

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka ini digunakan sebagai pembandingan antara penelitian yang sudah dilakukan dan yang akan dilakukan penulis di antaranya sebagai berikut:

Rinawati (2014) dalam skripsinya yang berjudul “Pengembangan Sistem Informasi Penggajian Karyawan PT. BPR Bumi Asih”. Penelitian ini menjelaskan bahwa PT. BPR Bumi Asih dengan jumlah karyawan 27 masih menggunakan cara konvensional untuk melakukan proses penghitungan gaji karyawan. Tujuan dari penelitian ini adalah untuk menghasilkan sistem penggajian karyawan yang terkomputerisasi sehingga membantu bendahara dalam melakukan proses penggajian dan pembuatan laporan yang cepat, akurat, efektif dan efisien. Penelitian ini berkonsentrasi pada pembuatan laporan cetak seperti laporan data karyawan, laporan data gaji karyawan dan laporan karyawan yang sudah gaji. Perancangan sistem tersebut menggunakan metode *Flow Map* sistem sedangkan pembuatan aplikasi memakai *Visual Basic 6.0* dan *SQL Server 2000* sebagai *database*.

Ardian (2011) dalam penelitiannya yang berjudul “Sistem Informasi Penggajian Karyawan Berbasis Web pada Kejaksaan Negeri Tangerang”. Penelitian ini menjelaskan bahwa kantor Kejaksaan Negeri Tangerang sering mengalami kesalahan dalam melakukan proses penghitungan gaji karyawan, persoalan yang muncul karena banyaknya 41 karyawan sehingga menyebabkan salah penghitungan dan pembayaran gaji yang tidak maksimal. Isi dari penelitian ini adalah membuat beberapa golongan karyawan, pembuatan master data karyawan, pembuatan transaksi pembayaran gaji, laporan golongan, laporan data karyawan dan laporan penggajian. Tujuan dari penelitian ini adalah untuk membuat sistem penggajian karyawan yang memberikan informasi data yang akurat dan laporan gaji karyawan secara transparan. Perancangan sistem pada penelitian menggunakan metode *Flow Map* sistem dan pembuatan aplikasinya menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *datasenya*.

Suryo (2012) dengan skripsinya yang berjudul “Sistem Informasi Penggajian Pada Primagama Cabang Rungkut Surabaya”. Penelitian ini menjelaskan bahwa Primagama cabang rungkut masih kesulitan dalam *maintenance* data yang masih menggunakan cara manual, sehingga pengolahan penggajian membutuhkan waktu yang cukup lama yang mengakibatkan sering terjadinya keterlambatan pada saat proses pembayaran gaji. Tujuan penelitian ini ialah membuat sistem informasi penggajian karyawan yang lebih terkomputerisasi agar proses penghitungan gaji tidak lagi membutuhkan waktu yang lama sehingga proses pembayaran gaji tidak terjadi keterlambatan. Perancangan sistem pada penelitian ini menggunakan metode *Flowchart* dan pembuatan aplikasinya menggunakan *VB.Net* dengan *database Microsoft Acces*.

Dari beberapa tinjauan pustaka di atas ada beberapa perbedaan dengan penelitian yang akan dilakukan penulis yaitu:

