BAB 2
TINJAUAN PUSTAKA

2.1. Tinjauan Umum Lembaga

Tinjauan umum lembaga dimaksudkan untuk mengetahui keadaan instansi antara lain tentang sejarah, visi dan misi, fungsi dan tujuan, logo dan arti dari lambangnya dari Lembaga Informasi Perburuhan Sedane.

2.1.1. Sejarah Singkat Lembaga Informasi Perburuhan Sedane

Lembaga Informasi Perburuhan Sedane (LIPS) adalah organisasi sosial, yang berfokus pada demokratisasi dan politisasi serikat buruh dengan menyediakan informasi dan melakukan dokumentasi melalui kegiatan penelitian partisipatif dan publikasi reguler.

LIPS lahir dari situasi yang mencekam ketika krisis politik dan ekonomi bergejolak berpadu dengan meningkatnya geliat gerakan sosial, pada 1991. Bersama organisasi non pemerintah yang bergerak di bidang agraria, lingkungan, bantuan hukum dan hak asasi manusia, LIPS menempatkan diri sebagai pelaksana kegiatan asistensi peningkatan kapasitas aktivis buruh, dokumentasi dan penyediaan informasi untuk penguatan gerakan buruh.

Kini, secara umum situasi ekonomi dan politik berubah, termasuk keadaan gerakan buruh. Paling mencolok dari perubahan tersebut adalah kebebasan mendirikan organisasi dan tren sistem kerja fleksibel dalam wujud buruh kontrak dan outsourcing.

Dalam perjalannya LIPS terus mengembangkan kapasitasnya dengan melembagakan sistem kerja yang efektif dan efisien. Hal ini tecermin dari perubahan status yayasan menjadi perkumpulan pada 2007. Dari perubahan tersebut, LIPS berupaya menjalankan sistem pengawasan internal, demokratisasi pengambilan keputusan dan menjaga akuntabilitas lembaga, sebagaimana tecermin dalam struktur kelembagaan dan pengambilan keputusan lembaga. Awal 2012 lalu, keuangan LIPS dinyatakan present fairly oleh lembaga audit independen Kantor Audit Publik Amir Hadly Jakarta.

2.1.2. Visi dan Misi Lembaga Informasi Perburuhan Sedane (LIPS)

Lembaga Informasi Perburuhan Sedane (LIPS) mempunyai visi dan misi sebagai berikut:

Visi LIPS adalah:

1. Mendorong gerakan buruh agar mampu memperjuangkan kepentingan ketika berhadapan dengan pemodal dan negara sehingga dapat mencapai kondisi hidup sejahtera, bebas dari sistem kerja yang eksploitatif dan tidak adil baik dalam hubungan jender maupun kelas.
2. Mendukung kondisi yang menghormati, memajukan dan melindungi nilai-nilai demokrasi dan hak asasi manusia.

Sedangkan Misi LIPS adalah:

1. Mendokumentasikan kegiatan-kegiatan yang berhubungan dengan gerakan buruh di berbagai sektor.
2. Melakukan penelitian-penelitian yang berbasis metodologi parsitipatif.
3. Membangun jaringan gerakan buruh di tingkat nasional maupun internasional.
4. Menyediakan informasi tentang dinamika keserikat buruh di Indonesia maupun luar negeri, tentang pola pengembangan organisasi, tentang modul pendidikan, dan informasi yang lainnya untuk melakukan advokasi dan kampanye.
5. Menyediakan analisis kontemporer perburuhan.

