

## BAB IV

### DISKRIPSI PEKERJAAN

#### 4.1 Metode Pelaksanaan

Pelaksanaan Proyek Akhir dilaksanakan 3 (tiga) bulan pada IT SAP *Department*. Di PT ECCO Tannery Indonesia yang pelaksanaannya pada:

tanggal : 11 January 2016 – 11 April 2016

peserta : Adhaji Vidananto

NIM : 13.39015.0004

Dalam pelaksanaan Proyek Akhir yang berlangsung dalam kurun waktu tiga bulan di PT ECCO Tannery Indonesia, berikut ini adalah rincian kegiatan yang dilakukan di bagian IT SAP *Department* dapat dilihat pada tabel 4.1:

Tabel 4.1 Daftar Kegiatan Selama Kerja Praktik

NO	Kegiatan/Pekerjaan
1	Memasukan data <i>Bill Of Material</i> (BOM) ke dalam aplikasi Java Antara
2	Pembuatan Aplikasi IT <i>Issue Log</i> menggunakan <i>Microsoft Access</i> pada bagian IT SAP <i>Department</i>
3	Pelayanan SPT Pajak dan Pembuatan G-mail
4	Pembuatan aplikasi <i>Training</i> menggunakan <i>Microsoft Access</i> pada bagian HRD
5	Pembuatan aplikasi Surat Peringatan (SP) menggunakan <i>Microsoft Access</i> pada bagian HRD

## 4.2 Metode Penulisan

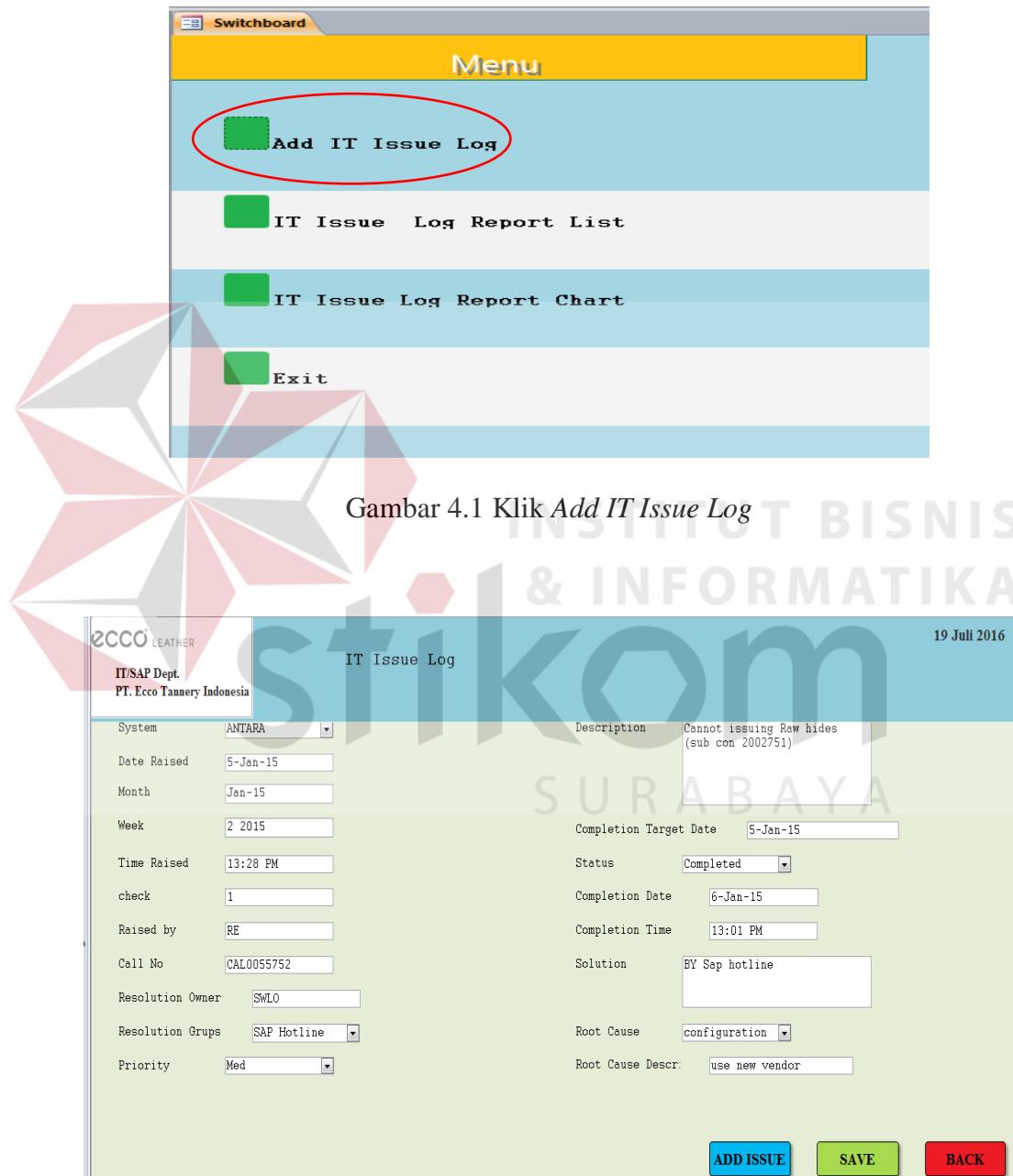
Metode penulisan yang digunakan untuk menyelesaikan laporan Proyek Akhir pada Program Studi DIII Komputerisasi Perkantoran dan Kesekretariatan Institut Bisnis dan Informatika Stikom Surabaya adalah:

- a. Studi *Observasi*, yaitu dengan pengamatan dan mempelajari secara langsung pada Bagian IT SAP *Department* di PT ECCO Tannery Indonesia.
- b. Wawancara, yaitu dengan mengadakan tanya jawab dengan pembimbing yaitu pada bagian IT SAP *Department* di PT ECCO Tannery Indonesia.
- c. Studi *Literatur* atau perpustakan, yaitu dengan mencari dan membaca *literatur* dan buku-buku yang mendukung penyelesaian laporan Proyek Akhir yang tersedia di perpustakaan.
- d. Penyusunan laporan, yaitu setelah melakukan kegiatan Proyek Akhir, langkah selanjutnya adalah menyusun laporan yang menjadi prasarat dalam menyelesaikan mata kuliah masa studi.

## 4.3 Pembuatan Aplikasi IT *Issue Log*

Aplikasi IT *Issue Log* adalah aplikasi yang berfungsi untuk menyimpan data masalah pada aplikasi Antara dan SAP yang digunakan pada PT ECCO Tannery Indonesia. Kemudian hasil dari masalah tersebut akan dilaporkan kepada Direktur berupa *report pivot chart*. Data yang dimasukkan ke dalam Aplikasi IT *Issue Log* adalah data masalah yang sering terjadi setiap hari tentang sistem Antara dan SAP. Sehingga, data yang dimasukkan ke dalam aplikasi IT *Issue Log* merupakan data yang selalu *update*. Jika sehari di dalam perusahaan tidak ada

masalah mengenai sistem Antara dan SAP maka tidak akan ada data yang dimasukkan ke dalam aplikasi IT Issue Log jika sehari terdapat kendala yang terjadi pada sistem Antara dan SAP maka data masalah tersebut akan dimasukkan ke dalam aplikasi *IT Issue Log* melalui *Form add IT Issue Log*.



Gambar 4.2 *Form Add IT Issue Log*

Setelah memasukkan data kedalam *Form Add IT Issue Log* dapat disimpan dengan klik tombol *save* yang terdapat pada bagian bawah *Form IT Issue Log*. Sehingga data tentang masalah sistem Antara atau SAP telah tersimpan kedalam *database* aplikasi *IT Issue Log*.

Data yang sudah terisi ke dalam aplikasi *IT Issue Log* dapat dilihat pada menu *IT Issue Log Report*. Pada menu *IT Issue Log Report* juga dapat digunakan untuk mengedit data dan *print* data *IT Issue Log Report*.



Gambar 4.3 Klik *IT Issue Log Report List*

The image shows a SAP application window titled 'IT/SAP Dept. PT. Ecco Tannery Indonesia'. The window contains a search bar and buttons for 'PRINT' and 'BACK'. Below the search bar is a table with 13 rows of data. The table has the following columns: Issue No, Created by, Company, System, Date Raised, Month, Week, Time Raised, check, Date Time R, and Raised by. The data in the table is as follows:

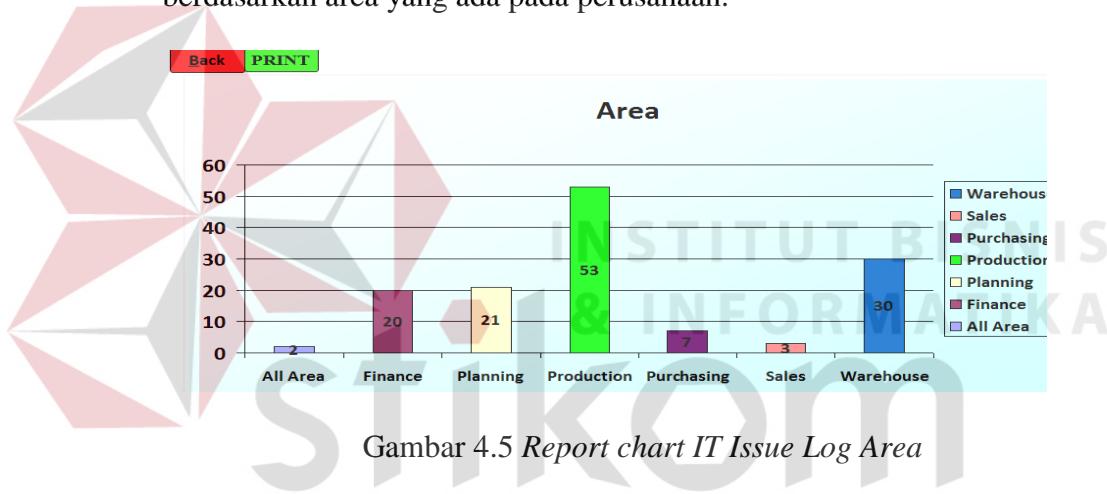
Issue No	Created by	Company	System	Date Raised	Month	Week	Time Raised	check	Date Time R	Raised by
1	SUD	PTKTT	ANTARA	5-Jan-15	Jan-15	2 2015	13:28 PM	1		RE
2	SUD	ETI	SAP	6-Jan-15	Jan-15	2 2015	10:21	1		UMB
3	SUD	ETI	SAP	6-Jan-15	Jan-15	2 2015	12:40	1		DEKA
4	SUD	PTKTT	ANTARA	7-Jan-15	Jan-15	2 2015	15:12 PM	1		HEK
5	SUD	ETI	ANTARA	8-Jan-15	Jan-15	2 2015	9:44	1		OTR
6	SUD	ETI	ANTARA	8-Jan-15	Jan-15	2 2015	13:33 PM	1		OTR
7	SUD	ETI	ANTARA	9-Jan-15	Jan-15	2 2015	16:27 PM	1		SIZ
8	SUD	ETI	ANTARA	10-Jan-15	Jan-15	2 2015	7:34	1		RE
9	SUD	ETI	ANTARA	14-Jan-15	Jan-15	3 2015	9:49	1		SIZ
10	SUD	ETI	ANTARA	15-Jan-15	Jan-15	3 2015	9:42	1		OTR
11	SUD	PTKTT	ANTARA	21-Jan-15	Jan-15	4 2015	9:42	1		RE
12	SUD	ETI	SAP	22-Jan-15	Jan-15	4 2015	10:08	1		UMB
13	SUD	PTKTT	ANTARA	24-Jan-15	Jan-15	4 2015	10:40	1		RE

Gambar 4.4 *IT Issue Log Report List*

Data IT *Issue Log* yang sudah diisi kedalam aplikasi dapat dijadikan *report* menggunakan *report pivot chart* yang sudah disediakan oleh *Software Microsoft Access*. *Report pivot chart* berfungsi untuk melaporkan data masalah sistem Antara dan SAP yang sudah diisikan kedalam aplikasi IT *Issue Log* berupa *chart*. Dalam aplikasi IT *Issue Log* terdapat 5 *Report chart* yang berbeda pelaporannya antara lain:

### 1. Report chart IT Issue Log Area

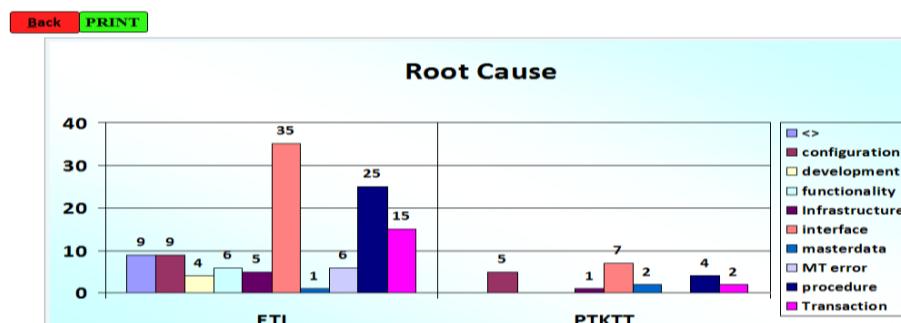
*Report chart* yang melaporkan masalah sistem Antara dan SAP berdasarkan area yang ada pada perusahaan.



Gambar 4.5 Report chart IT Issue Log Area

### 2. Report chart IT Issue Log Root Cause

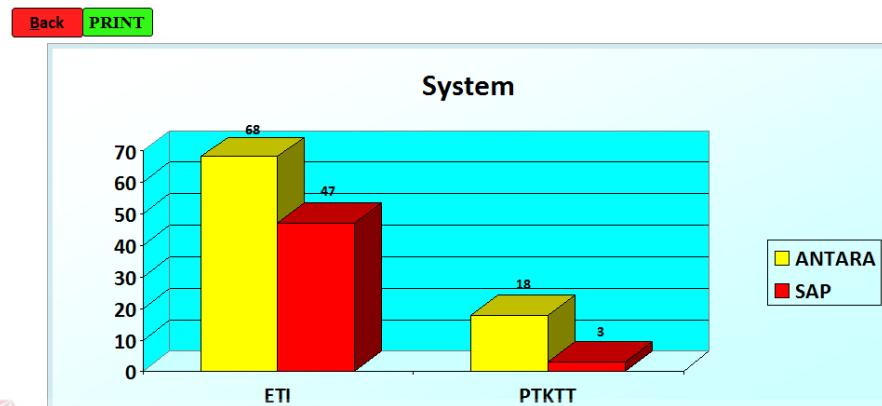
*Report chart* yang melaporkan akar masalah sistem Antara dan SAP pada perusahaan ETI dan perusahaan PTKTT.



Gambar 4.6 Report chart IT Issue Log Root Cause

### 3. Report chart IT Issue Log System

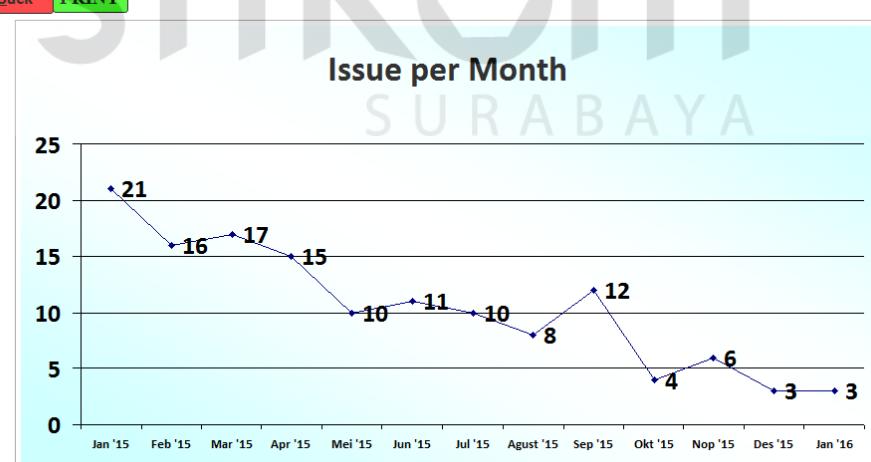
Report chart yang melaporkan masalah sistem Antara dan SAP pada perusahaan ETI dan PTKTT.



Gambar 4.7 Report chart IT Issue Log System

### 4. Report chart IT Issue Log Issue Per Month

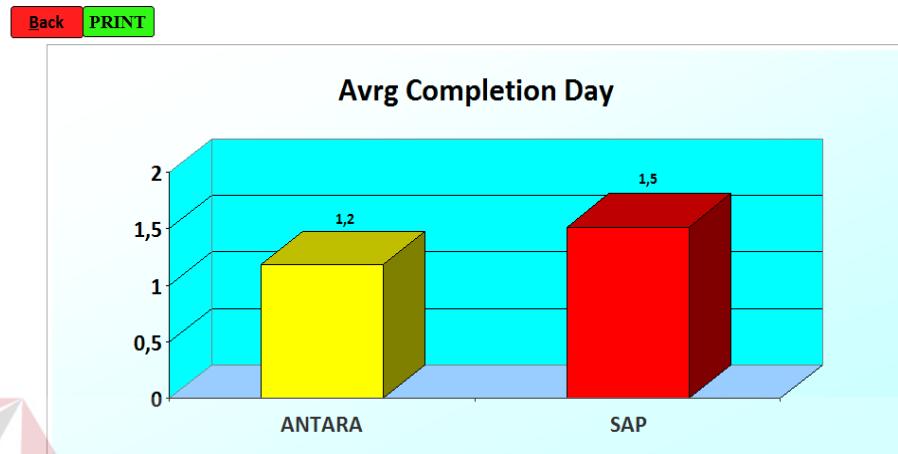
Report chart yang melaporkan masalah sistem Antara dan SAP setiap bulannya.



Gambar 4.8 Report chart IT Issue Log Issue Per Month

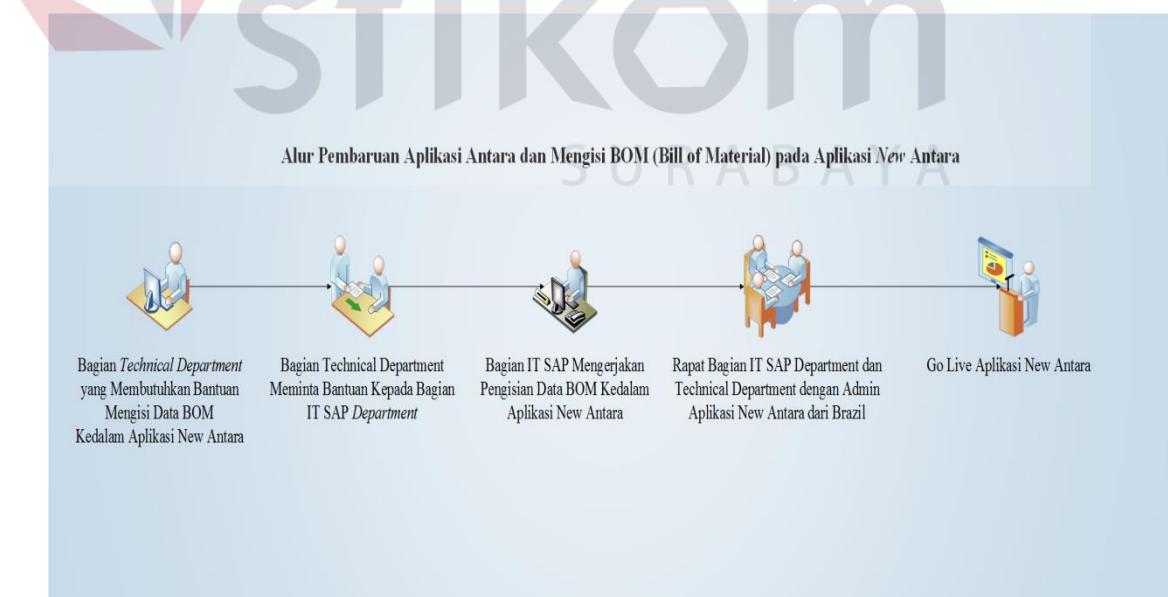
## 5. Report chart IT Issue Log Average Completion Day

Report chart yang melaporkan rata-rata penyelesaian masalah sistem Antara dan SAP.



Gambar 4.9 Report chart IT Issue Log Average Completion Day

### 4.4 Memasukan Data Bill Of Material (BOM) ke dalam aplikasi New Antara



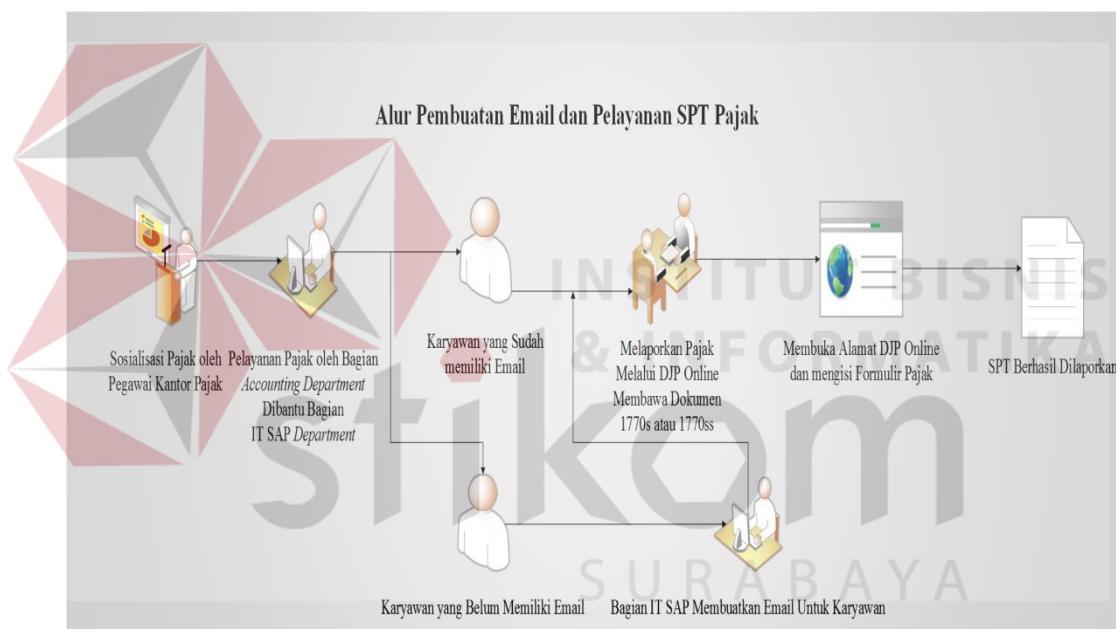
Gambar 4.10 Alur Memasukkan Data BOM Ke dalam Aplikasi New Antara

Pihak IT SAP *Depertment* juga turut menyiapkan program kerja untuk memperbarui aplikasi yang sudah digunakan oleh perusahaan. Aplikasi tersebut bernama Antara. Antara merupakan aplikasi buatan dari negara Brazil. Aplikasi ini tercipta dari bahasa pemrograman Java. Antara digunakan oleh semua karyawan baik yang bekerja dibagian produksi maupun *staff*. Antara berfungsi untuk meng-*capture* pergerakan *lot* yang diproduksi, dari mulai berupa bahan baku hingga menjadi kulit siap jual. Antara berisi banyak data yang dibutuhkan dalam memproduksi kulit, diantaranya adalah *Bill of Material* (BOM) menyiapkan semua data yang dibutuhkan mulai dari bahan baku yang digunakan, *Recipe* (mewarnai kulit), *operasional flow* dan *specification sheet* sehingga kulit siap diproduksi. Selama saya berada di PT Ecco Tannery Indonesia pekerjaan yang dapat dibantu adalah menyiapkan semua data yang dibutuhkan dalam pembuatan BOM.

