

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Sekolah-sekolah yang menyediakan perpustakaan di Indonesia, masih ada yang mengolah data mereka menggunakan sistem manual atau belum terkomputerisasi, oleh sebab itu sistem administrasi perpustakaan lebih efektif jika menggunakan sistem komputerisasi. Sistem terkomputerisasi perpustakaan sudah banyak dikembangkan sebelumnya dengan metode dan perangkat lunak yang berbeda-beda.

Jenar Kuswidiardi (2012) mengembangkan aplikasi tentang “*Sistem informasi perpustakaan berbasis Microsoft Visual Basic 6.0 dengan Database SQL Server 2000 di perpustakaan SMK YPKK 1 Sleman*”. Pengembangan aplikasi ini menggunakan metode *Research and Development (R&D)*. Aplikasi ini dibangun menggunakan *software Microsoft Visual Basic 6.0 dan Microsoft SQL Server 2000*. Sistem yang dibangun memudahkan petugas perpustakaan dalam mendata siswa yang ingin meminjam buku di perpustakaan tersebut.

Herbowo Tri Nugroho dan Bhekti Pamilih (2011) mengembangkan aplikasi tentang “*Perancangan Sistem Informasi Perpustakaan Pada SMP N 1 Seyegan dengan Menggunakan Java*”. Mereka mengembangkan aplikasi dengan menggunakan metode iterasi dan menggunakan diagram ERD, DFD untuk pemodelan aplikasi yang dibangun dengan menggunakan *software Netbeans 6.1.7* dan menggunakan *MySQL* sebagai basis data. Sistem yang dibangun membuahkan hasil yaitu aplikasi *desktop* yang dapat membantu para anggota perpustakaan dalam mendata siswa yang hendak meminjam buku di perpustakaan tersebut.

Barra Sulthani (2008) mengembangkan aplikasi tentang “*Sistem Informasi Perpustakaan Fakultas Kedokteran dan Kesehatan Universitas Muhammadiyah Jakarta*”. Pengembangan aplikasi ini menggunakan metode Pengumpulan data. Seperti wawancara, pengamatan, dan studi pustaka. Pemodelan aplikasi ini,

dibangun dengan menggunakan bahasa pemrograman *PHP* dan menggunakan *MySQL* sebagai basis data. Sistemnya menghasilkan sebuah aplikasi *desktop* untuk mempermudah petugas perpustakaan mendata siswa peminjam buku diperpustakaan tersebut. Sistem informasi di perpustakaan beberapa tahun terakhir ini semakin meningkat dengan adanya teknologi baru seperti sistem informasi berbasis web pada perpustakaan untuk membantu pengunjung maupun pegawai perpustakaan itu sendiri dalam memenuhi kebutuhannya, hal tersebut tidak terlepas dari semakin mudahnya untuk mendapatkan akses ke suatu teknologi informasi secara cepat. Adanya hal tersebut timbul ketertarikan sebagian besar orang untuk melakukan penelitian terkait evaluasi kinerja sistem informasi perpustakaan.

Mahdia (2011) dalam sebuah penelitian dengan judul “Evaluasi Kinerja SLiMS Sebagai Sarana Temu Kembali Informasi di Perpustakaan DPR RI” evaluasi kinerja SLiMS ini di fokuskan terhadap keefektifan *Online Public Access Catalog* (OPAC) sebagai sarana temu kembali informasi. *Public Access Catalog* (OPAC) sering di sebut sebagai katalog dari sebuah perpustakaan yang berisikan daftar-daftar buku yang ada diperpustakaan untuk memudahkan pengunjung dan pegawai perpustakaan untuk mengetahui letak atau posisi buku yang sedang dicari. Semua informasi mengenai buku yang diinginkan akan mudah untuk ditemukan menggunakan OPAC. Untuk mengevaluasi kinerja dari OPAC peneliti mencoba untuk mengupas sejauh mana tingkat *recall* dan *precision*, *respon's time*, upaya pengguna, dan segi penyajian pada software SLiMS di perpustakaan DPR RI.

Cahyono (2013) dalam sebuah penelitian dengan judul “Analisis Pemanfaatan Senayan Library Management System (SLiMS) di Kantor Perpustakaan dan Arsip Daerah Kota Salatiga” analisis ini bertujuan untuk mengetahui pemanfaatan teknologi informasi dari SLiMS yang diterapkan di Perpustakaan dan Arsip Daerah Kota Salatiga sejak 2010. Teori *Technology Acceptance Model* (TAM) digunakan untuk mengetahui persepsi pengguna terhadap sistem yang akan mempengaruhi sikap pengguna, dengan

menggunakan 2 buah *variable* yaitu kemanfaatan (*usefulness*) dan kemudahan penggunaan (*case of use*).

