

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Tinjauan Teori Umum**

##### **2.1.1. Pengertian Sistem**

Sebelum membahas sistem informasi, maka ada baiknya kita membahas sistem. Adapun sistem dapat didefinisikan menjadi kumpulan/grup dari bagian atau komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu (Susanto, 2017, p. 22).

Pendapat lain menurut (Ermatita, 2016, p. 967), Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu.

Sedangkan menurut (Rahmad & Setiady, 2014, p. 1332), sistem adalah sekumpulan elemen atau subsistem yang saling bekerjasama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan.

##### **2.1.1.1. Karakteristik Sistem**

Menurut (Fatta, 2009, p. 5), untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya.

Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya:

1. Batasan (*Boundary*).  
Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*Environment*).  
Segala sesuatu diluar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukkan (*Input*).  
Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*Output*).  
Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*Component*).  
Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*), komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*Interface*).  
Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

#### 7. Penyimpanan (*storage*).

Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga diantara komponen tersebut bekerja dengan tinkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

### **2.1.2. Pengertian Informasi**

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta-fakta yang ada (Rahmad & Setiady, 2014, p. 1333).

Sedangkan menurut (Husda, 2012, p. 117) secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu objek nyata seperti tempat, benda dan orang yang betul-

betul ada dan terjadi. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

Adapun fungsi-fungsi informasi adalah sebagai berikut (Husda, 2012, p. 118):

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

#### **2.1.2.1. Kualitas Informasi**

Informasi yang berkualitas harus akurat, tepat pada waktunya dan relevan, (Husda, 2012, p. 118).

1. Akurat

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat Waktu

Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena

informasi meruoakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal bagi organisasi. Saat ini mahalny nilai informasi disebabkan harus cepatnya informasi itu didapat sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkannya.

### 3. Relevan

Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda-beda.

#### **2.1.3. Pengertian Sistem Informasi**

Sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi (Rahmad & Setiady, 2014, p. 1333).

Pendapat lain menurut (Fitriana, Kurniawan, & Sarah, 2016, p. 63), Sistem informasi adalah kombinasi dari pengguna teknologi, data, proses, dan teknologi informasi yang saling berhubungan untuk mengumpulkan, memproses, menyimpan dan memberikan informasi untuk mendukung sebuah organisasi.

Sedangkan menurut (Suryati & Purnama, 2012, p. 74), sistem informasi sendiri adalah suatu kumpulan elemen yang saling berhubungan satu sama lain yang dalam bentuk suatu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Sistem informasi dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari

beberapa komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

#### **2.1.3.1. Komponen Sistem Informasi**

Menurut (Hutahaean, 2015, p. 13), Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

1. Blok masukan (*input block*)

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

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk 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 (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu mengendalikan dari sistem secara keseluruhan.

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

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

6. Blok kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu di rancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

#### **2.1.4. Pengertian Pemasaran**

Menurut Fandy Tjiptono (2012) di dalam (Rahmawati & Mulyono, 2017, p. 105) pemasaran adalah penentuan apa yang akan dijual kepada konsumen berupa produk atau jasa dengan mendapatkan laba, melalui cara-cara, kondisi dan saluran distribusi tertentu, serta penciptaan dan pengolahan program untuk menghasilkan, melayani dan memperluas penjualan.

Pendapat lain menurut Philip Kotler (2015) di dalam (Rahmawati & Mulyono, 2017, p. 105) pemasaran sendiri adalah bagaimana mengenali dan mempertemukan manusia dengan kebutuhan sosialnya. Salah satu definisi terpendek dari pemasaran adalah mempertemukan kebutuhan yang menguntungkan.

Sedangkan menurut (Lupi & Nurdin, 2016, p. 21), pemasaran adalah kegiatan yang menghubungkan penjual dan pembeli sesuai dengan kebutuhan dan bukanlah sekedar perluasan penjualan tetapi keseluruhan bisnis dengan keuntungan sebagai imbalannya.

#### **2.1.5. Pengertian Sistem Informasi Pemasaran**

Menurut Kenneth P. Laudon (2009) di dalam (Rahmawati & Mulyono, 2017, p. 105) Sistem Informasi Pemasaran adalah suatu struktur, interaksi secara kompleks antara orang, mesin, dan prosedur untuk dapat menghasilkan alur informasi yang teratur, tepat dari sumber-sumber dalam dan luar perusahaan untuk dapat digunakan dalam pengambilan keputusan yang dapat dipertanggung jawabkan bagi pimpinan.

