

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam pembuatan Sistem Informasi Geografis tempat pendidikan Al-Qur'an di Kabupaten Sukoharjo ada beberapa penelitian yang terkait sebagai dasar acuan dalam penelitian ini. Penelitian-penelitian tersebut sebagai berikut.

Penelitian yang dilakukan oleh Hamdi, dkk (2018) dengan judul “Perancangan Sistem Informasi Geografis Pemetaan Taman Di Kabupaten Indragiri Hilir Berbasis *Web*”. Perancangan Sistem Informasi Geografis pemetaan taman di Kabupaten Indragiri Hilir berbasis *web* ini bertujuan untuk menjadi media promosi tempat rekreasi yang telah di rancang dan dibangun oleh pemerintah serta dapat mempermudah masyarakat dalam mencari taman dan ruang terbuka hijau dengan pencarian letak koordinat yang tepat dan akurat serta informasi yang lengkap. Bahasa pemrograman yang digunakan adalah PHP, dan media pemrograman pendukung lainnya menggunakan Googel Maps API dan MySQL. Metode yang digunakan pada penelitian ini menggunakan metode *waterfall*.

Penelitian Mustaqim dan Ariyanto (2016) dengan judul “Sistem Informasi Geografis Berbasis *Web*” Untuk Pemetaan Komoditas Pertanian Di Kabupaten XYZ. Tujuan dari penelitian ini adalah membangun sebuah aplikasi Sistem Informasi Geografis berbasis *web* yang mampu menyajikan data spasial dalam bentuk peta wilayah dan data non spasial secara akurat mengenai komoditas pertanian yang ada, untuk membantu pihak-pihak yang membutuhkan informasi seperti dinas pertanian atau masyarakat umum. Sistem Informasi Geografis ini menggunakan perpaduan bahasa pemrograman antara PHP, HTML, *Javascript*, *Mapfile*. Metodologi dasar yang digunakan dalam penelitian ini adalah metodologi pengembangan sistem (SDLC) yaitu *waterfall*.

Penelitian dari Anggraeni dan Budisusanto (2016) dengan judul “Pembangunan Sistem Informasi Berbasis *Web* untuk Pemetaan Industri Kreatif Berbasis Budaya di Kota Surakarta”. Pada penelitian ini dilakukan pembangunan

Sistem Informasi Geografis (SIG) berbasis *Web* atau yang lebih dikenal sebagai WebGIS untuk menampilkan lokasi industri kreatif budaya yang ada di kota Surakarta. WebGIS Industri Kreatif Berbasis Budaya di Kota Surakarta ini memiliki fitur-fitur yang akan memberikan informasi kepada masyarakat serta pelaku bisnis tentang industri kreatif berbasis budaya di Kota Surakarta, seperti fitur pencarian dan menampilkan berdasarkan jenis serta tahun berdiri industri. Bahasa pemrograman yang dipakai dalam pembuatan WebGIS ini menggunakan PHP dan MySQL serta Google Maps API sebagai media pendukung pemrograman.

Penelitian dari Nugraha (2014) dengan judul “Pengembangan Sistem Informasi Geografis Daerah Tujuan Wisata Kota Surakarta Berbasis *Web*”. Penelitian ini dilakukan dengan tujuan untuk menghasilkan SIG dengan visualisasi data spasial yang berisi informasi letak obyek wisata dan fasilitas penunjang wisata yang disajikan secara jelas kepada wisatawan lokal maupun asing. Sistem ini dibuat menggunakan perpaduan bahasa pemrograman *framework Codeigniter*, HTML, MySQL dan Google Maps API.

2.2 Kerangka Pemikiran

Kerangka pemikiran yang dijalankan dalam penelitian ini adalah sebagai berikut :

1) Latar Belakang Masalah

Di Sukoharjo terdapat 45 tempat pendidikan Al-Qur'an, penyebaran informasi mengenai lokasi tempat pendidikan Al-Qur'an di Sukoharjo masih menggunakan pamflet yang ditempel di mading masjid dan di tiang atau tembok pinggir jalan serta dari mulut ke mulut sehingga masyarakat cukup kesulitan untuk mendapatkan informasi tentang tempat pendidikan Al-Qur'an.

2) Rumusan Masalah

Bagaimana membuat Sistem Informasi Geografis tempat pendidikan Al-Qur'an di Kabupaten Sukoharjo.

3) Penguasaan Dasar PHP, dan MySQL

Mempelajari dasar-dasar PHP, dan MySQL agar lebih menguasai perintah atau program yang nantinya akan digunakan dalam pembuatan Sistem Informasi Geografis.

4) Pengumpulan Data Tertulis dan Tidak Tertulis

Pengumpulan data ini dilakukan dengan cara observasi atau dengan wawancara langsung serta mencari referensi dari buku, jurnal, dan sumber lainnya.

5) Observasi Aplikasi PHP, dan MySQL

Pengamatan sampel-sampel yang telah ada, jurnal, buku, dan sumber lainnya yang dapat digunakan sebagai referensi pembuatan Sistem Informasi Geografis.

