



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1. Metode Pengumpulan Data

##### 2.1.1. Wawancara

Wawancara adalah proses tanya jawab lisan dimana dua orang atau lebih bertatap muka secara fisik untuk mengetahui tanggapan, pendapat, dan motivasi seseorang terhadap suatu obyek. Wawancara juga dapat digunakan sebagai cara untuk mengumpulkan data dengan tanya jawab sepihak yang dikerjakan secara sistematis dan berlandaskan kepada tujuan penelitian.

Wawancara dapat digunakan sebagai metode pengumpulan data primer, metode pelengkap, dan metode pendukung untuk menguji kebenaran sebuah hipotesis yang telah didapatkan sebelumnya. Wawancara sendiri terdiri dari 2 struktur, yaitu wawancara terstruktur dan wawancara tidak terstruktur. Wawancara terstruktur menggunakan pedoman wawancara sedangkan wawancara tidak terstruktur hanya memiliki hal penting sebagai pedoman. (Soegijono, 1993)

Penggunaan wawancara sebagai metode pengumpulan data memiliki beragam kelebihan. Berikut ini adalah kelebihan menggunakan metode wawancara yang berkaitan dengan penelitian:

1. *Flexibility*. Pewawancara dapat mengajukan pertanyaan sesuai dengan situasi yang dihadapi pada waktu itu. Apabila hasil yang didapat dirasa belum memuaskan, maka pewawancara dapat

melakukan pendalaman (*probing*). Bahkan jika sebuah pertanyaan dirasa kurang tepat atau kurang kontekstual, maka pewawancara tersebut dapat menundanya.

2. *Response Rate*. Wawancara dapat ditanggapi secara lebih baik dibandingkan dengan kuesioner yang disebar. Responden yang tidak mampu membaca atau menulis tetap dapat menjawab pertanyaan, demikian juga responden yang malas menulis. Banyak responden yang lebih menyukai mengeluarkan pandangan secara lisan dibanding tulisan.
3. *Nonverbal Behavior*. Pewawancara dapat mengobservasi perilaku nonverbal pada responden, seperti rasa suka atau tidak suka, atau perilaku lainnya saat pertanyaan diajukan kepada responden.

Penggunaan wawancara sebagai metode pengumpulan data dapat berhasil apabila peneliti memperhatikan aspek peranan pewawancara, peranan responden, dan teknik melakukan wawancara. Ketika peneliti memperhatikan ketiga aspek tersebut, maka kesalahan di dalam pengumpulan data dapat ditekan serendah mungkin (Bailey, 1978).

### **2.1.2. Kuesioner**

kuesioner atau biasa juga disebut daftar pertanyaan adalah sebuah set pertanyaan yang secara logis berhubungan dengan masalah penelitian, dan tiap pertanyaan merupakan jawaban jawaban yang mempunyai makna dalam menguji hipotesis. Daftar pertanyaan tersebut sebisa mungkin dibuat secara terperinci dan lengkap.

Kuesioner atau angket adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respon sesuai dengan permintaan pengguna (Arikunto, 2009). Kuesioner dapat dibedakan menjadi 3 jenis, yaitu:

1. Kuesioner terbuka yang memungkinkan responden memberikan jawaban sesuai dengan kehendak dan keadaannya. Kuesioner jenis ini digunakan apabila peneliti belum dapat memperkirakan atau menduga kemungkinan alternatif yang ada pada responden.
2. Kuesioner tertutup adalah kuesioner yang sudah memberikan pilihan terlebih dahulu pada setiap pertanyaan yang diberikannya, sehingga responden hanya perlu memilih dari pilihan yang ada pada tempat yang disediakan.
3. Kuesioner campuran adalah kuesioner yang jenis pertanyaannya terdiri dari kuesioner terbuka dan tertutup.

Penggunaan kuesioner memiliki kelebihan dan kekurangan yang terdapat pada tabel 2.1.