Proses untuk memperbarui bahan baku yang digunakan dilakukan oleh bagian *Technical Department* perusahaan dan dibantu oleh bagian IT SAP *Department*. Langkah pertama yaitu mempersiapkan semua data BOM yang ada pada *Microsoft Excel* kemudian memasukkan data yang terdapat pada *Excel* ke dalam antara yang diantaranya adalah WB *material* yang digunakan, *recipe*, *composition*, *operational flow*, dan *specification sheet*. *Operational Flow* ini berfungsi sebagai urutan proses yang akan digunakan dalam setiap pembuatan kulit. Setelah *Operational Flow* dibuat maka langkah selanjutnya yaitu membuat *Recipe Description* dan *Composition description*. Setelah semua data selesai disiapkan, maka langkah selanjutnya yaitu memasukannya semua data satu persatu ke dalam *Specification Sheet*. Tahap *Specification Sheet* merupakan tahap terakhir

memasukkan BOM kedalam aplikasi Antara. *Specification Sheet* merupakan penggabungan antara *Operational Flow* dan *Recipe* yang digunakan dalam setiap jenis kulit. Jika *Specification Sheet* sudah selesai dibuat di antara maka selanjutnya yaitu mengirim data *specification* tersebut ke *system SAP*, dan setelah semua data sudah masuk ke dalam SAP, maka kulit sudah siap diproduksi sesuai dengan jenis yang diinginkan.

#### 4.5 Pelayanan SPT Pajak dan Pembuatan E-mail



Gambar 4.11 Alur Pembuatan E-mail dan Pelayanan Pajak

Setiap karyawan wajib untuk melaporkan pajak setiap tahunnya kepada pemerintah. Pada tahun 2016 terdapat perubahan untuk pelaporan pajak seorang karyawan. Pelaporan pajak yang dulunya harus datang ke kantor pajak untuk melaporkan pajak tapi pada tahun 2016 pelaporan pajak bisa dilakukan lewat *online*. Melalui layanan DJP online yang sudah disediakan oleh dirjen pajak para

karyawan tidak perlu datang ke kantor pajak untuk melapor tapi pelaporan pajak sudah dapat dilakukan di tempat kerja. Oleh karena itu pihak IT SAP *Depertment* memfasilitasi para karyawan supaya dapat melaporkan pajak di tempat kerja.

Pelaporan pajak melalui DJP *online* harus memiliki E-mail terlebih dahulu untuk dapat mengakses layanan DJP *online*. Cukup banyak karyawan perusahaan yang belum mempunyai E-mail terutama karyawan pada bagian produksi. Sehingga pembuatan E-mail untuk karyawan perusahaan yang belum memiliki akun E-mail dapat dilakukan di perusahaan. Untuk pembuatan E-mail dapat dilihat dibawah ini.

1. Langkah pertama untuk pembuatan E-mail yaitu pastikan dulu PC atau Laptop sudah terhubung dengan internet.
2. Jika PC atau Laptop sudah terhubung ke internet maka langkah berikutnya membuka alamat [www.gmail.com](http://www.gmail.com). Alamat tersebut dapat dibuka melalui *Search Engine* yang ada pada PC atau laptop.
3. Setelah mengakses [www.gmail.com](http://www.gmail.com) maka pembuatan email dapat dilakukan dengan cara klik buat akun pada halaman gmail.
4. Pada halaman buat akun akan ada beberapa kolom. Setiap kolom diisi sesuai pertanyaan dan identitas diri lengkap. Kemudian klik langkah berikutnya.
5. Kemudian akan ada tahap verifikasi akun. Pastikan verifikasi akun berhasil. Verifikasi menggunakan nomor *handphone* yang sudah dimasukkan kedalam kolom identitas sebelumnya. Verifikasi dapat melalui pesan teks atau panggilan suara, pilih salah satu. Kemudian klik lanjutkan.

6. Masukkan kode verifikasi yang telah dikirim melalui pesan teks kedalam kolom. Kemudian klik lanjutkan.
7. Setelah itu akun google sudah dapat digunakan.

Setiap karyawan yang sudah memiliki E-mail bisa langsung untuk mendaftarkan diri di layanan DJP *online*. Langkah pertama mendaftar yaitu membuka situs DJP *online* dengan membuka situs web <https://djponline.pajak.go.id/account/login>. Setelah masuk halaman DJP *Online* bagi karyawan yang belum mendaftar bisa langsung mendaftar dengan klik “anda belum terdaftar? Daftar disini”. Kemudian membuka *link* tersebut maka langkah selanjutnya yaitu mengisi informasi data diri lengkap sesuai yang ada. Masukan nomor NPWP dan EFIN. Lalu sesuaikan kode keamanan dengan kode keamanan yang ada di bawah. Klik Verifikasi. Maka pembuatan akun DJP sudah berhasil.

Jika para karyawan sudah mendaftar DJP *online* sendiri maka pihak IT SAP *Department* dapat membantu untuk memasukkan Wajib lapor pajak. Untuk langkah pertama harus *login* terlebih dahulu. Kemudian tinggal membuat SPT Pajak dan mengisi data sesuai dokumen untuk lapor pajak. Sehingga pelaporan untuk wajib pajak tidak perlu datang ke kantor pajak.

#### **4.5 Pembuatan Aplikasi Pencatatan Administrasi Pada Bagian Human**

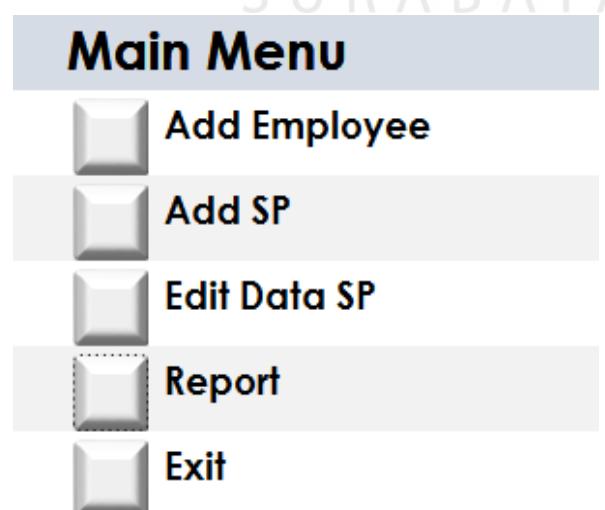
##### ***Resource Department PT Ecco Tannery Indonesia***

Aplikasi karyawan yang tercipta dari *Microsoft Access* dan digunakan oleh bagian *Human Resource Department* (HRD) untuk data pencatatan informasi administrasi karyawan. Aplikasi yang tercipta dari *Software Microsoft Access* ini dapat menyimpan data administrasi karyawan yang sudah melakukan training dan

telah terkena Surat Peringatan (SP). Pembuatan aplikasi menggunakan *Microsoft Access* diharuskan untuk membuat tabel terlebih dahulu. Pembuatan tabel berfungsi untuk menyimpan *database* yang akan disimpan kedalam aplikasi. *Database* merupakan kumpulan arsip data berbentuk tabel yang saling berkaitan untuk menghasilkan informasi, untuk aplikasi *Training* dan aplikasi Surat Peringatan (SP) data yang akan dimasukkan adalah data berapa kali seorang karyawan sudah melakukan *Training* dan berapa kali seorang karyawan mendapatkan SP. Setalah tabel tercipta maka langkah selanjutnya pembuatan *form*, *report*, dan *switchboard* dapat ditambahkan untuk memperindah aplikasi.

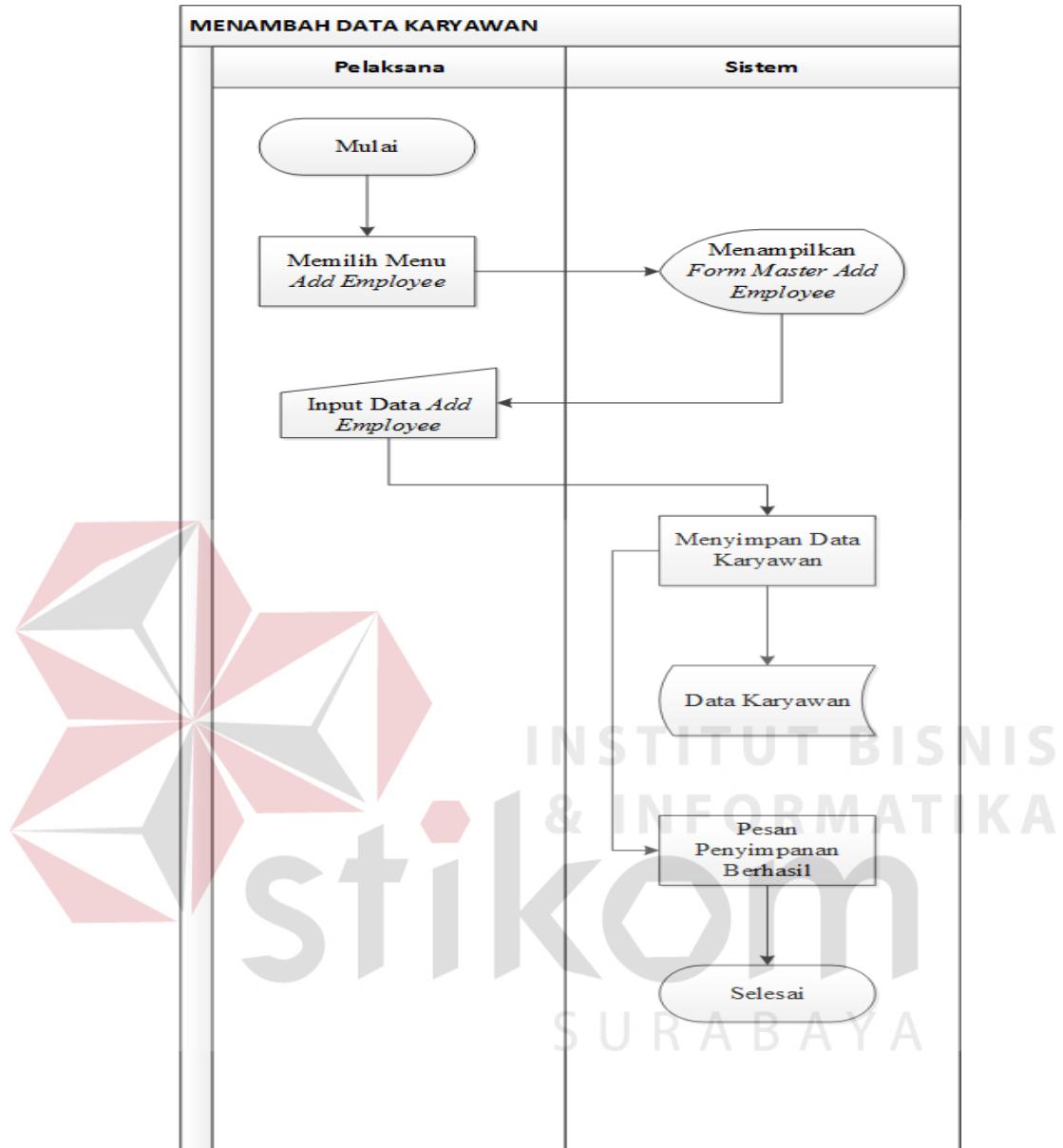
#### 4.6 Aplikasi SP

Aplikasi SP merupakan aplikasi yang digunakan oleh bagian HRD untuk menyimpan data karyawan yang sudah terkena SP, sehingga data untuk karyawan yang telah terkena SP dapat disimpan ke dalam aplikasi dan dapat dilihat *report history* untuk seorang karyawan. Pada aplikasi SP ini juga dapat dilihat karyawan tersebut masih aktif ataupun sudah keluar dari perusahaan.



Gambar 4.12 Tampilan *Main Menu* Aplikasi SP

#### 4.6.1 System Flow Aplikasi SP



Gambar 4.13 System Flow Menambah Data Karyawan

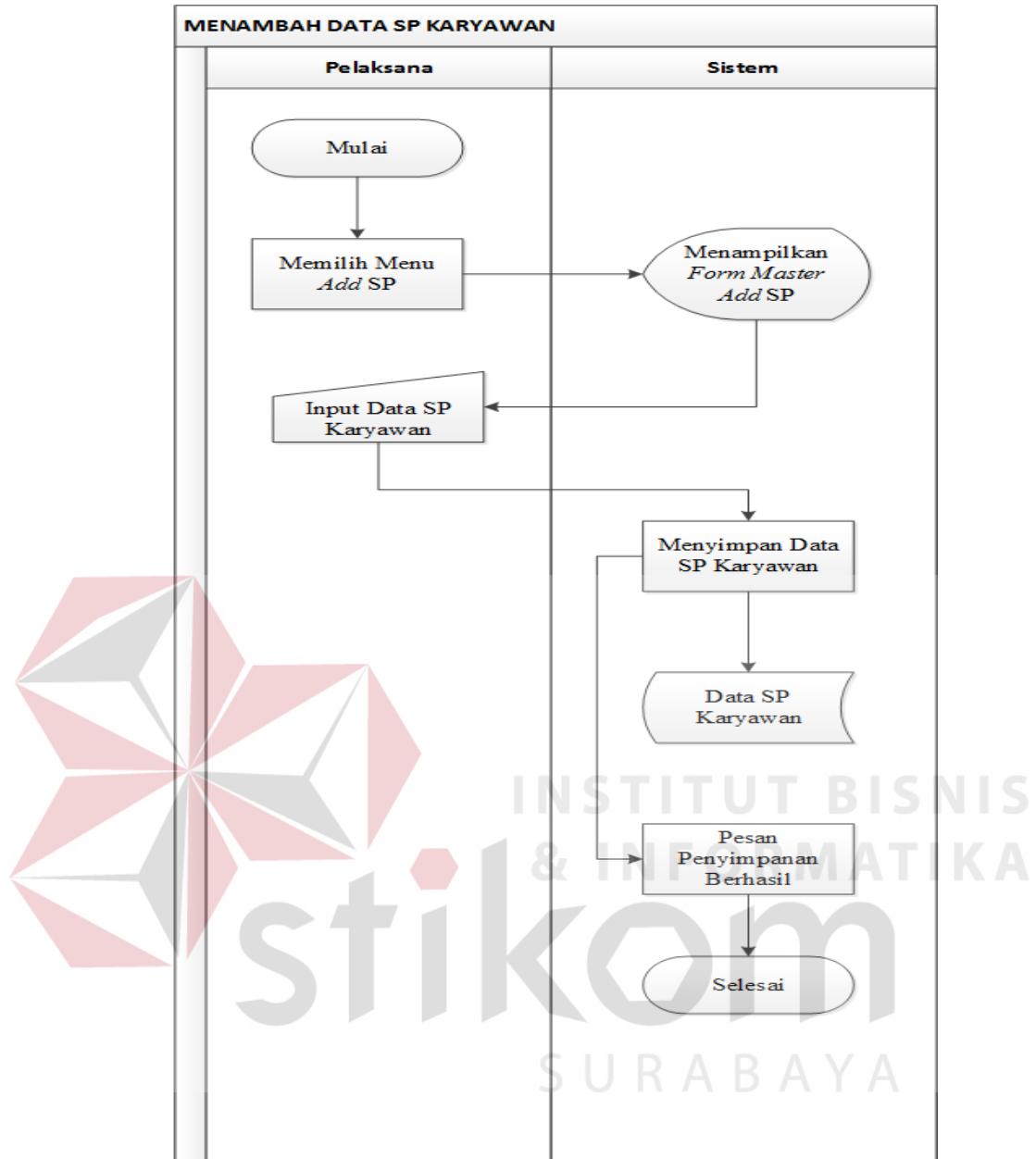
*Objective* : untuk menambah data karyawan

*Input* : data karyawan (*salary number* dan nama karyawan)

*Proses* : menyimpan data karyawan

*Output* : data pada tabel *employee* bertambah

*Actor* : pelaksana



Gambar 4.14 *System Flow Menambah Data SP Karyawan*

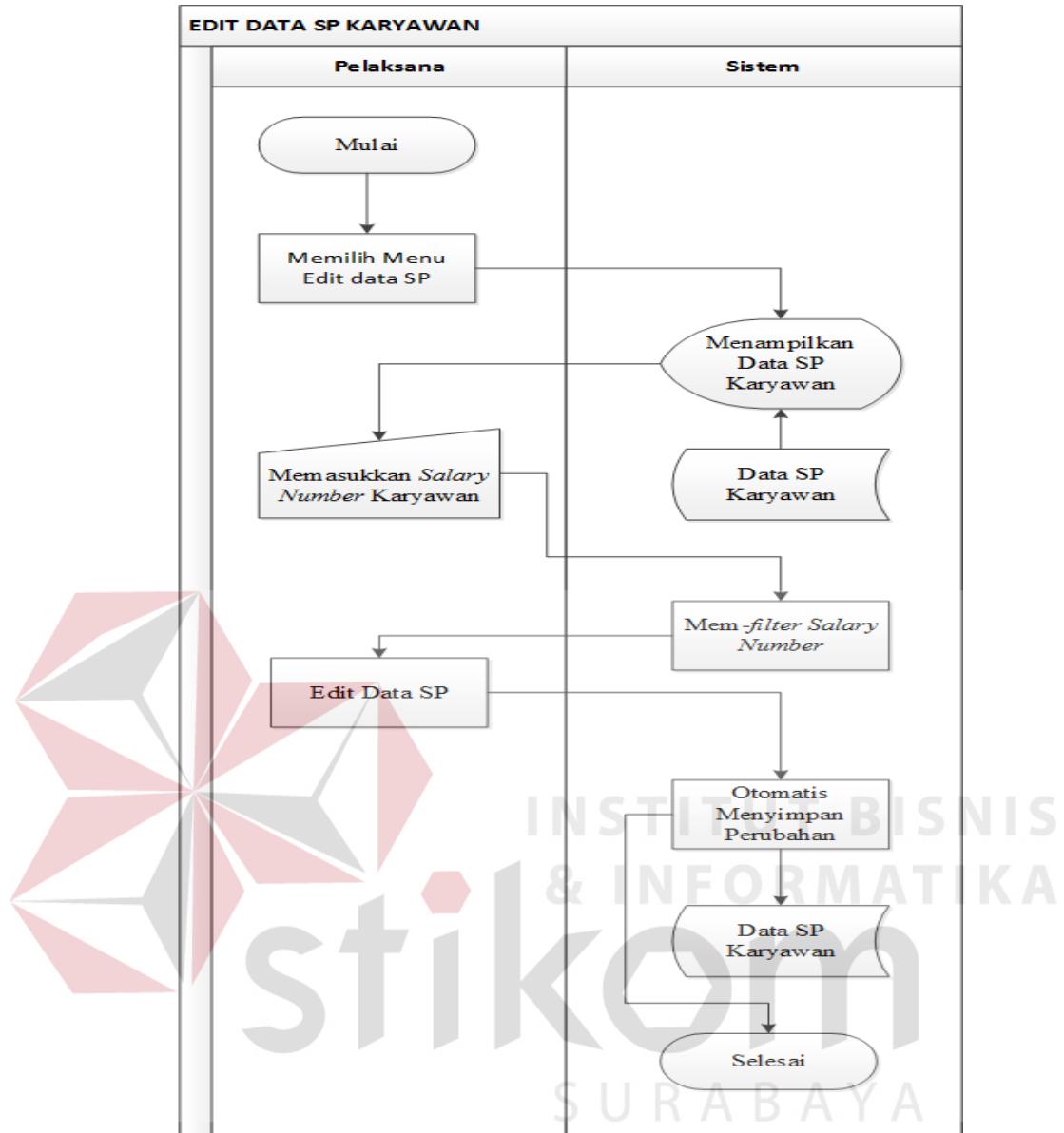
*Objective* : untuk menambah data SP karyawan

*Input* : data SP karyawan

*Proses* : menyimpan data SP karyawan

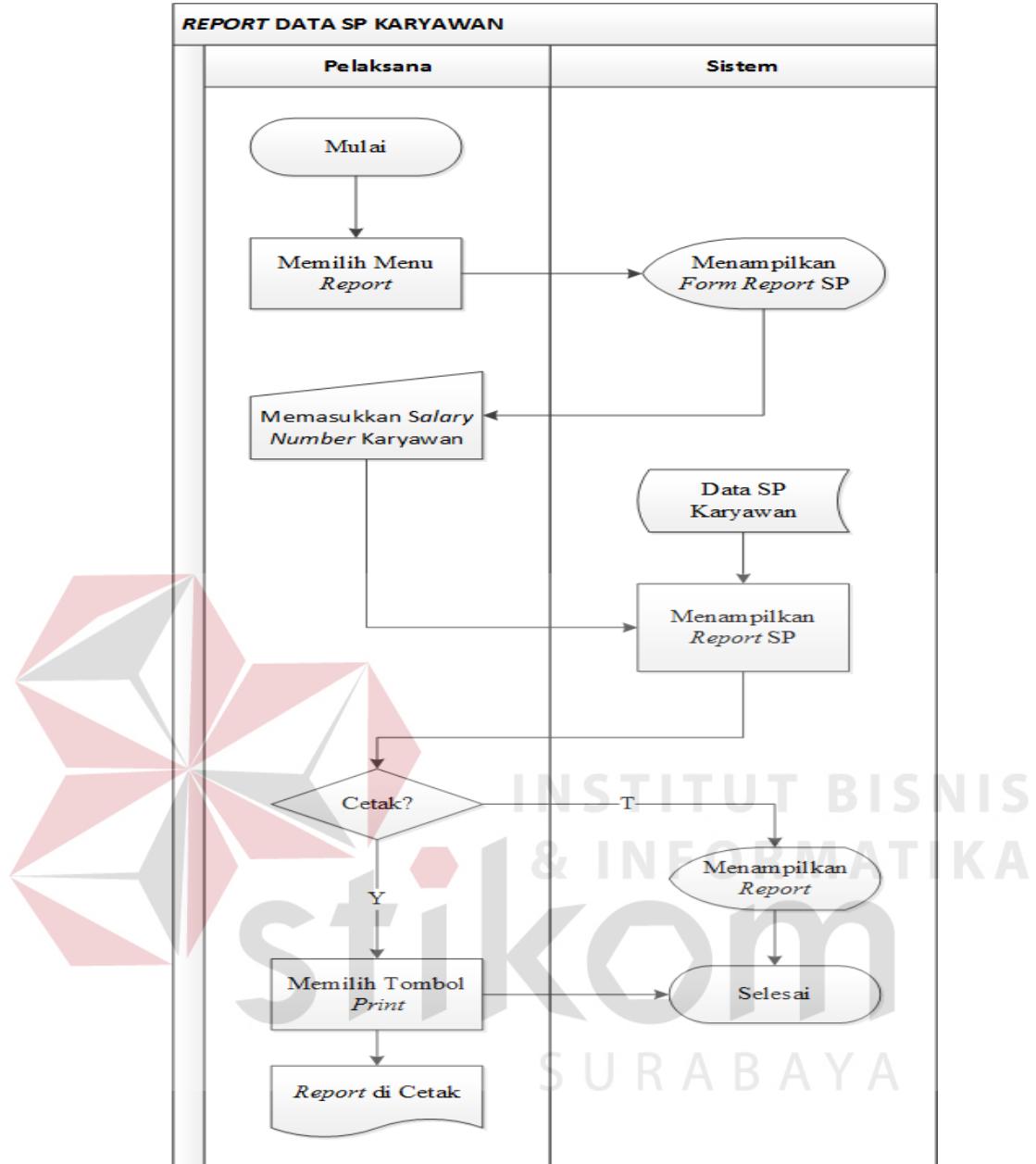
*Output* : data pada tabel SP bertambah

*Actor* : pelaksana



Gambar 4.15 System Flow Edit Data SP Karyawan

- Objective* : untuk edit data SP karyawan
- Input* : *salary number* karyawan
- Proses* : edit data SP karyawan pada tabel SP yang ditampilkan dengan *subform*, pencarian data dengan mengetikkan *salary number*
- Output* : data pada tabel SP berubah
- Actor* : pelaksana



