

BAB II

LANDASAN TERORI

2.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh penulis sebaiknya mencari referensi dari penelitian terdahulu yang pernah dilakukan, pada objek dan judul penelitian yang dilakukan di Dinas Kelautan dan Perikanan Jawa Barat sendiri khususnya pada sistem informasi konservasi belum ada penelitian atau perancangan sistem yang diusulkan sebelumnya sehingga membuat penulis tertarik untuk melakukan penelitian ini. Berikut beberapa Penelitian terdahulu yang memiliki sedikit berhubungan dengan topik penelitian yang dijadikan sebagai bahan referensi :

Table 1.1. Ringkasan Penelitian Terdahulu

1.	Judul Penelitian	PERANCANGAN DAN PEMBANGUNAN SISTEM INFORMASI GEOGRAFIS PARIWISATA DI KOTA BANDUNG BERBASIS WEBSITE [3]
	Nama Peneliti	Lusi Melian dan Hilman Agus
	Objek Penelitian	Dinas Kebudayaan dan Pariwisata Kota Bandung
	Tahun	Maret 2011
	Tujuan Penelitian	Untuk merancang Sistem Informasi Geografis Pariwisata di kota Bandung berbasis website yang bertujuan untuk memudahkan wisatawan untuk mencari informasi lokasi daerah-daerah pariwisata yang tersebar di kota Bandung dengan cepat akurat yang dapat di akses oleh siapa saja, dimana saja, kapan saja tanpa mengenal jarak dan waktu.
	Metodologi Penelitian	Metode pengembangan menggunakan prototipe yang menggunakan pendekatan untuk membuat sebuah

		program dengan cepat dan bertahap sehingga dapat segera di evaluasi oleh pemakai (user). Sedangkan pendekatan yang digunakan dalam merancang sistem ini adalah pendekatan terstruktur.
	Persamaan	Dalam penelitian ini persamaannya yaitu membangun sistem yang menentukan titik lokasi menggunakan sistem informasi geografis
	Perbedaan	Perbedaan penelitian ini pada sistem yang diusulkan yaitu lokasi, lokasi yang dikaji adalah lokasi pariwisata jawa barat bandung, sedangkan penelitian sekarang mencakup lokasi konservasi terumbu karang di jawa barat. Dan fokus penelitian serta yang membedakan juga yaitu usulan program yang berbasis Web.
2.	Judul Penelitian	Sistem Informasi Pendukung Keputusan Pariwisata Berbasis Web [4]
	Nama Peneliti	Marliana B. Winanti, S.Si, M.Si, Dewan Rahadyan, S.Kom
	Objek Penelitian	Kantor Dinas Pariwisata Dan Kebudayaan Provinsi Jawa Barat
	Tahun	September 2014
	Tujuan Penelitian	untuk membuat sebuah sistem informasi yang memiliki kemampuan manajemen data pariwisata yang baik dan mampu memberikan keputusan bagi wisatawan yang sedang mencari tempat wisata dan memberikan keputusan perkembangan dan minat wisata pada wisatawan bagi dinas pariwisata didasarkan pada hasil output keputusan yang berasal para wisatawan yang sedang mencari tempat wisata.

Metodologi Penelitian	Metode pengembangan menggambarkan komponen yang ada, proses dan pemecahan masalah yang terjadi pada saat analisis sistem dan perancangan sistem . Sedangkan pendekatan yang digunakan dalam merancang sistem ini adalah pendekatan terstruktur.
Persamaan	Persamaan yaitu membuat sebuah sistem informasi yang memiliki kemampuan manajemen data yang baik dan mampu memberikan keputusan bagi pihak instansi.
Perbedaan	Perbedaan dari penelitian sebelumnya manajemen yang dilakukan pada manajemen kelola wilayah pariwisata di bandung, sedangkan penelitian sekarang tentang kegiatan konservasi yang mengkaji penyediaan dan pengelolaan unit bibit.

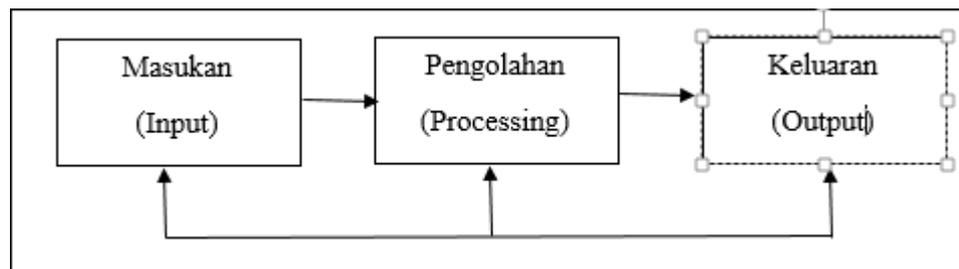
2.2. Definisi Sistem

Sistem adalah sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan.

