



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. Teori Pengumpulan Data

2.1.1. *Web Crawling*

Web Crawling disebut juga robot internet yang menelusuri dan mencari informasi melalui *World Wide Web* (WWW) yang bertujuan untuk melakukan indeks halaman *website* yang diperlukan. *Web crawling* biasanya dilakukan untuk mencari informasi tertentu yang diperlukan untuk maksud tertentu (Nidhi Singh, 2015). Peneliti menerapkan metode *web crawling* ini untuk mendapatkan informasi dari lima sumber toko *online* yang menjual berbagai macam komponen untuk perakitan komputer dengan cara mendapatkan *tag* HTML dari kelima *website* tersebut. Setelah nantinya *tag* HTML dari kelima *website* didapatkan, akan ditentukan data yang ingin diambil dari *website* tersebut. Setelah mendapatkan *tag* dari data yang ingin didapatkan dari suatu *website* selanjutnya akan dilanjutkan dengan melakukan metode *web scraping*.

2.1.2. *Web Scraping*

Selain *web crawling*, metode lain yang dipakai dalam pengambilan data dari halaman *website* adalah *web scraping*. *Web scraping* adalah kumpulan teknik yang dipakai untuk mendapatkan informasi dari *website* secara otomatis sehingga tidak perlu melakukan penarikan data secara satu persatu (Eloisa Vargiu, 2013).

Pendapat lain menyatakan bahwa *Web scraping* adalah sebuah proses yang memanfaatkan dokumen berbentuk *semi – structured* yang didapatkan dari internet, yang dimana dokumen tersebut berbentuk sebuah halaman *website* yang dibangun

oleh bahasa *markup* seperti HTML ataupun XHTML yang kemudian dianalisis untuk mendapatkan informasi yang berguna yang dapat dilakukan untuk konteks lain (Turland, 2010).

Implementasi *web scraping* dapat diterapkan untuk mendapatkan informasi tertentu dari sebuah *website* yang nantinya akan diolah untuk dapat ditampilkan ke dalam bentuk lain yang menghasilkan informasi yang berguna untuk para penggunanya. Pada perancangan aplikasi yang dilakukan peneliti ini *web scraping* yang dilakukan adalah dengan mengambil data dari kelima sumber *website* yang menjual komponen komputer. Setelah berhasil memperoleh data yang valid dari kelima *website* tersebut, aplikasi yang dibuat akan melakukan perbandingan harga dari tiap *website* dan menghasilkan rekomendasi harga yang paling rendah sebagai *output* informasi yang dihasilkan oleh aplikasi.

2.2. Teori Pengolahan Data

2.2.1. Data Mining

Data mining merupakan sebuah proses dalam menemukan pola dan tren yang berguna dalam kumpulan data dalam jumlah yang besar (Daniel T. Larose, 2014). *Data mining* yang dilakukan oleh peneliti dalam melakukan perancangan aplikasi rekomendasi harga untuk pembelian komponen komputer ini adalah dengan menggali informasi dari sejumlah sumber data yang nantinya akan menghasilkan komparasi harga komponen komputer yang dicari dari lima toko sumber yang sudah ditentukan. Dengan melakukan komparasi harga dari tiap komponen ini nantinya akan menghasilkan informasi harga komponen dari harga tertinggi sampai harga yang paling rendah.

Untuk mendapatkan hasil informasi sesuai dengan yang dibutuhkan, sebelumnya kita perlu melakukan *preprocessing data*. Di dalam *preprocessing data* terdapat 2 aktifitas inti yang dilakukan yaitu *data cleaning* dan *data transformation*. Terkadang dalam aktifitas pengambilan data terdapat data yang nilainya tidak konsisten, terdapat redundansi data, ataupun data yang informasinya tidak relevan.

Maka dari itulah kita perlu melakukan *preprocessing data* sebelum melakukan *data mining*.

2.2.2. Data Cleaning dan Data Transformation

Data Cleaning merupakan proses pembersihan data yang dilakukan setelah kita melakukan pengambilan data (Daniel T. Larose, 2014). Terkadang dalam proses pengambilan data terdapat data yang memberikan informasi yang tidak relevan, nilai dari data yang tidak konsisten, dan redundansi data. Maka dari itu untuk dapat mendapatkan data dengan kualitas yang valid dan akurat perlu dilakukan *data cleaning* sebelum melakukan penarikan data.

Setelah data yang diambil telah selesai melalui tahap *data cleaning*, untuk memberikan informasi data yang valid diperlukan adanya penentuan format untuk data tertentu agar nilai yang dihasilkan konsisten. Proses penentuan format untuk setiap data inilah yang disebut dengan tahap *Data Transformation* (Daniel T. Larose, 2014). *Output* yang dihasilkan dari tahapan *data transformation* inilah yang dapat dijadikan sumber data oleh aplikasi yang dirancang oleh peneliti.

2.3. Teori Web Based Application

2.3.1. Pengertian Web

World Wide Web (WWW) atau disebut juga *web* adalah sebuah sistem berbasis *hypermedia* yang menyediakan sebuah informasi yang didapatkan dengan cara melakukan *browsing* dengan internet dengan cara *non-sequential* dengan menggunakan sebuah *hyperlink* (Connolly dan Begg, 2014). Perancangan aplikasi yang dilakukan oleh peneliti juga memafaatkan teknologi *web* sehingga untuk dapat mengakses aplikasi ini dilakukan melalui *web browser* seperti google chrome, mozilla firefox, internet explorer, dan *web browser* lainnya.

