

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Informasi Geografis**

Amnah, (2016 : 01) Menyatakan bahwa Sistem informasi geografis pertama dibangun pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis. Sistem informasi geografis merupakan sistem informasi berbasis komputer yang menggabungkan antara unsur peta (geografis) dan informasinya tentang peta tersebut (data atribut).

Sistem Informasi geografis merupakan suatu sistem tentang informasi geografis yang telah sangat berkembang. SIG memiliki kemampuan yang baik dalam memvisualisasikan data spasial berikut atribut-atributnya, memodifikasi bentuk, warna, ukuran, dan symbol. (Yulmaini, 2014 : 02)

Sistem informasi geografis adalah system informasi khusus mengelola data yang memiliki informasi special. Arti yang lebih sempit SIG adalah system komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola, dan menampilkan informasi geografis. (Yulir, 2013 : 08)

#### **2.2 Rancang Bangun**

Pressman (2005 : 334) Menyatakan bahwa mendefinisikan rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik (Ladjamudin 2002 : 156).

Garis besar yang dapat disimpulkan dari definisi diatas perancangan adalah penggambaran, perencanaan dan pembuatan sketsa yang kemudian diterjemahkan ke dalam sebuah konsep rancangan sebagai pemecahan

masalah berbasis komputer yang telah dipilih selama tahap analisis. Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian.

Kesimpulannya pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

### **2.3 Pencarian**

Munir (2011 : 05) Menyatakan bahwa Pencarian (*Searching*) merupakan proses yang fundamental dalam pengelolaan data. Proses pencarian nilai (data) tertentu di dalam sekumpulan data yang bertipe sama

### **2.4 Lokasi**

Yulir (2013 : 12) Menyatakan bahwa dalam bukunya yang berjudul “Geografi 1 untuk SMA/MA Kelas X” menuliskan definisi Lokasi (*Location*) adalah konsep utama yang digunakan untuk mengetahui fenomena geosfer. Dalam Geografi lokasi mempunyai dua makna yaitu lokasi *absolut* dan lokasi relatif.

- 1) Lokasi *absolut* adalah lokasi di permukaan yang ditentukan oleh sistem koordinat garis lintang dan garis bujur, disebut juga lokasi mutlak.
- 2) Lokasi relatif adalah lokasi sesuatu objek yang nilainya ditentukan oleh objek-objek lain di luarnya.

### **2.5 Geografis**

Yulir (2013 : 02) Menyatakan bahwa Perkataan geografi berasal dari bahasa Yunani: *geo* berarti bumi dan *graphein* berarti tulisan, gambaran, atau tulisan tentang bumi. Jadi, secara harfiah, geografi berarti tulisan tentang bumi. Oleh karena itu, geografi sering juga disebut ilmu bumi. Dalam arti luas geografi adalah ilmu pengetahuan yang mempelajari uraian tentang

bumi, dengan titik fokus mengenai ruang dan waktu. Dalam geografi kita akan mempelajari hal – hal yang tampak dimuka bumi, meliputi bentuk fisik beserta gejala – gejalanya termasuk bentuk kehidupan serta factor lain yang mempengaruhinya.

Berikut ini beberapa batasan atau definisi dari beberapa pakar Geografi adalah

- 1) Geografi adalah studi tentang pengaruh lingkungan alam pada aktivitas manusia.
- 2) Gografi adalah suatu penyajian dengan peta bagi sebagian permukaan bumi yang menunjukkan kenampakan secara umum.
- 3) Geografi adalah ilmu pengetahuan yang menceritakan, menerangkan sifat – sifat bumi, menganalisa gejala – gejala alam dan penduduk, serta mempelajari corak khas mengenai kehidupan dan berusaha mencari fungsi dari unsur – unsur bumi dalam ruang dan waktu.
- 4) Gografi adalah ilmu yang berusaha menemukan dan memahami persamaan – persamaan dan perbedaan – perbedaan yang ada dalam ruang muka bumi
- 5) Geografi adalah ilmu yang mempelajari persamaan dan perbedaan fenomena geosfer dengan sudut pandang kelingkungan dan kewilayahan dalam konteks keruangan.

