

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka Umum

Pada subbab 2.1 ini peneliti ingin membahas teori-teori yang dikategorikan secara umum, yang artinya teori-teori umum yang akan dibahas dibawah ini, jika sebuah pernyataan umum dinyatakan fakta, maka pernyataan tersebut berlaku untuk semua atau bisa disebut dengan pernyataan yang universal.

2.1.1 Sistem

Menurut (Sihotang, 2019) Sistem merupakan sebuah jaringan kerja yang terdiri dari prosedur-prosedur atau proses yang saling terintegrasi dan berhubungan satu sama lain untuk melakukan sebuah kegiatan untuk mencapai sebuah tujuan tertentu.

Menurut (Fridayanthie & Charter, 2016) Sistem merupakan sebuah komponen yang terdiri dari bagian-bagian terpisah namun saling berhubungan secara harmonis demi mencapai suatu tujuan.

Dari kedua teori tersebut bisa disimpulkan bahwa Sistem adalah sekumpulan jaringan yang saling berhubungan atau terintegrasi dimana terdiri dari bagian bagian-bagian ataupun prosedur-prosedur yang bekerja sama untuk mencapai sebuah tujuan.

2.1.2 Informasi

Menurut (Astuti, 2013) Informasi merupakan sebuah hasil dari pengolahan data-data yang bisa digunakan sebagai dasar dari pengambilan keputusan bagi

penerima dimana penerima tersebut bisa merasakan secara langsung ataupun tidak langsung akibat yang akan datang.

Menurut (Sihotang, 2019) Informasi ada sekumpulan data yang diolah menjadi data yang berguna untuk penerima memproses informasi agar pada saat pengambilan keputusan, resiko dan profit bisa diukur secara efektif.

Dari kedua teori yang telah dibahas , bisa disimpulkan bahwa Informasi merupakan hasil olahan data-data ataupun laporan yang diolah jadi satu, dimana data-data tersebut bisa digunakan untuk penerima mengambil keputusan berdasarkan resiko dan profit yang sudah dihitung olehnya untuk mengetahui akibat yang akan datang dengan keputusan yang telah diambil oleh penerima.

2.1.3 Sistem Informasi

Menurut (Puspitasari, 2016) Sistem Informasi adalah sebuah sistem yang mempunyai kemampuan untuk mengumpulkan data dari berbagai sumber ataupun media dimana Sistem akan mengolah Informasi untuk menampilkan hasil dari prosedur tersebut.

Menurut (Wardani, 2013) Sistem Informasi merupakan sebuah proses pengumpulan data dari sumber-sumber yang tersedia dimana data tersebut akan diproses menjadi data yang layak digunakan untuk penerima yang kemudian penerima akan menganalisis data tersebut untuk mengetahui kalkulasi resiko dimana data tersebut akan disimpan olehnya dan data atau informasi yang telah dianalisis tersebut akan disebarkan ke perusahaan ataupun organisasi untuk mengetahui kelemahan dari perusahaan atau organisasi tersebut supaya ada kemajuan.

Adapula teori menurut (Husda, 2013) bahwa Sistem informasi memiliki tugas untuk menangkap dan mengelola data-data dari organisasi tersebut dimana data tersebut akan menghasilkan informasi yang efektif dan berguna demi mendukung pentingnya kegiatan ataupun sebuah pekerjaan.

Berdasarkan dari kedua teori tersebut, bisa disimpulkan bahwa Sistem Informasi merupakan sekumpulan elemen-elemen yang bekerja sama secara terintegrasi satu sama lain dengan mengumpulkan informasi atau data dari berbagai sumber media yang akan digunakan oleh penerima untuk mengambil keputusan yang akan dituju ke perusahaan atau organisasi dimana akibat yang akan datang sudah dikalkulasi risikonya.

