

**ANALISIS IMPLEMENTASI XSS *CLEAN* UNTUK SISTEM INFORMASI  
MANAJEMEN BERBASIS *CODEIGNITER*  
(STUDI KASUS: CV. AZATA INTERIOR)**



**TUGAS AKHIR**

**Disusun dan Diajukan Oleh:**

**MUH. MUHTASAN  
D421 15 515**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2022**

**LEMBAR PENGESAHAN SKRIPSI**

**ANALISIS IMPLEMENTASI XSS CLEAN UNTUK SISTEM INFORMASI  
MANAJEMEN BERBASIS CODEIGNITER (STUDI KASUS: CV. AZATA  
INTERIOR)**

**Disusun dan diajukan oleh**

**MUH.MUHTASAN**

**D42115515**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 19 Oktober 2022 dan dinyatakan telah memenuhi syarat kelulusan.

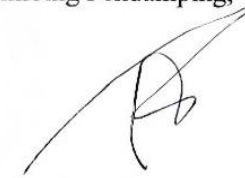
Menyetujui,

Pembimbing Utama,



Dr. Eng. Ir. Zulkifli Tahir, ST.,M.Sc.  
Nip. 198404032010121004

Pembimbing Pendamping,



Dr.Eng. Ady Wahyudi Paundu, ST.,MT.  
Nip. 197503132009121003

Ketua Program Studi,



Dr. Indrabayu, S.T., M.T., M.Bus.Sys.  
Nip. 19750716 200212 1 004

## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : MUH. MUHTASAN

NIM : D421 15 515

Program Studi : TEKNIK INFORMATIKA

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**ANALISIS IMPLEMENTASI XSS CLEAN UNTUK SISTEM  
INFORMASI MANAJEMEN BERBASIS CODEIGNITER  
(STUDI KASUS: CV. AZATA INTERIOR)**

Adalah karya tulis saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 19 Oktober 2022

Yang Menyatakan



MUH. MUHTASAN

## KATA PENGANTAR

*Bismillahirrahmanirrahim*

*Assalamualaikum Warahmatullahi Wabarakatuh*

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga tugas akhir yang berjudul “ANALISIS IMPLEMENTASI XSS *CLEAN* UNTUK SISTEM INFORMASI MANAJEMEN BERBASIS *CODEIGNITER*” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

1. Kedua Orang tua penulis, Bapak Nasrun S.Sos dan Ibu Alm. Nurhaedah yang selalu memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendidik penulis sejak kecil;
2. Bapak Dr-Eng. Zulkifli Tahir, S.T., M.Sc. selaku pembimbing I dan Bapak Dr. Eng. Ady Wahyudi Paundu, S.T., M.T. selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;

3. Bapak Dr. Indrabayu, S.T., M.T., M.Bus.Sys. selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
4. Teman-teman Hyperv15or atas dukungan dan semangat yang diberikan selama ini;
5. Segenap Staf dan Dosen Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis.
6. Orang-orang berpengaruh lainnya yang tanpa sadar telah menjadi inspirasi bagi penulis.

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

Wassalam

Makassar, 28 September 2022

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>Error! Bookmark not defined.</b>
<b>PERNYATAAN KEASLIAN.....</b>	<b>Error! Bookmark not defined.</b>
<b>ABSTRAK .....</b>	<b>1</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>DAFTAR TABEL.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
<b>1.1 Latar Belakang .....</b>	<b>2</b>
<b>1.2 Rumusan Masalah.....</b>	<b>4</b>
<b>1.3 Tujuan Penelitian .....</b>	<b>4</b>
<b>1.4. Manfaat Penelitian .....</b>	<b>4</b>
<b>1.5. Batasan Masalah Penelitian.....</b>	<b>5</b>
<b>1.6. Sistematika Penulisan.....</b>	<b>5</b>
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
<b>2.1. Landasan Teori.....</b>	<b>7</b>
<b>2.1.1. Keamanan Data.....</b>	<b>7</b>
<b>2.1.2. Framework .....</b>	<b>7</b>
<b>2.1.3. CodeIgniter.....</b>	<b>8</b>
<b>2.1.4. XSS .....</b>	<b>9</b>
<b>2.1.5. Sistem Informasi .....</b>	<b>10</b>

2.1.7. <i>Website</i> .....	12
2.1.8. <i>Database</i> .....	14
2.1.9. <i>MySQL</i> .....	15
2.1.10. <i>HTML</i> .....	20
2.1.11. <i>PHP</i> .....	21
2.1.12. <i>CSS</i> .....	22
2.1.13. <i>Black-Box</i> .....	26
2.1.14. <i>UML</i> .....	28
2.1.15. <i>Quality of Service (QoS)</i> .....	30
2.1.15. <i>Throughput</i> .....	31
2.1.16. <i>Packet lost</i> .....	31
2.2. <b>Penelitian Terdahulu</b> .....	32
<b>BAB III METODE PENELITIAN</b> .....	33
3.1. <b>Tahapan Penelitian</b> .....	33
3.2. <b>Waktu dan Lokasi Penelitian</b> .....	35
3.3. <b>Instrumen Penelitian</b> .....	35
3.4. <b>Analisis Kebutuhan</b> .....	35
3.5. <b>Metode Perancangan Sistem</b> .....	36
3.6. <b>Pembangunan Sistem</b> .....	36
3.7. <i>Use Case Diagram</i> .....	36
3.8. <i>Class Diagram</i> .....	37
3.9. <b>Perancangan <i>Interface</i></b> .....	38
3.10 <b>Hasil Analisis</b> .....	41

