

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1. Analisis Sistem**

Analisis sistem adalah kegiatan penguraian suatu sistem informasi yang utuh dan nyata mengenai konsep-konsep dasar sistem sehingga memberikan solusi pemecahan masalah dalam merancang dan mengimplementasikan sistem yang dibuat. Tahap ini bertujuan untuk mengetahui mekanisme aplikasi, proses-proses yang terlibat dalam aplikasi serta hubungan-hubungan proses dalam aplikasi tersebut. Hasil dari sistem ini adalah menjadi sebuah aplikasi yang dapat memberikan informasi yang berguna dan bermanfaat bagi pengguna.

Jaringan Saraf Tiruan (JST) adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel saraf biologi, sama seperti otak manusia yang memproses suatu informasi. Ada banyak metode JST yang dapat diimplementasikan. Masing-masing metode JST memiliki kelebihan dan kekurangan.

*Backpropagation* adalah algoritma pembelajaran terawasi yang menggunakan pola penyesuaian bobot untuk mencapai nilai kesalahan minimum antara keluaran hasil dengan keluaran yang nyata dan memiliki satu atau lebih *layer* tersembunyi. Metode *Backpropagation* banyak digunakan untuk peramalan.

Dalam pembelajaran terawasi, terdapat sejumlah pasangan data (masukan- target - keluaran) yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan.

### 3.1.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem ini dilakukan dalam dua bentuk antara lain :

#### 3.1.1.1. Analisis Peran Sistem

Analisis peran sistem bertujuan untuk memberikan informasi mengenai aplikasi yang dibuat agar dapat melayani kebutuhan pengguna sistem. Dalam aplikasi yang dibuat, fungsi utama sistem ini berada pada pelatihan dan pengujian data gaji pegawai negeri sipil di SMP N 4 Kupang menggunakan metode jaringan syaraf tiruan yakni metode *Backpropagation*.

#### 3.1.1.2. Analisis Peran Pengguna

Sistem yang akan dibangun ini akan digunakan oleh pihak kantor SMP N 4 Kupang, lebih khususnya kepada Bendahara yang bertugas melakukan penggajian terhadap pegawai di SMP N 4 Kupang.

#### 3.1.1.3. Analisis Metode Jaringan Syaraf Tiruan

JST merupakan salah satu sistem pemrosesan informasi atau data yang didisain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses

belajar melalui perubahan bobot sinapsisnya. Metode yang digunakan untuk penelitian ini adalah metode *Backpropagation*.

#### 1. Analisis Metode *Backpropagation*

*Backpropagation* adalah algoritma pembelajaran untuk memperkecil tingkat *error* dengan cara menyesuaikan bobotnya berdasarkan perbedaan *output* dan target yang diinginkan. *Backpropagation* termasuk *multilayer network* yang merupakan perkembangan dari *single layer network* yaitu *Perceptron*.

##### a. Arsitektur Jaringan Metode *Backpropagation*

Arsitektur jaringan metode *Backpropagation* terdiri dari tiga lapisan dalam proses pembelajarannya, yaitu lapisan *input*, *hidden* dan *output*.

##### 1. *Input Layer*

*Input layer* berfungsi untuk menerima nilai dari ciri pola *input* dari *neuron* pada *input layer* adalah numerik. Banyaknya *neuron* pada *input layer* dipengaruhi oleh banyaknya pengambilan pola *input* yang akan dimasukan kedalam JST.

##### 2. *Hidden Layer*

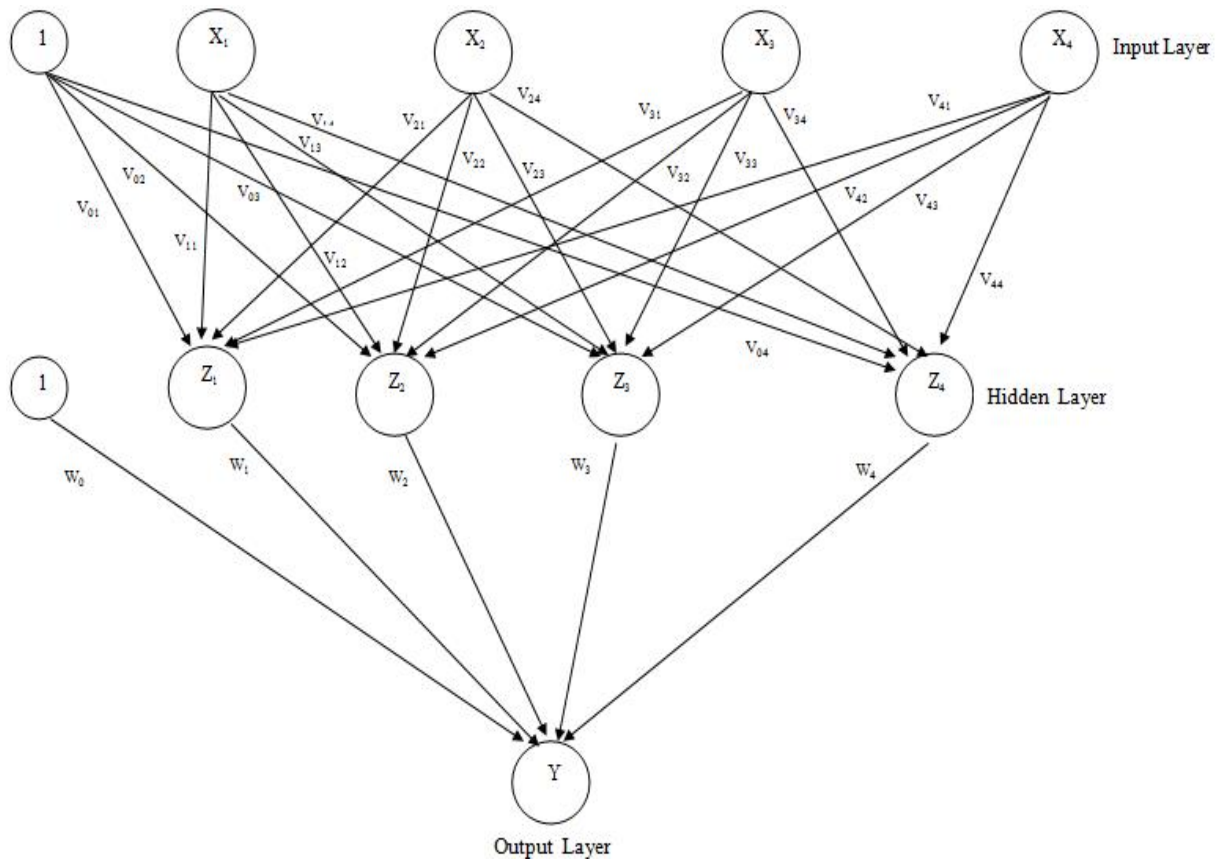
Pada *hidden layer* terjadi proses komputasi terhadap bobot dan bias dan dihitung pula besarnya *output* dari

