

BAB III. LANDASAN TEORI

Bab ini membahas dan menjelaskan tentang dasar-dasar teori yang akan melandasi penulis dalam pembuatan penelitian yang akan dilakukan.

3.1. Sistem Informasi

Sistem informasi merupakan sistem yang mengkombinasikan antara aktivitas manusia dengan melibatkan berbagai macam komponen teknologi informasi seperti, komputer (perangkat keras), *windows* (perangkat lunak), *database*, internet, dll, agar dapat berinteraksi ataupun memberikan informasi kepada organisasi dan masyarakat umum. Komponen tersebut berfungsi untuk mengumpulkan, menyimpan, dan memproses data untuk menyediakan informasi maupun pengetahuan [11].

3.2. Audit Internal

Audit internal merupakan mekanisme kontrol yang digunakan untuk membantu sebuah manajemen dalam mencapai tujuan perusahaan ataupun organisasi dengan menggunakan wawasan yang menarik. Menurut beberapa teori, audit internal diperkenalkan sebagai mekanisme pengendalian internal untuk mengetahui apakah manajemen benar-benar melaksanakan tanggung jawab yang ada. Perusahaan harus bertanggung jawab dalam memandu aktivitas dan keputusan yang akan dibuat oleh manajemen [12].

Perusahaan yang memiliki aktivitas, keputusan, dan tata kelola yang baik cenderung akan menghasilkan laporan audit internal yang berkualitas, ini dikarenakan laporan tersebut dapat meningkatkan nilai perusahaan dan memudahkan auditor dalam memahami sistem internal perusahaan. Perusahaan dengan tata kelola yang baik akan membuat audit internal yang berkualitas tinggi, sehingga berdampak pada peningkatan kinerja bagi perusahaan. Dengan adanya audit internal yang baik, maka tata kelola perusahaan dapat

meningkatkan transparansi dan efektivitas kinerja manajemen sehingga perusahaan dapat bertahan dan bersaing di kemudian hari [13].

3.3. Microsoft Visual Studio

Microsoft Visual Studio adalah perangkat lunak pada lingkungan pengembangan terintegrasi (*Software IDE/Software Integrated Development Environment*) yang menawarkan pengembangan fitur-fitur dalam pembuatan program komputer, situs, aplikasi, dan layanan website [14]. Evolusi pengembangan visual studio dapat dilihat pada Gambar 3.1.

Release	Version	.NET Framework	Release Date
Visual Studio 2019	16.0	3.5–4.8	April 2, 2019
Visual Studio 2017	15.0	3.5–4.7	March 7, 2017
Visual Studio 2015	14.0	2.0–4.6	July 20, 2015
Visual Studio 2013	12.0	2.0–4.5.2	October 17, 2013
Visual Studio 2012	11.0	2.0–4.5.2	September 12, 2012
Visual Studio 2010	10.0	2.0–4.0	April 12, 2010

Gambar 3.1. Pengembangan Visual Studio tahun 2010-2019 [14]

Visual studio juga menyediakan kode editor dan *debugger* yang terintegrasi, perangkat lunak ini juga mendukung Bahasa pemrograman yang sangat banyak, termasuk bahasa bawaan yang ada. Contohnya Bahasa C, C# (*C Sharp*), *JavaScript*, *HTML*, *CSS*, dll.

3.4. ASP.NET

ASP.NET merupakan pengembangan dari *framework Active Server Page (ASP)* yang memberikan tampilan untuk membuat aplikasi website lebih baik dari sebelumnya dan sebagian struktur kalimat, kerangka kerja, dan keamanan lebih bagus daripada *ASP*. Dengan menggunakan *framework ASP.NET*, programmer dapat menikmati kemajuan peningkatan website, yaitu dapat menggunakan *PHP* dan *JSP* [15].

Kerangka kerja *ASP.NET* dapat digunakan untuk menghindari kloning dan mempercepat pengembangan aplikasi *website* dengan menyediakan akses

database. Saat pembuatan file pun, secara otomatis akan terbuatnya komponen *model*, *view*, dan *controller* pada kerangka kerja *ASP.NET* tersebut, sehingga proses pembuatan aplikasi *website* dapat lebih mudah dan terstruktur [16].