2.1.4 Rekayasa Perangkat Lunak

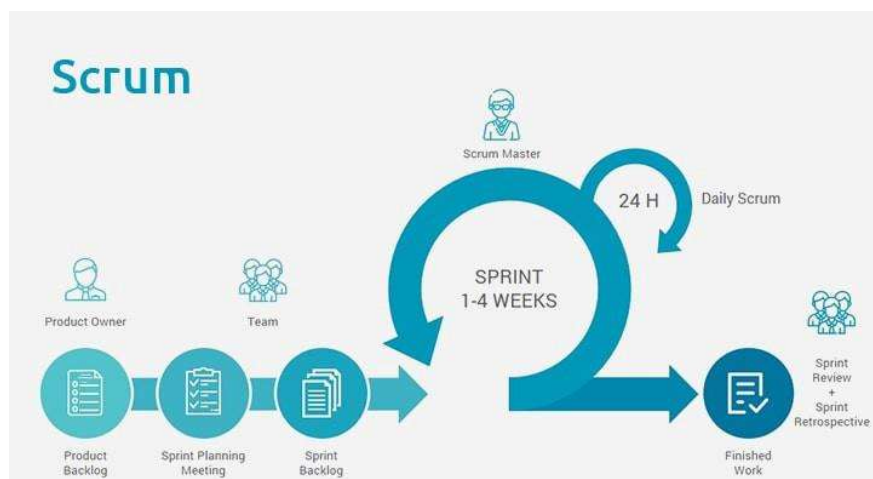
Menurut (Swara, Kom, & Pebriadi, 2016) Rekayasa Perangkat Lunak adalah proses pengembangan sekumpulan perintah-perintah pada mesin tersebut yang akan digunakan untuk mengolah informasi dimana didalam mesin tersebut sudah ada prosedur yang mudah dimengerti oleh mesin tersebut yang akan memproses informasi dimana data tersebut akan digunakan oleh pengguna.

Menurut (Yehendra & Yulianto, 2015) RPL atau Rekayasa Perangkat Lunak adalah sebuah kegiatan yang mendalami dalam bidang pengembangan perangkat lunak yang terdiri dari proses pembuatan aplikasi, pemeliharaan aplikasi yang telah dibuat, manajemen organisasi dan anggota dalam pengembangan perangkat lunak dan *training* pengguna (*User*) dalam menggunakan aplikasi yang telah dibuat.

Dari kedua teori yang telah disebut, bisa disimpulkan bahwa Rekayasa Perangkat Lunak adalah proses pendalaman dalam pengembangan aplikasi dimana perintah atau prosedur yang akan dibuat oleh pengembang tersebut akan diimplementasikan ke dalam mesin dimana mesin tersebut akan memproses informasi ke pengguna.

2.1.5 Metode *Scrum*

Metode *Scrum* menurut (Ford et al., 2019) merupakan sebuah metode yang menggunakan iterasi dan kerangka tambahan pada sebuah *project* yang akan dikembangkan. Siklus-Siklus kegiatan atau pekerjaan pada *Scrum* dimana aplikasi tersebut akan dikembangkan disebut dengan *Sprint* dimana kegiatan *Sprint* biasanya dilakukan dalam jangka waktu 1 bulan atau artinya kegiatan *Sprint* dilakukan tanpa berhenti.



Gambar 2. 1 Metode Scrum

(Sumber : Wahyudi, 2018)

Scrum menurut (Wahyudi, 2018) adalah salah satu metode pengembangan aplikasi dengan pengimplementasian proses *Agile Development*. *Scrum* mempunyai

perbedaan yang signifikan dikarenakan produk yang dihasilkan akan menyesuaikan dengan lingkungan seiring waktu proses pengembangan berlalu.

Scrum Method in Application Development menurut (Hadinata & Nasir, 2017) merupakan sebuah contoh metode pengembangan aplikasi dimana tiap kegiatan atau pekerjaan mengandung *Sprint* dan didorong oleh *Prioritized Product Backlog*. *Prioritized Product Backlog* memiliki fitur koreksi *bug/glitch* pada sistem yang dikembangkan dimana hal ini akan berkontribusi pada produk akhir nanti.

