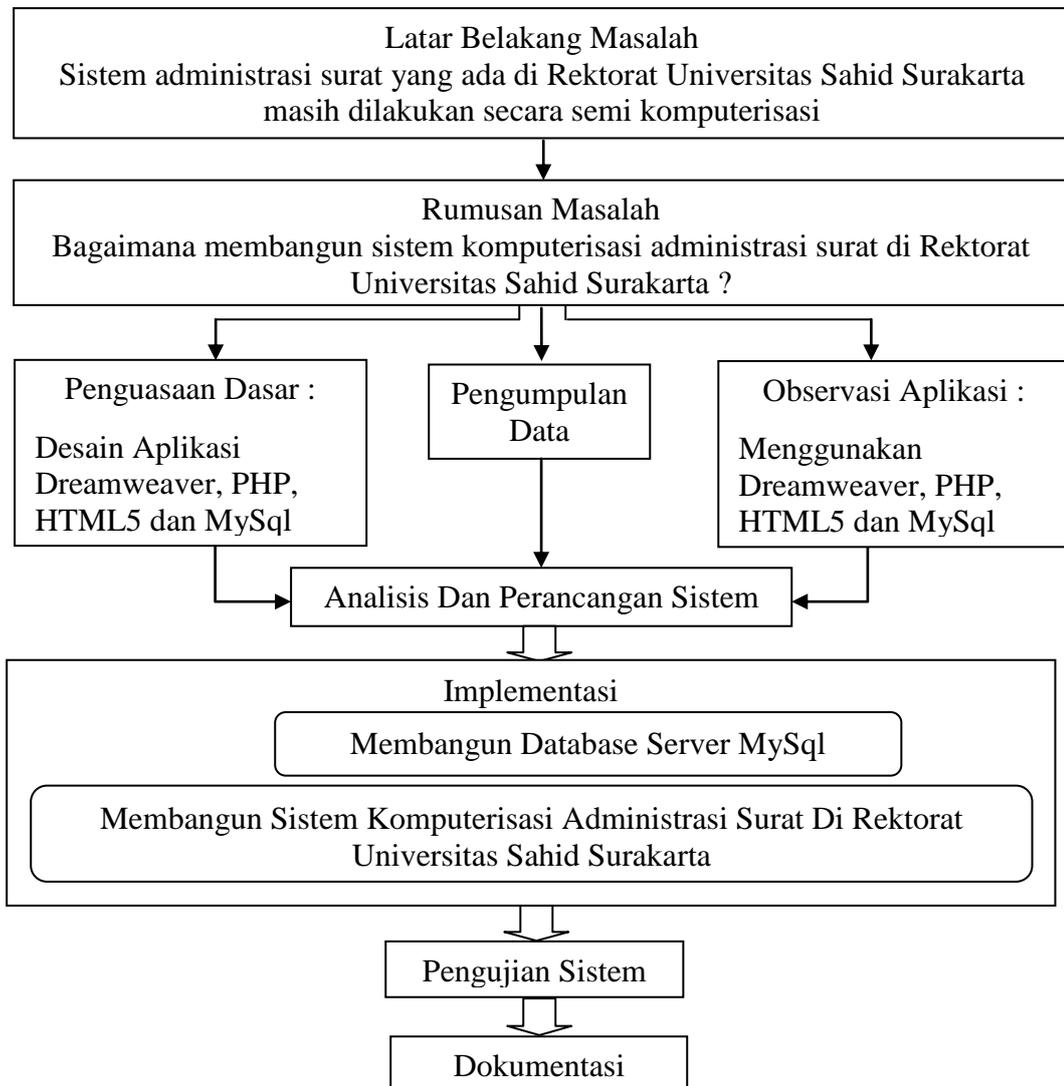
Gambar 2.2. Jaringan Informasi kearsipan Nasional

Aplikasi Pengelolaan Arsip Kementerian Kesehatan, aplikasi ini memiliki menu antara lain informasi aplikasi dan arsip yang terdiri dari menu *login*. Aplikasi Pengelolaan Arsip Kementerian Kesehatan disajikan pada Gambar 2.3.