Gambar 4.16 System Flow Report Data SP Karyawan

*Objective* : untuk *report* data SP karyawan

*Input* : *salary number* karyawan

*Proses* : *report* data SP karyawan dengan mengetikkan *salary number*

*Output* : *report* menemukan data SP karyawan

*Actor* : pelaksana

#### 4.6.2 Rancangan Tabel Aplikasi SP

##### A. Tabel *Employee*

Nama Tabel : *Employee*

Primary Key : Salnum

Fungsi : Menyimpan data Karyawan

Tabel 4.2 Struktur Tabel *Employee*

NO	Field	Type Data	Field Size	Format	Constraint
1	Salnum	Text	25		PK
2	Employee_name	Text	120		
3	Status	Text	255		

##### B. Tabel SP

Nama Tabel : SP

Primary Key : -

Fungsi : Menyimpan data SP Karyawan

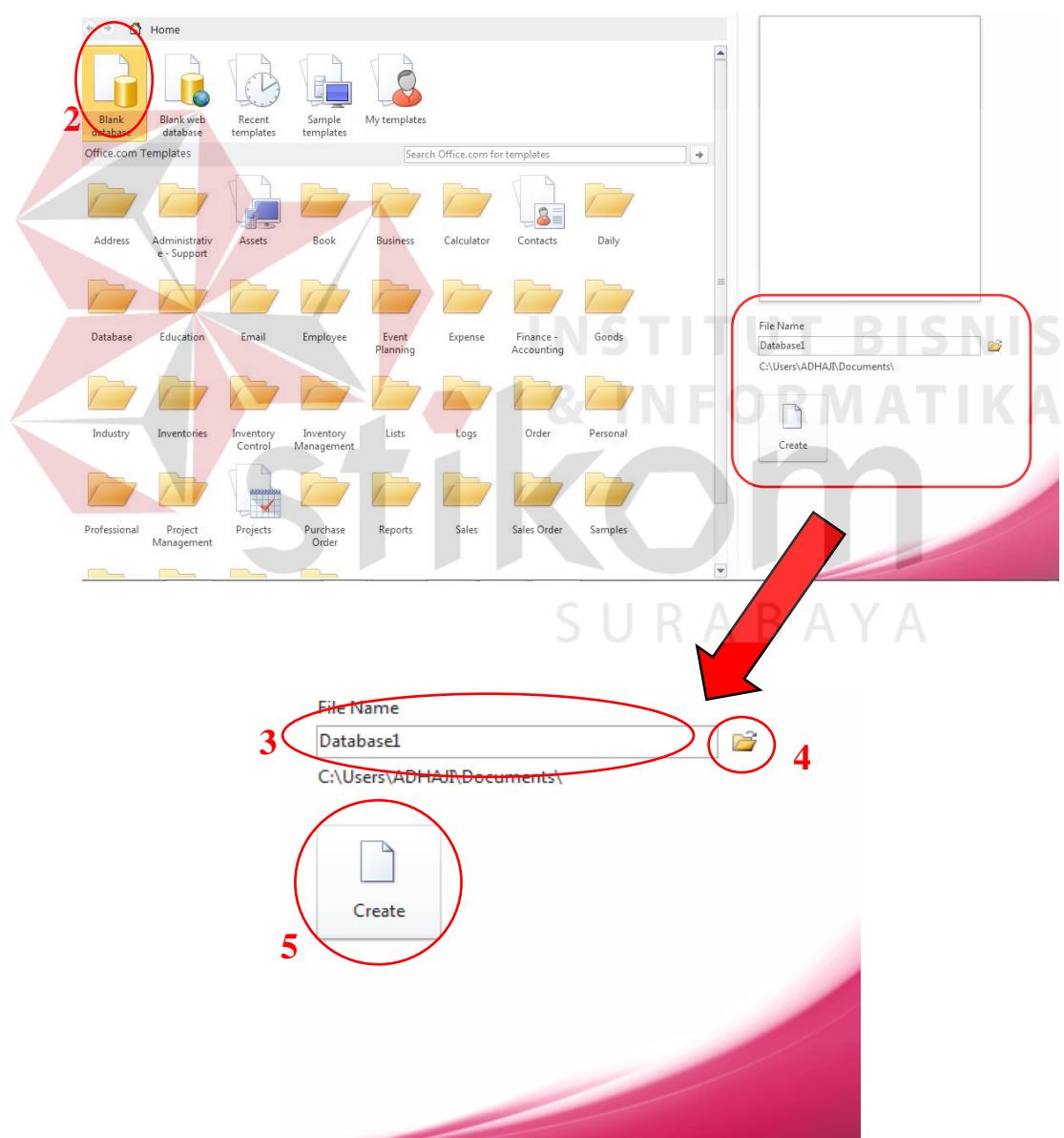
Tabel 4.3 Struktur Tabel SP

NO	Field	Type Data	Field Size	Format	Constraint
1	Salnum	Text	25		FK
2	Department	Text	100		
3	Section/Working Area	Text	100		
4	Positions	Text	100		
5	Type of SP	Text	20		
6	Date SP	Date/Time		Medium Date	
7	Expired	Date/Time		Medium Date	
8	Detail	Memo			

### 4.6.3 Pembuatan Aplikasi SP

#### A. Membuat Database

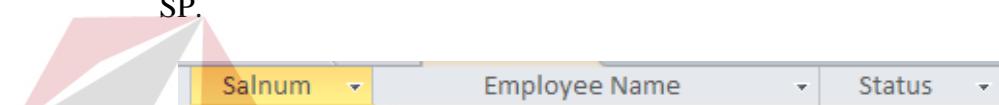
1. Buka aplikasi *Microsoft Access*
2. Pilih *blank database*.
3. Berikan nama *database*.
4. Pilih tempat penyimpanan *database*
5. Klik *Create*



Gambar 4.17 *Create Database*

### B. Create Table

Untuk aplikasi Karyawan terdapat 2 tabel yaitu tabel *employee* dan tabel SP. Tabel *employee* berfungsi untuk menyimpan ID karyawan, nama karyawan dan status karyawan yang ada pada PT Ecco Tannery Indonesia sedangkan tabel SP berfungsi untuk menyimpan data pelanggaran yang dilakukan oleh karyawan. Pada tabel SP terdapat 8 *field* yaitu Salnum (*link* dari ID Karyawan), *Department*, *section/working area*, *Position*, *Type of SP*, *Date SP*, *Expires*, dan *Detail*. Berikut *field* dari tabel *Employee* dan SP.



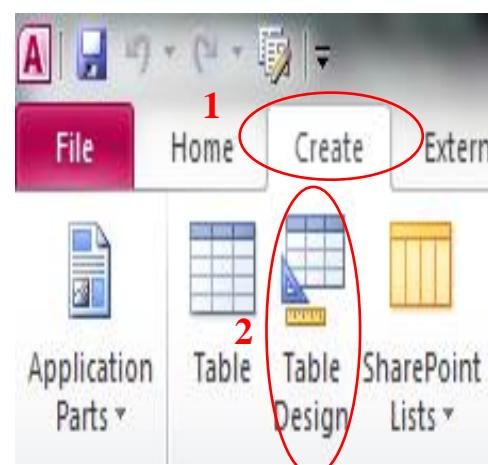
Gambar 4.18 Field Employee



Gambar 4.19 Field SP

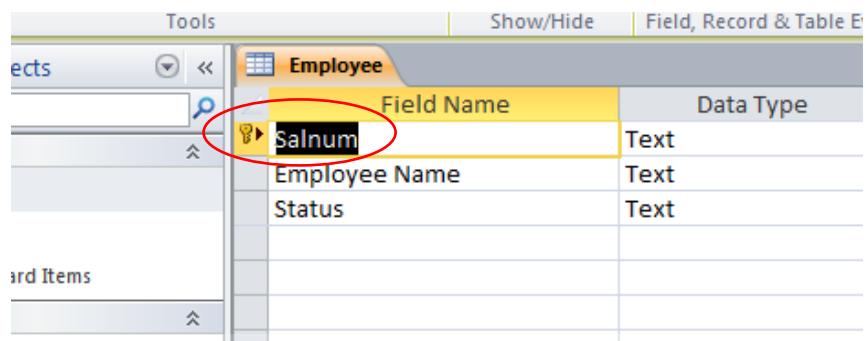
Untuk langkah pembuatan tabel bisa dilihat dibawah ini.

1. Pilih Tab *create* kemudian pilih *toolbar table design*.



Gambar 4.20 Create Table Design

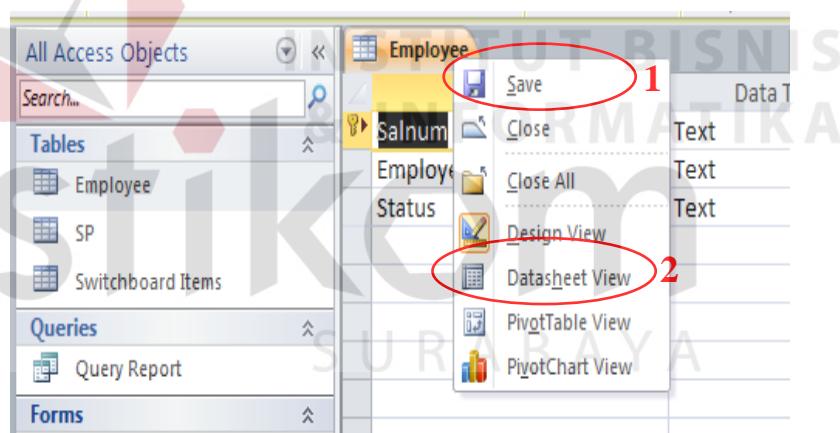
2. Isikan nama *field Name* dan *Data Type*
3. Jadikan salah satu *field* menjadi *Primary Key* Untuk tabel *employee* yang dijadikan *Primary key* adalah Salnum. Karena Salnum adalah Nomor ID karyawan Pada PT Ecco Tannery Indonesia.



Field Name	Data Type
Salnum	Text
Employee Name	Text
Status	Text

Gambar 4.21 Pembuatan *Primary Key*

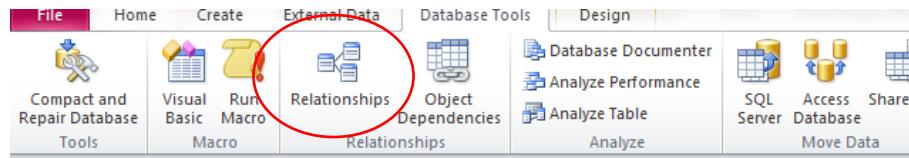
4. Simpan tabel kemudian klik *datasheet view*



Gambar 4.22 *Save table*

Langkah pembuatan tabel SP sama dengan pembuatan tabel *employee*. Jika tabel *employee* dan tabel SP sudah dibuat maka langkah selanjutnya yaitu menghubungkan kedua tabel dengan *relationship* supaya antara tabel *Employee* dan tabel SP dapat terhubung. Untuk langkah pembuatan *Relationship* dapat dilihat di bawah ini:

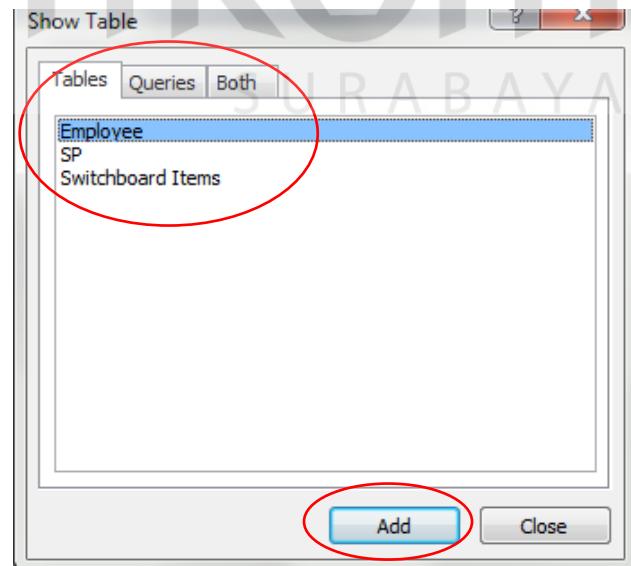
1. Buka toolbar database tools. Pilih Relationships.



Gambar 4.23 Pilih Toolbar Relationships

2. Jika tabel yang telah dibuat tidak muncul, dapat menampilkannya dengan cara klik kanan kemudian *show table*, pilih tabel *employee* dan SP yang akan dijadikan *relationship* kemudian klik *adds*.

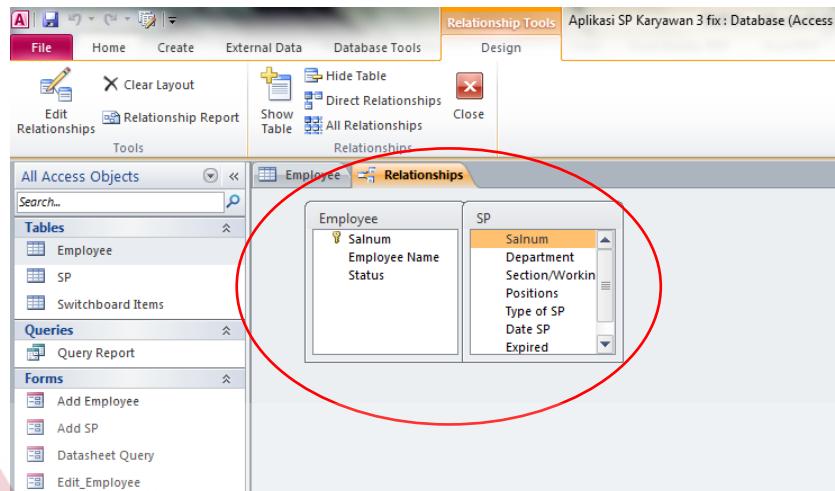
Gambar 4.24 Show Table



Gambar 4.25 Add Table

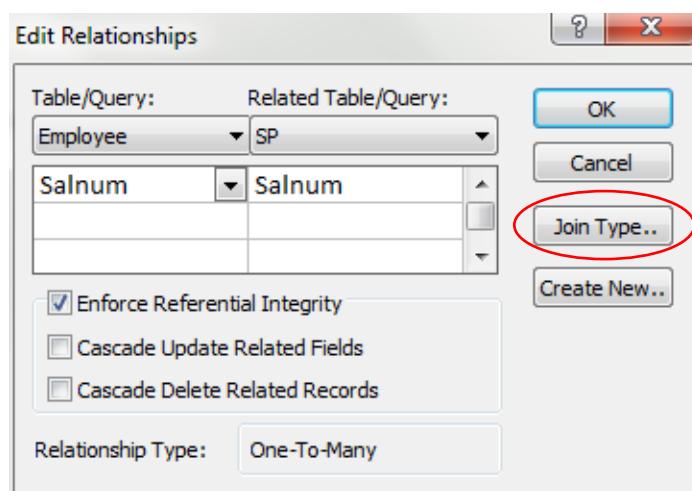
3. Kemudian akan muncul 2 tabel yang belum ada *relationships* nya.

Untuk menambahkan *relationship drag and drop* nama *field* yang sama kemudian akan muncul jendela baru.

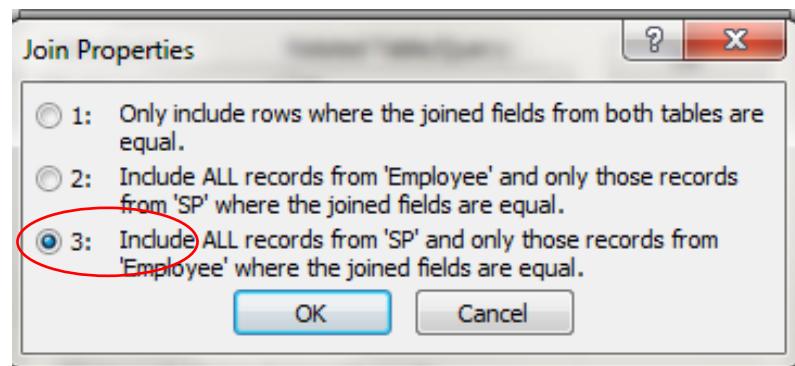


Gambar 4.26 Drag and Drop Field

4. Gunakan join *type* jika data yang digunakan mempunyai *setting-an* sendiri. Pilih join *type* yang ketiga untuk memasukkan semua *field* yang ada pada tabel SP ke dalam salnum pada employee melalui *relationships* kemudian klik *OK*.



Gambar 4.27 Tampilan Edit Relationships

Gambar 4.28 *Join Table*

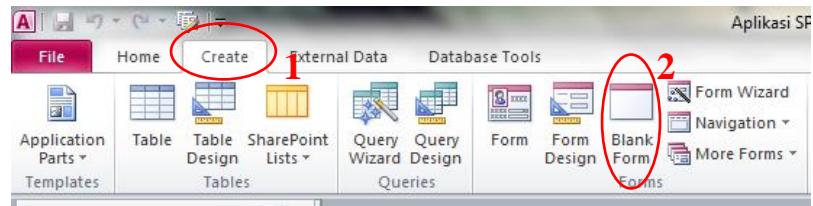
5. Hasil tabel yang telah diberi *relationships*.

Gambar 4.29 Hasil *Relationships*

### C. *Create Form Add SP*

Setelah tabel dan *relationship* tercipta maka langkah selanjunya yaitu membuat formulir *entri* yang berfungsi untuk memasukkan data baru ke dalam tabel yang telah tercipta. Untuk pembuatan *Form add SP* dapat dilihat di bawah ini.

1. Buka toolbar *create*, pilih *blank form*.

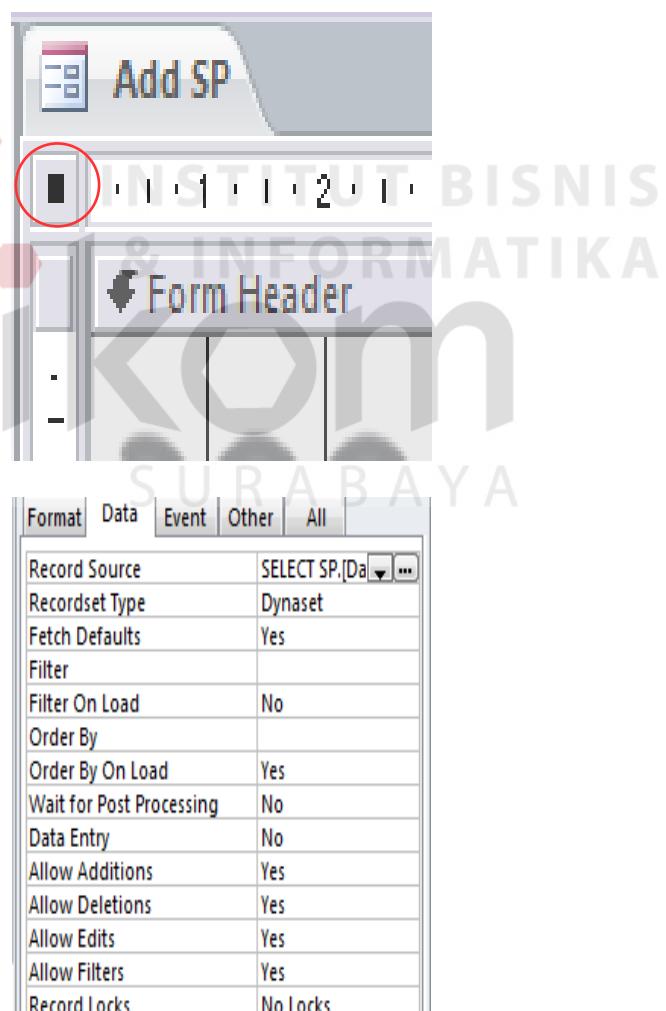


Gambar 4.30 Pilih *Blank Form*

2. *Setting record source* formulir dengan tabel SP pada *properties*.

Perhatikan tanda pada ujung kiri atas *form* hidup untuk men-setting

*form*. Kemudian pada *record source* pilih tabel SP.



