

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

a. Konsep Dasar Sistem Informasi

1. Pengertian Sistem

Menurut Mustakini (2009:34), “Sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu”. Sedangkan menurut Sucipto (Sucipto, 2010:2), “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu .

2. Pengertian Informasi

Menurut Mustakini (2009:36), “Informasi adalah data yang diolah menjadi bentuk yang berguna bagi pemakainya”. Informasi berarti data yang telah dibentuk dalam suatu format yang mempunyai arti dan berguna bagi manusia. Sebaliknya data merupakan sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau lingkungan fisik sebelum diolah dalam suatu format yang dapat dipahami dan digunakan orang.

3. Pengertian Sistem Informasi

Menurut Mulyanto (2009:29), “Sistem informasi merupakan suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisa, dan menyebarkan informasi untuk

mencapai suatu tujuan.” Dari pengertian-pengertian diatas dapat disimpulkan bahwa sistem informasi adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisa, dan menyebarkan informasi untuk mencapai suatu tujuan yang digunakan untuk pengambilan keputusan.

b. Website

Menurut Yuhfizar (2013:2), “*website* adalah keseluruhan halaman-halaman *web* yang terdapat dari sebuah domain yang mengandung informasi”. Sedangkan Arief (2011:9) menjelaskan bahwa “*web* dinamis adalah jenis *web* yang *content* atau isinya dapat berubah-ubah setiap saat”. Dalam teknologi pembuatan *web* dinamis sudah dirancang semudah mungkin bagi *user*, untuk perubahan *content* atau isi dokumen *web* dinamis *user* hanya perlu masuk kebagian *Control Panel* atau bagian *Administrator web* yang telah disediakan oleh teknologi *web* dinamis. Pengubahan *content* atau isi dokumen dalam *web* dinamis tidak perlu memiliki keahlian *programming* atau seorang *programmer* yang dapat mengubah isi dokumen *web* dinamis.

c. Web Browser

Menurut Arief (2011:19) “*Web browser* merupakan program yang berfungsi untuk menampilkan dokumen-dokumen *web* dalam format HTML”. Sedangkan Menurut Sibero (2013:12) “*Web browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*. Sumber informasi *web* diidentifikasi dengan *Uniform Resource Identifier* (URI) yang terdiri dari halaman *web*, video, gambar ataupun konten lainnya”.

d. *Web Server*

Menurut Sibero (2013:11) “*Web server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak. Secara bentuk dan fisik dan cara kerjanya, perangkat keras *web server* tidak berbeda dengan komputer rumah atau PC, yang membedakan adalah kapasitas dan kapabilitasnya”. *Web server* adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML (Sadeli, 2011:2).

e. *.NET Framework*

Menurut Faisal (2014:3) Microsoft *.NET Framework* adalah sebuah komponen yang dapat ditambahkan ke sistem operasi *Microsoft Windows* atau telah terintegrasi ke dalam *Windows* (mulai dari *Windows Server* 2003 dan versi-versi *Windows* terbaru). Kerangka kerja (*Framework*) menyediakan sejumlah besar solusi-solusi program untuk memenuhi kebutuhan-kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk *framework*. *.NET Framework* adalah kunci penawaran utama dari *Microsoft*, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk platform *Windows*.

Pada dasarnya, *.NET Framework* memiliki 2 komponen utama: *Common Language Runtime CLR* dan *.NET Framework Class Library*. Program - program yang ditulis untuk *.NET Framework* dijalankan pada suatu lingkungan *software* yang mengatur persyaratan-persyaratan runtime program. Runtime environment ini, yang juga merupakan suatu bagian dari *.NET Framework*, dikenal sebagai

Common Language Runtime (CLR). CLR menyediakan penampilan dari *application virtual machine*, sehingga para programmer tidak perlu mengetahui kemampuan CPU tertentu yang akan menjalankan program. CLR juga menyediakan layanan-layanan penting lainnya seperti jaminan keamanan, pengaturan memori, *garbage collection* dan *exception handling* / penanganan kesalahan pada saat runtime. *Class library* dan CLR ini merupakan komponen inti dari *.NET Framework*. Kerangka kerja itu pun dibuat sedemikian rupa agar para programmer dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga komputer dari beberapa ancaman keamanan. CLR adalah turunan dari CLI (*Common Language Infrastructure*) yang saat ini merupakan standar ECMA.