Dari ketiga teori yang telah dikemukakan diatas, bisa disimpulkan bahwa metode *Scrum* adalah salah satu metode dalam pengembangan aplikasi yang mengimplementasikan *Agile Development* dimana masing masing kegiatan ataupun pekerjaan dikerjakan tanpa berhenti (*Sprint*) kemudian meeting antara anggota akan dilakukan tiap hari untuk mengetahui kelemahan dan kebutuhan lain dari *client* supaya aplikasi bisa menyesuaikan dengan permintaan-Nya.

2.2 Tinjauan Pustaka Khusus

Di subbab 2.2 ini peneliti ingin membahas teori ataupun tinjauan pustaka yang dikelompokkan secara khusus, dimana arti tinjauan pustaka khusus ini berlaku dengan fakta-fakta yang berkaitan dengan individu tertentu ataupun variabel tertentu. Teori ini juga menjelaskan pernyataan berdasarkan fakta-fakta yang dikumpulkan dari berbagai narasumber tertentu.

2.2.1 Laravel Framework

Berdasarkan teori menurut (Sari, Kurniadi, & Irfan, 2018), *Framework* Laravel merupakan sebuah kerangka kerja dari bahasa pemrograman PHP (*Hypertext Preprocessor*) dengan *framework* laravel ini berada dibawah lisensi

MIT dimana Laravel ini biasanya digunakan untuk meningkatkan kualitas perangkat lunak.

Tetapi menurut (Somya & Nathanael, 2019) Laravel merupakan sebuah *framework* laman web yang dikembangkan oleh Taylor Otweel yang bersifat gratis (*Open Source*) dimana kegunaan utama dari Laravel ini untuk mengembangkan aplikasi web dengan menggunakan kerangka arsitektur berbentuk MVC (*Model-View-Controller*) dimana fitur-fitur seperti, otentikasi (*authentication*), *routing*, *session manager*, *caching* dan komponen lain yang telah disediakan oleh Laravel ini, kemudian Laravel juga mempunyai fitur migrasi database (*database migration*) dan *Unit Testing Support* yang terintergrasi agar memudahkan *developer* web dalam pengembangan aplikasi yang kompleks.

Berdasarkan dari kedua teori tersebut, bisa disimpulkan bahwa *framework* Laravel merupakan sebuah kerangka kerja yang dikembangkan oleh Taylor Otweel di MIT dengan basis bahasa pemograman PHP (*Hypertext Preprocessor*) yang bersifat *Open Source* dimana Laravel ini menggunakan kerangka arsitektur MVC (*Model-View-Controller*) dimana komponen pada Laravel sangat mudah untuk dipahami seperti fitur *authentication*, *session manager*, *routing*, dan *caching*, kemudian fitur *Unit Testing Support* yang telah terintergrasi untuk seorang pengembang laman web agar lebih mudah dalam mengembang aplikasi yang kompleks.

2.2.2 Hypertext Preprocessor (PHP)

PHP menurut (Yehendra & Yulianto, 2015) merupakan singkatan dari *Hypertext Preprocessor* yang merupakan sebuah bahasa pemograman umumnya

digunakan dalam pembuatan web dinamis dimana PHP adalah bahasa *script server-side* dalam pengembangan web dinamis yang disisipkan ke dalam dokumen HTML.

Menurut (Rini Sovia dan Jimmy Febio, 2011) PHP adalah *script* pemrograman berbasis web dinamis dimana *script* koding tersebut dibuat secara *on the fly* yang artinya dokumen HTML yang dikeluarkan oleh aplikasi tersebut tidak dibuat dengan dokumen HTML melainkan editor teks ataupun editor HTML.

```

1  <?php
2  require_once('dbconfig.php'); // file that stores our database connection variabl
3  require_once('functions.php'); // resize script
4  try {
5  if (!isset($_GET['id'])) {
6  throw new Exception('ID not specified');
7  }
8  $id = (int) $_GET['id'];
9  if ($id <= 0) {
10 throw new Exception('Invalid ID specified');
11 }
12
13 $query = sprintf('select * from PHOTOS where id = %d', $id);
14 $result = mysql_query($query, $db);
15 if (mysql_num_rows($result) == 0) {
16 throw new Exception('Image with specified ID not found');
17 }
18 $image = mysql_fetch_array($result);
19 }
20 catch (Exception $ex) {
21 header('HTTP/1.0 404 Not Found');
22 exit;
23 }
24 header('Content-type: ' . $image['mime_type']);
25
26 $width = 140;
27 $height = 140;
28
29 $folder = 'scripts';
30 $conflen=strlen($folder);