### **2.5.1 Ruang Lingkup Geografi**

Yulir (2013 : 07) Menyatakan bahwa Ruang lingkup geografi konsepnya sangat luas dan mencakup beberapa aspek yang dimana satu sama lain saling berkaitan dan sulit untuk dipisahkan, aspek tersebut meliputi hal – hal berikut :

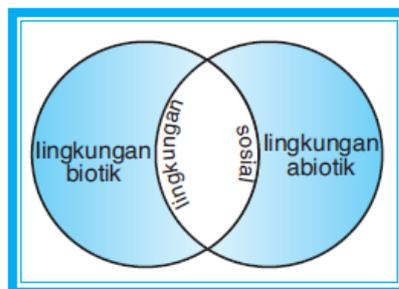
- 1) Aspek fisik (physical geopgraphy atau geografi fisik) meliputi gejala – gejala alam di permukaan bumi yaitu atmosfer, litosfer, hidrosfer, dan biosfer.

- 2) Aspek social ( human geografi atau geografi manusia ) meliputi segala aktifitas kehidupan manusia dan interaksinya dengan lingkungan baik lingkungan social, ekonomi maupun budaya.
- 3) Aspek regional ( geografi regional ) mempelajari topik yang mencakup wilayah tertentu. Geografi regional merupakan bahasan yang menyeluruh baik aspek fisik maupun aspek sosial.

Berdasarkan ketiga aspek di atas, maka ruang lingkup geografi tidak dapat dilepaskan dari aspek manusia dan lingkungannya. Menurut teori lingkungan hidup, lingkungan permukaan bumi dapat dikelompokkan menjadi tiga macam yaitu sebagai berikut :

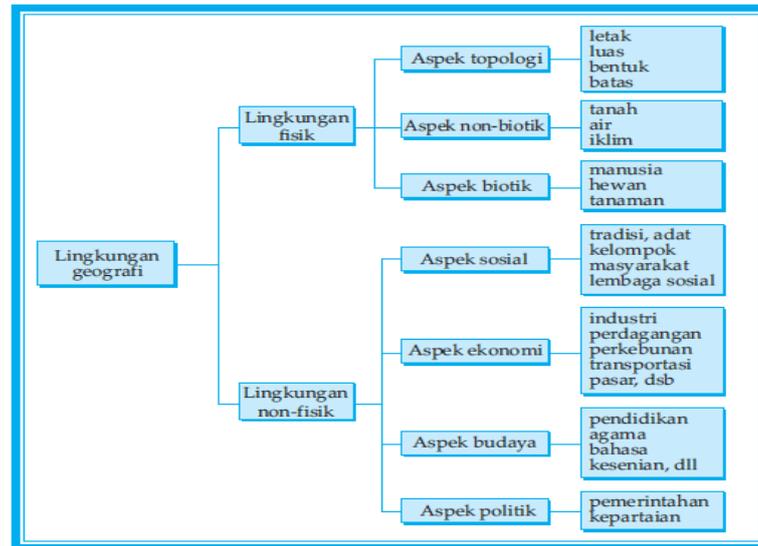
- 1) Lingkungan fisik (*physical environment*) atau abiotik adalah segala sesuatu di sekitar manusia yang berupa makhluk tak hidup, misalnya tanah, udara, air, dan sinar matahari.
- 2) Lingkungan biologis (*biological environment*) atau biotik adalah segala sesuatu di sekitar manusia yang berupa makhluk hidup, seperti binatang, tumbuh-tumbuhan termasuk di dalamnya adalah manusia.
- 3) Lingkungan sosial (*social environment*) adalah segala sesuatu di sekitar manusia yang berwujud tindakan atau aktivitas manusia baik dalam hubungannya dengan lingkungan alam maupun hubungan antar manusia.

Ketiga lingkungan itu dapat diilustrasikan seperti gambar 2.1 berikut.



Gambar 2.1 Lingkungan pada permukaan bumi

Berkaitan dengan teori lingkungan, William Kirk telah menyusun struktur lingkungan geografi yang digolongkan menjadi lingkungan fisik dan lingkungan nonfisik. Untuk lebih jelasnya perhatikan Gambar 2.2 berikut.



Gambar 2.2 Bagan lingkungan pada permukaan bumi.

