# BAB V

# IMPLEMENTASI DAN PERANCANGAN SISTEM

## 5.1. Implementasi

Implementai merupakan sebuah tahap lanjutan setelah perancangan sistem yang akan dibuat dan di implementasikan di dalam kehidupan sehari-hari sesuai dengan kebutuhan. Tahap implementasi ini dilakukan sebagai akhir dari desain Sistem Informasi Perhitungan Pajak. Pada PT. Pos Indonesia (Persero) Implementasi bertujuan untuk mengetahui apa saja yang akan diterpakan dalam sistem.

### 5.1.1. Batasan Implementasi

Dalam sebuah perangkat lunak, tak semuanya perangkat lunak dapat mengerjakan apa yang diperintahkan oleh user. Setiap kelebihan sebuah perangkat lunak pasti mempunyai seuatu batasan. Begitu pula dalam Sistem Informasi Perhitungan Pajak. Pada PT. Pos Indonesia (Persero). Adapun batasannya adalah sebagai berikut:

1. Sistem ini membahas mengenai perhitungan pajak pegawai tetap.
2. Perhitungan pajak pegawai tetap meliputi data pegawai,data penghasilan, dan data potongan penghasilan.
3. Implementasi *database* dari sistem ini mengguakan MySQL.

### 5.1.2. Implementasi Perangkat Lunak

Implementasi Perangkat Lunak dari Sistem Informasi ini memerlukan beberapa perangkat pendukung diantaranya :

a. Kebutuhan Untuk Server
    1. Sistem Operasi : Microsoft Windows 7/Xp Service Pack 3
    2. Database Menggunakan MySQL
    3. Laporan menggunakan iReport 3.7.5
b. Kebutuhan Client
    1. Sistem Operasi : Microsoft Windows 7/Xp Service Pack 3
    2. Laporan menggunakan iReport 3.7.5

### 5.1.3 Implementasi Perangkat Keras

Implementasi Perangkat Keras untuk mendukung dari sistem ini adalah sebagai berikut :

a. Server
    1. Menggunakan Processor Intel Pentium Dual Core 2.0 Ghz *or Higher.*
    2. Menggunakna RAM minimal 3Gb
    3. HardDisk minimal 80 Gb.
    4. VGA minimal 256 *or Higher.*
    5. Monitor,Mouse,Keyboard
    6. Printer

b. Client
    1. Menggunakan Processor Intel Pentium Dual Core 2.0 Ghz *or Higher.*

