BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

2.1.1 Pengertian Rancang

Dalam jurnal Rosadi & Andriawan (2016:53) Setelah tahap analisis sistem dilakukan dan mendapatkan gambaran dengan jelas tentang apa yang harus dikerjakan, maka dilakukan tahan berikutnya yaitu Perancangan Sistem. Definisi Perancangan Sistem menurut Robert J Verzello/ John Reuter III, Data Processing System and Concept (Dalam buku Analisis Sistem Informasi, Jogiyanto, 2005:196), mengemukkan bahwa : "Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem; pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untung rancang bangun implementasi; menggambarkan bagaimana sustu sistem dibentuk".

Perancangan atau desain merupakan yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesalahan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem (Husda, 2012:137).

Menurut Taylor dalam buku Prahasta (2014:488) Perancangan merupakan proses penggunaan berbagai prinsip dan teknik untuk tujuan pendefinisian

perangkat, proses, atau sistem hingga tingkat detil tertentu yang memungkinkan realisasi fisiknya.

2.1.2 Website

Website atau situs dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya itu, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling berkait dimana masing masing dihubungkan dengan jaringan jaringan halaman *(hyperlink)*, (Surajino, S.H.R. 2004 dalam jurnal Utama, 2011:360).

Website merupakan salah satu media yang dapat digunakan untuk memberikan informasi kepada masyarakat umum secara cepat dan mudah melalui internet. Biaya yang murah, kemudahan akses dan efisiensi menjadi alasan semakin luasnya penggunaan *website* (Saputro, Hamzah, & Triyono, 2013:32).

Website adalah lokasi di internet yang menyajikan kumpulan informasi sehubungan dengan profil pemilik situs. *Website* adalah suatu halaman yang memuat situs-situs *web page* yang berada di internet yang berfungsi sebagai media penyampaian informasi, komunikasi, atau transaksi (Hastanti, Purnama, & Wardati, 2015:1).

2.1.3 Sistem

Sistem *(system)* adalah kumpulan dari sub-sub sistem, elemen-elemen, prosedur-prosedur, yang saling berintegrasi untuk mencapai tujuan tertentu, seperti informasi, target atau goal. Karakter suatu sistem terdiri dari: Komponen *(Components)*, Batas Sistem *(Boundary)*, Lingkungan luar sistem *(Environments)*,

Penghubung *(Interface), input, process* dan *output* Sasaran *(Objectives),* Tujuan *(Goal).*(Ali & Wangdra, 2010:7).

Menurut Husda (2012:111) Sistem berasal dari bahasa Latin *(systema)* dan bahasa Yunani *(sustema)* adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energy. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, dimana suatu model matematika seringkali bisa dibuat.

Menurut Kadir, A. 2012 dalam jurnal Fatkhudin & Novianti (2015) Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang di maksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

2.1.4 Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata (Husda, 2012:117).

Informasi *(Information)* adalah data yang telah diolah menjadi suatu bentuk yang penting bagi sipenerima dan mempunyai nilai yang nyata atau dapat dirasakan manfaatnya dalam keputusan-keputusan yang akan datang (Ali & Wangdra, 2010:10).

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya (Sutabri, T. 2012 dalam jurnal Fatkhudin & Novianti (2015).

2.1.5 Sistem Informasi

Menurut Ali & Wangdra (2010:13) Sistem Informasi *(information system)* merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Sistem informasi dapat juga dikatakan sebagai suatu totalitas terpadu terdiri dari prosedur, tenaga pengolah *(brainware)*, perangkat lunak *(software)*, perangkat keras *(hardware)*, pangkalan data *(database)*, pernagkat telekomunikasi *(telecommunication)* yang saling ketergantungan dan saling menentukan dalam rangka menyediakan informasi untuk mendukung proses pengambilan keputusan.

Menurut Husda (2012:119) Sistem Informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi

operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Sutabri, T. 2012 dalam jurnal Fatkhudin & Novianti, 2015).

2.1.6 Data

Data adalah keterangan tertulis mengenai suatu fakta yang masih berdiri sendiri-sendiri, belum mempunyai pengertian sebagai kelompok, belum terkoordinasi satu sama lain dan belum diolah sesuai keperluan tertentu (Zulkifli, 2001 dalam jurnal Fatkhudin & Novianti, 2015).

2.2 Tinjauan Teori Khusus

2.2.1 Akomodasi Jangka Panjang

Menurut Kamus Besar Bahasa Indonesia (KBBI) Akomodasi adalah sesuatu yang disediakan untuk memenuhi kebutuhan, misalnya tempat menginap atau tempat tinggal sementara bagi orang yang bepergian.

Menurut Undang-Undang Republik Indonesia Tahun 2008 tentang penyelenggaraan ibadah haji, Akomodasi adalah perumahan atau pemondokan yang disediakan bagi Jemaah haji selama di embarkasi atau di debarkasi dan di Arab Saudi.

Akomodasi bisa juga diartikan tempat penginapan saat bepergian dimana orang yang harus tinggal jauh dari rumah lebih dari satu hari keperluan untuk tidur, istirahat, keselamatan, penyimpanan barang, serta akses ke fungsi umum pada rumah tangga. Penginapan dapat dilakukan pada hotel, apartemen, rumah komersial. Menurut Undang-Undang Republik Indonesia Tahun 2011 Tentang

Perumahan dan Kawasan Pemukiman, Rumah komersial adalah rumah yang diselenggarakan dengan tujuan mendapatkan keuntungan.