## 2.6 Layanan darurat

W, E.,Wishnu (2012 : 13) Menyatakan bahwa Layanan darurat atau *emergencyservices* merupakan organisasi yang menjamin keselamatan publik dan kesehatan dengan mengatasi berbagai keadaan darurat, beberapa lembaga ini ada untuk mengatasi beberapa jenis keadaan darurat, seperti Polisi, Rumah Sakit, Puskesmas dan Pemadam Pebakaran. Keadaan darurat merupakan keadaan yang lain dari keadaan normal yang mempunyai kecenderungan atau potensi tingkat yang membahayakan baik bagi keselamatan manusia, harta benda maupun lingkungan jenis-jenis keadaan darurat seperti kebakaran, ledakan, kecelakaan, perampokan dan kerusakan mesin.

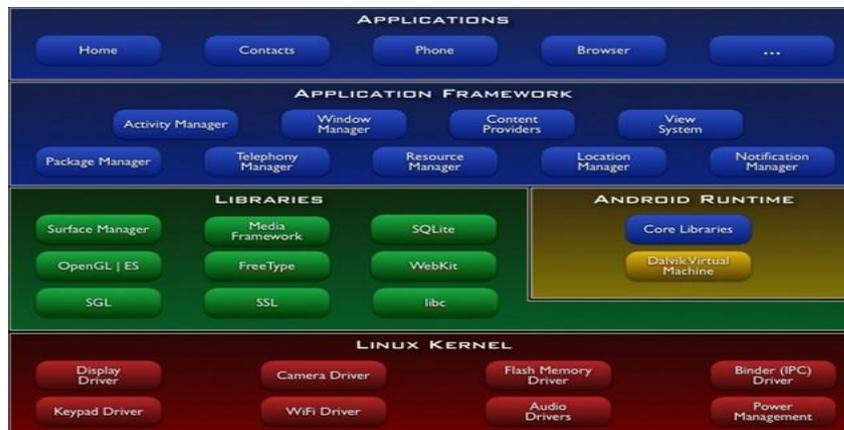
## 2.7 Android

Juhara, P.Z (2016: 01) Menyatakan bahwa dalam buku yang berjudul “Panduan Lengkap Pemrograman Android” Android adalah sistem operasi

berbasis Linux yang dimodifikasi untuk perangkat bergerak ( mobile devices ) yang terdiri dari sistem operasi, middleware, dan aplikasi – aplikasi utama. Awalnya android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi Open Handset Al-liance tahun 2007. Selain Google, beberapa nama – nama besar juga ikut seta dalam Open Handset Alliance, antara lain Motorola, Samsung, LG, Sony Ericsson, T- Mobile, Vodafone, Toshiba, dan Intel.

### 2.7.1 Arsitektur Android

Juhara, P.Z (2016 : 01) Menyatakan bahwa Secara umum, arsitektur Android terdiri dari lapisan software, yaitu lapisan aplikasi, framework aplikasi, pustaka, android runtime, dan kernel linux. Gambar 2.3 merupakan Penjelasan arsitektur dari Android :



Gambar 2.3 Arsitektur Android

- 1) *Framework Aplikasi* Dibawah aplikasi terdapat sejumlah software pendukung, meliputi layanan pengelola activity, view, sumber daya, pemberitahuan ( notifikasi ) dan lain lain.
- 2) Pustaka Android menyediakan sejumlah pustaka dasar seperti pustaka grafis 2D dan 3D, multimedia playback, browser engine, percetakan

font, database dan lain – lain. Aplikasi yang memanfaatkan pustaka fungsi ini melalui lisan framework aplikasi.

- 3) *AndroidRunTime* menyediakan pustaka inti bagi pemrograman Java. Dibawahnya terdapat Dalvik Virtual Machine yang akan menjalankan aplikasi. Tiap aplikasi akan dijalankan pada proses terpisah dengan Virtual Machine yang berbeda sehingga terisolasi satu sama lain. Aplikasi masih dapat berkomunikasi dengan aplikasi lain melalui mekanisme yang disediakan framework aplikasi. Dalvik VM bergantung pada lapisan dibawahnya ( kernel linux ) untuk multi-threading dan pengelolaan memori tingkat rendah.
- 4) *Kernel linux* bertanggung jawab menyediakan layar dasar seperti keamanan, pengelolaan proses, pengelolaan file, pengelolaan sumber daya memori dan hardware