Gambar 2.3. Aplikasi Pengolahan Arsip Kementerian Kesehatan

### 1.8. Kerangka Pemikiran



Gambar 2.4. Diagram Kerangka Pemikiran

Penjelasan kerangka pemikiran :

#### 1. Latar Belakang Masalah

Latar belakang Tugas Akhir ini adalah sistem administrasi surat di Rektorat Universitas Sahid Surakarta masih dilakukan secara semi komputerisasi.

#### 2. Rumusan Masalah

Bagaimana membangun sistem komputerisasi administrasi surat di Rektorat Universitas Sahid Surakarta ?

### 3. Pengumpulan Data Tertulis dan Tidak Tertulis

Mengumpulkan semua data yang diperlukan dalam penelitian, baik melalui *interview*, observasi maupun dokumentasi.

### 4. Penguasaan Dasar

Penulis mulai membuat tampilan atau tema dan pemrograman dasar PHP serta HTML5 menggunakan dreamweaver dan pembuatan database MySql dengan tujuan supaya lebih menguasai tentang bahasa pemrograman tersebut.

### 5. Observasi Aplikasi

Mencari beberapa aplikasi atau tinjauan pustaka yang berkaitan dengan administrasi surat baik melalui internet, karya ilmiah, buku yang dapat dijadikan referensi dalam membangun sistem komputerisasi administrasi surat di Rektorat Universitas Sahid Surakarta.

### 6. Analisis dan Perancangan Sistem

Menganalisis sistem yang berjalan saat ini serta merancang sistem yang akan di bangun seperti apa, seperti perancangan tampilan dan menu apa saja yang akan ada dalam sistem tersebut.

### 7. Implementasi

Membangun database server MySql sesuai dengan data-data yang didapatkan dengan kebutuhan sistem dan membangun sistem komputerisasi administrasi surat di Rektorat Universitas Sahid surakarta.

### 8. Pengujian Sistem

Pada tahap ini dimana sistem yang telah siap digunakan kemudian dilakukan pengujian untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada sistem yang telah di buat.

### 9. Dokumentasi

Pada tahap akhir dimana sistem telah siap digunakan di bagian Rektorat Universitas Sahid Surakarta dan membuat dokumentasi dari keseluruhan penelitian tugas akhir ini.

## **1.9. Sistem Komputerisasi Administrasi Surat**

### **2.3.1. Sistem**

Hanif Al Fatta (2007 : 3) memberikan beberapa definisi sistem secara umum :

1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa di lihat sebagai satu kesatuan yang di rancang untuk mencapai satu tujuan.

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain (Hanif Al Fatta, 2007 : 3).

### **2.3.2. Komputerisasi**

Komputerisasi adalah kegiatan pengolahan data yang sebagian besar prosesnya dilakukan menggunakan komputer, yang sudah terprogram dengan berbagai program yang akan menangani suatu aplikasi. Kegiatan komputerisasi adalah sebuah kegiatan pengelolaan data dalam rangka menghasilkan informasi-informasi penting bagi manajemen, agar yang bersangkutan mampu mengendalikan perusahaan yang menjadi tanggung jawabnya dengan baik (Muhammad Khoirul, 2013 : 80).

### **2.3.3. Administrasi**

Administrasi dalam arti sempit berasal dari kata "*administratie*" (bahasa Belanda) yang meliputi kegiatan : catat-mentcatat, surat-menyurat, pembukuan ringan, ketik-mengetik, agenda dan sebagainya yang bersifat teknis ketatausahaan / *clerical work*. Jadi tata usaha adalah bagian kecil dari kegiatan administrasi dalam arti luas. Administrasi dalam arti luas, yaitu dari kata "*administration*" (bahasa Inggris) (Maringan Masry Simbolon, 2004:14).

### **2.3.4. Surat**

Surat adalah pernyataan tertulis dari pihak lain, atas nama perseorangan ataupun nama jabatan (Rahardi, 2008:12).