6) Analisis dan Perancangan Sistem Informasi Geografis Berorientasi Objek

Penguasaan analisis dan perancangan sistem akan membantu dalam melakukan analisis dan pembuatan Sistem Informasi Geografis tempat pendidikan Al-Qur'an di Kabupaten Sukoharjo. Analisis sistem meliputi : *Use case Diagram, Activity Diagram, Sequence Diagram, Class Diagram*. Pembuatan sistem ini menggunakan PHP dan MySQL.

7) Implementasi Sistem Informasi Geografis Tempat Pendidikan Al-Qur'an di Kabupaten Sukoharjo.

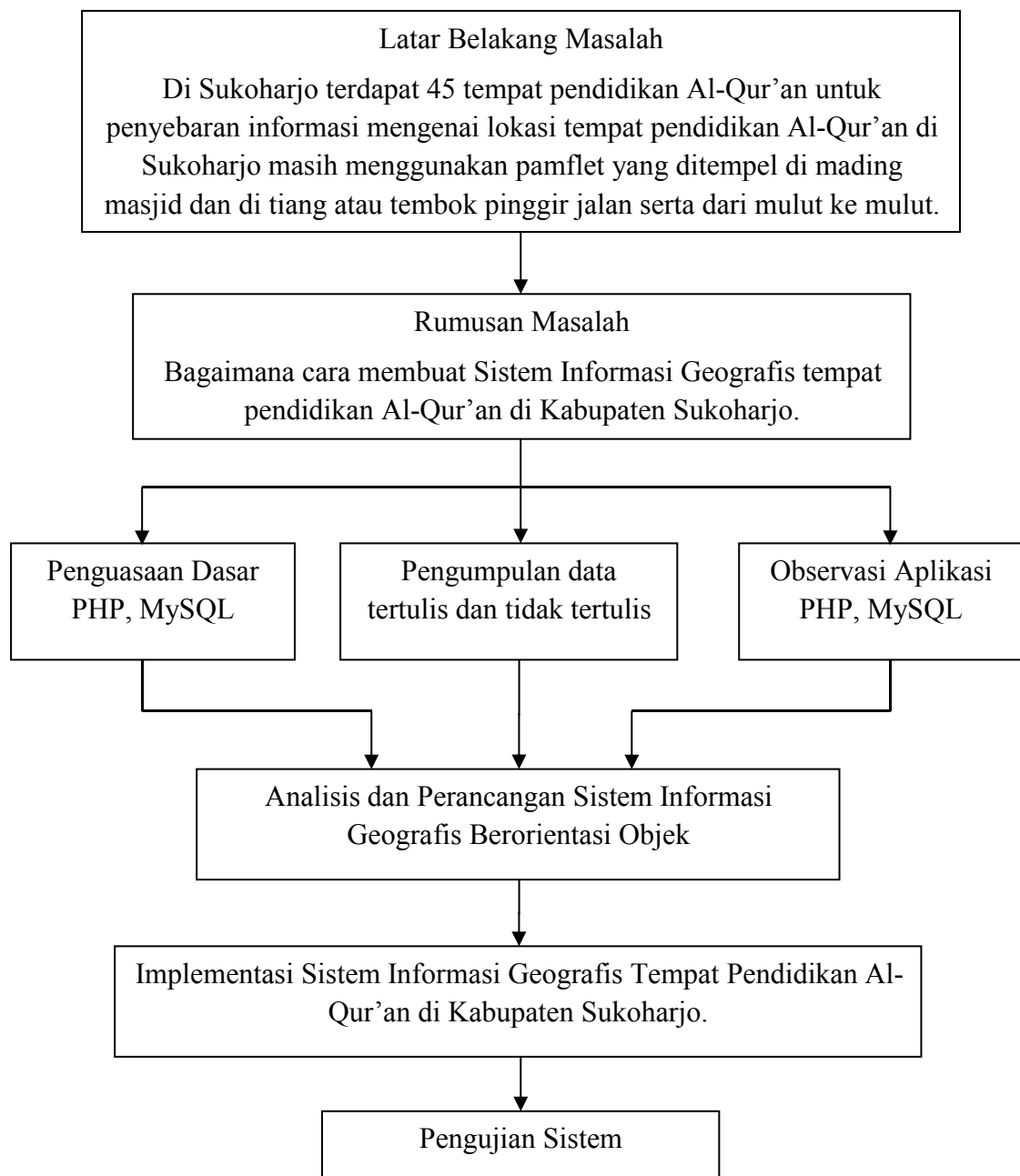
Implementasi sistem adalah langkah-langkah atau prosedur-prosedur yang dilakukan dalam menyelesaikan desain sistem yang telah disetujui, untuk menginstal, menguji dan memulai sistem baru atau sistem yang diperbaiki. Lingkungan implementasi *website* ini meliputi kebutuhan perangkat lunak, perangkat keras, *form* program yang sesuai, *query* yang digunakan, pemrograman, pengujian program dan pengujian *website* yang telah dirancang sesuai dengan kebutuhan dalam pembuatan Sistem Informasi Geografis ini.