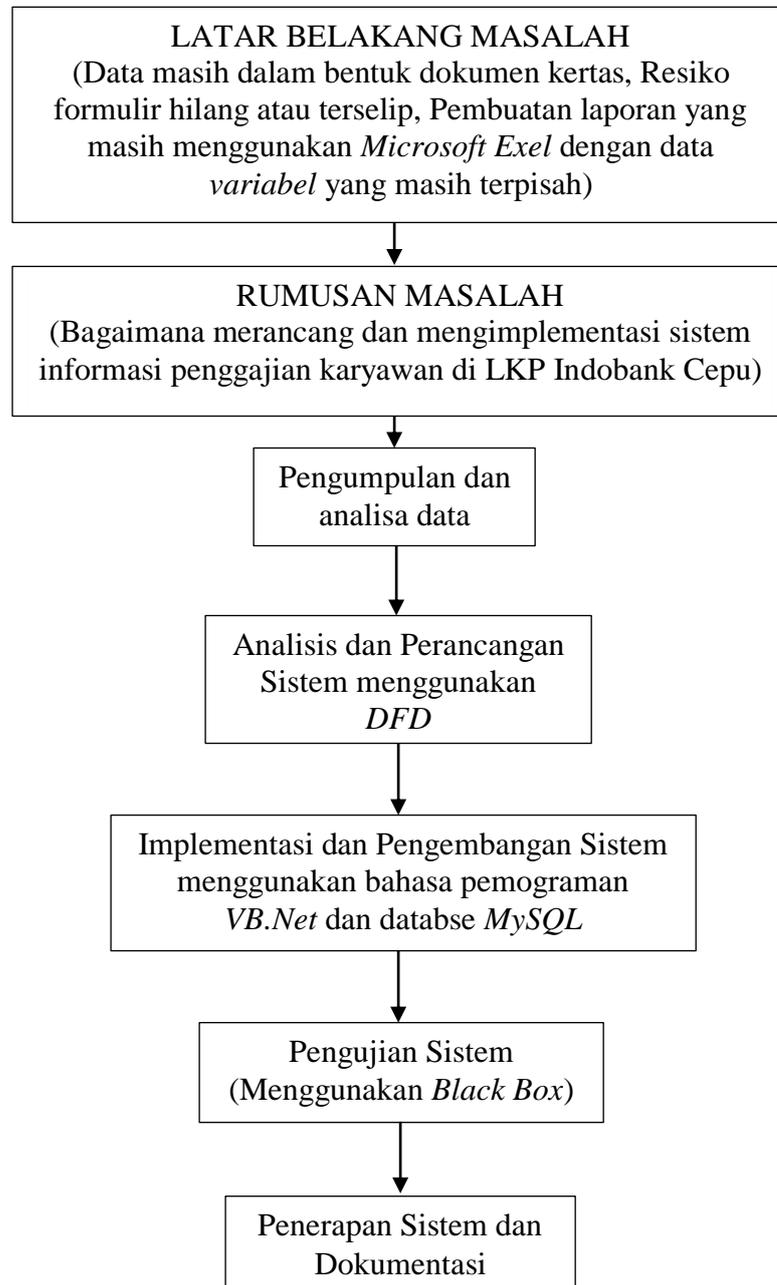
- 1) Rinawati: Dalam penelitiannya menggunakan *Visual Basic 6.0* sedangkan penelitian yang akan dilakukan menggunakan *VB.Net*.
- 2) Purnamasari: Perbedaannya penelitian purnamasari menggunakan bahasa pemograman *PHP* sedangkan penulis menggunakan bahasa pemograman *VB.Net*
- 3) Suryo: Dalam penelitiannya, pembuatan basis datanya menggunakan *Microsoft Acces* sedangkan penulis menggunakan *Mysql*.

Persamaan penelitian terdahulu dan penelitian yang akan dilakukan penulis adalah sama-sama bertujuan untuk memanfaatkan sistem informasi semaksimal mungkin, agar lebih memudahkan saat proses penggajian.

Penulis berharap dapat merancang dan membuat sistem informasi penggajian yang dapat melengkapi atau menyempurnakan dari penelitian yang terdahulu.

2.2 Kerangka Pemikiran

Kerangka pemikiran dalam penelitian ini dapat digambarkan pada Gambar 2.1.



Gambar 2. 1. Kerangka Pemikiran

Keterangan Diagram Kerangka Pemikiran adalah sebagai berikut:

1) Latar Belakang Masalah

Identifikasi adanya pengolahan sistem secara manual yang menjadi sebuah permasalahan di LKP Indobank Cepu. Permasalahan yang dibahas merupakan permasalahan pendataan karyawan yang hanya meliputi proses *input*, hitung gaji dan pembuatan laporan.