3.10.1 <i>Activity Diagram</i> .....	41
3.10.2 <i>Use Case Diagram</i> .....	42
3.10.3 <i>Class Diagram</i> .....	43
3.10.4 <i>Sequence Diagram</i> .....	44
3.10.5 <i>ERD (Entity Relationship Diagram)</i> .....	45
3.11. Hasil Implementasi.....	47
3.11.1. Implementasi Tabel.....	47
3.11.2 Hasil Pengujian Sistem .....	49
3.11.3. Implementasi <i>XSS Clean</i> .....	50
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	63
4.1. Uji Performa .....	63
4.1.1. <i>Throughput</i> .....	63
4.1.2. <i>Packet Lost</i> .....	65
4.1.3. <i>Latency/Delay</i> .....	67
<b>BAB V PENUTUP</b> .....	68
5.1 Kesimpulan .....	68
5.2 Saran.....	69
<b>DAFTAR PUSTAKA</b> .....	70



## DAFTAR TABEL

Tabel 2. 1 Packet Lost.....	31
Tabel 3. 1 Akun Pengguna.....	47
Tabel 3. 2Akun Pengguna.....	47
Tabel 3. 3 Data Gaji.....	48
Tabel 3. 4 Hasil Pengujian Menu.....	49

## DAFTAR GAMBAR

Gambar 3. 1 Rencana Penelitian .....	33
Gambar 3. 2 Metode Perancangan Sistem .....	36
Gambar 3. 3 Laman Registrasi .....	38
Gambar 3. 4. Tampilan Login .....	39
Gambar 3. 5 Tampilan Dashboard .....	39
Gambar 3. 6 Tampilan Data Pegawai .....	40
Gambar 3. 7 Tampilan Data Gaji .....	40
Gambar 3. 8 Activity Diagram.....	41
Gambar 3. 9 Use Case Diagram .....	42
Gambar 3. 10 Class Diagram .....	43
Gambar 3. 11 Sequence Diagram.....	44
Gambar 3. 12 Entity Relationship Diagram .....	46
Gambar 3. 13 Remove Invisible Characters (Non-Printable Characters) .....	50
Gambar 3. 14 URL Decode.....	50
Gambar 3. 15 Fungsi _urldecodespaces pada url_decode .....	51
Gambar 3. 16 Convert Character Entities to ASCII.....	51
Gambar 3. 17 Fungsi Convert attribute.....	51
Gambar 3. 18 Fungsi Decode Entity .....	52
Gambar 3. 19 Convert All Tabs to Spaces .....	52
Gambar 3. 20 Remove Strings That Are Never Allowed .....	53

Gambar 3. 21 Array data_never_allowed_str .....	53
Gambar 3. 22 Array data_never_allowed_regex .....	54
Gambar 3. 23 Make PHP Tags Safe.....	54
Gambar 3. 24 Company Any Exploded Words .....	55
Gambar 3. 25 Fungsi _compact_exploded_words .....	55
Gambar 3. 26 Remove Disallowed JavaScript in Links or Img Tags .....	56
Gambar 3. 27 . Fungsi _is_link_removal .....	56
Gambar 3. 28 Fungsi _is_img_removal .....	57
Gambar 3. 29 . Fungsi Attributes (Filters tag attributes for consistency and safety .....	58
Gambar 3. 30 Sanitize Naughty HTML Elements .....	59
Gambar 3. 31 List tags dan attributes.....	59
Gambar 3. 32 Fungsi yang digunakan.....	60
Gambar 3. 33 Sanitize Naughty Scripting Elements.....	61
Gambar 3. 34 Hasil Implementasi XSS Clean .....	61
Gambar 3. 35 Tampilan perbedaan data pada website.....	62
Gambar 3. 36 Tampilan perbedaan data pada Database .....	62

## ABSTRAK

XSS adalah singkatan dari *Cross-Site Scripting*, adalah kerentanan umum dalam aplikasi web, terjadi ketika kode web berbahaya dikirim atau dijalankan, biasanya dalam bentuk skrip, dari browser di komputer. Sistem informasi dalam pengelolaan data sangat perlu diterapkannya proses filter input data agar setiap inputan yang masuk tidak merusak halaman sistem yang digunakan serta data yang masuk akan tampil sesuai yang dimasukkan oleh pengguna. Cara seperti ini perlu digunakannya keamanan data seperti *XSS Clean Filtering* untuk mengecek setiap data inputan yang masuk pada sistem. Dengan *XSS Clean* keamanan data menggunakan *XSS Clean* dapat diterapkan guna meminimalisis kecurangan, penipuan, maupun pencurian data. Penerapan *XSS Clean* yang dimaksud akan diterapkan pada sistem informasi manajemen kepegawaian. Adanya penerapan keamanan data dengan *XSS Clean* diharapkan membantu menambahkan keamanan data pada sistem informasi manajemen kepegawaian yang akan dirancang dan dibangun dengan memanfaatkan *XSS Clean* dalam pembangunan sistem informasi manajemen kepegawaian.

**Kata kunci:** *XSS Clean*, Keamanan, Sistem.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknologi Informasi berbasis komputer mempunyai pengaruh yang cukup besar dalam masyarakat modern terutama bagi organisasi perusahaan. Perusahaan dihadapkan dalam lingkungan yang berubah-ubah dan sangat kompetitif. Untuk itu peran teknologi informasi sangatlah penting bagi perusahaan untuk membantu dalam perbaikan proses bisnis dan pengambilan keputusan (Elisabeth, 2019).