U  
M  
N

Tabel 2.1. Kelebihan dan Kekurangan Kuesioner

| Kelebihan  | Kekurangan   |
|--|--|
| <ul style="list-style-type: none"> <li>- Tidak memerlukan kehadiran peneliti.</li> <li>- Dapat dibagikan serentak kepada responden.</li> <li>- Dapat dijawab oleh responden dengan menyesuaikan waktu yang dimiliki responden.</li> <li>- Dapat dibuat secara anonim.</li> <li>- Memiliki standar yang sama karena setiap responden dihadapkan dengan pertanyaan yang sama.</li> </ul> | <ul style="list-style-type: none"> <li>- Responden seringkali tidak teliti sehingga memungkinkan adanya pertanyaan yang terlewat.</li> <li>- Ada kemungkinan kesulitan mencari validitas jawabannya</li> <li>- Responden mungkin memberikan jawaban yang tidak jujur atau tidak betul</li> <li>- Cakupan respondennya terbatas.</li> </ul> |

### 2.1.3. Observasi

Observasi merupakan teknik atau pendekatan untuk mendapatkan data primer dengan cara mengamati langsung objek datanya. Observasi merupakan cara pengumpulan data melalui proses pencatatan perilaku subjek, objek, atau kejadian yang sistematis tanpa adanya pertanyaan atau komunikasi dengan individu yang diteliti. Observasi sebagai teknik pengumpulan data tidak terbatas hanya kepada orang, tetapi juga pada objek-objek lainnya, seperti interaksinya terhadap individu lain (Hadi, 1986).

Observasi merupakan suatu proses yang kompleks dan tersusun dari berbagai proses biologis dan psikologis. Teknik penumpulan data dengan

cara observasi digunakan apabila penelitian berkaitan dengan perilaku manusia, proses kerja, gejala alam, dan bila responden yang diamati tidak terlalu besar. Observasi digunakan untuk memperoleh data dari tangan pertama dengan mengamati orang dan tempat pada saat dilakukan penelitian (Creswell, 2012).

Pada dasarnya teknik observasi digunakan untuk melihat dan mengamati fenomena sosial yang tumbuh dan berkembang yang kemudian dapat dilakukan perubahan atas penilaian tersebut bagi pelaksana observasi sehingga mampu memisahkan antara yang diperlukan dan tidak diperlukan (Margono, 2007).

## **2.2. Teknik Pengolahan Data**

### **2.2.1 Rapid Application Development**

Metode *Rapid Application Development* (RAD) memproses *requirement* yang didapat dari pengguna untuk menentukan *key feature*, kemudian membuat desain sistem hingga menjadi aplikasi yang dapat digunakan oleh klien. Metode RAD banyak membantu perusahaan membangun program lebih cepat dibandingkan dengan metode tradisional seperti *Waterfall*. Para *programmer* bisa menghemat waktu dengan menggunakan metode RAD karena metode ini menggunakan metode *prototyping* dan visualisasi sehingga dapat membuat *Graphical User Interface* (GUI) lalu kemudian *backend* dari aplikasi tersebut. Meskipun lebih cepat, namun metode ini tidak sesuai dengan beberapa fase yang

biasanya ada di dalam metode *Waterfall* atau metode tradisional lainnya (Mary, 1995).

Metode RAD / *prototyping* melibatkan model aplikasi yang akan dibuat. Kelebihan menggunakan metode RAD selain kecepatan pembuatan aplikasi juga dapat menyesuaikan dengan *requirement* yang berubah. Selain itu, calon pengguna juga akan dilibatkan dalam proses pembangunan aplikasi ini. Kekurangan metode ini meliputi kecenderungan pengguna yang mungkin akan terlalu mengikuti contoh dari *prototype* yang sudah dibuat sebelumnya, dan juga dokumentasi program menjadi kurang terfokus (Stuart, 1991).