2. Menggunakan RAM minimal 1Gb.
3. HardDisk minimal 40Gb
4. VGA minimal 256 *or Higher*
5. Monitor,Mouse,Keyboard

c. Konfigurasi Jaringan

1. Port Hub
2. Kabel UTP
3. Connector RJ 45

5.1.4. Implementasi Basis Data

Implementasi Basis Data Sebagai berikut :

```
CREATE TABLE IF NOT EXISTS `gaji` (
  `KdGol` varchar(20) NOT NULL,
  `KdGrade` varchar(10) NOT NULL,
  `Gapok` int(50) NOT NULL,
  KEY `KdGol` (`KdGol`),
  KEY `KdGrade` (`KdGrade`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Table structure for table `gajibruto`
CREATE TABLE IF NOT EXISTS `gajibruto` (
```

```
  `Tanggal` date NOT NULL,
  `Nip` varchar(30) NOT NULL,
  `GajiBruto` int(50) NOT NULL,
  KEY `Nip` (`Nip`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Table structure for table `gender`
CREATE TABLE IF NOT EXISTS `gender` (
  `KdGender` varchar(5) NOT NULL,
  `Gender` varchar(20) NOT NULL,
  PRIMARY KEY (`KdGender`)
-- Table structure for table `golongan`
CREATE TABLE IF NOT EXISTS `golongan` (
  `KdGol` varchar(20) NOT NULL,
  `NmGol` varchar(10) NOT NULL,
  PRIMARY KEY (`KdGol`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Table structure for table `grade`
```

```
CREATE TABLE IF NOT EXISTS `grade` (
  `KdGrade` varchar(10) NOT NULL,
  `NmGrade` varchar(10) NOT NULL,
  PRIMARY KEY (`KdGrade`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Table structure for table `jiwa`
--

CREATE TABLE IF NOT EXISTS `jiwa` (
  `KdJiwa` varchar(10) NOT NULL,
  `JmlTunjangan` int(50) NOT NULL,
  PRIMARY KEY (`KdJiwa`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Table structure for table `karyawan`
--

CREATE TABLE IF NOT EXISTS `karyawan` (
```

```
  `Nip` varchar(30) NOT NULL,
  `Nama` varchar(50) NOT NULL,
  `KdGender` varchar(5) NOT NULL,
  `KdStatus` varchar(5) NOT NULL,
  `KdJiwa` varchar(10) NOT NULL,
  `Alamat` varchar(50) DEFAULT NULL,
  `KdGol` varchar(20) NOT NULL,
  `KdGrade` varchar(10) NOT NULL,
  `Jabatan` varchar(10) NOT NULL,
  PRIMARY KEY (`Nip`),
  KEY `KdGender` (`KdGender`),
  KEY `KdStatus` (`KdStatus`),
  KEY `KdGol` (`KdGol`),
  KEY `KdGrade` (`KdGrade`),
  KEY `Jabatan` (`Jabatan`),
  KEY `KdJiwa` (`KdJiwa`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- Table structure for table `posisi`

CREATE TABLE IF NOT EXISTS `posisi` (

  `KdJabatan` varchar(10) NOT NULL,

  `NmJabatan` varchar(50) NOT NULL,

  PRIMARY KEY (`KdJabatan`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `status`

CREATE TABLE IF NOT EXISTS `status` (

  `KdStatus` varchar(5) NOT NULL,

  `Status` varchar(10) NOT NULL,

  PRIMARY KEY (`KdStatus`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `t_prestasi`

CREATE TABLE IF NOT EXISTS `t_prestasi` (

  `KdJabatan` varchar(10) NOT NULL,

  `JmlTunjangan` int(50) NOT NULL,

  KEY `KdJabatan` (`KdJabatan`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `user`

CREATE TABLE IF NOT EXISTS `user` (
  `id_user` varchar(30) NOT NULL,
  `Username` varchar(50) NOT NULL,
  `Password` varchar(50) NOT NULL,
  `Bagian` varchar(20) NOT NULL,
  KEY `id_user` (`id_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Constraints for table `gaji`

ALTER TABLE `gaji`
  ADD CONSTRAINT `gaji_ibfk_1` FOREIGN KEY (`KdGol`) REFERENCES `golongan` (`KdGol`),
  ADD CONSTRAINT `gaji_ibfk_2` FOREIGN KEY (`KdGrade`) REFERENCES `grade` (`KdGrade`);

-- Constraints for table `gajibruto`

ALTER TABLE `gajibruto`
```

```
 ADD CONSTRAINT `gajibruto_ibfk_1` FOREIGN KEY (`Nip`) REFERENCES `karyawan` (`Nip`);

-- Constraints for table `karyawan`

ALTER TABLE `karyawan`

 ADD CONSTRAINT `karyawan_ibfk_1` FOREIGN KEY (`KdGender`) REFERENCES `gender` (`KdGender`),

 ADD CONSTRAINT `karyawan_ibfk_2` FOREIGN KEY (`KdStatus`) REFERENCES `status` (`KdStatus`),

 ADD CONSTRAINT `karyawan_ibfk_3` FOREIGN KEY (`KdGol`) REFERENCES `golongan` (`KdGol`),

 ADD CONSTRAINT `karyawan_ibfk_4` FOREIGN KEY (`KdGrade`) REFERENCES `grade` (`KdGrade`),

 ADD CONSTRAINT `karyawan_ibfk_5` FOREIGN KEY (`Jabatan`) REFERENCES `posisi` (`KdJabatan`),

 ADD CONSTRAINT `karyawan_ibfk_6` FOREIGN KEY (`KdJiwa`) REFERENCES `jiwa` (`KdJiwa`);

-- Constraints for table `t_prestasi`

ALTER TABLE `t_prestasi`
```

```
 ADD CONSTRAINT `t_prestasi_ibfk_1` FOREIGN KEY (`KdJabatan`) REFERENCES `posisi` (`KdJabatan`);

-- Constraints for table `user`

ALTER TABLE `user`

 ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `karyawan` (`Nip`);
```

### 5.1.5. Implementasi Antar Muka

Implementasi antar muka bertujuan untuk memenuhhi kebutuhan user agar dapat berinteraksi dengan sebuah sistem. Fasilitas yang diberikan memungkinkan *user* dapat menggunakan fungsi yang terdapat pada sistem. Adapun *interface* dalam sistem ini adalah sebagai berikut :

a. Login

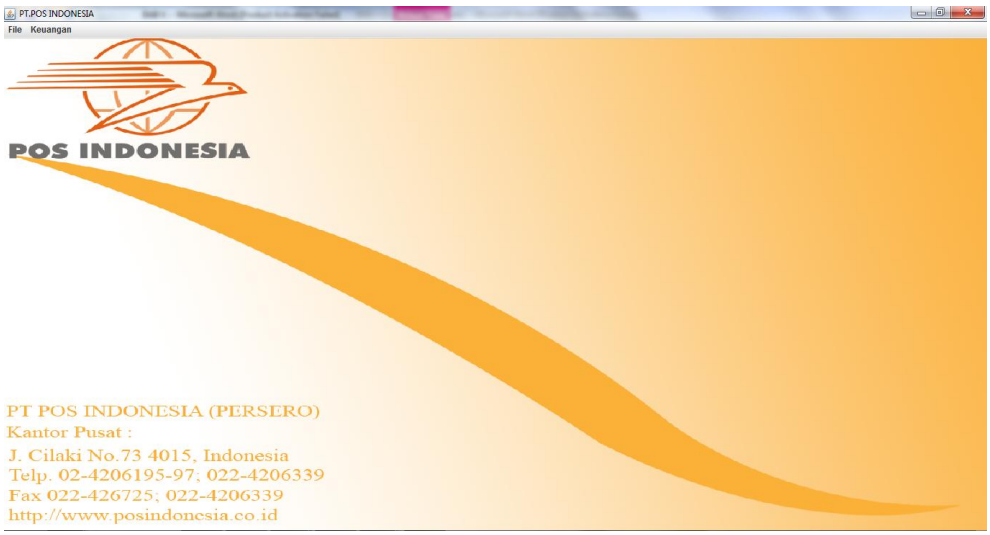
Fungsi login merupakan sebuah fungsi untuk memberikan hak akses kepada user yang menggunakan sistem ini.




