

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem Informasi Geografis**

SIG (Sistem Informasi Geografis) atau sistem informasi berbasis pemetaan dan geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang terkait dengan sistem pemetaan dan analisis terhadap segala sesuatu, serta peristiwa-peristiwa yang terjadi di muka bumi. Teknologi SIG (Sistem Informasi Geografis) mengintegrasikan operasi pengolahan data berbasis *database* yang biasa digunakan, seperti pengambilan data berdasarkan kebutuhan serta analisis statistic dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar tertentu. Konsep SIG (Sistem Informasi Geografis) telah diperkenalkan di Indonesia sejak pertengahan tahun 1980-an, dan kini telah dimanfaatkan di berbagai bidang baik negeri maupun swasta. Kemampuan dasar dari SIG (Sistem Informasi Geografis) adalah mengintegrasikan berbagai operasi basis data seperti *query*, menganalisisnya, dan menyimpan serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan SIG (Sistem Informasi Geografis) dengan sistem informasi lain. Komponen SIG (Sistem Informasi Geografis) terdiri atas *hardware*, *software*, data, dan *user*. Dengan adanya SIG (Sistem Informasi Geografis) diharapkan tersedia informasi yang cepat, benar dan akurat tentang keadaan di lingkungannya (Hersa Farida Qoriani ;

Jurnal Sistem Informasi Geografis Untuk Mengetahui Tingkat Pencemara  
Limbah Pabrik Di Kabupaten Sidoarjo : 2012 : 2)

### **II.1.1. Komponen Sistem Informasi Geografis**

Menurut Bramantiyo Marjuki (2014 : 2) Sebagai salah satu jenis sistem informasi, SIG (Sistem Informasi Geografis) mempunyai sub sistem atau komponen yang bekerja secara bersama untuk menghasilkan fungsionalitas SIG (Sistem Informasi Geografis). Komponen SIG (Sistem Informasi Geografis) terdiri dari *hardware*, *software*, data dan metode.

1. *Hardware* atau perangkat keras merupakan media tempat pelaksanaan proses SIG (Sistem Informasi Geografis). *Hardware* yang diperlukan dalam sebuah SIG (Sistem Informasi Geografis) meliputi perangkat keras untuk masukan data, penyimpanan data, pengolahan dan analisa data dan pembuatan keluaran.
2. *Software* atau perangkat lunak merupakan alat pelaksana pekerjaan SIG (Sistem Informasi Geografis). *Software* standar SIG (Sistem Informasi Geografis) harus mempunyai kapabilitas data input, penyimpanan, manajemen data, transformasi dan konversi data, analisa dan pembuatan keluaran.
3. Data atau representasi dari sebuah obyek/fenomena adalah bahan yang dianalisa di dalam SIG (Sistem Informasi Geografis). SIG (Sistem Informasi Geografis) memerlukan sebuah jenis data yang spesifik agar dapat memberikan keluaran seperti fungsionalitasnya. Data yang digunakan dalam SIG (Sistem Informasi Geografis) adalah data geospasial atau data yang berefrensi geografis (mempunyai informasi lokasi).

4. Metode adalah cara bagaimana data diolah untuk menjadi sebuah informasi. Metode meliputi aspek pemasukan data ke dalam sistem, bagaimana data dikelola dan disimpan, bagaimana data dianalisis, dan bagaimana informasi ditampilkan.

### **II.1.2. Data Spasial**

Sebagian besar data yang akan ditangani dalam SIG (Sistem Informasi Geografis) merupakan data spasial yaitu sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (attribute) yang dijelaskan berikut ini :

1. Informasi lokasi (spasial), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi datum dan proyeksi.
2. Informasi deskriptif (atribut) atau informasi non spasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya, contohnya: jenis vegetasi, populasi, luasan, kode pos, dan sebagainya (G. Manjela Eko Hartoyo ; 2010 : 2).

