BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1. Perancangan

Menurut Jogiyanto dalam (Hatmoko, 2014:60), perancangan merupakan tahap persiapan untuk rancang bangun implementasi suatu sistem, yang menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan, dan pembuatan sketsa atau peraturan dari beberapa elemen terpisah ke dalam suatu kesatuan yang utuh dan berfungsi termasuk mengkonfigurasi komponen-komponen perangkat lunak dan perangkat dari suatu sistem.

2.1.2. Sistem

Menurut Husda (2013:111) Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari berbagai elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai

penggeraknya yaitu rakyat yang berada dinegara tersebut. Contoh lainnya adalah sistem komputer terdiri dari, *software, hardware, brainware*.

Menurut Jogiyanto, H.M., dalam (Andi Muh. Lukman, 2016:44), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Menurut Jogianto dalam (Hatmoko, 2014:60), mengemukakan bahwa "sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul Bersama-sama untuk melakukan suatu kegiatan untuk menyelesaikan suatu sasaran tertentu". Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi.

Syarat-syarat, (Husda, 2013:112). yaitu:

1) Sistem harus dibentuk untuk menyelesaikan tujuan.
2) Elemen sistem harus mempunyai rencana yang ditetapkan.
3) Adanya hubungan diantara elemen sistem.
4) Unsur dasar dari proses (arus informasi, energi, dan material) lebih penting dari pada elemen sistem.
5) Tujuan organisasi lebih penting dari pada tujuan elemen.

a. Karakteristik Sistem

Karakteristik yang harus dimiliki oleh suatu sistem, (Husda, 2013:112). yaitu:

1) Komponen (*Component*)

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau *subsistem-subsistem*. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2) Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari *subsistem* yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3) Lingkungan Luar Sistem (*Environment*)

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang

menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

4) Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu *subsistem* dengan *subsistem* yang lainnya. Untuk membentuk suatu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain output dari suatu subsistem akan menjadi input subsistem yang lainnya.

5) Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa:

a) Masukan perawatan (*maintenance input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi.
b) Masukan sinyal (*signal input*) adalah energi yang diproses untuk didapatkan keluaran.

Sebagai contoh didalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan

komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

6) Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem. Meliputi:

a) Keluaran yang berguna, contohnya informasi yang dikeluaran oleh komputer.

b) Keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

7) Pengolah Sistem *(Process)*

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi, bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8) Tujuan Sistem *(Objective)*

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalua pengoperasian sistem itu mengenai sasaran atau tujuannya.

Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

b. Klasifikasi Sistem

Klasifikasi suatu sistem, (Husda, 2013:115). yaitu:

1) Sistem Abstrak (*Abstract System*), Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (Sistem *Teologia* yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dan Manusia).

2) Sistem Fisik (*Physical System*), Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya (sistem komputer, sistem akuntansi, sistem produksi, dll.)

3) Sistem Alamiah (*Natural System*), Sistem yang terjadi melaui proses alam dalam artian tidak dibuat oleh manusia. (sistem tata surya, sistem *galaxy*, sistem reproduksi, dll.)

4) Sistem Buatan Manusia (*Human Made System*), Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system* (contoh: sistem informasi).

5) Sistem Tertentu (*Deterministic System*), Sistem beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh: sistem komputer).

6) Sistem tak tentu (*Probabilistic System*), Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. (contoh: sistem manusia)

7) Sistem Tertutup (*Close System*), Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relativeliy closed system* (secara relativ tertutup, tidak benar-benar tertutup).

8) Sistem Terbuka (*Open System*), Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomatis, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat *modern*.

2.1.3. Informasi

Menurut Husda (2013:117), secara umum 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 informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah suatu yang terjadi pada saat tertentu.

Menurut G. Davis dalam (Andi Muh. Lukman, 2016:44), informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

Menurut Hartono dalam (Hatmoko, 2014:60), informasi adalah suatu data yang telah diproses sehingga dapat mengurangi ketidakjelasan tentang keadaan atau kejadian". Maksud dari kata data itu sendiri adalah fakta atau kenyataan yang sebenarnya. Pengertian informasi juga dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian *(event)* yang nyata *(fact)* yang digunakan untuk pengambilan keputusan.