2) Rumusan Masalah

Bagaimana merancang dan mengimplementasi sistem informasi penggajian Karyawan LKP Indobank Cepu.

3) Pengumpulan dan Analisa data

Semua data yang di butuhkan dikumpulkan dan dianalisa, baik melalui *interview* dengan pihak terkait antara lain dengan Bapak EdySuwito SE., MM selaku direktur, Ibu Siti Wahidah SE dan beberapa karyawan. Selain itu juga melakukan observasi sistem yang sedang berjalan saat ini.

4) Analisis dan Perancangan Sistem

Analisis dan perancangan sistem pada tugas akhir ini menggunakan metode *DFD*.

5) Implementasi dan Pengembangan Sistem

Menerapkan dan membangun sistem yang telah dirancang, mulai dari penulisan kode program menggunakan aplikasi *VB.Net*, dan *database MySQL*.

6) Pengujian Sistem

Pengujian sistem ini dilakukan bertujuan untuk mengetahui jika ternyata masih ada kesalahan ataupun kekurangan pada sistem yang telah dibuat. Metode pengujian sistem yang digunakan adalah *black box*.

7) Penerapan Sistem dan Dokumentasi

Sistem telah siap digunakan setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan kegiatan penyusun tugas akhir.

2.3 Teori Pendukung.

2.3.1 Sistem Informasi Penggajian

2.3.1.1 Pengertian Sistem

Pengertian Sistem menurut Ardian (2011) dalam penelitiannya menyebutkan Sistem Informasi adalah: “Kumpulan atau sekelompok dari sistem atau bagian atau komponen apapun baik berbentuk fisik maupun *non*-fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”.

2.3.1.2 Pengertian Sistem Informasi

Sistem informasi merupakan suatu perkumpulan data yang terorganisasi beserta tatacara penggunaannya yang mencakup lebih jauh dari pada sekedar penyajian. Keberhasilan suatu sistem informasi yang diukur berdasarkan maksud pembuatannya tergantung pada tiga faktor utama, yaitu: keserasian dan mutu data, pengorganisasian data, dan tata cara penggunaannya. Untuk dapat menggabungkan data yang berasal dari berbagai sumber suatu sistem alih rupa (*transformation*) data sehingga jadi tergabungkan (*compatible*). Berapa pun ukurannya dan apapun ruang lingkupnya suatu sistem informasi perlu memiliki ketergabungan (*compatibility*) data yang disimpannya (Al Fatta, 2010).

2.3.1.3 Pengertian Sistem Penggajian

Sistem penggajian adalah mengembangkan sekumpulan prosedur yang memungkinkan perusahaan untuk menarik, menahan dan memotivasi staf yang diperlukan, serta untuk mengendalikan biaya pembayaran gaji. Karena tidak ada satu pola yang dapat digunakan secara *universal* maka prosedur ini harus disesuaikan dengan kebijakan gaji tiap-tiap organisasi, dan hendaknya didasari atas kebijakan yang dianggap adil (Mulyadi, 2010). Sistem penggajian mempunyai beberapa prosedur yaitu:

1) **Prosedur pencatatan waktu hadir**

Prosedur ini bertujuan untuk mencatat waktu hadir karyawan. Pencatatan waktu hadir ini diselenggarakan oleh fungsi pencatat waktu dengan menggunakan daftar hadir pada pintu masuk kantor administrasi atau pabrik.

Pencatatan waktu hadir karyawan ini diselenggarakan untuk menentukan gaji dan upah karyawan.

2) Prosedur pencatat waktu kerja

Dalam perusahaan manufaktur yang produksinya berdasarkan pesanan, pencatat waktu kerja diperlukan bagi karyawan yang berkerja di fungsi produksi untuk keperluan distribusi biaya dan upah karyawan kepada produk atau pesanan yang menikmati jasa karyawan tersebut. Jika misalnya seorang karyawan pabrik hadir ke perusahaan selama 7 jam dalam suatu hari kerja, jumlah jam hadir tersebut dirinci menjadi waktu kerja dalam tiap-tiap pesanan yang dikerjakan. Dengan demikian waktu kerja ini dipakai sebagai dasar pembebanan biaya tenaga kerja langsung kepada produk yang diproduksi.