```

Gambar 2. 2 Contoh Script PHP

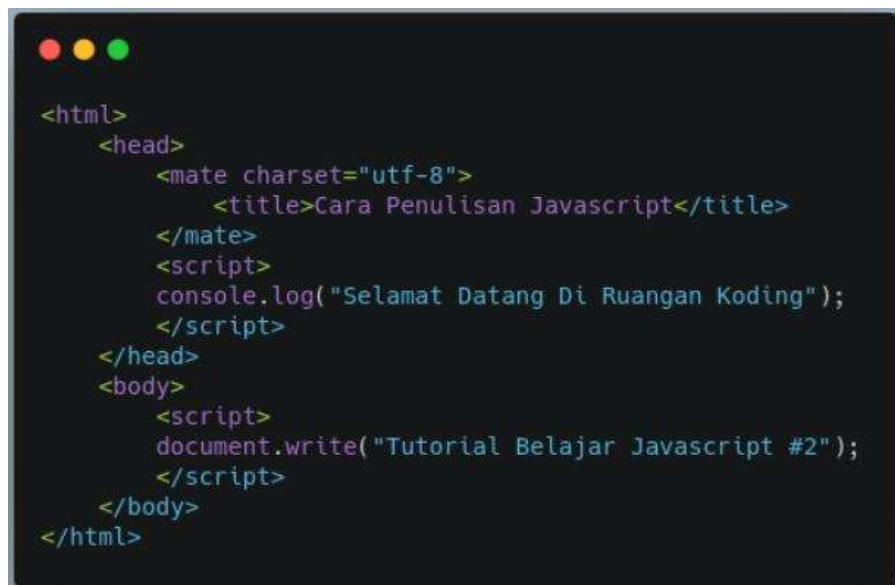
(Sumber : Rini Sovia dan Jimmy Febio, 2011)

Berdasarkan kedua teori mengenai PHP tersebut, bisa disimpulkan bahwa PHP atau *Hypertext Preprocessor* merupakan sebuah bahasa pemrograman dalam pengembangan web dinamis dimana *script* PHP disisipkan ke dalam dokumen HTML namun *script* tersebut bersifat *on the fly* yang artinya dokumen HTML yang diolah oleh aplikasi tersebut tidak diproses dengan dokumen HTML tetapi diproses ke dalam editor teks ataupun editor HTML.

2.2.3 *Hypertext Markup Language (HTML)*

Menurut (Recky T. Djaelangara, Rizal Sengkey, 2015) HTML adalah kepanjangan dari *Hypertext Markup Language*. HTML merupakan sebuah bahasa markah yang biasanya digunakan dalam pengembangan web statis maupun dinamis karena HTML mempunyai format hiperteks yang sederhana dimana format tersebut bisa ditulis dengan format ASCII.

Konsep dasar pada HTML dimana pada umumnya dikenal dengan *web scripting* dikarenakan koding hasil olahan HTML tidak akan jadi berkas yang *executable* namun akan jadi berkas yang di *interpreted*. HTML juga merupakan dasar dari pengembangan web statis ataupun dinamis dimana HTML mampu menampilkan informasi dalam bentuk multimedia seperti teks, grafik serta penghubung antar tampilan sebuah halaman web (*hyperlink*) (Rini Sovia dan Jimmy Febio, 2011)

A screenshot of a code editor window with a dark background and light-colored text. The code is an HTML document structure with JavaScript snippets. The code is as follows:

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Cara Penulisan Javascript</title>
  </meta>
  <script>
    console.log("Selamat Datang Di Ruangan Koding");
  </script>
</head>
<body>
  <script>
    document.write("Tutorial Belajar Javascript #2");
  </script>
</body>
</html>
```

Gambar 2. 3 Contoh Script HTML

(**Sumber** : Rini Sovia dan Jimmy Febio, 2011)

Berdasarkan teori dari kedua penulis tersebut, bisa disimpulkan bahwa HTML atau *Hypertext Markup Language* adalah bahasa pemrograman yang umumnya digunakan dalam pengembangan web dinamis maupun statis dimana HTML memiliki format hiperteks dengan menggunakan bahasa ASCII tetapi HTML tidak mampu diolah menjadi berkas yang *executable* namun berkas yang di *interpreted*.

2.2.4 Database

Menurut (Yehendra & Yulianto, 2015) Database merupakan sebuah kumpulan-kumpulan data yang sudah terorganisir dimana data tersebut akan digunakan ke sistem yang sudah terkomputerisasi.

Menurut (Recky T. Djaelangkara, Rizal Sengkey, 2015) Database atau dalam Bahasa Indonesia Basis Data adalah tempat dimana sekumpulan data-data ditampung di dalam satu wadah kemudian data tersebut akan diorganisir kembali sehingga informasi-informasi yang tertera pada data tersebut dapat dijelajah untuk menyusun informasi dalam berbagai bentuk sesuai dengan kebutuhan perusahaan tersebut.

Menurut (Puspitasari, 2016) Database merupakan sebuah kumpulan data-data yang sudah saling terintegrasi secara logikal dan deskripsi yang telah dirancang dan diorganisir supaya informasi yang dibutuhkan suatu organisasi atau perusahaan dipenuhi.

Setelah dijelaskan teori dan konsep Database atau Basis Data dari ketiga penulis tersebut, bisa disimpulkan bahwa Database merupakan sebuah kumpulan

data-data yang sudah saling berhubungan secara logikal dan data-data tersebut ditampung ke dalam satu wadah dimana data-data tersebut akan diolah untuk menyusun informasi yang akan digunakan sesuai dengan kebutuhan perusahaan atau organisasi tersebut.

2.2.5 SQL

SQL atau *Structured Query Language* menurut (Rini Sovia dan Jimmy Febio, 2011) adalah bahasa *Query* terstruktur yang digunakan untuk memanggil database.

SQL sendiri menurut (Pamungkas, 2018) merupakan sekumpulan bahasa yang terstruktur yang umumnya digunakan untuk mengakses ke database.

SQL didefinisikan menjadi 3 konsep dasar yaitu :

1. DDL (*Data Definition Language*)

DDL merupakan sekumpulan perintah SQL yang menggambarkan desain dari database tersebut, selain itu DDL juga digunakan untuk membuat (*CREATE*), merubah (*ALTER*) ataupun menghapus (*DROP*).

Contoh-Contoh perintah DDL pada SQL :

- a. *CREATE db_mahasiswa* → Digunakan untuk membuat database baru
- b. *ALTER TABLE tb_mahasiswa ADD npm [FIRST] atau [AFTER]* → Digunakan untuk menambah kolom pada sebuah table, *FIRST* jika anda ingin menempatkan kolom pada urutan pertama, *AFTER* jika anda ingin menempatkan kolom sesudah kolom yang diinginkan
- c. *DROP db_mahasiswa* → Digunakan untuk menghapus database tersebut dan struktur-struktur table-Nya

2. DML (*Data Manipulation Language*)

DML merupakan kumpulan perintah *Query* SQL dimana umumnya digunakan untuk proses pengolahan isi data seperti memasukkan data (*INSERT*), merubah data (*UPDATE*) dan menghapus isi data (*DELETE*).

Contoh-Contoh perintah DML pada SQL :

- a. *SELECT * FROM tbl_mahasiswa* → Digunakan untuk menampilkan seluruh data pada tabel mahasiswa
- b. *INSERT INTO tbl_mahasiswa (npm, nama_mhs, jenis_kelamin, gol_darah, jurusan, fakultas) VALUES ('000001', 'Erik Sebastian', 'Laki-Laki', 'B', 'Sistem Informasi', 'Teknik dan Komputer')* → Digunakan untuk menginput data ke dalam tabel mahasiswa