2.1.4. Sistem Informasi

Menurut Sutabri, Tata. dalam (Andi Muh. Lukman, 2016:44), sistem informasi adalah suatu sistem di dalam 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 kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Menurut Husda (2013:119), sistem informasi dapat didefinisikan sebagai suatu sistem di dalam 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.

a. Sifat-sifat Sistem Informasi

Menurut Husda (2013:120), sistem informasi harus mempunyai beberapa sifat seperti:

1) Pemrosesan informasi yang efektif. Hal ini berhubungan dengan pengujian terhadap data yang masuk, pemakaian perangkat keras dan perangkat lunak yang sesuai.
2) Manajemen informasi yang efektif. Dengan kata lain, operasi manajemen, keamanan dan keutuhan data yang ada harus diperhatikan.
3) Keluwesan. Sistem informasi hendaknya cukup luwes untuk menangani suatu macam operasi.

4) Kepuasan pemakai. Hal yang paling penting adalah pemakai mendapatkan manfaat dan puas terhadap sistem informasi.

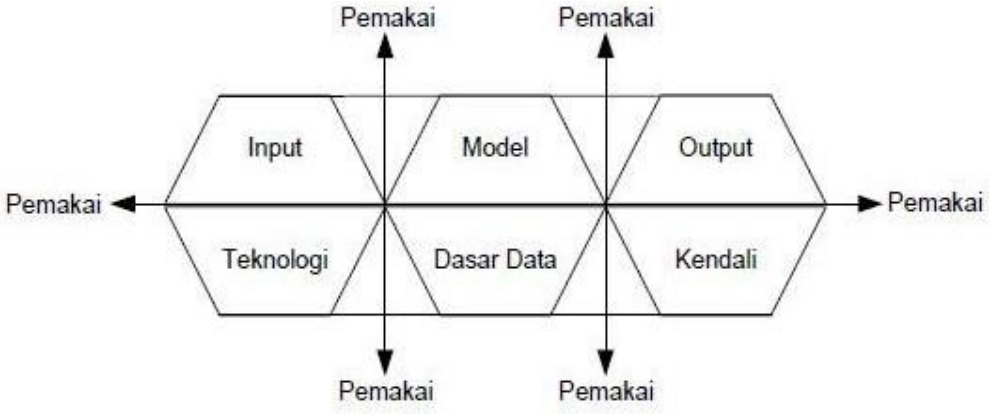
b. Komponen Sistem Informasi

Menurut Husda (2013:120), Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), yaitu:

1) Blok Masukan (*Input Block*), Input merupakan data yang masuk ke dalam sistem informasi.
2) Blok Model (*Model Block*), kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3) Blok Keluaran (*Output Block*), keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingakatan manajemen serta semua pemakai sistem.
4) Blok Teknologi (*Technology Block*), teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.
5) Blok Basis data (*Database Block*), merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di

perangkat keras computer dan digunakan perangkat lunak untuk memanipulasinya.

6) Blok Kendali (*Control Block*), beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.



Gambar 2. 1 Komponen Sistem Informasi

2.1.5. Pariwisata

Menurut Hatmoko (2014:60), pariwisata adalah suatu perjalanan yang dilakuan untuk rekreasi atau liburan dan juga persiapan yang dilakukan untuk kegiatan ini. Seorang wisatawan atau turis adalah seseorang yang melakukan perjalanan paling tidak sejauh 80 km (50 mil) dari rumahnya dengan tujuan rekreasi, adalah pengertian definisi dari Organisasi Pariwisata Dunia.