3) Prosedur pembuatan daftar gaji

Dalam prosedur ini fungsi pembuat daftar gaji dan upah membuat daftar gaji dan upah karyawan. Data yang dipakai sebagai dasar pembuatan daftar gaji adalah surat-surat keputusan mengenai pengangkatan karyawan baru, kenaikan pangkat, penurunan pangkat, pemberhentian karyawan, daftar gaji bulan sebelumnya dan daftar hadir.

4) Prosedur pembayaran gaji

Prosedur pembayaran gaji dan upah melibatkan fungsi akuntansi dan fungsi keuangan. Fungsi akuntansi membuat perintah pengeluaran kas kepada fungsi keuangan untuk menulis cek guna pembayaran gaji dan upah. Fungsi keuangan kemudian menguangkan cek tersebut ke bank dan memasukan uang ke amplop gaji dan upah. Jika jumlah karyawan perusahaan banyak. Pembagian amplop dan upah dapat dilakukan dengan membagikan cek gaji dan upah kepada karyawan.

2.3.1.4 Pengertian Penggajian

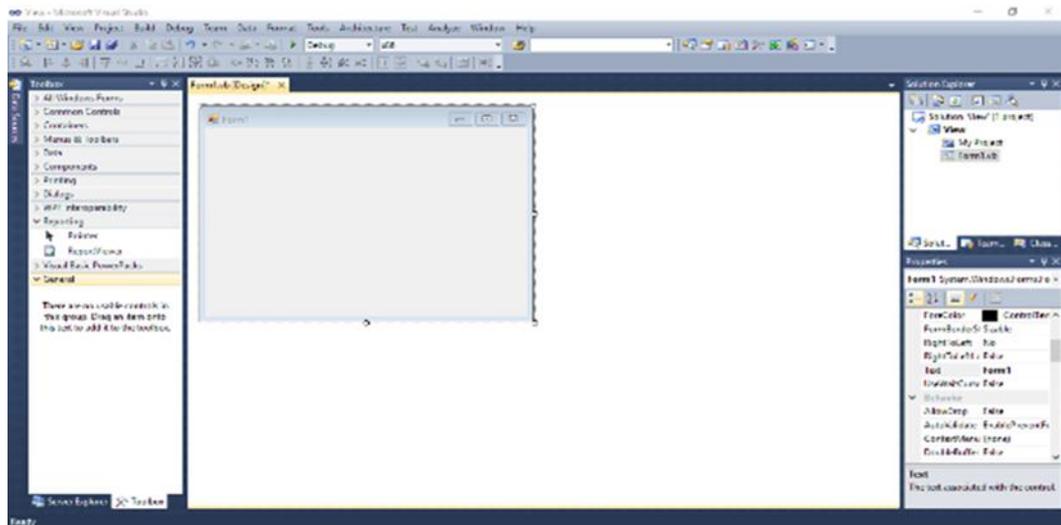
Dalam jurnal Damayanti (2013) menyebutkan, gaji adalah sebuah komponen yang mutlak dikeluarkan oleh perusahaan sebagai kompensasi bagi karyawan, yang mana hal ini untuk menjamin keberlangsungan perusahaan itu sendiri. Berikut ini contoh pendekatan dalam penentuan elemen gaji. Secara

umum sasaran penggajian ditujukan karena pertimbangan beberapa hal sebagai berikut:

- 1) *Pay for Position*, yaitu penetapan gaji karyawan disebabkan pertimbangan nilai suatu pekerjaan yang menjadi tanggung jawabnya.
- 2) *Pay for Person*, yaitu penetapan gaji karyawan disebabkan pertimbangan orangnya yang dapat dibedakan dari aspek kompetensinya atau juga karena faktor kelangkaan ketersediaan orang dengan kualifikasi yang sesuai di pasar tenaga kerja.
- 3) *Pay for Performance*, adalah penetapan gaji karyawan disebabkan pertimbangan pencapaian kinerja seseorang.
- 4) *Pay for Year of Service*, adalah penetapan gaji karyawan disebabkan pertimbangan masa kerja seseorang.