8) Pengujian Sistem

Pengujian sistem adalah prosedur yang dilakukan untuk menyelesaikan rancangan sistem yang telah disetujui, menguji sistem, menginstal serta

memulai penggunaan sistem baru atau sistem yang telah diperbaiki. Dalam pengujian sistem ini menggunakan metode *black-box testing* dan pengujian kuesioner.

Berdasarkan penjabaran dari kerangka pemikiran penelitian tentang Sistem Informasi Geografis tempat pendidikan Al-Qur'an di Kabupaten Sukoharjo, maka dapat digambarkan diagram alur seperti yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran Penelitian

2.3 Landasan Teori

Penjelasan dari beberapa istilah yang berhubungan dengan pembuatan Sistem Informasi Geografis tempat pendidikan Al-Qur'an di Kabupaten Sukoharjo.

2.3.1 Pengertian Sistem

Menurut Kadir (2014), sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran jika dalam sebuah sistem terdapat sebuah elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

2.3.2 Pengertian Informasi

Menurut Sutabri (2014), Informasi adalah hasil pemrosesan, manipulasi, dan pengorganisasian/penataan dari sekelompok data yang mempunyai nilai pengetahuan bagi penggunanya. Informasi juga disebut data yang diproses atau data yang memiliki arti. Informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakannya.

2.3.3 Pengertian Geografis

Menurut Prahasta (2009) bahwa geografis mengandung pengertian suatu persoalan atau hal mengenai wilayah di permukaan bumi, baik permukaan dua dimensi atau tiga dimensi. Pengertian geografis dapat disimpulkan sebagai suatu ilmu yang mempelajari tentang lokasi di mana suatu objek terletak di permukaan bumi beserta keterangan objek tersebut.

2.3.4 Sistem Informasi Geografis (SIG)

Menurut Prahasta (2009) bahwa SIG merupakan sejenis perangkat lunak, perangkat keras (manusia, prosedur, basis data dan fasilitas jaringan komunikasi) yang dapat digunakan untuk memfasilitasi proses pemasukan, penyimpanan,

manipulasi, menampilkan dan keluaran data atau informasi geografis berikut atribut-atribut terkait.

Menurut Ekadinata (2008) bahwa SIG adalah sebuah sistem atau teknologi berbasis komputer yang dibangun dengan tujuan untuk mengumpulkan, menyimpan, mengolah dan menganalisa, serta menyajikan data dan informasi dari suatu objek atau fenomena yang berkaitan dengan letak atau keberadaannya di permukaan bumi.

2.3.5 Subsistem SIG

Menurut Prahasta (2009) SIG dapat diuraikan menjadi beberapa sub-sistem berikut :

- 1) Data *Input* : sub-sistem ini bertugas untuk mengumpulkan, mempersiapkan dan menyimpan data spasial dan atribut dari berbagai sumber. Sub-sistem ini pula yang bertanggung jawab dalam mengkonversikan atau mentransformasikan format-format data aslinya ke dalam format (*native*) yang dapat digunakan oleh perangkat SIG yang bersangkutan.
- 2) Data *Output* : sub-sistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk *softcopy* maupun bentuk *hardcopy* seperti halnya tabel, grafik, *report*, peta dan lain sebagainya.
- 3) Data *Management* : sub-sistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil kembali atau di-*retrive*, di-*update* dan di-*edit*.
- 4) Data *Manipulation & Analysis* : sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG dan melakukan manipulasi serta pemodelan data untuk menghasilkan informasi yang diharapkan.

2.3.6 Komponen SIG

Menurut Prahasta (2009) SIG terdiri dari beberapa komponen berikut dengan berbagai karakteristiknya :

- 1) Perangkat Keras. *digitizer, scanner, monitor, Central Processing Unit (CPU), mouse, printer, and plotter.*

- 2) Perangkat Lunak. Dari sudut pandang lain, SIG bisa juga merupakan suatu sistem perangkat lunak yang tersusun secara modular di mana sistem basis datanya memegang peranan kunci.
- 3) Manajemen. Teknologi SIG tidaklah bermanfaat tanpa manusia yang mengelola sistem dan membangun perencanaan yang dapat diaplikasikan sesuai kondisi nyata.
- 4) Data dan informasi geografis.

2.3.7 Jenis Data Masukan Dalam SIG

Menurut Ekadinata (2008) Data geografis pada dasarnya tersusun oleh dua komponen penting yaitu data spasial dan data atribut :

1) Data Spasial

Data spasial merepresentasikan posisi atau lokasi geografis dari suatu objek di permukaan bumi. Data spasial berasal dari peta analog, foto udara, citra satelit, survei lapangan dan pengukuran dengan *global positioning systems* (GPS). Format data spasial secara umum dapat dikategorikan dalam format digital dan analog. Dalam format digital terdapat dua model representasi data, yaitu model data *vektor* dan model data *raster*. Kedua model mampu menyimpan detail informasi tentang lokasi serta atributnya.