Keamanan data merupakan salah satu aspek penting dari sebuah sistem informasi. Sehingga masalah keamanan ini harus diperhatikan oleh para pemilik dan pengelola sistem informasi sebagai salah satu masalah yang dianggap penting dan harus dicari solusi pemecahan masalah (M.S Hasibuan, 2019). Perkembangan teknologi informasi pada saat ini membuat setiap pemilik dan pengelola sistem informasi harus dan wajib memikirkan bagaimana cara untuk melindungi keamanan sistem informasi yang dimilikinya agar terhindar dari berbagai resiko yang mungkin saja dapat menyebabkan kerugian. Resiko yang bisa saja menimbulkan dampak kerugian termasuk kedalam bagian dari tiga komponen yang memiliki kontribusi terhadap resiko yang diantaranya *Asset*, *Threat*, dan *Vulnerabilities* (M.S Hasibuan, 2016). Teknologi informasi seperti sekarang ini data menjadi sesuatu yang penting yang harus diperhatikan

oleh para pengolah data informasi, sehingga data tersebut tidak disalahgunakan oleh pihak-pihak tertentu yang tidak bertanggung jawab (Gunadhi, 2016).

XSS adalah singkatan dari *Cross-Site Scripting*, adalah kerentanan umum dalam aplikasi web, terjadi ketika kode web berbahaya dikirim atau dijalankan, biasanya dalam bentuk *script*, dari browser di komputer korban. Dengan eksekusi ini penyerang dapat mengambil informasi pribadi atau mencuri data pengguna (Melinda dkk, 2021).

Sistem informasi dalam pengelolaan data sangat perlu diterapkannya proses filter *input* data agar setiap inputan yang masuk tidak merusak halaman sistem yang digunakan serta data yang masuk akan tampil sesuai yang dimasukan oleh pengguna. Cara seperti ini perlu digunakannya keamanan data seperti *XSS Clean Filtering* untuk mengecek setiap data inputan yang masuk pada sistem. Dengan *XSS Clean* keamanan data menggunakan *XSS Clean* dapat diterapkan guna meminimalisis kecurangan, penipuan, maupun pencurian data. Penerapan *XSS Clean* yang dimaksud akan diterapkan pada sistem informasi manajemen kepegawaian. Adanya penerapan keamanan data dengan *XSS Clean* diharapkan membantu menambahkan keamanan data pada sistem informasi manajemen kepegawaian yang akan dirancang dan dibangun dengan memanfaatkan *XSS Clean* dalam pembangunan sistem informasi manajemen kepegawaian.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang dari penelitian diatas, maka rumusan masalah pada penelitian kali ini adalah

1. Bagaimana merancang sistem informasi manajemen kepegawaian berbasis *web* dengan menerapkan XSS *Clean* sebagai implementasi keamanan data terhadap XSS *injection attack*.
2. Bagaimana hasil dari pengujian QoS pada system manajemen kepegawaian berbasis *web*.

## **1.3 Tujuan Penelitian**

Pada penelitian yang dilakukan memiliki tujuan seperti berikut:

1. Mengetahui hasil dari perancangan sistem informasi manajemen kepegawaian berbasis *web* dengan menerapkan XSS *Clean* sebagai implementasi keamanan data terhadap XSS *injection attack*.
2. Mengetahui hasil dari pengujian QoS pada system manajemen kepegawaian pada *web*.

## **1.4. Manfaat Penelitian**

Manfaat dari tugas akhir ini adalah:

1. Bagi CV AZATA FURNITURE, dapat memudahkan CV AZATA FURNITURE dalam melakukan pengimputan data secara aman.
2. Bagi Peneliti, dapat mengetahui hasil dari pengujian QoS pada System manajemen kepegawaian pada *web*.

### **1.5. Batasan Masalah Penelitian**

Penelitian yang dilakukan adalah implementasi *XSS Clean* dan pengujian QoS pada sistem informasi manajemen kepegawaian memiliki ruang lingkup penelitian seperti berikut:

1. Penerapan keamanan data yang digunakan adalah *XSS Clean*.
2. Penerapan keamanan data diterapkan pada sistem informasi manajemen kepegawaian.
3. Sistem informasi berbasis *website*.

### **1.6. Sistematika Penulisan**

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

## **BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan sistem yang dianalisis, antara lain mengenai modul praktikum, metode yang digunakan, hingga bahasa yang digunakan dalam analisis tersebut.

## **BAB III METODOLOGI PENELITIAN**



Bab ini berisi tentang tahapan penelitian, waktu dan lokasi penelitian, instrumen penelitian, tahap persiapan, gambaran umum sistem, skenario pengujian dan analisis performa.

#### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil performa serta pembahasan yang disertai pemaparan hasil penelitian.

#### **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Landasan Teori**

##### **2.1.1. Keamanan Data**

Keamanan data dalam pengaksesan komputer adalah salah satu bentuk yang harus diperhatikan baik secara fisik maupun non fisik. Salah satu bentuk kejahatan pencurian data adalah merekam jejak keyboard komputer dengan bantuan hardware maupun *software*. Keamanan data atau *data security* adalah sebuah prosedur dengan dukungan dari regulasi dan teknologi untuk melindungi data dari perusakan data, modifikasi data, serta penyebaran data baik yang disengaja maupun tidak (Hasibuan, 2016).