2.3.2. Pengertian Internet

Internet atau terkadang sering disingkat menjadi *net* adalah jaringan global yang menghubungkan jutaan komputer di seluruh dunia sehingga dengan adanya jaringan internet setiap *user* yang mempunyai izin untuk menggunakan satu komputer dapat mengakses dan mengumpulkan informasi dari komputer lain (Gary B. Shelly, 2010).

Penggunaan internet dalam perancangan aplikasi yang dilakukan peneliti ini mempunyai peranan yang vital karena aplikasi yang dirancang adalah aplikasi berbasis *web* yang berarti untuk dapat menggunakan aplikasi yang dibuat, para pengguna dari aplikasi ini harus terhubung dengan koneksi internet terlebih dahulu.

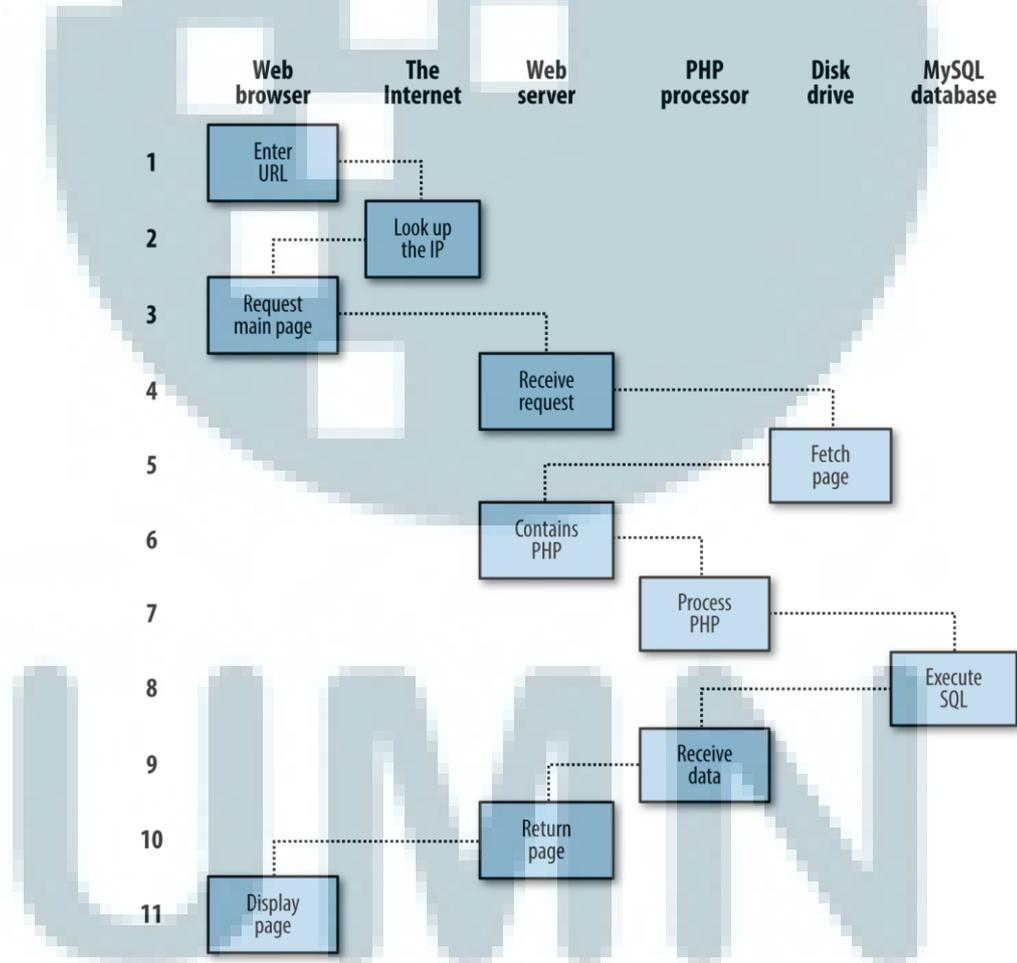
2.3.3. *Web Based Application*

Web Based Application atau yang apabila diartikan ke dalam bahasa Indonesia mempunyai arti aplikasi berbasis internet ini merupakan aplikasi yang mempunyai sebuah keunggulan gabungan dari aplikasi *desktop* dan aplikasi *server* yang memiliki kemudahan akses dan fleksibilitas yang dapat diakses melalui sebuah *web* (Athina Lazakidou, 2010). Aplikasi berbasis internet ini dikembangkan dengan arsitektur *client server* yang artinya aplikasi ini menempatkan *web browser* di sisi *client* dimana untuk mengakses data yang diinginkan para pengguna aplikasi berbasis internet ini harus memiliki *web browser* untuk mengaksesnya. Setelah membuka aplikasi melalui *web browser*, nantinya data yang ingin diakses oleh *user* akan diproses ke *web server*, dan setelah *web server* menampilkan data yang diminta oleh *user* barulah *user* dapat mengakses konten yang diminta tersebut.