Pengertian pariwisata juga diperkuat dengan adanya Perpres (Peraturan Presiden) Republik Indonesia, menurut Undang-Undang Nomor 10 Tahun 2009 tentang Kepariwisataan:

a. Pariwisata adalah berbagai macam kegiatan wisata dan dudukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, pemerintah pusat, dan pemerintah daerah.
b. Kepariwisataan adalah keseluruhan kegiatan yang terkait dengan pariwisata bersifat multidimensi serta multidisiplin yang muncul sebagai wujud kebutuhan setiap orang dan negara serta interaksi antara wisatawan dan masyarakat ketempat, sesame wisatawan, Pemerintah, Pemerintah Daerah, dan pengusaha.
c. Pengawasan dan Pengendalian Kegiatan Kepariwisataan adalah sistem dan mekanisme pencegahan dan penanggulangan dampak negatif dari kegiatan kepariwisataan.
d. Setiap Orang adalah perorangan, kelompok orang, masyarakat, badan usaha berbadan hukum dan/atau badan usaha bukan badan hukum.
e. Wisatawan adalah orang yang melakukan wisata.
f. Pengusaha Pariwisata adalah orang atau sekelompok orang yang melakukan kegiatan usaha pariwisata.
g. Pemerintah adalah Presiden Republik Indonesia yang memegang kekuasaan pemerintahan Negara Republik Indonesia sebagaimana dimaksud dalam UndangUndang Dasar Negara Republik Indonesia Tahun 1945.
h. Pemerintah Daerah adalah Gubernur, Bupati, atau Walikota dan perangkat daerah sebagai unsur penyelenggara pemerintahan daerah.

i. Menteri adalah menteri yang menyelenggarakan urusan pemerintahan di bidang kepariwisataan.

2.1.6. *Android*

Menurut Hatmoko (2014:61), *android* adalah sistem operasi berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Menurut Nasruddin, S.H. dalam (Santoso & Rais, 2015:31), *android* adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, *Google Inc*. mengakusisi *Android Inc*. Yang mengembangkan *software* untuk ponsel yang berada di Palo Alto, California Amerika Serikat.

Menurut (Udayana, Wirawan, & Sunarya, 2015), *android* merupakan *platform mobile* pertama yang lengkap, terbuka dan bebas. Lengkap (*complete platform*) berarti para pengembang dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform android*. Terbuka (*Open Source Platform*) berarti *platform android* berlisensi *open source* sehingga pengembang dapat dengan bebas mengembangkan aplikasi. Sedangkan bebas *(free)* yaitu tidak ada lisensi atau biaya royalti untuk pengembangan *platform android*, tidak ada biaya keanggotaan yang diperlukan, aplikasi *android* dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Menurut Andi Muh. Lukman (2016:45), *android* adalah sistem informasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

2.2. Tinjauan Teori Khusus

2.2.1. *MySQL*

Menurut (Haryanti & Irianto, 2011) *MySQL* merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data terutama dapat dikerjakan dengan mudah dan otomatis. Kepopuleran *MySQL* dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah kecil. *MySQL* merupakan database yang digunakan oleh situs-situs terkemuka di Internet untuk menyimpan datanya. *Software* database *MySQL* kini dilepas sebagai *software* manajemen *database* yang *open source*, sebelumnya merupakan *software database* yang *shareware*.

Menurut Februariyanti & Zuliarso (2012:128), *MySQL* adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan Batasan

perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Menurut Peranginangin, Kasiman dalam (Andi Muh. Lukman, 2016:45), *MySQL* adalah suatu *relational database management system* (RDBMS) yang mendukung *database* yang terdiri dari sekumpulan relasi atau tabel. SQL distandarisasi sebagai bahasa untuk menciptakan *database*, dan menyimpan informasi ke dalam *database*, dan mendapatkan kembali informasi darinya.