*hidden layer* tersebut berdasarkan fungsi aktivasi tertentu. Untuk menentukan jumlah *hidden layer* tidak ada ketentuan yang pasti. Namun jumlah *hidden layer* berpengaruh terhadap akurasi dan kecepatan. Dengan adanya *hidden layer* dapat menyebabkan tingkat *error* pada *Backpropagation* lebih kecil.

### 3. *Output Layer*

Banyaknya *layer* yang digunakan adalah 1 *layer*. Nilai *neuron* merupakan bilangan *sigmoid biner*. *Output layer* terdiri dari beberapa *neuron input*. Kombinasi dari semua *neuron* tersebut dapat digunakan untuk mengidentifikasi dan juga *learning* sebagai *output* yang seharusnya.

Berikut tampilan Arsitektur Jaringan Metode *Backpropagation* :



Gambar 3.1. Arsitektur Jaringan Metode *Backpropagation*

Keterangan :

$X_1$  = *Input* terdiri dari 4 *neuron* yaitu gaji pokok ( $x_1$ ),  
tunjangan ( $x_2$ ), potongan ( $x_3$ ) dan hutang ( $x_4$ ).

$V_{ij}$  = Bobot pada *hidden layer*

$V_{0j}$  = Bias pada *hidden layer*

$W_i$  = Bobot pada *output layer*

$W_0$  = Bias pada *output layer*

$i, j = 1, 2, 3, \dots, n$

$n$  = Jumlah *neuron* dalam suatu lapisan

$Y$  = *Output* terdiri dari 1 *neuron* yaitu gaji diterima

$1$  = Konstanta bias

#### 3.1.1.4. Analisis Data

Secara umum data yang akan menjadi masukan dalam sistem peramalan adalah data historis. Data historis digunakan sebagai variabel masukan untuk menentukan atau meramalkan jumlah gaji diterima pada bulan berikutnya. Data historis dibagi menjadi dua yaitu data simulasi dan data pengujian. Data historis diambil pada tahun 2015 dengan 4 variabel masukan yaitu gaji pokok, tunjangan, potongan, dan hutang. Dan terdapat 1 variabel sebagai target. Data yang dikumpulkan adalah data historis dari bulan Januari sampai Desember 2015. Data bersumber dari Bendahara SMP N 4 Kupang untuk tahun 2015.

### 3.1.1.5. Analisis Kriteria *Warning*

$x$  = Besar gaji diterima

Tabel 3.1. Tabel Analisis Kriteria *Warning*

Kriteria	Keterangan
$x < 1.500.000$	Warning ! Tidak bisa melakukan kredit tambahan
$1.600.000 \leq x \leq 2.000.000$	Dapat melakukan kredit tambahan dengan angsuran minimal 100.000 dan maksimal 600.000
$2.000.000 \leq x \leq 2.500.000$	Dapat melakukan kredit tambahan dengan angsuran minimal 700.000 dan maksimal 1.000.000
$x > 2.500.000$	Dapat melakukan kredit tambahan dengan angsuran minimal 1000.000 dan maksimal 1500.000

## 3.2. Sistem Perangkat Pendukung

### 3.2.1. Analisis Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) merupakan salah satu komponen komputer yang berfungsi untuk mendukung proses komputerisasi yang bekerja berdasarkan perintah yang telah ditentukan oleh pengguna. Perangkat keras (*hardware*) bagi menjadi beberapa bagian yaitu unit masukan (*input device*), unit pemrosesan (*process device*), unit keluaran (*output device*), unit penyimpanan (*backing storage*) dan unit tambahan (*peripheral*). Komponen perangkat keras (*hardware*) yang digunakan dalam pembuatan aplikasi antara lain :

1. *Processor Intel® Pentium® Dual T4500*
2. *RAM 2 GB*
3. *Hard Disk 500GB*
4. *Printer*
5. *Keyboard*
6. *Mouse*

### 3.2.2. Analisis Perangkat Lunak (*Software*)

Perangkat lunak (*software*) merupakan program komputer yang merupakan suatu susunan instruksi yang akan dijalankan oleh perintah, maupun catatan – catatan yang diperlukan oleh komputer untuk menjalankan perintah yang dijalankannya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi ini adalah sebagai berikut :

1. *Sistem Operasi Windows 7*
2. *Bahasa Pemrograman JAVA Netbeans 8.1.*
3. *Database MySQL Xampp V3.2.2.*
4. *Microsoft Office 2007*

### 3.3. Perancangan Sistem

Perancangan aplikasi ini di bangun bertujuan untuk memudahkan proses implementasi sistem yang akan dibuat.