3.5. Bahasa Pemrograman C# (C Sharp)

Bahasa pemrograman *C#* (*C Sharp*) adalah bahasa pemrograman yang sangat fleksibel, dan terkenal sebagai bahasa penulisan sistem operasi *UNIX*. Selain itu, *C#* juga merupakan bahasa pemrograman berorientasi objek yang memungkinkan dapat membuat program yang memiliki tingkat keandalan dan stabilitas yang tinggi [17].

Dalam Bahasa pemrograman *C#* ada beberapa struktur dasar yang harus selalu diingat, yaitu *library*, *namespace*, *class*, *method declaration*, dan *method (command)*. *Library* disini adalah koleksi program yang bermanfaat untuk mengembangkan program, *namespace* adalah nama project yang dibuat, *class* adalah objek untuk membuat kerangka dasar, *method declaration* adalah pembuatan badan program dan berisi kode program yang akan dijalankan, lalu *method* adalah isi program yang nantinya akan dieksekusi oleh compiler. Contoh *method* tersebut dapat dilihat pada Gambar 3.2.

```
using System;
class GlazerCalc
{
    static void Main()
    {
        double width, height, woodLength, glassArea;
        string widthString, heightString;

        widthString = Console.ReadLine();
        width = double.Parse(widthString);

        heightString = Console.ReadLine();
        height = double.Parse(heightString);

        woodLength = 2 * ( width + height ) * 3.25 ;
        glassArea = 2 * ( width * height ) ;

        Console.WriteLine ( "The length of the wood is " +
            woodLength + " feet" );
        Console.WriteLine( "The area of the glass is " +
            glassArea + " square metres" );
    }
}
```

Gambar 3.2. Struktur Program Pada C# [17]

3.6. PHP Hypertext Preprocessor

PHP Hypertext Preprocessor (PHP) adalah salah satu bahasa pemrograman yang paling populer, banyak digunakan oleh komunitas *open source* maupun industri dalam bidang teknologi untuk membangun kerangka kerja aplikasi besar yang berfokus pada pembuatan platform *website* [18]. File *PHP* pun dapat berisikan teks, *HTML*, *JavaScript*, dan kode *PHP* itu sendiri.

Untuk menjalankan *PHP* dibutuhkan *web server*, yang bertanggung jawab untuk memproses file *php* dan mengirimkan hasil pemrosesan ke *browser client* untuk ditampilkan. Maka dari itu, *PHP* merupakan *server-side scripting* (script yang diproses pada sisi server). Aturan penulisan script pada *PHP* pun ada beberapa macam, seperti `<?php` dan `?>`, `<script language='php'>` dan `</script>`, dan masih ada lagi. Akan tetapi, tanda yang resmi dan sering digunakan adalah `<?php` dan `?>`, dan pada setiap akhir perintah harus diakhiri dengan tanda titik koma (;). Adapun contoh dari penulisan script pada *PHP* dapat dilihat pada Gambar 3.3 [19].

```
<?php
echo 'Halo, Dunia! <br />';
echo 'Ini <i>script</i> php pertamaku';
?>
```

Gambar 3.3. Penulisan Script pada PHP [19]

3.7. Microsoft SQL Server

Microsoft SQL Server adalah sistem manajemen basis data relasional, dengan *MS-SQL* dan *Transact-SQL* sebagai bahasa pemrograman yang terstruktur. Sistem manajemen ini digunakan untuk melakukan penyisipan, modifikasi, penghapusan, pengambilan, serta pengendalian akses data dalam basis data tersebut. Adapun kueri yang sering digunakan saat pengambilan data yaitu, perintah *SELECT* [20].