##### **2.1.2. Framework**

*Framework* adalah kumpulan intruksi-intruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu. *Framework* adalah kerangka kerja untuk mengembangkan aplikasi berbasis *website* maupun *desktop*. Kerangka kerja disini sangat membantu *developer* dalam menuliskan sebuah dengan lebih terstruktur dan tersusun rapi. Kerangka kerja diciptakan untuk mempermudah kinerja dari programmer. Sehingga, seorang *programmer* tidak perlu untuk menuliskan kode secara berulang-ulang. Karena di dalamnya sendiri anda hanya

perlu menyusun komponen–komponen pemrograman saja (Sallaby dan Kanedi, 2020).

### **2.1.3. CodeIgniter**

*CodeIgniter* adalah Sebuah *Framework PHP* yang bersifat *open source* dan menggunakan metode *MVC (Model, View, Controller)* untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuatnya dari awal. Kerangka kerja ini pertama kali dibuat oleh *programmer* dari *EllisLab* yang kini menjadi proyek dari *the British Columbia Institute of Technology* sejak tahun 2014. *Framework CI* memiliki serangkaian fungsionalitas mumpuni yang akan mempercepat proses kerja pengembangan *website* dinamis (Sallaby dan Kanedi, 2020).

#### *A Kelebihan CodeIgniter*

##### 1. Sederhana

Banyak *programmer* yang menyadari apa itu *CodeIgniter* sebagai platform *Framework* sederhana dan cara penggunaan yang mudah. Disamping itu, sebagai *Framework open source*, *Framework CI* memungkinkan pengguna (*programmer*) untuk bisa menyesuaikan, mengatur, mengkonfigurasi, dan mengintegrasikan fitur secara mudah.

##### 2. Konfigurasi Sederhana

Sehubungan dengan poin kelebihan *Framework CI* sebelumnya, *syntax PHP XSS Clean* adalah bagian yang didasarkan pada

perkembangan linier dan struktur folder file yang mudah digunakan. Jadi, kode yang dimiliki platform ini lebih sederhana dan mudah diubah sesuai kebutuhan pengembangan web. Sehingga memudahkan proses konfigurasi sistem untuk mendapatkan performa fitur aplikasi web terbaik.

### 3. Fungsionalitas yang Menguntungkan

Pengguna mengetahui apa itu *CodeIgniter* memiliki fungsi yang sedikit berbeda dengan *Framework* lainnya, tapi ia lebih bermanfaat. *Framework* CI menyediakan fungsi migrasi yang dapat dilakukan dari satu *server Database* ke *server Database* lain.

#### 2.1.4. XSS

*Cross-site scripting* (juga dikenal sebagai XSS) adalah kerentanan keamanan *web* yang memungkinkan penyerang membahayakan interaksi yang dimiliki pengguna dengan Aplikasi yang rentan. Memungkinkan penyerang untuk menghindari kebijakan asal yang sama, yang dirancang untuk memisahkan Website yang berbeda satu sama lain. XSS adalah singkatan dari *Cross-Site Scripting*, adalah kerentanan umum dalam aplikasi web, terjadi ketika kode web berbahaya dikirim atau dijalankan, biasanya dalam bentuk skrip, dari *browser* di komputer korban. Dengan eksekusi ini penyerang dapat mengambil informasi pribadi atau mencuri data pengguna (Lisa, 2021).

##### A. Jenis-jenis XSS

1. *Reflected XSS (Cross-Site Scripting)*

Adalah tempat skrip berbahaya berasal dari permintaan *HTTP* saat ini.

2. *Stored XSS (Cross-Site Scripting)*

Yaitu di mana skrip berbahaya berasal dari Database Website.

B. Tujuan XSS (*Cross-Site Scripting*)

- Untuk menyamar sebagai atau menyamar sebagai pengguna korban.
- Melakukan tindakan apa pun yang dapat dilakukan pengguna.
- Untuk membaca Data apa pun yang dapat diakses pengguna.
- Menangkap kredensial *login* pengguna.
- Melakukan kerusakan *virtual* situs *web*.
- Menyuntikkan fungsionalitas *trojan* ke situs *web*.

### **2.1.5. Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan–laporan yang diperlukan oleh pihak luar tertentu. Sistem Informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang, fasilitas, teknologi, media, prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian internal

dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengembalian keputusan yang cerdas. Sistem Informasi adalah suatu sistem yang terorganisir yang berfungsi untuk mengolah informasi yang bermanfaat yang mempunyai tujuan tertentu dan informasi yang akan disampaikan dapat diterima dengan baik oleh penerima sehingga tujuan tersebut dapat tercapai (Sallaby dan Kanedi, 2015).

#### **2.1.6. *Computer Based Information System (CBIS)***

*Computer Based Information System (CBIS)* atau yang dalam Bahasa Indonesia disebut juga Sistem Informasi Berbasis Komputer merupakan sistem pengolah data menjadi sebuah informasi yang berkualitas dan dipergunakan untuk suatu alat bantu pengambilan keputusan. Sistem Informasi yang akurat dan efektif. Secara teori, penerapan sebuah Sistem Informasi memang tidak harus menggunakan komputer dalam kegiatannya. Tetapi pada prakteknya tidak mungkin sistem informasi yang sangat kompleks itu dapat berjalan dengan baik jika tanpa adanya komputer. Sistem Informasi merupakan sistem pembangkit informasi. Dengan integrasi yang dimiliki antar subsistemnya, sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya (Sutopo dkk, 2016).

Sistem informasi merupakan gabungan dari 4 (empat) bagian utama, yaitu: *Software, Hardware, Infrastruktur, dan Sumber Daya Manusia* yang terlatih untuk menciptakan sebuah sistem yang dapat mengelola data menjadi informasi yang bermanfaat (Juliany dkk, 2018).