Sedangkan menurut (Cvijanovic, Mihailovic, & Nikolic, 2014, p. 196), Sistem informasi pemasaran pada dasarnya adalah sebuah sistem terkomputerisasi, yang dirancang untuk menyediakan alur informasi untuk menyediakan dan mendukung aktifitas pemasaran suatu organisasi. Sistem informasi pemasaran cenderung memuaskan kebutuhan kerjasama, analitis, dan bisnis. Mengenai kerjasama, sistem informasi pemasaran memungkinkan manager untuk berbagi informasi dan berkejasama secara virtual. Selain itu, sistem informasi pemasaran memungkinkan karyawan di bagian pemasaran untuk berkerjasama dengan pengguna dalam menampilkan produk dan kepuasannya. Sistem informasi pemasaran memiliki 4 komponen utama, yaitu tampilan antarmuka dengan konsumen, perangkat lunak, basis data, dan pendukung sistem.

### 2.1.6. Pengertian Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Penerapan *database* dalam sistem informasi disebut dengan *database system* (Rice Novita, 2016, p. 2).

Pendapat lain menurut (Iskandar & Haris Rangkuti, 2008, p. 126), Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan teknik-teknik formal dan manajemen basis data. Dari definisi diatas, maka dapat dikatakan bahwa basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

Sedangkan menurut (Rahmad & Setiady, 2014, p. 1333), Basis data adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap *file* yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*.

### 2.1.6.1. Kegunaan Basis Data

Penyusunan satu basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data (Rahmad & Setiady, 2014, p. 1334), yaitu:

1. Redudansi dan inkonsistensi data

Jika *file-file* dan program aplikasi diciptakan oleh programmer yang berbeda pada waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada *file-file* yang berbeda. Penyimpanan data yang berulang-ulang di beberapa file juga dapat mengakibatkan inkonsistensi (tidak konsisten).

2. Kesulitan Pengaksesan Data

Suatu saat dibutuhkan untuk mencetak data siapa saja, padahal belum tersedia program yang telah tertulis untuk mengeluarkan data tersebut maka kesulitan tersebut timbul, dan penyelesaiannya untuk itu adalah kearah Sistem Manajemen Basis Data yang mengambil data secara langsung dengan bahasa yang familian dan mudah digunakan.

3. Isolasi data untuk standarisasi

Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data, maka haruslah data dalam satu basis data dibuat satu format sehingga mudah membuat program aplikasinya.

4. Masalah keamanan atau *Security*

Setiap pemakai sistem basis data tidak semuanya diperbolehkan untuk mengakses semua data. Misalnya data mengenai gaji pegawai hanya boleh

dibuka oleh bagian keuangan dan personalia. Keamanan ini dapat diatur lewat program yang dibuat oleh pemrogram atau fasilitas keamanan dari operating sistem.

5. Masalah Integrasi (Kesatuan)

Basis Data berisi file yang saling berkaitan, masalah utama adalah bagaimana kaitan antara file tersebut terjadi. Meskipun diketahui bahwa file A berkaitan dengan file B, namun secara teknis maka ada file kunci yang mengaitkan kedua file tersebut.

6. Masalah data *independence* (kebebasan data)

Aplikasi yang dibuat dengan bahasa yang diciptakan dari Sistem Manajemen Basis Data, apapun yang terjadi pada struktur file, setiap kali hendak melihat data cukuplah dengan *utility USE*, hendak menambah data cukup dengan *APPEND*, ini berarti perintah-perintah dalam paket Sistem Manajemen Basis Data bebas terhadap basis data. Perubahan apapun dalam basis data, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

#### **2.1.6.2. Tingkatan Basis Data**

Menurut (Husda, 2012, p. 152), sebuah data pada basis data mempunyai beberapa tingkatan diantaranya adalah:

1. *Characters* (Karakter)

Karakter merupakan bagian data yang paling kecil, dapat berupa angka, huruf atau karakter khusus yang membentuk suatu *item* data atau *field*.

2. *Field* (Data Item)

Field merupakan representasi suatu atribut dari *record* yang sejenis yang menunjukkan suatu *item* dari data.

3. *Record*

*Record* adalah kumpulan elemen yang saling berkaitan yang menginformasikan tentang satu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.