## **1.10. Software Pendukung**

### **2.4.1. Macromedia Dreamweaver**

Dreamweaver merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah *website*, baik secara grafis maupun dengan menulis kode sumber secara langsung. “Adobe dreamweaver merupakan program untuk membuat atau mengedit *web* yang dikeluarkan oleh Adobe System yang juga di kenal sebagai Macromedia Dreamweaver. *Software* ini digunakan karena memiliki fitur-fitur yang menarik dan cenderung mudah dalam penggunaanya (Wahana Komputer, 2011).

### **2.4.2. Adobe Photoshop**

Adobe Photoshop adalah perangkat lunak editor citra buatan Adobe System yang dikhususkan untuk pengeditan foto atau gambar dan pembuatan efek. Perangkat ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*Market Leader*) untuk perangkat lunak pengolahan gambar atau foto dan bersama Adobe Acrobat dianggap sebagai produk terbaik yang pernah di produksi oleh Adobe System (Jennar Simartata, 2010).

### **2.4.3. XAMPP Control Panel**

XAMPP adalah 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 sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat di modifikasi *website* tanpa harus *online* atau terakses dengan internet (Wicaksono, 2008 : 7).

### **2.4.4. MySql**

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil. MySQL juga

bersifat *open source* dan *free* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali pada Windows, yang bersifat *shareware*). MySQL di distribusikan dengan lisensi *open source GPL (General Public Licence)* mulai versi 3.23, pada bulan juni 2000. *Software* MySQL bisa di unduh di <http://www.mysql.org> atau <http://www.mysql.com> (M. Rudyanto Arief, 2011 : 151).

#### **2.4.5. HTML (Hypertext Markup Language)**

HTML atau Hypertext Markup Language merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini di kenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam *web browser*. (M. Rudyanto Arief, 2011 : 23).

#### **2.4.6. PHP**

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side scripting* maka *sintaks* dan perintah-perintah PHP akan di eksekusi di server kemudian hasilnya dikirimkan ke *browser* dalam format HTML. Dengan demikian kode program yang di tulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan *web* lebih terjamin. PHP di rancang untuk membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web* (M. Rudyanto Arief, 2011 : 43).

### **1.11. Analisis Sistem**

Analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem menjadi fondasi menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini sangat penting karena menentukan bentuk sistem yang harus di bangun. Tahapan ini bisa merupakan tahap yang mudah jika klien sangat paham dengan masalah yang di hadapi dalam organisasinya dan tahu betul fungsionalitas dari sistem informasi yang akan di buat. Tetapi tahap ini bisa

menjadi tahap yang paling sulit jika klien tidak bisa mengidentifikasi kebutuhannya atau tertutup terhadap pihak luar yang ingin mengetahui detail proses-prose bisnisnya (Hanif Al Fatta, 2007 : 44).

### **1.12. Perancangan Sistem**

Perancangan Sistem adalah merancang atau mendesain suatu sistem yang baik, yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem (Al-bahra Bin Ladjamuddin, 2005).

### **1.13. Metode Perancangan Sistem Berorientasi Objek**

Metode Berorientasi Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya (Adi Nugroho, 2005).

Sebuah sistem yang di bangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya di bungkus (*dienkapsulasi*) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya serta dapat berinteraksi satu sama lainnya (Arif Rachman, dkk, 2012 : 2).

Metode perancangan metodologi berorientasi objek menggunakan diagram UML (*Unified Modeling Language*) adalah sebuah “bahasa” yang telah menjadi *standart* dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak (Adi Nugroho, 2002).

#### **2.7.1. Use Case Diagram**

*Use case diagram* menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya di lihat dari luar oleh *actor* (keadaan lingkungan sistem yang di lihat user) dan bukan bagaimana fungsi yang ada dalam sistem (Dana S. Kusumo 2004:3). Simbol-simbol yang digunakan pada *use case diagram* disajikan pada Tabel 2.1.