### 2.1.7. Website

*Web* merupakan terobosan baru sebagai teknologi sistem informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di internet, *Web* cepat sekali populer di lingkungan pengguna internet, karena kemudahan yang diberikan kepada pengguna internet untuk melakukan penelusuran, penjelajahan, dan pencarian informasi (Sutopo dkk, 2016).

Secara makna sebuah *website* adalah sekumpulan halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan sebuah komponen yang terdiri dari teks, gambar, suara animasi sehingga menjadi media informasi yang menarik untuk dikunjungi oleh orang lain maka dari makna itu, bisa kita fahami bahwa definisi *website* secara sederhana adalah informasi apa saja yang bisa diakses dengan menggunakan koneksi jaringan internet. Secara umum, situs web digolongkan menjadi 3 jenis yaitu: *Website Statis*, *Website Dinamis*, *Website Interaktif* (Zufria dan Ashari, 2017).

*Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses dimana pun selama anda terkoneksi dengan jaringan internet *website* merupakan komponen atau kumpulan kompeonen terdiri dari teks, gambar, suara, animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi (Rochman dkk, 2018).

*World Wide Web* atau W3W atau juga dikenal dengan *WEB* adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. *Web* ini

menyediakan informasi bagi pemakai komputer yang terhubung ke internet dari sekedar informasi “*Negative/Positive*” atau informasi yang tidak berguna sama sekali sampai informasi yang serius; dari informasi yang gratisan sampai informasi yang komersial.

#### A. *Website Statis*

Dari kata statis atau saja, sudah dapat dipahami tidak berubah. Mudah bukan? Tapi tentu tidak ada salahnya jika kita juga ketahui makna sebenarnya dari situs statid ini. *Website Statis* adalah *web* yang mempunyai halaman tidak berubah. Artinya adalah untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengedit *code* yang menjadi struktur dari situs itu.

#### B. *Website Dinamis*

*Website Dinamis* merupakan *website* yang secara struktur diperuntukan untuk *update* sesering mungkin. Biasanya selain utama yang bisa diakses oleh user pada umumnya, juga disediakan halaman backend untuk mengedit kontent dari *website*. Contoh umum mengenai *website* dinamis adalah *web* berita atau *web portal* yang didalamnya terdapat fasilitas berita, *polling* dan sebagainya.

#### C. *Website Interaktif*

*Website Interaktif* adalah *web* yang saat ini memang sedang booming. Salah satu contoh *website* interaktif adalah *blog* dan forum. Di *website* ini user bisa berinteraksi dan beradu argumen mengenai apa yang



menjadi pemikiran mereka. Biasanya *website* seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak melenceng dari alur pembicaraan.

#### **2.1.8. Database**

Basis data merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lain. Basis data atau *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi pemakainya, Sistem basis data adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan dengan yang lainnya dan untuk membuatnya tersedia beberapa aplikasi yang bermacam-macam dalam suatu sistem organisasi. Sistem basis data adalah suatu sistem menyusun dan mengelola record-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 keputusan (Sutopo dkk, 2016).

*Database* (Basis Data) merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Basis data adalah kumpulan *data* yang mempunyai kaitan antara satu *data* dengan *data* lain sehingga membentuk suatu

bangunan data untuk menginformasikan suatu perusahaan atau instansi dalam batasan tertentu (Hanafiah dan Pirmansyah, 2019). Beberapa hal yang dimaksud unsur-unsur dari basis data adalah sebagai berikut:

### **2.1.9. MySQL**

MySQL adalah salah satu jenis *Database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *Databasenya*. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL, sebuah *Database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom (Sutopo dkk, 2016).

MySQL didefinisikan nama *Database server*. *Database server* adalah server yang berfungsi untuk menangani *Database*. *Database* adalah suatu pengorganisasian data. Dengan menggunakan MySQL, kita bisa menyimpan data dan kemudian data bisa diakses cara yang mudah dan cepat (Rochman dkk, 2018).

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan

pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Hanafiah dan Pirmansyah, 2019).

#### A. Entitas

Entitas adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang kesehatan Entity adalah Pasien, Dokter, Kamar.

#### B. *Field*

Setiap entity mempunyai atribut atau sebutan untuk mewakili suatu entity. Seorang siswa dapat dilihat dari atributnya misalnya, NIM, Nama\_siswa, Alamat.

#### C. *Record*

Record adalah kumpulan isi elemen data (atribut) yang saling berhubungan menginformasikan tentang suatu entity secara lengkap.

Contoh Kumpulan atribut NIP, Nama, dan alamat berisikan “01001245566”, Sanusi, Jl. Hati suci No 2 Kupang.

#### D. *Data Value*

Merupakan data aktual atau informasi yang disimpan di tiap data elemen. Isi atribut disebut nilai data.

#### E. Kunci Elemen Data (*Key Data Element*)

Komponen-komponen Sistem Basis Data (*Database*)

Basis data merupakan sistem yang terdiri atas kumpulan file atau tabel yang saling berhubungan dan *Database Management System* (DBMS) yang

memungkinkan beberapa pemakai untuk mengakses dan manipulasi file-file tersebut (Fathansyah, 1999). Dalam Sistem Basis data memiliki beberapa komponen yaitu:

A. Perangkat Keras (*Hardware*)

Perangkat keras yang biasanya terdapat dalam sistem basis data adalah memori sekunder *harddisk*.

B. Sistem Operasi (*Operating System*)

Sistem Operasi (*Operating System*) merupakan program yang mengaktifkan atau mengfungsikan sistem komputer, mengendalikan seluruh sumber daya (*resource*) dan melakukan operasi-operasi dalam komputer. Sistem Operasi yang banyak digunakan seperti: MS-DOS, MS-*Windows* 95 MS *Windows* NT, dan Unix.