2.1.3. Logo Lembaga Informasi Perburuhan Sedane (LIPS)

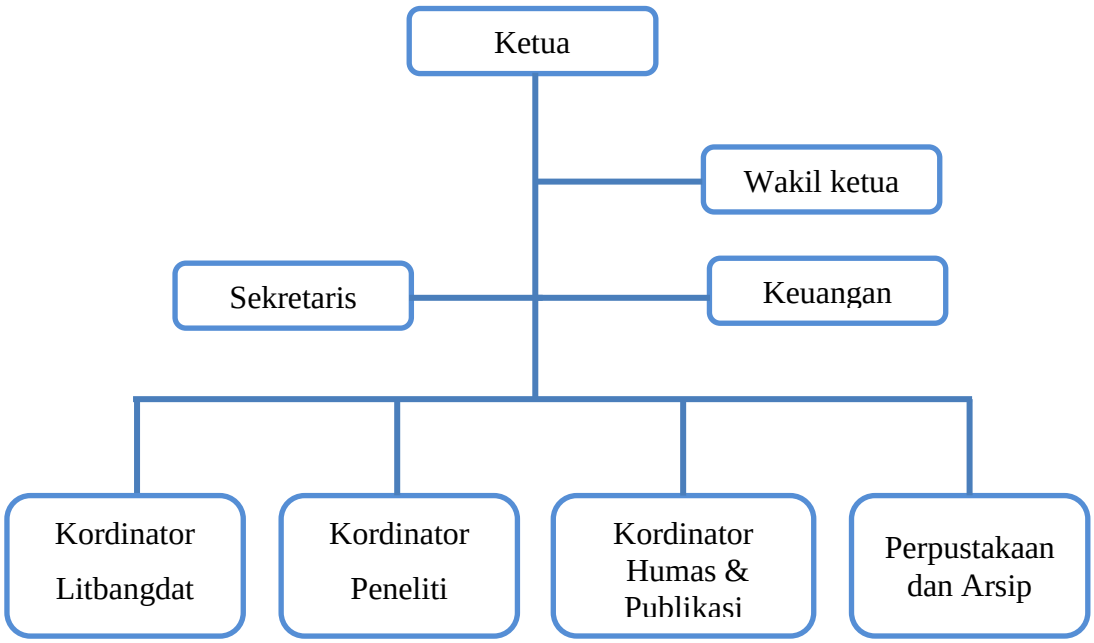
Setiap instansi, organisasi atau lembaga mempunya logo masing-masing uuntuk membedakan dan menjadi salah satu karakter atau ciri dari suatu organisasi. Adapun logo dari Lembaga Informasi Perburuhan Sedane (LIPS) dapat diihat pada Gambar 2.1.

Gambar 2.1 Logo Lembaga Informasi Perburuhan Sedane (LIPS)

Logo Lembaga Informasi perburuhan Sedane (LIPS) secara keseluruhan adalah berupa roda bergerigi (*gear)* yang melambangkan dunia industri dimana dunia industri adalah ruang lingkup kerja dari Lembaga Informasi Perburuhan Sedane (LIPS).

2.1.4. Struktur Organisasi

Berikut adalah struktur organisasi Lembaga Informasi Perburuhan Sedane (LIPS) bisa dilihat pada Gambar 2.2.



Gambar 2. 2 Struktur Organisasi Lembaga Informasi perburuhan Sedane (LIPS)

2.2. Landasan Teori

Landasan teori adalah seperangkat definisi, konsep serta proposisi yang telah disusun rapi serta sistematis tentang variabel-variabel dalam sebuah penelitian. Landasan teori sangat penting dalam sebuah penelitian terutama dalam penulisan karya tulis ilmiah. Penelitian tidak bisa mengembangkan masalah yang mungkin ditemui di tempat penelitian. Jika tidak memiliki acuan landasan teori yang mendukungnya.

2.2.1. Sistem Informasi

Pengertian sistem menrut Jogianto mengemukakan bahwa sistem adalah kumpulan dari elemen elemen yang berinteraksi untuk mecapai suatu tujuan tertetu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi[1]. Dengan demikian sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan,berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu.

Sedangkan Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event)* adalah kejadian yang terjadi pada saat tertentu. Menurut Gordon B. Davis: Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang[2]. Nilai informasi ditentukan oleh dua hal yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan lebih bernilai jika manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

Secara umum dapat di definisikan sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu[2]:

1. Blok masukan (*input block*)

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen dasar.

2. Blok model (*model block*)

   Blok ini terdiri darikombinasi prosedur, logika dan metode matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentuuntuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technologi block*)

   Teknologi digunakan untuk menerima input, menjalankan model,menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan.

   Teknologi terdiri dari unsur utama:

   a. Teknisi (*human ware* atau *brain ware*)
   b. Perangkat lunak (*software*)
   c. Perangkat keras (*hardware*)

5. Blok basis data (*data base block*)

   Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras computer dan digunakan perangkat lunak uuntuk memanipulasinya.