Pada model *vektor*, posisi suatu objek didefinisikan oleh rangkaian koordinat x dan y. Data *vektor* terdiri dari titik, garis (*arc/line*) dan poligon. Titik bisa digunakan sebagai lokasi sebuah kota atau posisi *tower* radio. Garis bisa digunakan untuk menunjukkan jalur kabel atau menggambarkan batasan daerah. Poligon bisa digunakan untuk menggambarkan sebuah danau atau sebuah Negara pada peta dunia. Contoh penggunaan data *vektor* misalkan jaringan kabel optik, jaringan transmisi *tower*, pola air sungai, dan garis kontur.

Sedangkan pada model *raster*, data spasial direpresentasikan dengan *pixel-pixel* sebagai unit terkecil. Foto digital seperti foto satelit merupakan bagian dari data *raster* pada peta. Data *raster* terdiri dari kolom dan baris, dimana tiap *cell* menyimpan nilai warna. Data *raster* disimpan dalam berbagai format seperti TIF, JPEG, BMP dan sebagainya.

2) Data Non Spasial / Atribut

Data atribut memberikan deskripsi atau penjelasan dari suatu objek. Biasanya data atribut diperoleh dari statistik, sensus, pengukuran, foto, narasi, lapangan dan data tabular. Data atribut bisa dilihat berdasarkan kualitas dan kuantitasnya. Contoh data atribut misalkan tanah, geologi, geomorfologi, penggunaan lahan, populasi, dan transportasi.

2.3.8 Kemampuan SIG

Menurut Prahasta (2009) Kemampuan SIG saat ini mencakup kemampuan untuk menampilkan, mencetak dan memanipulasi berbagai lapisan data termasuk gambar foto udara, informasi keselamatan demografi dan publik, kepemilikan properti, pajak, penggunaan lahan, dan informasi zona, lokasi utilitas, jalan, fitur alam, topografi dan fitur buatan manusia serta lingkungan lainnya.

Secara eksplisit, kemampuan SIG juga dapat dilihat dari pengertian atau definisinya. Berikut adalah kemampuan-kemampuan SIG yang diambil dari beberapa definisi-definisi SIG yang telah dituliskan pada bagian sebelumnya :

- 1) Memasukkan dan mengumpulkan data atribut dan spasial.
- 2) Mengintegrasikan data atribut dan spasial.
- 3) Memeriksa dan meng-*update* (meng-*edit*) data atribut dan spasial.
- 4) Menyimpan dan memanggil kembali data atribut.
- 5) Mempresentasikan atau menampilkan data spasial dan atribut.
- 6) Mengelola data geografis atribut dan spasial.
- 7) Memanipulasi data geografis atribut.
- 8) Menganalisa data geografis (spasial dan atribut).
- 9) Menghasilkan keluaran (*output*) data geografis dalam bentuk-bentuk peta tematik (*view* dan *layout*), tabel, grafik (*chart*), laporan (*report*) dan lainnya baik dalam bentuk *hardcopy* maupun *softcopy*.

2.3.9 Google Maps API

Menurut Julzarika dan Andi (2013), Google Maps adalah layanan gratis Google yang cukup populer. Kita dapat menambahkan fitur Google Maps dalam *web* kita sendiri dengan Google Maps API. Google Maps API merupakan *library JavaScript*. Untuk melakukan pemrograman Google Maps API dapat terbilang mudah. Yang kita butuhkan adalah pengetahuan tentang HTML dan *JavaScript*, serta koneksi *Internet*. Dengan menggunakan Google Maps API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang diperlukan. Data peta dunia menjadi urusan Google.

Google Maps API adalah fungsi-fungsi pemrograman yang disediakan oleh Google Maps agar Google maps bisa diintegrasikan kedalam *Web* atau aplikasi yang sedang dibuat. Google maps API sendiri menyediakan fungsi yang banyak, berikut ini adalah pembagiannya :

- 1) Google Maps *Javascript* API v3, ini adalah google maps API yang tersedia dalam bahasa *javascript*.
- 2) Google Maps API *Webservice*, layanan Google maps API untuk fungsi lanjutan seperti *direction*, *Geocoding*, *Distance Matrix API* dan *elevation API*.
- 3) Google Place API, ini adalah API wajib yang harus dikuasai untuk membuat aplikasi *Location based service*.

Semua aplikasi yang membutuhkan layanan peta dapat dibuat. Aplikasi tidak terbatas pada *website*, namun juga bisa aplikasi *desktop* ataupun aplikasi di *smartphone* bahkan aplikasi di *embeded* sistem misalkan aplikasi navigasi di mobil. Untuk mendapatkan Google API Key, bisa didapatkan melalui situs resmi Google.

2.3.10 Website

Menurut Gamaliel (2014), *Website* adalah merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke halaman lain

(*hypertext*), baik diantara halaman yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* atau halaman diakses dan dibaca melalui *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi *browser* lainnya.