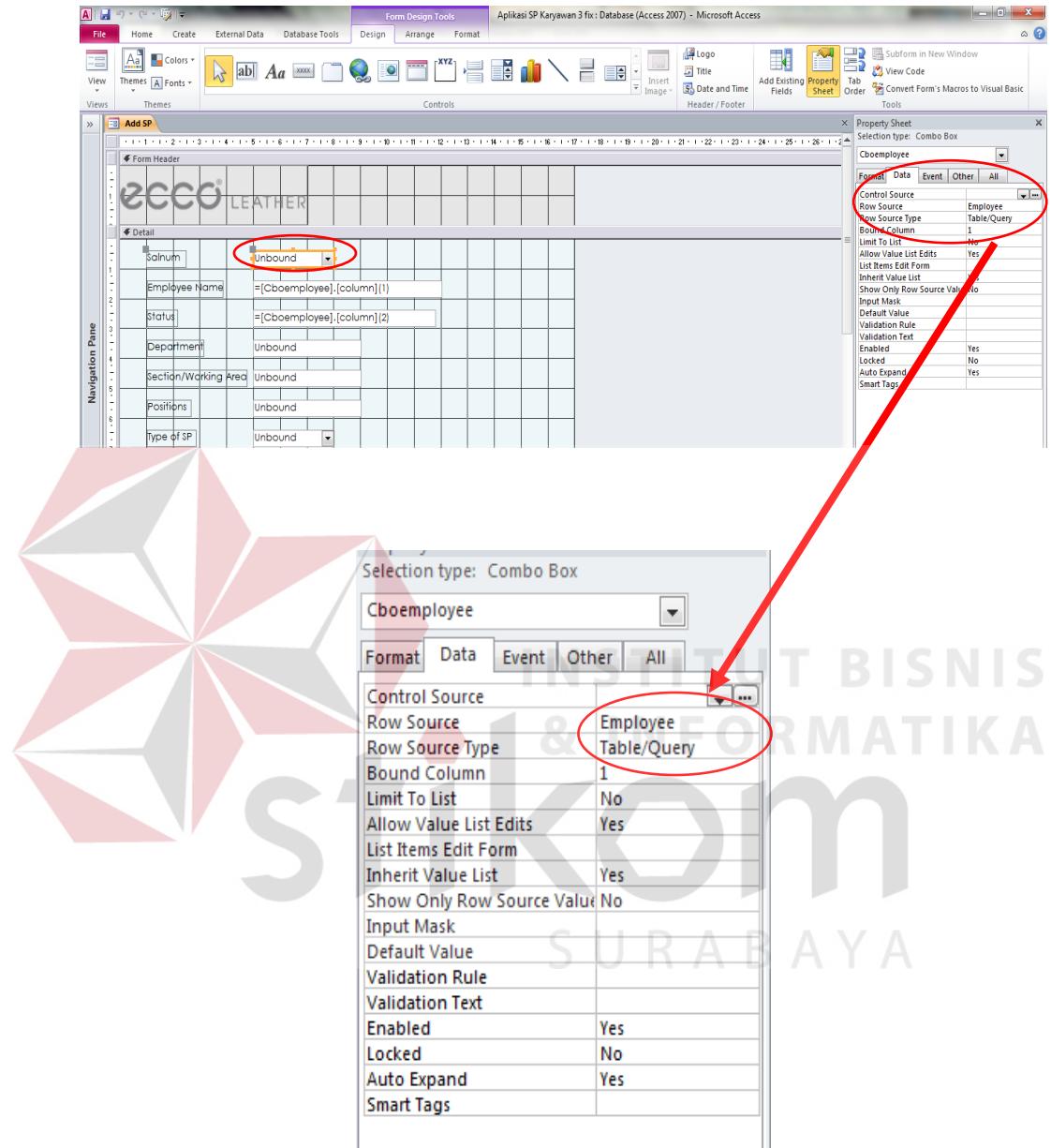
Gambar 4.31 *Setting Form SP*

3. Isikan *blank form* dengan *text box* sesuai jumlah *field* yang ada pada tabel SP (Salnum dan *Type Of SP* menggunakan *combo box*) untuk *combo box* akan ada perbedaan dengan *text box*. *Combo box* memiliki arah panah kebawah sedangkan *text box* tidak memiliki arah panah kebawah.

Salnum	Unbound	▼	Combo Box
Employee Name	=[Cboemployee].[column](1)		
Status	=[Cboemployee].[column](2)		
Department	Unbound	▼	Text Box
Section/Working Area	Unbound		
Positions	Unbound		
Type of SP	Unbound	▼	Text Box
Date SP	Unbound		
Expired	Unbound		
Detail	Unbound		

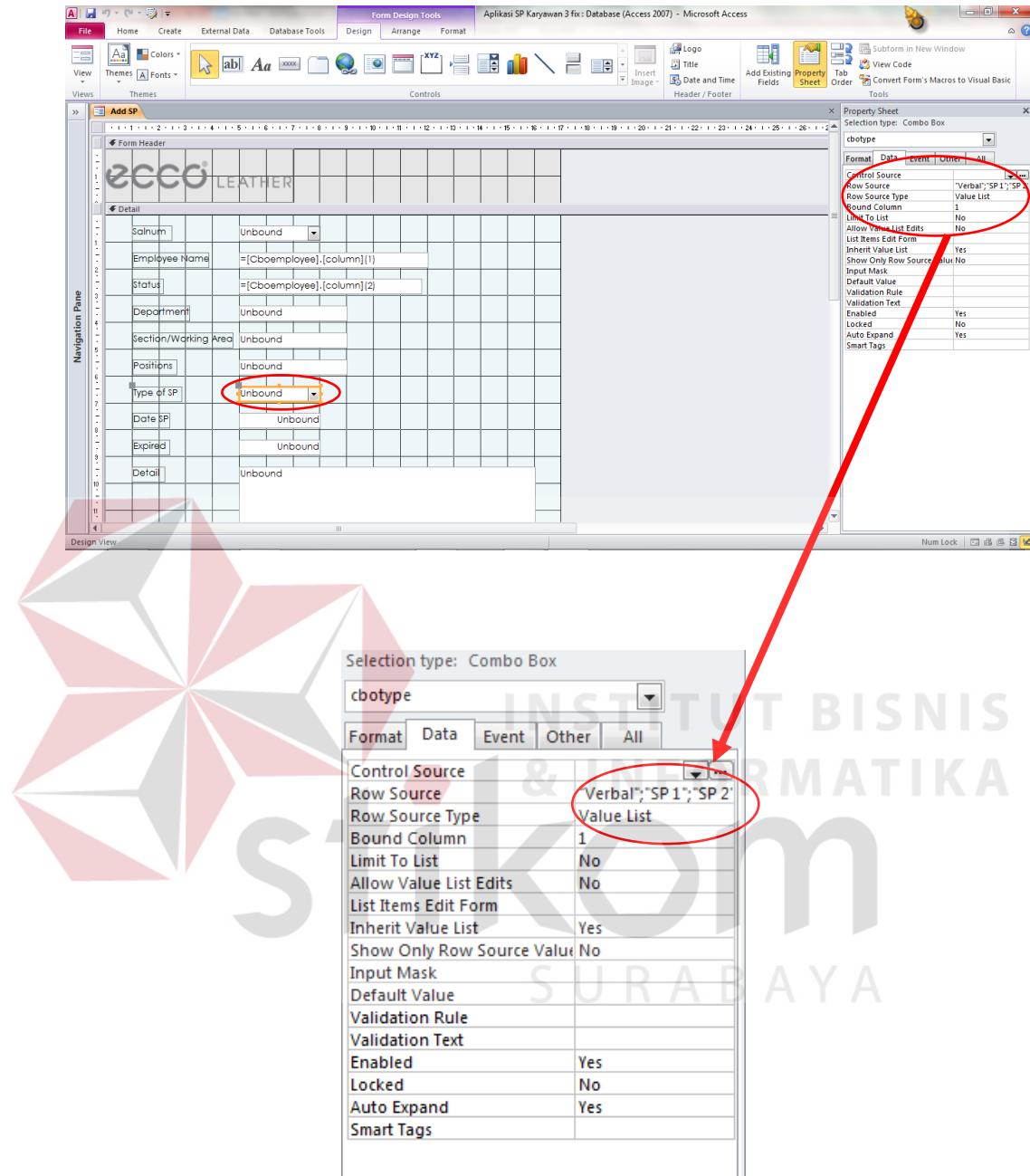
Gambar 4.32 Isikan Field

4. Untuk salnum *combo box* data diambil dari salnum pada tabel *Employee*.



Gambar 4.33 *Combo Box* Salnum

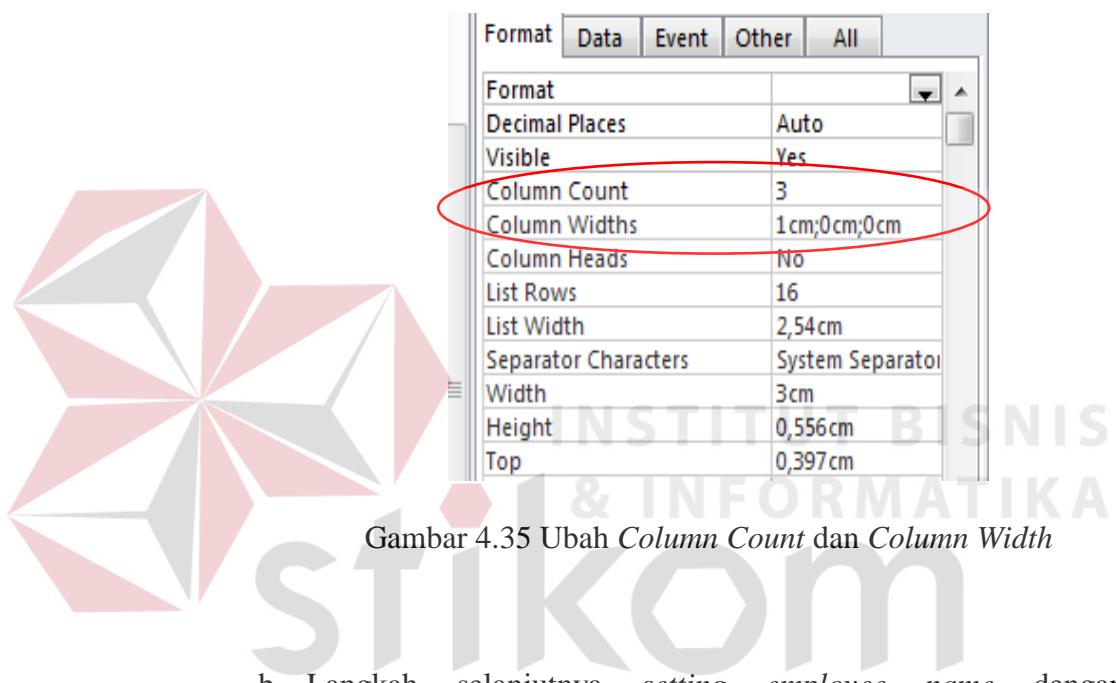
5. *Combo Type of SP pada properties setting menggunakan value list.*



Gambar 4.34 *Combo Box Type of SP*

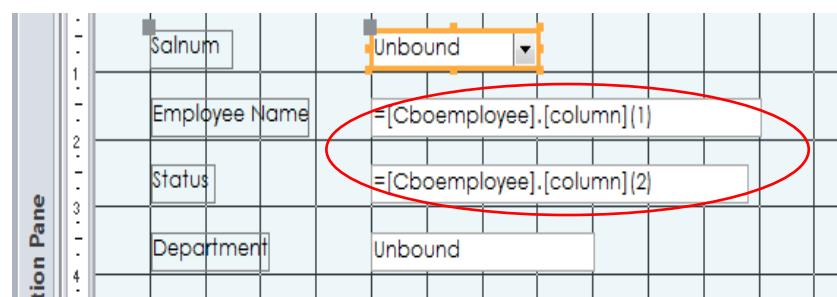
6. Untuk *Employee Name* dan Status supaya dapat otomatis keluar nama dan status karyawan setelah memilih salnum dapat dilakukan dengan cara.

- a. Setting terlebih dahulu nama dan *properties combo box* salnum, *column count* dari 1 menjadi 3 dan *column widths* 1 cm menjadi 1 cm ; 0 cm ; 0 cm, Perintah diatas berfungsi untuk menampilkan *database* pada tabel *Employee* akan tetapi tidak akan nampak pada *combo box* salnum



Gambar 4.35 Ubah *Column Count* dan *Column Width*

- b. Langkah selanjutnya setting *employee name* dengan menuliskan pada “*Unbound*” ganti dengan “=[Cboemployee].[column](1)” dan status “*Unbound*” ganti dengan “=[Cboemployee].[column](2)”.



Gambar 4.36 Setting *Unbound*

Perintah diatas berfungsi untuk menampilkan otomatis data ketika Salnum terisi. *CBOemployee* berfungsi memanggil data pada *combo box employee*, *column 1* berfungsi memanggil data *column* keberapa pada *CBOemployee*. Setelah semua di *setting* seperti langkah diatas maka akan secara otomatis muncul data nama karyawan dan status ketika Salnum terisi.

7. Jika lembar formulir sudah diisi seperti langkah diatas selanjutnya menambahkan tombol *button*, pada aplikasi karyawan menggunakan 4 tombol *button* yaitu *Add*, *Save*, *Cancel*, *Back*.



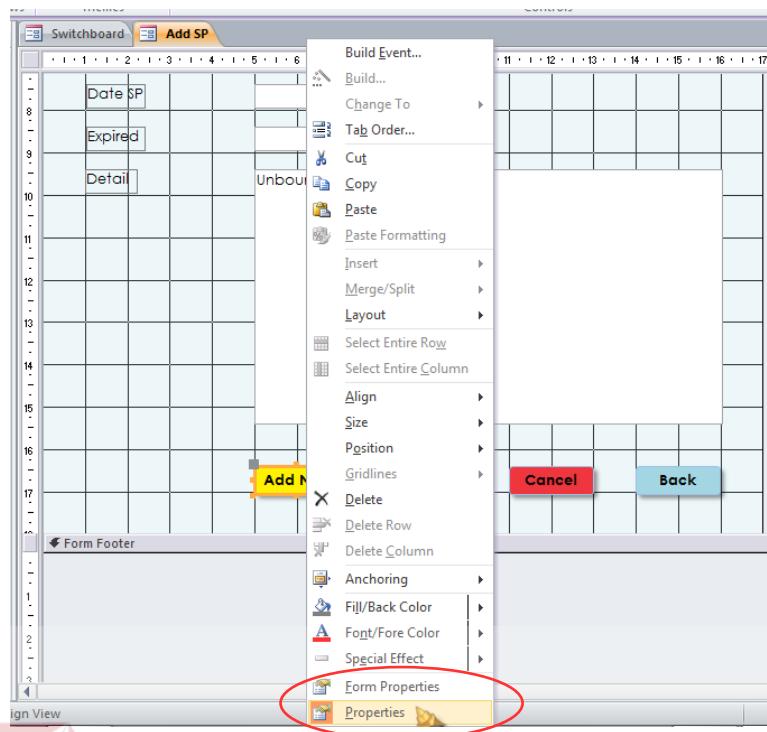
Gambar 4.37 Tombol *Button*

Supaya tombol *button* dapat berfungsi untuk kedua tabel *Employee* dan SP maka terlebih dahulu membuat *event* pada setiap tombol *button*. Setiap tombol *button* memiliki fungsi masing-masing dan langkah pembuatan *event* dapat dilihat dibawah ini.

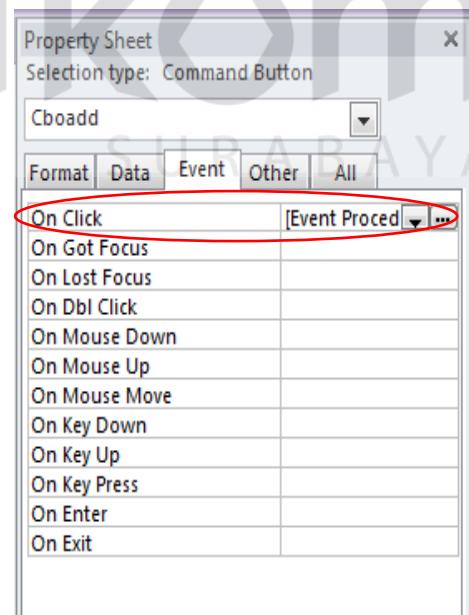
a. *Event* Tombol *Add new*

Tombol *Add New* berfungsi untuk menambahkan data baru atau membuat lembar formulir menjadi kosong dan siap untuk diisi kembali.

- 1) Untuk *event* tombol *add new* dapat ditambahkan melalui klik kanan pilih *properties*.

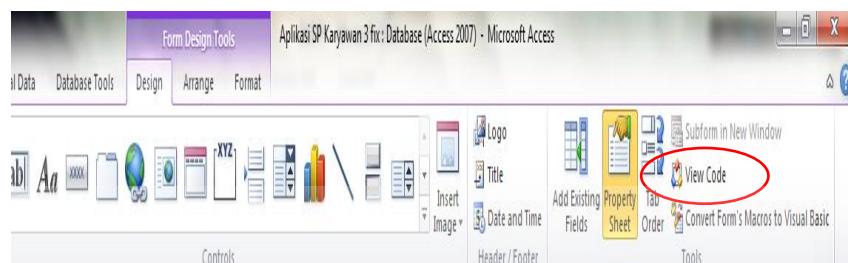
Gambar 4.38 Klik *Properties*

2) Pilih tab *event* lalu pilih *On Click*.

Gambar 4.39 *On Click* pada Tab *Event*

- 3) Pada *event On Click* berfungsi untuk *add new* secara otomatis.

Tombol *add new* akan menjalankan *code* yang telah dibuat dalam *event On Click*. Sebelum memasukkan *code* untuk tombol *add new* langkah pertama untuk men-*setting* formulir agar dapat menggunakan 2 atau lebih *database* yaitu klik tombol *View Code* pada pojok kanan atas *form design*.



Gambar 4.40 *View Code*

- 4) Setelah itu akan muncul jendela baru dengan halaman kosong.

Pada halaman kosong tersebut isikan *code* seperti dibawah ini.

```
Option Compare Database
Dim db As DAO.Database
Dim rssp As DAO.Recordset
Dim rse As DAO.Recordset
```

 A screenshot of the Microsoft Visual Basic Editor. The window title is 'Aplikasi SP Karyawan 3 fix - Form\_Add SP (Code)'. The code window shows the following VBA code:
 

```
(General)
Option Compare Database
Dim db As DAO.Database
Dim rssp As DAO.Recordset
Dim rse As DAO.Recordset
Private Sub Cboadd_Click()
```

 The first four lines of code ('Option Compare Database' and 'Dim db As DAO.Database' followed by two empty lines) are highlighted with a red circle.

Gambar 4.41 *Code Setting Database*

Perintah *code* diatas berfungsi untuk menggabungkan *database* tabel *employee* dan tabel *SP* kedalam formulir. Sehingga formulir dapat menggunakan lebih dari 1 *database*. Setelah *setting-an* untuk *database* formulir dibuat maka formulir dapat menggunakan lebih dari 1 *database*,

- 5) Sebelum memasukkan *code* untuk tombol *add new* pertama *setting* formulir untuk menjalankan *database* dengan cara *properties event* pada formulir atau langsung masuk ke dalam *view code* lalu masukan *code* seperti dibawah ini.



```

Private Sub Form_Load()
Set db = CurrentDb
Set rssp = db.OpenRecordset("SP",
dbOpenDynaset)
Set rse = db.OpenRecordset("Employee",
dbOpenDynaset)
End Sub

```

---

```

Private Sub Form_Load()
Set db = CurrentDb
Set rssp = db.OpenRecordset("SP", dbOpenDynaset)
Set rse = db.OpenRecordset("Employee", dbOpenDynaset)
End Sub

```

Gambar 4.42 *Code Setting Form Load*

Perintah *code* diatas berfungsi ketika saat formulir pertama kali dibuka akan menggunakan *database* yang telah di-*setting* dalam formulir. Untuk aplikasi karyawan *database* yang

digunakan yaitu RSSP (*Record Set SP*) dan RSE (*Record Set Employee*). Jika telah di-setting *code* seperti diatas maka formulir akan dapat membaca dan menggunakan 2 atau lebih *database* untuk digunakan di formulir. Seperti *code* yang digunakan pada tombol *save* yang pasti berguna menyimpan data yang akan dimasukkan kedalam tabel yang akan dituju.

- 6) Setelah formulir di-setting seperti diatas maka langkah selanjutnya untuk pembuatan fungsi *code* tombol *add new* dapat dilakukan.

Langkah pertama yaitu memberi *event on* klik pada tombol *add new* kemudian masukkan *code* pada tombol *add new*.

```
Private Sub Cboadd_Click()
    CboEmployee.Value = ""
    'txtempLOYEE.Value = ""
    'txtstatus.Value = ""
    txtdept.Value = ""
    txtsec.Value = ""
    txtPCC.Value = ""
    cbotype.Value = ""
    txtdate.Value = ""
    txtexp.Value = ""
    txtdet.Value = ""
    CboEmployee.SetFocus

End Sub
```

Gambar 4.43 *Code* Tombol *Add New*.

### b. Event Tombol Save

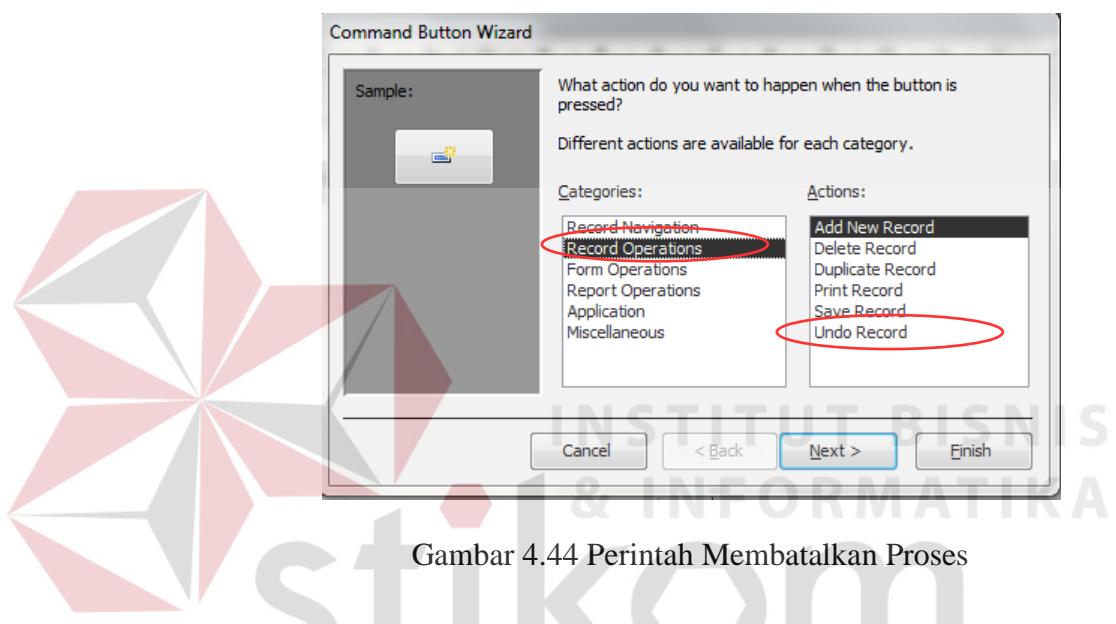
Tombol *Save* berfungsi untuk menyimpan *database* yang telah terisi pada *field* formulir sehingga dapat tersimpan kedalam tabel SP. Untuk langkah pembuatan *event* sama seperti langkah pembuatan *event add new*. Akan tetapi untuk *code* yang digunakan dapat dilihat dibawah ini.

```
Private Sub Save_Click()
    rssp.AddNew
    rssp.Fields("salnum") = CboEmployee.Value
    rssp.Fields("Department") = txtdePTValue
    rssp.Fields("Section/Working Area") = txtsec.Value
    rssp.Fields("Positions") = txtpcc.Value
    rssp.Fields("Type of SP") = cbotype.Value
    rssp.Fields("Date SP") = txtdate.Value
    rssp.Fields("Expired") = txtexp.Value
    rssp.Fields("detail") = txtdet.Value
    rssp.Update
    MsgBox "Save Succes"
End Sub
```

RSSP (*Record Source SP*) adalah nama tabel *database* yang akan dijadikan sebagai penyimpanan *database*. Kemudian *Fields* berfungsi sebagai tanda *fields* pada tabel mana yang akan digunakan. Kemudian pilih nama *field* setelah itu hubungkan dengan nama *field* yang ada pada formulir. MsgBox “Save Success” adalah perintah ketika *save* berhasil masuk maka akan muncul notifikasi bahwa *save* berhasil.

