

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang berjudul “Aplikasi Pendataan Medical Check Up (MCU) Pada RST Ciremai Cirebon”. Bertujuan untuk membantu MCU dalam melakukan penyimpanan data ke dalam basis data, melakukan pendataan untuk pemeriksaan fisik dan laboratorium, dan menyajikan data dalam bentuk grafik. Perancangan sistem tersebut menggunakan metode *Prototyping*. Sedangkan pembuatan aplikasi menggunakan bahasa pemrograman *Java* dan *My Sql* sebagai basis data. (Juliyanti, Ika,2014)

Penelitian yang berjudul “Sistem Informasi Pendataan Barang Berbasis Komputerisasi Pada Perusahaan Manufaktur Anggun Rottan”. Penelitian tersebut bertujuan untuk pengembangan sistem komputerisasi dan mempercepat proses pengolahan data barang di Perusahaan Anggun Rottan. Pembuatan aplikasi tersebut menggunakan bahasa pemrograman *Visual Basic* dan *Microsoft SQL Server* sebagai *basis data*. (Brigita Panca Indriarini, Umi Safiatun, Nur Hidayati 2010).

Penelitian yang berjudul “Pembuatan Sistem Informasi Administrasi Berbasis Desktop Pada Bimbingan Belajar Citra Bagus Grup Sleman”. Penelitian tersebut bertujuan untuk menyederhanakan dan mempercepat melaksanakan administrasi kegiatan dan untuk memfasilitasi presentasi dari pengolahan data dan mempercepat informasi ketika diperlukan. Pembuatan aplikasi tersebut menggunakan *Netbens* dan *MySQL* sebagai databasenya. (Ana Wati Ndarbeni, 2013).

Berdasarkan penelitian terlebih dahulu yang telah dijelaskan diatas, dapat disimpulkan bahwa persamaan nya adalah:

1. Membuat aplikasi berbasis *Desktop* sebagai media untuk memudahkan pendataan di instansi masing-masing.
2. Mempercepat informasi yang diperlukan di instansi masing-masing.
3. Menggunakan *MySQL* sebagai databasenya

Sedangkan perbedaan yang dapat disimpulkan penulis adalah:

1. Penulis menggunakan Bahasa Pemrograman *c#* sedangkan penelitian sebelumnya menggunakan bahasa pemrograman *java*.
2. Penelitian sebelumnya menggunakan Bahasa Pemrograman *Visual Basic*, sedangkan penulis menggunakan Bahasa Pemrograman *Visual Studio*.

2.2 Landasan Teori

Untuk mendukung pembuatan laporan ini, maka perlu dikemukakan hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam membangun laporan ini.

2.2.1 Pengertian Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna.

Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas dapat disimpulkan bahwa aplikasi adalah berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. (Anisyah, 2000:3).

2.2.2 Pengertian Aplikasi Desktop

desktop application atau aplikasi *desktop* adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet disuatu komputer otonom. (Dew Omenn, 2013)

Aplikasi berbasis *Desktop* merupakan aplikasi yang dijalankan pada masing – masing komputer atau klien. Aplikasi berbasis *Desktop* harus di install terlebih dahulu ke dalam komputer agar dapat digunakan. Berdasarkan pengertian diatas penulis menyimpulkan bahwa aplikasi berbasis *Desktop* adalah aplikasi yang berjalan pada komputer yang dapat digunakan secara langsung ketika kode program selesai dikompilasi. (Rafy101, 2013)

2.2.3 Pengertian Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau komponen yang terorganisir dan saling berinteraksi satu sama lainnya. Sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang berfungsi dan dengan tujuan yang sama. (James A, Hall, 2007:6). Sistem terdiri dari unsur-unsur seperti masukan (*input*) pengolahan (*processing*), serta keluaran (*output*). (Scoot,1996).

2.2.4 Pengertian Informasi

Informasi adalah data yang sudah diolah, dibentuk, atau dimanipulasi sesuai dengan keperluan tertentu. (Drs Zulfikar Amsyah, MLS, 2003:2). Informasi berarti data yang telah dibentuk menjadi sesuatu yang memiliki arti dan berguna bagi manusia. (Kenneth C. Laudon dan Jane P. Laudon, 2008:16).