Aplikasi berbasis internet ini dapat dikembangkan dengan menerapkan berbagai bahasa skrip seperti HTML, JavaScript, JQuery, AJAX, XML, DOM, dan lain – lain yang dapat diterjemahkan oleh *web browser* untuk mendapatkan informasi yang diinginkan. Tren aplikasi berbasis internet di Indonesia ini semakin berkembang karena ditunjang juga oleh penyebaran koneksi internet di Indonesia

sehingga cukup mudah untuk mendapatkan akses internet. Aplikasi berbasis internet ini memiliki beberapa keunggulan, keunggulan pertama yaitu aplikasi ini dapat diakses oleh berbagai macam *platform* yang digunakan selama terdapat *web browser* dan koneksi internet untuk mengaksesnya. Selain itu, aplikasi berbasis internet juga menawarkan kemudahan dengan dapat diakses kapanpun, dimanapun selama ada koneksi internet tanpa harus melakukan instalasi terlebih dahulu apabila kita ingin mengaksesnya.

2.3.4. Website Dinamis



Gambar 2.1 Proses *request* dan *response* pada *website* dinamis

Sumber: (Robin Nixon, 2014)

Website dinamis merupakan *website* yang secara struktur diperuntukan untuk di-*update* sesering mungkin (Rudyanto Arief, 2011). Pada saat ini sudah banyak *website* dinamis yang dikembangkan untuk berbagai macam kebutuhan seperti *web portal*, *toko online*, dan berbagai macam media sosial. Salah satu hal yang membedakan *website* dinamis dengan *website* statis adalah penggunaan *database* yang berfungsi sebagai penyimpan data yang diterima dari interaksi *client-server*.

2.3.5. Database

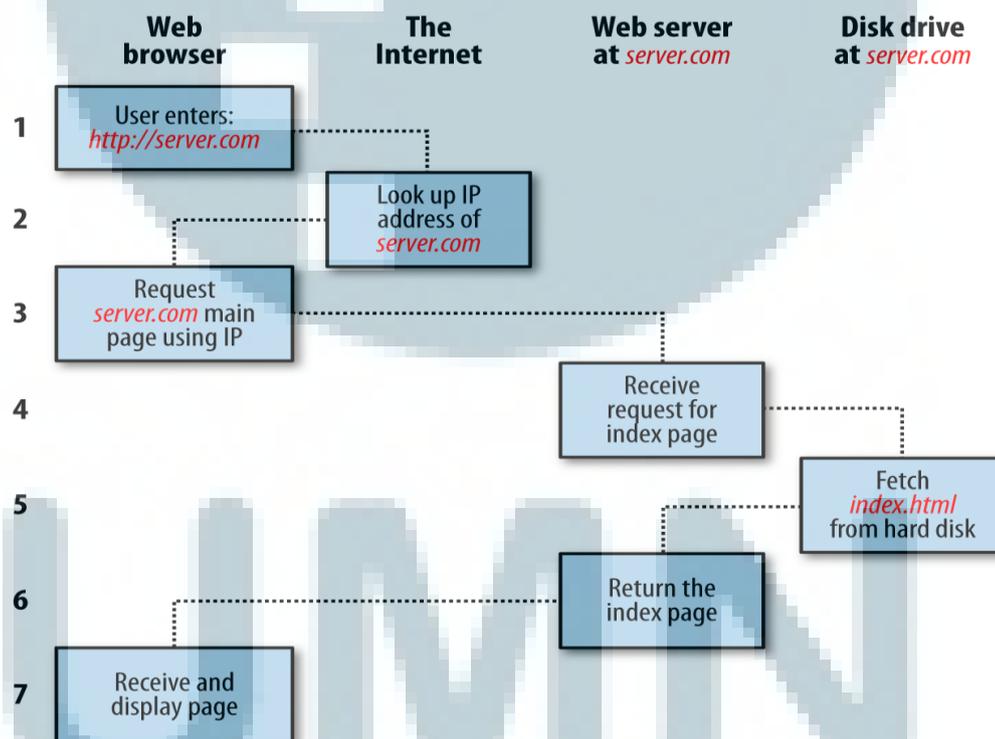
Salah satu keunggulan dari *website* dinamis seperti yang sudah disebutkan sebelumnya adalah penerapan *database*. *Database* digunakan untuk penyimpanan data yang menunjang untuk fungsional suatu *website*. Sementara itu pengertian lain menyatakan *database* merupakan kumpulan dari data-data yang mempunyai relasi logis dan penjelasan tentang data yang saling terhubung tersebut dan perancangan data sampai sedemikian rupa sehingga dapat dianalisa untuk menghasilkan informasi yang berguna (Connolly, 2010). Definisi lain dari *database* menurut adalah kumpulan terstruktur yang terdiri dari *record* ataupun data yang disimpan di dalam sistem komputer dan diorganisir dengan cara tertentu agar informasi yang disimpan dapat diakses dengan cepat (Robin Nixon, 2014).

Pada perancangan aplikasi yang dilakukan, peneliti memilih MySQL sebagai *database* yang digunakan. Salah satu alasan peneliti memilih MySQL untuk *database* yang dipakai dalam perancangan aplikasi adalah MySQL merupakan *relational database* yang bersifat *open source* terbesar di bidangnya (Sahil Bahl, 2014). Selain itu, MySQL juga memiliki keunggulan dalam hal kompatibilitas yang berarti MySQL dapat dijalankan pada berbagai *platform* sistem operasi.