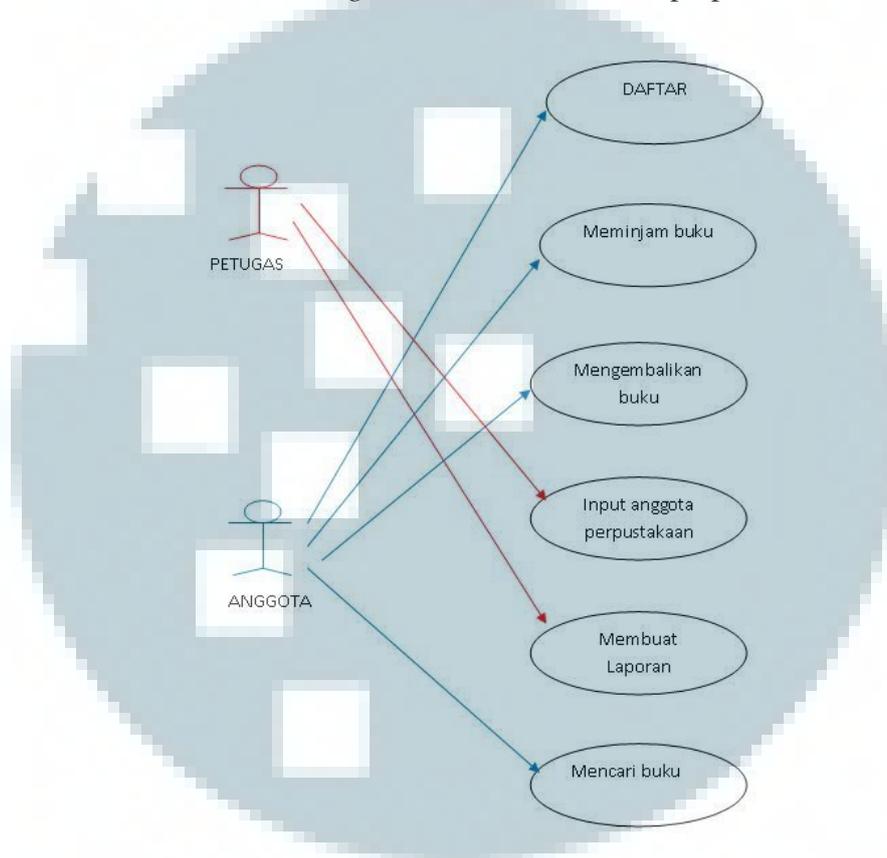
### **2.2.2 Unified Modeling Language**

*Unified Modeling Language* (UML) adalah sebuah bahasa yang memiliki standar tertentu untuk memvisualisasikan sebuah logika aplikasi atau sistem yang bisa dikondisikan kepada beberapa situasi tertentu (Addison-Wesley, 2005). Hasil akhir dari UML adalah *use case diagram*, *activity diagram*, dan *sequence diagram*.

#### **1. Use Case Diagram**

*Use case diagram* adalah diagram yang menggambarkan interaksi antara sebuah sistem dengan *user* atau dengan sistem lainnya. Diagram ini juga dapat membantu memetakan *system requirements*. Pada *use case* terdapat *system requirements* dan aktor. *System requirements* adalah fungsi-fungsi yang diharapkan ada oleh *user* di dalam suatu sistem yang

akan dibangun. Sedangkan aktor adalah peran (*role*) yang nantinya akan berinteraksi dengan sistem yang akan dibuat. Pada gambar 2.1 terdapat contoh *use case diagram* dari sebuah sistem perpustakaan.



Gambar 2.1. Contoh *use case* perpustakaan

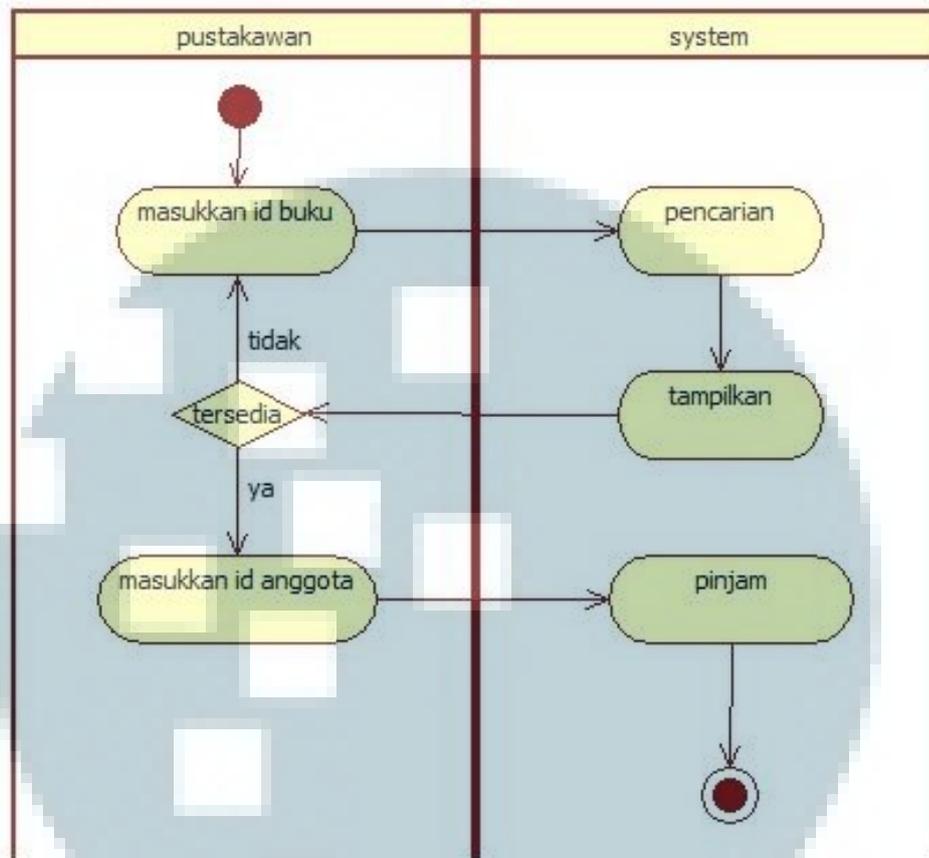
Pada gambar 2.1 dapat dilihat peran dan *requirements* apa saja yang terdapat di dalam sistem perpustakaan itu. Kemudian juga ada pemetaan terhadap peran-peran yang ada dan aksesnya terhadap masing-masing *requirement* yang ada.