Robert G. Murdick (1993), mendefinisikan sistem sebagai seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan suatu tujuan bersama. [5, p.4]

Pendekatan sistem yang lebih menekankan pada prosedur didefinisikan bahwa sistem yaitu suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Gerald J. 1991). [6, p.3]

menurut Gapsert sistem memiliki ciri pokok yaitu sistem ini dioperasikan didalam suatu lingkungan, sistem ini terdiri dari unsur-unsur, sistem ditandai dengan adanya keterhubungan, dan sistem ini memiliki satu fungsi dan atau tujuan utama [6,p.3]. Untuk menggambarkan sistem yang dimaksud yaitu model sebuah sistem ditunjukkan dalam gambar berikut:



Gambar 2. 1. Model Sistem [6, p.4]

2.3. Karakteristik Sistem

Agar sistem dikatakan sebagai sistem yang baik maka sistem tersebut harus memiliki karakteristik sebagai berikut : [6, p.4-5]

1. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan Sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem

Lingkungan luar sistem merupakan bagian luar batas sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga agar tidak mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem

Penghubung sistem merupakan media penghubung antara subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan Sistem (*input*)

Masukan merupakan energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). Maintenance input merupakan energi yang dimasukkan agar sistem dapat beroperasi. Signal Input merupakan energi yang diproses untuk didapatkan keluaran

6. Keluaran Sistem (*output*)

Keluaran sistem merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

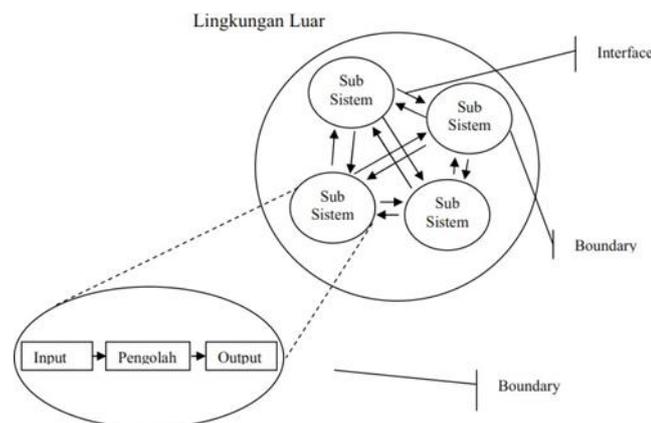
7. Pengolahan Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Berikut merupakan gambaran karakteristik sistem:



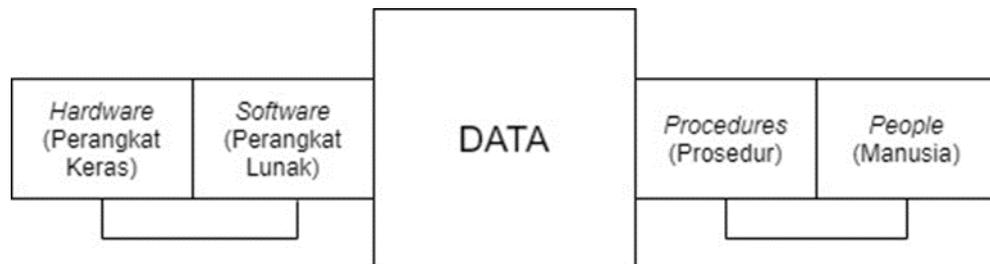
Gambar 2. 2. Karakteristik Sistem Informasi [6, p.4-5]

2.4. Definisi Sistem Informasi

Sistem Informasi dapat didefinisikan sebagai berikut : [6, p.13]

- Suatu sistem yang di buat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mendapatkan suatu tujuan yaitu menyajikan informasi.
- Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan organisasi.
- Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manejerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Adapun komponen dari sistem informasi dapat dilihat dari gambar berikut :



Gambar 2. 3. Lima komponen Sistem Informasi
(Sumber: Buku: Analisis dan Desain SISTEM INFORMASI [6, p.15])

Kelima komponen tersebut dapat didefinisikan sebagai berikut.

- a. *Hardware* dan *Software* yang berfungsi sebagai mesin.
- b. *People* dan *Procedures* yang merupakan manusia dan tat cara menggunakan mesin.
- c. Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.