Akomodasi atau penginapan jangka panjang adalah perumahan atau rumah komersial atau bangunan yang disediakan oleh pemilik untuk kebutuhan menginap atau tempat tinggal sementara dengan jangka waktu yang ditentukan bagi orang yang bepergian atau yang merantau bisa berupa rumah atau kamar kos untuk mendapatkan keuntungan.

Akomodasi adalah tempat bagi seseorang untuk tinggal sementara, dapat berupa hotel, losmen, *guest house*, pondok, *cottage inn*, perkemahan, *caravan*, *bag packer* dan sebagainya. (Suwithi, Erwin, & Boham, 2008:20).

Akomodasi adalah suatu usaha yang menggunakan suatu bangunan atau sebagian bangunan yang disediakan secara khusus, di mana setiap orang dapat menginap dengan atau tanpa makan dan memperoleh pelayanan serta menggunakan fasilitas lainnya dengan pembayaran. Akomodasi lainnya meliputi: hotel melati yaitu hotel yang belum memenuhi persyaratan sebagai hotel berbintang seperti yang ditentukan oleh Diparda, penginapan remaja, pondok wisata, dan jasa akomodasi lainnya. (Wardhani, Viverawati, & Mustafa, 2008:37).

Sarana akomodasi (penginapan) dapat diartikan sebagai bangunan dalam bentuk apapun yang memiliki kamar-kamar tidur untuk menginap, baik dengan tambahan pelayanan makanan dan minum atau tidak. (Wardhani et al., 2008:36).

Kesimpulannya Akomodasi atau penginapan jangka panjang adalah perumahan atau rumah komersial atau bangunan yang disediakan oleh pemilik untuk kebutuhan menginap atau tempat tinggal sementara dengan jangka waktu

yang ditentukan bagi orang yang bepergian atau yang merantau bisa berupa rumah atau kamar kos baik dengan tambahan pelayanan makanan/minuman atau tidak untuk mendapatkan keuntungan.

Perumahan adalah kelompok rumah yang berfungsi sebagai lingkungan tempat tinggal atau tempat hunian yang dilengkapi dengan sarana dan prasarana (Undang-Undang Republik Indonesia, 2011 tentang perumahan dan pemukiman).

Dalam jurnal (Adibhadiansyah, 2016:69) kos merupakan sejenis kamar sewa yang disewa selama kurun waktu tertentu sesuai dengan perjanjian pemilik kamar dan harga yang disepakati. Namun demikian ada pula yang hanya menyewakan selama satu bulan, tiga bulan, dan enam bulan sehingga sebutannya menjadi sewa tahunan, tri bulanan dan tengah tahunan.

2.2.2 GPS (*Global Positioning System)*

Global Positioning System merupakan sistem navigasi yang dapat memberikan infomasi tentang sebuah lokasi. GPS merupakan teknologi yang awalnya digunakan untuk kepentingan militer dan sekarang dapat digunakan untuk kepentingan biasa (Fikri, Herumurti, & H, 2016:48).

Menurut Abidin (2006) dalam jurnal Adibhadiansyah (2016:69). *Global Positioning System* (GPS) merupakan sebuah sistem satelit navigasi dan penentuan posisi dengan menggunakan satelit. GPS dapat memberikan informasi tentang posisi, kecepatan, dan waktu secara cepat, akurat, murah dimana saja dibumi ini pada setiap saat tanpa tergantung cuaca.

2.2.3 *Google Maps*

Google Maps merupakan aplikasi antarmuka yang dikeluarkan oleh Google yang dapat diakses lewat *JavaScript. Google Maps* menyediakan layanan berbasis peta yang sangat *responsife* dan mudah dalam penggunaanya. Dengan menggunakan *Google Maps* ini pengguna dapat dengan mudah mencari suatu lokasi serta dapat melakukan penelusuran *route* menuju lokasi yang diinginkan (Sirenden dan Dachi: 2012 dalam jurnal Adibhadiansyah, 2016:69). Ditingkat pemrograman, *Google Maps* dapat dikembangkan dengan basis data, semua data yang terkait dengan titik lokasi disimpan dalam tabel dan ditampilkan sesuai keinginan pengguna. Isi tabel yang berisi data posisi peta dapat ditampilkan dengan menyajikan informasi lokasi yang menggunakan *Google Maps*. Pengguna tentunya akan mendapatkan informasi yang lebih detail terutama informasi lokasi perumahan dan kost tersebut.

Google Maps merupakan sebuah aplikasi GPS *online* yang menggunakan paket data untuk mengakses peta, karena menggunakan data secara langsung dari *Google Maps* sebagai *server*nya. Bedanya dengan GPS *Offline,* yaitu jika GPS *offline* petanya tersimpan di SD Card, sedangkan peta pada *Google Maps* ini tidak perlu melakukan penyimpanan pada memori tetapi langsung dari data *server* Google (Komputer, 2013:16).

2.2.4 Google Maps API

Google Maps API merupakan perkembangan dari *Google Maps*. Dengan menggunakan *Google Maps API* ini, dimungkinkan untuk dapat menggunakan *Google Maps* di dalam *website*. Meski awalnya hanya *JavaScript API, Maps API*

diperluas untuk menyertakan sebuah API untuk aplikasi *Adobe Flash*. Keberhasilan *Google Maps API* telah melahirkan sejumlah pesaing antara lain *Yahoo! Maps API, Bing Maps Platforn, MapQuest Development Platform* dan *OpenLayers* (Masykur, 2014:182).