Perkembangan teknologi *website* saat ini sudah memasuki generasi ketiga yaitu *Web 3.0* atau disebut *semantic web*. Adapun generasi *website* sebelumnya dan masih berjalan yaitu *Web 1.0* dan *Web 2.0* yang saat ini dikenal dengan sebutan *web science*. *Web 1.0* adalah generasi pertama dari *website* internet. Pada generasi ini pengguna hanya dapat mencari (*searching*) dan melihat (*browsing*) data informasi yang ada pada *website*. Era pengembangan *web* kedua (*Web 2.0*) dimana pengunjung mulai dapat melakukan interaksi dengan diatur oleh sistem yang ada pada *web*. *Web 2.0* memiliki beberapa ciri mencolok yaitu *share*, *collaborate* dan *exploit*. Di era *Web 2.0* sekarang, penggunaan *web* untuk berbagi, pertemanan, kolaborasi menjadi sesuatu yang penting. *Web 3.0*. Teknologi *web* generasi ketiga ini merupakan perkembangan lebih maju dari *Web 2.0* dimana disini *web* seolah-olah sudah seperti kehidupan di alam nyata. *Web 3.0* memiliki ciri-ciri umum seperti *suggest*, *happen* dan *provide*.

2.3.11 HTML (*Hypertext Markup Language*)

Menurut Sidik (2015), *Hyper Text Markup Language* (HTML) merupakan sebuah dokumen yang disajikan dalam sebuah *browser web server* dan pada umumnya berisi informasi ataupun sebuah *interface* dari sebuah aplikasi di dalam internet. Penggunaan HTML lebih mudah dimengerti dibanding bahasa pemrograman lainnya dan kerana bentuknya lebih sederhana maka HTML dapat dibaca oleh berbagai *platform* seperti *Windows*, *Linux*, *Macintosh*. Contoh *script* HTML disajikan pada Gambar 2.2.

```

<html>
  <head>
    <title>Bismillah</title>
  </head>
  <body bgcolor = "black">
    <p style="color:white">Assalamua'laikum</p>
  </body>
</html>

```

Gambar 2.2 Contoh *script* HTML

2.3.12 PHP (*Perl Hypertext Preprocessor*)

Menurut Lukmanul (2014), PHP (*Hypertext Preprocessing*) merupakan bahasa pemrograman yang biasa digunakan untuk membuat halaman HTML. File *php* yang dibuat akan diproses di dalam *server*, sedangkan halaman yang akan dikirim ke *browser* pengunjung hanyalah tampilan HTML-nya. Dengan PHP, halaman *website* yang dibuat akan menjadi dinamis, yakni dapat selalu berubah tanpa harus mengubah isi *website* secara manual. Informasi akan diproses ulang oleh *web server* sehingga akan didapatkan isi paling mutakhir dari halaman *web*. Contoh *script* PHP disajikan pada Gambar 2.3.

```

<html>
  <head>
    <title>Contoh list dengan PHP</title>
  </head>
  <body>
    <h2>Daftar Absensi Mahasiswa</h2>
    <ol>
      <?php
        for ($i= 1; $i <= 1000; $i++)
        {
          echo "<li>Nama Mahasiswa ke- $\$i$ </li>";
        }
      ?>
    </ol>
  </body>
</html>

```

Gambar 2.3 Contoh *script* PHP

2.3.13 *Bootstrap*

Menurut Utomo (2016), *Bootstrap* merupakan salah satu *framework* HTML, CSS, dan JS yang cukup populer, serta banyak digunakan oleh para pengembang *web* saat ini. *Framework* ini banyak digunakan untuk membuat

website bersifat *responsive*. Artinya bisa menyesuaikan tampilan *layout*-nya berdasarkan pada fakta di lapangan yang menunjukkan bahwa perkembangan penggunaan *handheld* seperti *smartphone* dan *tablet* semakin meningkat. Dengan *website responsive* tersebut dapat memudahkan pengguna melakukan *resizing*, *scrolling* dan *panning* dengan lebih optimal.

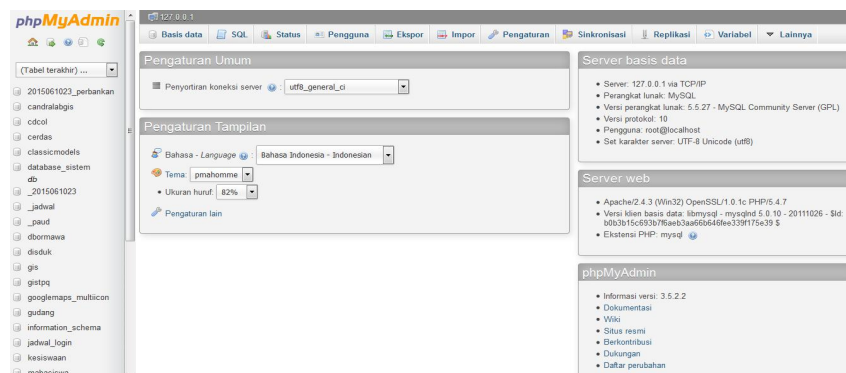
2.3.14 Database

Menurut Rosa dan Shalahuddin (2015), Basis data (*Database*) merupakan salah satu bagian dalam rekayasa perangkat lunak yang terkomputerisasi dan bertujuan utama memelihara data yang sudah diolah atau media penyimpanan informasi agar dapat diakses dengan mudah dan cepat.