2.3.2 Visual Basic .NET

Menurut Stefano (2014), *Visual Basic Net* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *Net Framework*, dengan menggunakan bahasa *Basic* (Marlon, 2010). Dengan menggunakan alat ini, para programmer dapat membangun aplikasi *Windows Forms*, Aplikasi web berbasis *Asp Net*, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#*), atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio*. Bahasa *Visual Basic Net* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang di implementasikan di atas *Net Framework*. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.

Gambar 2. 2. Tampilan *Vb.Net*

Program-program yang ditulis dengan *VB.Net* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi luar tambahan. Dalam pemrograman untuk bisnis, *VB.Net* memiliki pasar yang sangat luas. Dalam sebuah survei yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Basic*, yang diikuti oleh *C++*, *JavaScript*, *C#*, dan *Java*. Berikut ini Gambar 2.3 merupakan contoh kode *VB.Net*

```

1  Sub pesan()
2      Dim nama As String
3      Dim alamat As String
4      nama = Val(txtNama.Text)
5      alamat = Val(txtAlamat.Text)
6      Label3.Text = "Terimakasih" & Chr(13) & "Nama anda : " & txtNama.Text & Chr(13) & "Alamat : " & txtAlamat.
       Text
7  End Sub
8  Sub kosong()
9      txtNama.Text = ""
10     txtAlamat.Text = ""
11     Label3.Text = ""
12 End Sub
13

```

Gambar 2. 3. Contoh Kode *Vb.Net*

2.3.2.1 MySQL

Menurut Kustiyahningsih (2011), “*MySQL* adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel”. *MySQL* dalam operasi *client-server* melibatkan server *daemon MySQL* di sisi server dan berbagai macam program serta *library* yang berjalan di sisi *client*. *MySQL* mampu menangani data yang cukup besar. *SQL* adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini awalnya dikembangkan oleh *IBM*, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan *SQL*,

proses akses database menjadi lebih *user-friendly* dibandingkan dengan menggunakan *dBASE* atau *clipper* yang masih menggunakan perintah-perintah pemrograman. *MySQL* merupakan program yang *multi-threaded*, sehingga dapat dipasang pada *server* yang memiliki *multi-CPU*.