## 2. Activity Diagram

Setelah dibuat *use case*, maka dapat dibuat *activity diagram*. Diagram ini dapat memvisualisasikan proses-proses dari setiap *requirement* untuk mencapai tujuan yang diharapkan oleh *user*. Setiap *requirement* memiliki *initial node* yang menandakan dimulainya sebuah aktifitas dan *activity final node* yang menandakan bahwa sebuah aktifitas sudah selesai dikerjakan.

Dalam prakteknya sehari-hari, setiap aktifitas dapat memiliki *decision* yang menyebabkan user memiliki beberapa opsi yang dapat dilakukan tergantung kondisinya pada saat itu. *Activity diagram* ini juga dapat menggambarkan proses-proses yang dapat berlangsung berulang kali. Proses yang digambarkan juga tidak terbatas kepada 1 *actor* saja, melainkan dapat menampung beberapa *actor* sekaligus. Dibawah ini adalah contoh *activity diagram* dari sistem perusahaan pada aktifitas peminjaman buku.

U  
M  
M  
N



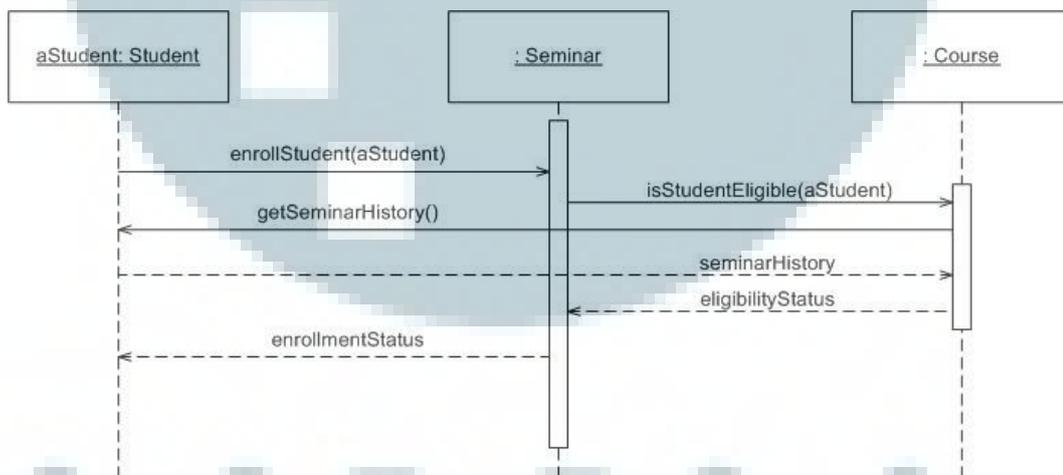
Gambar 2.2. Contoh *activity diagram* peminjaman buku perpustakaan

Pada gambar 2.2 dapat terlihat proses yang terjadi ketika *user* hendak meminjam buku kepada petugas perpustakaan. Proses yang berlangsung terjadi pada 2 *actor* yang berbeda. Pada gambar juga dapat terlihat ada *decision* yang diambil ketika buku tidak ada dan ketika *user* belum terdaftar sebagai *member*.

### 3. Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan urutan pekerjaan, pesan (*value*) yang disampaikan, dan bagaimana setiap elemen yang ada berinteraksi dengan elemen lainnya untuk menyelesaikan sebuah tugas. Setiap urutan digambarkan di dalam diagram horizontal dan pesan yang disampaikan dapat maju dan mundur sesuai dengan proses yang ada.

Pada gambar 2.3 merupakan contoh dari *sequence diagram* murid dengan mata kuliah yang diambil.



Gambar 2.3. Contoh *sequence diagram* murid dengan mata kuliah

Pada gambar 2.3 dapat dilihat bagaimana murid (*student*) berinteraksi dengan seminar dan *course*. Setiap murid dapat mendaftarkan diri pada seminar yang ada (*enrollStudent*). Kemudian setelah murid berhasil mendaftar, akan ada pengecekan yang mengecek apakah murid tersebut berhak untuk mengambil seminar tersebut (*isStudentEligible*). Jika

pengecekan tersebut berhasil, maka *course* akan memasukkan seminar tersebut ke dalam catatan yang berisi *course* apa saja yang pernah diambil oleh murid tersebut.