### 2.7.2 Komponen Aplikasi Android

Juhara, P.Z (2016 : 02) Menyatakan bahwa Aplikasi android ditulis dengan bahasa pemrograman java. Semua file kode intermediate dan asset disatukan dalam satu paket berupa file berekstensi, APK, sebuah file yang dapat didistribusi. Tiap file.APK adalah sebuah aplikasi tunggal. Komponen aplikasi android terdiri dari beberapa jenis, antara lain :

- 1) *Activity* adalah istilah yang digunakan dalam pemrograman android untuk mengacu pada satuan interaksi dengan pengguna melalui antarmuka grafis ( graphical user-interface, GUI ). Sebagai satuan interaksi, Activity adalah tampilan yang anda lihat di layar seperti Windows atau kotak dialog pada pemrograman aplikasi desktop. Tiap aplikasi dapat terdiri dari Nol atau lebih activity. Selain sebagai satuan interaksi dengan pengguna, Activity juga satuan eksekusi. Sebagai satuan eksekusi, Activity selalu memiliki paling tidak satu buah thread, yakni thread utama yang digunakan untuk mempengaruhi tampilan user-interface.

- 2) *Intent* adalah istilah yang digunakan dalam pemrograman Android untuk mengacu pada mekanisme berbagi pesan pemberitahuan atau bertukar data activity atau untuk menjalankan aplikasi lain.
- 3) *Service* adalah komponen aplikasi yang berjalan dibelakang layar tanpa user interface untuk menyediakan layana tertentu seperti mengecek RSS feed secara kontinu atau memainkan music. Service tetap berjalan meski Activity yang mengendalikannya telah berhenti. Media player adalah sebuah contoh aplikasi yang menggunakan service.
- 4) *Content Provider* adalah membuat suatu aplikasi dapat berbagi sejumlah data tertentu kepada aplikasi lain. jika membutuhkan data nama – nama kontak, aplikasi anda tinggal meminta data tersebut.
- 5) *Broadcast Receiver* adalah komponen yang memantau, menerima, dan bereaksi terhadap pesan yang disebarkan, baik oleh sistem maupun aplikasi lain, misalnya ketika baterai lemah, android akan mengirim pesan “baterai lemah” kepada semua broadcast receiver pada dasarnya, anda hanya perlu membuat turunan tipe broadcast receiver, melengkapi metode onReceived() dan mendaftarkannya di Android Manifest.xml atau dengan metode Context.registerReceiver(). Instance broadcast receiver hanya valid selama pemanggilan metode Context.registerReceiver() sehingga onReceive dipanggil adalah mewakilkannya ke komponen lain, misalnya dengan memanggil metode start Activity () atau start Service () milik Content

### 2.7.3 Android Studio

Satyaputra, A., & Aritonang, M, E,( 2016 : 01) Menyatakan bahwa Android studio adalah IDE resmi untuk membangun aplikasi android berdasarkan IntelliJ IDEA. IntelliJ IDEA sendiri adalah java Integrated Development Environment (IDE) yang dikembangkan oleh JetBrains, untuk mengembangkan perangkat lunak komputer. IntelliJ IDE berfungsi dalam membantu anda dalam dunia per-coding-an baik dari segi navigasi,

penyokong produktivitas, hingga code editor yang cerdas. android studio dibangun dengan tujuan mempercepat proses pembangunan maupun pengembangan aplikasi yang berkualitas tinggi untuk setiap device Android

## 2.8 GoogleMaps

Shoddiq, A (2017 : 17) Menyatakan bahwa *GoogleMaps* adalah layanan gratis yang diberikan oleh Google dan sangat populer. *GoogleMaps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *GoogleMaps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *Google Maps* dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *GoogleMapsAPI*. *GoogleMapsAPI* adalah suatu *library* yang berbentuk *JavaScript*.

## 2.9 Global Positioning System (GPS)

W, E,Wishnu (2012 : 32) Menyatakan bahwa GPS adalah singkatan dari *Global Positioning System* yang merupakan system navigasi dengan penggunaan teknologi satelit yang dapat mengirim dan menerima sinyal radio. GPS dapat menentukan posisi dimanapun anda berada dimuka bumi ini.