6. Block kendali (*control block*)

   Banyak factor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperature tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan,

sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

2.2.2. Sistem Informasi Geografis

Geografi adalah ilmu yang mempelajari permukaan bumi dengan menggunakan pendekatan keruangan, ekologi, dan kompleks wilayah. Fenomena yang diamati merpakan dinamika perkembangan dan pembangunan wilayah yang ada dalam keseharian, misalnya informasi mengenai letak dan persebaran dari kejadian-kejadian alamiah maupun fenomena terdapat sumberdaya. Ketersediaan data yang bersifat geografi, dimna memiliki atribut utama keruangan, akan memudahkan banyak kepentingan.

Arnoff mendefinisikan Sistem Informasi Geografis sebagai sebuahsistem berbasisan komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis. Sistem Informasi Geografis dirancang untuk mengumpulkan, menyimpan dan menganalisa objek-objek dan fenomena dimana lokasi geografi merupakan karakteristik yang penting atau kritis untuk dianalisis.

Dalam pengelolaan Sistem Informasi Geografis yang perlu mandapat perhatian tidak hanya sekedar aspek peta digital, meskipun hal ini yang utama. Hal lain yang tidak kalah penting adalah aspek pengelolaan *database* yang dikandungnya yang merupakan atribut peta. Selain itu Sistem Informasi Geografi juga dapat menyerap dan mengolah data dari bermacam sumber yang memiliki skala dan struktur yang berbeda dan dapat melakukan operasi data-data keruangan yang bersifat kompleks.

Sistem Informasi Geografis, dapat diuraikan menjadi beberapa subsistem sebagai berikut[3]:

1. Data Input

   Sub-sistem ini bertugas untuk mengumpulkan, mempersiapkan dan menyimpan data spasial dan atributnya dari berbagai sumber. Sub-sistem ini pula yang bertanggug jawab dalam mengkonversikan atau mentransformasikan format-format data aslinya ke dalam format yang

dapat digunakan oleh perangkat sistem informasi geografi yang bersangkutan.

2. Data Output

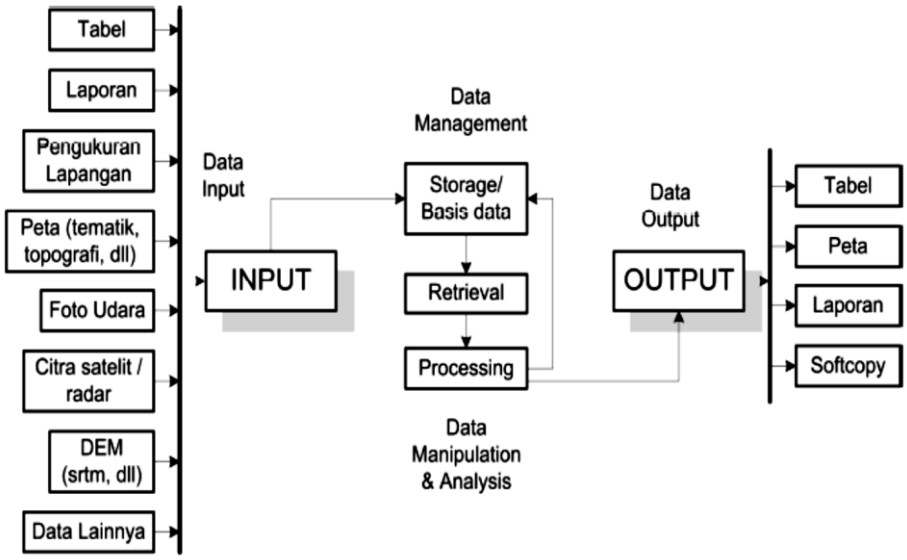
   Sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengekspornya ke format yang dikehendaki) seluruh atau sebagian basis data (spasial) baik dalam bentuk softcopy maupun hardcopy seperti halnya tabel, grafik, report, peta dan lainsebagainya.

3. Data Management

   Sub-sistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait dalam sebuah sistem basis data sedemikian rupa sehingga mudah dipanggil kembali atau di-*retrive,*di-*update* dan diedit.

4. Data Manipulation dan Analisis

   Sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh sistem informasi geografi. Selain itu sub-sistem ini juga melakukan manipulasi (evaluasi dan penggunaanfungsi-fungsi dan operator matematis dan logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan.