Google Maps adalah merupakan SIG yang berbasis internet yang disediakan oleh *Google* secara gratis (bukan untuk kepentingan komersial), termasuk di dalamnya *website Google Maps, Google Ride Finder, Google Translate,* dan peta yang dapat disisipkan pada *website* lain melalui *Google Maps API*. Saat ini *Google Maps* adalah layanan pemetaan berbasis *web* yang popular. *User* dapat menambahkan layanan *Google Maps* ke sebuah *website* menggunakan *Google Maps API. Google Maps API* dapat ditambahkan ke sebuah *website* menggunakan *JavaScript.* API tersebut menyediakan banyak fasilitas dan utilitas untuk memanipulasi peta dan menambahkan konten ke peta melalui berbagai layanan, memungkinkan *user* untuk membuat aplikasi peta yang kuat pada sebuah *website*. Namun untuk dapat mengakses *Google Maps*, terlebih dahulu *user* harus melakukan pendaftaran API *key* dengan data pendaftaran berupa nama domain *web* yang kita bangun (Triansah, Cahyadi, & Astuti, 2015:59).

2.2.5 MySQL

Structure Query Language (SQL) adalah suatu bahasa komputer yang melekat pada suatu DBMS dan digunakan untuk mengakses atau melakukan permintaan data dalam sebuah sistem *database relational* (Husda, 2012:158).

Menurut Rosa & Shalahuddin (2011:146) *Structure Query Language* (SQL) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori *Aljabar* relasional dan kalkulus.

MySQL tergolong sebagai DBMS *(Databse Management System).* Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut:

1. Menyimpan data kedalam tabel
2. Menghapus data dalam tabel
3. Mengubah data dalam tabel
4. Mengambil data yang tersimpan dalam tabel
5. Memungkinkan untuk memilih data tertentu yang diambil
6. Memungkinkan untuk melakukan pengaturan hak akses terhadap data.

MySQL banyak dipakai untuk kepentingan penanganan *database* karena selain handal juga bersifat *open source*. Konsekuensi dari *open source* perangkat lunak ini dapat dipakai oleh siapa saja tanpa membayar dan *source code*-nya bisa diunduh oleh siapa (Kadir, 2010:10).

Sedangkan menurut Saputra (2012:77) MySQL merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. MySQL bekerja menggunakan bahasa SQL *(Structure Query Language)* yang merupakan bahasa standar yang digunakan untuk memanipulasi *database.*

2.2.6 *JSON*

JSON adalah struktur data yang *universal*, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh *JSON* dalam berbagai *format*. Hal ini memungkinkan *format* data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur *JSON*. *Format* data *JSON* mempunyai aturan sebagai berikut: *Object* adalah satu set nama/nilai yang tidak terurut *(An object is an unordered set of name/value pairs)*. Penulisan *object* dimulai dengan tanda { *(left brace)* dan diakhiri dengan tanda } *(right brace)*. Setiap nama diikuti oleh tanda : *(colon)* dan pasangan nama/nilai dipisahkan dengan tanda , *(comma)*. (Anggraini, Ardianty, & Widiyanto, 2014:242).

JSON *(JavaScript Object Notation)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat *(generate)* oleh komputer. *Format* ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript, Standar ECMA-262* Edisi ke-3 - Desember 1999. *JSON* merupakan *format* teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga *C* termasuk *C, C++, C#, Java, JavaScript, Perl, Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran data. (Herdiana, 2014:2).

JSON *(Java Script Object Notation)* adalah format pertukaran data yang bersifat ringan, disusun oleh *Douglas Crockford*. Fokus *JSON* adalah pada representasi data di *website*. *JSON* dirancang untuk memudahkan pertukaran data

pada situs dan merupakan perluasan dari fungsi-fungsi *JavaScript*. (Wijaya, Fenty, & Fiade, 2015:2).

2.2.7 *JQuery*

JQuery adalah sebuah *framework* berbasiskan *Javascript*. *JQuery* sama dengan *Javascript Library* yaitu kumpulan kode atau fungsi *Javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *Javascript*. Hal yang menarik dari *JQuery* adalah penekanan interaksi antara *javascript* dan HTML. (Warman & Zahni, 2013:33).

JQuery merupakan salah satu dari sekian banyak *framework* yang digunakan dan memilik penggunaan yang paling banyak. *JQuery* merupakan pustaka *JavaScript* yang dibangun untuk mempercepat dan memperingkas serta menyederhanakan manipulasi dokumen HTML, penangan *event*, animasi dan interaksi *Ajax* untuk mempercepat pengembangan web. Dengan *JQuery*, *developer* akan dimanjakan dengan suatu pemrograman *Javascript* yang sangat sederhana jika dibandingkan *native JavaScript*. Sebelum anda memulai mempelajari *JQuery*, anda harus mempunyai pengetahuan dasar mengenai HTML, CSS, dan *JavaScript*. (Komputer, 2012:2).

Dalam jurnal Warman & Zahni (2013:33) ada beberapa kemampuan yang dimiliki oleh *JQuery* sebagai berikut:

1. Memanipulasi elemen HTML
2. Memanipulasi CSS
3. Penanganan event HTML
4. Efek-efek *javascript* dan animasi

5. Modifikasi HTML DOM
6. AJAX

Kelebihan dan kekurangan *JQuery* yakni *Writeless, do more* yaitu menyederhanakan penggunaan *javascript* yang ada, karena kita cukup menggunakan fungsi *library javascript* yang ada, juga mempercepat *coding javascript* dalam sebuah *website*, dibandingkan kita harus memulai sebuah *script javascript* satu persatu.