Satelit sebagai salah satu kunci penting dalam penggunaan teknologi GPS bergerak diorbitnya dengan ketinggian lebih dari 20.000 km di atas permukaan bumi. Ada 24 satelit yang beredar mengitari bumi, 21 satelit beroperasi dan 3 sebagai spare.

Sebagai gambaran umum, satelit mempunyai tiga bagian penting :

1. Komputer : umum untuk mengontrol orbit dan fungsi yang lain
2. Jam atom : untuk keakuratan waktu dengan tingkat kecermatan sampai nano second
3. Radio transmitter : untuk mengirim sinyal kebumi

## 2.10 Elemen GoogleMapsAPI

Shoddiq, A (2017 : 21) Menyatakan bahwa Beberapa elemen dan objek yang terdapat pada Google Maps API yang diterapkan di aplikasi :

- 1) *Markers* adalah fungsional simbol yang menandakan untuk titik koordinat lokasi. *Markers* digunakan untuk menandai lokasi seperti klub dan kejuaraan olahraga pada peta *GoogleMaps*.
- 2) *Directions* adalah *utility* yang menangani masalah pencarian rute. *Directions* ini digunakan untuk memandu rute jalan dari pengguna menuju tempat lokasi klub olahraga dan kejuaraan olahraga pada peta *GoogleMaps*.
- 3) *GeoCoding and Reverse GeoCoding* adalah *utility* untuk memproses konversi alamat yang dapat digunakan untuk menempatkan penanda atau posisi peta.
- 4) *Location Manager* adalah API yang paling penting, dalam memberikan akses ke layanan sistem lokasi dengan pemetaan dan GPS. Dalam hal ini *Location Manager* berfungsi untuk mendapatkan *update* periodik lokasi geografis perangkat dengan memetakan nilai *latitude* dan *longitude* kedalam *GoogleMaps*.
- 5) *Overlays Overview* adalah objek di *GoogleMaps* yang terkait dengan koordinat lintang dan bujur, sehingga objek ini bergerak saat ditarik atau *zoom* pada peta. *Overlays* mencerminkan benda yang ada pada peta untuk menunjukkan titik, garis, area, atau *markers* benda.
- 6) *Loading GoogleMapsAPI* koneksi *script* yang dibuat keserver *GoogleMapsAPI* dengan menggunakan *APIkey* yang didapatkan pada saat mendaftar ke *Google Maps API*.
- 7) *Map DOM Elements* Untuk menampilkan Maps pada halaman web, umumnya menggunakan *div* tag untuk menampungnya. Diharuskan membuat sebuah *div* tag dengan nama "map\_canvas", dan mereferensikan *elements* ini ke *Document Object Model* (DOM).

## 2.11 Web Service

Kenali, W.E. (2013 : 06) Menyatakan bahwa *Web Service* adalah layanan layanan yang disediakan adalah layanan-layanan yang disediakan internet, menggunakan pengiriman pesan format XML, dan tidak bergantung pada satu macam system operasi atau bahasa pemrograman.

Secara umum web service memiliki tiga operasi yang terlibat di dalamnya yaitu :

- 1) Publish/unpublish yaitu yang menerbitkan atau menghapus layanan ke/dari registry
- 2) Find yaitu service requestor mencari dan menemukan layanan yang dibutuhkan
- 3) Bind yaitu interaksi dan mengakses layanan yang disediakan layananservice provider.

Di dalam layanan web service terdapat Tiga komponen utama yaitu :

- 1) Service provider
- 2) Service requestor
- 3) Service registry

## 2.12 JSON

Perangsinangin, K. (2006:12) JSON (*JavaScriptObjectNotation*) 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 Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman 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 JSON terbuat dari dua struktur :

- 1) Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hashtable*), daftar berkunci (*keyedlist*), atau *associativearray*.
- 2) Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

- 1) Objek  
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).
- 2) Larik  
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [ (kurung kotak buka) dan diakhiri dengan ] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).
- 3) Nilai  
Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.
- 4) String  
String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash *escapes* "`\`" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.

5) Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksa desimal tidak digunakan.