2.3.15 MySQL

Menurut Dianing (2012), MySQL adalah sistem manajemen *database* SQL yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan SQL *database* manajemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan.

Menurut Winarto dan Zaki (2014), MySQL adalah sebuah *software database*. MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan. Tampilan MySQL dapat dilihat pada Gambar 2.4.

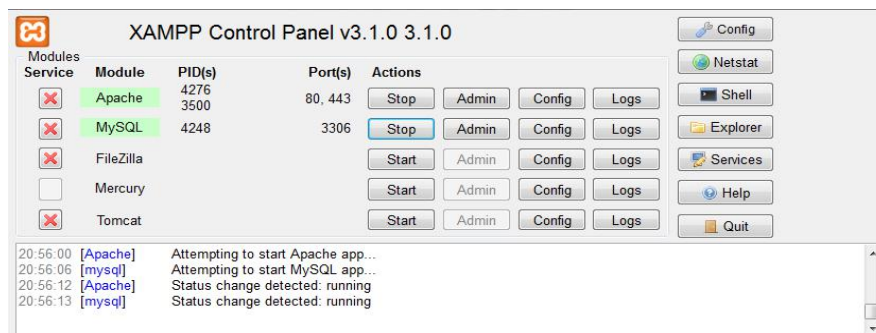


Gambar 2.4 Tampilan MySQL

2.3.16 Xampp

Menurut Supono (2016) bahwa Xampp merupakan sebuah perangkat lunak gratis sehingga bebas digunakan. Xampp berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari *Apache*, *HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan *PHP* dan *Perl*. Xampp dikembangkan oleh perusahaan *Apache Friends* yang biasanya digunakan untuk simulasi pengembangan.

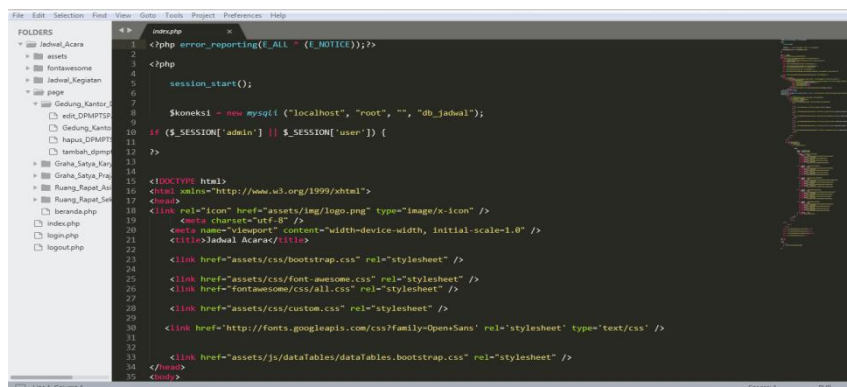
Menurut Riyanto (2011), Xampp merupakan paket *PHP* dan *MySQL* berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis *PHP*. Xampp mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket. Tampilan Xampp dapat dilihat pada Gambar 2.5.



Gambar 2.5 Tampilan Xampp

2.3.17 Sublime Text

Menurut Faridl (2015), *Sublime text* adalah teks *editor* berbasis *Python*, sebuah teks *editor* yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan *developer* (pengembang), penulis dan *desainer*. Para *programmer* biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang ia kerjakan. Sampai saat ini *sublime text* sudah mencapai versi 3 yang ditunjukkan pada Gambar 2.6. *Sublime text* mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*.



Gambar 2.6 Tampilan *Sublime Text*

2.3.18 UML (*Unified Modeling Language*)

Menurut Rosa dan Shalahudin (2015) *Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.





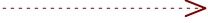

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek.