4. *File*

*File* merupakan kumpulan dari *record-record* dalam basis data yang menggambarkan satu kesatuan data yang sejenis.

5. *Database* (Basis Data)

*Database* merupakan kumpulan dari beberapa *file* atau *table* yang saling berhubungan sehingga membentuk satu basis data.

### **2.1.7. Pengertian Bahasa Pemrograman**

Bahasa pemrograman adalah *software* bahasa komputer yang digunakan dengan cara merancang atau membuat program sesuai dengan struktur dan metode yang dimiliki oleh bahasa program itu sendiri. Komputer mengerjakan transformasi data berdasarkan kumpulan perintah program yang telah dibuat oleh program. Kumpulan perintah ini harus dimengerti oleh komputer, berstruktur

tertentu (*syntax*), dan bermakna. Bahasa pemrograman merupakan notasi untuk memberikan secara tepat program komputer. Berbeda dengan bahasa, misalkan Bahasa Indonesia dan Inggris yang merupakan bahasa alamiah (*natural language*), sintaksis dan semantik bahasa pemrograman komputer ditentukan secara jelas dan terstruktur, sehingga bahasa pemrograman juga disebut sebagai bahasa formal (*formal language*)(Husda & Wangdra, 2016, p. 42).

### **2.1.8. Pengertian Website**

Menurut M. Rudyanto Arief (2011) di dalam (Suhartanto, 2012, p. 2) *Website* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) didalamnya yang menggunakan protokol *HTTP (Hyper Text Transfer Protocol)* dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser. Beberapa jenis browser yang populer saat ini di antaranya : Internet Explorer yang diproduksi oleh Microsoft, Mozilla Firefox, Opera dan Safari yang diproduksi oleh Apple. *Browser* (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen web dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat didalam aplikasi *browser* yang biasa disebut *web engine*. Semua dokumen web ditampilkan dengan cara diterjemahkan. Sebuah *website* pada umumnya terdiri dari komponen-komponen yang ditulis dengan bahasa pemrograman HTML, CSS, JavaScript, PHP, dan lain-lain.

Sedangkan menurut (Sovia & Febio, 2011, p. 39), *Website* adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang

mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman web yang saling berhubungan. Hubungan antara satu halaman web dengan halaman web lainnya disebut dengan *hiperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

Dan menurut (Husda, 2012, p. 166), *website* atau situs web adalah setiap komputer atau tempat (*space*) dalam sebuah komputer yang terhubung dengan internet dan menjalankan fungsi dan proses sebagai server web yang berisi dokumen-dokumen dalam format HTML.

#### **2.1.9. Pengertian Software Development Life Cycle (SDLC)**

Dalam perancangan sebuah sistem informasi, tentunya diperlukan sebuah metode dalam proses perancangan. Salah satu metode yang bisa dipakai adalah metode SDLC. SDLC mengacu pada model dan proses yang digunakan untuk mengembangkan sistem perangkat lunak dan menguraikan proses, yaitu pengembang menerima perpindahan dari permasalahan ke solusi. Pengembangan rekayasa sistem informasi (*system development*) dan atau perangkat lunak (*software engineering*) dapat berarti menyusun sistem atau perangkat lunak yang benar-benar baru atau yang lebih sering terjadi menyempurnakan yang sebelumnya.

Tahap awal yaitu perencanaan (*planning*) adalah menyangkut studi tentang kebutuhan pengguna (*user specification*), studi-studi kelayakan (*feasibility study*) baik secara teknik maupun secara teknologi serta penjadwalan suatu proyek

sistem informasi atau perangkat lunak. Pada tahap ini pula, kita menggunakan perkakas (*tool*) yang penulis gunakan yaitu UML.

Tahap kedua, adalah tahap analisis (*analysis*), yaitu tahap dimana kita berusaha mengenai segenap permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan use case diagram lebih lanjut, mengenai komponen-komponen sistem atau perangkat lunak, objek-objek, hubungan antar objek dan sebagainya.

Tahap ketiga, adalah tahap perencanaan (*design*) dimana penulis mencoba mencari solusi dari permasalahan yang didapat dari tahap analisis.

Tahap keempat, adalah tahap implementasi dimana penulis mengimplementasikan perencanaan sistem ke situasi nyata yaitu dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (*pengkodean / coding*).