Berikut ini contoh kode dalam bentuk Gambar 2.4 untuk menghubungkan *MySQL* ke *VB.Net*:

```

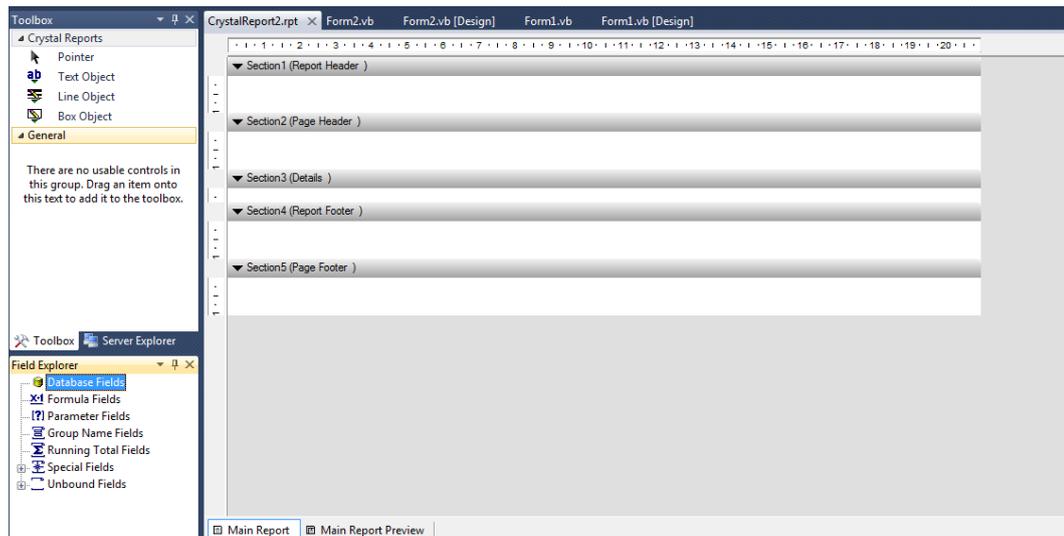
1 Imports MySql.Data.MySqlClient.MySqlConnection
2 Imports System.Data
3 Public Class Form1
4
5     '-- MySQL Connection
6     Public connDB As New MySql.Data.MySqlClient.MySqlConnection
7
8     Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
9         Call conecDB()
10        'Close Database
11        connDB.Close()
12    End Sub
13
14    Public Sub conecDB()
15        Try
16
17            'This is the server IP/Server name. If server is intalled on your local machine, your IP should be
18            '127.0.0.1 or you may use localhost
19            Dim strServer As String = "localhost"
20            Dim strDbase As String = "mysql" 'Database name
21            Dim strUser As String = "root" 'Database user
22            Dim strPass As String = "" 'Database password
23
24            'MySQL Connection String
25            If connDB.State <> ConnectionState.Open Then connDB.ConnectionString = "server=" & strServer.Trim &
26            ";database=" & strDbase.Trim & ";user=" & strUser.Trim & ";password=" & strPass
27            If connDB.State <> ConnectionState.Open Then connDB.Open() Else connDB.Close()
28            MsgBox("Connected")
29        Catch ex As Exception
30            MsgBox(ex.Message)
31        End Try
32    End Sub
33 End Class

```

Gambar 2. 4. Contoh Kode Mysql

2.3.2.2 Crystal Report

Menurut Madcoms (2010), *Crystal Report* merupakan program yang terpisah dengan program Microsoft Visual Basic, tetapi keduanya dapat dihubungkan (*Linkage*). Membuat laporan dengan *Crystal Report* hasilnya lebih baik dan lebih mudah, karena pada *Crystal Report* banyak tersedia objek-objek maupun komponen yang mudah digunakan. Tampilan *Crystal Report* dapat dilihat pada Gambar 2.5 berikut.



Gambar 2. 5. Tampilan *Crystal Report*

Crystal Report merupakan salah satu program yang memiliki spesialisasi untuk membuat laporan dengan tidak dapat melakukan *input* di dalamnya, *Crystal Report* mampu membaca sumber data yang dihasilkan melalui *Microsoft Access*, *Fox Pro*, *SQL* ataupun program-program *database* lainnya. Gambar 2.6 di bawah ini merupakan kode untuk menghubungkan *crystal report* ke *vb.net*

```

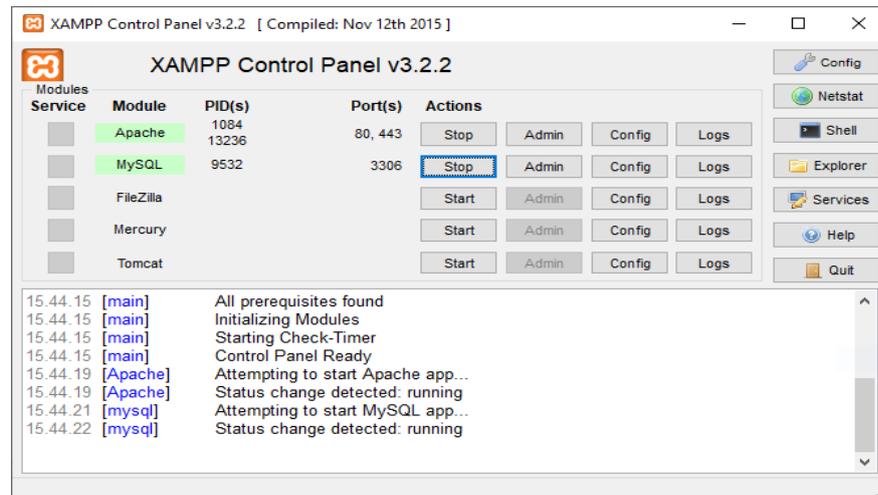
1 CrystalReport1.ReportFileName = App.Path & "\nama file crystalreport.rpt"
2 CrystalReport1.WindowState = crptMaximized
3 CrystalReport1.RetrieveDataFiles
4 CrystalReport1.Action = 0
5