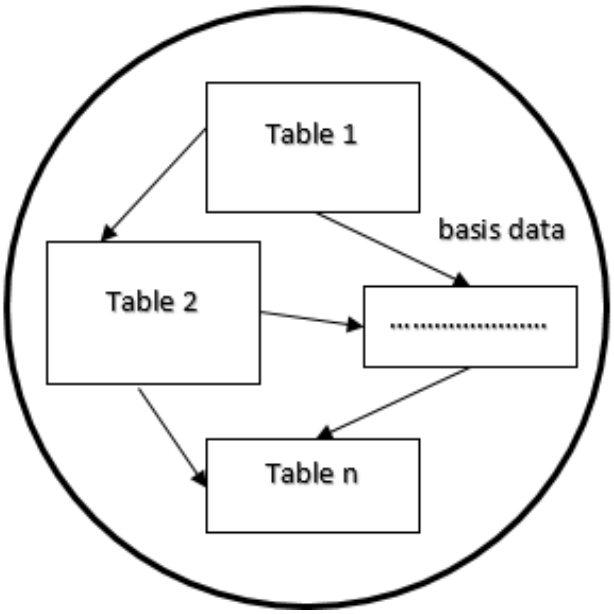
2.2.2. Basis Data

Menurut Marlinda dalam (Andi Muh. Lukman, 2016:45), basis data adalah suatu susunan atau kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Menurut Abdul Kadir dalam (Andi Muh. Lukman, 2016:45), basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktifitas untuk memperoleh informasi yang optimal. Database managemen sistem (DBMS) merupakan perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses

basis data dengan cara yang praktis dan efisien. DBMS digunakan untuk mengakomodasi berbagai macam yang memiliki kebutuhan akses yang berbeda-beda.

Menurut A.S & Shalahuddin (2011:44) Sistem basis data adalah sistem terkomputasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Pada buku ini menggunakan basis data relasional yang diimplementasikan dengan table-tabel yang saling memiliki relasi seperti pada gambar berikut.



Gambar 2. 2 Ilustrasi Basis Data

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa *file* teks ataupun *database management system* (DBMS).

Kebutuhan basis data dalam sistem informasi meliputi:

a. Memasukkan, menyimpan, dan mengambil data.

b. Membuat laporan berdasarkan data yang telah disimpan.

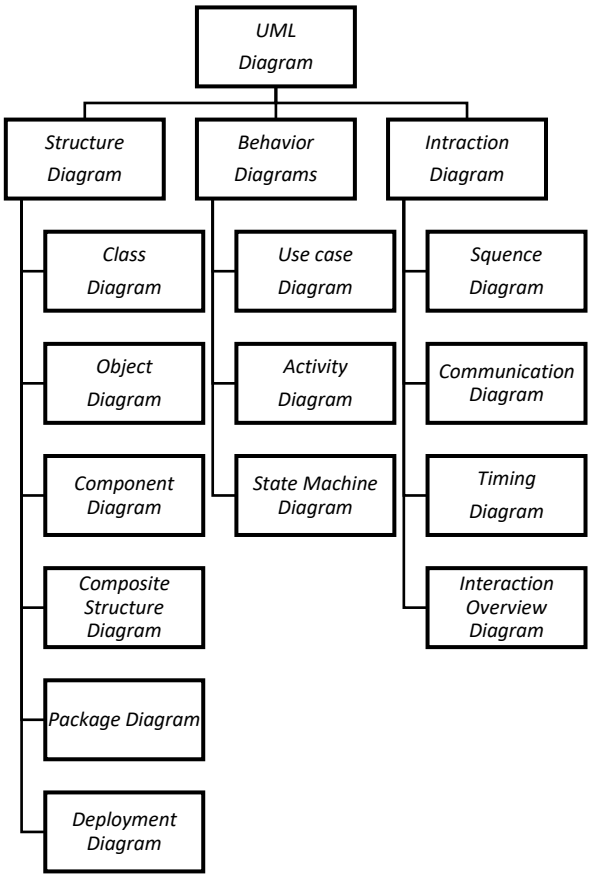
2.2.3. UML (*Unified Modeling Language*)

Menurut A.S & Shalahuddin (2011:117), pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karna itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *unified modeling language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML 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.

UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah.



Gambar 2. 3 Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

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

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

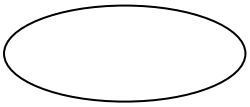
c. *Interaction* diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Penjelasan tentang beberapa diagram akan dilakukan pada *submodul-submodul* berikutnya.