Pemodelan UML sering disamakan dengan diagram (*flowchart*) sebuah sistem. Diagram adalah gambar yang mewakili sebagian model dari suatu sistem. Gabungan dari seluruh diagram kadang belum tentu mencakup seluruh model yang ada dalam sebuah sistem.

### **2.2.3 MySQL Database**

Dikutip dari situs MySQL ([dev.mysql.com](http://dev.mysql.com)), MySQL adalah *Database Management System* (DBMS) yang bersifat *open source* yang banyak digunakan untuk penyimpanan data, baik data pribadi maupun data korporat. MySQL dikembangkan, didistribusikan, dan didukung oleh Oracle Corporation. MySQL banyak digunakan untuk menyimpan data-data berbagai jenis *web application* yang menggunakan bahasa pemrograman PHP dan Perl.

MySQL tidak memiliki *Graphical User Interface* (GUI) dalam penggunaannya untuk mengatur database. Untuk mengakses MySQL dapat menggunakan terminal (Linux) ataupun command line (Windows). Seiring dengan berkembangnya zaman, banyak bermunculan *third-party application* yang menciptakan GUI untuk mengakses database MySQL seperti phpmyadmin yang terdapat pada XAMPP server.

#### 2.2.4 PHP (*PHP Hypertext Preprocessor*)

Diambil dari situs [www.php.net](http://www.php.net), PHP adalah bahasa pemrograman yang berjalan pada sisi *server*. PHP banyak digunakan sebagai bahasa pemrograman dalam pembuatan *website* yang interaktif dan dinamis. Karena diproses pada sisi *server*, maka pengguna (*client*) hanya akan menerima halaman *website* (data) yang sudah diproses oleh PHP, sehingga klien tidak dapat melihat proses apa yang dijalankan dalam menjalankan *request* dari klien. Berbeda dengan JavaScript yang proses pengolahan datanya berjalan pada sisi klien, sehingga klien bisa melihat logika program yang berjalan pada *web browser*.

Dalam penggunaannya, *script* PHP menggunakan tag pembuka `<?php` dan tag penutup `?>` dimana setiap baris di dalam tag ini tidak akan ditampilkan kepada *client* kecuali jika memang dimaksudkan untuk dibaca oleh *client*. Selain digunakan untuk membuat *website*, PHP juga dapat digunakan untuk pembuatan *desktop application* dan *Command Line Scripting*.

Bahasa pemrograman PHP dapat digunakan pada berbagai sistem operasi, seperti Linux, Unix, Windows, Mac OS X, dan lain lain. Selain itu PHP juga kompatibel dengan teknologi yang digunakan pada berbagai Web Server masa kini seperti Apache, IIS, dan teknologi *web server* lainnya. Dengan demikian, pembuatan *website* dengan menggunakan bahasa pemrograman PHP mampu memberikan kemudahan kepada penggunanya, baik *end user* maupun *developer* karena dapat berjalan di berbagai sistem

operasi dan *web server*. Selain itu, PHP juga mendukung 2 jenis pemrograman seperti *procedural programming* dan *object oriented programming* (OOP).

### 2.2.5 AJAX (Asynchronous JavaScript and XML)

Dikutip dari halaman [www.w3schools.com](http://www.w3schools.com), Ajax adalah teknik pembuatan halaman website yang menggunakan kombinasi dari beberapa teknologi yaitu HTML, CSS, Javascript, jQuery, hingga PHP. Website yang menggunakan Ajax memungkinkan penggunanya melakukan transaksi data (*data transaction*) secara asinkronus sehingga memungkinkan user bertukar data secara *real time* tanpa harus *me-reload* ulang halaman. Ajax mendukung semua jenis transaksi data yaitu *Create, Read, Update, Delete*.

Teknologi Ajax menggabungkan beberapa teknologi yang sudah banyak digunakan dalam pembuatan *website*. Komponen Ajax terdiri dari HTML dan CSS untuk persentasi kepada klien, *Document Object Model* (DOM) untuk desain yang dinamis, *JavaScript Object Notation* (JSON) sebagai format pertukaran data, dan JavaScript yang menggabungkan semua komponen diatas. Penggunaan Ajax tidak hanya terbatas pada tampilan HTML saja tetapi juga mendukung transaksi data dengan *database* secara *real time*.