Tahap kelima, adalah tahap pengujian (*testing*), yang dapat digunakan untuk menentukan apakah sistem atau perangkat lunak yang dibuat sudah sesuai dengan kebutuhan pengguna atau belum, jika belum, proses selanjutnya adalah bersifat iteratif, yaitu kembali ke tahap-tahap sebelumnya. Dan tujuan dari pengujian itu sendiri adalah untuk menghilangkan atau meminimalisasi cacat program (*defect*) sehingga sistem yang dikembangkan benar-benar akan membantu para pengguna saat mereka melakukan aktivitas-aktivitasnya.

Tahap keenam, adalah tahap pemeliharaan (*maintenance*) atau perawatan dimana pada tahap ini dimulainya proses pengoperasian sistem dan jika

diperlukan melakukan perbaikan-perbaikan kecil. Kemudian jika waktu penggunaan sistem habis, maka akan masuk lagi pada tahap perencanaan.

Dari pendapat diatas dapat disimpulkan bahwa *System Development Life Cycle* (SDLC) adalah keseluruhan proses dalam membangun sistem melalui beberapa langkah. Ada beberapa model SDLC. Model yang cukup populer dan banyak digunakan adalah *waterfall*. Beberapa model lain SDLC misalnya *fountain, spiral, rapid, prototyping, incremental, build & fix*, dan *synchronize & stabilize* (Sofyan, Puspitorini, & Yulianto, 2016, p. 1).

Model *Waterfall* atau yang sering disebut juga disebut *classic life cycle* atau *model sequential linier* terdiri dari pendefinisian masalah, studi kelayakan, rancangan sistem, penerapan, pemeliharaan dan evaluasi (Oryza, Aziz, & Purnama, 2012, p. 24).

## **2.2. Tinjauan Teori Khusus**

### **2.2.1. Pengertian PHP**

PHP singkatan dari *PHP HyperText Processor* yang digunakan di dalam bahasa *script server side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan web dapat dibuat dinamis sehingga *maintenance* situs web tersebut menjadi lebih mudah dan efisien. PHP merupakan *software Open Source* yang disebar dan dilisensikan secara gratis serta dapat di unduh secara bebas dari situs resminya <http://www.php.net>. PHP ditulis dengan menggunakan bahasa *C* (Suhartanto, 2012, p. 2).

Sedangkan Menurut Agus Saputra di dalam (Rahmawati & Mulyono, 2017, p. 106) PHP atau yang memiliki kepanjangan *PHP Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, web akan sangat mudah di *maintenance*. Linux.PHP juga dibangun sebagai modul pada *web server apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

Menurut (Hariyanto, 2017, p. 13) ada beberapa kelebihan PHP dibandingkan bahasa pemrograman *web* lainnya, diantaranya adalah:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana, dari mulai *Apache, IIS, Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara

secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

Selain itu ada banyak kelebihan lain dari *PHP* ini, diantaranya:

1. Mampu terkoneksi dengan banyak pilihan *database* seperti *MySQL*, *ORACLE*, *PostgresSQL*, *dBase*, *IBM*, *DB2*, *ODBC*, *Ms. Access*, *SQL Lite*, dan *database* lainnya.
2. Gratis, karena *PHP* bersifat *open source*, sehingga kita menggunakannya kapan saja dan dimana saja tanpa harus bayar.
3. Komponen *PHP* sangat handal dan ringan. Ibaratnya tenaga sekuat gajah, tetapi konsumsi *BBM* nya sedikit semut. Perintah-perintah dalam *PHP* sangat mumpuni dan dapat dijalankan tanpa membenahi *server* terlalu berat dan cepat.
4. Dari banyak hal di atas, yang paling penting *PHP* itu ternyata adalah bahasa pemrograman yang paling banyak digunakan dalam pengembangan aplikasi-aplikasi berbasis *internet*.

Mengacu pada data survei yang dilakukan per tanggal 1 Juli 2016, *PHP* dinobatkan sebagai *most popular server-side programming languages*. Artinya *PHP* adalah bahasa pemrograman *server-side* paling populer yang digunakan lebih dari 82% *website* yang ada.

### **2.2.2. Pengertian HTML**

Menurut M. Rudyanto Arief di dalam (Suhartanto, 2012, p. 2) *HTML* atau *HyperText Markup Language* merupakan salah satu format yang digunakan dalam

pembuatan dokumen dan aplikasi yang berjalan di halaman web. Halaman ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan pada *web browser*.