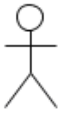
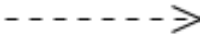
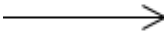
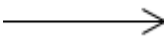
a. *Use-case* Diagram

Menurut A.S & Shalahuddin (2011:130), *Use case* 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 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. Simbol-simbol yang digunakan dalam diagram *Use case* adalah:

Tabel 2. 1 Simbol *Use case* Diagram

GAMBAR	DESKRIPSI
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama *use case.*

Tabel 2. 1 Lanjutan

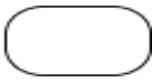
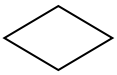
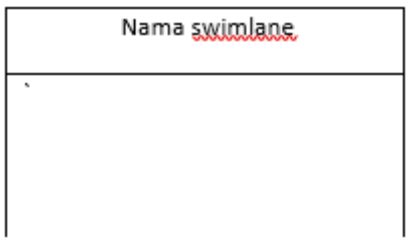
Aktor/*Actor* 	Orang, proses, atau sistem lain yang beriteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi/*Association* 	Komunikasi actor dan *use case* yang berpartisipasi pada *use case* atau *use case* memliki interaksi dengan actor.
Ekstensi/*Extend* <<Extend>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *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 yang lain.
Menggunakan/*Include*/*Uses* <<include>> <<uses>> 	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.

b. *Activity* diagram

Menurut A.S & Shalahuddin (2011:134), Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Berikut adalah simbol-simbol yang digunakan dalam diagram aktifitas:

Tabel 2. 2 Simbol *Activity* diagram

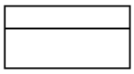
GAMBAR	DESKRIPSI
Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/*join*	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  atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Tabel 2. 2 Lanjutan

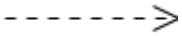
c. *Class* Diagram

Menurut A.S & Shalahuddin (2011:122), Diagram Kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut adalah simbol-simbol yang digunakan dalam diagram kelas:

Tabel 2. 3 Simbol *Class* Diagram

GAMBAR	DESKRIPSI
Kelas 	Kelas pada struktur sistem
Antarmuka/*interface*	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi/*assosiation*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.

Tabel 2. 3 Lanjutan

Asosiasi berarah/*directed assosiation*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan/*dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi/*aggregation*	Relasi antar kelas dengan makna semua-bagian (*whole-part*).

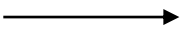
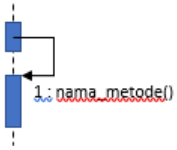
d. *Sequence* Diagram

Menurut A.S & Shalahuddin (2011:137), 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 menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstasiasi menjadi objek itu. Berikut adalah simbol-simbol yang digunakan dalam diagram sekuen:

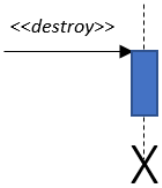
Tabel 2. 4 Simbol *Sequence* Diagram

GAMBAR	DESKRIPSI
Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi aktor belum tentu merupakan orang;

Tabel 2. 4 Lanjutan

Atau nama aktor Tanpa waktu aktif	biasanya dinyatakan menggunakan kata benda di awal frase nama akor.
Garis hidup/*lifeline* 	Menyatakan kehidupan suatu objek
Objek nama objek : nama kelas	Menyatakan objek yang beriteraksi pesan
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan beriteraksi pesan.
Pesan tipe *create* <<*create*>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *call* 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.

Tabel 2. 4 Lanjutan

Pesan tipe *send* 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
Pesan tipe *return* 1 : masukan 	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.
Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

2.2.4. SDLC

Menurut A.S & Shalahuddin (2011:24), SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Seperti halnya proses metamorfosis pada kupu-kupu, untuk menjadi kupu-kupu yang indah maka dibutuhkan beberapa tahap untuk dilalui, sama halnya

dengan membuat perangkat lunak, memiliki daur tahapan yang dilalui agar menghasilkan perangkat lunak yang berkualitas.

Tahapan-tahapan yang ada pada *SDLC* secara global adalah sebagai berikut:

a. Inisiasi (*Initiation*)

   Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

b. Pengembangan konsep sistem (*System concept development*).