2.3.6. Arsitektur *Client – Server*

Komponen dari arsitektur *client-server* terbagi menjadi 2 yaitu sisi *client* dan sisi *server* (Satzinger, 2010). *Server* yang berarti sebuah proses, modul, objek atau komputer yang menyediakan layanan melalui jaringan dan sedangkan *client* adalah sebuah proses, modul, objek atau komputer yang meminta layanan dari satu atau lebih *server*. Pada aplikasi yang dikembangkan oleh peneliti ini juga menerapkan arsitektur *client-server* dalam pengembangannya dimana pihak klien melakukan *request* terhadap komponen yang ingin diakses dan sisi *server* kemudian memproses permintaan tersebut.

2.3.7. PHP Hypertext Preprocessor



Gambar 2.2 Proses *request* dan *response* dasar pada arsitektur *client – server*

Sumber: (Robin Nixon, 2014)

PHP Hypertext Preprocessor merupakan bahasa skrip yang didesain secara spesifik untuk membangun sebuah *web* yang dimana skrip PHP ini dituliskan di dalam sebuah halaman HTML untuk mengeksekusi setiap permintaan informasi yang diakses didalam halaman HTML (Luke Welling dan Laura Thomson, 2010).

PHP Hypertext Preprocessor merupakan salah satu bahasa skrip yang dipakai dalam membangun suatu *web* dinamis. *Web* dinamis merupakan sebuah halaman *website* dimana di dalamnya terdapat interaksi antara sisi *client* dengan *server*. Interaksi yang dilakukan misalnya sisi *client* dapat menulis data, meng-*edit* informasi dari suatu halaman *website*, maupun menghapus konten yang ada di dalam *website*.

PHP memiliki beberapa keunggulan, salah satu keunggulannya yaitu PHP merupakan bahasa skrip yang dapat dipelajari semua orang tanpa harus mengeluarkan biaya karena PHP bersifat *open source*. Selain itu, kelebihan PHP yang lainnya adalah PHP juga sudah banyak didukung dengan banyak *web server* seperti apache, IIS, lighttpd, dan xitami. PHP juga memiliki banyak *mailing list* dan komunitas-komunitas yang aktif apabila kita memerlukan bantuan dalam melakukan pengembangan aplikasi berbasis internet dengan menggunakan PHP.

2.4. Tools yang Digunakan

2.4.1. Adobe Dreamweaver

Dreamweaver adalah sebuah *editor* HTML profesional untuk mendesain *web* secara visual dan mengelola situs atau halaman *web* (Sigit, 2010). Adobe Dreamweaver merupakan salah satu *software* yang dikembangkan oleh Adobe yang mempunyai fungsi untuk melakukan pengembangan atau pembuatan suatu *website* yang memanfaatkan bahasa skrip seperti HTML, PHP, Javascript, dan bahasa skrip yang lainnya. *Software* ini dipakai oleh peneliti dalam tahap *coding* pada saat perancangan aplikasi yang dilakukan dimana aplikasi yang dibuat adalah aplikasi berbasis *web*.

2.4.2. Pentaho Data Integration

Pentaho Data Integration atau yang dikenal juga dengan nama Kettle merupakan *software* yang bersifat *open source* yang menyediakan fungsionalitas untuk memberikan solusi integrasi data. Dengan Pentaho Data Integration kita dapat mengambil data dari berbagai sumber, menyesuaikan, dan mengubah datanya sesuai dengan kebutuhan kita dalam pengembangan sistem (Alex Meadows, 2013).

Pentaho Data Integration merupakan *software* yang dimanfaatkan peneliti dalam melakukan perancangan aplikasi untuk melakukan pengumpulan dan penarikan data. Selain bersifat *open source*, pentaho data integration memiliki keunggulan lain yaitu *software* ini dapat digunakan di berbagai *platform* yang berbeda. Keunggulan lain yang dimiliki oleh perangkat lunak ini yaitu kemudahan mentransformasikan data ke dalam bentuk yang diinginkan seperti bentuk *spreadsheet*, *SAP input*, *dbf input*, *web text file*.

2.4.3. Yed Graphic Editor

Yed Graph Editor merupakan sebuah aplikasi *desktop* yang dapat dipakai secara cepat dan juga efektif dalam melakukan pembuatan diagram – diagram (<http://www.yworks.com/en/products/yfiles/yed/>). Perangkat lunak ini bersifat *open source* dan juga dapat dijalankan di berbagai *platform* sistem operasi yang dijalankan oleh komputer. dipakai oleh peneliti pada tahap *design* untuk membuat diagram-diagram *UML* untuk melakukan sebagian pemodelan aplikasi yang akan dibuat yaitu *activity diagram* dan *use case diagram*.

2.4.4. Microsoft Visio

Microsoft visio merupakan salah satu produk dari perusahaan microsoft yang merupakan aplikasi penting yang digunakan untuk membuat diagram yang

bersifat teknis maupun yang bersifat bisnis (Scott A. Helmers, 2013). Microsoft Visio merupakan *software* yang digunakan oleh peneliti dalam membuat diagram – diagram yang dibuat untuk melakukan pemodelan terhadap aplikasi yang dikembangkan oleh peneliti. Diagram – diagram yang dibuat oleh peneliti menggunakan *software* ini adalah *sequence diagram* dan *flowchart diagram*. Versi Microsoft Visio yang digunakan oleh peneliti dalam melakukan pembuatan *sequence diagram* dan *flowchart diagram* adalah Microsoft Visio 2013.