Gambar 5.1. Login

b. Tampilan Menu Utama

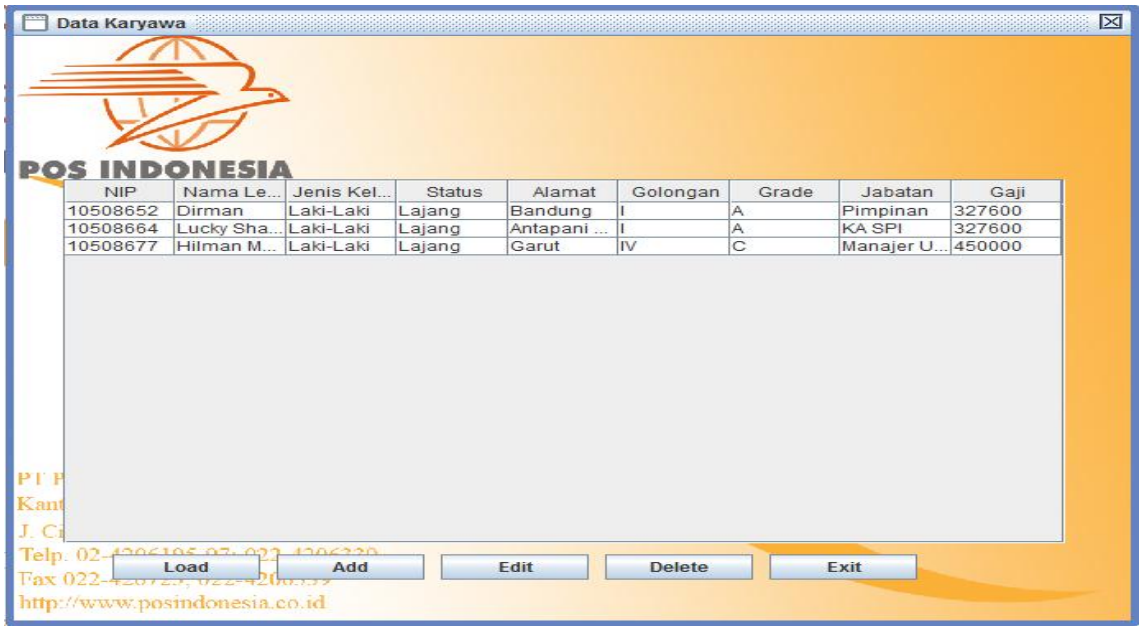
Setelah tombol login ditekan dan verifikasi username dan password valid, maka muncul menu utama dari program sesuai dengan hak akses masing-masing user yang masuk pada login. Dalam tampilan menu, pada Menu bar terdapat beberapa fungsi yang diantaranya : File, Data,Keuangan.




Gambar 5.2 Menu Utama

c. Tampilan Data Karyawan

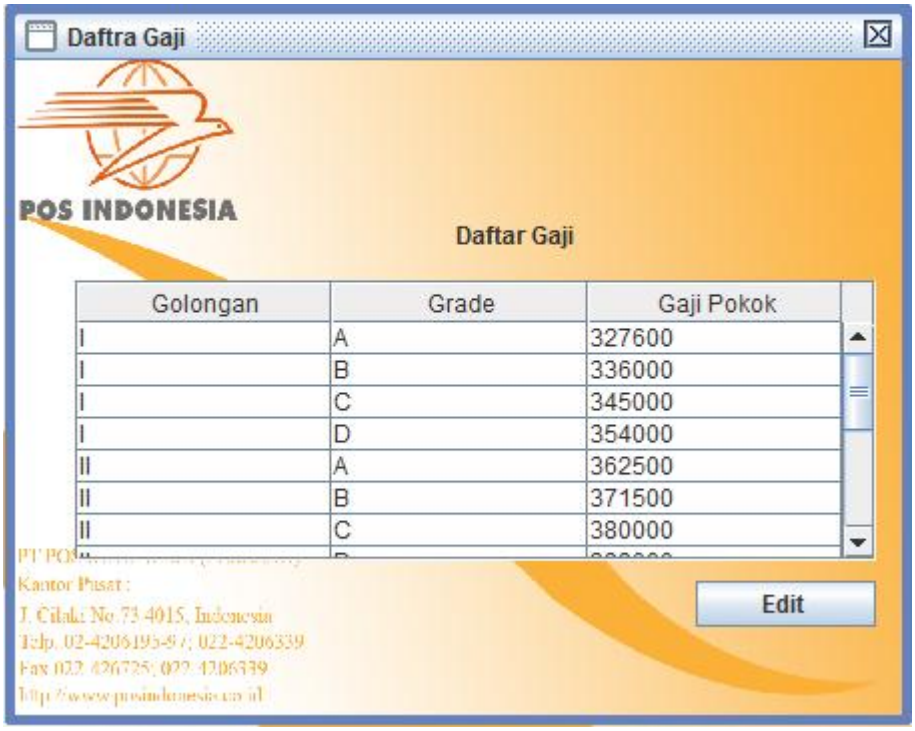
Tampilan data Karyawan merupakan sebuah tampilan yang fungsinya untuk menampilkan semua data Karyawan yang terdapat pada sistem.