Kelima penelitian sebelumnya, aplikasi yang dibuat memberikan kemudahan dalam mendata siswa yang ingin meminjam buku pada perpustakaan, sesuai keinginan pengguna. Namun dari ketiga aplikasi yang telah mereka buat, hanya mempermudah petugas perpustakaan dalam mendata peminjam buku. Berdasarkan penelitian yang sudah ada, maka penulis menambah beberapa spesifikasi yang menjadi perbedaan dengan sistem yang telah dikembangkan.

Persamaan dari kelima penelitian sebelumnya adalah sebagai berikut :

1. Aplikasi dapat mendata mahasiswa yang ingin meminjam buku di perpustakaan tanpa menulis di buku tamu.
2. Aplikasi dapat mendata buku yang ada di perpustakaan.
3. Aplikasi dapat mendata karyawan yang bertugas pada perpustakaan.
4. Aplikasi dapat membuat laporan transaksi pengembalian buku.
5. Aplikasi dapat menampilkan informasi buku (Katalog).
6. Aplikasi dapat memberi informasi keterlambatan pengembalian buku.

Pembeda dari penelitian ini dengan kelima penelitian sebelumnya antara lain :

1. Pembaharuan menu katalog yang lebih mudah digunakan.
2. Studi kasus jelas berbeda.

## **2.2 Landasan Teori**

### **2.2.1 Pengertian Sistem**

Pengertian Sistem adalah kumpulan unsur-unsur yang bergabung menjadi satu kesatuan dan mempunyai tujuan yang sama. Unsur-unsur dalam sistem tersebut saling berhubungan satu sama lain untuk memudahkan arus informasi agar dicapai suatu tujuan bersama. Di dalam sistem terdapat unsur-unsur penggerakannya sehingga penggerak tersebut saling berinteraksi satu sama lain untuk mencapai tujuan yang diinginkan. (Jogiyanto 2010)

### **2.2.2 Sistem Informasi**

Sistem informasi merupakan suatu perkumpulan data yang terorganisasi beserta tatacara penggunaannya yang mencangkup lebih jauh dari pada sekedar penyajian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tatacara penggunaannya. Keberhasilan suatu sistem informasi yang diukur berdasarkan maksud pembuatannya tergantung pada tiga faktor utama, yaitu : keserasian dan mutu data, pengorganisasian data, dan tatacara penggunaannya. Untuk memenuhi permintaan penggunaan tertentu, maka struktur dan cara kerja sistem informasi berbeda-beda bergantung pada macam keperluan atau macam permintaan yang harus dipenuhi. Suatu persamaan yang menonjol ialah suatu sistem informasi menggabungkan berbagai ragam data yang dikumpulkan dari berbagai sumber. Untuk dapat menggabungkan data yang berasal dari berbagai sumber suatu sistem alih rupa (transformation) data sehingga jadi tergabungkan (compatible). Berapa pun ukurannya dan apapun ruang lingkungannya suatu sistem informasi perlu memiliki ketergabungan (compatibility) data yang disimpannya. (Hanif Al Fatta, 2009:9).

### **2.2.3 Pengertian Perpustakaan Digital**

Perpustakaan digital itu tidak berdiri sendiri, melainkan terkait dengan sumber-sumber informasi lain dan pelayanan informasinya terbuka bagi pengguna di seluruh dunia. Koleksi perpustakaan digital tidak terbatas pada dokumen elektronik pengganti bentuk tercetak saja, ruang lingkup koleksinya malah sampai pada artefak digital yang tidak bisa tergantikan oleh bentuk cetak. (Widyawan, 2005)

### **2.2.4 Pengertian Web**

Pengertian *web* menurut Hakim Lukmanul adalah fasilitas internet yang menghubungkan dokumen dalam lingkup local maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari suatu *page* ke *page* lain (*hyper text*), baik diantara

page yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca melalui *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi browser lainnya.