Menurut (Husda, 2012, p. 165), HTML merupakan bahasa program yang digunakan untuk menulis format dokumen yang dapat diakses dalam web. Dengan menggunakan format HTML setiap dokumen yang dibuat dapat dibaca oleh semua sistem operasi dan tipe komputer yang ada di dunia. Disamping itu, kekuatan utama file HTML adalah adanya fungsi *hypertext link (hyperlink)*.

Dan menurut (Harison & Syarif, 2016, p. 43), *HyperText Markup Language (HTML)* adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan ke dalam format ASCII normal sehingga menjadi *home page* dengan perintah-perintah HTML.

### **2.2.3. Pengertian CSS**

CSS (*Cascading Style Sheet*) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga lebih terstruktur dan seragam. Pada umumnya CSS dipakai sebagai memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh teks, warna tabel, ukuran *border*, warna *hyperlink*, warna *mouse over*, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya (Muari, 2015, p. 33).

Menurut (Agustian, 2016, p. 78), CSS adalah kependekan dari Cascading Style Sheet. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996.

Sedangkan menurut (Maudi, Nugraha, & Sasmito, 2014, p. 103), CSS merupakan singkatan dari *Cascading Style Sheet*. CSS biasa digunakan dalam dokumen HTML untuk menciptakan suatu style yang dipakai untuk mengatur penampilan elemen HTML. Dengan menggunakan *style*, suatu elemen dapat diformat dengan fitur yang jauh lebih kaya daripada yang disediakan oleh elemen HTML itu sendiri. Sebagai contoh, pengaturan seperti warna tulisan bisa ditangani melalui *style* tanpa melibatkan tag HTML yang berfungsi untuk mengatur warna.

#### **2.2.4. Pengertian *Javascript***

*JavaScript* adalah bahasa pemrograman yang digunakan untuk membuat interaksi atau menambah fitur web dinamis kedalam sebuah web (Muari, 2015, p. 33).

Menurut (Handayaningsih & Pujiyono, 2010, p. 199) *JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi-objek dan lintas *platform*. *JavaScript*

tidak dapat berjalan dengan baik sebagai bahasa mandiri, melainkan dicancang untuk ditanamkan pada produk.

Sedangkan menurut (Maudi et al., 2014, p. 33), *JavaScript* merupakan bahasa pemrograman web *client side*. Kalau HTML digunakan untuk membuat halaman web statis, maka *JavaScript* digunakan untuk membuat halaman web yang interaktif dan dinamis. Karena sebagai bahasa pemrograman, *JavaScript* dapat digunakan untuk membuat aplikasi matematis, efek animasi sederhana, bahkan juga untuk membuat *game*. Hampir *browser* yang ada saat ini sudah *support JavaScript*. Dokumen *JavaScript* dapat dibuat dengan *text editor* biasa, seperti: Notepad, Wordpad, Notepad++, dll, yaitu dengan menyimpannya kedalam format \*.js.

Dan menurut (Yatini, 2014, p. 2), *JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. *JavaScript* digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. *JavaScript* dapat merespon perintah *user* dengan cepat dan menjadikan halaman web menjadi responsif. *JavaScript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi.

#### **2.2.5. Pengertian MySQL**

Menurut M. Rudyanto Arief di dalam (Suhartanto, 2012, p. 3) MySQL adalah sebuah sistem basis data yang dikembangkan oleh sebuah perusahaan

Swedia bernama MySQL AB yang pada saat itu bernama TcX DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. TcX merupakan perusahaan pengembang *software* dan konsultan *database*. Saat ini MySQL sudah diakuisisi oleh Oracle Corp. MySQL adalah salah satu jenis *database* server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang memiliki *database* sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* nya sehingga mudah untuk digunakan. MySQL juga bersifat *open source* dan *free* pada berbagai *platform* kecuali pada sistem operasi *windows* yang bersifat *shareware*. MySQL didistribusikan dengan lisensi *open source GPL (General Public License)* mulai versi 3.23, pada bulan Juni 2000.

Sedangkan menurut (Ramadhani, Anis, & Masruro, 2013, p. 480), MySQL adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*).

### 2.2.6. Pengertian Laravel

Laravel adalah sebuah *framework* web berbasis PHP yang *open source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di laravel terdapat *routing* yang menjembatani antara *request* dari *user* dan *controller*. Jadi *controller* tidak langsung menerima *request* tersebut (Yudanto, Tolle, & Brata, 2017, p. 630).