Gambar 5.3. Data Karyawan

d. Tampilan Data Gaji

Tampilan data Gaji merupakan sebuah tampilan yang fungsinya untuk menampilkan semua data Gaji Karyawan yang terdapat pada sistem.




Gambar 5.4. Data Gaji

e. Tampilan Gaji Bruto

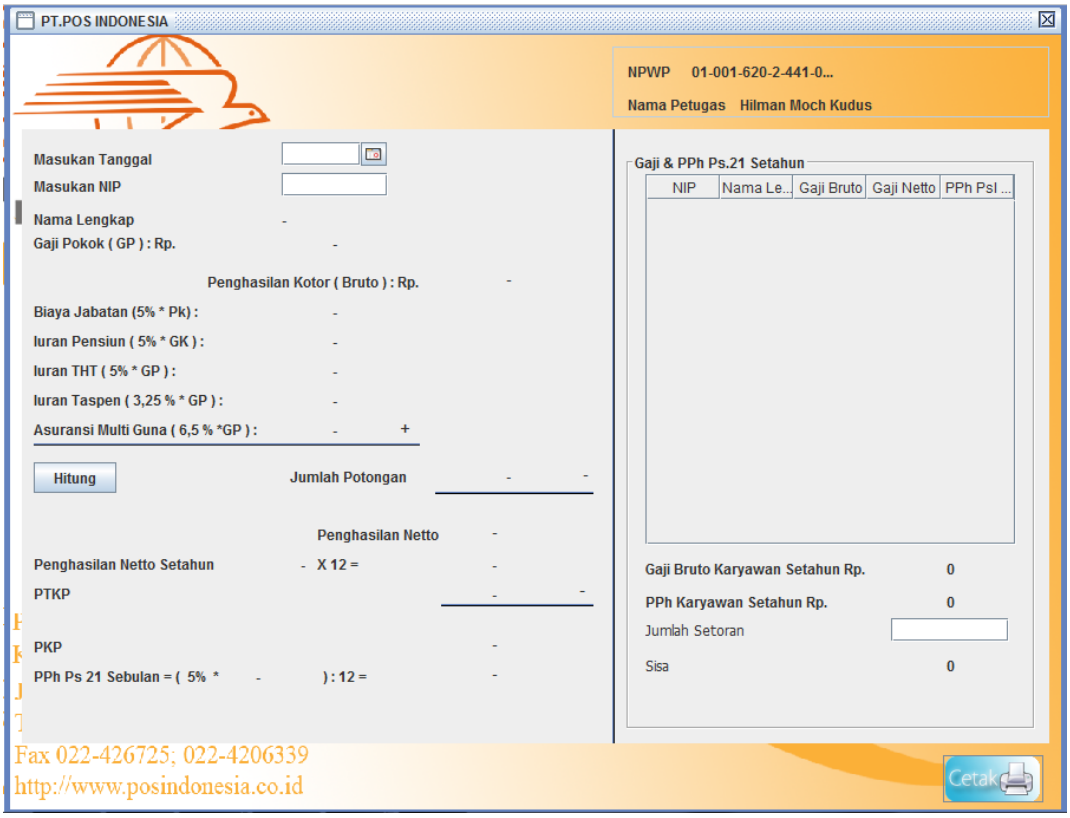
Tampilan Gaji Bruto merupakan sebuah tampilan yang fungsinya untuk Menghitung Gaji Bruto Karyawan yang terdapat pada sistem.




Gambar 5.5. Data Gaji Bruto

f. Tampilan Gaji Netto dan Pph pasal 21

Tampilan Gaji Netto dan Pph pasal 21 merupakan sebuah tampilan yang fungsinya untuk Menghitung Gaji Netto dan Pph pasal 21 Karyawan yang terdapat pada sistem.