### **2.2.5 Metode Pengujian Menggunakan *WebQual***

Menurut Sanjaya (2012) *WebQual* merupakan salah satu metode pengukuran *website* berdasarkan persepsi pengguna akhir, yang dikembangkan oleh Stuart Barnes & Richard Vidgen. *WebQual* berdasar pada konsep *Quality Function Deployment* (QFD) yaitu suatu proses yang berdasar pada “*voice of customer*” dalam pengembangan dan implementasi suatu produk atau jasa. Dari konsep QFD tersebut, *WebQual* disusun berdasar pada persepsi pengguna akhir (*end user*) terhadap suatu *website*. *WebQual* telah mengalami beberapa iterasi dalam penyusunan kategori dan butir – butir pertanyaannya. Versi terbaru adalah *WebQual* 4.0 yang menggunakan tiga dimensi untuk mewakili kualitas dari *website*, yaitu dimensi kemudahan penggunaan (*Usability Quality*), dimensi kualitas informasi (*Information Quality*), dan kualitas interaksi (*Interaction Quality*).

### **2.2.6 Perpustakaan Universitas Sahid Surakarta**

Selaras dengan sejarah pendirian USAHID, guna menunjang kegiatan akademis perpustakaan menjadi hal yang wajib dimiliki oleh sebuah Universitas seperti pada pasal 55 Peraturan Pemerintah No. 17 Tahun 2010 tentang Pengelolaan Penyelenggaraan tentang Pendidikan Tinggi, menyebutkan bahwa salah satu syarat untuk penyelenggaraan perguruan tinggi harus memiliki perpustakaan. Meskipun hanya sebagai unit pelaksana teknik yang merupakan unsur penunjang sebagai kelengkapan bagi pendidikan, penelitian dan pengabdian kepada masyarakat, keberadaan perpustakaan cukup penting dalam mendukung kegiatan Tri Dharma perguruan tinggi, dimana secara umum tugas perpustakaan perguruan tinggi adalah melakukan tugas rutin untuk mengadakan, mengolah dan merawat bahan pustaka serta mendayagunakan untuk kepentingan sivitas akademika pada khususnya dan masyarakat pada umumnya.

Pendirian perpustakaan Universitas Sahid Surakarta pada awalnya bertempat satu ruang dengan Bagian Administrasi Akademik (B.A.A.K), karena alasan perluasan ruangan kemudian pada tahun 2005 perpustakaan Universitas Sahid Surakarta dan Sekolah Tinggi Pariwisata Sahid Surakarta disatukan dengan alasan efisiensi penggunaan ruang dengan luas ruangan 64 m<sup>2</sup>. Penggabungan perpustakaan kedua instansi tersebut hanya bertahan 1 (satu) tahun karena pada tahun 2006 perpustakaan Universitas Sahid Surakarta dipindah berdekatan dengan ruang Bagian Administrasi Akademik (B.A.A.K) dengan luas ruang 4 x 8 m. Sebelum akhirnya perpustakaan dipindah ke lantai 3 (tiga) pada tahun 2011 sampai sekarang ini, perpustakaan sempat dipindah ke laboratorium komputer selama 2 (dua) bulan.

Kegiatan pengembangan dimulai dengan ditata ulang ruang perpustakaan. Perpustakaan Universitas Sahid Surakarta kini mempunyai 2 ruangan dengan luas masing-masing 81 m<sup>2</sup> yang terdiri dari ruang koleksi umum dan ruang koleksi skripsi.

Kegiatan pengolahan dan pelayanan bahan pustaka mulai berkembang dari sistem manual ke sistem otomatisasi perpustakaan dengan menggunakan SLiMS (*Senayan Library Management System*) 3-stable 14 untuk aktivitas di perpustakaan dan SLiMS Cendana untuk *Digital Library*. Serta sudah tersedianya OPAC (*Online Public Access Catalogue*) yang memudahkan pemustaka dalam menelusur informasi.

#### **2.2.1.1. Visi Dan Misi**

##### **1. VISI**

Visi Perpustakaan Universitas Sahid Surakarta adalah mewujudkan Perpustakaan Universitas Sahid Surakarta sebagai pusat sumber informasi, ilmu pengetahuan dan budaya lokal dalam menunjang proses belajar mengajar di lingkungan Universitas Sahid Surakarta.