2.2.5 Pengertian Sistem Informasi

Secara garis besar Sistem Informasi dapat diartikan sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan yang diperlukan. Sistem Informasi didefinisikan sebagai sekumpulan komponen yang saling berhubungan, mengumpulkan (mendapatkan), memproses, menyimpan, dan mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi serta. (Kenneth C Laudon dan Jane P Laudon, 2008:15)

Sistem Informasi adalah serangkaian prosedur formal dimana data dikumpulkan, diproses menjadi informasi dan didistribusikan kepada pengguna. (James A. Hall, 2007:9)

2.2.6 Siklus Sistem Informasi

Dalam pengembangan sebuah sistem informasi memerlukan proses – proses dan urutan yang standar, yaitu Analisis, Desain, Implementasi, Pemeliharaan. Proses-proses tersebut dituangkan dalam satu metode yang dikenal dengan nama *System Development Life Cycle (SDLC)*. *SDLC* merupakan metodologi umum dalam pengembangannya sistem yang memadai kemajuan usaha analisis dan desain. (Youssef Bassil)

SDLC meliputi fase – fase sebagai berikut:

1. *Analysis*
2. *Design*
3. *Implementation*
4. *Testing*
5. *Maintenance*

2.3 Teknologi Pengembangan Aplikasi







2.3.1 Unified Markup Language (UML)

Unified Markup Language (UML) merupakan bahasa permodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan seluruh rancangan aplikasi lunak. Penggunaan model ini bertujuan untuk mengidentifikasi bagian – bagian yang termasuk dalam lingkup sistem didalam aplikasi. *Model UML* yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case Diagram*, *Activity Diagram*, *Flowchart*, dan *Class Diagram*.

a. *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang dibutuhkan





Tabel 2.1 *Use Case* diagram


No	Gambar	Nama	Keterangan
1		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem untuk menghasilkan suatu hasil yang terukur bagi suatu actor
2		Actor	Menspesifikasikan himpunan peran yang penggunaan mainkan ketika berinteraksi dengan use case
3		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya
4		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
5		Include	Menspesifikasikan bahwa use case sumber secara eksplisit
6		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan

b. *Activity Diagram*

Activity Diagram adalah menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses *parallel* yang mungkin terjadi beberapa eksekusi, *activity diagram* merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antara *subsistem*) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur- jalur aktivitas dari level secara umum.

Tabel 2.2 Simbol *Activity Diagram*

No	Gambar	Nama Gambar	Keterangan
1		Initial Node	Bagaimana objek dibentuk atau diawali
2		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
3		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
4		Decision	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu

No	Gambar	Nama Gambar	Keterangan
5		Swimlane	Digunakan untuk pembagian activity diagram yang menunjukkan siapa yang melakukan aktivitas

c. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi *objek*. *Class* menggambarkan keadaan (*atribut/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda/fungsi*). *Class* diagram menggambarkan struktur dan *deskripsi class, package* dan objek beserta hubungan satu sama lain seperti *containment, pewarisan asosiasi*, dan lain-lain. *Class* memiliki tiga area pokok:

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

2.3.2 Basis Data

Basis data merupakan merupakan kegiatan sistem program komputer untuk berbagai aplikasi komputer. Dalam basis data dibutuhkan suatu media simpan komputer yang terorganisasi sedemikian rupa dan juga pemeliharaan data baik dalam fungsi manajemen sistem.

Sedangkan pandangan lain basis data adalah suatu pengetahuan tentang organisasi data, sehingga database merupakan salah satu komponen yang penting dalam sistem informasi. Penerapan database dalam sistem informasi disebut dengan sistem basis data (*database system*).