```

Gambar 2. 6. Contoh Kode *Crystal Report*

2.3.3 Pengertian XAMPP

XAMPP adalah sebuah *software web server apache* yang di dalamnya sudah tersedia *database server MySQL* dan dapat mendukung pemrograman *PHP*. *XAMPP* merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support (PHP 4 dan PHP 5)* dan beberapa modul lainnya (Februariyanti, 2012). Berikut tampilan *xampp* ditunjukkan pada Gambar 2.7.



Gambar 2. 7. Tampilan *Xampp*

2.3.4 Pengembangan Sistem Testruktur

Pengembangan sistem terstruktur adalah metode yang menggunakan aliran data atau informasi yang mengalir pada suatu sistem (Kristanto, 2012). Sedangkan alat yang digunakan adalah sebagai berikut:

2.3.4.1 Diagram Kontek

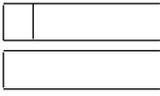
Menurut Nugroho (2013), Diagram kontek merupakan alat bantu dalam perancangan global bagi program aplikasi yang akan dibuat, dengan tujuan untuk mencerminkan keadaan sistem yang akan dibangun secara umum.

Diagram kontek berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram kontek ini berisi “siapa saja yang memberi data (dan apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem”.

2.3.4.2 *Data Flow Diagram (DFD)*

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Berikut adalah simbol-simbol DFD bisa dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol-Simbol *Data Flow Diagram*

NO	NAMA SIMBOL	SIMBOL	KETERANGAN
1	Proses		Menunjukkan sebagai proses komputerisasi
2	Alir		Digunakan untuk menunjukkan arus dari proses
3	Penyimpanan		kumpulan data atau paket data hasil dari proses maupun penyediaan data untuk proses
4	<i>Link</i>		Digunakan untuk menghubungkan <i>entity</i> dengan atribut

DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem. Ada dua teknik dasar DFD yang umum dipakai yaitu *Gane and Sarson* dan *Yourdon and De Marco* (Kristanto, 2012).

2.3.4.3 *Flowchart*

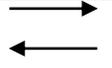
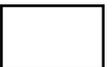
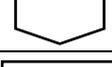
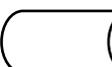
Menurut Indrajani (2017), *flowchart* adalah bagian yang menunjukkan aliran data didalam program atau prosedur sistem secara logika. Tujuan utama dari penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, dan jelas. Tahap penyelesaian masalah yang disajikan harus jelas, efektif, dan tepat.

Indrajani menyarankan untuk membuat model *flowchart* yang berbeda sesuai dengan perspektif pemakaian, sehingga dikenal ada empat jenis diagram alir secara umum:

- 1) Diagram Alir Dokumen, menunjukkan kontrol dari sistem aliran dokumen.
- 2) Diagram Alir Data, menunjukkan kontrol dari sebuah sistem aliran data.
- 3) Diagram Aliran Sistem, menunjukkan kontrol dari sistem aliran secara fisik.
- 4) Diagram Aliran Program, menunjukkan kontrol dari program sebuah sistem.