2.4.5. MySql

MySql adalah sebuah perangkat lunak sistem mengenai basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MySQL, Postagre SQL, dan lain lain (Anhar, 2010). MySql merupakan *database* yang bersifat *open source* sehingga dapat dengan bebas digunakan oleh semua orang. Meskipun bersifat *open source*, MySql tetap menjadi *database* yang handal dan sudah banyak diimplementasikan di berbagai pengembangan aplikasi. Selain itu, keunggulan dari MySql adalah MySql dapat dijalankan di berbagai *platform* sistem operasi sehingga tidak akan menjadi hambatan apabila ingin digunakan di sistem operasi apapun.



Gambar 2.3 Logo mysql

Sumber: www.mysql.com

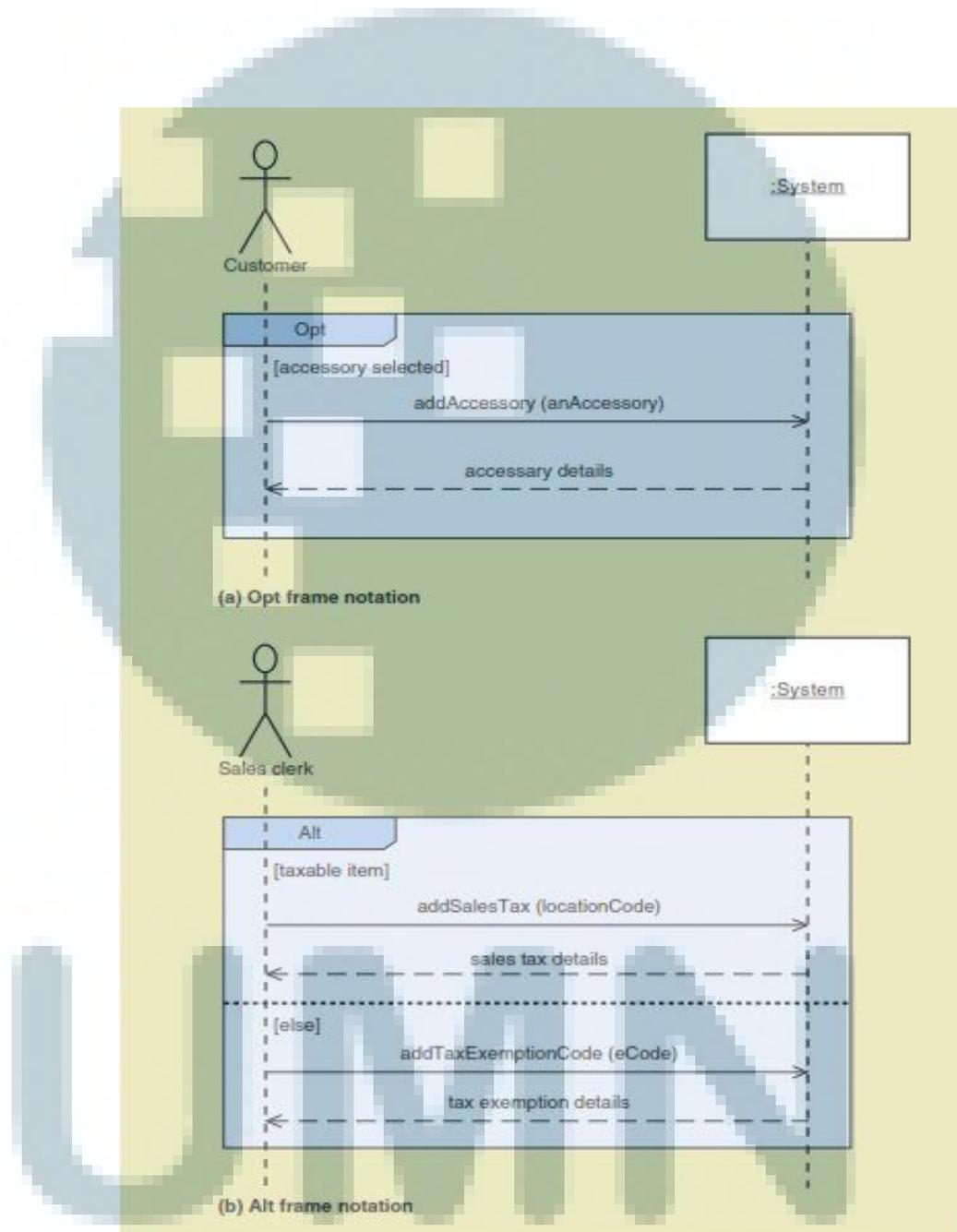
2.5. *Unified Modeling Language Diagram*

Unified modeling language atau yang biasa disingkat dengan UML merupakan bahasa pemodelan untuk sistem ataupun perangkat lunak yang berorientasikan objek dimana tujuan dari dilakukannya pemodelan dari sistem atau perangkat lunak ini adalah untuk menyederhanakan permasalahan – permasalahan yang bersifat kompleks sehingga dapat lebih mudah untuk dipahami dan dipelajari (Nugroho, 2010). Pendapat lain tentang pengertian UML di dalam sebuah buku yang ditulisnya dengan judul “object-oriented analysis and design through unified modeling language” UML itu diumpamakan sebagai Bahasa Inggris dimana di dalamnya terdapat kosa kata dan aturan untuk mengkombinasikan kata – kata sehingga menjadi kalimat yang bermakna (Gandharba Swain, 2010). Begitu pula dengan UML yang diumpamakan sebagai bahasa dimana juga memiliki kosa kata dan aturan dalam mengkombinasikan kata – kata di dalamnya yang berfokus untuk merepresentasikan sebuah sistem baik secara fisik maupun secara konseptual. Di dalam perancangan aplikasi berbasis *web* yang dilakukan oleh peneliti, ada tiga diagram UML yang digunakan yaitu *activity diagram*, *sequence diagram*, dan *use case diagram*.