- c. *UPDATE tbl_mahasiswa SET gol_darah = "A" WHERE npm = "000001"* → Digunakan untuk mengupdate golongan darah ke 'A' dengan kondisi NPM 000001
- d. *DELETE FROM tbl_mahasiswa WHERE npm = "000001"* → Digunakan untuk menghapus seluruh data pada tabel mahasiswa dengan kondisi NPM 000001

3. DCL (*Data Control Language*)

DCL merupakan bahasa *Query* SQL dimana fungsi utamanya memberikan akses (*GRANT*) ataupun menghapus akses user kepada database (*REVOKE*)

Contoh perintah *Query* DCL sebagai berikut :

- a. *GRANT SELECT ON db_mahasiswa TO user_admin* → Digunakan untuk memberi Admin untuk mengakses ke database mahasiswa saja.

- b. *REVOKE INSERT ON db_mahasiswa FROM user_admin* → Digunakan untuk mencabut akses input data ke dalam database mahasiswa untuk user Admin

Berdasarkan teori yang telah dijelaskan diatas, bisa disimpulkan bahwa SQL merupakan bahasa *Query* yang berlogika dan terstruktur dimana fungsi utamanya yaitu sebagai bahasa perantara antara database dan pengguna. SQL juga mempunyai 3 konsep dasar dalam *Query*, yang pertama ; DDL atau *Data Definition Language* merupakan bahasa *query* pada SQL yang digunakan untuk menggambarkan desain dari database tersebut , yang kedua ; DML atau *Data Manipulation Language* merupakan bahasa query dimana fungsi utamanya adalah untuk memanipulasi data atau dalam arti lain, mengubah ataupun menambah data pada database tersebut, yang terakhir ; DCL atau *Data Control Language* merupakan bahasa *query* yang digunakan untuk mengontrol user yang menggunakan database tersebut.

2.2.6 Object Oriented Programming (OOP)

Pemrograman berbasis objek atau *Object Oriented Programming* (Retnoningsih, Shadiq, & Oscar, 2017) merupakan sebuah metode pendekatan bahasa pemrograman yang menggunakan *object dan class*. OOP juga member keuntungan kepada programmer seperti :

1. *Reusability*

Kode atau *Script* yang telah dibuat dapat didaur ulang kembali atau dapat digunakan kembali untuk *Form* selanjutnya ataupun lain-Nya.

2. *Extensibility*

Koding yang telah dibuat oleh *programmer* bisa sisipkan *method* yang baru ke dalam *script* yang telah dibuat tanpa harus membuat koding baru dari awal lagi.

3. *Maintanability*

Script yang telah dibuat secara berskala besar dapat diorganisir dengan teratur yang memungkinkan *error* pada *script* tersebut dapat teratasi dengan OOP karena metode pemograman OOP sudah menggunakan konsep modularitas.

Adapula teori menurut (Fridayanthie & Charter, 2016) bahwa OOP merupakan sebuah teknik pemograman yang sering digunakan oleh *programmer* sebagai pengganti teknik pemograman yang menggunakan metode prosedural. OOP juga merupakan sebuah teknik pendekatan dalam pengembangan software dimana struktur *software* tersebut berdasarkan pada interaksi objek dalam menyelesaikan suatu proses yang sedang berjalan.

Namun menurut (Cahyanto & Augta, 2010) Pemograman berbasis Objek atau OOP dikategorikan menjadi beberapa bagian yaitu :

1. Kelas (*Class*)

Kelas atau *Class* pada OOP merupakan sebuah kumpulan dari data-data ataupun fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.

2. Objek (*Objek*)

Objek pada pemograman OOP merupakan sebuah proses dimana sekumpulan data akan dibungkus menjadi satu yang akan disisipkan ke dalam sebuah program komputer. Objek juga merupakan sebuah konsep dasar dari

modularitas dan struktur-struktur yang tertera kedalam sebuah program pada komputer yang berbasis objek.

3. Abstraksi (*Abstract*)