Gambar 5.6. Data Gaji Netto dan Pph pasal 21

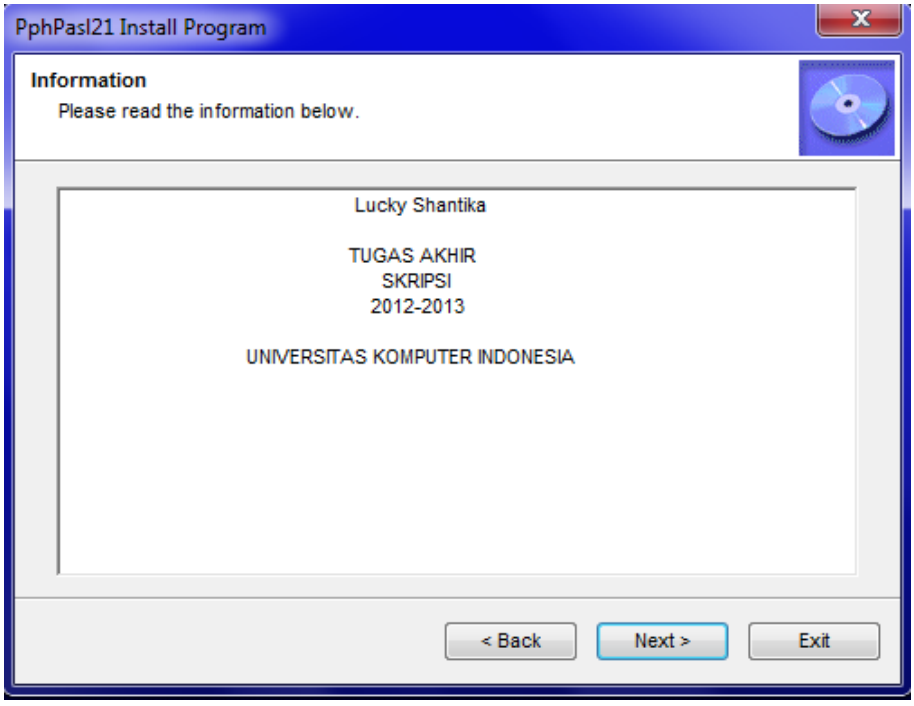
### 5.1.6. Implementasi Instalasi Program

1. Pilih Setup_Mustikasari,Kemudian klik dua kali atau tekan tombol enter
2. Kemudian Muncul gambar dialog Instalasi seperti dibawah ini.