JQuery Selector Selector adalah fungsi utama pada *JQuery*. Semua fungsi di *JQuery* dapat diakses melalui *selector*. Penggunaan paling dasarnya adalah mempassingkan sebuah *ekspressi* yang kemudian selanjutnya *JQuery* akan mencari elemen yang cocok. Pada intinya ini adalah fungsi untuk memilih elemen-elemen pada halaman web. (Warman & Zahni, 2013:33).

2.2.8 *JavaScript*

Dalam jurnal Yatini (2014:2) *JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. *JavaScript* digunakan dalam pembuatan website agar lebih *interaktif* dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. *JavaScript* dapat merespon perintah *user* dengan cepat dan menjadikan halaman web menjadi responsif. *JavaScript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi.

JavaScript merupakan skrip yang paling banyak digunakan dalam pemrograman web pada sisi client dewasa ini. Dengan adanya *JavaScript* sebuah

akan menjadi lebih hidup, cepat dan tampil lebih menawan dengan sebuah animasi. Dikarenakan lebih luasnya penggunaan *JavaScript*, banyak pengembang yang membangun sebuah *library*/pustaka *JavaScript* untuk memudahkan para *programmer website. Library*/pustaka inilah yang selanjutnya disebut sebagai *framework*, yaitu kumpulan fungsi-fungsi *JavaScript* yang siap digunakan untuk memanipulasi DOM *(Document Object Model)*. Banyak *framework* yang telah dikembangkan, antara lain *YUI, prototype, mootools,* dan *JQuery*. (Komputer, 2012:2).

2.2.9 UML *(Unified Modeling Language)*

Menurut Sholiq (2006) dalam jurnal Haryanti & Irianto (2011:11), UML *(Unified Modeling Language)* adalah sebuah bahasa yang berdasarkan grafik /gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO *(Object Oriented)*. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*.

Menurut Triansah et al. (2015:59) *Unified Modeling Languange* (UML) dirilis tahun 1987 sebagai sebuah metode untuk menggambarkan desain software. *Unified Modelling Language* (UML) sebagai notasi pemodelan standar industri untuk visualisasi sistem berorientasi obyek dan juga sebagai *platform* untuk mempercepat proses pengembangan aplikasi.

Keuntungan menggunakan UML adalah :

1. *Software* terdesain dan terdokumentasi secara profesional sebelum dibuat, dan dapat diketahui secara persis apa yang nantinya didapatkan.
2. Oleh karena mendesain terlebih dahulu, *reusable code* dapat dikode dengan tingkat efesiensi tinggi.
3. Lubang dapat diketemukan pada saat menggambar desain.
4. Dengan UML, dapat dilihat gambaran besarnya.

Menurut Rosa & Shalahuddin (2011:118), UML *(Unified Modeling Language)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. 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.

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa pemodelan berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek *(Object Oriented Programming).*

2.2.9.1 Langkah-langkah penggunaan *Unified Modeling Language* (UML)

Menurut Tantra (2012:152), tahapan penerapan *Unified Modeling Language* (UML) sebagai berikut:

1. Berdasarkan hasil analisis, susunlah semua proses bisnis secara *top down* dari *level* abstraksi yang tertinggi untuk mendefenisikan aktivitas dan proses yang ada.
2. Berdasarkan proses bisnis yang telah disusun tersebut, buat diagram *use case* sehingga semua fungsi sistem yang harus ada dapat terdidentifikasi. Tambahkan juga informasi penting lainnya yang diperlukan dalam pemrograman.
3. Defenisikan arsitektur fisik *software* dengan membuat *diagram deployment*. Kemudian juga defenisikan *requirement* lainnya seperti non-fungsional, *security* dan sebagainya yang harus dimiliki oleh sistem.
4. Susun *diagram activity* berdasarkan digram use case yang telah disusun sebelumnya.
5. Buat defenisi objek-objek yang ada pada *level* sistem yang paling umum. Kemudian berdasarkan *diagram use case*, untuk setiap aliran sistem, buat *diagram sequence* dan kolaborasi.
6. Lakukan desain antarmuka pengguna dari model yang sudah dibuat sehingga pengguna nantinya dapat melakukan simulasi berdasarkan skenario dari *use case*.
7. *Diagram class* dapat juga dibuat berdasarkan model yang sudah ada. Untuk itu maka setiap *package* harus diuraikan secara hirarkis menjadi *class*, termasuk juga mendefenisikan metode dan atribut *class* tersebut. Untuk membantu pengujian *class*, bisa juga dibuat suatu tes, untuk

memastikan *fungsionalitas class* tersebut dan interaksinya dengan *class* lainnya.

8. Berdasarkan susunan *class diagram*, maka lakukan pengelompokan *class* yang memiliki kesamaan karakteristik fungsionalitasnya menjadi komponen. Dengan demikian, diagram komponen dapat dibuat, termasuk juaga tes integrasi untuk memastikan interaksi antar komponen dapat berfungsi dengan baik.
9. Diagram *deployment* yang telah disusun sebelumnya pada tahap 3 lebih diperinci lagi dengan menambahkan informasi *requirement* dan kemampuan sistem, *platform*. Kemudian lakukan pemetaan komponen-komponen kedalam *node*.
10. Pada tahap ini, *software* sudah dapat mulai dibangun. Ada dua pendekatan yang dapat dipakai saat mulai melakukan pemrograman berdasarkan penggunaan UML ini, yaitu:
    a. Pendekatan secara *use case*, dimana *developer* melakukan pemrograman berdasarkan setiap *use case* yang ditugaskan pada tim *developer* tersebut untuk dibuatkan program beserta pengujiannya.
    b. Pendekatan secara komponen, dimana *developer* melakukan pemrograman berdasarkan komponen-komponen yang ditugaskan pada tim *developer* tersebut.
11. Setiap modul yang dihasilkan kemudian di uji per modul maupun secara terintegrasi. Lakukan perbaikan atau perubahan jika diperlukan,

berdasarkan hasil testing yang dilakukan. Perbaikan harus meliputi baik kode program maupun modelnya, karena model harus sesuai dengan kode program yang terbaru.