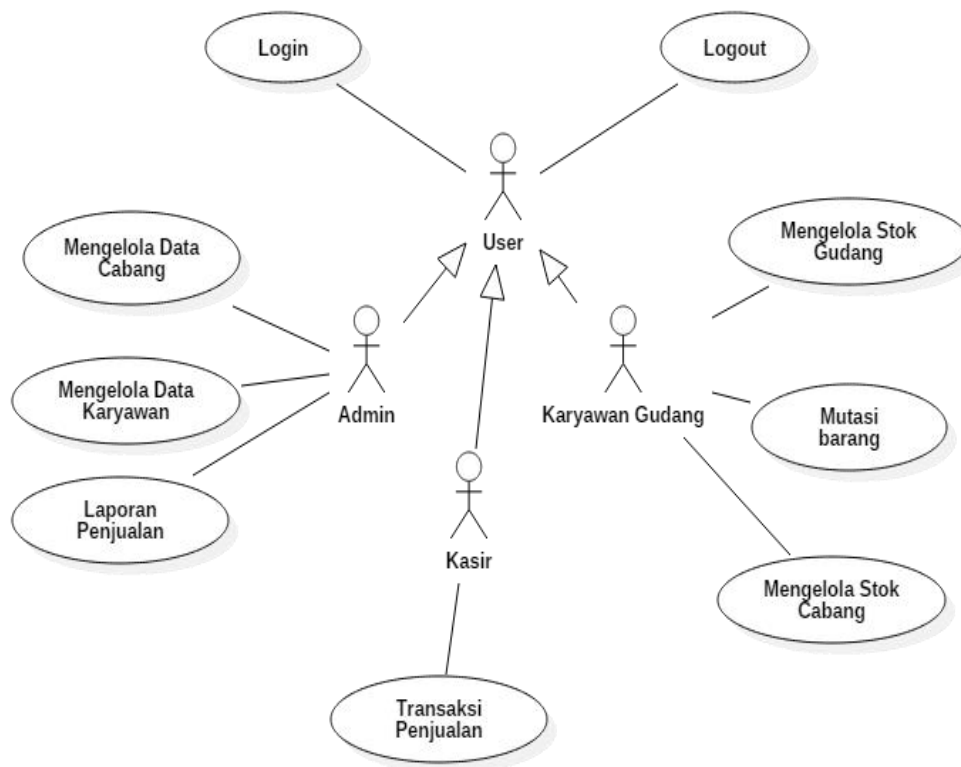
2.3.18.1 *Use case Diagram*

Menurut Rosa dan Shalahudin (2015), *Use case Diagram* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Simbol *Use case Diagram*

No	Gambar	Keterangan
1.	<p data-bbox="395 353 512 383"><i>Use case</i></p> 	<p data-bbox="775 353 1359 544">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
2.	<p data-bbox="395 566 469 595"><i>Actor</i></p> 	<p data-bbox="775 566 1359 723">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>
3.	<p data-bbox="395 779 544 808"><i>Association</i></p> 	<p data-bbox="775 779 1359 887">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4.	<p data-bbox="395 909 587 938"><i>Generalization</i></p> 	<p data-bbox="775 909 1359 1059">Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>
5.	<p data-bbox="395 1081 491 1111"><i>Include</i></p> <p data-bbox="448 1133 576 1162"><<include>></p> 	<p data-bbox="775 1081 1359 1272">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 atau sebagai syarat dijalankan <i>use case</i> ini.</p>
6.	<p data-bbox="395 1305 483 1335"><i>Extend</i></p> <p data-bbox="448 1357 576 1386"><<extend>></p> 	<p data-bbox="775 1305 1359 1617">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> yang ditambahkan.</p>

Contoh penggunaan *Use case Diagram* ditunjukkan pada Gambar 2.7.



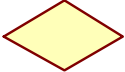


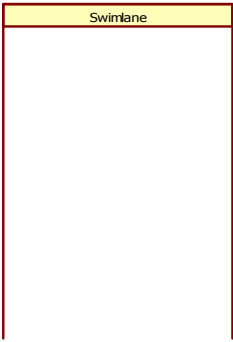


Gambar 2.7 Contoh Use case Diagram (Hendini, 2016)

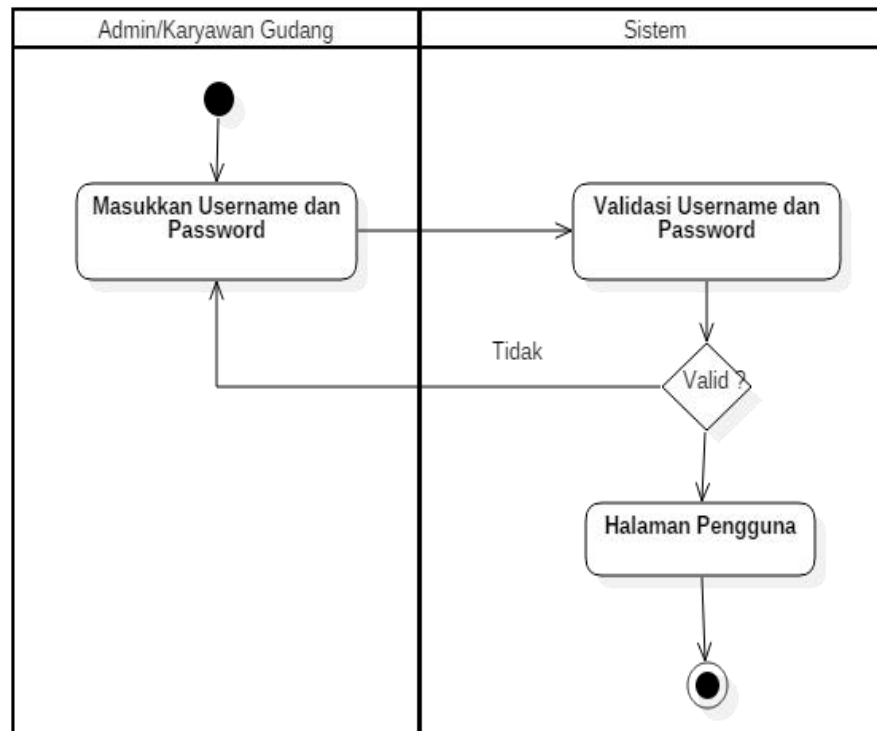
2.3.18.2 Activity Diagram

Menurut Rosa dan Shalahudin (2015), 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 di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas yang ditunjukkan pada Tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram*

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

Contoh penggunaan *Activity Diagram* ditunjukkan pada Gambar 2.8.