Solusi-solusi program pembentuk class library dari *.NET Framework* mengatasi area yang luas dari kebutuhan program pada bidang *user interface*, pengaksesan data, koneksi basis data, kriptografi, pembuatan aplikasi berbasis *web*, algoritma numerik, dan komunikasi jaringan. Fungsi-fungsi yang ada dalam *class library* dapat digabungkan oleh programmer dengan kodenya sendiri untuk membuat suatu program aplikasi baru.

Pada berbagai literatur dan referensi di Internet, *.NET Framework* seringkali disingkat menjadi *.NET* saja. *.NET* seringkali juga dapat diartikan sebagai *platform*, yang merupakan suatu lingkungan terpadu untuk pengembangan dan eksekusi untuk berbagai macam bahasa pemrograman dan kumpulan library untuk bekerja sama membuat dan menjalankan aplikasi berbasis *Windows* yang lebih mudah untuk dibuat, diatur, didistribusikan, dan diintegrasikan dengan sistem jaringan lain.

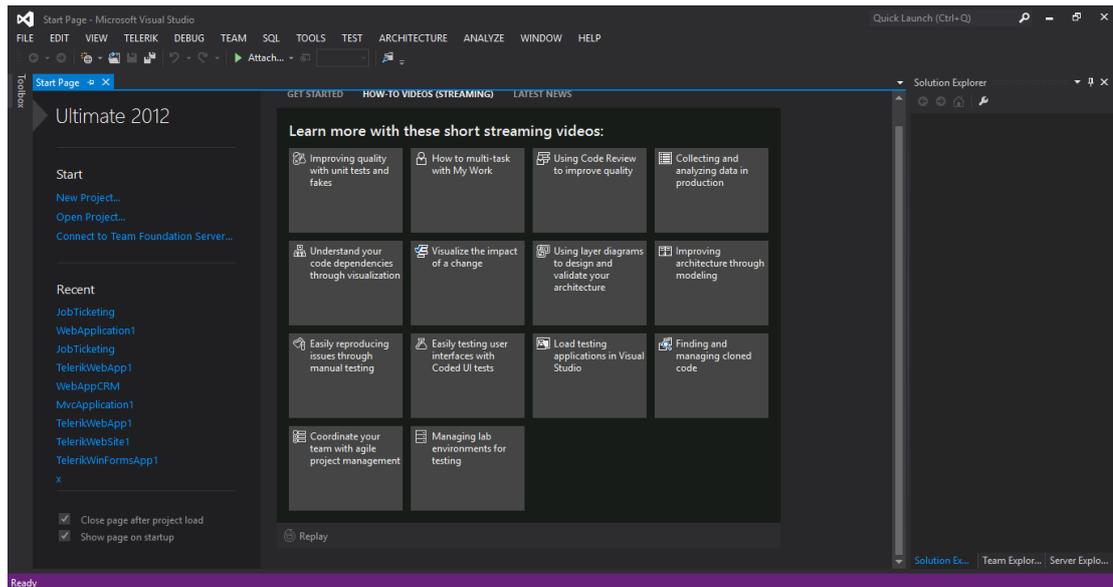
Dalam perkembangannya, *.NET* seringkali dikaitkan pula dengan versi Visual Studio yang sesuai dengan dukungan versi yang bersangkutan untuk pengembangan aplikasi.

f. Visual Studio 2012

Menurut Faisal (2014:5) “Visual Studio adalah *Integrated Development Environment* (IDE) dari untuk membangun aplikasi *console* dan *Graphical user interface* (GUI) dengan menggunakan bahasa yang didukung pada *.NET Framework*”. Aplikasi GUI yang dapat dibangun diantaranya adalah *Windows Form*, *Website*, *Web Application*, *Windows Mobile*. Visual Studio juga mempunyai *feature* untuk :

1. *Designer* antarmuka untuk *Winform*, *WPF* dan *Web*. Selain itu juga dapat digunakan untuk membuat *Class*, *Data* dan *Mapping*.
2. *Code editor* dengan dukungan *IntelliSense*.
3. *Debugger*.