Gambar 2.3 Ilustrasi Uraian Sub-sistem Sistem Informasi Geografis [4]

2.2.2.1. Komponen Sistem Informasi Geografis

Sistem informasi geografis salah satu sistem yang kompleks dan pada umumnya juga (selain yang *stand-alone*) terintegrasi dengan lingkungan sistem computer lainnya ditingkat fungsional dan janringan (*network*). Jika diuraikan sistem informasi geografis sebagai sistem terdiri dari beberapa komponen dengan berbagai karakteristiknya[4]:

1. Perangkat Keras

   Pada saat ini sistem informasi geografis sudah tersedia bagi berbagai *platform* perangkat keras, mulai dari kelas PC *desktop*, *workstation*, hingga *multi-user host* yang bahkan dapat digunakan oleh banyak orang bersamaan (simultan) dalam jaringan komputer yang luas, tersebar, berkemampuan tinggi, memiliki ruang penyimpanan (*hardisk*) yang besar dan mempunyai kapasistas memori (RAM) yang besar. Adapun perangkat keras yang sering digunakan untuk aplikasi sistem informasi geografis adalah komputer (PC), *mouse*, monitor (plus VGA-*card* grafik) yang beresolusi tinggi, *digitizer, printer, plotter, reciver* GPS dan *scanner.*

2. Perangkat Lunak

   Dari sudt pandang yang lain, sistem informasi geografis bisa juga merupakan sistem perangkat lunak yang tersusun secara modular dimana sistem basis datanya memegang peranan kunci. Pada kasus perangkat sistem informasi geografis tertentu, setiap sub-sistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari ratusan modul program (*.exe) yang masing-masing dapat dieksekusi tersendiri.

3. Data dan Informasi Geografis

   Sistem informasi geografis dapat mengmpulkan dan menyimpan data atau informasi yang diperlukan baik secara tidak langsung (dengan cara meng-*import*-nya dari format-format perangkat lunak sistem informasi geografis yang lain) maupun secara langsung dengan cara melakukan dijitasi data spasialnya (dijitasi *on-screen* atau *heads-ups* diatas tampilan layar monitor, atau manual dengan menggunakan *digitizer*) dari peta analog dan

kemudian memasukan data atributnya dari tabel-tabel atau laporan dengan menggunakan *keyboard*

4. Manajemen

   Suatu proyek sistem informasi geografis akan berhasil jika dikelola dengan baik dan dikerjakan oleh orang-orang memiliki keahlian (kesesuaian dengan *job-description* yang bersangkutan) yang tepat pada semua tingkatan.

2.2.2.2. Model Data Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) adalah kesatuan sistem yang terdiri dari berbagai komponen namun tidak hanya perangkat keras komputer beserta dengan perangkat lunaknya saja tetapi harus tersedia data geografis yang benar dan sumber daya manusia untuk melaksanakan perannya dalam memformulasikan dan menganalisa persoalan yang menentukan keberhasilan SIG. dalam SIG terdapat dua jenis data yaitu data spasial dan non-spasial.

2.2.2.3. Model Data Spasial Sistem Informasi Geografis

Data spasial merupakan data yang memuat tentang lokasi suatu objek dalam peta berdasarkan posisi geografis objek tersebut di atas bumi dengan menggunakan sistem koordinat data spasial deirepresentasikan dengan model vector dan model raster.

Model data vector diwakili oleh symbol-simbol yang terdiri atas interkoneksi garis dan titik yag merepresentasikan lokasi dan garis batas dari entitas geografi, diantaranya *lines* (garis), *polylines (polygon)*, *points* (titik), *area* (daerah) dan *nodes* (titk potong). Model data raster dihasilkan dari teknologi pemotretan melalui satelit dan udara, yang mempresentasikan objek geografi sebagai struktur *cell* dikenal sebagai *pixel*.[5]

2.2.2.4. Model Data Non-Spasial Sistem Informasi Geografis

Data non-spasial merupakan data yang memuat karakteristik atau keterangan dari suatu objek yang terdapat dalam peta yang sama sekali tidak berkaitan

dengan posisi geografi objek tertentu. Sebagai contoh luas wilayah, jumlah penduduk, kepadatan penduduk, tingkat kriminalitas dan sebagainya.[5]

2.2.3. Basis Data

Menurut sejarah sistem pemrosesan basis data terbentuk setelah masa sistem pemrosesan manal dan sistem pemrosesan berkas. Sistem pemrosesan manual (berbasis kertas) merupakan bentuk pemrosesan yang menggunakan dasar berupa setumpuk rekaman yang disimpan pad rak-rak berkas. Jika hal seperti itu dikehendaki, tidak ada cara lainselain dengan menuliskannya kembali. Tentu saja hal itu tidak praktis.