### **II.1.3. Format Data Spasial**

Secara sederhana format dalam bahasa komputer berarti bentuk dan kode penyimpanan data yang berbeda antara file satu dengan lainnya. Dalam SIG

(Sistem Informasi Geografis), data spasial dapat direpresentasikan dalam dua format, yaitu :

- a) Data Raster Data raster (atau disebut juga dengan sel grid) adalah data yang dihasilkan dari sistem Penginderaan Jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan pixel (picture element).
- b) Data Vektor Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (merupakan titik perpotongan antara dua buah garis). Keuntungan utama dari format data vektor adalah ketepatan dalam merepresentasikan fitur titik, batasan dan garis lurus (G. Manjela Eko Hartoyo ; 2010 : 2)

#### **II.1.4. Sumber Data Spasial**

Salah satu syarat SIG (Sistem Informasi Geografis) adalah data spasial, yang dapat diperoleh dari beberapa sumber antara lain :

- a) Peta Analog Peta analog (antara lain peta topografi, peta tanah dan sebagainya) yaitu peta dalam bentuk cetak. Pada umumnya peta analog dibuat dengan teknik kartografi, kemungkinan besar memiliki referensi spasial seperti koordinat, skala, arah mata angin dan sebagainya. Dalam tahapan SIG (Sistem Informasi Geografis) sebagai keperluan sumber data, peta analog dikonversi menjadi peta digital dengan cara format raster diubah menjadi

format vektor melalui proses digitasi sehingga dapat menunjukkan koordinat sebenarnya di permukaan bumi.

- b) Data Sistem Penginderaan Jauh Data Penginderaan Jauh (antara lain citra satelit, foto udara dan sebagainya), merupakan sumber data yang terpenting bagi SIG (Sistem Informasi Geografis) karena ketersediaanya secara berkala dan mencakup area tertentu.
- c) Data Hasil Pengukuran Lapangan Data pengukuran lapangan yang dihasilkan berdasarkan teknik perhitungan tersendiri, pada umumnya data ini merupakan sumber data atribut contohnya : batas administrasi, batas kepemilikan lahan, batas persil, batas hak pengusahaan hutan dan lain-lain.
- d) Data GPS (*Global Positioning System*) Teknologi GPS memberikan terobosan penting dalam menyediakan data bagi SIG (Sistem Informasi Geografis). Keakuratan pengukuran GPS semakin tinggi dengan berkembangnya teknologi satelit navigasi. Pengolahan data yang bersumber dari GPS biasanya dilakukan dalam format vector (G. Manjela Eko Hartoyo ; 2010 : 4)

### **II.1.5. Arsitektur Sistem Informasi Geografis**

Untuk dapat melakukan komunikasi dengan komponen yang berbeda-beda di lingkungan web maka dibutuhkan sebuah *web server*. Karena standart dari geo data berbeda beda dan sangat spesifik maka pengembangan arsitektur system mengikuti arsitektur "*Client Server*".

Applikasi berada disisi client yang berkomunikasi dengan Server sebagai penyedia data melalui web Protokol seperti HTTP (*Hyper Text Transfer*

*Protocol*). Aplikasi seperti ini bisa dikembangkan dengan *web browser* (Mozilla Firefox, Opera, Internet Explorer, dll). Untuk menampilkan dan berinteraksi dengan data GIS, sebuah browser membutuhkan *Pug-In* atau *Java Applet* atau bahkan keduanya. Web Server bertanggung jawab terhadap proses permintaan dari client dan mengirimkan tanggapan terhadap respon tersebut. Dalam arsitektur web, sebuah web server juga mengatur komunikasi dengan server side GIS Komponen. Server side GIS Komponen bertanggung jawab terhadap koneksi kepada database spasial seperti menterjemahkan *query* kedalam SQL dan membuat representasi yang diteruskan ke server. Dalam kenyataannya Side Server GIS Komponen berupa software libraries yang menawarkan layanan khusus untuk analisis spasial pada data. Selain komponen hal lain yang juga sangat penting adalah aspek fungsional yang terletak di sisi client atau di server (Denny Charter ; 2011 : 1 – 2).