##### **2. MISI**

- a. Menunjang pelaksanaan Tridharma Perguruan Tinggi di Universitas Sahid Surakarta

- b. Memberikan layanan informasi kepada pemustaka dengan menerapkan sistem informasi perpustakaan yang berbasis teknologi informasi.
- c. Mengadakan pengolahan dan pengembangan informasi dari koleksi yang dimiliki dan menyajikan hasilnya dengan memanfaatkan teknologi dan sistem sebagai bagian integral proses belajar mengajar dalam lingkungan Universitas Sahid Surakarta.
- d. Menumbuhkan iklim akademis di sivitas Universitas Sahid Surakarta melalui pengembangan :
  - a) Budaya membaca
  - b) Kemampuan berdiskusi
  - c) Budaya menulis
- e. Memasyarakatkan budaya ilmiah di sivitas Universitas Sahid Surakarta melalui:
  - a) Publikasi karya ilmiah dari hasil penelitian Dosen Universitas Sahid Surakarta
  - b) Menumbuhkan atensi masyarakat kampus Universitas Sahid Surakarta terhadap peristiwa ilmiah dan pengkajian ilmiah.

#### **2.2.1.2. TENAGA PERPUSTAKAAN**

Jumlah staf Perpustakaan Universitas Sahid Surakarta sebanyak 2 orang, terdiri dari ibu Erna Setyaningsih sebagai kepala perpustakaan dan bapak Sunu Wahyudi sebagai staff perpustakaan.

#### **2.2.7 Bahasa pemrograman Java**

Menurut M. Shalauddin dan Rosa A.S. (2010), Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan.

Java berdiri di atas sebuah mesin penterjemah (*interpreter*) yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca kode bit (*bytecode*) dalam file *.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai

bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (*platform independent*). Itulah yang menyebabkan dalam dunia pemrograman *Java* dikenal adanya istilah *write once, run everywhere*”, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (*platform*) manapun, tanpa harus melakukan perubahan kode program.

### **2.2.8 Pemrograman Berorientasi Objek (*Object Oriented Programming*)**

Menurut Rosa A.S dan M.Shalahuddin (2016), metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Menurut Rosa A.S dan M.Shalahuddin (2016) 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.

### **2.2.4 Basis Data (*Database*)**

Menurut Rosa A.S dan M. Shalahuddin (2014) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat di butuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks ataupun

*Database Management System (DBMS).*

Kebutuhan basis data dalam sistem informasi meliputi memasukan, menyimpan, dan mengambil data, membuat laporan berdasarkan data yang telah disimpan.

Salah satu aplikasi yang sering di gunakan yaitu *MySQL*, *MySQL* adalah sebuah implementasi dari sistem manajemen basis data relasional yang di distribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tidak boleh dijadikan produk turunan yang bersifat komersial. Berikut beberapa keistimewaan dari *MySQL* menurut beberapa sumber sebagai bahan pertimbangan :

- a) *Portabilitas*, maksudnya adalah *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
- b) Perangkat lunak sumber terbuka (*open source*), maksudnya adalah *MySQL* didistribusikan sebagai open source sehingga dapat digunakan secara gratis.
- c) *Multi-user*, maksudnya adalah *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah.
- d) *Performance tuning*, maksudnya adalah *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.
- e) *Ragam tipe data*, maksudnya adalah *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed or unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
- f) *Perintah dan Fungsi*, maksudnya adalah *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
- g) *Keamanan*, maksudnya adalah *MySQL* memiliki beberapa lapisan keamanan seperti password yang terenkripsi.
- h) *Skalabilitas dan Pembatasan*, maksudnya adalah *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60

ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

- i) *Konektivitas*, maksudnya adalah *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP*, *Unix Soket (UNIX)*, atau *named pipes (NT)*.
- j) *Lokalisasi*, maksudnya adalah *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
- k) *Antar Muka*, maksudnya adalah *MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *API (Application Programming Interface)*.

### **2.2.5 UML (*Unified Modelling Language*)**

UML (*Unified Modelling Language*) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. (Nugroho 2010)

UML ini menggunakan *software* yaitu *Edraw max*, *Edraw max* adalah *software* untuk penggambaran sebuah susunan dalam perancangan sistem. *Software* ini dikeluarkan oleh *Edrawsoft* yang kini sudah tersebar ke penjuru dunia. *Software* ini dirancang khusus untuk para developer dalam pengembangan sistem. Penggunaannya pun sangat mudah karena kita sudah di berikan banyak pilihan fitur.