12. Setelah memalui tahap pengujian maka *software* sudah siap untuk *deliver.*

2.2.9.2 Konsep Pemodelan Menggunakan UML

Menurut Nugroho (2010:10), Sesungguhnya tidak ada batasan yag tegas diantara berbagai konsep dan konstruksi dalam UML, tetapi untuk menyederhanakannya, kita membagi sejumlah besar konsep dan dalam UML menjadi beberapa *view*. Suatu *view* sendiri pada dasarnya merupakan sejumlah konstruksi pemodelan UML yang merepresentasikan suatu aspek tertentu dari sistem atau perangkat lunak yang sedang kita kembangkan. Pada peringkat paling atas, *view-view* sesungguhnya dapat dibagi menjadi tiga area utama, yaitu: klasifikasi struktural *(structural classification)*, perilaku dinamis *(dinamic behaviour)*, serta pengolahan atau manajemen model *(model management).*

2.2.9.3 Bangunan dasar Metodologi *Unified Modeling Language* (UML)

Menurut Tantra (2012:149) Bangunan dasar yang menyusun UML adalah benda-benda *(things)* dan relasinya yang dikombinasikan dalam berbagai aturan untuk membentuk jenis-jenis diagram yang berbeda.

Menurut Yasin (2012:197) abstraksi konsep dasar UML terdidri dari *structural classification, dynamic behavior,* dan *model management,* bisa kita pahami dengan mudah. *Main concept* bisa kita pandang sebagai *term* yang akan muncul pada saat kita membuat diagram. Dan *view* adalah kategori dari diagram.

Lalu darimana kita mulai? Untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan.

1. menguasai pembuatan diagram UML
2. menguasai langkah-langkah dalam analisa dan pengembangan UML

2.2.9.4 Tujuan Penggunaan UML

Menurut Yasin (2012:197) ada beberapa tujuan penggunaan UML, sebagai berikut:

1. Pemodelan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. menciptakan bahasa pemodalan yang dapat digunakan baik oleh manusia maupun mesin.

Dalam jurnal Haviluddin (2011:2) Tujuan dari penggunaan diagram seperti diungkapan oleh Schmuller J. (2004), *"The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model"*. Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.

3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat *independen* terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

2.2.9.4 Jenis-jenis diagram UML *(Unified Modeling Language)*

Menurut Rosa & Shalahuddin (2011:121), Berikut ini adalah definisi mengenai 4 diagram UML:

1. Use Case Diagram

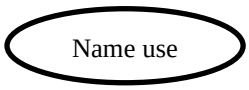
Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

a. **Aktor**: merupakan orang, proses, atau sistem lain 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.

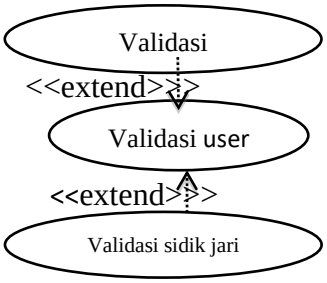
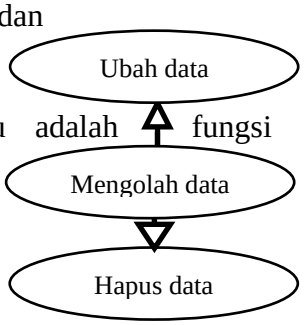
b. ***Use case***: merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Simbol dalam *Use case Diagram*:

Berikut adalah simbol-simbol yang ada pada diagram *use case* menurut Rosa & Shalahuddin (2011:130).

Tabel 2.1 Tabel Simbol *Use Case Diagram*

Simbol	Deskripsi
Use case 	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama *use case*.
Aktor / *actor* 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi / *association* 	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.

Tabel lanjutan 2.1

Simbol	Deskripsi
Ektensi / *extend* <<extend>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan yaitu , mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek, biasanya *use case* tambahan memiliki nama depan sama dengan *use case* yang ditambahkan misal Arah panah mengarah pada *use case* yang ditambahkan 
Generalisasi/*generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainya, misalnya Arah panah mengarah pada *use case* yang menjadi generalisasi (umum) 
Menggunakan / include / *uses* <<include>>  <<uses>> 	Relasi *use case* tambah ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case.* Ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini Ada dua sudut pandang yang cukup besar mengenai *include* di *use case*: *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* yang ditambahkan dijalankan. *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan. Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan

2. Class Diagram

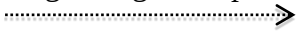
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Kelas Memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam suatu sebuah kelas. Dalam diagram kelas terdapat beberapa simbol dalam penggunaannya (Rosa & Shalahuddin, 2011:122).