C. Basis data (*Database*)

Sebuah basis data (*Database*) dapat memiliki beberapa basis data. Setiap basis data dapat berisi atau memiliki sejumlah objek basis data seperti *file* atau table *Database*

D. *Management System* (DBMS)

Pengolahan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang disebut DBMS yang menentukan bagaimana data disimpan, diubah dan diambil kembali.

E. Pemakai (*User*)

Bagi pemakai dapat berinteraksi dengan basis data dan memanipulasi data dalam program yang ditulis dalam bahasa pemrograman. Tujuan utama dalam pengolahan data dalam sebuah basis data adalah agar kita dapat memperoleh data yang kita cari dengan mudah dan cepat (Fathansyah,1999).

Pemanfaatan basis data dilakukan dengan tujuan yaitu:

A. Kecepatan dan kemudahan (*Speed*)

Pemanfaatan *Database* memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan (manipulasi) dan menampilkan kembali data tersebut dengan cepat dan mudah, dari pada kita menyimpan data secara manual.

B. Efisien ruang penyimpanan (*Space*)

Dengan *Database* penggunaan ruang penyimpanan data dapat dilakukan karena kita dapat melakukan penekanan jumlah pengulangan data dengan menerapkan jumlah pengkodean.

C. Keakuratan (*Acuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data dengan penerapan aturan atau batasan tipe data dapat diterapkan dalam *Database* yang berguna untuk menentukan ketidakakuratan pemasukan atau penyimpanan.

D. Keamanan (*Security*)

Dalam jumlah sistem (apilkasi) pengelola *Database* tidak menerapkan aspek keamanan dalam penggunaan *Database*. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan. Dengan begitu kita dapat

menentukan siapa yang boleh menggunakan *Database* dan menentukan jenis operasi-operasi apa saja yang boleh dilakukan.

E. Terpeliharanya keselarasan data (*Consitant*)

Apabila ada perubahan data pada aplikasi yang berbeda maka secara otomatis perubahan itu berlaku untuk keseluruhan.

F. Data dapat dipakai secara bersama (*shared*)

Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi (secara *batch* maupun *online*) pada saat bersamaan.

G. Dapat diterapkan standarisasi (*standardization*)

Dengan adanya pengontrolan yang terpusat maka DBA dapat menerapkan standarisasi data yang disimpan sehingga memudahkan pemakaian, pengiriman maupun pertukaran data.

H. Kelemahan Sistem Basis Data

- Memerlukan tenaga spesialis
- Kompleks
- Memerlukan tempat yang besar
- Mahal
- Pengguna Basis Data
- *System Engineer*

I. *Database Administrator* (DBA)

Tenaga ahli yang mempunyai tugas untuk mengontrol sistem basis data secara keseluruhan, meramalkan kebutuhan akan sistem basis data, merencanakannya dan mengaturnya.

#### **2.1.10. HTML**

HTML adalah singkatan dari *HyperText Markup Language*. HTML merupakan *file* teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke user melalui suatu aplikasi *web browser*. Dapat disimpulkan bahwa HTML (*HyperText Markup Language*) ialah bahasa pemrograman untuk membuat desain *body* atau *layout* (tampilan) dari sebuah *website*. Bahasa pemrograman HTML inilah yang akan dieksekusi oleh *browser* dan ditampilkan di jendela *browser* (Hanafiah dan Pirmansyah, 2019).

*Hypertext Markup Language* (HTML) adalah sebuah bahasa *formatting* yang digunakan untuk membuat sebuah halaman *website*. HTML memiliki beberapa element yang tersusun dari *tag-tag* yang memiliki fungsinya masing-masing. *Hypertext Markup Language* atau HTML adalah bahasa *markup* standar yang digunakan untuk membuat halaman *website* dan aplikasi *web*. Sejarah HTML dimulai oleh Tim Berners-Lee, seorang ahli fisika di lembaga penelitian CERN yang berlokasi di Swiss. Versi pertamanya dirilis pada tahun 1991, dengan 18 *tag*. Sejak saat itu, setiap kali ada versi barunya, pasti akan selalu ada *tag* dan *attribute* (*tag modifier*) yang juga baru. Ketika bekerja dengan bahasa *markup* ini, Anda menggunakan struktur kode sederhana (*tag dan attribute*) untuk *markup* halaman

*website*. Misalnya, Anda membuat sebuah paragraf dengan menempatkan *enclosed text* di antara *tag* pembuka `<p>` dan *tag* penutup `</p>`. File *HTML* diakhiri dengan ekstensi. *HTML* atau *.htm*. Ekstensi file ini bisa dilihat dengan menggunakan *web browser* apa pun (seperti *Google Chrome*, *Safari*, atau *Mozilla Firefox*). *Browser* tersebut membaca file *HTML* dan merender kontennya sehingga *user internet* bisa melihat dan membacanya. Biasanya, rata-rata situs *web* menyertakan sejumlah halaman *HTML* yang berbeda-beda. Contohnya, halaman beranda, ‘Tentang Kami’, halaman kontak memiliki dokumennya sendiri-sendiri. Masing-masing halaman tersebut terdiri atas serangkaian *tags* (bisa disebut juga *elements*), yang tersusun untuk membentuk sebuah halaman *website*. Tag tersebut membuat hierarki yang menyusun konten hingga menjadi bagian, paragraf, *heading*, dan *block* konten lainnya (Juliany dkk, 2018).