Laravel adalah salah satu *web application framework* yang bersifat *open source*. *Framework* ini berjalan diatas PHP 5 dan berbasis MVC (*Model View Controller*). Laravel pertama kali dirilis pada 22 Februari 2012, dan versi stabil terbaru adalah versi 4.2.11 yang dirilis pada 4 Oktober 2014 (Irfandi Susanto, Darwiyanto, & Gede Agung Ary, 2015, p. 7734).

Menurut (Irfandi Susanto et al., 2015, p. 7734) *Model View Controller* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Terdapat tiga komponen pembangun suatu MVC yaitu:

1. Model

*Model* adalah bagian yang berperan menghubungkan *controller* dengan *database*. Tugas dari *model* adalah melakukan manipulasi data ke *database* seperti CRUD (*create, read, update, delete*).

## 2. View

*View* mengatur bagaimana data akan ditampilkan kepada pengguna. Data yang didapat dari *model* akan diproses oleh *controller* kemudian oleh *View* ditampilkan kepada pengguna.

## 3. Controller

*Controller* merupakan bagian yang menjadi penghubung antara *model* dan *view*. *Controller* berfungsi memroses fungsi atau perintah dari pengguna kemudian menentukan bagaimana aplikasi dijalankan.

Framework Laravel ini pada dasarnya akan memudahkan *programmer* untuk mengembangkan aplikasi secara mandiri apalagi jika bekerja dalam sebuah tim. Laravel akan sangat membantu sekali dalam *team work* dengan menggunakan aturan-aturan yang berlaku dalam *framework* setiap *programmer* akan dibuat patuh terhadap aturan yang berlaku, sehingga pembacaan kode antar *programmer* akan lebih mudah (Triyono, 2016, p. 6).

### **2.2.7. Pengertian *Bootstrap***

*Bootstrap* adalah sebuah kerangka kerja *front-end* untuk membangun sebuah situs web yang responsif. *Bootstrap* dapat menjadi pilihan yang dianggap memenuhi kriteria, baik itu digunakan untuk aplikasi kerangka kerja, blog, atau aplikasi CMS lainnya, seperti halnya penggunaan pada umumnya. *Bootstrap* merupakan kombinasi dari HTML, CSS, dan JavaScript, memudahkan pembuatan sebuah situs yang kompeten tanpa menambahkan banyak kode. Dengan sistem *grid* standar, tata letak pengaturan kata dihadirkan secara bersamaan dengan

mudah, dan penataan tombol, *navs*, dan tabel membuat *markup* dasar terlihat sangat bagus dari awal. Beberapa grup atau lebih dari plugin *JavaScript* memudahkan anda untuk menambahkan elemen interaktif pada situs anda. (Spurlock, 2013, p. ix).

*Bootstrap* merupakan kerangka kerja *front-end* yang bersifat *open source* dikelola oleh Twitter untuk mengembangkan aplikasi web dan situs web responsif. Hal ini termasuk kode HTML, CSS, dan *JavaScript* untuk membangun komponen-komponen antarmuka pengguna, merupakan cara yang mudah dan cepat untuk mengembangkan sebuah antarmuka pengguna portabel pertama yang mumpuni. Sistem *grid Bootstrap* memungkinkan anda untuk menciptakan *grid* 12 kolom yang responsif, tata letak, dan komponen, termasuk di dalamnya standar *class* untuk pilihan tata letak yang mudah (*fixed width* dan *full width*). *Bootstrap* mempunyai beberapa grup komponen *reusable prestyled* dan *plugin jQuery* kustom, seperti tombol, kotak peringatan / *alerts*, *dropdown*, modal, *tooltip*, *pagination*, *carousal*, *badges*, ikon, dan masih banyak lagi (Banks & Porcello, 2016, p. 12).

*Bootstrap* merupakan situs web pengembangan kerangka kerja yang membantu para pengembang untuk membangun antarmuka situs web. Antarmuka yang pada awalnya dicetuskan oleh Mark Otto dan Jacob Thornton pada tahun 2011 untuk antarmuka Twitter, saat ini bersifat *open source* dan telah berkembang menjadi salah satu web pengembang antar muka yang populer hingga saat ini. Penggunaannya yang tersedia secara bebas untuk penggunaan yang bersifat pribadi, edukasional, dan untuk tujuan komersial, membuat *Bootstrap* menjadi

populer dengan sangat cepat. Saat ini, ribuan organisasi bergantung kepada *Bootstrap*, diantaranya termasuk NASA, Walmart, dan Bloomberg. Mengacu kepada BuiltWith.com, lebih dari 10% dari satu milyar situs web teratas dibangun dengan menggunakan *Bootstrap*. (Jakobus & Marah, 2016, p. 6).