Gambar 2.8 Contoh *Activity Diagram* (Hendini, 2016)

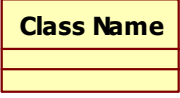


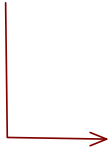

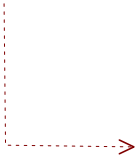

2.3.18.3 Class Diagram

Menurut Rosa dan Shalahudin (2015), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

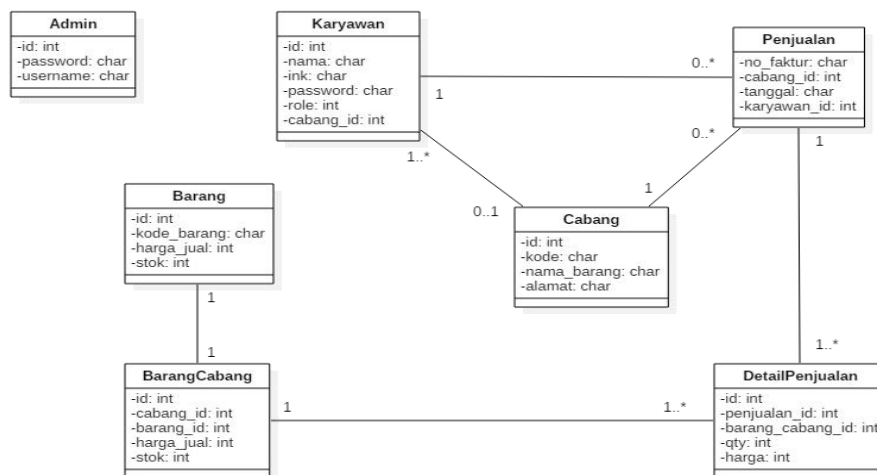
- 1) Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
- 2) Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

No	Gambar	Keterangan
1.	Kelas/ <i>Class</i> 	Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/ <i>Directed Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Contoh penggunaan *Class Diagram* dapat dilihat pada Gambar 2.9.


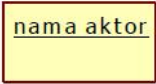

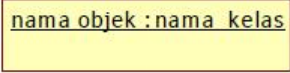

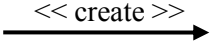
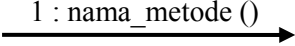
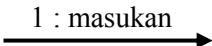
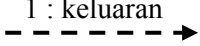
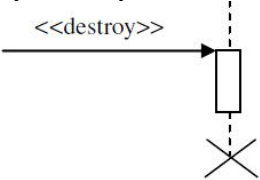


Gambar 2.9 Contoh *Class Diagram* (Hendini, 2016)

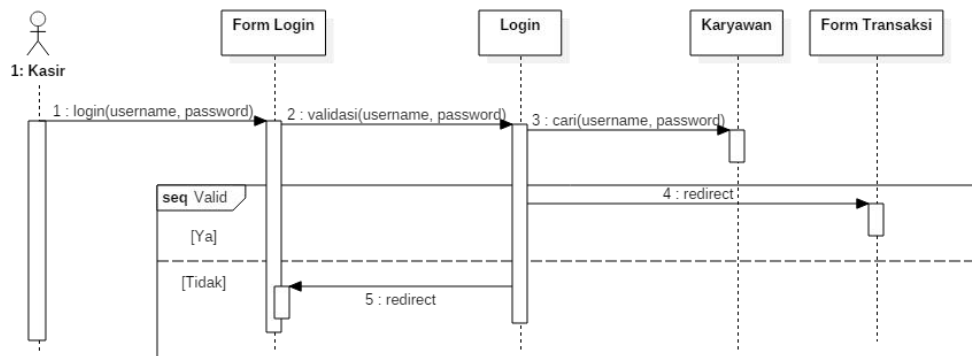
2.3.18.4 *Sequence Diagram*

Menurut Rosa dan M. Shalahudin (2015), *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *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*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Simbol-Simbol *Sequence Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol *Sequence Diagram*

No.	Simbol	Keterangan
1.	Aktor  Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor 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.	<i>Object</i> 	Menyatakan objek yang berinteraksi pesan.
4.	Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data / masukkan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	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> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

Contoh penggunaan *Sequence Diagram* dapat dilihat pada Gambar 2.10.



Gambar 2.10 Contoh *Sequence Diagram* (Hendini, 2016)

2.3.19 *Black-box Testing*

Menurut Shalahuddin dan Rosa (2015), *Black-box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

2.3.20 Pengujian Kuesioner

Menurut Sugiyono (2017), Kuesioner adalah teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner merupakan teknik pengumpulan data yang efisien bila peneliti tahu dengan pasti variabel yang akan diukur dan tahu apa yang bisa diharapkan dari responden. Selain itu, kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar di wilayah yang luas.