2.5. Komponen Sistem Informasi

Sistem informasi tersusun dari beberapa komponen atau elemen. Komponen sistem informasi tersebut adalah sebagai berikut [6, p.15]:

- a. Perangkat keras atau yang disebut *hardware* , perangkat ini mencakup piranti-piranti fisik misalnya komputer, printer, keyboard, dan lain-lain.
- b. Perangkat Lunak atau yang disebut *software* (program) merupakan sekumpulan instruksi-instruksi yang memungkinkan perangkat keras dalam melakukan proses.
- c. Prosedur, merupakan sekumpulan aturan yang digunakan dalam mewujudkan pemrosesan data dan menghasilkan sebuah keluaran yang dikehendaki.

- d. Orang, merupakan semua pihak yang memiliki tanggung jawab di dalam pengembangan sebuah sistem, pemrosesan, dan penggunaan keluaran sistem informasi.
- e. Basis Data atau yang disebut database merupakan tabel, hubungan, dan lain-lain yang berhubungan dengan penyimpanan data.
- f. Jaringan komputer dan komunikasi data merupakan sistem penghubung yang memungkinkan sumber dipakai secara bersama atau diakses oleh sejumlah pemakai. [6, p.16]

2.6. Basis Data

Basis Data (*Database*) adalah pengorganisasian sekumpulan data yang saling berkaitan satu sama lain sehingga mempermudah dalam aktivitas untuk memperoleh sebuah informasi. Basis data dibuat bertujuan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. [7]

Basis data merupakan suatu sistem yang memiliki karakteristik seperti berikut:

1. Kumpulan-kumpulan interaksi data yang disimpan bersama-sama dan tidak saling mengganggu satu sama yang lainnya atau membentuk duplikat data.
2. Kumpulan-kumpulan data tersebut dapat digunakan oleh sebuah program secara optimal.
3. Penambahan data baru, modifikasi dan pengambilan kembali dari data dapat dilakukan dengan mudah dan terorganisasi.

Dalam sebuah arsitektur database terdapat tiga tingkatan yang ketiganya tersebut saling mendukung. Berikut ini adalah penjelasannya [7]:

1. Level Internal yaitu level dimana berhubungan langsung dengan basis data dan menjelaskan bagaimana data disimpan dalam basis data. Level ini berurusan dengan hal yang antarlain sebagai berikut: alokasi ruang penyimpanan data dan indeks, deskripsi rekaman dalam penyimpanan, kompresi data dan teknik enkripsi data.
2. Level Eksternal merupakan level yang berhubungan langsung dengan pengguna. Pada level ini, pengguna cukup mengenal struktur data yang sederhana dalam basis data supaya dapat mengakses basis data.
3. Level konseptual menjelaskan data apa saja yang tersimpan dalam basis data dan juga menjabarkan hubungan- hubunganantar data. Level ini biasa dipakai oleh administrator data.

Operasi yang dilakukan pada sebuah database berdasarkan atas tabel-tabel dan hubungannya. Beberapa istilah dalam model relasional antara lain *query* ,*table*, *record*, *indeks*, *field* yang akan dijelaskan lebih rinci sebagai berikut [7]:

1. Entity atau tabel dalam model relasional digunakan untuk mendukung antar muka komunikasi antara pemakai dengan professional komputer.
2. Record atau baris atau dalam istilah model relasional yang formal disebut tuple adalah kumpulan data yang terdiri dari satu atau lebih.
3. Field atau kolom, dalam model relasional formal disebut attribute adalah sekumpulan data-data yang mempunyai atau menyimpan sebuah fakta yang sama atau sejenis untuk setiap baris yang terdapat pada tabel.
4. Indeks adalah salah satu jenis tipe dari suatu tabel tertentu yang berisi sebuah nilai-nilai field kunci.

5. Query adalah sekumpulan perintah *Structure Query Language* (SQL) yang dibentuk untuk memanggil kelompok record tertentu dari satu tabel atau tabel lainnya untuk melakukan operasi pada tabel.

2.7. Teori Dasar Cakupan Penelitian

Ada pun penjelasan teori yang berkaitan dengan sistem yang dirancang :

2.7.1. Terumbu Karang

Terumbu karang adalah sekumpulan hewan karang yang bersimbiosis dengan sejenis tumbuhan alga yang disebut dengan zooxanthellae. Terumbu karang merupakan salah satu komponen utama sumber daya pesisir dan laut, disamping hutan mangrove dan padang lamun. Terumbu karang dan segala kehidupan yang ada didalamnya merupakan salah satu kekayaan alam yang dimiliki bangsa Indonesia yang tak ternilai harganya. Diperkirakan luas terumbu karang yang terdapat di perairan Indonesia adalah lebih dari 60.000 km², yang tersebar luas dari perairan Kawasan Barat Indonesia sampai Kawasan Timur Indonesia.[2]

2.7.2. Konservasi

Konservasi adalah pelestarian atau perlindungan. Secara harfiah, konservasi berasal dari bahasa Inggris, conservation yang artinya pelestarian atau perlindungan [14].