2.3.3 *SQL Server*

Microsoft SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur client server. Fitur pada *Microsoft*

SQL Server yaitu mempunyai kemampuan untuk membuat *basis data mirroring* dan *clustering*. *Microsoft SQL Server* juga mendukung *SQL* sebagai bahasa untuk memproses query ke dalam database dan kita tahu bahwa *SQL* ini sudah digunakan secara umum pada semua produk *database server*. Kelebihan *Microsoft SQL Server* adalah sebagai berikut:

- a. Dengan kemampuannya untuk mengolah data yang besar maka *DBMS* ini sangat cocok untuk perusahaan mikro, menengah hingga perusahaan besar sekalipun
- b. *DBMS* jenis memiliki kelebihan mamanager user *dataset* masing – masing user dapat diatur hak aksesnya terhadap pengaksesan *database* oleh *DBA*
- c. Mempunyai tingkat keamanan data yang sangat baik
- d. Dapat melakukan *back up*, *recovery* dan *rollback data* yang mudah
- e. Mempunyai kelebihan untuk membuat *database mirroring* dan *clustering*

2.3.4 Bahasa Pemrograman C#

C# (C sharp) adalah sebuah bahasa pemrograman berbasis objek yang didukung oleh *Microsoft .Net Framework*. *Microsoft .NET Framework* adalah perantara agar aplikasi dengan bahasa pemrograman yang didukung dapat berkomunikasi dengan sistem operasi yang digunakan oleh komputer. Selain itu, *.NET Framework* juga memungkinkan *c#* untuk berkomunikasi dengan bahasa pemrograman lainnya yang juga didukung oleh *.NET Framework* seperti *VB .NET*, *F#*, atau *C++*.

Dengan kata lain, aplikasi yang kita buat dapat menggunakan komponen – komponen lain yang dibuat dengan menggunakan *VB.NET*, *J#*, atau *C++*.

2.3.5 Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk *aplikasi console*, *aplikasi windows* ataupun *aplikasi web*.

Visual Studio mencakup kompiler, *SDK*, *Integrated Development Environment (IDE)*, dan *dokumentasi* (umumnya berupa *MSDN Library*). *Kompiler*

yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual c#*, *Visual Basic*, *Visual Basic.NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan diatas Windows) ataupun managed code (dalam bentuk *Microsoft Intermediate Language* diatas *.NET Framework*) selain itu, *visual studio* juga dapat digunakan untuk 28 mengembangkan aplikasi *Silverlight*, *aplikasi Window Mobile* (yang berjalan diatas *.NET Compact Framework*).

2.3.6 Black Box Testing

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, pada skripsi dipilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

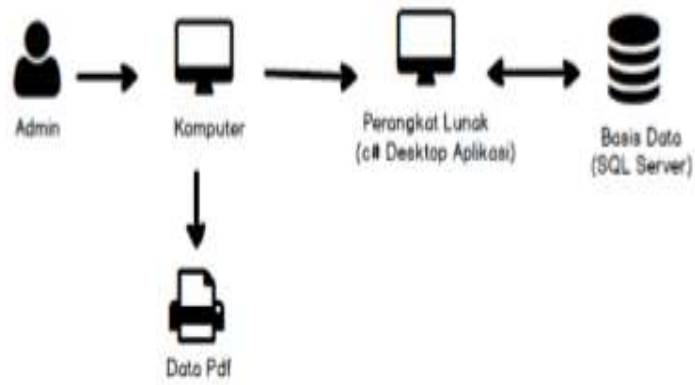
Black box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineer untuk memperoleh input yang sepenuhnya akan melaksanakan persyaratan fungsional. (Roger S. Pressman, 2010).

untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

- a. Fungsi yang tidak benar atau fungsi yang hilang.
- b. Kesalahan antarmuka.
- c. Kesalahan dalam struktur data atau akses database eksterna.
- d. Kesalahan kinerja.
- e. Kesalahan inisialisasi dan pemutusan kesalahan

2.3.7. Arsitektur Perangkat Lunak

Dalam mengembangkan aplikasi diperlukan perancangan arsitektur perangkat lunak yang bertujuan untuk menggambarkan bagaimana sistem dikembangkan dan dijalankan. Arsitektur perangkat lunak pada aplikasi dapat dilihat pada Gambar 2.1



Gambar 2.1 Arsitektur Perangkat Lunak