### 2.13 PHP

Perangsinangin, K.(2006 : 28) Menyatakan bahwa PHP merupakan singkatan dari Hypertext Preprocessor yang digunakan sebagai bahasa script server-side dalam pengembangan web yang disisipkan pada dokumen HTML. PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya PHP di gunakan untuk mencatat jumlah serta mengetahui siapa saja pengunjung homepage-nya dan tak lama keluar lah PHP versi 1.0. Pada tahun 1996 PHP telah banyak dgunakan dalam website di sunia. Sebuah kelompok pengembang software yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken, Shane Caraveo bekerja sama untuk menyempurnakan PHP 2.0, akhirnya pada tahun 1998 keluarlah PHP versi 3.0, penyempurnaan terus dilakukan tahun 2000 di luncurkan PHP versi 4.0 todak berhenti sampai disitu kemampuan PHP semakin meningkat hingga versi terbaru telah dikeluarkan PHP versi 5.0.

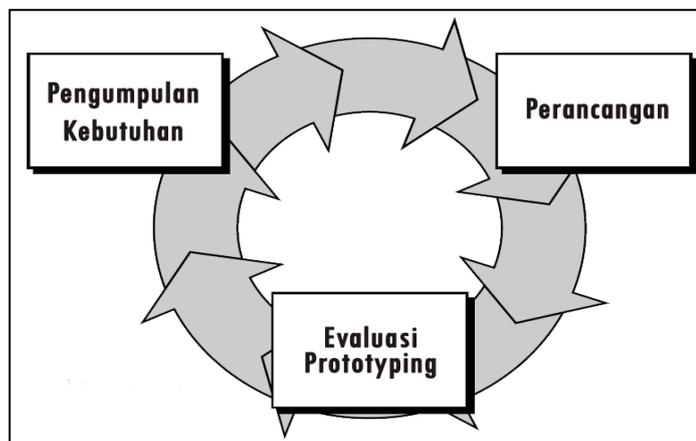
PHP memiliki kelebihan yang tidak dimiliki oleh bahasa script sejenis. PHP difokuskan pada pembuatan script server-side yang bisa melakukan apa saja yang dapat dilakukan oleh CGI seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis dan kemampuan mengirim serta menerima cookies bahkan lebih dari pada kemampuan CGI

PHP tidak terbatas pada hasil keluaran HTML ( Hypertext Markup Languages ) tetapi juga memiliki kemampuan keluaran gambar, file PDF, dan movie flash, PHP juga menghasilkan teks seperti XHTML dan file XML lainnya.

## 2.14 Metode Pengembangan Perangkat Lunak

### 2.14.1 Model *Prototype*

A.S.Rosa (2016 : 31) menyatakan bahwa *Prototyping* paradigma dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi *prototyping* yang dapat dilihat pada gambar 2.4 berikut :



Gambar 2.4 *Model Prototype*

Tahap–tahap rekayasa *software* dalam *prototype model* pada gambar 2.4 di atas adalah sebagai berikut :

#### 1) Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan. Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis kebutuhan pelayanan darurat.

2) Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan

3) Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan/kerusakan yang tidak ditemukan pada langkah sebelumnya

Kelebihan dari *Prototype Model* adalah sebagai berikut :

- 1) End user dapat berpartisipasi aktif.
- 2) Penentuan kebutuhan lebih mudah diwujudkan.
- 3) Mempersingkat waktu pengembangan *software*.

Kekurangan dari *Prototype Model* adalah sebagai berikut:

- 1) Proses analisis dan perancangan terlalu singkat.
- 2) Mengesampingkan alternatif pemecahan masalah.
- 3) Biasanya kurang fleksibel dalam menghadapi perubahan.
- 4) *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
- 5) *Prototype* terlalu cepat selesai.

## 2.15 UML (*Unified Modeling Language*)

### 2.15.1 Pengertian UML

UML merupakan bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak ) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak. Selain itu UML adalah bahasa

pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Crop. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namu hampir dalam semua bidang yang membutuhkan pemodelan.

### 2.15.2 Bagian-Bagian UML

Bagian-bagian utama dari UML adalah *view*, diagram, model element, dan *general mechanism*. Diagram berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain:

1) *Use Case Diagram*

*Use case* adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe intraksi antara lain user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan *Use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan clinet.