Pada saat awal pemrosesan sistem komputer, sekelompok rekaman disimpan pada sejumlah berkas secara terpisah. Sistem yang menggunakan pedekatan seperti ini biasa disebut sebagai sistem pemrosesan berkas. Sistem ini tentu saja memiliki kelebihan daripada sistem pemrosesan manual, yaitu dalam hal kecepatan dan keakuratannya. Kelemahannya, perancangan sistem masih didasarkan pada kebutuhan individu pengguna, bukan kebutuhan sejumlah pengguna. Setiap kali ada kebutuhan baru dari seorang pengguna, kebutuhan segera diterjemahkan ke program komputer. Alhasil, setiap program aplikasi menuliskan data tersendiri.

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain lain. Data dinyatakan dengan nilai (angka, deretan karekter, atau *symbol*).[6]

2.2.3.1. Bahasa Basis Data

Bahasa yang digunakan di dalam basis data antara lain:[7]

1. *Data Definition Leanguage* (DDL), merujuk pada kumpulan perintah yang dapat digunakan untuk mendefinisikan objek-objek basis data seperti membuat sebuah tabel basis data atau indeks primer atau sekunder.
2. *Data Manipulation Language* (DML), mengacu pada kumpulan perintah yang dapat digunakan melakukan manipulasi data, seperti penyimpanan data ke suatu tabel, kemudian mengubahnya dan menghapus atau hanya sekedar menampilkannya kembali DBML dibagi atas dua jenis:

a. Prosedural

   Prosedural menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkannya.

b. Non-prosedural

   Non-prosedural menuntut pengguna menentukan pengguna menentukan data apa yang diperlukan tetapi perlu menyebutkan cara mendapatkannya.

2.2.3.2. Sistem Basis Data

Sistem basis data adalah suatu sistem menyusun dan mengelola record menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil kepuasan.

Manajemen Sistem Basis Data (*Database Management System*/DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dalam jumlah besar. DBMS dapat menjadi alternative penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya. DBMS memiliki beberapa sifat-sifat antara lain: [7]

1. Mengolah file-file yang saling berhubungan.
2. Program yang disediakan meliputi fungsi untuk:
   a. Memanipulasi data yang telah ada dalam file.
   b. Mengorgansisasi dan mengontrol data dalam jumlah yang besar.
   c. Memasukan data atau *write* data.
   d. Melakukan *backup, recovery* dan loging terhadap data.
   e. Menyederhanakan dan memperluas pengamanan data.

2.2.4. Pemodelan Analisis

Model analisis merupakan representasi tejnis yang pertama dari sistem, tetapi saat ini ada yang mendominasi landasan pemodelan analisis. Analisis terstruktur adalah metode pemodelan klasik dan analisis berorientasi objek..

Dasar pertimbangan untuk mengembangkan kemampuan *object oriented* dari PHP adalah perkembangan aplikasi web sebagai sebuah *platform* yang terus meluas dengan cepat karena ruang lingkup aplikasi web semakin meluas sehingga aplikasi web yang dibangun menjadi semakin besar, rumit, dan kompleks. Aplikasi web telah diimplementasikan mulai dari tingkatan yang paling sederhana seperti online *news* hingga ke tingkatan *enterprise* seperti aplikasi *online banking*, *Supply Chain Management* (SCM), *Customer Relationship Management* (CRM), *Geographic Information System* (GIS), dan *Knowlwdge Management system*.

Dengan kemampuan untuk mengimplementasikan konsep *object oriented*, PHP akan lebih mudah untuk dapat berhubungan dan mengakses komponen-komponen yang dibuat dengan bahasa-bahasa *object oriented* lain seperti Java atau C++ dan bahkan Visual Basic.[8]

2.2.4.1. Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (*Object Oriented Programming*/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.[9]

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan objek yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman sistem yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement.