Perancangan aplikasi ini dibangun bersifat *Object Oriented* dengan menggunakan *Unified Modeling Language* (UML) sebagai bahasa pemodelan.



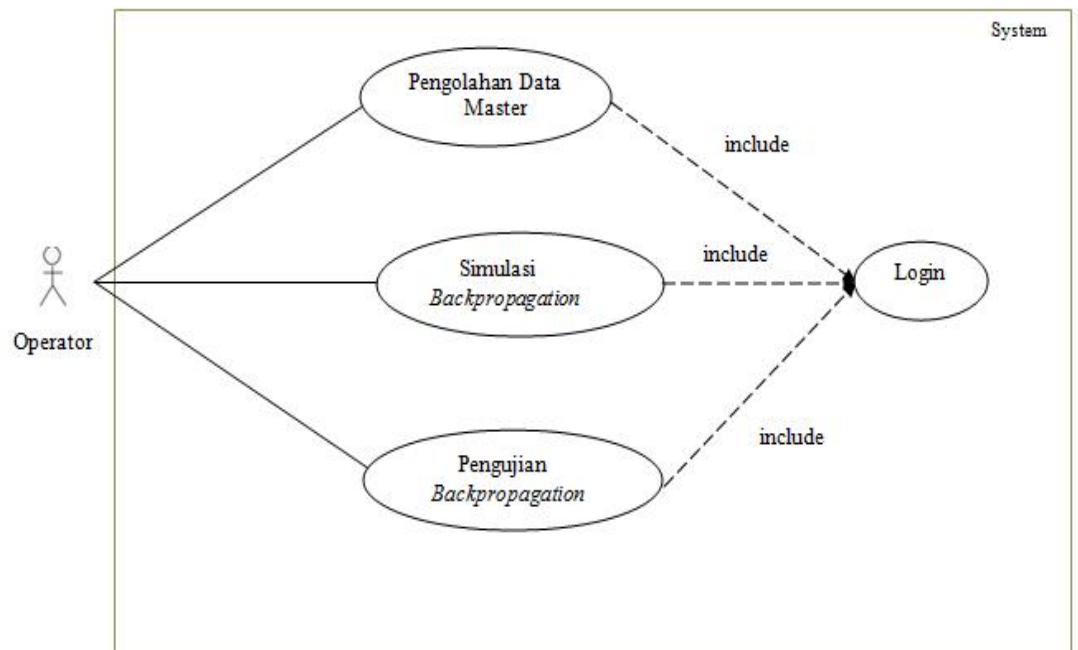
### 3.3.1. Model *Use Case*

Model *Use Case* menjelaskan mengenai aktor – aktor yang terlibat dengan perangkat lunak yang dibangun beserta proses – proses yang ada didalamnya.

#### 3.3.1.1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk menggambarkan kalakuan sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem.

Berikut adalah gambaran *use case diagram* :



Gambar 3.2. *Use Case Diagram Backpropagation*

### 3.3.1.2. Use Case Skenario

*Use Case* skenario menjelaskan skenario dari setiap proses yang terdapat pada *use case diagram*. Berikut ada tabel dari *Use Case Skenario* :

Tabel 3.2. *Use Case Skenario Pengolahan Data Master*

Nama <i>Use Case</i>	Pengolahan Data Master
Deskripsi	Pengguna Mengolah Data Master
Aktor	Pengguna / Bendahara
<i>Extend</i>	Simpan, Ubah, Hapus, Baru, Keluar
Reaksi Aktor	
Aksi Aktor	Reaksi Sistem
1. Pengguna masuk dan berada pada tampilan penggajian metode JST 2. Pengguna memilih menu data master	3. Menampilkan form menu data master

Tabel 3.3. *Use Case Skenario Simpan Data Master*

Nama <i>Use Case</i>	Simpan Data Master
Deskripsi	Pengguna Memasukan Data
Aktor	Pengguna

Reaksi Aktor	
Aksi Aktor	Reaksi Sistem
1. Mengisi pola data baru 2. Memilih tombol <b>Simpan</b>	3. Menyimpan ke dalam <i>database</i> 4. Menampilkan pesan “Data sukses disimpan”

Tabel 3.4. *Use Case* Skenario Edit Data Master

Nama <i>Use Case</i>	Edit Data Master
Deskripsi	Pengguna Mengubah Data
Aktor	Pengguna
Reaksi Aktor	
Aksi Aktor	Reaksi Sistem
1. Memilih pola data yang ingin diubah 2. Mengganti atau menambah dengan pola data baru 3. Memilih tombol <b>Edit</b>	4. Menyimpan ke dalam <i>database</i> 5. Menampilkan pesan “ Data sukses diubah”