### 2.2.8. Pengertian XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Priyanti, 2013, p. 56).<sup>3</sup>

Menurut (Maudi et al., 2014, p. 102), XAMPP merupakan paket PHP berbasis *open source*. Informasinya dapat diperoleh di *website* resminya: <http://www.apachefriends.com>. XAMPP membantu memudahkan dalam mengembangkan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket *software* berbeda kedalam satu paket. Adapun lisensi masing-masing paket *software* tersebut dapat ditemukan di direktori `\xampp\licence`. XAMPP menyediakan antar muka *control panel* tersendiri yang dapat digunakan untuk menjalankan semua *service* (paket *software* pendukung) yang telah terinstal. Pada

sistem operasi windows, *control panel* dapat diakses melalui menu *Start, Program, Apachefriends, xampp, control xampp server panel*.

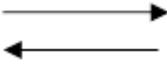
Sedangkan menurut (Palevi & Krisnawati, 2013, p. 4) XAMPP merupakan *software* bundel yang terdiri dari kumpulan paket program yang berhubungan dengan *database server, web server*, dan lain sebagainya. Didalam XAMPP terdapat Apache sebagai *server* web, MySQL sebagai *server* basis data, Filezilla sebagai *FTP server*, dan beberapa fitur tambahan seperti Mercury dan Tomcat.

### 2.2.9. Pengertian Aliran Sistem Informasi

Aliran Sistem Informasi (ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam system Adapun simbol-simbol yang dapat digunakan pada Aliran Sistem Informasi (ASI) ini adalah sebagai berikut, (Ismael, 2017, p. 149).

**Tabel 2. 1** Simbol-simbol Aliran Sistem Informasi

Simbol	Keterangan
	Proses Komputer
	Dokumen

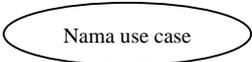
Simbol	Keterangan
	Penghubung
	Garis alur
	<i>Output</i> di monitor
	Manual Proses
	<i>File Storage</i>
	Penghubung antar halaman
	<i>Input</i> menggunakan <i>keyboard</i>
	<i>Input output</i> menggunakan disket

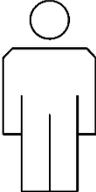
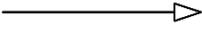
### 2.2.10. Pengertian UML

Menurut (Ariani Sukamto & Shalahuddin, 2011, p. 113) *Unified Modeling Language* merupakan salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. *UML* merupakan standar yang relatif terbuka di kontrol oleh *Object Management Group* (*OMG*), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. Dimana *OMG* dibentuk untuk membuat standar-standar yang mendukung interoperabilitas khususnya untuk sistem berorientasi objek. Ada banyak diagram *UML* (*Unified Modeling Language*), penulis akan membahas diagram yang digunakan dalam melakukan rancangan sistem pendukung keputusan penentu pajangan, sebagai berikut:

1. *Use Case Diagram* adalah diagram yang menggambarkan kebutuhan sistem dari sudut pandang pengguna, yang memperlihatkan hubungan-hubungan yang terjadi antara *actors* dengan *use case* dalam sistem.

**Tabel 2. 2** Simbol *Use Case*

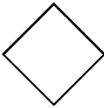
Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor,

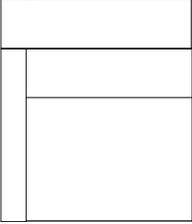
		biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i>
Aktor / <i>actor</i>  		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi / <i>association</i>  		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi / <i>extend</i>  		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
Generalisasi / <i>generalization</i>  	/	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan	/	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana

<p><i>include</i></p> <p>--      --&gt;</p>	<p><i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>
---	--

2. *Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi di dalam suatu *use case*.