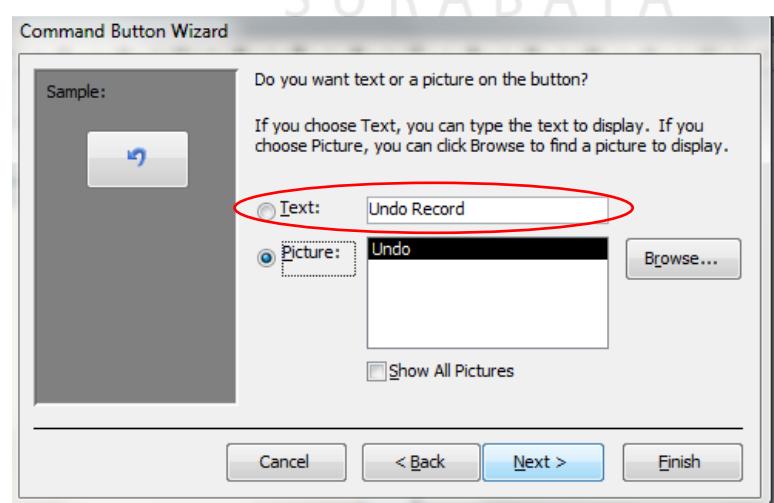
c. *Event Tombol Cancel*

Tombol *cancel* berfungsi untuk membatalkan pengisian formulir menjadi lembaran formulir kosong. Langkah pembuatannya seperti biasa menggunakan perintah yang telah disediakan oleh *Microsoft Access* yaitu menggunakan *command button wizard*. Pilih *Record Operation* pilih *undo record*. Lalu klik *next*.



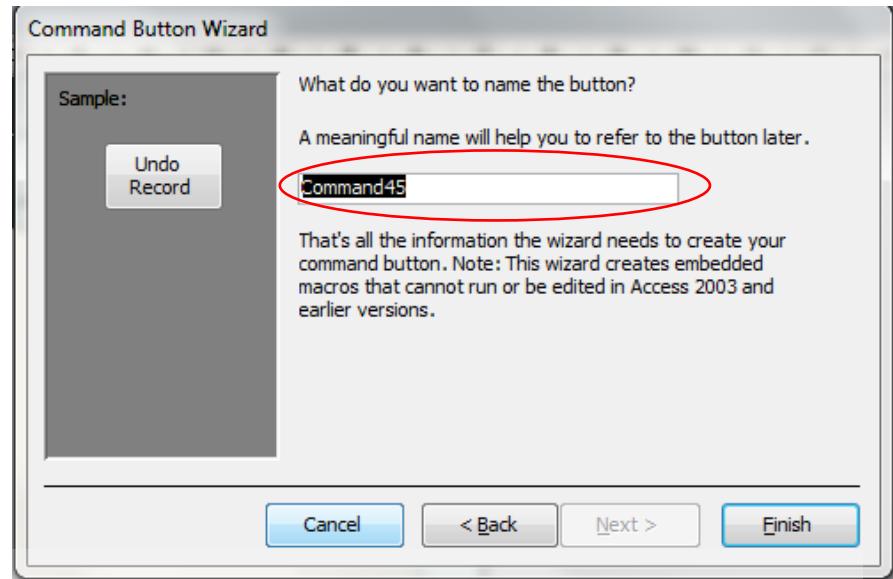
Gambar 4.44 Perintah Membatalkan Proses

Ubah nama *undo record* menjadi *Cancel*. Kemudian klik *next*.



Gambar 4.45 Ubah Nama *Text Buton*

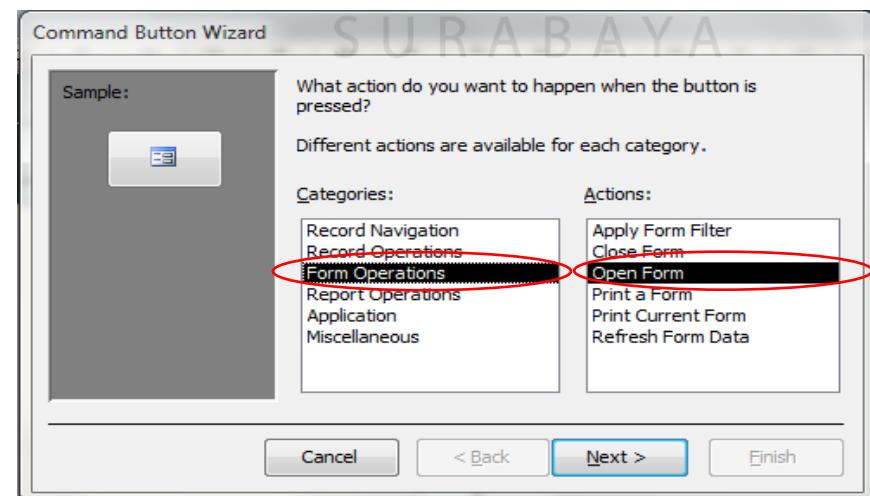
Ubah nama *command* menjadi *cancel record* lalu klik *finish*.



Gambar 4.46 Ubah Nama *Command Button*

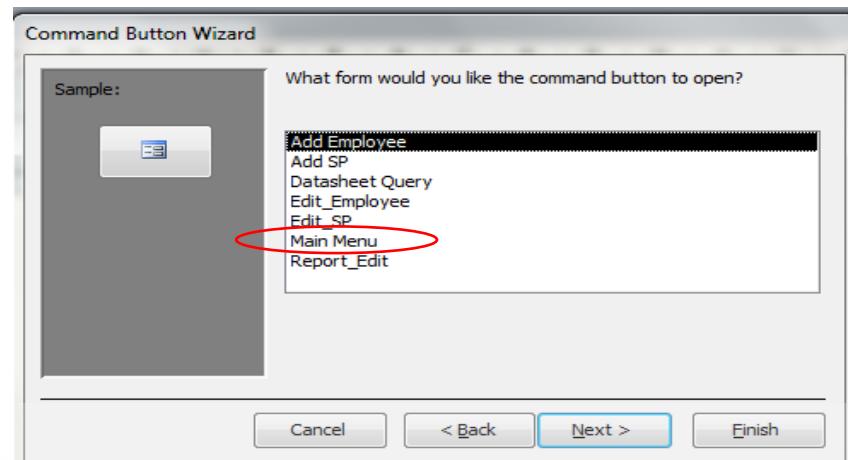
d. *Event Tombol Back*

Tombol *back* berfungsi untuk kembali ke menu utama. Dan langkah pembuatannya menggunakan *command button wizard*. Pilih *form operations* lalu pilih *open form*. Klik *next*.



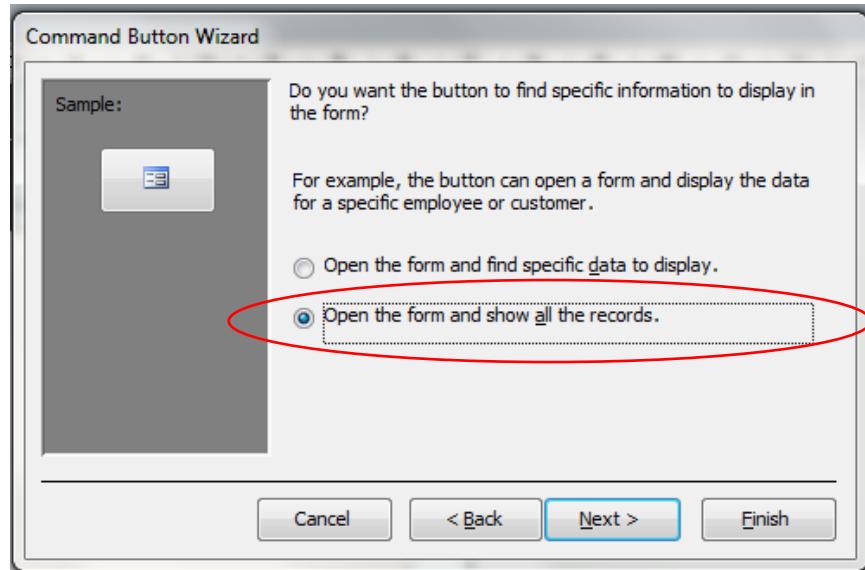
Gambar 4.47 Perintah Membuka *Form*

Pilih *form* yang akan dibuka klik *next*. Untuk *form* pada aplikasi pilih *form main menu*. *Form main menu* akan muncul ketika sudah membuat *switchboard*. Kemudian klik *next*.



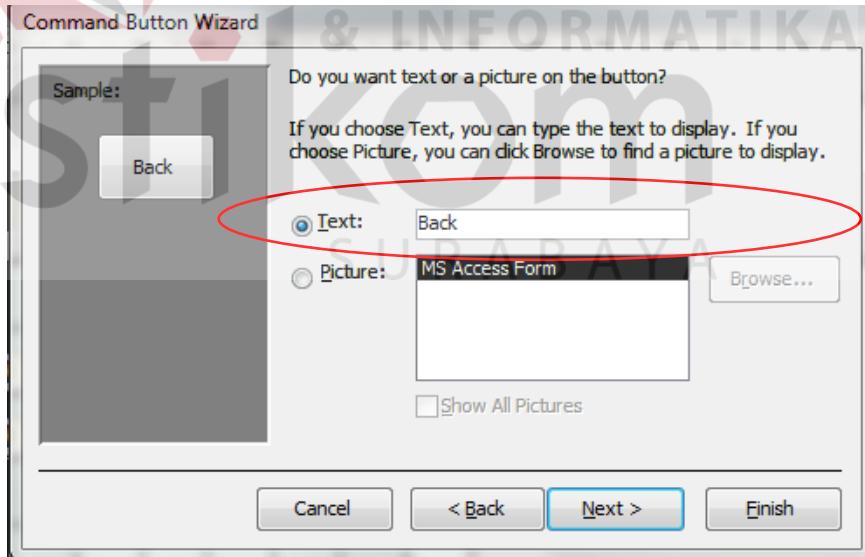
Gambar 4.48 Pilih *Form* yang Dibuka

Setelah memilih formulir maka akan muncul 2 pilihan. Yaitu “*Open the form and find specific data to display*”, pilihan pertama berfungsi membuka formulir dengan mencari data yang sama antara formulir yang sedang berjalan dengan formulir yang akan dibuka. Pilihan yang kedua “*Open the form and show all the record*”, pilihan yang kedua berfungsi menampilkan formulir dengan semua data tanpa *filter*. Lalu klik *next*.



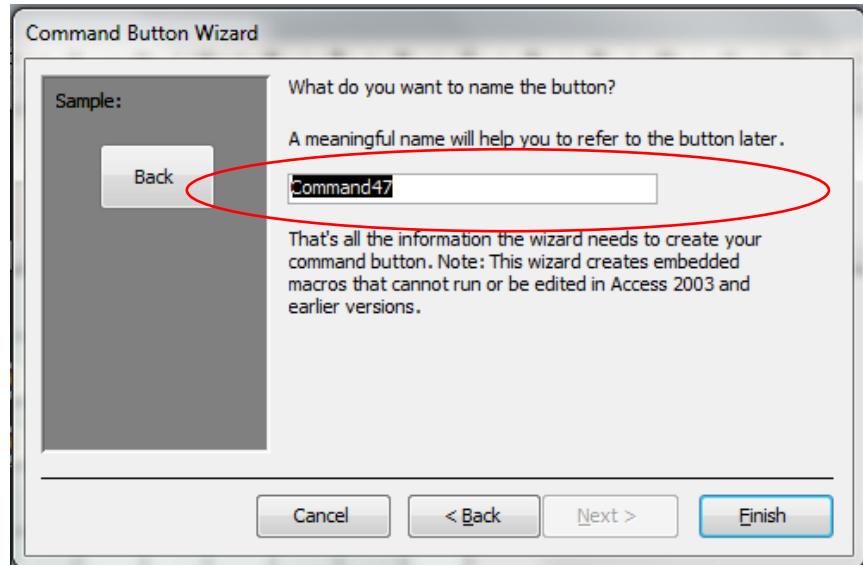
Gambar 4.49 Pilih *Open the Form and Show All the Record*

Kemudian beri nama untuk tombol *button* dengan nama *Back*. Lalu klik *next*.



Gambar 4.50 Beri Nama *Back*

Langkah terakhir untuk tombol *back* yaitu memberi nama tombol dari *command* diubah menjadi *Back*. Lalu klik *finish*.

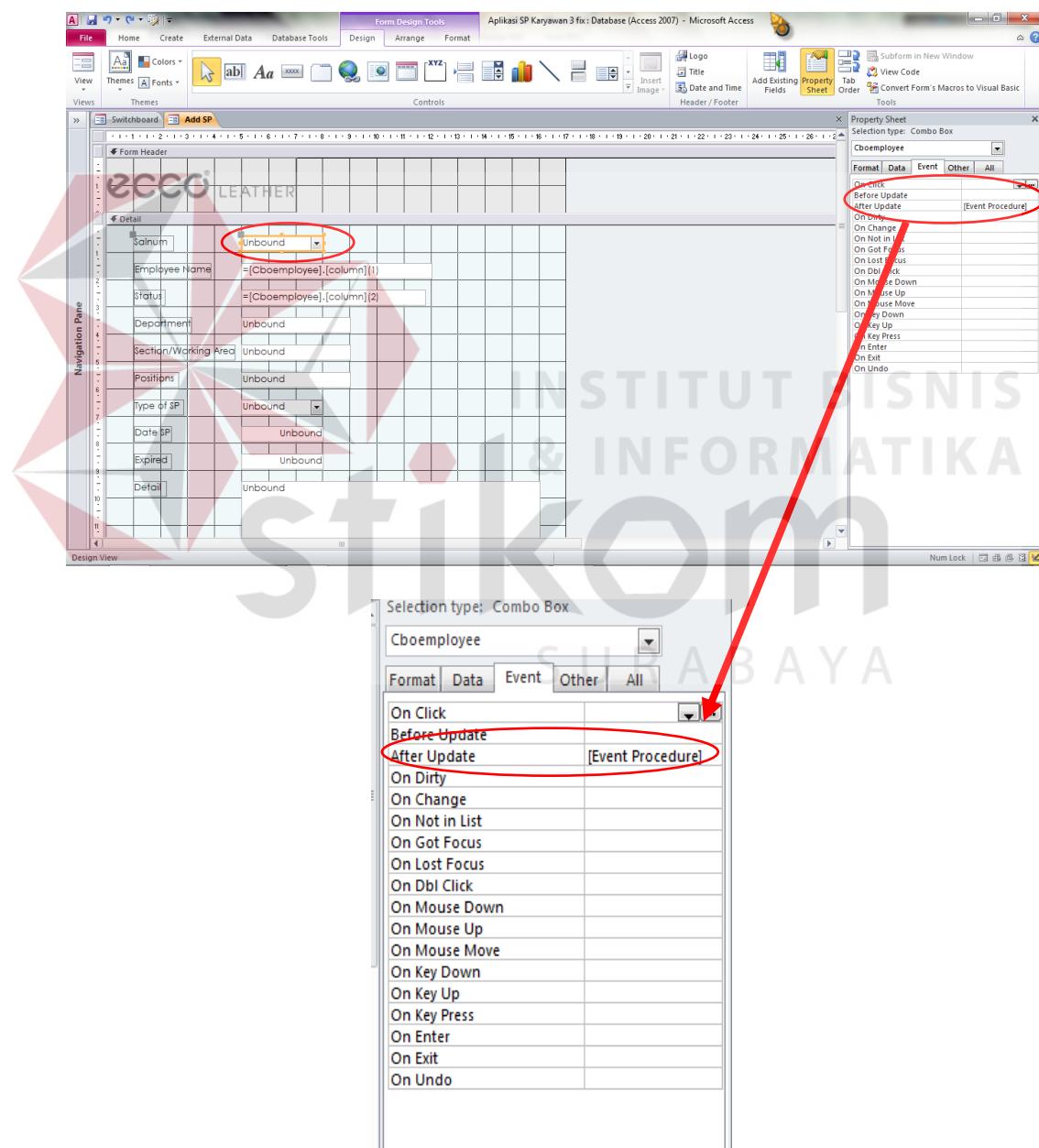


Gambar 4.51 Beri Nama *Command Button*

Setelah semua langkah berhasil dilakukan maka akan membentuk tampilan *Form Design* dibawah ini.

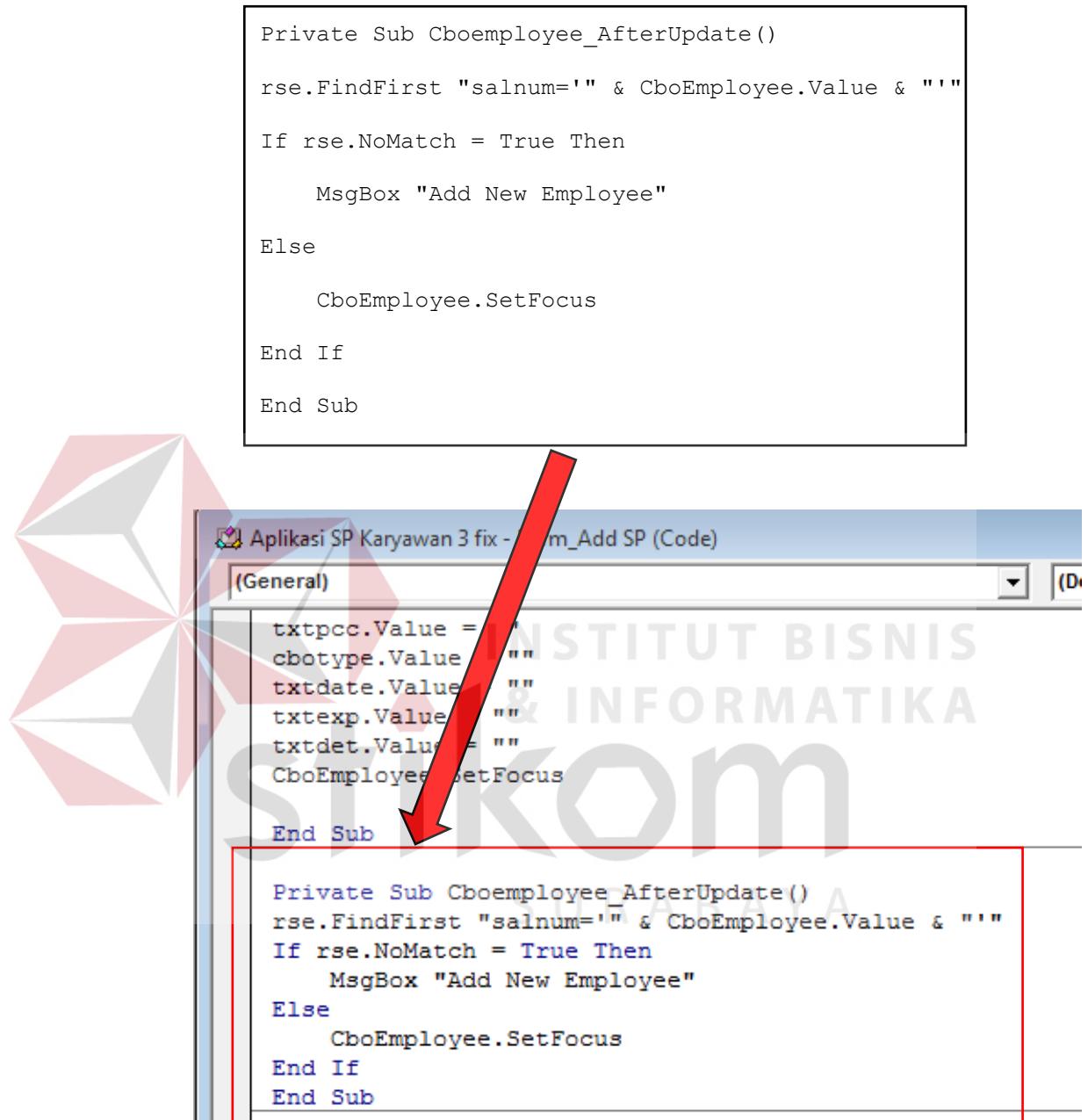
Gambar 4.52 Tampilan *Form Add SP*

8. Langkah terakhir Pembuatan formulir *add new SP* sebelum digunakan adalah memasukkan *code IF* yang berfungsi jika pada saat mengisi SP karyawan dan salnum karyawan belum ada atau belum terdaftar maka akan muncul pemberitahuan. Sehingga tidak terjadi *error* pada *database* yang akan dilihat *report* karyawannya. Tambahkan *event after Update* pada *cboemployee*.



Gambar 4.53 *Combo Box Salnum*

Kemudian masukkan *code* seperti dibawah ini.



```

Private Sub Cboemployee_AfterUpdate()

rse.FindFirst "salnum=''' & CboEmployee.Value & '''"

If rse.NoMatch = True Then

    MsgBox "Add New Employee"

Else

    CboEmployee.SetFocus

End If

End Sub

```

```

Private Sub Cboemployee_AfterUpdate()
rse.FindFirst "salnum=''' & CboEmployee.Value & '''"
If rse.NoMatch = True Then
    MsgBox "Add New Employee"
Else
    CboEmployee.SetFocus
End If
End Sub

```

Gambar 4.54 *Code IF*

*Code* diatas dimasukkan pada *event after update* pada *field Cboemployee*. Kemudian pada *database RSE* *field* *salnum* harus sama dengan yang diisikan pada *field* *CboEmployee*. Jika tidak terdapat

kesamaan maka aplikasi akan menampilkan pesan untuk menambahkan karyawan terlebih dahulu. Tapi jika data yang diisi sama dengan data pada *field* salnum maka *field* *CboEmployee* menjadi fokus untuk diisi. Jangan lupa setiap menggunakan *code* pastikan diakhiri dengan perintah *End*. Supaya *code* dapat berjalan sesuai perintah dan tidak *error*. Itulah langkah-langkah pembuatan formulir *add new SP*.

#### D. Create Form Add Employee

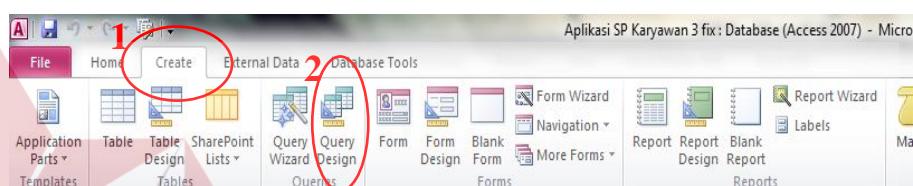
Langkah pembuatan *form add employee* hampir sama dengan pembuatan *form add SP* untuk pembuatan awal form menggunakan *blank form*. Kemudian masukkan *field* yang terdapat pada tabel *employee*. sehingga tampilan form akan sama seperti dibawah ini.