Tabel 2.2 Tabel Simbol *Class Diagram*

Simbol	Deskripsi
Kelas **nama_kelas** +atribut +operasi()	Kelas pada struktur system
Antarmuka / *Interface* **nama_interface**	Sama dengan konsep *interface* dalam pemograman berorientasiobjek
Asosiasi / *association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*

Tabel lanjutan 2.2

Simbol	Deskripsi
Asosiasi berarah / *ditected association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Kebergantungan / *dependency* 	Relasi antara kelas dengan makna kebergantungan antar kelas
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua bagian (*whol –part*)

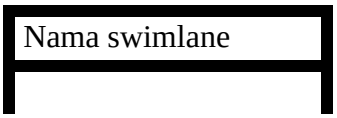
3. Activity Diagram

Menurut Rosa & Shalahuddin (2011:134) *Diagram Activity* memilki beberapa simbol dalam penggunannya. Berikut adalah simbol-simbol yang ada pada *Diagram Activity*:

Tabel 2.3 Tabel Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

Tabel Lanjutan 2.3

Simbol	Deskripsi
Penggabungan / *joint* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Activity Diagram (diagram aktivitas) menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis. *Activity Diagram* juga banyak digunakan untuk mendefinisikan hal-hal berikut:

a. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
b. Urutan atau pengelompokan tampilan dari sistem*/userinterface* dimana setiap aktivitas di anggap memiliki sebuah rencanngan antarmuka tampilan.
c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan khusus ujinya.

4. Sequence Diagram

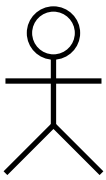
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk

menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

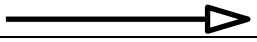
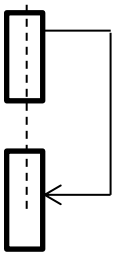
Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefenisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sihingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen (Rosa & Shalahuddin, 2011:138).

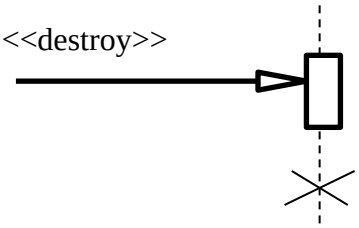
Tabel 2.4 Tabel Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor / *actor*  Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.

Tabel lanjutan 2.4

Simbol	Deskripsi
Garis hidup	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
pesan tipe create <<include>> 	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi yang ada pada objek lain atau dirinya sendiri 
Pesan tipe send 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data ke objek lainnya
pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu

Tabel Lanjutan 2.4

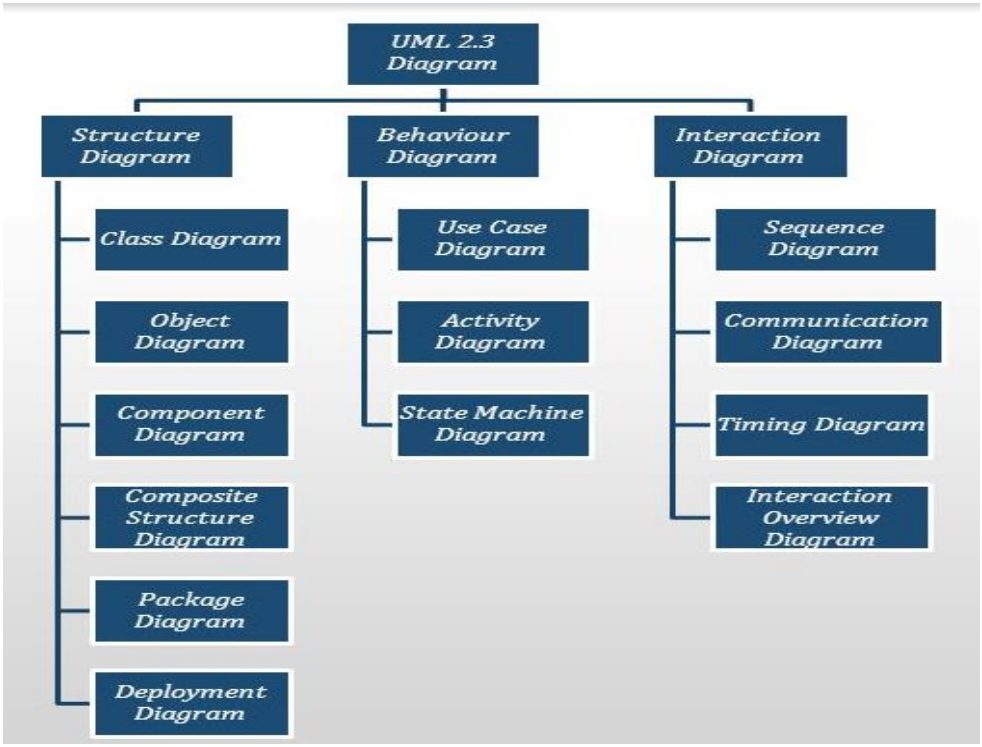
Simbol	Deskripsi
Pesan tipe destroy <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri.

Menurut Rosa & Shalahuddin (2011:120) UML terdiri dari 13 macam diagram yang dikelompokkan kedalam 3 kategori:

a. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari seistem yang dimodelkan.

b. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



Gambar 2.1 Diagram UML

2.2.7 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metodologi berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek. (Rosa & Shalahuddin, 2013:100).

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut, (Rosa & Shalahuddin, 2011:83):

1. meningkatkan produktivitas

   karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut *(reusable).*

2. kecepatan pengembangan

   karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. kemudahan pemeliharaan

   karena dengan model objek, pola-pola yang cendrung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin yang sering berubah-ubah.

4. adanya konsistensi

   karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. menungkatkan kualitas perangkat lunak

   karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangakat lunak yang sihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Berikut beberapa contoh bahasa pemroraman yang mendukung pemrograman berorientasi objek penulis:

1. Bahasa pemrograman (web) PHP

   PHP dibuat pertama kali oleh seorang perekayasa perangkat lunak yang bernama Rasmus Lerdoff. Rasmus Lerdoff membuat halaman web PHP pertamanya pada tahun 1994. PHP4 dengan versi-versi akhir menuju PHP5 sudah mendukung pemrograman berorientasi objek. PHP merupakan bahasa pemrograman yang digunakan untuk pemrograman web.