**Tabel 2. 3** Simbol *Activity Diagram*

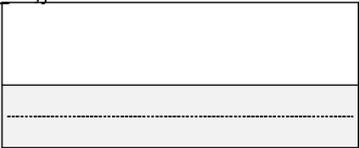
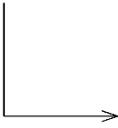
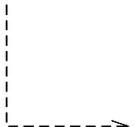
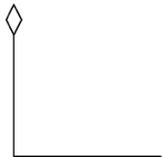
Simbol	Deskripsi
<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Percabangan / <i>Decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>
<p>Penggabungan / <i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.</p>
<p>Status akhir</p>	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>

	
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

3. *Class* adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode / fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti *containtment*, perwarisan, asosiasi, dan lain-lain.

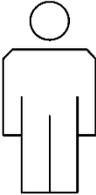
**Tabel 2. 4** Simbol *Class Diagram*

Simbol	Aktivitas
<p>Kelas</p> 	<p>Kelas pada struktur sistem.</p>

<p>Antarmuka / <i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Relasi antar kela dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas.</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Relasi antar kelas dengan makna semua-bagain (<i>whole-part</i>).</p>

4. *Sequence Diagram* menggambarkan interaksi antara objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan (Mahdiana, 2011, p. 38).

**Tabel 2. 5** Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek.

	
<p>Objek</p> 	Menyatakan objek yang beinteraksi pesan.
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan beinteraksi pesan.
<p>Pesan tipe <i>create</i></p> <p>&lt;&lt;create&gt;&gt;</p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<p>Pesan tipe <i>call</i></p> <p>l : nama_metode()</p> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarahkan pada objek yang dikirim.
<p>Pesan tipe <i>return</i></p>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan sesuatu kembalian ke objek

		tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i>  <<destroy>>  →		Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

### 2.3. Penelitian Terdahulu

No	Nama Peneliti	Judul	ISSN	Kesimpulan/Hasil
1	(Suryati & Purnama, 2012)	Pembangunan Sistem Informasi Pendataan Rakyat Miskin Untuk Program Beras Miskin (Raskin) Pada Desa Mantren Kecamatan Kebonagung Kabupaten Pacitan	1979-0330	Sistem informasi yang dibuat memberikan kemudahan proses pengelolaan dan pengolahan data mengenai masyarakat yang berhak mendapatkan beras miskin (raskin) serta mempermudah petugas dalam membuat laporan akan hal itu.
2	(Indah, 2013)	Pembuatan Sistem Informasi Penjualan Pada Toko Sehat Jaya Elektronik Pacitan	1979-9330	Pembuatan sistem informasi pada Toko Sehat Jaya Elektronik Pacitan dari sistem konvensional menjadi terkomputerisasi sudah sesuai dengan harapan dari pimpinan toko

				dan data yang tadinya tidak dapat tersimpan dengan baik secara konvensional sekarang sudah dapat tersimpan dengan baik.
3	(Astuti, 2013)	Sistem Informasi Penjualan Obat Pada Apotek Jati Farma Arjosari	1979-9330	Sistem informasi yang dibangun dapat membantu memperbaiki kesalahan yang kerap terjadi saat penulisan data dan dapat membantu asisten apoteker dalam mencari data.
4	(Latifah & Triyono, 2013)	Rancang Bangun Sistem Informasi Manajemen Pendonoran Darah Pada Udd PMI Kabupaten Pacitan	1979-9330	Sistem informasi yang dibangun dapat membantu pengolahan data pada UDD PMI Kabupaten Pacitan menjadi lebih mudah, akurat, efektif, dan efisien.
5	(Adeoti, Olawale, & Abolarinwa, 2016)	Impact of Marketing Information System on Product Performance in Nigerian Bottling Company Limited	1996902-3	Perusahaan yg mempunyai sistem informasi pemasaran yg efisien akan mempunyai produk yg bisa bersaing dan akan bertemu dengan pembeli yg potensial drpd organisasi yg tidak mempunyai sistem informasi pemasaran.
6	(Singla, 2015)	Designing Marketing	2395-6410	Sebuah sistem informasi

		Information System : An Overview		pemasaran adalah alat yang sangat berguna di tangan manajemen dan harus dapat menjawab kebutuhan pelanggan.
7	(Queiroz & Oliveira, 2014)	Benefits Of The Marketing Information System In The Clothing Retail Business	1807-1775	Hasil dr penelitian ini menunjukkan bahwa informasi dan proses sistematisnya mempunyai kedudukan yg penting dalam pengembangan operasi bisnis dan untuk menciptakan kesempatan bisnis yang baru.