Adapun menurut ilmu biologi, konservasi adalah:

1. Efisiensi penggunaan, produksi, transmisi, atau distribusi energi yang berakibat pada turunnya konsumsi energi dengan tetap menghasilkan manfaat yang sama;
2. Pelestarian dan pengelolaan lingkungan dan sumber daya alam secara

bijaksana;

3. Pelestarian dan perlindungan jangka panjang terhadap lingkungan, memastikan bahwa habitat alami suatu area dapat dipertahankan, sementara keanekaragaman genetik dari suatu spesies dapat tetap ada dengan mempertahankan lingkungan alaminya.

2.7.3. Transplantasi

Transplantasi berasal dari bahasa Inggris yaitu *To Transplant*, yang berarti *To Move From One Place to Another*, artinya berpindah dari satu tempat ke tempat lain. [17] Menurut kamus besar bahasa Indonesia kata transplantasi mempunyai arti pemindahan jaringan tubuh dari satu tempat ke tempat lain dan atau pencangkokan.

2.7.4. Sistem Informasi Geografis (GIS)

Sistem Informasi Geografis (SIG) atau *Geographic Information System (GIS)* merupakan suatu sistem informasi yang berbasis komputer, dirancang untuk bekerja dengan menggunakan data yang memiliki informasi spasial (bereferensi keruangan) [3, p.2]. Sistem ini menangkap, mengecek, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang secara spasial mereferensikan kepada kondisi bumi. CGIS merupakan sistem pertama di dunia dan hasil dari perbaikan aplikasi pemetaan yang memiliki kemampuan tumpang susun (*overlay*), penghitungan, pendijitalan/pemindaian (*digitizing/scanning*), mendukung sistem koordinat nasional yang membentang di atas benua Amerika, memasukkan garis sebagai arc yang memiliki topologi dan menyimpan atribut dan informasi lokasional pada berkas terpisah. Pengembangnya, seorang geografer bernama Roger Tomlinson kemudian disebut "Bapak SIG". [13, p.16-19]

2.7.5. Penyediaan

Merupakan proses kegiatan untuk pemenuhan kebutuhan dan pasokan barang atau jasa di bawah kontrak atau pembelian langsung untuk memenuhi kebutuhan bisnis. Penyediaan dapat mempengaruhi keseluruhan proses arus barang karena merupakan bagian penting dalam proses tersebut. [4]

2.7.6. Pengelolaan

Nugroho (2003:119) mengemukakan bahwa Pengelolaan merupakan istilah yang dipakai dalam ilmu manajemen. Secara etimologi istilah pengelolaan berasal dari kata kelola (*to manage*) dan biasanya merujuk pada proses mengurus atau menangani sesuatu untuk mencapai tujuan tertentu. Jadi pengelolaan merupakan ilmu manajemen yang berhubungan dengan proses mengurus dan menangani sesuatu untuk mewujudkan tujuan tertentu yang ingin dicapai. [16]

2.8. *Unified Modeling Language* (UML)

Menurut Nugroho, *Unified Modeling Language* (UML) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berorientasi objek. Pemodelan sesungguhnya digunakan untuk menyederhanakan permasalahan-permasalahan yang rumit sehingga dapat lebih mudah dipahami dan dimengerti. Sedangkan menurut Martin Fowler, *Unified Modeling Language* keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.[7]

UML Diagram dapat disimpulkan sebagai sebuah rancangan untuk membangun sebuah perangkat lunak yang didalamnya berisi alur dari proses yang sudah ada untuk kemudian diterjemahkan kedalam bahasa pemrograman.[1]

1. Use Case Diagram

Use case adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan aktor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case juga digunakan untuk membentuk perilaku (*behaviour*) sistem yang akan dibuat. Sebuah use case menggambarkan sebuah interaksi antara pengguna (aktor) dengan sistem yang ada [1, p.14-15] .