Tabel 2.1. Tabel simbol *use case diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Aktor</i>	Idealization orang eksternal, proses, atau hal yang berinteraksi dengan sistem, subsistem atau kelas.
2.		<i>Use case</i>	Sebuah <i>use case</i> menggambarkan interaksi dengan <i>actor</i> sebagai urutan pesan antara sistem dan aktor satu atau lebih.
3.		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
4.		<i>Generalization</i>	Hubungan antara <i>use case</i> umum dan <i>use case</i> yang lebih spesifik yang mewarisi dan menambahkan fitur.
5.		<i>Comunication Association</i>	Jalur komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi didalam.
6.		<i>Extend</i>	Penyisipan perilaku tambahan ke dalam basis <i>use case</i> yang tidak tahu tentang hal itu.
7.		<i>include</i>	Penyisipan perilaku tambahan ke dalam basis <i>use case</i> yang secara eksplisit menggambarkan penyisipan.

### 2.7.2. Class Diagram

*Class diagram* menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang di tangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram* (Dana S. Kusumo 2004:3). Simbol-simbol yang digunakan pada *class diagram* disajikan pada Tabel 2.2.

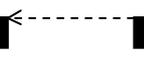
Tabel 2.2. Tabel simbol *class diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagai perilaku dan struktur data dari objek yang ada di atas objek induk ( <i>ancestor</i> ).
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang terbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan 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 yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yan tidak mandiri.
7.		<i>Association</i>	Untuk menghubungkan objek satu dengan objek yang lainnya.

### 2.7.3. *Sequence Diagram*

*Sequence diagram* menggambarkan kolaborasi dinamis antara sejumlah objek. Kegunaan untuk menunjukkan rangkaian pesan yang di kirim Antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *sequence diagram* disajikan pada Tabel 2.3.

Tabel 2.3. Tabel simbol *sequence diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Message</i>	<i>Message</i> ditampilkan sebagai anak panah dari lifeline dari satu objek ke objek yang lain.
3.		<i>Return</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

#### 2.7.4. Activity Diagram

*Activity diagram* menggambarkan rangkaian aliran aktifitas, digunakan untuk mendeskripsikan aktifitas yang di bentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *use activity diagram* disajikan pada Tabel 2.4.

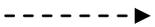
Tabel 2.4. Tabel simbol *activity diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Actifity State</i>	Aktivitas yang mewakili pelaksanaan dalam pernyataan dalam prosedur atau pelaksanaan kegiatan dalam alur kerja.
2.		<i>Branch/Merge</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.		<i>Initial State</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
4.		<i>Final State</i>	Status akhir yang dilakukan sistem.
5.		<i>Fork/Join</i>	Sebuah control yang ditampilkan dengan cara yang sama seperti statechart, oleh beberapa panah masuk

### 2.7.5. Component Diagram

*Component diagram* menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *sourcecode*, 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* (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *component diagram* disajikan pada Tabel 2.5.

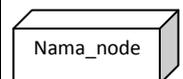
Tabel 2.5. Tabel simbol *component diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Component</i>	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel di sebelah kirinya.
2.		<i>Dependency</i>	Sebuah <i>Dependency</i> digunakan untuk menotasikan relasi antara dua komponen.

### 2.7.6. Deployment Diagram

*Deployment Diagram* menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang di eksekusi oleh *node* tertentu dan ketergantungan komponen (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *deployment diagram* disajikan pada Tabel 2.6.

Tabel 2.6. Tabel simbol *deployment diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Component</i>	Komponen-komponen yang ada diletakkan di dalam <i>node</i> .
2.		<i>Node</i>	<i>Node</i> menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem.
3.		<i>Association</i>	Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara element-elemen <i>hardware</i> .

#### 1.14. Metode Pengujian Sistem

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi transaksi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang tipikal, kritis, atau *abnormal*. Isu kunci dalam pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil, dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus menyangkut unit *testing*, yang mengecek *validasi* dari prosedur dan fungsi-fungsi secara *independent* dari komponen sistem yang lain. Kemudian modul *testing* harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan (Hanif Al Fatta, 2007 : 170-171). Kategori *test* yang dilakukan pada pengujian ini adalah unit *testing*.

##### 2.8.1. Unit Testing

Unit *testing* merupakan pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Metode yang digunakan untuk melakukan unit *testing* yaitu *black box testing*.

#### **2.8.1.1. *Black Box Testing***

*Black box testing* terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian di amati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.