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu:

1. *Attribute*

   Atribut dari sebuah kelas adalah sistem global yang dimiliki sebuah kelas, Atribut dapat memiliki hak akses *private, public* maupun *protected*. Sebuah atribut yang dinyatakan sebagai *private* hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak

dapat mengakses atribut ini secara langsung.

Sebuah atribut yang dinyatakan sebagai *public* dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya. Sebuah atribut yang dinyatakan sebagai *protected* tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

Atribut juga biasa disebut *state/ properties/ field* merupakan bagian yang dimiliki oleh sebuah *class* yang tidak melakukan operasi, tetapi kepadanya dilakukan operasi sehingga dapat merubah nilai dari *attribute* tersebut.

2. *Method*

Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu c*lass*. Dengan kata lain, *method* pada sebuah kelas sistem sama dengan fungsi atau prosedur pada pemrograman sistem.

Pada sebuah *method* di dalam sebuah kelas juga memiliki izin akses seperti halnya atribut pada kelas, izin akses itu antara lain private, public dan protected yang memiliki arti sama pada izin akses atribut yang telah dibahas sebelumnya. Sebuah kelas boleh memiliki lebih dari satu method dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga sistem atau interpreter dapat mengenali *method* mana yang dipanggil.

Di dalam sebuah kelas, terdapat juga yang disebut sebagai *method* atau atribut statis yang memiliki kata kunci statis. Maksud dari statis di sini adalah *method* yang dapat diakses secara berbagi dengan semua objek lain tanpa harus membuat objek yang memiliki method statis tadi (tanpa proses *new*), tapi sebuah *method* statis mempunyai keterbatasan yaitu hanya dapat mengakses atribut atau *method* lain di dalam kelas yang membungkusnya yang juga bersifat statis. *Method* statis biasanya diimplementasikan untuk *method main.*

Didalam suatu *class*, *behavior* disimpan dalam bentuk *method*. *Method* merupakan serangkaian statemen / perintah (perintah = baris program)

dalam suatu *class* yang *menghandle* task tertentu. *Method* merupakan hal-hal yang bisa dilakukan oleh *object* dari suatu *class.*

Method didefinisikan pada *class* akan tetapi dipanggil melalui *object*. *Method* memiliki peranan diantaranya:

a. Merubah nilai dai suatu *object,*
b. Menerima informasi dari *object* lain, dan
c. Mengirim informasi ke objek lain untuk melakukan suatu *task* (cara obyek berkomunikasi dengan obyek lain adalah dengan menggunakan *method*).

3. *Class*

   Merupakan model yang berisi kumpulan *attribute* dan *method* dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *class* manusia memiliki *attribute* berat, tinggi, usia kemudian memiliki *method* makan, minum, tidur. *Method* dalam sebuah *class* dapat merubah *attribute* yang dimiliki oleh *class* tersebut. Sebuah *class* merupakan dasar dari modularitas dan struktur dalam pemrograman berorientasi *object*.

4. *Object*

   Merupakan perwujudan dari *class*, setiap *object* akan mempunyai attribute dan method yang dimiliki oleh *class*-nya, contohnya: amir, ahmad, yani merupakan *object* dari *class* manusia. Setiap *object* dapat berinteraksi dengan *object* lainnya meskipun berasal dari *class* yang berbeda.

   Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun *object* lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya. Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah *object.* Rumah, mobil, sepeda motor, meja, dan sistem merupakan contoh-contoh object yang ada di dunia nyata. *Object* adalah implementasi dari *class*. Secara sederhananya, dapat dikatakan terdiri dari sistem (atribut) dan *method.*

2.2.4.2. UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri piranti lunak dan pengembangan sistem[9].

2.2.4.3. Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:[9]

1. *Use Case* Diagram untuk memodelkan proses bisnis.
2. *Conceptual* Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence* Diagram untuk memodelkan pengiriman pesan (*message*) antar objects.
4. *Collaboration* Diagram untuk memodelkan interaksi antar objek.
5. *State* Diagram untuk memodelkan perilaku objects di dalam sistem.

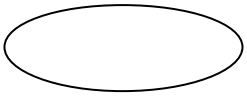
6. *Activity* Diagram untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system.*
7. *Class* Diagram untuk memodelkan struktur kelas.
8. *Object* Diagram untuk memodelkan struktur object.
9. *Component* Diagram untuk memodelkan komponen object.
10. *Deployment* Diagram untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi *object*, yaitu use case diagram, *sequence* diagram, collaboration diagram, dan *class* diagram.