2.5.1. *Sequence Diagram*

Sequence diagram merupakan interaksi diagram yang menjelaskan waktu pada saat sistem melakukan suatu aktifitas (Gandharba Swain, 2010). Sementara itu pendapat lain tentang *sequence diagram* pada pengembangan sistem yang bersifat *object – oriented*, aliran informasi dilakukan melalui pengiriman pesan dari sebuah aktor kepada suatu objek, disitulah fungsi dari *sequence diagram* (Satzinger *et al*, 2010). Diagram ini dapat juga dijadikan sebagai pendeskripsian lebih lanjut apabila penggambaran aktifitas sistem *melalui use case* dianggap cukup rumit. Diagram ini berfokus untuk menggambarkan aktifitas – aktifitas yang dilakukan oleh para *actor* yang terlibat terhadap aplikasi yang dikembangkan oleh peneliti.

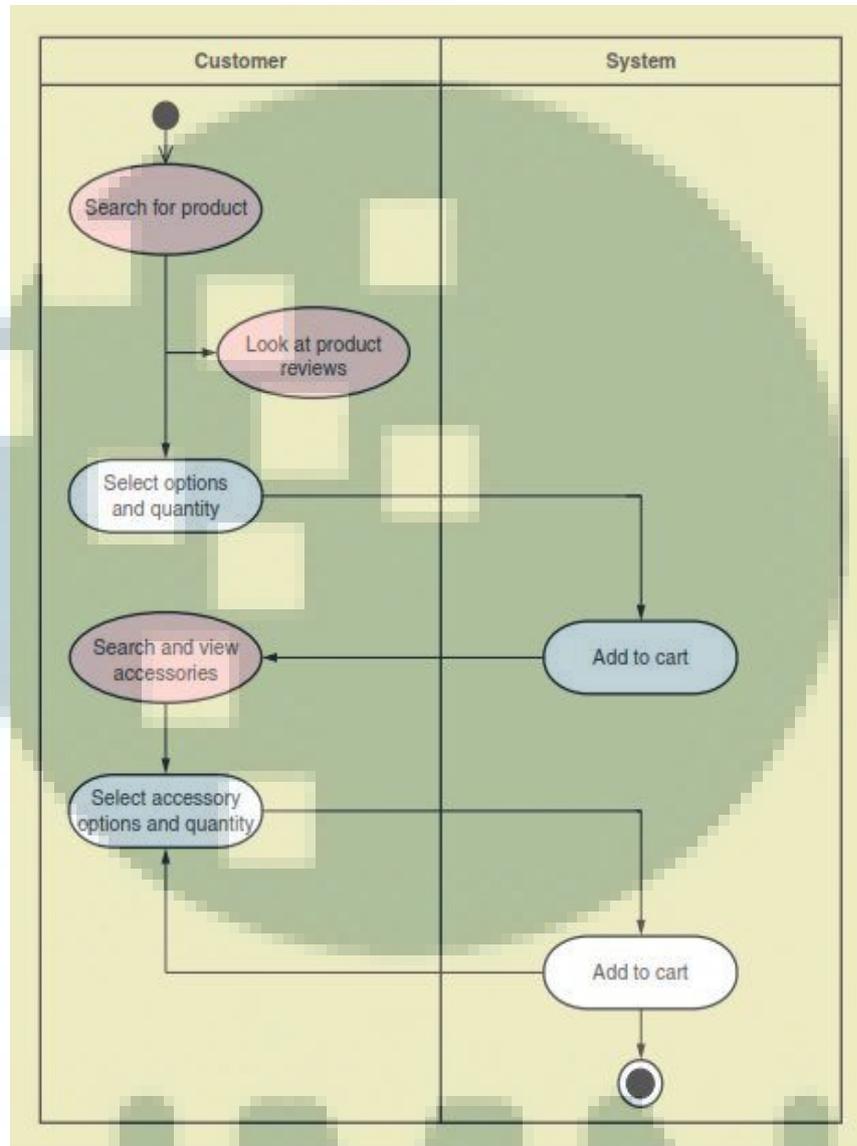
Dimulai dari menjelaskan dimana sebuah aktifitas yang dilakukan sistem dimulai, lalu melihat perubahan apa saja yang terjadi akibat aktifitas yang dilakukan sistem dan *output* apa yang dihasilkan dari aktifitas yang dijalankan oleh sistem.