Gambar 4.55 Tampilan Form Design Add Employee

### E. Create Edit Data SP

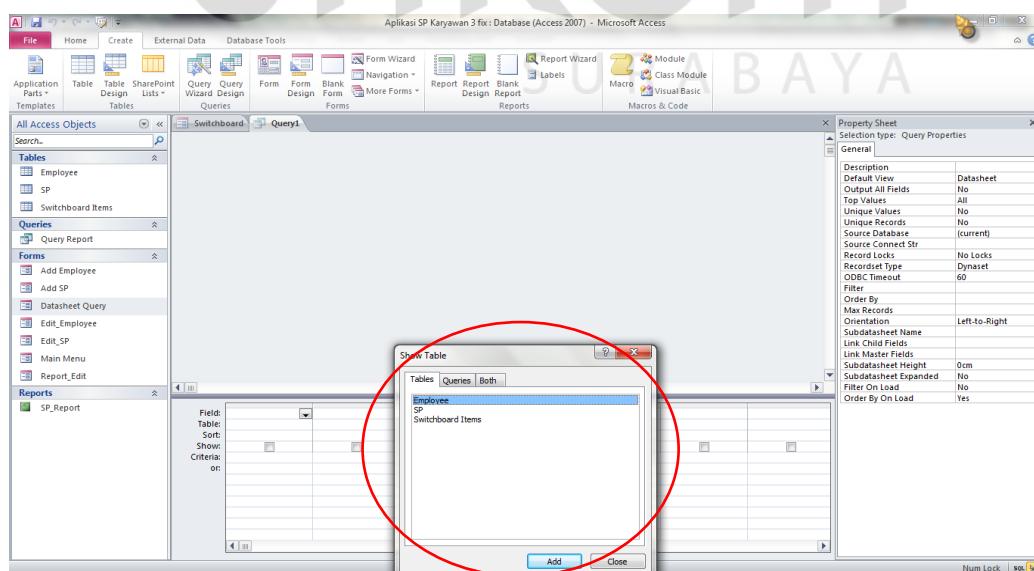
Setelah formulir *add SP* dan *Add Employee* tercipta maka langkah selanjutnya membuat *edit SP*. *Edit SP* dibuat melalui *form* yang datanya diambil dari data *query*. Langkah pembuatannya dapat dilihat dibawah ini.

a. Langkah pertama untuk pembuatan *edit list* pada aplikasi karyawan yaitu dengan membuat *query* terlebih dahulu. *Query* berfungsi untuk mengambil *database* dari banyak tabel. Pembuatan *query* dapat dilakukan dengan memilih *toolbar* lalu pilih *query design*.



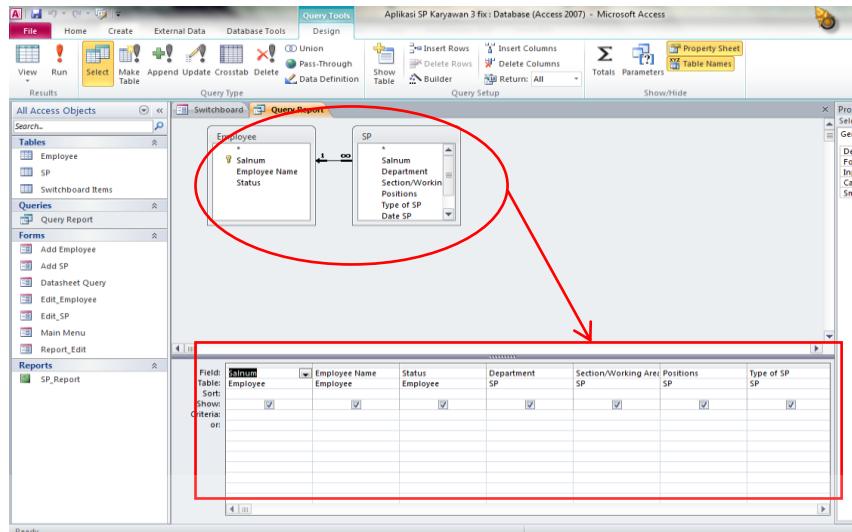
Gambar 4.56 Create Query Design

b. Kemudian pada lembar kerja *query* tampilkan tabel *employee* dan tabel SP. Klik *add*.



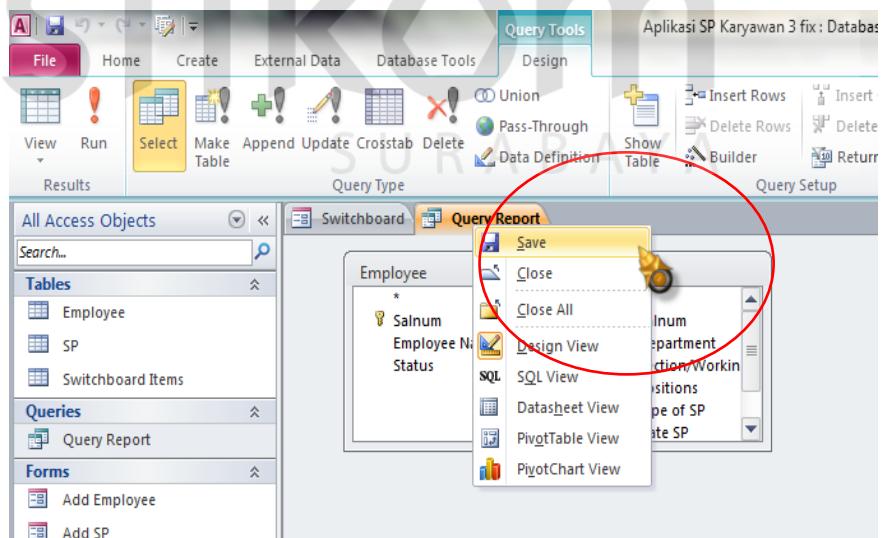
Gambar 4.57 Add Table Query

- c. Setelah tampil tabel SP dan *employee* maka langkah selanjutnya yaitu memasukkan *field* ke dalam tabel *query*.



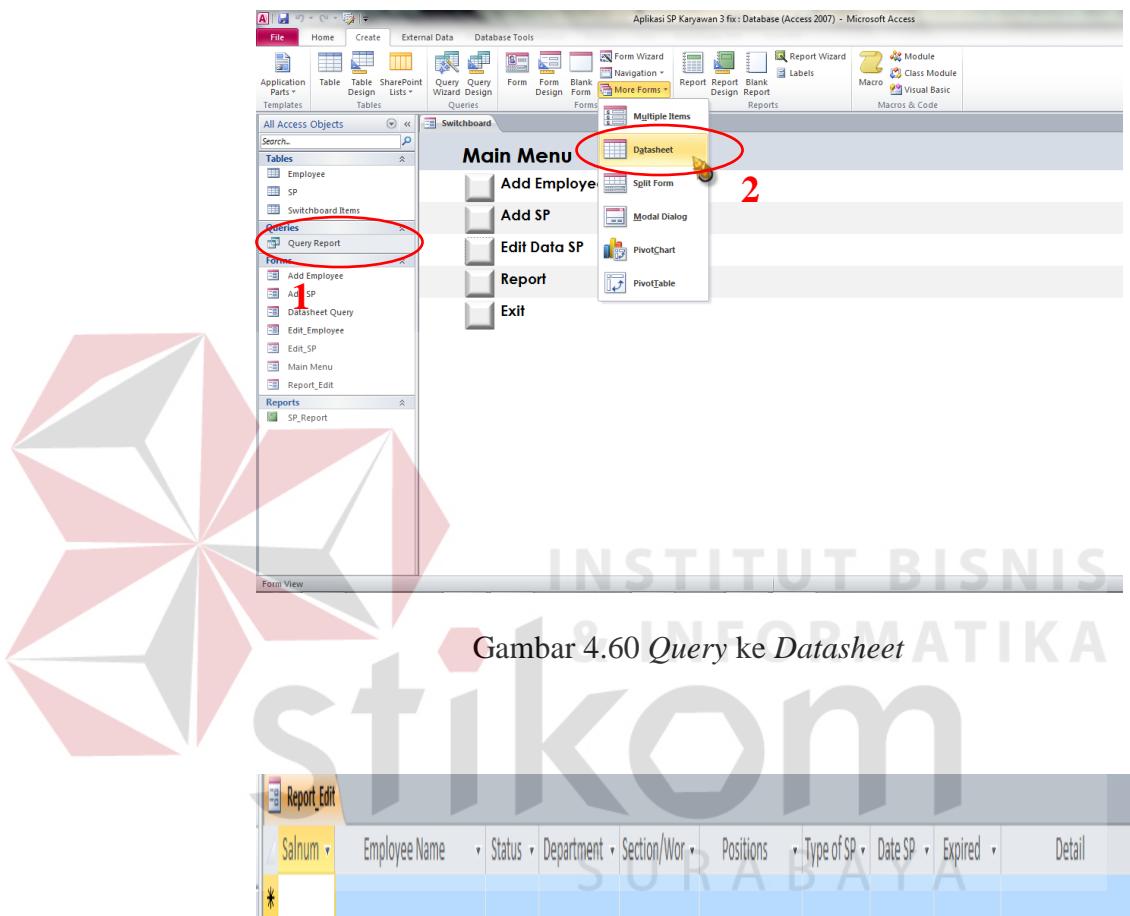
Gambar 4.58 Masukkan *Field* dalam *Query*

- d. Setelah memasukkan *field* ke dalam tabel *query*, save data dengan nama *query\_report*.



Gambar 4.59 Save *Query*

- e. Setelah *query* tercipta jadikan *query* menjadi *datasheet* dengan menggunakan *datasheet form*. Pastikan pilih dahulu *query* kemudian klik *datasheet*. Setelah jadi *datasheet* simpan dengan nama *Report\_edit*.



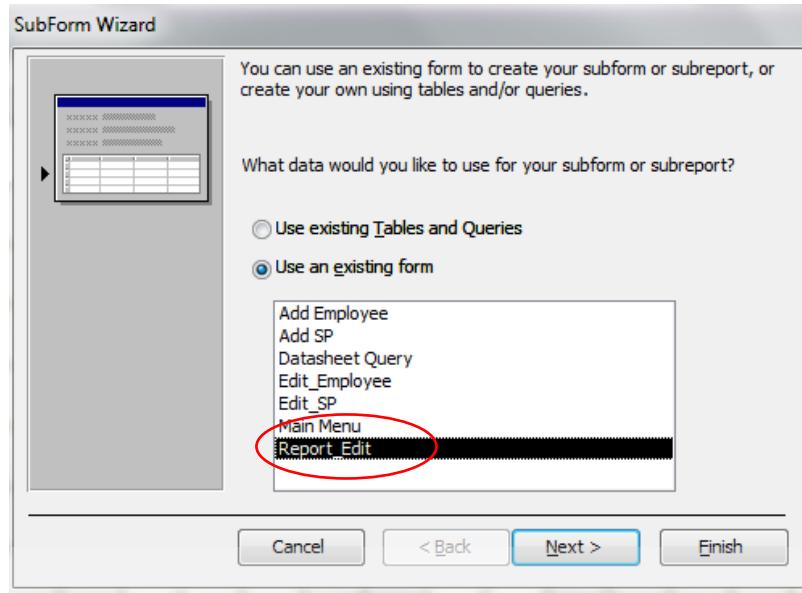
Gambar 4.60 *Query ke Datasheet*

- f. *Datasheet* telah tercipta maka dapat dijadikan sebagai *subform/subreport* pada *edit list*. Masukkan *subform/subreport* ke dalam *blank form*.



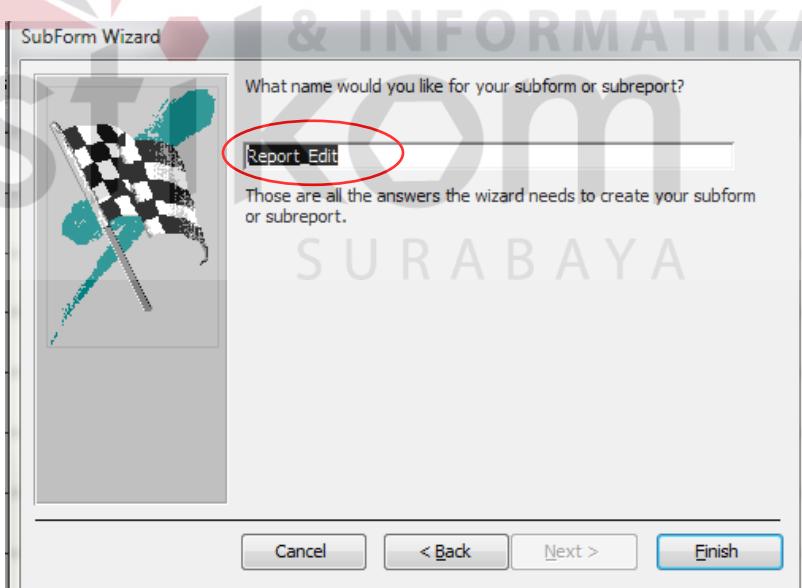
Gambar 4.62 Masukkan *Subform/subreport*

g. Lalu pilih *form report\_edit*, kemudian klik *next*.



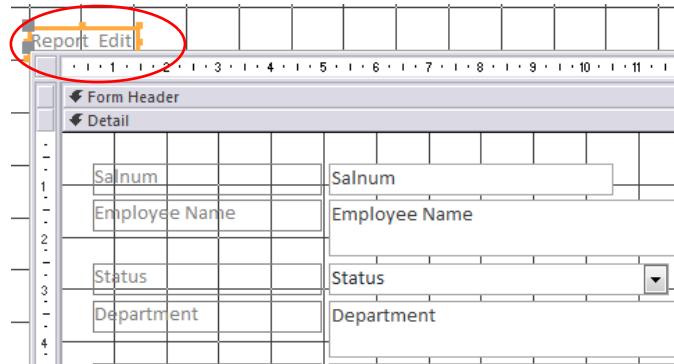
Gambar 4.63 Pilih Form Report>Edit

h. Beri nama *subform* dengan *Report\_edit* kemudian klik *finish*.



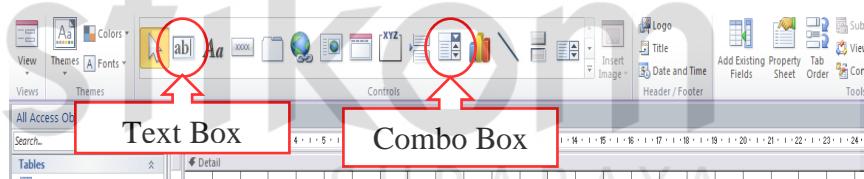
Gambar 4.64 Save Nama Report>Edit

- i. Nama untuk *subform/subreport* dapat dihapus jika tidak perlu. Klik nama *subform/subedit* lalu *delete*.

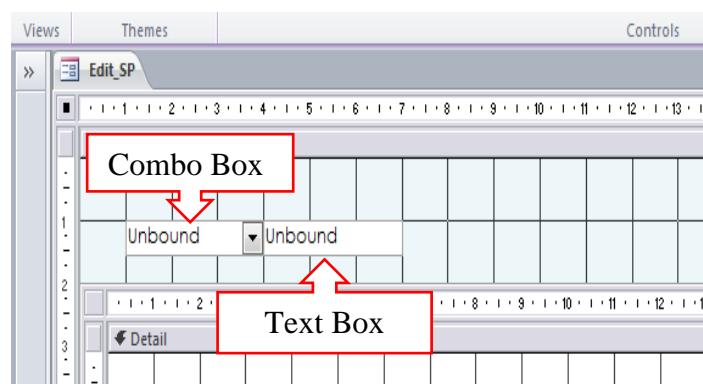


Gambar 4.65 Hapus Nama Subform/Subreport

- j. Jika *subform/subreport* telah tercipta maka langkah selanjutnya yaitu membuat *text box* dan *combo box* sebagai *filter* nya. Letakan *combo box* dan *text box* diatas *subform/subreport*. Beri nama *combo box* dengan *Cbosalnum* dan *text box* dengan *Txtsalnum*.

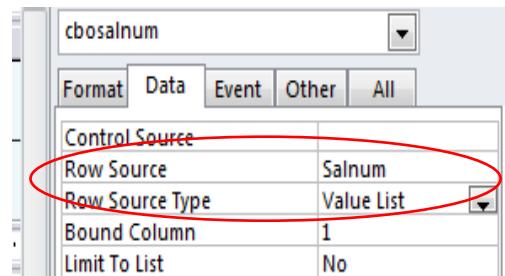


Gambar 4.66 Text Box dan Combo Box



Gambar 4.67 Text Box dan Combo Box

- k. Setelah *combo box* dan *text box* dibuat maka beri *control* untuk kedua *button*. Untuk *combo box* seperti pada *report list IT Issue Log* beri *control* pada pada *properties* tab data. *Row source* isikan dengan “*salnum*” dan *row source type* ubah menjadi *value/list*.



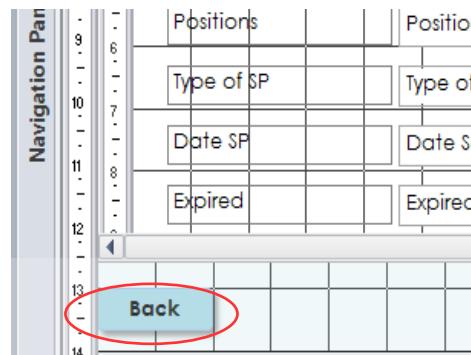
Gambar 4.68 Setting Row Source dan Row Type

```

Private Sub txtsalnum_Change()
    Dim STR_CBosalnum As String
    If Nz(Me.cbosalnum, "") = "" Then
        MsgBox "Entry Key", vbCritical, "Warning"
    Else
        STR_CBosalnum = "[" & Me.cbosalnum & "]"
        With Me.Datasheet_Query.Form
            If Me.txtsalnum.Text = "" Then
                .Filter = ""
                .FilterOn = False
            Else
                .Filter = STR_CBosalnum & " LIKE '*' &
                Me.txtsalnum.Text & "*"
                .FilterOn = True
            End If
        End With
    End If
End Sub

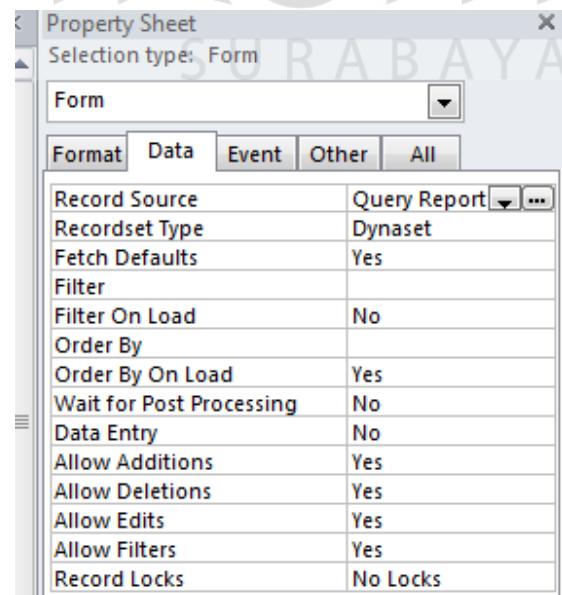
```

1. Langkah terakhir yaitu memberi tombol *button back* yang berfungsi untuk kembali ke menu utama. Untuk langkah pembuatan tombol *button* sama seperti tombol *button back* pada *form add SP*.

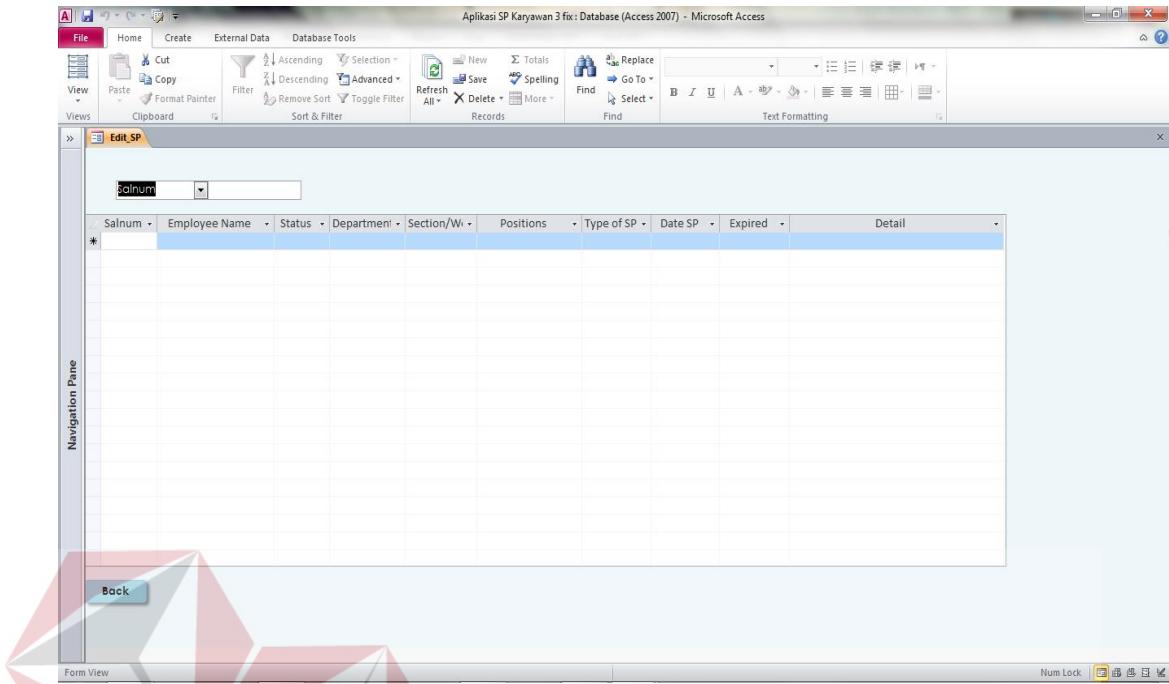


Gambar 4.69 Tombol Back

Untuk menampilkan *edit list* supaya dapat diedit maka *setting subform/subreport* pada *properties* data seperti ini. Sehingga *edit list* dapat ditampilkan.