Visual Studio 2012 adalah IDE dengan multi target *.NET Framework*, artinya developer dapat membangun aplikasi dengan menggunakan *.NET Framework* 2.0, 3.0, 3.5 dan 4.0.



Gambar II.1

Microsoft Visual Studio 2012

g. ASP.NET

ASP.NET adalah teknologi *Microsoft* yang dikhususkan untuk pengembangan aplikasi berbasis *web* dinamis berbasis *platform .NET Framework*. (Kurniawan, 2010:1). ASP.NET di desain untuk memberikan kemudahan pada pengembang *web* untuk membuat aplikasi berbasis *web* dengan cepat, mudah, dan efisien karena meminimalkan penulisan kode program dengan bantuan komponen-komponen yang tersedia, sehingga dapat meningkatkan produktivitas. ASP.NET memiliki beberapa kelebihan dibandingkan teknologi terdahulu antara lain:

1. Kemudahan mengakses berbagai *library .NET* secara konsisten dan *powerful* yang mempercepat pengembangan aplikasi.
2. Penggunaan berbagai pemrograman secara penuh, misalnya VB.NET, C#, J# dan C++. Dalam ASP.NET bahasa-bahasa ini dapat digunakan secara penuh sebagaimana layaknya bekerja di *Windows Application*.

3. Tersedia berbagai *Web Control* yang dapat digunakan membangun aplikasi secara cepat.
4. Model yang digunakan ASP.NET bukan model *request/respons*, melainkan model *event-driven*. Pada model ini, ASP.NET secara aktif memeriksa dan menunggu apa yang terjadi di *client*. Bila sesuatu terjadi, *server* dengan segera melakukan tindakan sesuai yang diinginkan oleh *client*.

h. Basis Data

Menurut Simarmata dan Paryudi (2010:1) “Basis Data adalah mekanisme yang digunakan untuk menyimpan informasi atau data”. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi, dan dihapus. Data adalah fakta mengenai objek, orang, dan lain-lain. Sedangkan Informasi adalah hasil analisis dan sintesis terhadap data. Sistem basisdata adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan. Simarmata dan Paryudi (2010:8) menyatakan "Sistem manajemen basisdata (DBMS) adalah perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar”. Kebutuhan akan sistem termasuk pula penggunaanya yang berkembang pesat. Alternatif penggunaan DBMS adalah menyimpan data dalam file dan menulis kode aplikasi tertentu untuk mengaturnya. Penggunaan DBMS memiliki beberapa manfaat penting. Ada pun keuntungan dan kerugian menggunakan DBMS dibawah ini:

1. Keuntungan DBMS

DBMS memungkinkan perusahaan maupun pengguna individu untuk

a. Mengurangi pengulangan data

Apabila dibandingkan dengan file-file komputer yang disimpan terpisah di setiap aplikasi komputer, DBMS mengurangi jumlah total file dengan menghapus data yang terduplikasi diberbagai file. Data terduplikasi selebihnya dapat ditempatkan dalam satu file.

b. Mencapai indenpedensi data

Spesifikasi data simpan dalam skema pada tiap program aplikasi. Perubahan dapat dibuat pada struktur data tanpa mempengaruhi program yang mengakses data.

c. Mengintegrasikan data beberapa file

Saat file dibentuk sehingga menyediakan kaitan logis, maka organisasi fisik bukan merupakan kendala. Organisasi logis, pandangan pengguna, dan program aplikasi tidak harus tercemin pada media penyimpanan fisik.

d. Mengambil data dan ifnromasi dengan cepat

Hubungan-hubungan logis, bahasa manipulasi data, serta bahasa query memungkinkan pengguna mengambil data dalam hitungan detik atau menit.

e. Meningkatkan keamanan

DBMS mainframe maupun komputer mikro dapat menyertakan beberapa lapis keamanan seperti kata sandi (*password*), direktori pemakai, dan bahasa sandi (*encryption*) sehingga data yang dikelola akan lebih aman.