Tabel 3.5. *Use Case* Skenario Hapus Data Master

Nama <i>Use Case</i>	Hapus Data Master
Deskripsi	Pengguna Menghapus Data
Aktor	Pengguna
Reaksi Aktor	

Aksi Aktor	Reaksi Sistem
1. Memilih pola data yang ingin dihapus 2. Memilih tombol <b>Hapus</b> 4. Tekan <b>OK</b> 6. Tekan <b>OK</b>	3. Menampilkan pesan “ Apakah data akan dihapus?” 5. Menampilkan pesan “Data berhasil dihapus”

Tabel 3.6. *Use Case* Skenario Baru Data Master

Nama <i>Use Case</i>	Baru Data Master
Deskripsi	Pengguna Menambah Pola Data
Aktor	Pengguna
Reaksi Aktor	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol <b>Baru</b>	2. Mengosongkan panel Isi Data

Tabel 3.7. *Use Case* Skenario Keluar Data Master

Nama <i>Use Case</i>	Keluar Data Master
Deskripsi	Pengguna Keluar dari <i>Form</i> Pegawai
Aktor	Pengguna
Reaksi Aktor	

Aksi Aktor	Reaksi Sistem
1. Memilih tombol <b>Keluar</b>	2. Sistem keluar dari <i>Form</i> Pegawai

Tabel 3.8. *Use Case* Skenario Simulasi *Backpropagation*

Nama <i>Use Case</i>	Simulasi <i>Backpropagation</i>
Deskripsi	Pengguna melakukan simulasi dengan metode <i>Backpropagation</i>
Aktor	Pengguna
<i>Include</i>	Pengolahan Data Master
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Berada pada <i>form</i> simulasi <i>Backpropagation</i>	
2. Mengetik nama pegawai dan pilih bulan	
3. Memilih Tombol <b>Normalisasi</b>	4. Tampilan data hasil normalisasi beserta nilai max dan min
5. Memilih tombol <b>Proses</b>	6. Melakukan validasi semua data
	7. Random bobot <i>input layer</i>
	8. Random bobot <i>hidden layer</i>
	7. Hitung keluaran jaringan dari <i>input</i>
	8. Bangkitkan fungsi aktivasi <i>Layer</i>
	9. Hitung keluaran jaringan dari <i>hidden layer</i>

	10. Bangkitkan fungsi aktivasi
	11. Hitung perubahan bobot dan bias <i>hidden layer</i>
	12. Hitung perubahan bobot dan bias <i>input layer</i>
	13. Tampilkan data hasil simulasi dan tampilan pesan “Proses Simulasi Selesai”
14. Memilih tombol <b>Simpan</b>	15. Tampilan pesan “Data Sukses Disimpan”

Tabel 3.9. *Use Case* Skenario Pengujian *Backpropagation*

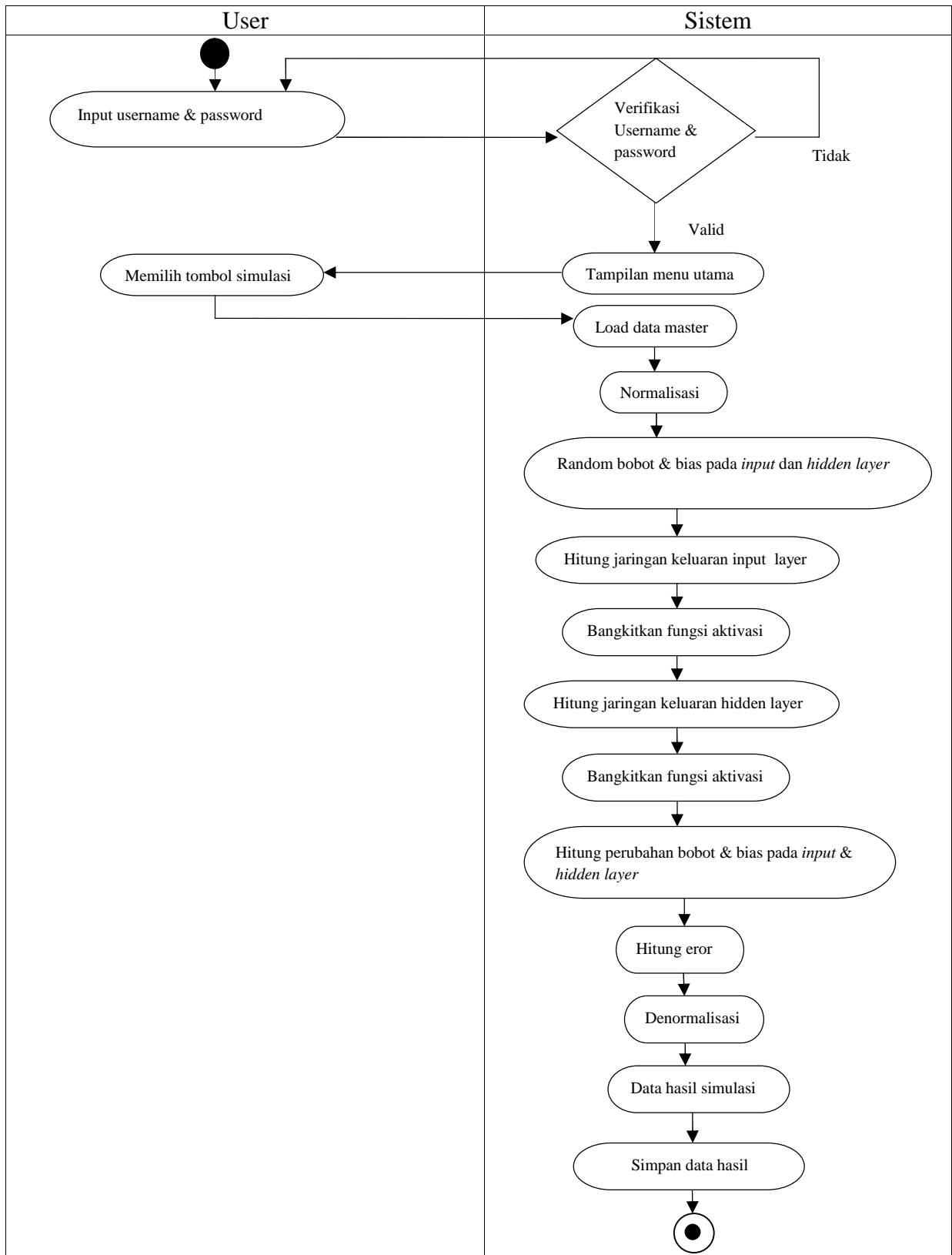
Nama <i>Use Case</i>	Pengujian <i>Backpropagation</i>
Deskripsi	Pengujian metode <i>Backpropagation</i>
Aktor	Pengguna
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memilih menu pengujian <i>Backpropagation</i>	
	2. Menampilkan form pengujian <i>Backpropagation</i>
3. Mengetik nama pegawai	
4. Memilih bulan	
5. Memilih tombol <b>Proses</b>	6. Melakukan validasi data
	7. Menampilkan pesan hasil pengujian
	8. Menyimpan data hasil pengujian <i>Backpropagation</i>