Abstract pada pemograman OOP merupakan sebuah kemampuan pada program yang akan melewati aspek informasi yang akan diproses olehnya dimana program tersebut akan langsung proses ke informasi inti. Proses yang dilakukan oleh *Abstract* dimana setiap objek yang tertera pada sistem akan dianggap sebagai model dari “pelaku” abstrak yang dapat melakukan sebuah pekerjaan ataupun proses dimana proses selanjutnya adalah pembuatan laporan dan keadaanya kemudian komunikasi ke objek lainnya dalam sistem tanpa mengungkapkan bagaimana kelebihan abstrak tersebut diterapkan

4. Enkapsulasi (*Encapsulation*)

Enkapsulasi merupakan konsep Pemograman berbasis objek dimana proses ini memastikan semua pengguna pada sebuah objek tidak akan dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak layak, hanya beberapa metode yang akan diberikan oleh pengguna untuk diberi akses ke objek tersebut untuk mengakses keadaanya. Setiap objek yang ada pada program tersebut yang telah menggunakan enkapsulasi akan disebutkan bagaimana objek lainnya dapat berinteraksi dengannya.

5. Polimorfisme (*Polymorphism*)

Polimorfisme pada OOP merupakan sebuah program pengiriman pesan berbasis objek dan tidak menggantung sama pemanggilan subrutin.

6. Warisan (*Inheritance*)

Inheritance pada pemrograman berbasis objek merupakan sebuah metode dimana program-program enkapsulasi, polimorfisme ataupun abstraksi dapat di sebar luaskan ke program lainnya agar tidak terjadi pengimplementasian ulang perilaku dari objek atau bahasa tersebut dengan catatan bahwa tidak semua objek pada pemrograman OOP dapat diwariskan ke program yang lain.

Berdasarkan teori-teori yang telah dijelaskan bisa disimpulkan bahwa OOP atau Pemrograman Berbasis Objek, merupakan sebuah metode pendekatan bahasa pemrograman dimana *programmer* tidak perlu menggunakan proses yang prosedural namun bisa menggunakan OOP dimana OOP dikategorikan menjadi beberapa bagian yaitu yang pertama ; Kelas atau *Class* ; Objek atau *Objek* ; Abstrak ; Enkapsulasi ; Polimorfisme ataupun Warisan (*Inheritance*) dikarenakan OOP memiliki sifat dimana koding yang telah dibikin dapat didaur ulang untuk *script* koding form lainnya, kemudian OOP juga memiliki sifat untuk menambahkan koding yang telah jadi agar pembuatan koding yang sama pada form lainnya tidak perlu ada penulisan ulang koding dan yang terakhir OOP memiliki sifat dimana *script* berskala besar bisa diorganisir secara teratur untuk mengurangi *error* pada *script* tersebut.

2.2.7 *Cascading Style Sheet (CSS)*

Cascading Style Sheet menurut teori (Puspitasari, 2016) CSS merupakan sebuah bahasa dasar HTML dan XHTML yang digunakan untuk mengatur tata letak dokumen dan menu yang ada di halaman tersebut.

CSS menurut (Recky T. Djaelangkara, Rizal Sengkey, 2015) merupakan sekumpulan aturan yang digunakan untuk mengendalikan komponen pada sebuah halaman web agar terlihat lebih terstruktur dan seragam namun CSS bukan sebuah bahasa pemrograman.

```
<style>
#inputnyamasbro{
  border:1px solid black;
  width:200px;
  padding:5px;
  margin:2px;
  border-radius:3px;
  font-weight:bold;
  font-family:cursive;
}
#buttonLogin{
  border: 1px solid rgb(52, 200, 253);
  width: 212px;
  padding:5px;
  margin:2px;
  border-radius:3px;
  background:rgb(52, 200, 253);
  color:white;
  font-weight:bold;
  font-family:cursive;
}
</style>
```

Gambar 2. 4 Contoh Script CSS

(Sumber : Puspitasari, 2016)

Berdasarkan kedua teori diatas, bisa disimpulkan bahwa CSS atau *Cascading Style Sheet* merupakan sebuah kumpulan aturan yang berdasarkan dari HTML dimana fungsi utama CSS yaitu untuk mengatur komponen yang ada pada sebuah halaman agar terlihat lebih terstruktur dan seragam dengan catatan bahwa CSS bukan sebuah bahasa pemrograman.