Tujuan Use Case adalah Memetakan kebutuhan sistem ,Merepresentasikan interaksi pengguna terhadap sistem ,Untuk mengetahui kebutuhan diluar sistem . Adapun Komponen use case diagram adalah sebagai berikut:

a. Aktor

Aktor merupakan hal terpenting dari use case diagram, akan tetapi aktor bukanlah bagian dari use case diagram, namun untuk dapat terciptanya suatu use case diagram diperlukan beberapa aktor. Itulah pengertian Actor pada komponen use case diagram secara singkat .

b. *Use Case*

Use case merupakan sebuah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem tersebut paham secara jelas mengenai kegunaan sistem atau aplikasi yang akan dibangun. Itulah pengertian use case pada komponen use case diagram secara singkat.

c. Relasi

Use case diagram memiliki relasi sebagai berikut:

- a. *Association* : Menghubungkan link antar element.
- b. *Generalization* : Sebuah elemen dapat merupakan spesialis dari elemen lainnya bisa disebut dengan pewarisan sifat (*inheritance*).
- c. *Dependency* : Sebuah elemen bergantung dalam beberapa cara ke element lain.
- d. *Aggregation* : Bentuk association dimana sebuah elemen berisi elemen lainnya.

Tipe Relasi/ Stereotype yang mungkin terjadi pada use case diagram :

- a. <<*include*>>, kelakuan yang harus terpenuhi agar sebuah event dapat terjadi.
- b. <<*extends*>>, kelakuan yang hanya berjalan pada kondisi tertentu.
- c. <<*communicates*>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah communicates association.

2. Skenario Use Case

Skenario Use Case digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut.[1, p.14]

3. Activity Diagram

Activity Diagram adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah

bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem [1, p.15].

Komponen-komponen yang ada pada activity diagram adalah sebagai berikut:

- a. *Nodes* (initial dan final) adalah simbol untuk memulai (initial) dan mengakhiri (final) suatu activity diagram.
- b. *Activity* (aktivitas) adalah proses komputasi yang bisa berupa kata kerja atau ekspresi dan bersifat atomik atau tidak dapat didekomposisi
- c. *Flow* adalah awal dari proses yang paralel dan mampu menggambarkan aktivitas yang mungkin terjadi secara concurrent.
- d. *Join* adalah akhir dari suatu proses paralel.
- e. *Decision* adalah pilihan untuk mengambil keputusan.
- f. *Partition* digunakan untuk menjelaskan siapa yang melakukan aktivitas dalam *activity diagram*. Untuk melakukan partisi dapat dilakukan dengan menggunakan Swim Lane.
- g. *Signal* adalah tanda untuk memulai sebuah aktivitas.

4. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan kelakuan objek pada usecase 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 use case beserta metode-metode yang dimiliki kelas yang diinisialisasi dalam objek itu. Jumlah dari diagram sekuen yang harus digambar sebanyak usecase yang memiliki proses sendiri atau yang penting semua usecase yang telah didefinisikan interaksinya pesan sudah

dicakup pada diagram sekuen sehingga semakin banyak usecase yang ada maka diagram sekuen yang dibuat semakin banyak.

Komponen Sequence Diagram terdiri dari [1, p.17-18] :

- a. *Aktor* : Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
- b. *Boundary* : Menggambarkan interaksi antara satu atau lebih aktor dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.
- c. *Control* : Menggambarkan, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
- d. *Entity* : Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).
- e. *Object Message* : Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.
- f. *Message to Self* : Menggambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
- g. *Return Message* : Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi
- h. *Lifeline* : Eksekusi obyek selama sequence (*message* dikirim atau diterima dan aktifasinya).

5. *Class Diagram*

Class Diagram adalah diagram yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class* diagram menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML. [1, p.18-19]

1) *Class* :

Blok – blok pembangun pada pemrograman berorientasi objek. Sebuah class digambarkan sebagai sebuah kotak yang terdiri dari 3 bagian. Bagian atas adalah bagian nama dari class, Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan method-method dari sebuah class.

2) *Association* :

Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe relationship dan juga menampilkan hukum-hukum multiplisitas pada sebuah relationship (Contoh : one-to-one, one-to-many, many-to-many)

3) *Composition* :

Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi / solid.

4) *Dependency* :

Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain.

5) *Aggregation* :

Mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi.

6. *Deployment Diagram*

Deployment diagram merupakan gambaran proses-proses berbeda pada suatu sistem yang berjalan dan bagaimana relasi di dalamnya. Hal inilah yang mempermudah user dalam pemakaian sistem yang telah dibuat dan diagram tersebut merupakan diagram yang statis. Misalnya untuk mendeskripsikan sebuah situs web, deployment diagram menunjukkan komponen perangkat keras ("*node*") apa yang digunakan (misalnya, *web server*, *server aplikasi*, dan *database server*), komponen perangkat lunak ("*artefak*") apa yang berjalan pada setiap node.(misalnya, *aplikasi web*, *database*), dan bagaimana bagian-bagian yang berbeda terhubung (misalnya JDBC, REST, RMI). [1, p.21]