2) *Class Diagram*

*Class* adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *Class diagram* dapat memeberikan pandangan global atas sebuah system. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat memebantu dalam visualisasi setruktur kelas dari suatu sistem.

3) *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau intraksi.

### **2.15.3 Tujuan dan Keunggulan UML**

Tujuan UML adalah sebagai berikut: Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object, menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Keunggulan menggunakan UML dibandingkan menggunakan metodologi terstruktur:

1) *Uniformity*

Pengembangan cukup menggunakan satu metodologi dari tahap analisis hingga perancangan. Memungkinkan merancang komponen antarmuka secara terintegrasi bersama perancangan perangkat lunak dan perancangan struktur data.

2) *Understandability*

Kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah yang sesungguhnya sehingga lebih mudah untuk dipahami.

3) *Stability*

Kode program yang dihasilkan relatif stabil sepanjang waktu, karena mendekati permasalahan yang sesungguhnya.

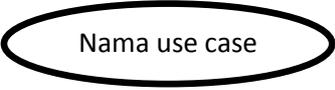
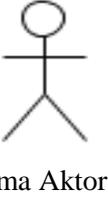
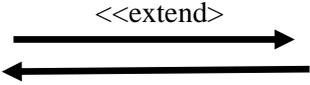
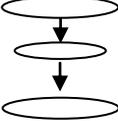
4) *Reusability*

Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada akhirnya akan sangat mempercepat waktu pengembangan perangkat lunak (atau sistem informasi).

### 2.15.4 Simbol-Simbol pada UML

Simbol-simbol yang terdapat dalam diagram UML. Dapat dilihat pada tabel 2.1 dibawah ini :

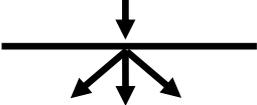
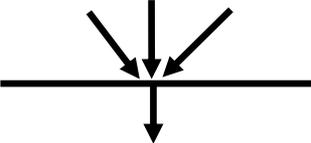
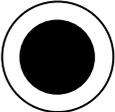
Tabel 2.1 Simbol Pada *Diagram* UML.

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anatar unit atau aktor, biasanya akan diterangkan dengan menggunakan kata kerja diawal-diawal frase nama <i>use case</i>.</p>
<p><i>Aktor/Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berintraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang. Biasanya akan dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.</p>
<p><i>Asosiasi/Association</i></p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p><i>Ekstensi/Extend</i></p> 	<p>Case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal .</p>  <p>Arah panah mengarah pada use case yang ditambahkan.</p>
<p><i>Uses</i></p> 	<p>Digunakan sebagai kegiatan utama atau syarat menuju use case berikutnya.</p>

### 2.15.5 Activity Diagram

Diagram aktivitas digunakan untuk menggambarkan alur kerja suatu sistem informasi. Sebuah diagram aktivitas menunjukkan suatu alur kegiatan secara berurutan. Tabel 2.2 dibawah ini adalah simbol-simbol yang ada pada diagram aktifitas:

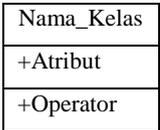
Tabel 2.2 Simbol-Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Setatus awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem. Aktivitas biasanya diawali dengan kata kerja.
Pencabangan / decision 	Asosiasi penggabungan dimana lebih satu aktivitas.
Fork 	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.
Penggabungan / Join 	Digunakan untuk menunjukkan kegiatan yang digabungkan.
End Point 	Mengakhiri aktivitas sistem.

### 2.15.6 Class Diagram

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut yaitu variabel-variabel yang dimiliki oleh suatu kelas dan operasi atau metode yaitu fungsi-fungsi yang dimiliki oleh suatu kelas. Tabel 2.3 dibawah ini adalah simbol-simbol yang ada pada *clas diagram*:

Tabel 2.3 Bagan *Class Diagram*

Simbol	Deskripsi
<p><b>Kelas</b></p> 	Kelas pada Struktur
<p><b>Interface</b></p>  <p>Nama <i>Interface</i></p>	Metode pada <i>interface</i> yang digunakan pada suatu kelas sama persis dengan yang ada pada <i>interface</i> .
<p><b>Asosiasi</b></p> 	Relasi antara kelas dengan makna umum.
<p><b>Asosiasi Berarah</b></p> 	Relasi antar kelas dengan makna kela yang satu digunakan pada kelas lain.