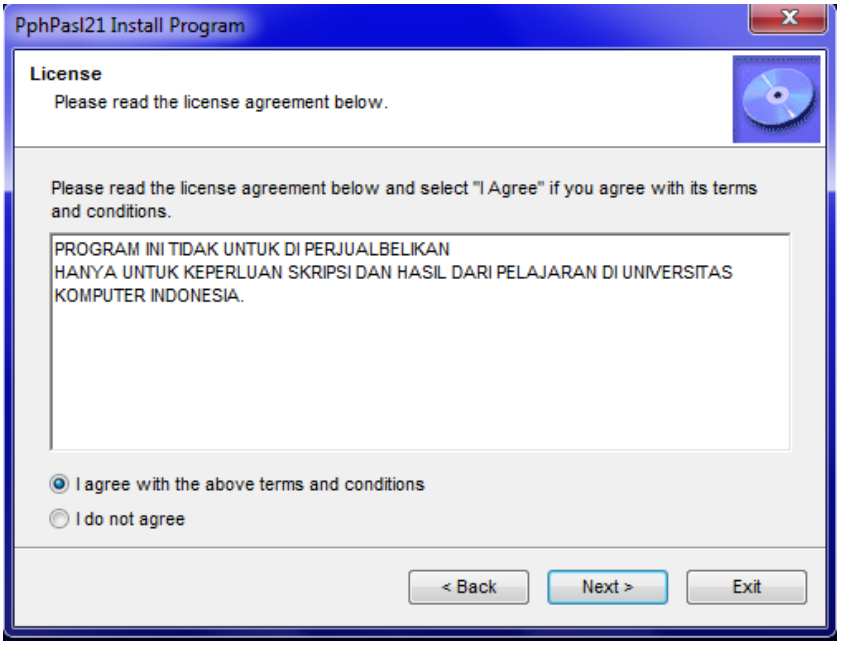

Gambar 5.7. Installasi tahap 1

3. Kemudian klik tombol Next, dan muncul informasi program seperti di bawah ini



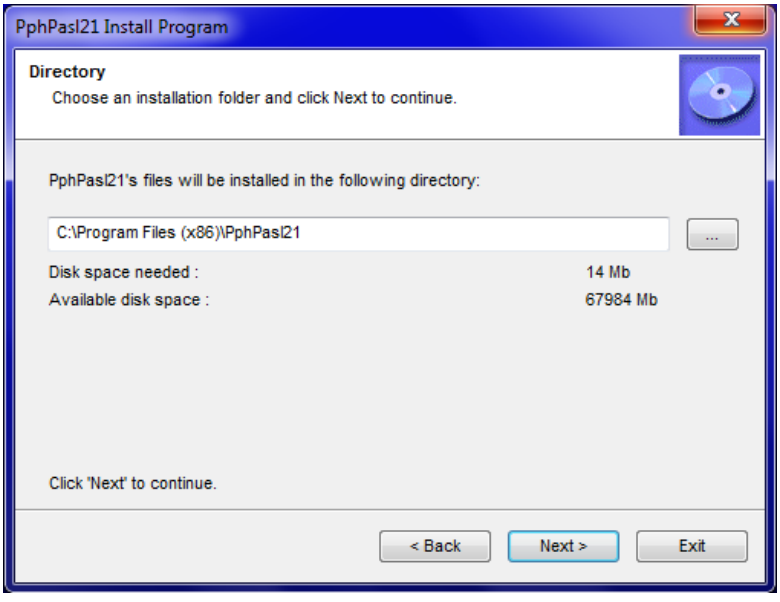

Gambar 5.8. Installasi tahap 2

4. Setelah itu keluar license program, Kemudian tekan tombol Next



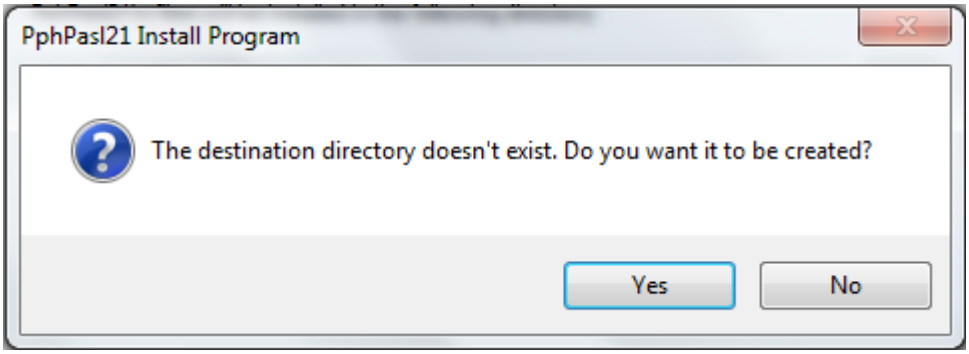

Gambar 5.9. Installasi tahap 3

5. Tampil tempat penyimpanan program yang dimana nantinya temmpat dimana program tersebut di install. Pilih lokasi penyimpanan kemudian tekan tombol Next.



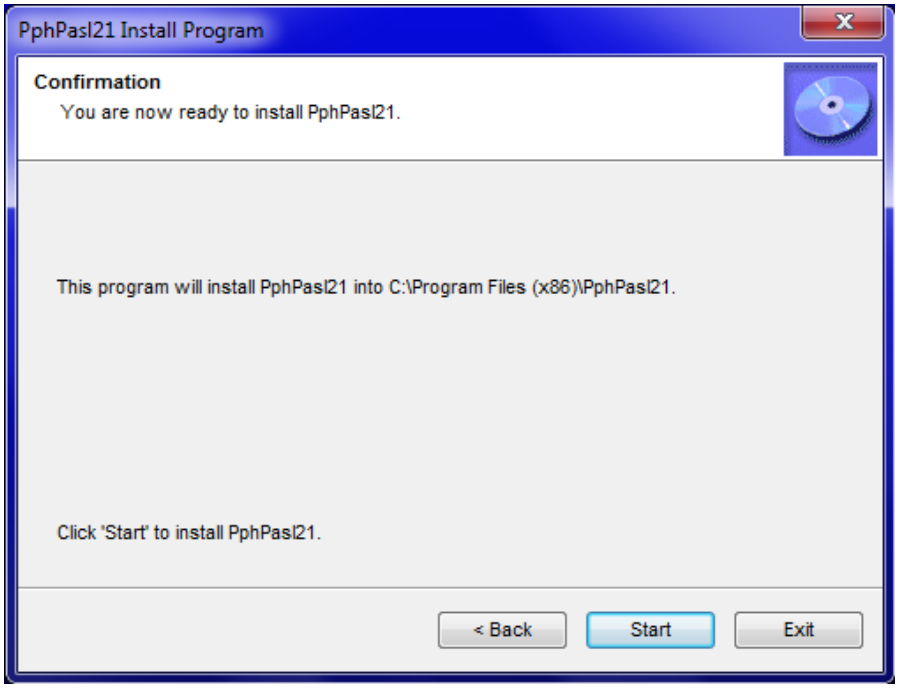

Gambar 5.10. Installasi tahap 4

6. Setelah kita menentukan tempat penyimpanan program untuk di install maka keluar dialog untuk member pemberitahuan bahwa penyimpanan program telah dibuat seperti gambar dibawah. Kemudian tekan tombol Yes