Gambar 4.70 Setting Edit List



Gambar 4.71 Hasil Jadi Form Edit List

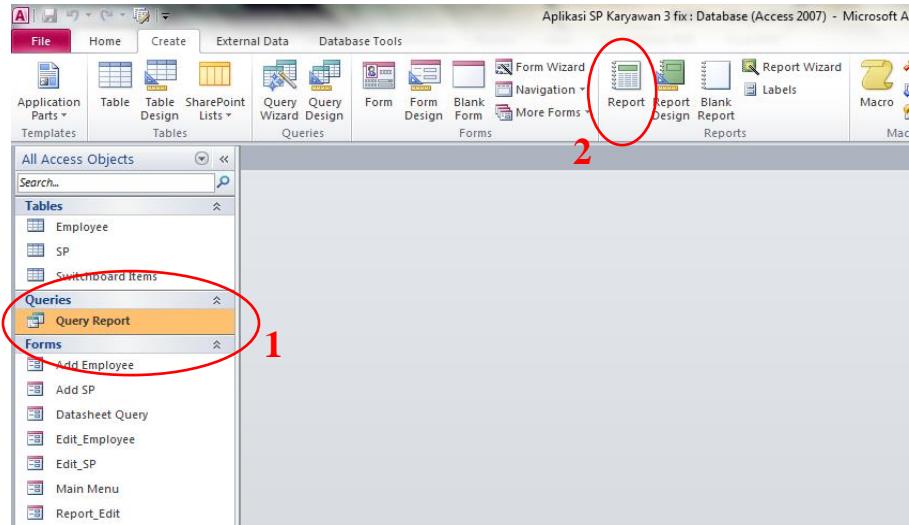
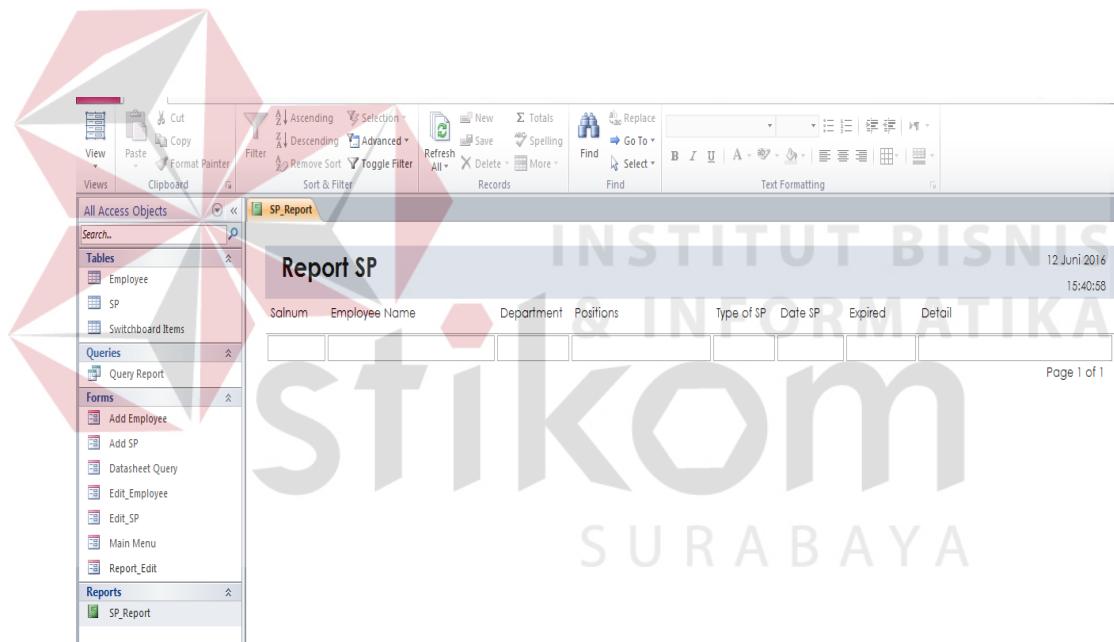
#### F. Create Report

Setelah *edit list* tercipta maka langkah selanjutnya yaitu pembuatan *report*.

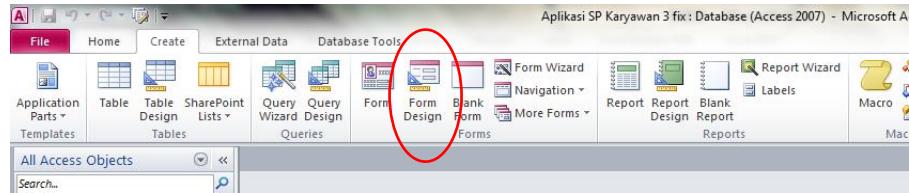
Report ini berfungsi untuk melihat status karyawan hanya dengan melihat dari salnumnya.

1. Pembuatan *report* yaitu menggunakan *report* pada *toolbar create*.

Pastikan sebelum klik *report query* yang sudah dibuat berwarna atau klik kiri terlebih dahulu pada *query*. Kemudian simpan dengan nama *SP\_report*.

Gambar 4.72 Pembuatan *report*Gambar 4.73 Hasil Jadi *Report SP*

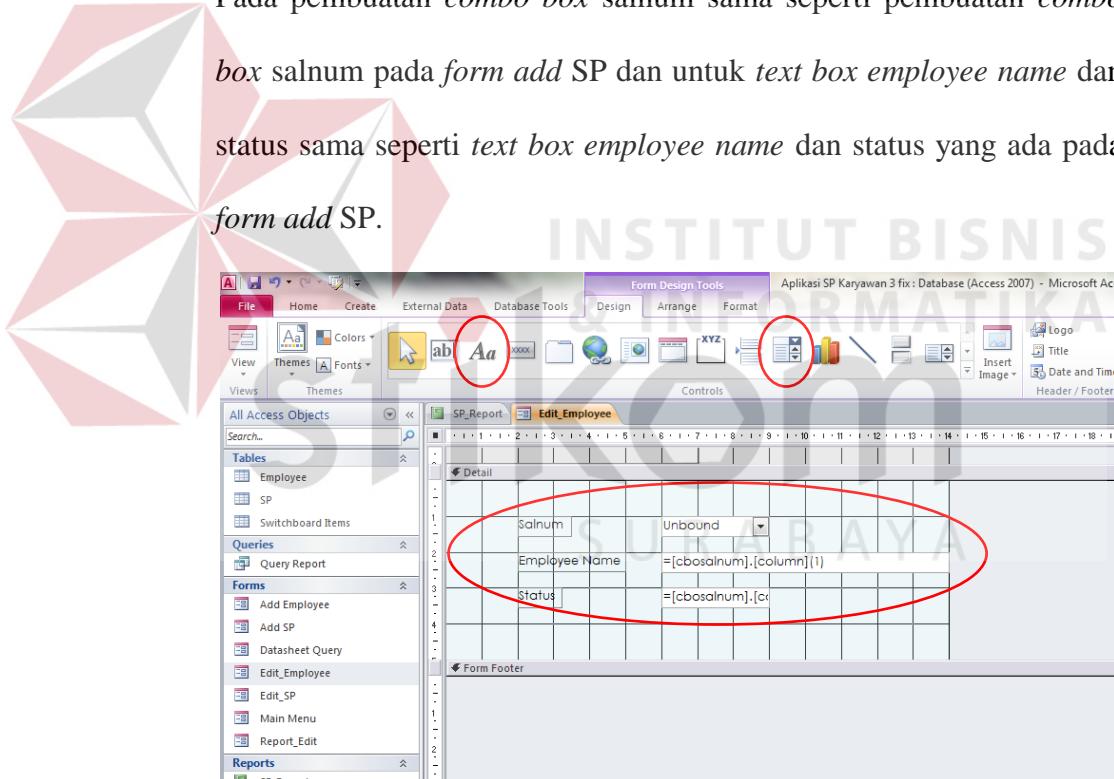
2. Setelah *report* terbuat maka langkah selanjutnya membuat halaman yang bertujuan untuk membuka *report* tersebut. Untuk pembuatan halaman menggunakan *form design* yang ada pada *toolbar create*.



Gambar 4.74 Pembuatan *Form Report*

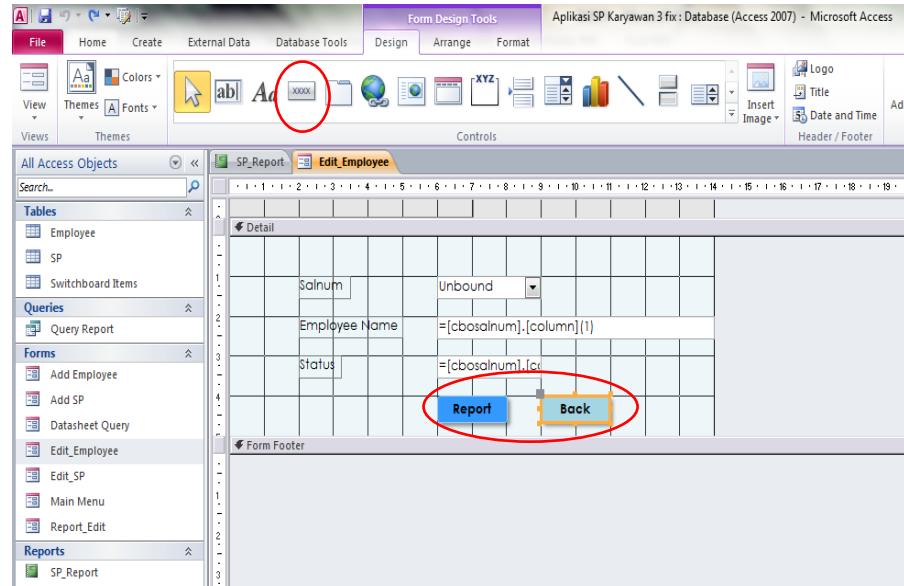
Pada lembar kerja *form design* menggunakan *combo box* dan *text box* sebagai langkah awal sebelum membuka *report*. *Combo box* dan *text box* ini berfungsi untuk mem-filter data yang terdapat pada *report*.

Pada pembuatan *combo box* salnum sama seperti pembuatan *combo box* salnum pada *form add SP* dan untuk *text box employee name* dan status sama seperti *text box employee name* dan status yang ada pada *form add SP*.



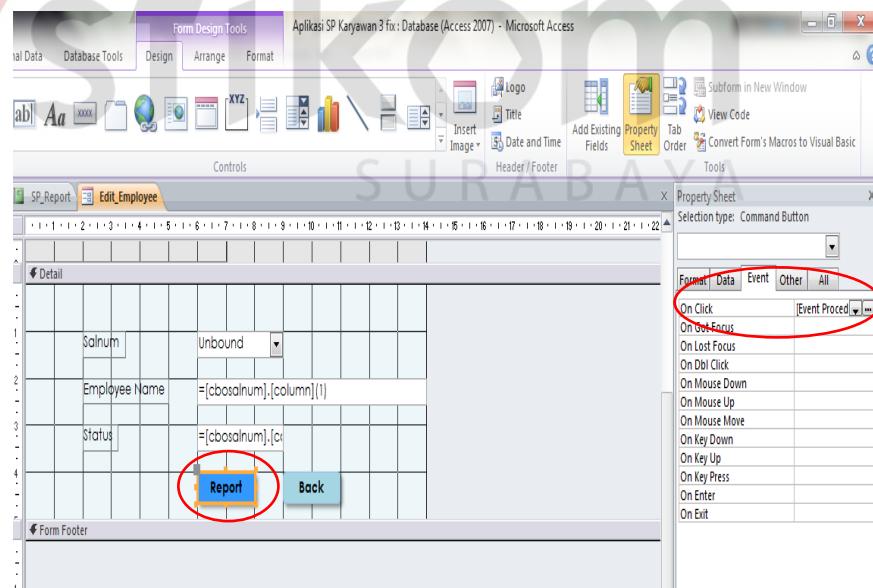
Gambar 4.75 *Text Box* dan *Combo Box*

Setelah itu buat tombol *button*, tombol pertama digunakan untuk membuka *report karyawan* dan tombol kedua berfungsi untuk kembali ke menu utama atau tombol *back*.

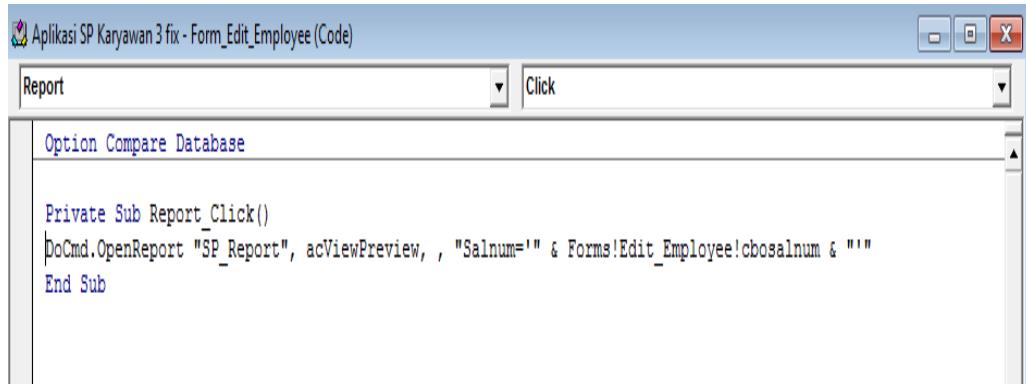


Gambar 4.76 Tombol *Button Report* dan *Back*

Pada tombol *report* beri *event* supaya dapat membuka *report* dari salnum yang ada pada *combo box*. Pada *event* “on klik” masukkan *code*.



Gambar 4.77 *Event* pada Tombol *Report*



A screenshot of a Microsoft Access VBA editor window. The title bar says 'Aplikasi SP Karyawan 3 fix - Form\_Edit\_Employee (Code)'. The code in the editor is:

```

Private Sub Report_Click()
DoCmd.OpenReport "SP_Report", acViewPreview, , "Salnum=' " & Forms!Edit_Employee!cbosalnum & "'"
End Sub

```

Gambar 4.78 *Code Filter Report*

Jika *code* sudah dimasukkan maka formulir akan berfungsi pembuka

*report* karyawan dengan *filter* yang terdapat pada *salnum*.



A screenshot of an 'Edit Employee' form. The form has three text input fields and two buttons at the bottom:

- Salnum: 1800027
- Employee Name: RUDIYATI
- Status: Active

At the bottom are two buttons: 'Report' (highlighted in blue) and 'Back'.

Gambar 4.79 Hasil *Form Report*

#### G. *Create Switchboard*

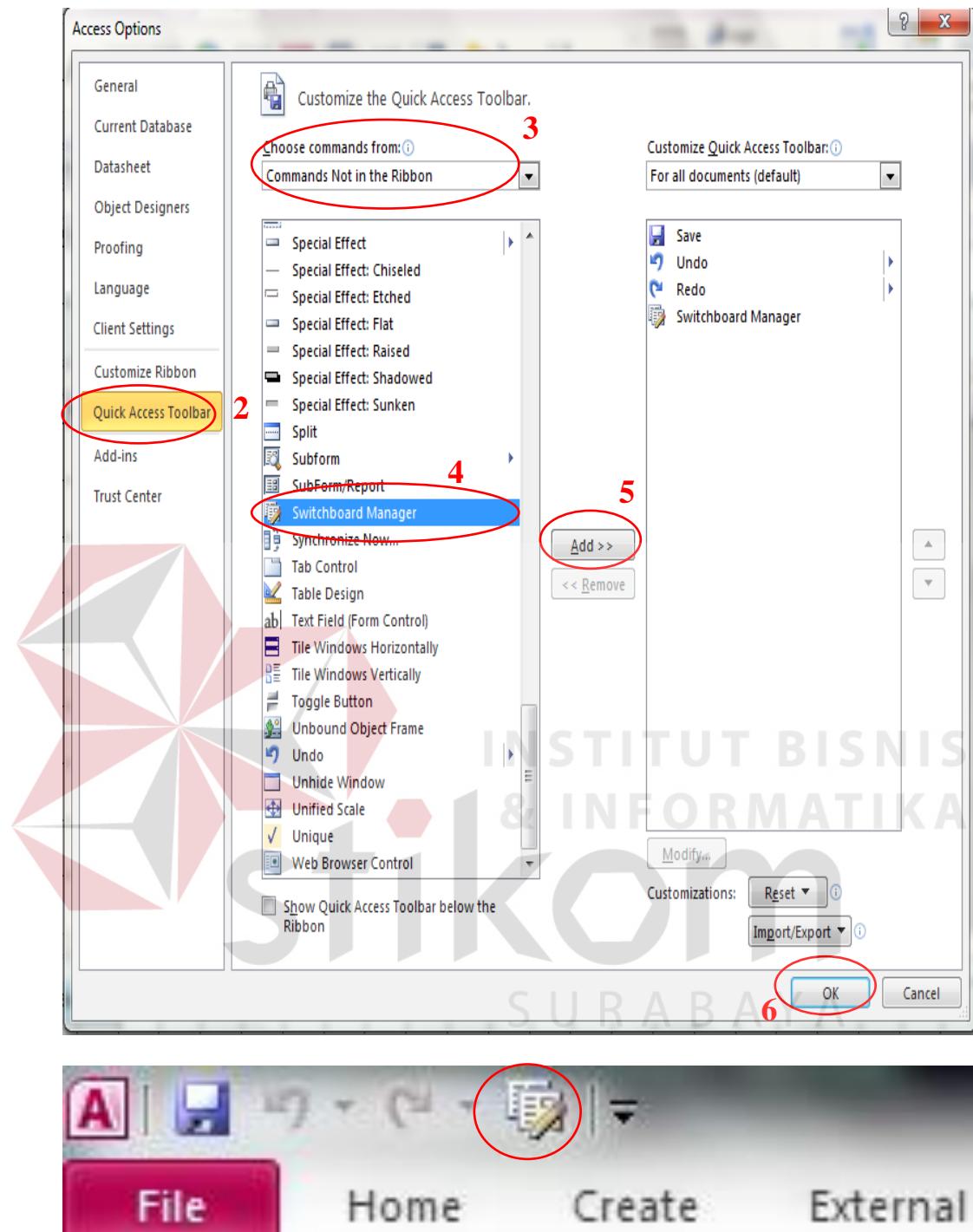
Langkah terakhir pembuatan aplikasi karyawan yaitu pembuatan *switchboard*. Pembuatan *Switchboard* dapat dilihat dibawah ini.

1. Klik icon *Switchboard* pada atas *microsoft access*. Jika belum muncul dapat ditampilkan melalui *options* pada tab *file*.



Gambar 4.80 *Options*

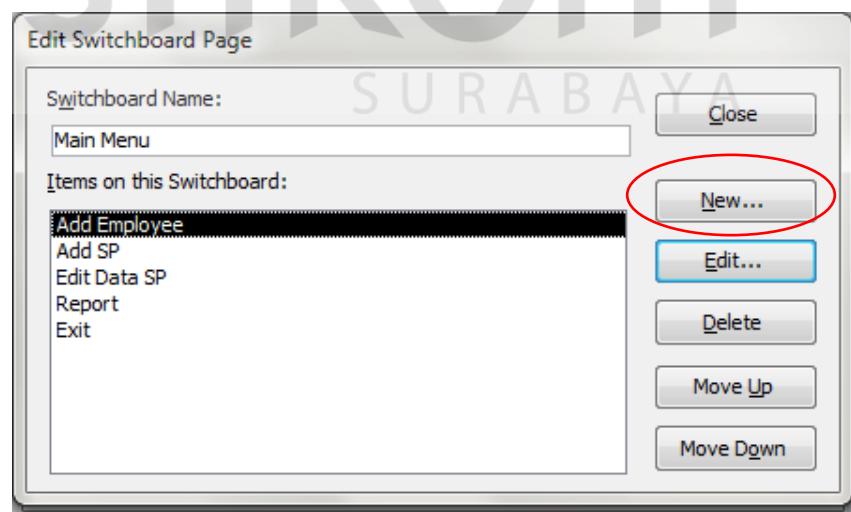
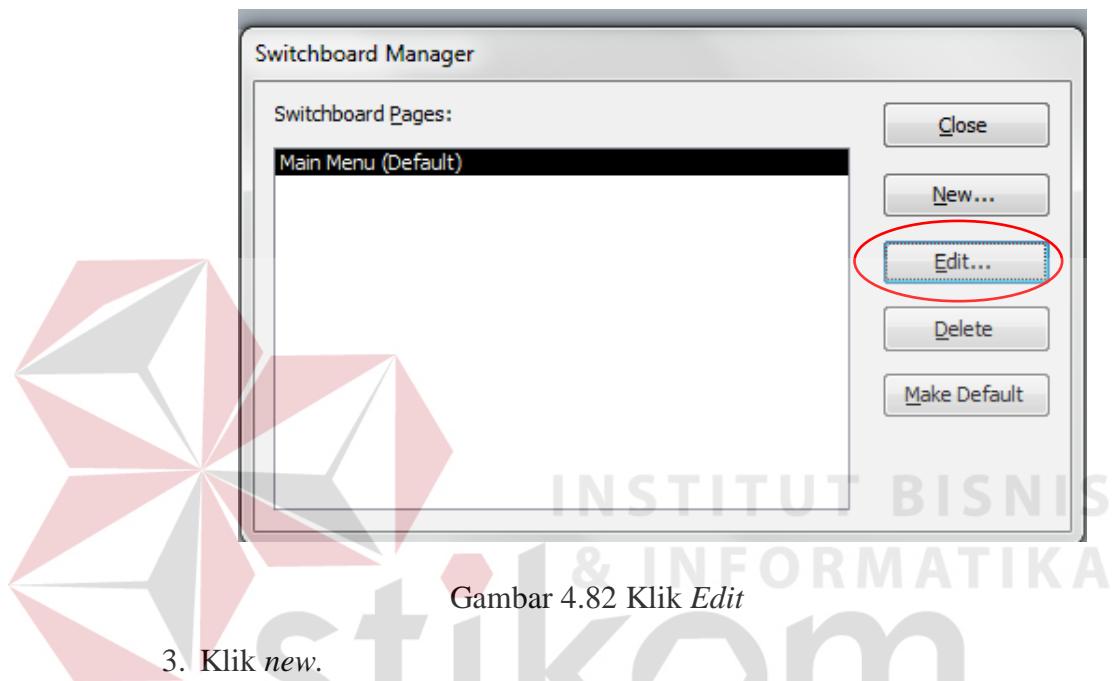
2. Pilih *Quick Acces Toolbar*
3. Ubah *choose commands form* menjadi *command not in the ribbon*
4. Pilih *Switchboard Manager*
5. Klik *add*
6. Klik *OK*



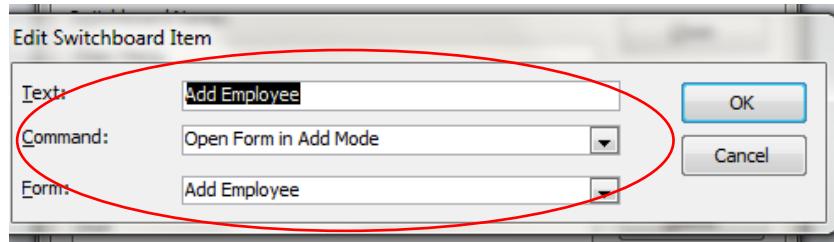
Gambar 4.81 Add Icon Switchboard

Setelah *icon switchboard* muncul maka langkah selanjutnya membuat *switchboard* dapat dilakukan. Untuk langkah pembuatan dapat dilihat dibawah ini.