   Mendefinisikan lingkup konsep termasuk dokumen lingkup *system* analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

c. Perencanaan (*planning*)

   Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*recource*) yang dibutuhkan untuk memperoleh solusi.

d. Analisis Kebutuhan (*requirement analysis*)

   Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

e. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

f. Pengembangan (*development*)

Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program, peninjauan pengujian.

g. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

h. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan

resolusi dari permasalahan yang teridentifikasi dari integrasi dan pengujian.

i. Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*) termasuk implementasi akhir dan masuk pada proses peninjauan.

j. Disposisi (*disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

Ada beberapa model SDLC yang dapat digunakan. Semuanya memiliki kelemahan dan kelebihan pada setiap model SDLC. Hal terpenting adalah mengenali tipe pelanggan (*customer*) dan memilih menggunakan model SDLC yang sesuai dengan karakter pelanggan (*customer*) dan sesuai dengan karakter pengembang,

2.2.5. *JavaScript*

Menurut Sidik dalam (Prayitno & Safitri, 2015:2) menjelaskan bahwa "JavaScript adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif, tidak sekedar indah saja".

Menurut Sidik dalam (Prasetio, 2014) menjelaskan "*JavaScript* adalah program dalam bentuk script, yang dijalankan oleh interpreter yang telah ditanamkan ke dalam browser web, sehingga browser web dapat mengeksekusi program *JavaScript*."

2.2.6. HTML

Menurut Winarno dan Utomo dalam (Prayitno & Safitri, 2015:2) "HTML singkatan dari *Hypertext Markup Language* dan berguna untuk menampilkan halaman web"

2.2.7. PHP (*PHP Hypertext Preprocessor*)

Menurut Anhar dalam (Prayitno & Safitri, 2015:2) PHP adalah (PHP Hypertext Preprocessor) adalah bahasa pemrograman web berupa *script* yang dapat diintegrasikan dengan HTML.

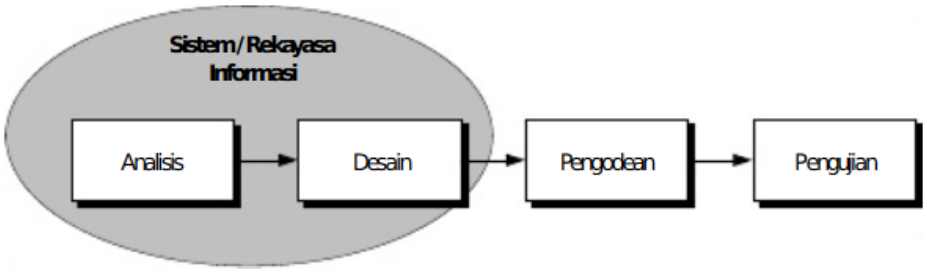
2.2.8. XAMPP

Menurut Wahana dalam (Prayitno & Safitri, 2015:2) XAMPP adalah salah satu paket instalasi *apache*, PHP, dan *MySql* secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut.

2.2.9. Model *Waterfall*

Menurut A.S & Shalahuddin (2011:26) Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik

(*classic life cycle*)". Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). Berikut adalah gambar model air terjun:



Gambar 2. 4 Ilustrasi model *waterfall*

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain

perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari tahap analisis spesifikasi untuk perubahan perangkat lunak baru.

Dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai dengan alurnya karena sebab berikut:

a. Perubahan spesifikasi perangkat lunak terjadi ditengah alur pengembangan.
b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan. Pelanggan sering kali butuh contoh *(prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan model di akhir alur pengembangan.

Dengan berbagai kelemahan yang dimiliki model air terjun tapi model ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikan model pengembangan perangkat lunak.

Model air terjun sangat cocok digunakan kebutuhan pelanggan sudah sangat dipahami dan kemungkinan terjadi perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari model air terjun adalah struktur tahap pengembangan sistem jelas, dokumentasi di hasilkan di setiap tahap pengembangan dan sebuah tahap di jalankan setelah tahap sebelumnya selesai di jalankan (tidak ada tumpang tindih pelaksanaan tahap).

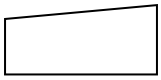
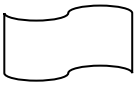
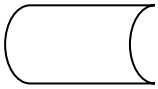
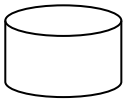
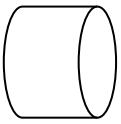
"Model *waterfall* adalah model *SLDC* yang paling sederhana. Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah".