2.2.4.4. *Use Case* Diagram

Use-case diagram merupakan pemodelan untuk kelakuan sistem informasi yang dibuat. *Use case* mendeskrisikan interaksi antara saru atau lebih aktor dengan sistem informasi yang akan dibangun. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan apa saja yang berhak menggunakan fungsi tersebut. Berikut adalah symbol simbol yang ada pada diahram *use case*[9].

Tabel 2.1 Simbol *Use Case* Diagram [9]

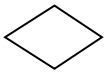
Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor / *Actor* 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
Asosiasi / *association*	Komunikasi antara aktor dan use case yang berpartisipasi ada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi / *extended* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan.
Generalisasi / *generalitation*	Hubungan generalisasi dan spesialisasi antara dua buah *use case* dimana fungsi yang satu

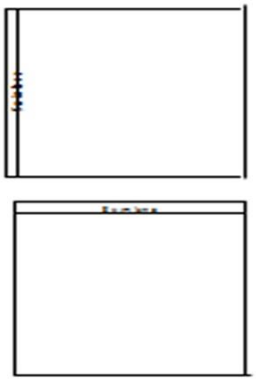
Simbol	Deskripsi
	adalah fungsi yang lebih umum dari lainya.
Menggunakan / *include* 	Relasi *use case* tambahan ke sebuah *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya.

2.2.4.5. *Activity* Diagram

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran aktivitas) dari sebuah sistem proses bisnis atau menu yang ada pada perangkat lunak. Berikit ini adalah simbol yang terdapat pada *activity diagram*:

Tabel 2.2 Simbol *Activity* Diagram [9]

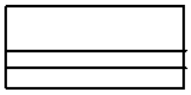
Simbol	Deskripsi
Status Awal 	Status awal aktivitas system sebuah diagram aktivitas memunyai status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / *decision* 	Asosiasi percabangandimana jika ada pilihan aktivitas lebih dari satu.
Percabangan / *join* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir.

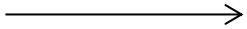
Swimline 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

2.2.4.6. *Class* Diagram

Class diagram dapat membantu dalam memvisualisasikan struktur kelaskelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* banyak memperhatikan hubungan antarkelas dan penjelasan detail tiap kelas dalam pemodelan desain (dalam *logical view*) dari suatu sistem. Selama proses analisa, *class diagram* memperhatikan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Berikut ini adalah simbol yang ada pada struktur *class diagram.*

Tabel 2.3 Simbol *Class* Diagram [9]

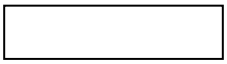
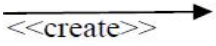
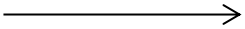
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka / *interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
Asosiasi / association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi 	Relasi antar kelas dengan makna genelasisai-spesialisasi (umum khusus)
Kebergantungan /	Kebergantungan antar kelas.

dependency 	
Agregasi / *agregation* 	Relasi kelas dengan semua bagian

2.2.4.7. *Sequence* Diagram

Sequence Diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use-case*. *Sequence* diagram memperlihatkan tahap demi tahap apa yang harus terjadi untuk menghasilkan suatu didalam *sequence* diagram.

Tabel 2.4 Simbol *Squence* Diagram [9]

Simbol	Deskripsi
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem.
Garis hidup	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi.
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi.
Pesan tipe create 	Menyatukan suatu objek membuat objek yang lain
Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data masukan ke objek lain.

Simbol	Deskripsi
Pesan tipe keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.

2.2.5. Pengertian Perangkat Lunak

Perangkat lunak atau *software* merupakan instruksi program yang memiliki fungsi-fungsi tertentu yang dapat menjalankan segala perintah sesuai dengan apa yang dibutuhkan oleh pengguna (*user*).