Berikut adalah simbol-simbol *flowchart* bisa dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		<i>Terminator</i>	Menandakan awal atau akhir dari suatu sistem.
2		Garis Alir	Arah aliran sistem.
3		<i>Input/Output</i>	Menyatakan simbol <i>input</i> dan <i>output</i> tanpa tergantung jenis peralatannya.
4		<i>Process Computer</i>	Menunjukkan kegiatan dari operasi program komputer.
5		<i>Decision</i>	Menggambarkan logika keputusan dengan nilai <i>true</i> atau <i>false</i> .
6		<i>On Page Connector</i>	Menunjukkan hubungan di halaman yang sama.
7		<i>Off Page Connector</i>	Menunjukkan hubungan di halaman yang berbeda.
8		<i>Document</i>	Merupakan simbol untuk data yang berbentuk kertas maupun informasi.
9		<i>Disk</i>	Simbol untuk menyatakan input yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i> .
10		<i>Manual Operation</i>	Untuk menunjukkan pekerjaan yang dilakukan secara manual.
11		Arsip	Menunjukkan <i>file non</i> komputer yang diarsip urut angka.

2.3.4.4 Entity Relationship Diagram (ERD)

ERD berguna untuk memodelkan sistem yang nantinya akan dikembangkan basis datanya. Model ini juga membantu perancang basis data karena model ini dapat menunjukkan jenis data yang dibutuhkan dan kerelasian antar data di dalamnya. Bagi pemakai, model ini sangat membantu dalam hal pemahaman model sistem dan rancangan basis data yang akan dikembangkan oleh perancang basis data.

Penentuan entitas dalam suatu sistem perlu dilakukan dengan cermat dan hati-hati. Tidak semua orang, benda atau hal dapat disebut entitas. Hanya orang, benda, dan hal yang terkait dengan sistem dan keterangannya perlu disimpan dalam basis data saja yang dapat disebut entitas (Kristanto, 2012).

Derajat relasi atau kardinalitas rasio adalah jumlah maksimum hubungan antara satu entitas dengan entitas lainnya. Hubungan dalam ERD umum terbagi menjadi 3 tipe yaitu:

1) *One to one* (1:1)

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

2) *One to Many* (1:M)

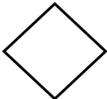
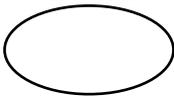
Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

3) *Many to Many* (M:M)

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

Hubungan ini ditentukan oleh *primary key*, *foreign key* dan *candidate key* yang ada pada tabel *database*. Berikut adalah simbol ERD bisa dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol-Simbol *ERD*

No	Simbol	Nama	Proses
1		Entitas	Simbol berfungsi untuk memberikan identitas pada entitas yang memiliki <i>label</i> dan nama.
2		Relasi	Simbol yang berfungsi untuk mengetahui jenis hubungan yang ada antara dua <i>file</i> .
3		Atribut	Karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut.
4		Alur	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

2.4 Pengujian Sistem

Menurut Quadri dan Farooq (2010), pengujian sistem adalah proses verifikasi dan validasi apakah sebuah aplikasi *software* atau program memenuhi persyaratan bisnis dan persyaratan teknis yang mengarahkan desain dan pengembangan beserta cara kerjanya seperti yang diharapkan dan juga mengidentifikasi kesalahan yang penting yang digolongkan berdasarkan tingkat *severity* pada aplikasi yang

harus diperbaiki. Teori pengujian dengan metode *Black Box* dapat dilihat pada Tabel 2.4

Tabel 2. 4 Teori Pengujian Metode *Black Box*

No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Status
1	Mengosongkan semua isian data login, lalu mengklik tombol 'Masuk'.	Nama: - Kata sandi: -	Sistem menolak akses login dan menampilkan pesan "Mohon isi nama dan kata sandi"	Sesuai harapan	Valid
2	Hanya mengisi nama admin dan mengosongkan data kata sandi, lalu mengklik tombol 'Masuk'.	Nama: admin Kata sandi: -	Sistem menolak akses login dan menampilkan pesan "Mohon isi dulu data yang masih kosong"	Sesuai harapan	Valid
3	Hanya mengisi data kata sandi dan mengosongkan nama admin, lalu mengklik tombol 'Masuk'.	Nama: - Kata sandi: 123	Sistem menolak akses login dan menampilkan pesan "Mohon isi dulu data yang masih kosong"	Sesuai harapan	Valid