Node digambarkan sebagai kotak, dan artefak yang dialokasikan ke setiap node digambarkan sebagai persegi panjang di dalam kotak. Node mungkin memiliki subnodes, yang digambarkan sebagai kotak nested. Sebuah node tunggal secara konseptual dapat mewakili banyak node fisik, seperti sekelompok database server. [1,p.22-23]

Komponen Deployment Diagram :

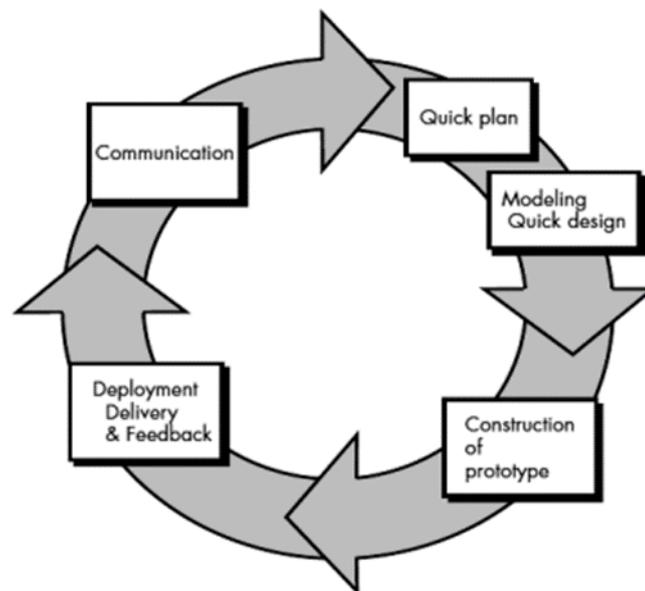
- a. *Component*: Pada deployment diagram, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.
- b. *Node* : Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
- c. *Association* : Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara komponen - komponen hardware.
- d. *Dependency* : Merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.
- e. *Generalization* : Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan generalization, class yang lebih spesifik (subclass) akan menurunkan atribut dan operasi dari class yang lebih umum (superclass).
- f. *Note* : Menunjukkan catatan untuk komentar dari suatu pesan antar elemen, sehingga bisa langsung terlampir dalam model.
- g. *Class* : Interface merupakan kumpulan operasi tanpa implementasi dari suatu class. Implementasi operasi dalam interface dijabarkan oleh operasi didalam class.

2.9. Metode Pengembangan Perangkat Lunak (*Prototype*)

Menurut Roger S. Pressman, dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode prototype. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal

ini pengguna dari perangkat yang dikembangkan adalah peserta didik. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar. [9, p.10]

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik. Gambar model *prototype* menurut Roger S. Pressman [9]:



Gambar 2. 4. Paradigma Pembuatan *Prototype*

(Sumber : Roger S. Pressman, *Software Engineering : A Practitioner's Approach*, 7th ed. 2010)

1. *Communication* (Komunikasi)

Pada tahap ini dilakukan komunikasi dengan stakeholder pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu

bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. *Quick Plan* (Perencanaan Cepat)

Pada tahap ini, dilakukan perencanaan cepat untuk menentukan kebutuhan apa saja yang akan dilibatkan dalam tahap selanjutnya dan dalam tahapan ini juga tim akan melakukan pemodelan secara kasar mengenai sistem yang akan dibangun.

3. *Modeling Quick Design* (Pemodelan Rancangan Cepat)

Pada tahap ini, dilakukan pemodelan perencanaan dari tahap sebelumnya dengan menggunakan alat bantu seperti DFD, ERD dan Flowchart untuk pendekatan terstruktur. Sedangkan untuk pendekatan objek, maka dapat menggunakan UML.

4. *Construction of Prototype* (Pembuatan Prototype)

Pada tahap ini, rancangan yang sebelumnya sudah dibuat akan mulai dikonversikan menjadi suatu program yang akan dibangun.

5. *Deployment Delivery & Feedback* (Penyerahan dan Memberikan Umpan Balik Terhadap Pengembangan Prototipe)

Pada tahap ini, prototype yang sudah dibangun akan diserahkan kepada pelanggan untuk dinilai dan diberikan umpan balik mengenai kinerja prototipe tersebut. Apabila dalam prototipe tersebut masih ada kekurangan, maka akan kembali ke langkah ke 1 untuk memperbaiki kekurangan tersebut.

2.10. Perangkat Lunak Pendukung

Penelitian ini menggunakan beberapa perangkat lunak untuk mengembangkan Sistem Informasi Konservasi Terumbu Karang Pada Dinas Perikanan dan Kelautan Provinsi Jawa Barat. Berikut adalah urutan perangkat lunak

atau *software* yang digunakan dalam membangun dan mengembangkan sistem tersebut [11]:

2.10.1. Perl Hypertext Preprocessor (PHP)

Menurut Arief, PHP (*Perl Hypertext Preprocessor*) adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis.