2. Kerugian DBMS

Keputusan menggunakan DBMS mengikat perusahaan atau pengguna untuk:

a. Memperoleh perangkat lunak yang mahal

DBMS mainframe masih sangat mahal. Walaupun harga DBMS berbasis komputer mikro lebih murah, tetapi tetap merupakan pengeluaran besar bagi suatu organisasi kecil.

b. Memperoleh konfigurasi perangkat keras yang besar

DBMS sering memerlukan kapasitas penyimpanan dan memori lebih besar daripada program aplikasi lain.

c. Mempekerjakan dan mempertahankan staf DBA

DBMS memerlukan pengetahuan khusus agar dapat memanfaatkan kemampuannya secara penuh. Pengetahuan khusus ini disediakan paling baik oleh para pengelola basisdata (DBA).

i. UML (*Unified Modelling Language*)

1. Definisi Unified Modelling Language (UML)

Menurut Gata dan Gata (2013:4), “UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.”

Berdasarkan pendapat yang dikemukakan diatas dapat disimpulkan bahwa “*Unified Modelling Language (UML)* adalah sebuah bahasa standar industri untuk memvisualisasikan, menspesifikasikan, membangun dan

pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*)”.

2. Definisi Diagram-diagram UML (*Unified Modeling Language*)

Berikut ini adalah definisi mengenai 6 diagram UML yaitu:

a. *Use Case Diagram*

Menurut Gata dan Gata, (2013:4) menyatakan bahwa "Use case diagram merupakan pemodelan untuk kelakuan (behavior) system informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan system informasi yang akan dibuat".

b. *Class Diagram*

Menurut Gata dan Gata (2013:8) menyatakan bahwa "Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

c. *Sequence Diagram*

Menurut Gata dan Gata, (2013:7) “Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek”.

d. *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

e. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

f. *Deployment Diagram*

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

k. ***Black Box Testing***

Menurut Rosa dan Shalahudin (2013:50) “*Black Box Testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Black Box Testing bertujuan untuk menemukan kesalahan-kesalahan yang ada dalam suatu sistem, diantaranya:

- a. Fungsi-fungsi yang salah atau hilang
- b. Kesalahan *interface*
- c. Kesalahan dalam struktur data atau akses *database eksternal*
- d. Kesalahan performa
- e. kesalahan inisialisasi dan terminasi

Black box testing dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan

spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah:

- a. Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.2 Penelitian Terkait

Penelitian sejenis pernah dilakukan oleh Rico (2016), Rico melakukan penelitian analisa dan perancangan sistem informasi *IT-Helpdesk* pada PT. Lontar Papyrus Pulp & Paper Industry. Dalam penelitiannya itu menjelaskan bahwa *Department Information Technology* mempunyai standar operasional prosedur (SOP) dalam menerima dan mengatasi masalah dari beberapa user. Dari beberapa masalah yang diterima oleh bagian *Department Information Technolgy* sering mengalami kesulitan dalam menentukan dan mengatasi skala pengerjaanya (*emergency, urgent, dan normal*) dikarenakan tidak adanya sistem penentuan skala prioritas dan manajemen layanan service user terpusat. Penanganan masalah tersebut terkait erat dalam hal pelayanan, sehingga memerlukan pengelolaan yang baik. Pihak manajemen dalam memberikan suatu nilai pelayanan tertentu kepada *user / pengguna layanan* dapat melalui fasilitas seperti *Helpdesk, Problem Tracking, Trouble Call, Technical support & service, dan hotline support*. Metode pengembangan perangkat lunak merupakan suatu metode untuk menyusun suatu

perangkat lunak sebagai solusi dari permasalahan yang sedang diteliti. Model yang model yang telah digunakan adalah *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Dalam penelitian tersebut dapat ditarik kesimpulan sebagai berikut:

1. Sistem Informasi *IT-Helpdesk* dengan prioritas kerja yang dibangun dapat membantu *Department Information Technology* dalam menangani permasalahan yang berkaitan dengan perangkat Teknologi Informasi dan Komunikasi pada PT. Lontar Papyrus Pulp & Paper Industry.
2. *Helpdesk solution* merupakan solusi yang dapat memberikan kemudahan dalam melakukan pencarian, analisa dan minimalisasi adanya permasalahan.
3. Sistem yang dibangun dapat memberikan laporan mengenai *availability* dan *reliability* perangkat Teknologi Informasi dan Komunikasi.

Penelitian mengenai *Helpdesk Ticketing System* pernah juga dilakukan oleh Qoyyimah dkk (2012) pada PT. Primus Indojaya. Penelitiannya menjelaskan bahwa *Helpdesk ticketing system* yang terkomputerisasi sangat diperlukan oleh PT. Primus Indojaya, dikarenakan dalam menjalankan proses usaha perusahaan seringkali ditemukan masalah-masalah pada peralatan penunjang yang ada dilapangan dan harus segera diselesaikan. Adapun masalah yang berkaitan adalah tentang *hardware* dan *software*. Namun penanganan masalah yang saat ini terjadi pada *helpdesk ticketing system* di PT. Primus Indojaya masih kurang baik, karena proses pencatatan *helpdesk* yang masih manual dan sederhana sehingga mempengaruhi kinerja perusahaan tidak cepat teratasi, dan prosesnya tidak terkontrol. Semua pencatatan laporan bulanan serta perhitungan masalah masih dilakukan manual. Hal ini dapat memberikan dampak negatif, yakni: terdapat

masalah yang dicatat secara berulang, perhitungan masalah tidak akurat karena dilakukan manual dan perlunya manajer operasional menunggu untuk mendapatkan laporan yang diinginkan dan juga belum adanya sistem yang mampu mengukur tingkat masalah pada *helpdesk* dalam menangani permasalahan yang terjadi pada PT. Primus Indojoya. Oleh karena itu dibutuhkan sistem *helpdesk ticketing system* yang terintegrasi dengan baik dan cepat sehingga pengaksesan data pada *helpdesk* dapat dilakukan dengan mudah dan cepat guna pengukuran tingkat masalah pengaksesan laporan oleh manajer operasional, serta permasalahan dapat tertangani dengan baik dalam cakupan batasan masalah yang menghasilkan solusi tepat untuk mengatur sumberdaya yang ada. Penelitian tersebut berhasil membangun sistem informasi yang memiliki alur kerja bagian *helpdesk* login pada sistem dan membuat tiket yang berisi keluhan atau pengaduan customer, teknisi menerima tiket tersebut dan melakukan pekerjaan, kemudian supervisor bisa melihat laporan dari sistem tersebut. Solusi yang dihasilkan bagi perusahaan dari penelitiannya adalah sebagai berikut :

1. Proses pencatatan sudah terkomputerisasi dan langsung masuk ke sistem database.
2. Proses dokumentasi menjadi mudah dan aman.
3. Dapat memberikan informasi sesuai kebutuhan pengguna.
4. Pembuatan *format* slip gaji sudah terotomatisasi.
5. Pemecahan masalah yang *realtime*, dan keakuratan laporan bulanan.

Berdasarkan penelitian sebelumnya maka penulis membuat penelitian dengan beberapa pembaruan. Aplikasi tidak hanya mengelola data *helpdesk ticketing* tetapi juga mengelola data *customer*, data kontrak dan penambahan *fitur*

tracking permasalahan pada mesin, penambahan menu monitoring yang bisa diakses melalui *internet*, sehingga judul dikembangkan dari sistem informasi *helpdesk ticketing* menjadi sistem informasi pengelolaan pekerjaan teknisi. Dalam hal ini obyek penelitian yang diteliti adalah PT Samafitro.