2.2.5.1. HTML

HTML (*Hyper Text Markup Language*) adalah Bahasa yang digunakan untuk menulis halaman *web. HTML* dirancang untuk digunakan tanpa tergantung dengan suatu *platform* tertentu (*platform independent*). Dokumen *HTML* adalah suatu dokumen teks biasa biasa, dan disebut sebagai *markup language* karena mengandung tanda-tanda (*tag)* tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen *HTML* merupakan pengembangan dari standar performatan dokumen teks yaitu *SGML* (*Standart Generalized Markup Language*). Sejak awal perkembangan sampai sekarang telah tersedia bermacam-macam versi *HTML*, yaitu HTML 1.0, HTML 2.0, *HTML* 3.0, *HTML* 4.0, dan *HTML* 5.0.[10]

2.2.5.2. PHP

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing'/Form Interpreter*. Dengan perilisan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

PHP adalah salah satu server side yang dirancang khusus untuk aplikasi web. PHP disisipkan diantara bahasa HTML dan karena bahasa server side, maka bahasa PHP akan dieksekusi di server, sehingga yang dikirimkan ke *browser* adalah hasil jadi dalam bentuk HTML, dan kode PHP tidak akan terlihat. PHP termasuk *open source* product. Jadi, dapat diubah *source code* dan mendistribusikanya secara bebas[11].

2.2.5.3. JavaScript

JavaScript merpupakan bahasa pemrograman web *client side*. Kalau HTML digunakan untuk membuat halaman web statis, maka JavaScript digunakan untuk membuat halaman web yang interaktif dan dinamis. JavaScript dapat digunakan untuk membuat aplikasi matematis, efek animasi sederhana, bahkan juga untuk membuat game. Hampir *browser* yang ada saat ini *support* JavaScript. Dokumen JavaScript dapat dibuat dengan *text editor biasa*, seperti: Notepad, wordpad, Notepad++, dan lain-lain, yaitu dengan menyimpannya ke dalam format *.js[11]

Sebagaimana HTML, sintaks di JavaScript juga menggunakan istilah TAG, tag yang digunakan adalah <script>. Terdapat 2 cara untuk membuka atau menyimpan dokumen JavaScript, yaitu[11]:

a. Disimpan dalam File sendiri

   Kumpulan perintah, fungsi JavaScript disimpan dalam *file* tersendiri (format*.js), tidak dicampur dengan kode HTML ataupun PHP.

b. Digabungkan ke dalam HTML

   Perintah atau kode JavaScript disisipkan langsung kedalam dokumen HTML. Cara seperti ini efektif, jika kode JavaScript tidak terlalu panjang dan tidak merupakan fungsi atau *method* berulang-ulang dipanggil.

2.2.5.4. *Cascading Style Sheet (CSS)*

Cascading *Style Sheet (CSS)* adalah aplikasi yang digunakan untuk membangun aplikasi berbasis web atau aplikasi *online* dan mempunyai kemampuan sebagai *code generator*. Aplikasi ini menyediakan banyak sekali fitur untuk membuat suatu aplikasi dengan cepat. CSS dapat mengendalikan ukuran

gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Ada dua sifat CSS yaitu internal dan eksternal. Jika internal yang dipilih, maka *skript* itu dimasukkan secara langsung ke halaman *website* yang akan didesain. Kalau halaman web yang lain akan didesain dengan model yang sama, maka *script CSS* itu harus dimasukkan lagi ke dalam halaman web yang lain itu. Sifat yang kedua adalah eksternal di mana *script CSS* dipisahkan dan diletakkan dalam berkas khusus[11].

2.2.5.5. MySQL

MySQL alah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan cepat, *multi user* serta menggunakan perintah standar *SQL (Structured Query Language).* MySQL memiliki beberapa kelebihan, antara lain:[12]

1. *Protability*

   MySQL dapat berjalan stabil pada berbagai sistem operasiseperti windows, Linux FreeBSD, Solaris dan lain-lain.

2. *Open Source*

   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga digunakan cuma-cuma.

3. *Multi User*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *integer, double, char, text, date* dan lain-lain.

6. *Command and Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam *query.*

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scability and Limits*

   MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *clients* menggunakan protokol TCP/IP, *Unix socket* (UNIX) atau *Named Pipes* (NT).

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan Bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients and Tools*

    MySQL dilengkapi dengan berbagai *tools* yang dapat digunakan untuk administrasi database dan pada setip *tool* yang dapat digunakan untuk administrasidatabase dan pada setiap *tool* yang ada disertaka petunjuk *online.*

13. *Struktur Tabe*

MySQL memiliki struktur tabel yang lebihfleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PosgreeSQL ataupun Oracle.