9. Memilih tombol <b>Simpan</b>	10. Menampilkan pesan data sukses disimpan
11. Memilih tombol <b>Cetak</b>	12. Menampilkan data laporan

### 3.3.2. Activity Diagram

*Activity diagram* memiliki fungsi yang sama dengan *use case* spesifikasi yaitu untuk menjelaskan *use case* secara lebih rinci. Perbedaan keduanya terletak pada cara pandang dalam hal menjelaskan *use case* tersebut. Pada *use case* spesifikasi, *use case* dijelaskan dengan narasi yang tanggung jawabnya dibuat terpisah antara pengguna dan reaksi sistem. Skenario utama pada *use case* spesifikasi identik dengan penulisan algoritma dalam bentuk naratif. Sedangkan pada *activity diagram*, *use case* dijelaskan dalam bentuk diagram alir yang juga memisahkan tanggung jawab pengguna dan sistem. Berikut adalah tampilan *Activity diagram* dari simulasi dan pengujian *Backpropagation* :

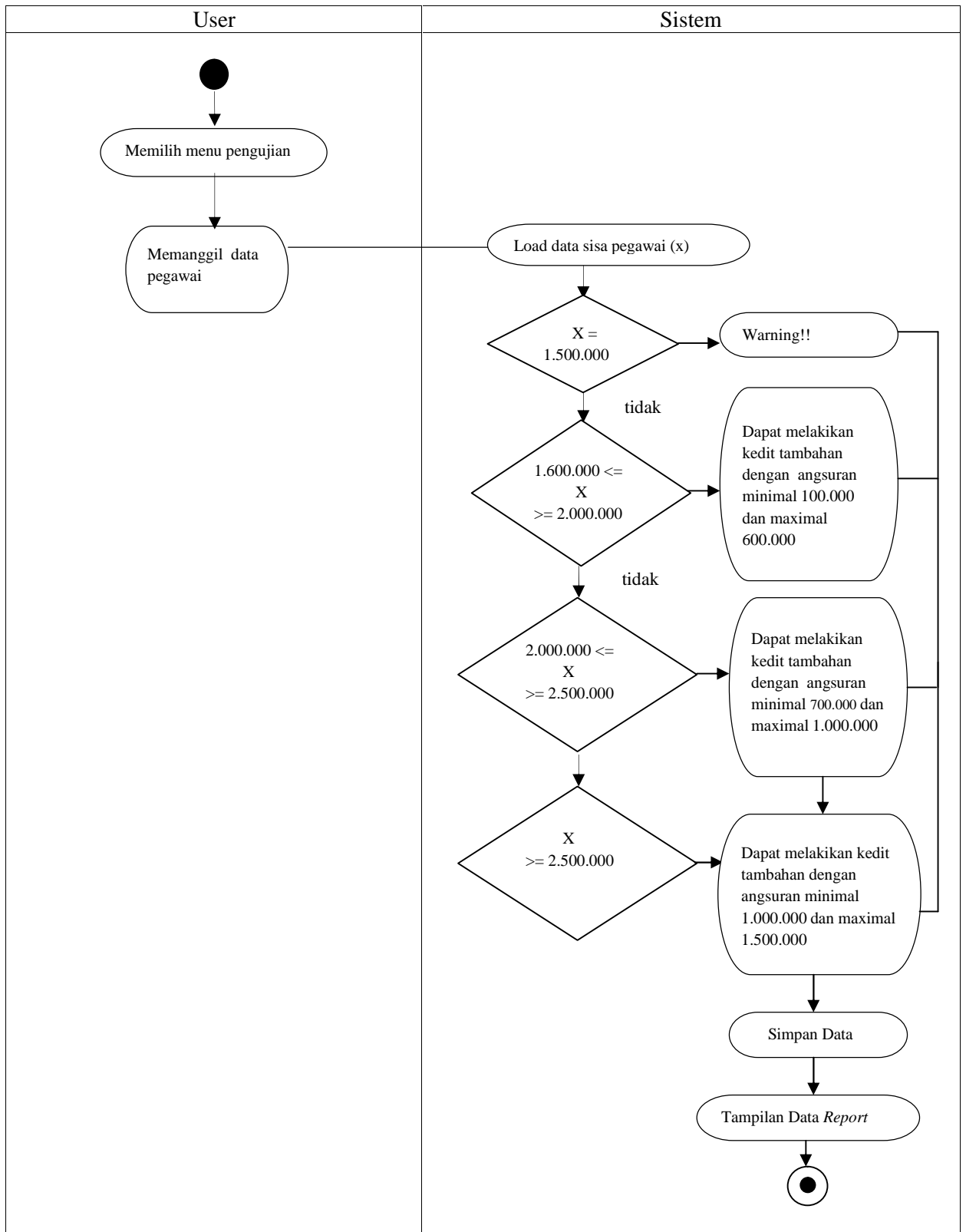
Tabel 3.10. Activity Diagram Simulasi Backpropagation