Gambar 2.4 Contoh *sequence diagram*

Sumber: Satzinger *et al*, 2010

2.5.2. Activity Diagram



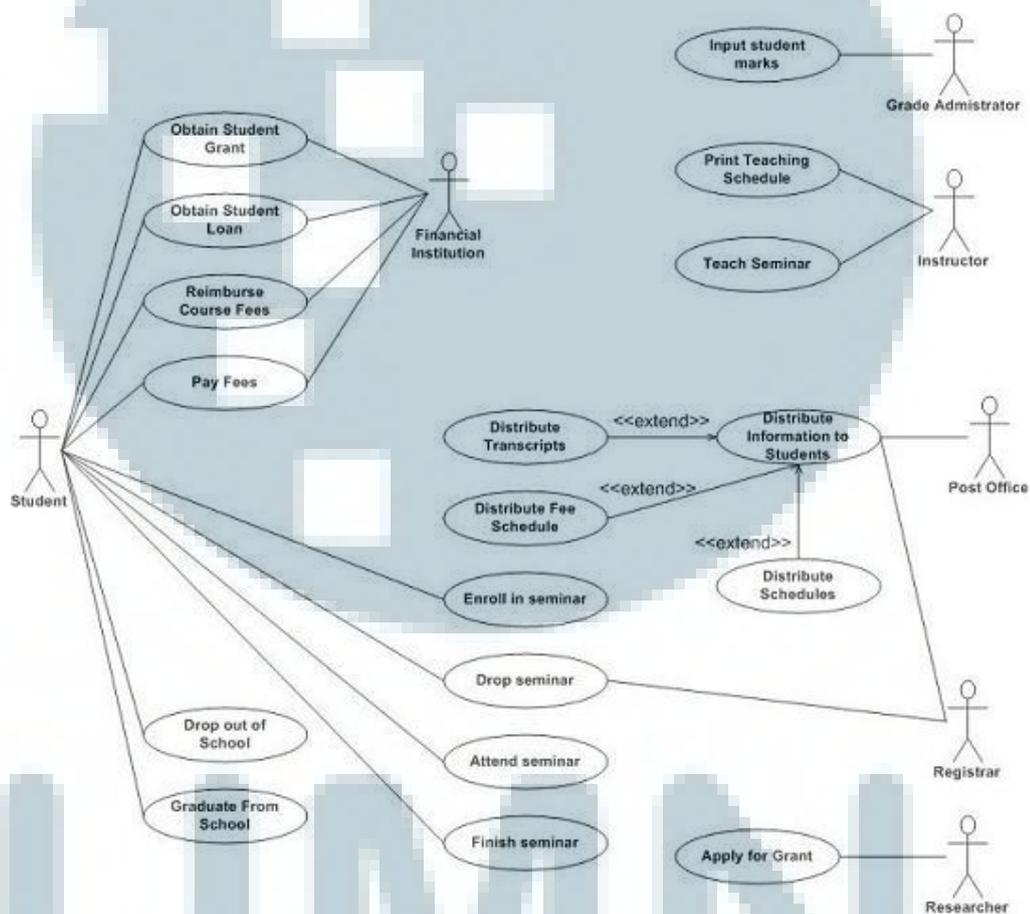
Gambar 2.5 Contoh *activity diagram*

Sumber: Satzinger *et al*, 2010

Activity diagram merupakan diagram yang menggambarkan aliran dari setiap aktifitas yang dijalankan oleh sistem (Gandharba Swain, 2010). *Activity Diagram* sangat efektif dipakai ketika *use case diagram* yang dibuat bersifat cukup kompleks sehingga dengan adanya *activity diagram* akan lebih mempermudah dalam menjelaskan aktifitas yang dilakukan sistem (Satzinger *et al*, 2010). *Activity*

diagram merupakan aktifitas pemodelan dalam bentuk diagram yang berfokus untuk menjelaskan secara *detail* urutan dari aktifitas yang berjalan di dalam aplikasi, dimana pada saat membuat *activity diagram* ini fokus kita adalah pada *use case* atau proses yang berjalan.

2.5.3. Use Case Diagram



Gambar 2.6 Contoh *use case diagram*

Sumber: <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>

Use case Diagram merupakan pemodelan dari *unified modeling language* yang digunakan untuk menggambarkan hubungan antara sebuah *use case* dengan *user* yang bersangkutan (Satzinger, 2010). Penggunaan *use case*

diagram sebagai bentuk pemodelan dalam melakukan penggambaran hubungan antara sistem dengan *user* biasanya diterapkan pada pengembangan sistem yang bersifat *object oriented programming*.

Use case diagram terdiri dari dua komponen dasar yaitu *actor* dan *use case*. *Actor* merupakan komponen yang merepresentasikan orang / divisi yang berhubungan dengan sistem. Sementara *use case* merupakan representasi dari sistem yang berjalan. Penggabungan kedua komponen ini dalam bentuk gambaran fungsionalitas akan menghasilkan penggambaran hubungan antara sistem dengan *user*.

2.6. ***Flowchart Diagram***

Flowchart diagram atau yang apabila diartikan ke dalam Bahasa Indonesia yaitu diagram arus merupakan teknik analitis yang digunakan untuk menjelaskan beberapa aspek di dalam sistem informasi secara jelas, tepat, dan logis (Romney dan Steinbart, 2010). *Flowchart Diagram* digunakan oleh peneliti di dalam perancangan sistem untuk menggambarkan secara jelas dan detail dari proses berjalannya aplikasi.