2.2.8 PHP (*Hypertext Prepocessor*)

Dalam jurnal Anggraini, Ardianty, & Widiyanto (2014:241) PHP merupakan singkatan dari PHP *Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnyalah yang dikirimkan ke klien tempat pemakai menggunakan *browser*.

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi PHP adalah forum (phpBB) dan *MediaWiki* (software di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari *ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl*. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah *Mambo, Joomla!, Postnuke, Xaraya,* dan lain-lain (Ramadhani, Anis, & Masruro, 2013:480).

PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan Web yang disisipkan pada dokumen HTML. (Kasiman Peranginangin, 2006 : 2. Dalam jurnal Nandari & Sukadi, 2014:44).

2.2.9 Pengertian Analisis SWOT

Dalam jurnal Remawati, n.d. menurut Rangkuti (2009: 18) Analisis SWOT adalah indentifikasi beberapa faktor secara sistematis untuk merumuskan strategi perusahaan. Analisis ini didasarkan pada logika yang dapat memaksimalkan kekuatan *(Strenghts)* dan peluang *(Opportunities)*, namun secara bersamaan dapat meminimalkan kelemahan *(Weaknesses)* dan ancaman *(Threats)*. Analisis SWOT secara sederhana dipahami sebagai pengujian terhadap kekuatan dan kelemahan internal sebuah organisasi, serta kesempatan dan ancaman lingkungan eksternalnya. SWOT adalah perangkat umum yang didesain dan digunakan sebagai langkah awal dalam proses pembuatan keputusan dan sebagai perencanaan strategis dalam berbagai terapan.

Penelitian ini menunjukkan bahwa kinerja perusahaan dapat ditentukan oleh kombinasi faktor *internal* dan *eksternal*. Kedua faktor tersebut harus dipertimbangkan dalam analisis SWOT. SWOT adalah singkatan dari lingkungan *Internal Strenghths* dan *Weaknesses* serta lingkungan *Eksternal Opportunities* dan *Threats* yang dihadapi dunia bisnis. Analisis SWOT membandingkan antara faktor *eksternal* Peluang *(opportunities)* dan Ancaman *(threats)* dengan faktor *internal* Kekuatan *(strengths)* dan Kelemahan *(weaknesses)*.

2.2.10 *Bootstrap*

Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat dengan alat bantu ini memiliki tampilan halaman yang mirip dengan tampilan halaman *Twitter*. Meskipun demikian, desainer juga dapat mengubah tampilan halaman website sesuai dengan kebutuhan.

Twitter Bootstrap dibangun dengan teknologi HTML dan CSS yang dapat membuat *layout* halaman *website*, tabel, tombol, *form*, *navigasi*, dan komponen lainnya dalam sebuah *website* dengan hanya memanggil fungsi CSS *(class)* dalam berkas HTML yang telah didefinisikan. Selain itu terdapat juga, komponen-komponen lainnya yang dibangun menggunakan *JavaScript*. (Zubaidi, 2015:9).

Bootstrap adalah salah satu *frameworks* yang bisa digunakan untuk membuat aplikasi web atau situs *web responsive* secara cepat, handal, mudah dan gratis. *Bootstrap* terdiri dari CSS dan HTML yang digunakan untuk mengelola *Grid, Layout, Tipografi, Tabel, Icon, Button, Navigation*, dan lain-lain. Didalam *Bootstrap* juga sudah ada *jQuery* untuk mengelola komponen UI seperti *Transition, Caousell, Dropdown, Tooltip* dan lain-lain, sehingga memudahkan pengguna untuk mendesain sebuah aplikasi atau situs berbasis web. (Hariyanto, 2017:146).

Pada awalnya, *Bootstrap* diciptakan pada desain web media sosial *Twitter* oleh dua orang *programmer*nya , yaitu Mark Otto dan Jacob Thronthon pada tahun 2011. Oleh karena itu walaupun nama resminya *Bootstrap*, tetapi

dikalangan *developer web* lebih dikenal dengan nama *Twitter Bootstrap*. Sejak awal diluncurkan sampai sekarang *Bootstrap* sudah berevolusi menjadi sebuah *tool framework* yang lebih lengkap dan juga berisi *JavaScript Plugin, Icon, Form* dan *Button*. *Bootstrap* dapat kita peroleh secara gratis disitus resminya dengan alamat http://getbootstrap.com. (Hariyanto, 2017:146).

Bootstrap is a free and open-source front-end web structure for arranging locales and web applications. It contains HTML-and CSS-based design formats for typography, shapes, gets, course and other interface parts, furthermore optional JavaScript developments. Not at all like various web frameworks, it stresses over front-end change in a manner of speaking. Bootstrap, at first named Twitter Blueprint, was delivered by Jacob Thornton and Mark Otto at Twitter as a structure to engage consistency transversely over internal mechanical assemblies. (Durganath & Edreena, 2016: 21116).

Bootstrap is a framework utilizing HTML, CSS, and JavaScript to develop webpages. It was originally created by Twitter in 2010. It is an open source project. Beginning from version 2, it is capable to design responsive webpages. Now, Bootstrap 4 is being developed. (Murai & Klyuev, 2016:60).

2.2.11 *Notepad++*

Notepad++ adalah sebuah aplikasi *text editor* yang bersifat gratis. *Notepad* menitikberatkan kegunaan aplikasi untuk *editing text* dalam waktu yang cepat dan praktis. *Notepad++* mendukung banyak *format* bahasa pemrograman seperti *PHP, HTML, JavaScript* dan *CSS*. (Palevi & Krisnawati, 2013:4).