Berdasarkan tabel *activity diagram* simulasi *Backpropagation*, dijelaskan bahwa pengguna harus menginput *username* dan *password* kemudian dilakukan verifikasi yaitu *login*, jika berhasil pengguna akan masuk ke tampilan menu utama, jika tidak pengguna harus menginput ulang *username* dan *password*. Selanjutnya, akan muncul tampilan menu utama dan pengguna memilih tombol simulasi dan meload data master. Setelah meload data master selanjutnya akan dinormalisasikan, membuat random bobot dan bias pada *input* dan *hidden layer*. Selanjutnya dilakukan perhitungan jaringan keluaran *input layer* dan bangkitkan fungsi aktifasinya dan begitu juga pada *hidden layer*. Setelah itu, menghitung perubahan bobot dan bias pada *input* dan *hidden layer*, menghitung nilai error, dinormalisasikan dan muncul data dari hasil simulasinya yang disimpan dalam *database*.

Tabel 3.11. Activity Diagram Pengujian *Backpropagation*

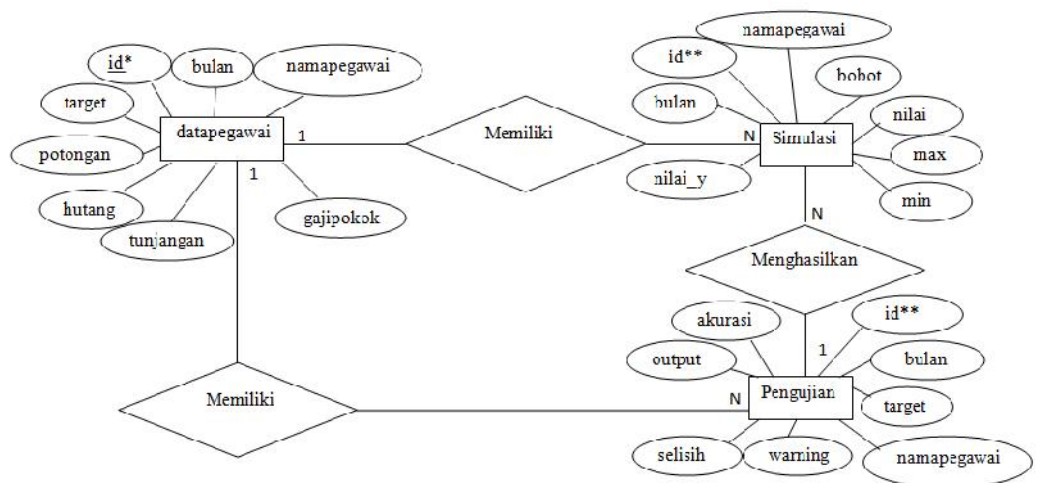


Berdasarkan tabel *activity diagram* pengujian *Backpropagation*, dijelaskan bahwa pengguna akan memilih menu pengujian dan mengisi data pegawai, sistem akan menjalankan sistem pengujiannya dan akan load data sisa pegawai (x). Kemudian, akan ada *warning*/ peringatan akan dilakukan kredit tambahan sesuai dengan besar gaji berdasarkan kriteria *warning* yang sudah berada pada sistem dan dicetak laporannya.

### 3.3.3. Perancangan Database

Perancangan *database* dibutuhkan untuk menyimpan data terkait aplikasi yang dibuat. Perancangan *database* pada dokumentasi ini terdiri dari *Entity Relational Diagram* (ERD), relasi antar tabel dan perancangan tabel. Berikut tampilan ERD, relasi antar tabel dan perancangan tabel :