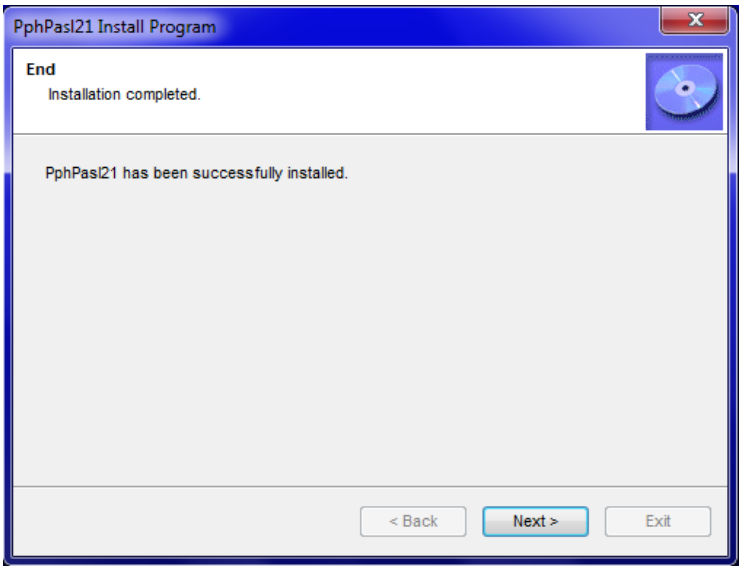

Gambar 5.11. Installasi tahap 5

7. Kemudian muncul tampilan untuk memulai proses instalasi seperti di bawah ini. Lalu tekan tombol Start.



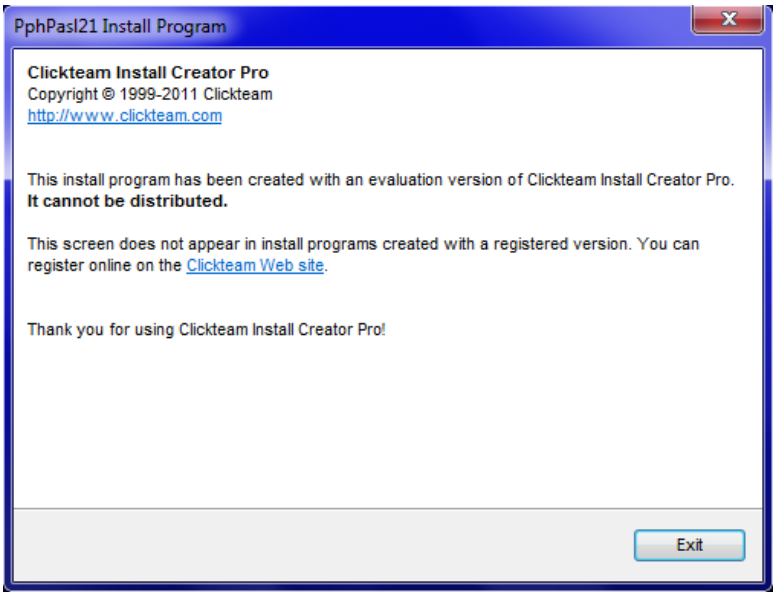

Gambar 5.12. Installasi tahap 6

8. Setelah selesai proses instalasi maka akan muncul pemberitahuan seperti di bawah ini. Lalu tekan tombol Next.




Gambar 5.13. Installasi tahap 7

9. Setelah selesai proses installasi maka akan muncul dialog seperti dibawah ini, lalu tekan tombol Exit.




Gambar 5.14. Installasi tahap 8

### 5.1.7. Penggunaan Program

Untuk Prosedur penghitungan gaji bruto, pada form Gaji Bruto bagian keuangan mennginputkan data karyawan berdasarkan nip dengan menekan tombol Baru lalu menekan tombol hitung untuk mendapatkan jumlah tunjangan – tunjangan pada field-field yang tersedia kemudian menyimpan dan mencetaknya dengan menekan tombol Gaji Bruto.

Untuk Prosedur selanjutnya Gaji Netto dan Pph Pasal 21, pada form Netto dan Pph Pasal 21 Bagian Keuangan menginputkan data dengan Memasukan tanggal input gaji bruto beserta nip, lalu menekan tombol hitungdan system secara otomatis memasukan pada daftar tabel gaji netto dan pph Pasal 21.

Prosedur selanjutnya yaitu Data Karyawan, pada form data aryawan. SDM mengelola data Karyawan, user menambahkan dengan menekan tombol Add. Dan menyimpannya dengan menekan tombol Save.

## 5.2. Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Dengan dilakukannya pengujian terhadap sistem yang dibuat, akan meminimalisir kesalahan yang mungkin akan muncul saat program dijalankan. Pengujian dilakukan untuk mengetahui keandalan program yang dibuat. Agar fungsional sistem bekerja dengan baik, dan tidak mengalami kesalahan fungsi atau error saat penggunaan program. Tujuan pengujian perangkat lunak ada dua macam yaitu tujuan langsung dan tidak langsung.

A. Tujuan Langsung

1. Identifikasi dan menemukan beberapa kesalahanyang mungkin ada dalam perangkat lunak yang di uji
2. Setelah perangkat lunak dibetulkan, diidentifikasi lagi kesalahan dan di tes ulang untuk menjamin kualitas level penerimaan.
3. Membentuk tes yang efisien dan efektid dengan anggaran dan jadwal yang terbatas.

B. Tujuan Tidak Langsung

Mengumpulkan daftar kesalahan untuk digunakan dalam daftar pencegahan kesalahan (tindakan *corrective* dan *preventive*).

5.2.1. Rencana Pengujian

Rencana pengujian yang dilakukan adalah pengujian terhadap faktor akses kontrol, kebenaran data, integrasi database, dan kelanjutan pemrosesan data.