## **II.2. Metode *Haversine Formula***

Metode Haversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*latitude*) sebagai variabel inputan. Haversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2,

lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut :

$$x = (\text{lon2}-\text{lon1}) \cdot \cos((\text{lat1}+\text{lat2})/2);$$

$$y = (\text{lat2}-\text{lat1});$$

$$d = \sqrt{x^2+y^2} \cdot R$$

Keterangan :

x = Longitude (Lintang)

y= Lattitude ( Bujur)

d= Jarak

R= Radius Bumi =6371 km

1 derajat= 0.0174532925 radian (Dwi Prasetyo ; 2012 : 2).

Perlu diketahui bahwa argumen untuk fungsi haversine diasumsikan di sini untuk diberikan dalam radian. Dalam derajat, haversin ( $d / R$ ) dalam rumus akan menjadi haversin ( $180^\circ d / \pi R$ ). Mengingat lingkup pada haversine merupakan segitiga pada permukaan bola yang didefinisikan oleh lingkaran – lingkaran besar yang menghubungkan tiga titik u , v , dan w pada bola.

Jika panjang dari ketiga sisi adalah A (u ke v) , b (dari u untuk w), dan c (dari v ke w), dan sudut sudut c sebaliknya adalah C, maka hukum haversines sebagai berikut :

$$\text{Haversin}(c) = \text{haversin}(a-b) + \sin(a) \sin(b) \text{haversin}(c)$$

Karena ini adalah lingkup unit, panjang a, b, dan c hanya sama dengan sudut (dalam radian) dengan sisi dari pusat bola sedangkan untuk lingkup non-unit, masing-masing sama panjang busur untuk sudut pusat yang dikalikan dengan jari-jari bola. Jika dalam satu kasus khusus dimana U adalah kutub utara,

sementara  $v$  dan  $w$  adalah dua poin pemisah yang akan ditentukan Dalam hal ini,  $a$  dan  $b$  adalah  $\pi / 2 - \phi_{1, 2}$  (yaitu,  $90^\circ$  - lintang),  $C$  adalah longitude  $\Delta\lambda$  pemisahan, dan  $c$  adalah dikehendaki  $d / R$ . Memperhatikan bahwa  $\text{havrsin}(\pi / 2 - \phi) = \cos(\phi)$ , maka Haversin-nya akan menjadi sebagai berikut :

$$\text{Cos}(c) = \text{cos}(a) \text{cos}(b) + \text{sin}(a) \text{sin}(b) \text{cos}(c) \text{ (Nurul Huda ; 2012 : 2).}$$

### II.3. Pengertian Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan oleh *Sun Microsystem* dan dirilis tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan *versi* yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, serta Sembilan pemrograman lainnya dari *Sun Microsystem*. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang (Wahana Komputer ; 2010 : 1)



## II.4. Pengertian Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (M. Ichwan ; 2011 : 15).

### II.4.1. Versi Android

Berikut adalah versi android dari tahun ke tahun :

1. Android versi 1.1 Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.
2. Android versi 1.5 (*Cupcake*) Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset

Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan system

3. Android versi 1.6 (Donut) Donut (versi 1.6) dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.
4. Android versi 2.0/2.1 (Eclair) Pada 3 Des 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.
5. Android versi 2.2 (Froyo: Frozen Yoghurt) Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market

6. Android versi 2.3 (Gingerbread) Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.
7. Android versi 3.0/3.1 (Honeycomb) Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.
8. Android versi 4.0 (ICS: Ice Cream Sandwich) Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan

berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

9. Android versi 4.1 (Jelly Bean) Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus7.
10. Android versi 4.2 (Jelly Bean) Mengusung slogan “smart, simple, and truly yours“, fitur pertama yang diangkat oleh Android 4.4 Kitkat adalah fitur voice search. Meski sudah agak lama memiliki fitur pencarian dengan suara, pada iterasi terbaru ini Google mengimplementasikan aplikasi voice search sebagai aplikasi yang bisa diakses dengan panggilan saja, tanpa harus menyentuh perangkat sama sekail. Cukup dengan mengucapkan “Ok Google”, maka aplikasi siap untuk melakukan pencarian yang Anda perintahkan. Google juga memberikan beberapa perubahan lain seperti dukungan multitasking yang lebih cepat (Nurul Khotimah ; 2014 : 2).