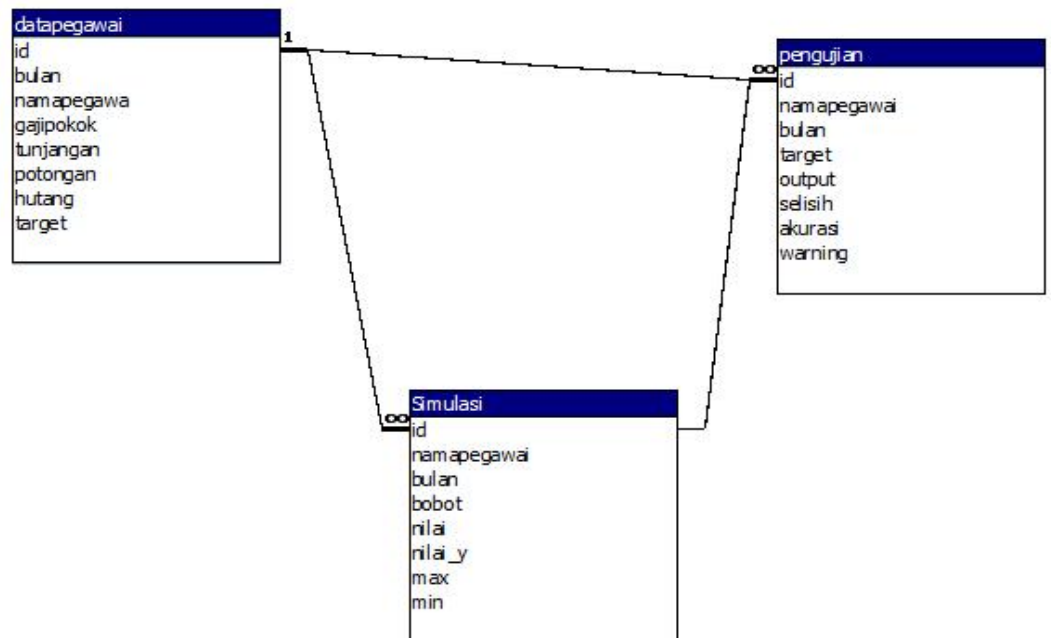
#### 3.3.3.1. Entity Relationship Diagram (ERD)



Gambar 3.3. Tampilan ERD

Tujuan dibuatnya *Entity Relationship Diagram* adalah untuk menghubungkan dan menggambarkan hubungan – hubungan antar entitas, dimana terdapat tiga entitas pada tampilan *entity relationship diagram* yaitu Data pegawai, simulasi dan pengujian dengan setiap entitas memiliki masing – masing atribut.

### 3.3.3.2. Relasi Antar Tabel



Gambar 3.4. Tampilan Relasi Antar Tabel

Sebuah tabel akan menyimpan data dari sebuah entitas dan satu *record* dalam tabel akan menyimpan nilai – nilai atribut dai sebuah entitas.

### 3.3.3.3. Perancangan Tabel

#### 1. Tabel Data Pegawai

Tabel 3.12. Tabel Data Pegawai

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Keterangan</i>
Id	<i>Varchar</i>	5	<i>Primary Key</i>
Bulan	<i>Text</i>		
Namapegawai	<i>Text</i>		
Gajipokok	<i>Integer</i>	8	
Tunjangan	<i>Integer</i>	8	
Potongan	<i>Integer</i>	8	
Hutang	<i>Integer</i>	8	
Target	<i>Integer</i>	8	

Pada tabel entitas Data Pegawai, terdapat 8 atribut yakni id sebagai *primary key*, bulan, namapegawai, gajipokok, tunjangan, potongan, hutang dan target.

## 2. Tabel Simulasi

Tabel 3.13. Tabel Simulasi

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Keterangan</i>
Id	<i>Varchar</i>	5	<i>Foreign Key</i>
Namapegawai	<i>Text</i>		
Bulan	<i>Varchar</i>	12	
Bobot	<i>Varchar</i>	11	
Nilai	<i>Double</i>		
Nilai_y	<i>Double</i>		
Max	<i>Double</i>		
Min	<i>Double</i>		

Pada tabel entitas Simulasi, terdapat 8 atribut yakni id sebagai *foreign key*, namapegawai, bulan, bobot, nilai, nilai\_y, max, dan min.

### 3. Tabel Pengujian

Tabel 3.14. Tabel Pengujian

Field Nama	Data Type	Field Size	Keteangan
Id	<i>Varchar</i>	5	<i>Foreign Key</i>
Namapegawai	<i>Varchar</i>	20	
Bulan	<i>Varchar</i>	10	
Target	<i>Varchar</i>	20	
Output	<i>Double</i>		
Selisih	<i>Double</i>		
Akurasi	<i>Double</i>		
Warning	<i>Double</i>		

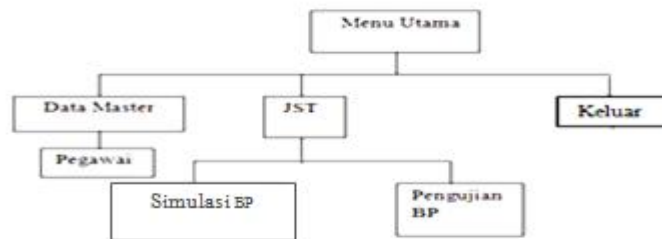
Pada tabel entitas Pengujian, terdapat 8 atribut yakni id sebagai *foreign key*, namapegawai, bulan, target, *output*, selisih, akurasi dan *warning*.

#### 3.3.4. Perancangan Antarmuka (*Design Interface*)

Didalam sistem yang sudah dibuat, hanya ada interaksi antara pengguna dengan komputer. Komponen antarmuka grafis adalah bentuk tampilan yang dapat dilihat dan umumnya bersifat tetap. Tujuannya dilakukan perancangan antarmuka pada dasarnya adalah untuk mendapatkan satu kriteria dalam pengoperasian pada sebuah program aplikasi. Saat ini antarmuka yang banyak dilakukan dalam *software* adalah GUI (*Graphical User Interface*).