Diagram *flowchart* terdiri dari berbagai simbol yang memiliki arti tersendiri dalam mendeskripsikan sebuah objek sehingga menghasilkan representasi atau gambaran dari kerja yang dilakukan sistem. simbol di dalam *flowchart diagram* dapat dibagi menjadi 4 kategori (Romney dan Steinbart, 2010), yaitu:

- a. Simbol *input / output*: simbol yang merepresentasikan perangkat atau media yang berperan sebagai *input* ataupun sebagai *output* perekam dari proses yang berjalan.
- b. Simbol pemrosesan: simbol yang menunjukkan apakah proses pengolahan data dilakukan secara manual ataukah dengan menggunakan bantuan suatu perangkat tertentu.

- c. Simbol penyimpanan: simbol yang merepresentasikan penggunaan perangkat untuk melakukan penyimpanan data.
- d. Simbol arus dan lain – lain: simbol yang digunakan untuk menggambarkan aliran data, menggambarkan dimana proses dimulai dan dimana proses berakhir, dimana keputusan dibuat, dan pada waktu kapan dibutuhkan penjelasan tambahan pada *flowchart* yang ada.

2.7. **Entity Relationship Diagram**

Entity relationship diagram merupakan diagram yang digunakan untuk menggambarkan *database* dari suatu organisasi yang nantinya akan dijadikan dasar sebagai pembuatan tabel-tabel yang ada di dalam *database* (Hall, 2011). *Diagram entity relationship diagram* digunakan juga untuk melihat hubungan dari setiap entitas yang ada di tabel – tabel di dalam *database*. *Entity relationship diagram* memiliki 3 komponen utama (Brady dan Loonam, 2010), yaitu:

- a. Entitas

Entitas adalah sebuah objek yang menarik di bidang organisasi yang dimodelkan. Lambang dari komponen entitas dalam *entity relationship diagram* digambarkan dalam bentuk persegi panjang.

- b. *Relationship*

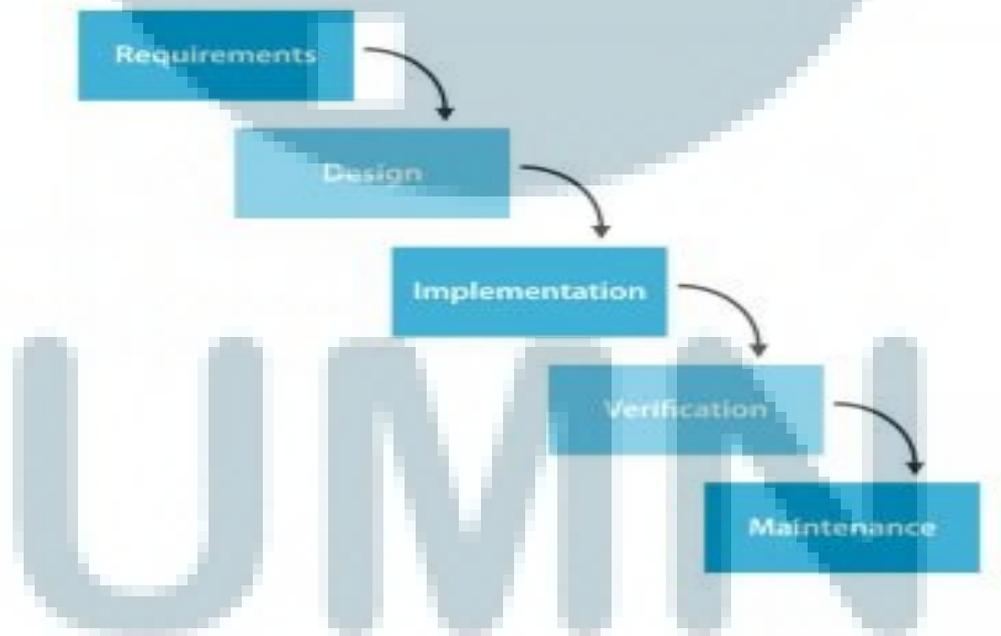
Relationship adalah komponen dari *entity relationship diagram* yang menggambarkan hubungan atau relasi yang ada antara dua jenis entitas.

- c. *Attribute*

Atribut menurut Connolly dan Begg (2010) adalah properti dari sebuah entitas atau satu jenis hubungan. Atribut memiliki nilai – nilai yang mendeskripsikan setiap *event* yang ada di dalam satu entitas dan juga merepresentasikan bagian utama dari data yang disimpan di *database*.

2.8. *Waterfall Model*

Metode yang akan digunakan untuk merancang dan mengembangkan aplikasi ini adalah metode *waterfall model*. Model *waterfall* adalah suatu model klasik yang bersifat sistematis dan berurutan dalam proses pengembangan *software* (Pressman, 2010). Sesuai dengan namanya yaitu *waterfall*, dalam pengembangan *software* metodologi menerapkan prinsip sebelum satu tahap yang sedang dikerjakan terselesaikan, maka kita tidak akan bisa masuk ke tahap selanjutnya. Pemaparan kebutuhan dari pengembangan *software* awal secara jelas pada tahap awal dari metode *waterfall model* ini menjadi salah satu alasan mengapa penulis memilih metode ini sebagai metode perancangan sistem. Dengan pemaparan semua kebutuhan untuk pengembangan aplikasi secara jelas pada tahap awal, diharapkan ketika melanjutkan ke tahap-tahap selanjutnya sudah dapat menjadi lebih fokus.



Gambar 2.7 *Waterfall model*

Sumber: www.waterfall-model.com