#### **2.1.11. 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 (Sutopo dkk, 2016).

PHP adalah bahasa pemrograman scripting yang pertama dikembangkan untuk meng-generate statement *HTML*. Bahkan program yang dikembangkan



dengan PHP seratus persen, tetap ditampilkan dalam bentuk kode HTML (Rochman dkk, 2018).

PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP wajib adanya seb server (Hanafiah dan Pirmansyah, 2019).

Merupakan singkatan recursive dari PHP: *Hypertext Preprocessor*. PHP termasuk bahasa pemrograman yang paling banyak digunakan untuk membuat aplikasi *web*. PHP memungkinkan pembuatan aplikasi *web* yang dinamis, dalam arti dapat membuat halaman *web* yang dikendalikan oleh data (Juliany dkk, 2017).

#### **2.1.12. CSS**

*CSS* atau *Cascading Style Sheets* adalah sebuah dokumen yang berisi aturan yang digunakan untuk memisahkan isi dengan *layout* dalam halaman-halaman *web* yang dibuat. *CSS* memperkenalkan “*template*” yang berupa style untuk dibuat dalam mengizinkan penulisan kode yang lebih mudah dari halaman-halaman web yang dirancang. *CSS (Cascading Style Sheet)* adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman *website* (situs). Singkatnya dengan menggunakan Metode *CSS* ini anda dengan mudah mengubah secara keseluruhan warna dan tampilan yang ada di situs anda, sekaligus memformat ulang situs anda (merubah secara cepat). *CSS* juga memungkinkan si pembuat web

untuk memodifikasi HTML untuk membentuk tampilan sebuah *website*. Jadi kalo kita ini adalah arsitek, maka HTML adalah bangunannya dan CSS adalah desain interior dan eksteriornya (Hanafiah dan Pirmansyah, 2019).

CSS merupakan singkatan dari *Cascading Style Sheets*. Sesuai dengan namanya CSS memiliki sifat *style sheet language* yang berarti bahasa pemrograman yang digunakan untuk *web design*. CSS adalah bahasa pemrograman yang di gunakan untuk *men-design* sebuah halaman *website*. dalam merancang halaman *website*, CSS menggunakan penanda yang kita kenal dengan *id* dan *class*. Kepanjangan CSS adalah singkatan dari *Cascading Style Sheets*, yang berguna untuk menyederhanakan proses pembuatan *website* dengan mengatur elemen di bahasa *markup*. Pemakaian CSS adalah untuk mendesain *website front end*. CSS adalah *standard* pembuatan dan pemakaian *style* untuk dokumen terstruktur, CSS digunakan untuk mempersingkat penulisan *tag* HTML seperti *font*, *color*, *text*, dan *table* menjadi lebih ringkas sehingga tidak terjadi pengulangan tulisan. CSS adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan dalam kode HTML atau sekedar menjadi rujukan oleh HTML dalam pendefinisian *style*. CSS menggunakan kode2 yang tersusun untuk menetapkan *style* pda elemen HTML atau dapat juga digunakan membuat *style* baru yang biasa disebut *class*. CSS dapat mengubah besar kecilnya *text*, mengganti warna *background* pada sebuah halaman, atau dapat pula mengubah warna *border* pada tabel, dan masih banyak lagi hal yang dapat dilakukan oleh CSS. Singkatnya, CSS digunakan untuk mengatur susunan tampilan pada halaman HTML. CSS dapat

mengatur banyak *atribut* pada sebuah halaman secara mudah. Misalnya: warna *background*, *border*, *shadow*, yang berbeda pada masing-masing *tag* yang digunakan. Satu dari beberapa kelebihan teknologi CSS ini dan merupakan alasan banyak orang menyukai penggunaannya adalah memperbolehkan kita untuk mendefinisikan sebuah *style-sheet global* yang dapat menerapkan aturan-aturan CSS tersebut untuk keseluruhan dokumen-dokumen HTML pada halaman web site anda. Keuntungannya jelas sekali, jika kita ingin mengubah tampilan situs, kita tinggal mengedit satu file saja sekarang, kita akan mencoba melihat bagaimana CSS itu bekerja. Untuk itu, kita butuh sebuah *text editor* (*notepad*, *wordpad*, *Ms word*, semua tergantung kesukaan masing2), dan sebuah *web browser* tentunya (Juliany dkk, 2018).

#### A. Sejarah CSS

CSS diperkenalkan untuk pengembangan *website* pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan *parent-child* pada setiap *style*. Setelah CSS distandarisasikan, *Internet Explorer* dan *Netscape* melepas *browser* terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS. Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian membentuk hubungan ayah-anak (*parent-child*) pada setiap *style*. CSS sendiri merupakan sebuah teknologi internet yang direkomendasikan oleh *World Wide Web*

*Consortium* atau W3C pada tahun 1996. Setelah CSS distandardisasikan, *Internet Explorer* dan *Netscape* melepas *browser* terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

## B. Fungsi CSS

Fungsi utama dari CSS ialah mendesain, merancang, merubah, dan juga membentuk halaman pada *website* ataupun blog dan isi dari halaman *website* itu ialah tag-tag HTML, logikanya CSS itu bisa mengubah *tag-tag HTML* yang sederhana sehingga bisa menjadi lebih fungsional dan juga menarik.