2.2.10. Aliran Sistem Informasi

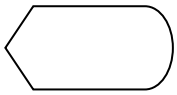
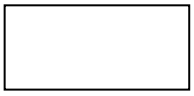
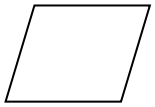
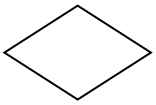
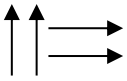
Menurut Iswandy (2015:73), *Flowchart* merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis.

Simbol yang digunakan untuk membuat aliran sitem informasi adalah sebagai berikut:

Tabel 2. 5 Simbol Aliran Sistem Informasi

SIMBOL	KETERANGAN
	*Manual Input* (*Keyboard*), sebagai *entry* data
	Card, sumber data berasal dari kartu data
	*Punched tape*, menunjukkan mesin pembolong pita
	*Stored data*, media penyimpanan data dapat berupa *harddisk*
	Sequential access storage, media penyimpanan data berupa pita yang dibaca berurut
	*Magnetic Disk*, media penyimpanan data seperti *floppy disk*
	*Direct access storage*, media penyimpanan data yang dapat dibaca/disimpan secara acak

Tabel 2. 5 Lanjutan

	*Display*, media untuk melihat hasil proses dilayar (*monitor*)
	Document, menunjukkan data akan dicetak ke kertas melalui mesin *printer*
	Terminator, menunjukkan awal dan akhir dari suatu alur program *flowchart*
	Proses, menunjukkan proses seperti perhitungan aritmatik, penulisan suatu formula atau dapat berisi pemberian nilai terhadap variabel
	*Read/Write*, menunjukkan sumber data yang akan diproses atau dapat juga menunjukkan data yang akan dicetak/ditulis
	*Decision*, menunjukkan suatu proses evaluasi atau pemeriksaaan terhadap nilai data dengan operator relasi
	Sub program, menunjukkan *sub program* yang akan diproses dapat berupa *procedure* dan *function*
	Connector, menujukkan tanda sambungan dari suatu *Flowchart* pada satu halaman kertas
	Off page connector, menunjukkan tanda sambungan dari suatu *Flowchart* untuk beda halaman kertas
	*Arrow*, menunjukkan arah dari suatu proses dapat ke atas, bawah, kanan, dan kiri.

2.3. Penelitian Terdahulu

Berikut ini adalah tabel kesimpulan beberapa jurnal yang dipakai dalam pembuatan dan penyusunan skripsi ini.

Tabel 2. 6 Penelitian Terdahulu

Judul Penelitian	Peneliti	Metode	Hasil Penelitian
Perancangan Sistem Informasi Pariwisata Berbasis *Android*	Mira Afrina, Ali Ibrahim, Tumpol Simarmata	*Rational Unified process* (RUP)	Dapat membantu dinas kebudayaan dan pariwisata untuk mempromosikan dan menyampaikan informasi wisata dan sarana pendukung yang tersedia.
Perancangan Aplikasi Sistem Navigasi Objek Wisata Berbasis *Android* pada Dinas Pariwisata Kota Makassar	Josseano Amakora Koli Parera, Suci Rahma Dani R	*Data Flow* Diagram (DFD)	Pengguna dimudahkan dalam memperoleh informasi mengenai tempat wisata di Kota Makassar secara lengkap dan detail.
Sistem Informasi ObyekWisata (*Tour Guide*) Secara *Real Time* Menggunakan *GPS* di Bogor Via *Mobile* Berbasis *Android*	Bondan Dwi Hatmoko	*Research and Developm ent* (R&D)	Membantu para wisatawan untuk mendapatkan informasi wisata secara efektif dan efisien
Implementasi Sistem Informasi Geografis Daerah Pariwisata Kabupaten Temanggung Berbasis *Android* dengan *Global Positioning System* (GPS)	Kartika Imam Santoso, Muhamad Nur Rais	*Waterfall*	Memudahkan wisatawan atau pengguna dalam memperoleh informasi obyek wisata.