Tabel 5.1 Rencana Pengujian Sistem Informasi Perhitungan Pajak Karyawan Tetap Pada PT.Pos (Persero)

| **Kelas Uji** | **Butir Uji** | **Tingkat Pengujian** | **Jenis Pengujian** |
|---|---|---|---|
| Akses kontrol | Pengecekan LOGIN user yang telah terdaftar | Integrasi | *Black box* |
| Integrasi Database | Data Pada Tabel Master dengan | Integrasi | *Black Box* |

| | | | |
|---|---|---|---|
| | Tabel Transaksi Sama | | |
| Kebenaran Data | Data yang diinputkan sesuai data yang pada output | Integrasi | *Black Box* |

5.2.2. Kasus dan Hasil Pengujian

Kasus uji yang dilakukan disesuaikan dengan rencana pengujian yang telah dijabarkan sebelumnya. Dibawah ini sampel kasus uji dan hasil pengujian yang dilakukan.

**Tabel 5.2.** Pengujian Akses Kontrol

| **Kasus Uji** | **Langkah Pengujian** | **Hasil Yang Diharapkan** | **Hasil Aktual** | **Kesimpulan** |
|---|---|---|---|---|
| LOGIN (SUKSES) | 1. Pengguna harus berada pada halaman login<br>2. Pada text box username dan password masukan username dan password yang valid<br>Misalnya<br>username : cemi dan password : a | Akan Terdisplay Sistem dengan menu – menu sesuai hak akses pengguna | Akan Tampil Sistem dengan menu – menu sesuai hak akses pengguna | [+] Diterima<br>[-] Ditolak |

| | | | | |
|---|---|---|---|---|
| | 3. Klik tombol login | | | |
| LOGIN (jenis user, username atau password yang dimasukan salah) | 1. Pengguna harus berada pada halaman login<br>2. Pada text box username dan password masukan username dan password yang tidak valid Misalnya username : abcde, password : 1234<br>3. Klik tombol login | Akan terdisplay pesan “Username / Password Salah, silahkan input ulang “ | Akan terdisplay pesan Username / Password Salah, silahkan input ulang ” | [+] Diterima<br>[-] Ditolak |

**Tabel 5.3.** Pengujian Integritas *Database* ( Sinkronisasi )

| **Kasus Uji** | **Langkah Pengujian** | **Hasil Yang Diharapkan** | **Hasil Aktual** | **Kesimpulan** |
|---|---|---|---|---|
| Data Pada Tabel Master sesuai Dengan Tabel | 1. User memasukan data pada tabel master<br>2. User lainnya melihat data pada tabel gaji | Jika data pada tabel master telah tersedia, maka pada tabel Gaji telah terdapat data sesuai tabel master | Data dapat disimpan | [+] Diterima<br>[-] Ditolak |
| Data Pada | 1. Misalnya user | misalnya tabel | Gaji Netto | [+] Diterima |

| | | | | |
|---|---|---|---|---|
| Tabel Master tidak sesuai Dengan Tabel Transkasi | memasukan data master<br>2. User melihat data pada tabel Gaji<br>( Jika tabel master tidak terdapat data, maka dalam tabel Gaji tidak dapat melihat hasil dari pemrosesan dari tabel master ) | master tidak diisi data nip karyawan maka pada tabel gaji tidak akan muncul karyawan yang dibutuhkan. | dan Pph diminta tidak muncul | [-] Ditolak |

**Tabel 5.4.** Pengujian Kebenaran Data

| **Kasus Uji** | **Langkah Pengujian** | **Hasil Yang Diharapkan** | **Hasil Aktual** | **Kesimpulan** |
|---|---|---|---|---|
| Kebenaran Data<br>( data diinputkan secara lengkap ) | 1. User berada pada form master (Karyawan)<br>2. Klik tombol tambah<br>3. Inputkan data pada semua text box yang berfungsi sebagai menginputkan data<br>4. Klik simpan | Akan Terdisplay bahwa data telah berhasil ditambahkan | Akan Terdisplay bahwa data telah berhasil ditambahkan | [+] Diterima<br>[-] Ditolak |

| | | | | |
|---|---|---|---|---|
| Kebenaran Data ( data diinputkan tidak lengkap ) | 1. User berada pada form master ( Karyawan )<br>2. Klik tombol tambah Inputkan data pada bebeapa text box saja yang<br>4. berfungsi menginputkan data, misalnya Nip Karyawan tidak diinputkan<br>Klik simpan | Data tidak akan tersimpan dan akan ada tampilan pesan “Data belum diisi“ | Data tidak akan tersimpan dan akan ada tampilan pesan “Data belum diisi “ | [+] Diterima<br>[-] Ditolak |

5.2.3. Kesimpulan Hasil Pengujian

Seperti yang telah dijelaskan sebelumnya dari tujuan pengujian, dengan sampel diatas diharapkan pengujian yang dilakukan akan menghasilkan sistem yang memiliki fungsional yang baik, sehingga kesalahan yang mungkin akan muncul menjadi lebih kecil kemungkinannya.