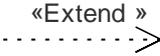
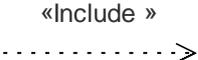
Ada 5 (lima) macam diagram dalam *Unified Modeling Language (UML)*, yaitu :

#### **2.2.5.1. Use Case Diagram**

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Mereka memisahkan sistem menjadi actor dan *end user*. Aktor mewakili peran yang dimainkan oleh pengguna

system. Pengguna tersebut bisa menjadi manusia, komputer lain, perangkat keras atau bahkan perangkat lunak lainnya. Satu-satunya kriteria adalah bahwa mereka harus berada di luar sistem yang dimasukkan menjadi *use case*. *Use case* mewakili kegiatan yang dilakukan actor dengan bantuan sistem dalam mencapai sebuah tujuan dalam sistem tersebut. Actor Perlu mendefinisikan apa yang dibutuhkan dari system. *Use case* harus mencerminkan kebutuhan dan sasaran pengguna, dan harus diprakarsai oleh seorang actor. Simbol-simbol *use case* dapat dilihat pada Tabel 2.2.

Tabel 2.2 simbol *Use Case Diagram*

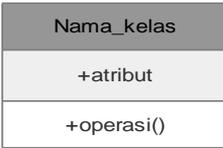
No	Simbol	Deksripsi
1	<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan menggunakan kata kerja di awal di awal frase nama <i>use case</i>
2	<p>Aktor</p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun aktor adalah gambar orang, tapi belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3	<p>Asosiasi</p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4	<p><i>Extend</i></p> 	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; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i>
5	<p>Generalisasi</p> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
6	<p><i>Include</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya

### 2.2.5.2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari sisi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. (Rosa A.S dan M.Shalahuddin, 2016)

Simbol-simbol *class diagram* dapat dilihat pada Tabel 2.3.

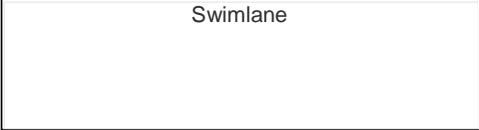
Tabel 2.3 simbol *class Diagram*

No	Simbol	Keterangan
1	Kelas 	Kelas pada struktur sistem
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4	Asosiasi Berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
7	Agregasi 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

### 2.2.5.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol *activity diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 simbol *Activity Diagram*

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem , sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas 	Aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja
3	<i>Decision</i> / Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	<i>Join</i> / penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

### 2.2.5.4. Deploymet Diagram

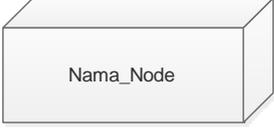
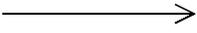
Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut :

- a) Sistem tambahan (*embedded sytem*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.

- b) Sistem *client / server*
- c) Sistem terdistribusi murni
- d) Rekayasa ulang aplikasi

Simbol-simbol *Deployment Diagram* dapat dilihat pada Tabel 2.5.

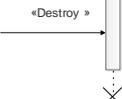
Tabel 2.5 simbol *Deployment Diagram*

No	Simbol	Deskripsi
1	<p><i>Package</i></p> 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
2	<p>Node</p> 	Biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
3	<p><i>Dependency</i> / kebergantungan</p> 	Kebergantungan antar node, arah panah mengarah pada node yang dipakai
4	<p><i>Link</i></p> 	Relasi antar node

#### 2.2.5.5. *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Simbol-simbol *Sequence Diagram* dapat dilihat pada tabel 2.6

Table 2.6 simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor  Actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun aktor adalah gambar orang, tapi belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
2	Garis Hidup/ <i>Lifeline</i> 	Menyatakan kehidupan suatu objek
3	Objek 	Menyatakan objek yang berinteraksi pesan
4	Waktu Aktif 	Meyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya
5	Pesan Tipe <i>Create</i> «Create » 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6	Pesan Tipe <i>Call</i> 1: nama_metode() 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi / metode
7	Pesan Tipe <i>Send</i> 1: masukan 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8	Pesan Tipe <i>Return</i> 1: keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	Pesan Tipe <i>Destroy</i> «Destroy » 	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>

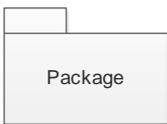
### 2.2.5.6. Component Diagram

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi atau ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. *Source code* program perangkat lunak
- b. Komponen *executable* yang dilepas ke user
- c. Basis data secara fisik
- d. Sistem yang harus beradaptasi dengan sistem lain

Simbol-simbol *component diagram* dapat dilihat pada Tabel 2.7

Tabel 2.7 simbol *Component Diagram*

No	Simbol	Deskripsi
1	<i>Package</i> 	<i>Package</i> merupakan sebuah bungkusuan dari satu atau lebih komponen
2	Komponen 	Komponen sistem
3	<i>Depedency / kebergantungan</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4	<i>Interface / interface</i>  Nama_interface	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