[11] Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis. Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. PHP dapat berjalan pada web server yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat open source.
3. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembangannya.
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. PHP dapat berjalan pada 3 operating sistem, yaitu: Linux, unix, dan windows, dan juga dapat dijalankan secara runtime pada suatu console.

2.10.2. HTML

Hyper Text Markup Language atau yang biasa disingkat HTML merupakan kumpulan bahasa pemrograman yang menyusun halaman di WWW (*World Wide Web*). HTML ini adalah bahasa perintah untuk meminta browser menjalankan,

menampilkan dan menangani jenis data berupa teks, gambar, grafik, suara dan elemen lainnya untuk suatu dokumen web. HTML tidak hanya menampilkan informasi yang statis tetapi juga dinamis. [10]

2.10.3. Bootstrap

Bootstrap adalah sebuah *framework CSS* yang menyediakan kumpulan komponen-komponen antarmuka dasar pada web yang telah dirancang sedemikian rupa untuk digunakan bersama-sama. Selain komponen antarmuka, *Bootstrap* juga menyediakan sarana untuk membangun layout halaman dengan mudah dan rapi, serta modifikasi pada tampilan dasar HTML untuk membuat seluruh halaman web yang dikembangkan senada dengan komponen-komponen lainnya. Bootstrap dibuat untuk memberikan sekumpulan perangkat yang dapat digunakan untuk membangun website sederhana dengan mudah. [10]

2.10.4. Cascading Style Sheet (CSS)

CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa markup seperti HTML, XHTML, XML, SVG (*Scalable Vector Graphics*) dan Mozilla XUL (*XML User Interface Language*). [10]

2.10.5. Database Management System (DBMS)

DBMS adalah singkatan dari “*Database Management System*” yaitu sistem pengorganisasian dan sistem pengolahan database pada komputer. DBMS atau

database management system ini merupakan perangkat lunak (*software*) yang dipakai untuk membangun basis data yang berbasis komputerisasi.

DBMS (*Database Management system*) ini juga dapat membantu dalam memelihara serta pengolahan data dalam jumlah yang besar, dengan menggunakan DBMS bertujuan agar tidak dapat menimbulkan kekacauan dan dapat dipakai oleh user sesuai dengan kebutuhan.

DBMS ialah perantara untuk user dengan basis data, untuk dapat berinteraksi dengan DBMS dapat memakai bahasa basis data yang sudah ditentukan oleh perusahaan DBMS. Bahasa basis data umumnya terdiri dari berbagai macam instruksi yang diformulasikan sehingga instruksi tersebut dapat di proses oleh DBMS.

Perintah atau instruksi tersebut umumnya ditentukan oleh user, adapun bahasa yang digunakan dibagi kedalam 2 (dua) macam diantaranya sebagaimana di bawah ini:

1. DDL (*Data Definition Language*)

Yang pertama adalah bahasa DDL atau kepanjangannya Data Definition Language, yaitu dipakai untuk menggambarkan desain dari basis data secara menyeluruh. DDL (*Data Definition Language*) dapat dipakai untuk membuat tabel baru, memuat indeks, maupun mengubah tabel. Hasil dari kompilasi DDL akan disimpan di kamus data. Itulah definisi dari DDL.

2. DML (*Data Manipulation Language*)

DML atau kepanjangannya Data Manipulation Language, yaitu dipakai untuk memanipulasi dan pengambilan data pada suatu basis data, misalnya seperti

penambahan data yang baru ke dalam suatu basis data, menghapus data pada suatu basis data dan mengubah data pada suatu basis data. Itulah definisi dari DML.[11]

2.10.6. MySQL

MySQL adalah perangkat lunak yang terhubung dengan RDMS (*Relational Database Management System*) untuk mengatur Database. MySQL merupakan turunan dari database SQL untuk pengoperasian database dalam pemilihan / seleksi dan inputan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah dan otomatis. MySQL mendukung berbagai platform dan mudah terintegrasi serta kecepatan akses terbilang dapat diandalkan. [11]

2.10.7. Visual Studio Code

Microsoft Visual Studio Code adalah one-stop shop yang memungkinkan kita fokus pada proses pengembangan dan melupakan tools baru. Lihat nih beberapa fitur Visual Studio Code [10]:

1. *Cross platform* – tersedia di macOS, Linux dan Windows artinya Anda dapat bekerja pada sistem operasi manapun tanpa khawatir belajar coding tools yang sama untuk sistem yang berbeda-beda.
2. *Lightweight* – tak perlu menunggu lama untuk memulai. Anda mengontrol sepenuhnya bahasa, tema, *debugger*, *commands* dan lain-lainnya sesuai keinginan. Ini dapat dilakukan melalui extentions untuk bahasa populer seperti python, node.js, java dan lain-lainnya di *Visual Studio Code Marketplace*.