SQL Server juga disebut dengan *Relational Database Management System (RDBMS)*. *SQL Server* bersifat fungsionalitas (menyediakan

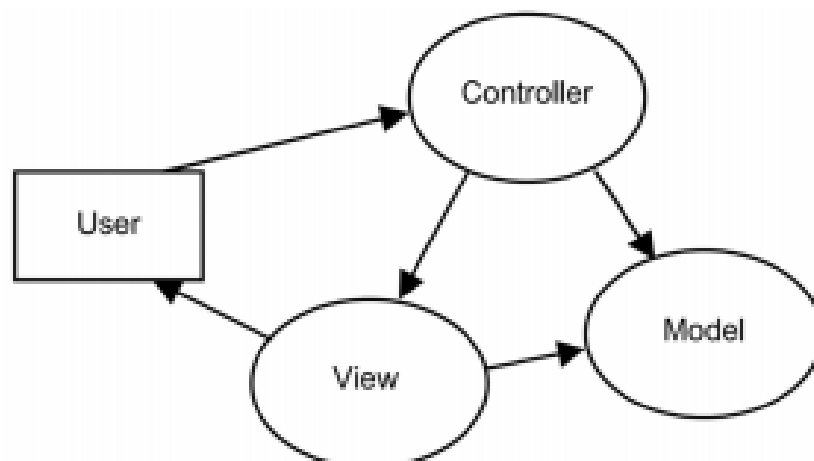
administrasi data lebih mudah), memberikan pemahaman yang baik, proses kontrol redundansi lebih efisien, dan prosedur pencarian lebih kompleks [21].

3.8. Web Server

Web server adalah server yang berfungsi dalam memberikan suatu layanan berbasis protokol http dalam internet, contohnya yaitu apache, Microsoft IIS (*Internet Information System*), Tomcat, dsb. Untuk *web server* sendiri ada beberapa cara dalam melakukan pengaksesan. Contohnya, *web server* membutuhkan *web client* yang biasa dipakai oleh pengguna internet, memanggil nama *localhost*, mengakses melalui jaringan lain dengan memanggil *IP (Internet Protocol)* pada server [22].

3.9. Arsitektur MVC (*Model View Controller*)

Arsitektur *MVC* pada awalnya merupakan antarmuka pengguna dalam aplikasi yang akan diimplementasikan dengan menggunakan bahasa pemrograman. Dalam arsitektur ini, dibagi menjadi tiga komponen, yaitu *model*, *view*, dan *controller*. *Model* berisi fungsi dan data yang mengungkapkan area logika pembuatan, *view* akan menyajikan antarmuka informasi kepada pengguna, sedangkan *controller* akan mengelola perubahan pada tampilan [23]. Komponen antarmuka pada *MVC* dapat dilihat pada Gambar 3.4.



Gambar 3.4. Komponen Arsitektur MVC [23]

Arsitektur MVC dibagi menjadi 3 komponen, antara lain:

1. *Model*

Model memiliki tugas (pedoman) untuk menyimpan data heterogen dalam satu format dan menangani fungsionalitas inti dari sistem yang tidak bergantung pada antarmuka pengguna. Komponen *model* sendiri menangani pengumpulan data mentah, proses Extract-Transform-Load, dan memelihara meta-data [24]. Disini *model* memiliki fungsi untuk berinteraksi dengan basis data agar dapat menangani data, logika, dan aturan.

2. *View*

View menyediakan dan mengontrol suatu tampilan agar informasi dapat disajikan dengan baik, komponen *view* menampilkan informasi yang telah disimpan pada gudang data ke dalam bentuk tampilan grafis dan tekstual. Setiap *view* memiliki komponen pengontrol yang terkait dan bertanggung jawab untuk menghubungkan dan menampilkan output kepada pengguna [24]. Disini *view* memiliki fungsi untuk menghubungkan pengguna ke *model*.

3. *Controller*

Controller merupakan komponen yang menangani dan mengimplementasikan fungsi *event handling* pada input pengguna. Controller selalu mengikuti mekanisme change-propagation, dimana semua tampilan yang terkait dengannya harus mencerminkan perubahan yang sama (perilaku dinamis) [24]. Disini *controller* memiliki fungsi untuk mengirim perintah ke *model* untuk memperbaharui data dan mengirim perintah ke *view* untuk merubah data.