2.2.12 HTML *(HyperText Markup Language)*

HTML adalah singkatan *HyperText Markup Language*, merupakan *file* teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi *web browser*. (Raharjo dalam jurnal Kuncoro, 2012:37).

Dalam jurnal Binarso, Sarwoko, & Bahtiar (2012:76) HTML *(Hyper Text Markup Language)* sebenarnya bukan sebuah bahasa pemrograman, karena HTML adalah bahasa *mark up*. HTML digunakan untuk *mark up* (penanda) terhadap suatu dokumen teks. Simbol *mark up* yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Kedua tanda ini disebut tag. Tag yang digunakan sebagai tanda penutup diberi karakter garis miring (</..>).

HTML *(Hyper Text Markup Language)* merupakan bahasa asli dari www, yang telah menjadi bahasa standar untuk menampilkan data di internet. Perkembangan html sangatlah pesat, saat ini versi terakhir dari html telah mencapai html 5. (Kusuma, 2015).

2.2.13 CSS *(Cascading Style Sheet)*

CSS merupakan kependekan dari *Cascading Style Sheet*, yang digunakan untuk membantu anda mendesain isi halaman web. Misalnya anda mempunyai halaman web yang terdiri dari beberapa *file*, untuk melakukan pemformatan pada halaman tersebut, anda tidak perlu memformat satu persatu, tetapi anda cukup membuat satu file CSS. CSS distandarisasi oleh W3C *(World Wide Web*

Consortium). CSS dapat dipasang pada dokumen HTML/XHTML yang telah jadi. (Madcoms, 2009:89).

CSS *(Cascading Style Sheet)* adalah *stylesheet language* yang digunakan untuk mendeskripsikan penyajian dari dokumen yang dibuat dalam *mark up language.* CSS merupakan sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman web, inti dari dokumen ini adalah memformat halaman web standar menjadi bentuk web yang memiliki kualitas yang lebih indah dan menarik. (Binarso et al., 2012:76)

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML, XML, dan XHTML. CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen dengan presentasi dokumen (yang ditulis dengan CSS). Memungkinkan juga untuk halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh *browser* basis-suara atau pembaca layar), dan juga alat pembaca *braille*. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

2.3 Penelitian Terdahulu

1. **(Pratikto, Sutanta, & Suraya, 2014)** dengan judul Sistem Pencarian dan Pemesanan Rumah Kos Menggunakan Sistem Informasi Geografis (SIG). Tujuan dari *website*nya tersebut untuk memberikan informasi rumah kos dengan memanfaatkan *Google Maps* untuk menunjuk lokasi rumah kos beserta detail infomasi kosnya serta objek yang terdekat dengan rumah kos tersebut. Kelebihannya yaitu pada penentuan lokasi pada petanya sudah lebih spesifik. Untuk kekurangannya yaitu pemesanannya hanya menunjuk rumah kosnya, belum menunjuk pada pemesanannya langsung dengan kamar kosnya saja.
2. **Fatkhudin & Novianti (2015)** dengan judul Sistem Informasi Pemesanan Rumah Kos Di Kota Pekalongan Berbasis Website. Beliau membuat sebuah sistem informasi yang menyediakan berbagai informasi, ketersediaan kamar kos dan bagaimana melakukan pemesanan kos secara online bagi para pencari kos serta membantu pemilik kos untuk mempromosikan rumah kosnya, serta akses jaringan yang luas, karena berbasis *website* sehingga dapat di akses kapan saja dan dimana saja.
3. **Ramadhani et al. (2013)** dengan judul Rancang Bangun Sistem Informasi Geografis Layanan Kesehatan Di Kecamatan Lamongan dengan PHP MySQL, dari hasil penelitian ini, dapat diambil kesimpulan bahwa telah dihasilkan suatu Sistem Informasi Geografis Layanan Kesehatan di Kecamatan Lamongan Dengan *PHP MySQL*,

yang dapat membantu memudahkan masyarakat memperoleh informasi letak layanan kesehatan di Kecamatan Lamongan kapanpun dan dimanapun melalui internet. Selain itu, sebagai teknologi alternatif dalam perkembangan dunia kesehatan serta mempermudah Dinas Kesehatan Kabupaten Lamongan dalam memantau sebaran dari layanan kesehatan di Kecamatan Lamongan.

4. **Kosasi (2014)** dengan judul Sistem Informasi Geografis Pemetaan Tempat Kos Berbasis Web. Dari hasil penelitian ini memperlihatkan bahwa dalam pengembangan sistem informasi geografis berbasis web untuk memetakan lokasi tempat (rumah) kos khusus di Kecamatan Pontianak Utara. Tahapan dalam perancangan sistem informasi geografis pemetaan lokasi tempat (rumah) kos berbasis web mengacu kepada metode pengembangan *waterfall*. Sistem informasi geografis pemetaan lokasi tempat (rumah) kos berbasis web terdiri dari dua halaman utama yaitu halaman yang dipergunakan untuk admin dan halaman yang dipergunakan untuk pemilik (rumah) kos. Sistem informasi geografis pemetaan lokasi rumah kos berbasis web adalah sebuah sistem yang menampilkan informasi letak tempat kos dalam bentuk peta yang disertai dengan informasi rumah kos. Sistem informasi geografis berbasis web dapat menjangkau kebutuhan masyarakat secara luas dan tidak hanya fokus kepada masyarakat di Kecamatan Pontianak Utara.