3. *Powerful editor* – memfungsikan fitur untuk source code editing yang sangat produktif, seperti membuat code snippets, IntelliSense, auto correct, dan formatting.
4. *Code Debugging* – salah satu fitur terkeren yang ditawarkan *Visual Studio Code* adalah membantu Anda melakukan debug pada kode dengan cara mengawasi kode, variabel, call stack dan expression yang mana saja.
5. *Source control* – *Visual Studio Code* memiliki integrated source control termasuk Git support in-the-box dan penyedia source code control lainnya di pasaran. Ini meningkatkan siklus rilis proyek Anda secara signifikan.
6. *Integrated terminal* – Tiada lagi *multiple windows* dan *alt-tabs*. Anda dapat melakukan command-line task sekejap dan membuat banyak terminal di dalam editor. [10]

2.10.8. XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL database, *Apache HTTP Server*, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), *Apache*, MySQL, PHP dan *Perl*. Program ini tersedia di bawah GNU *General Public License* dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan XAMPP dapat mendownload langsung dari situs resminya. [15]

Bagian-bagian XAMPP

- a. *htdocs* adalah folder di mana Anda meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
- b. *phpMyAdmin* adalah bagian untuk mengelola database MySQL yang dikomputer. Untuk membukanya, membuka browser dan ketik alamat `http://localhost/phpMyAdmin`, halaman phpMyAdmin akan muncul.
- c. *Control Panel* yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti *stop service* (berhenti), atau *start* (mulai).

2.11. Jaringan Komputer

Interconnection Networking atau biasa disebut Internet adalah jaringan komputer paling luas yang mencakup seluruh bumi. LAN, MAN, dan WAN semua ini dihubungkan oleh Internet didalamnya. Internet saat ini penggunaannya sangat diperlukan dan menjadi bagian pokok penting dalam umat manusia.

Hosting atau sering juga disebut dengan *WebHosting* merupakan suatu infrastruktur yang digunakan untuk menyimpan data dari website atau aplikasi sehingga bisa diakses menggunakan internet. [12]

2.12. Bahasa Pemrograman *JavaScript*

JavaScript merupakan sebuah bahasa pemrograman web yang bersifat *Client Side Programming Language* yang pemrosesannya dilakukan disisi *client*. Aplikasi client tersebut merujuk kepada aplikasi web browser seperti *Mozilla Firefox* dan *Google Chrome*. [10]

2.13. *CodeIgniter* (CI)

CodeIgniter adalah sebuah *framework* PHP yang dapat mempercepat pengembangan atau pembuatan sebuah website. Dengan dilengkapi *library* yang banyak dan helper yang berguna didalamnya sehingga dapat mempermudah proses development. Banyak *framework* PHP yang telah ada saat ini. Salah satunya adalah *framework codeIgniter*. *CodeIgniter* termasuk dalam aplikasi *open source* (gratis) berupa framework dengan model konsep MVC (*Model, View, Controller*) yang di memiliki fungsi untuk membuat sebuah halaman web dinamis yang berbasis PHP. Konsep MVC (*Model, View, Controller*) merupakan salah suatu konsep yang populer dalam pembangunan sebuah aplikasi web, konsep ini berawal dari bahasa pemrograman Small Talk. MVC memisahkan pengembangan aplikasi menjadi beberapa komponen utama untuk membangun sebuah aplikasi web seperti manipulasi data (*model*), *user interface* (*view*), dan bagian yang menjadi kontrol aplikasi (*Controller*). Berikut ini penjelasan lebih rinci dari ketiga bagian tersebut [3]:

1. *View*, bagian ini memiliki fungsi untuk menangani *presentation logic*. Pada sebuah aplikasi web bagian *view* ini biasanya berupa file template HTML, bagian ini diatur oleh sebuah *controller*. *View* berguna untuk menerima dan merepresentasikan sebuah data kepada *user*.
2. *Model*, bagian ini memiliki hubungan langsung dengan database untuk memanipulasi sebuah data (*creat, update, delete, read, search*), menangani validasi yang dikirim dari bagian *controller*, bagian ini tidak dapat berhubungan langsung dengan bagian *view*.

3. *Controller*, bagian ini merupakan bagian yang memiliki fungsi untuk mengatur hubungan antara bagian *view* dan bagian model, menerima permintaan data dari user, lalu bagian ini kemudian menentukan apa yang akan diproses pada aplikasi.[3]