## C. Tujuan CSS

Tujuan utama dari CSS dikembangkan ialah agar dapat membedakan konten dari dokumen dan dari tampilan dokumen. Maka dari itu, pembuatan dalam pemrograman ulang *web* pun akan lebih mudah dilakukan. Hal-hal yang terlibat dalam desain *web* itu adalah warna, ukuran dan *formatting*. Dengan adanya CSS ini, konten dan desain *web* akan tampak mudah saat dibedakan, jadi memungkinkan juga untuk melakukan pengulangan pada tampilan-tampilan tertentu pada suatu *web*, sehingga akan sangat memudahkan dalam hal membuat halaman *web* yang banyak, dimana pada akhirnya bisa menghemat waktu dalam pembuatan *web*.

### 2.1.13. *Black-Box*

Pengujian *software* sangat diperlukan untuk memastikan *software/aplikasi* yang sudah/sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang atau penguji *software* harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian atau *Testing* sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan *software* seperti halnya analisis, desain, dan pengkodean. Pengujian *software* haruslah dilakukan dalam proses rekayasa perangkat lunak atau *software engineering*. Sejumlah strategi pengujian *software* telah diusulkan dalam literatur. *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* merupakan strategi *Testing* dimana hanya memperhatikan/memfokuskan kepada faktor fungsionalitas dan spesifikasi perangkat lunak. Berbeda dengan *white box*, *black box Testing* tidak membutuhkan pengetahuan mengenai, alur internal (*internal path*), struktur atau implementasi dari *software under test* (SUT) (Mustaqbal dkk, 2015).

*Blackbox Testing* adalah metode merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan

kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Rochman dkk, 2018). Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap *form*-nya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan (Hanafiah dan Pirmansyah, 2019). Pengujian dengan metode *Blackbox Testing* memiliki dua teknik yaitu *Equivalence Partitioning* dan *Boundary Value Analysis* (Shadiq dkk, 2021). *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).

Proses *Black Box Testing*:

1. Menganalisis kebutuhan dan spesifikasi dari perangkat lunak.
2. Pemilihan jenis input yang memungkinkan menghasilkan output benar serta jenis input yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
3. Menentukan output untuk suatu jenis input.
4. Pengujian dilakukan dengan input-input yang telah benar-benar diseleksi.
5. Melakukan pengujian.
6. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan.

7. Menentukan fungsionalitas yang seharusnya ada pada perangkat lunak yang sedang diuji.

#### **2.1.14. UML**

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu *Unified Modeling Language* (UML) (Rochman dkk, 2018).

UML merupakan konsep pemodelan berbasis *Object Oriented* (OO) yang dapat membantu dalam perancangan dan penganalisaan sistem. Di dalam UML banyak diagram yang dapat digunakan untuk menganalisa sistem dan perancangan sistem. Komponen diagram UML yang digunakan ialah *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi (Affandi dan Syahputra, 2018).

UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada *system*. UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan

pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah *software*. UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor *scalability*, *robustness*, *security*, dan sebagainya (Yuliana dan Azizah, 2019). Jenis-Jenis Diagram *UML* dan Contoh Diagram *UML*:

1. *Use Case Diagram*

*Use Case Diagram* yaitu salah satu jenis diagram pada *UML* yang menggambarkan interaksi antara sistem dan aktor, *Use Case Diagram* juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

2. *Activity Diagram*

*Activity Diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada *UML* yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.

3. *Sequence Diagram*

*Sequence* diagram yaitu salah satu jenis diagram pada *UML* yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *Use Case Diagram*.

4. *Class diagram*



*Class diagram* yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.

Adapun tujuan dan fungsi perlu adanya UML yaitu sebagai berikut:

1. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
2. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
3. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
4. Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek.
5. Mempermudah pengguna untuk membaca suatu sistem.
6. Berguna sebagai *blueprint*, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa *coding* suatu program.

#### **2.1.15. Quality of Service (QoS)**

Dari segi *networking*, QoS mengacu kepada kemampuan memberikan pelayanan berbeda kepada lalu lintas jaringan dengan kelas-kelas yang berbeda. Tujuan akhir dari QoS adalah memberikan *network service* yang

lebih baik dan terencana dengan salah satu parameternya, yaitu *packet lost* dan *throughput*. Berikut adalah penjelasan mengenai parameter-parameter yang digunakan dalam penilaian QoS yang baik (Iskandar, 2015).

#### 2.1.15. *Throughput*

Yaitu kecepatan (*rate*) *transfer* data yang efektif yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut (Iskandar, 2015).

Untuk mengukur *throughput* digunakan persamaan

$$Throughput = \frac{\text{Jumlah data yang dikirim}}{\text{Lama pengamatan}}$$

#### 2.1.16. *Packet lost*

*Packet Lost* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket data yang hilang atau jumlah data yang gagal dikirim (Iskandar, 2015).

**Tabel 2. 1 Packet Lost**

<b>Kategori Degradasi</b>	<b>Packet Lost %</b>	<b>Indeks</b>
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

## **2.2. Penelitian Terdahulu**

Pada penelitian terdahulu merupakan gambaran penelitian relevan yang hampir sama permasalahan yang ada pada penelitian tersebut dengan penelitian saat ini yang sedang dilaksanakan. Penelitian terdahulu merupakan referensi mengenai penelitian yang sedang dikaji berdasarkan permasalahan yang ada di lokasi penelitian. Berikut adalah sumber referensi penelitian terdahulu:

Rifki Arif Kurniawan dengan judul “*IMPLEMENTASI FILTER CROSS SITE SCRIPTING MBIJON, VOKU DEPYCODE, SCRIPTZTEAM*” mengatakan bahwa filter dapat digunakan untuk mengendalikan data yang masuk dan keluar berdasarkan kriteria data yang dimasukkan (Kurniawan, 2018).