## II.5. Pengertian Basis Data

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*. (Agustinus Mujilan ; 2012 : 23)

## II.6. Pengertian Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan

### II.6.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1<sup>st</sup> Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing *cell* bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

c. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

d. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi  $X \rightarrow A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
  - Atau A merupakan bagian dari *primary key* pada tabel tersebut.
- e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

- f. Boyce Code Normal Form (BCNF)
- Memenuhi 1<sup>st</sup> NF
  - Relasi harus bergantung fungsi pada atribut superkey (Janner Simarmata ; 2010 : 76).

## II.6. Pengertian MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah *Oracle*, *Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, menurutkan dan mengambil data secara efisien. *Server MySQL* yang akan membantu melakukan

fungsionalitas tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah *SQL-standar* bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan dan disokong oleh sebuah perusahaan Swedia bernama MySQL AB. RDBMS ini berada di bawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial. Apabila menggunakan MySQL sebagai basis data dalam suatu situs Web. Anda tidak perlu membayar, akan tetapi jika ingin membuat produk RDBMS baru dengan basis MySQL dan kemudian menguálnua, anda wajib bertemu mudah dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10).

## **II.7. Pengertian PHP**

PHP merupakan suatu bahasa pemrograman sisi server yang dapat anda gunakan untuk membuat halaman Web dinamis. Contoh bahasa yang lain adalah *Microsoft Active Server Page (ASP)* dan *Java Server Page(JSP)*. Dalam suatu halaman HTML anda dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi. Karena kekayaannya akan fitur yang mempermudah perancangan dan pemrograman Web, PHP memiliki popularitas yang tinggi. Anda dapat mengecek *survey* popularitas yang dilakukan *netcraft* di URL [www.php.net/usage.php](http://www.php.net/usage.php).

PHP adalah kependekan dari *HyperText Preprocessor* (suatu akronim rekursif) yang dibangun oleh RasmusLerdorf pada tahun 1994. Dahulu, pada awal pengembangannya PHP disebut sebagai kependekan dari *Personal Home Page*. PHP merupakan produk *Open Sources* sehingga anda dapat mengakses *source*



*code*, menggunakan dan mengubahnya tanpa harus membayar sepeser pun. Gratis! (Antonius Nugraha Widhi Pratama ; 2010 : 9).

## **II.8. UML (*Unified Modeling Language*)**

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


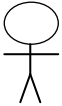

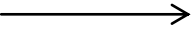
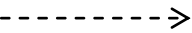
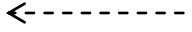
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah *use case* diagram, *activity* diagram, *sequence* diagram dan *class* diagram.

### **1. *Use case* Diagram**

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol *Use Case*



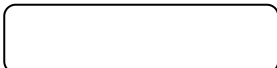
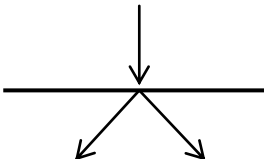
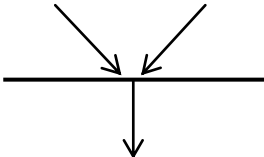
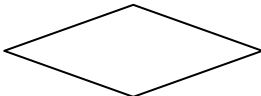
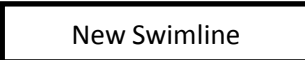
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol *Activity Diagram*

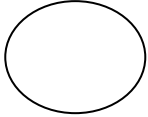
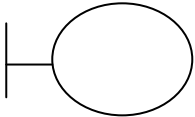
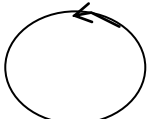

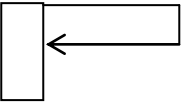
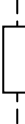

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.4. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)