### 3.3.4.1. Perancangan Struktur Menu

Struktur menu aplikasi metode *Backpropagation* diperlihatkan berikut :



Gambar 3.5. Struktur Menu Aplikasi

### 3.3.4.2. Perancangan Form Login

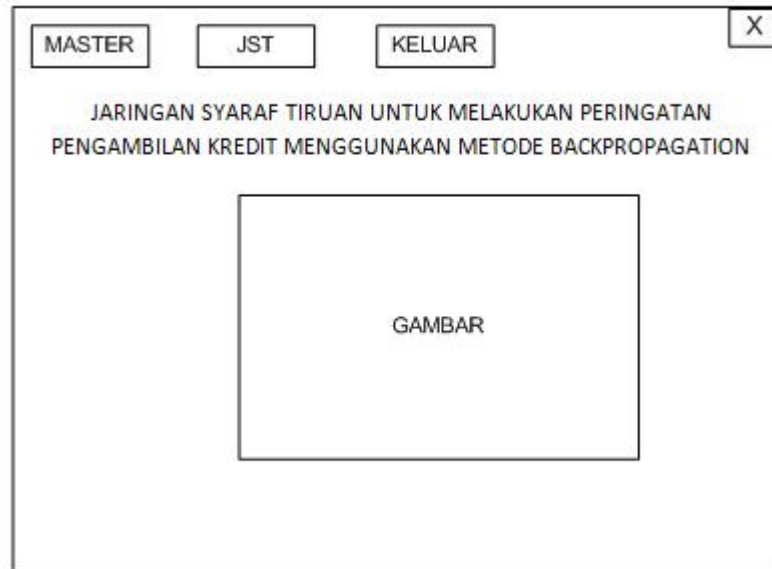
The screenshot shows a window titled 'Login' with a close button 'X' in the top right corner. The form contains two input fields: 'Username :' and 'Password :'. Below the input fields are two buttons: 'Login' and 'Exit'.

Gambar 3.6. Perancangan Form Login

Perancangan ini digunakan untuk pengguna melakukan *login* sebelum masuk ke menu utama, dengan mengisi *Username* dan *Password*, jika benar maka *login* berhasil dan pengguna akan memasuki langkah berikutnya yakni menu utama sistem.



### 3.3.4.3. Perancangan Menu Utama



Gambar 3.7. Perancangan Menu Utama

Perancangan Menu Utama diatas menampilkan beberapa item didalamnya antara lain Master yang berisi data Master Pegawai, JST berisi Simulasi dan Pengujian *Backpropagation* dan Keluar.

### 3.3.4.4. Perancangan *Form* Pegawai

FORM PEGAWAI			
Title 1	Title 2	Title 3	Title 4

Isi Data		Tombol	
Id	<input type="text"/>	<input type="button" value="Simpan"/>	<input type="button" value="Hapus"/>
Bulan	<input type="text"/>	<input type="button" value="Baru"/>	<input type="button" value="Edit"/>
Nama Pegawai	<input type="text"/>	<input type="button" value="Keluar"/>	
Gaji Pokok	<input type="text"/>		
Tunjangan	<input type="text"/>		
Potongan	<input type="text"/>		
Hutang	<input type="text"/>		
Target	<input type="text"/>		

Gambar 3.8. Perancangan *Form* Pegawai

Perancangan *Form* Pegawai diatas menjelaskan bahwa *user* akan memasukan data setiap pegawai per bulan untuk melakukan peringatan pengambilan kredit. Perancangan diatas terdiri dari tabel, panel Isi Data terdiri dari ID, Bulan, Nama Pegawai, Gaji Pokok, Tunjangan, Potongan, Hutang, dan Gaji diterima atau target dan tombol terdiri dari Simpan, Hapus, Baru, Edit dan Keluar.

### 3.3.4.5. Perancangan *Form* Simulasi

SIMULASI BACKPROPAGATION

Cari Nama  Pilih Bulan

Title 1	Title 2	Title 3	Title 4	Title 5

Normalisasi

Title 1	Title 2	Title 3	Title 4	Title 5

Perubahan Bobot

Title 1	Title 2	Title 3	Title 4

Parameter Pelatihan

Learning Rate

Epoch

Error Rate

Tombol

Proses

Keluar

Gambar 3.9. Perancangan *Form* Simulasi

Pada perancangan *Form* Simulasi diatas menjelaskan bahwa data pegawai yang sudah di *input*, akan dilakukan proses pencarian berdasarkan nama pegawai. Selanjutnya, data pegawai tersebut dilakukan proses normalisasi dan simulasi *Backpropagation* dengan menekan tombol proses. Setelah itu, akan menghasilkan bobot baru dan bias baru. Bobot baru dan bias baru tersebut selanjutnya akan disimpan pada *database*.

### 3.3.4.6. Perancangan *Form* Pengujian

Pengujian Backpropagation

Data Pegawai

Cari Nama pegawai

Nama Pegawai

Bulan

Target

Proses

Simpan

Cetak

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7

Gambar 3.10. Perancangan *Form* Pengujian

Pada Perancangan *Form* Pengujian, data pegawai yang akan diuji di *input* pada panel *form* pegawai, setelah itu diproses dan akan menghasilkan data dan *warning*.