1. Klik *icon switchboard manager*.
2. Pada jendela *switchboard* klik *edit*.

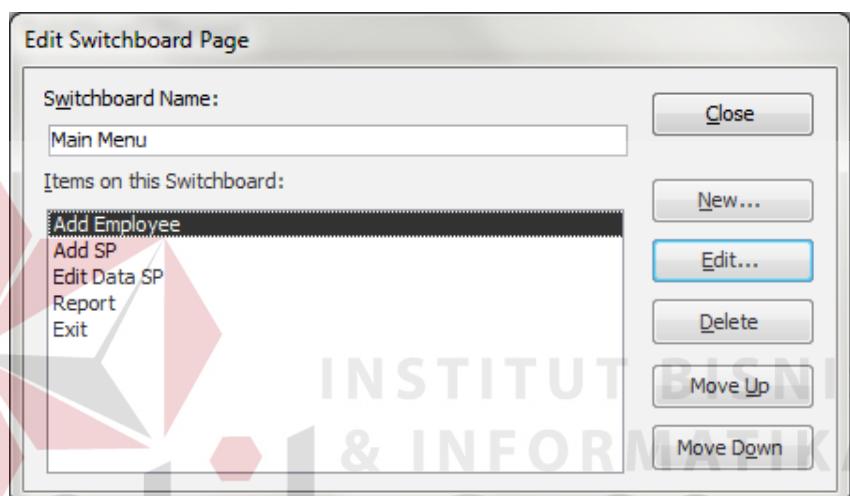


4. Masukkan nama *switchboard* dan berikan perintah *switchboard*.



Gambar 4.84 Berikan Nama dan perintah

5. Klik OK

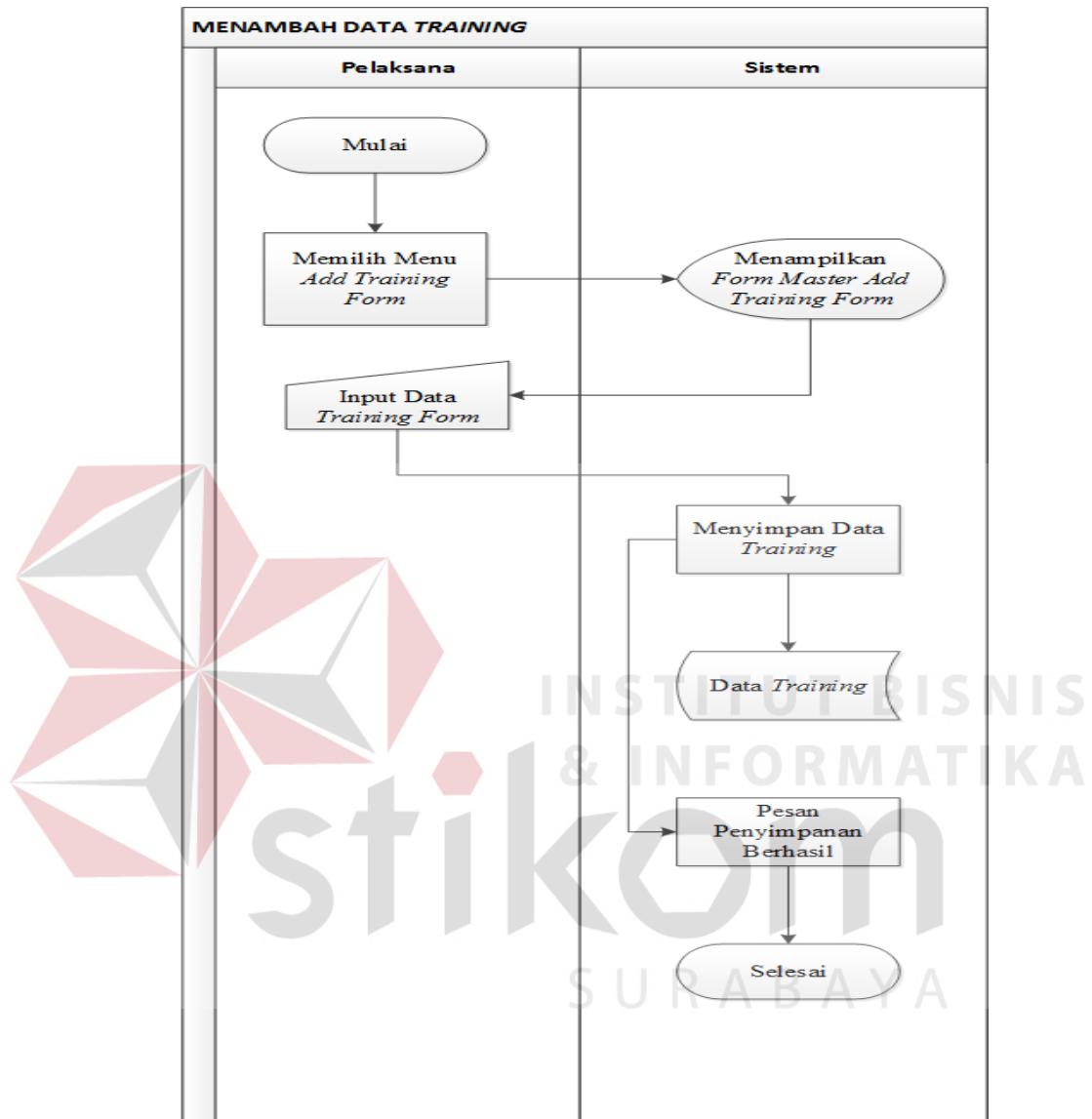


Gambar 4.85 Tampilan *Switchboard*

#### 4.7 Aplikasi *Training*

Aplikasi *Training* merupakan aplikasi yang digunakan oleh bagian HRD untuk menyimpan data karyawan yang sudah melakukan *training* sehingga setiap karyawan yang telah melakukan *Training* dapat dilihat *record*-nya melalui aplikasi *Training*. Aplikasi ini tercipta melalui *software microsoft access*. Cara pembuatannya hampir sama dengan pembuatan aplikasi SP. Untuk langkah pembuatannya dapat dilihat dibawah ini.

#### 4.7.1 System Flow Aplikasi Training



Gambar 4.86 System Flow Menambah Data Training

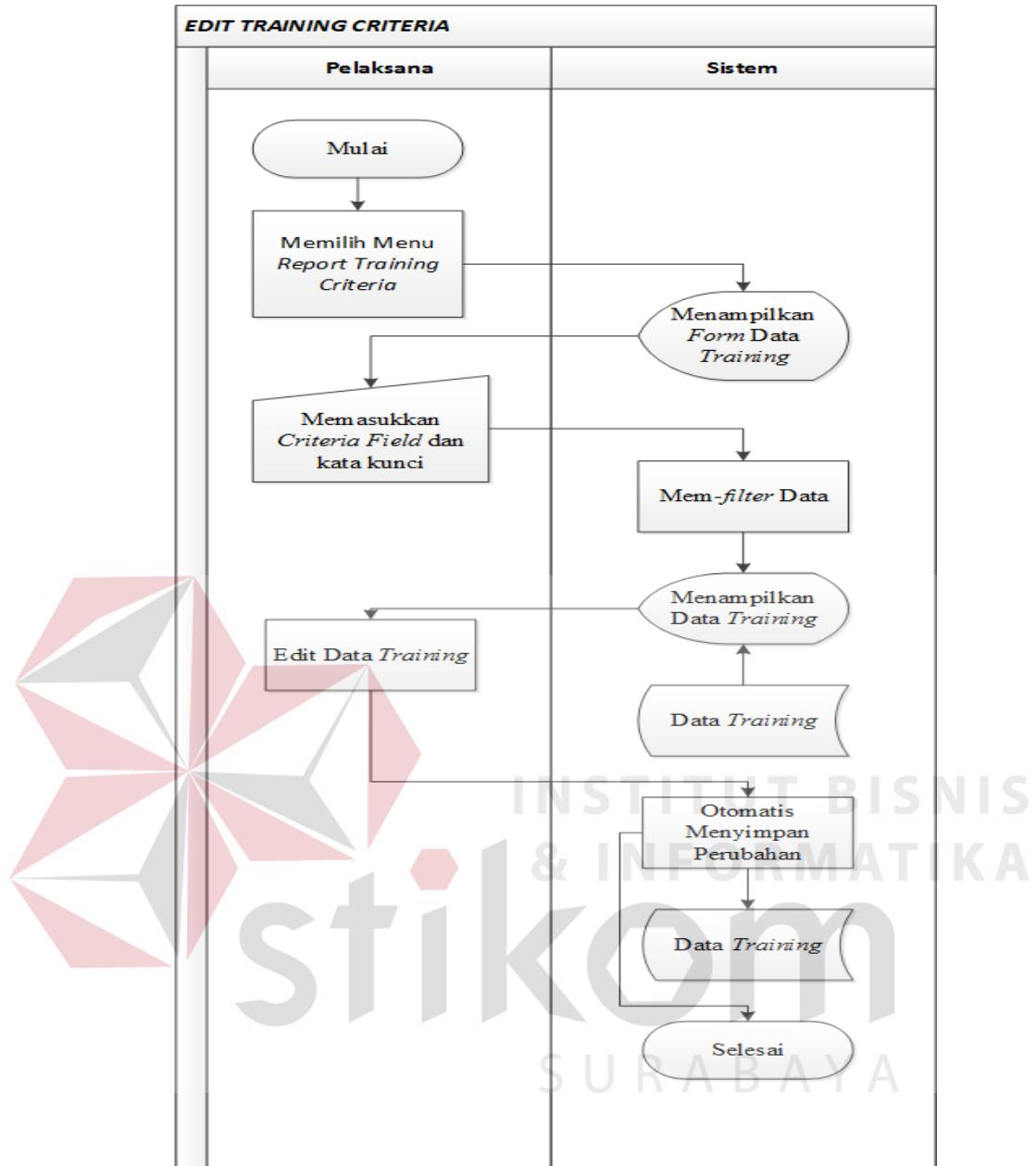
*Objective* : untuk menambah data *training*

*Input* : identitas karyawan dan nama *training*

*Proses* : menyimpan data *training*

*Output* : data pada tabel data pegawai *training* bertambah

*Actor* : pelaksana



Gambar 4.87 System Flow Report Training Criteria

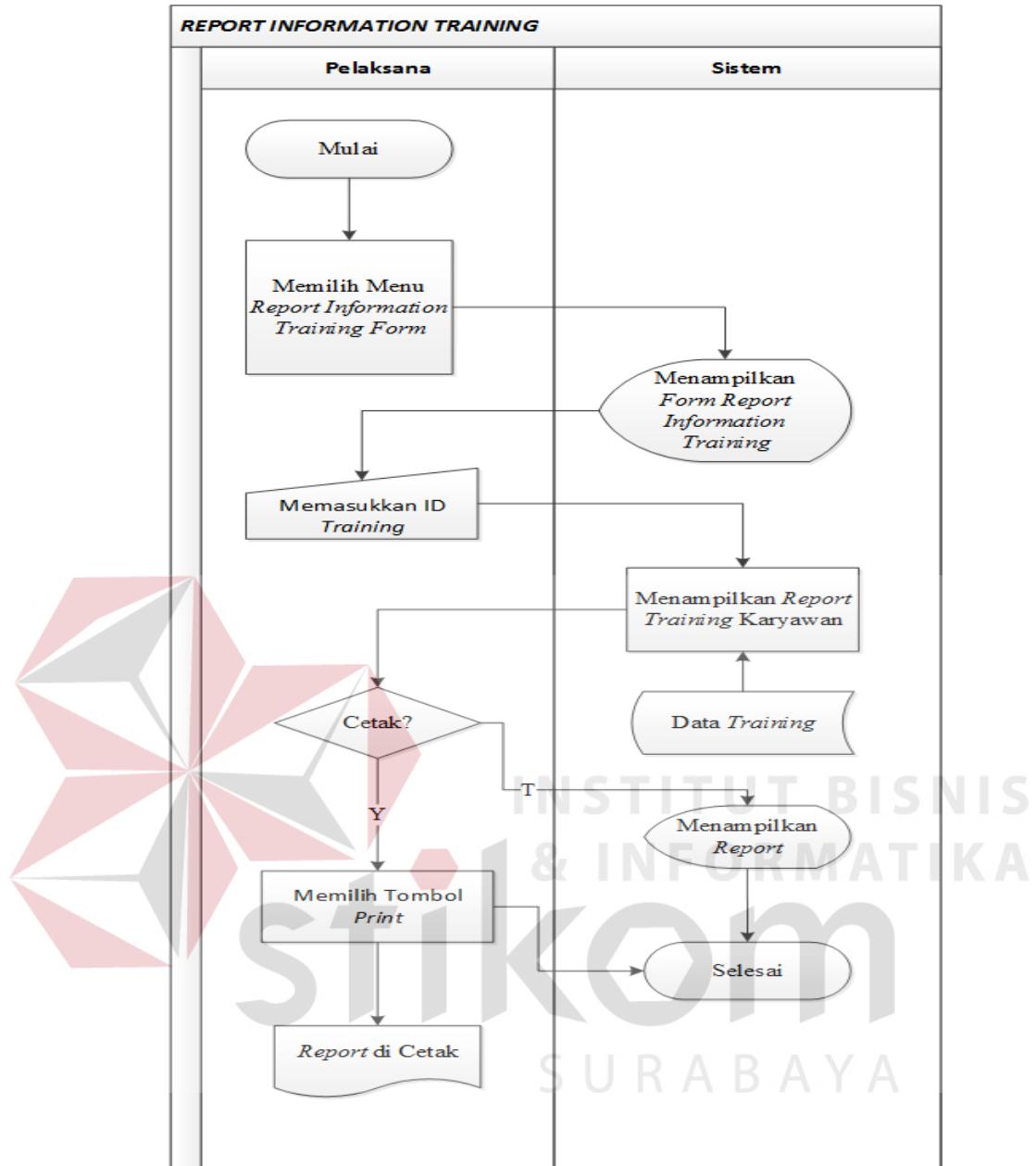
*Objective* : untuk *report* dan *edit data training*

*Input* : *criteria field*

*Proses* : *report* dan *edit data training*

*Output* : data pada tabel data pegawai *training* berubah jika di *edit*

*Actor* : pelaksana



Gambar 4.88 System Flow Report Information Training

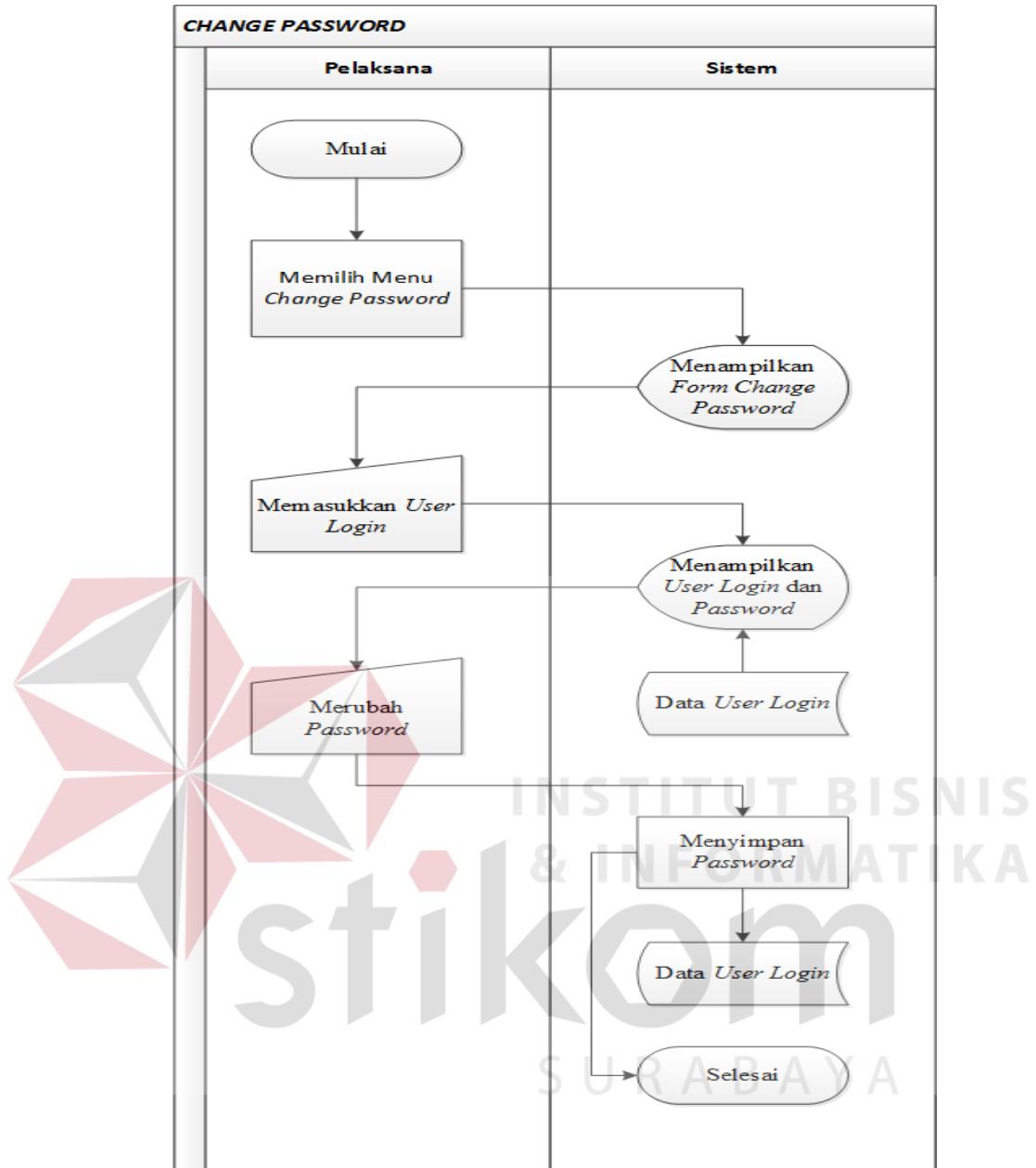
*Objective* : untuk report data training

*Input* : ID training

*Proses* : reportt data training

*Output* : report menemukan ID training

*Actor* : pelaksana



Gambar 4.89 System Flow Change Password

*Objective* : untuk mengubah *password*

*Input* : *user login*

*Proses* : mengubah *password*

*Output* : *field password* pada tabel *user login* berubah

*Actor* : pelaksana

#### 4.7.2 Rancangan Tabel Aplikasi *Training*

##### A. Tabel Data Pegawai *Traning*

Nama Tabel : tabel data pegawai *training*

Primary Key : *TrainingID*

Fungsi : menyimpan data *Training* karyawan

Tabel 4.4 Struktur Tabel Data Pegawai *Training*

NO	Field	Type Data	Field Size	Format	Constraint
1	<i>TrainingID</i>	<i>AutoNumber</i>	<i>Long integer</i>		PK
2	<i>Salnum</i>	<i>Text</i>	255		
3	<i>Employee_name</i>	<i>Text</i>	100		
4	<i>Department</i>	<i>Text</i>	50		
5	<i>Section/Working Area</i>	<i>Text</i>	50		
6	<i>Positions</i>	<i>Text</i>	50		
7	<i>Training Titles</i>	<i>Text</i>	255		
8	<i>Purposes of Training</i>	<i>Text</i>	255		
9	<i>Training Date</i>	<i>Date/Time</i>		<i>Medium Date</i>	
10	<i>Training End Date</i>	<i>Date/Time</i>		<i>Medium Date</i>	
11	<i>Trainer Name</i>	<i>Text</i>	100		
12	<i>Traning Location</i>	<i>Text</i>	100		
13	<i>Type of Training</i>	<i>Text</i>	50		
14	<i>Cettificate</i>	<i>Text</i>	10		

## B. Tabel *User Login*

Nama Tabel : *User Login*

Primary Key : *LoginID*

Fungsi : menyimpan *user login* dan *password*

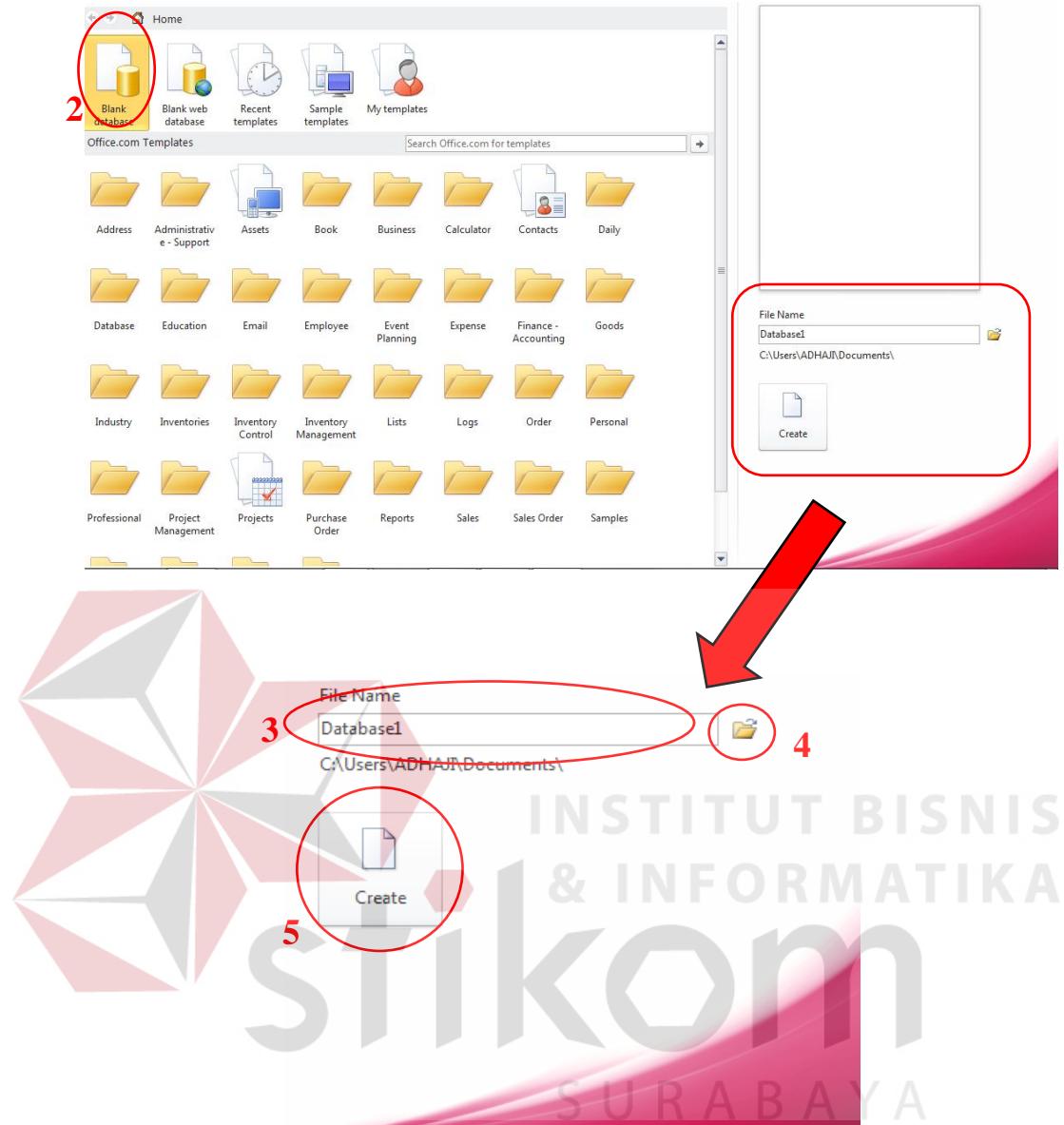
Tabel 4.5 Struktur Tabel *User Login*

NO	Field	Type Data	Field Size	Format	Constraint
1	<i>LoginID</i>	<i>Autonumber</i>	Long Integer		PK
2	<i>User Login</i>	<i>Text</i>	100		
3	<i>Password</i>	<i>Text</i>	50		

### 4.7.3 Pembuatan Aplikasi *Training*

#### A. Membuat *Database*

1. Buka aplikasi *Microsoft Access*
2. Pilih *blank database*.
3. Berikan nama *database*.
4. Pilih tempat penyimpanan *database*
5. Klik *Create*

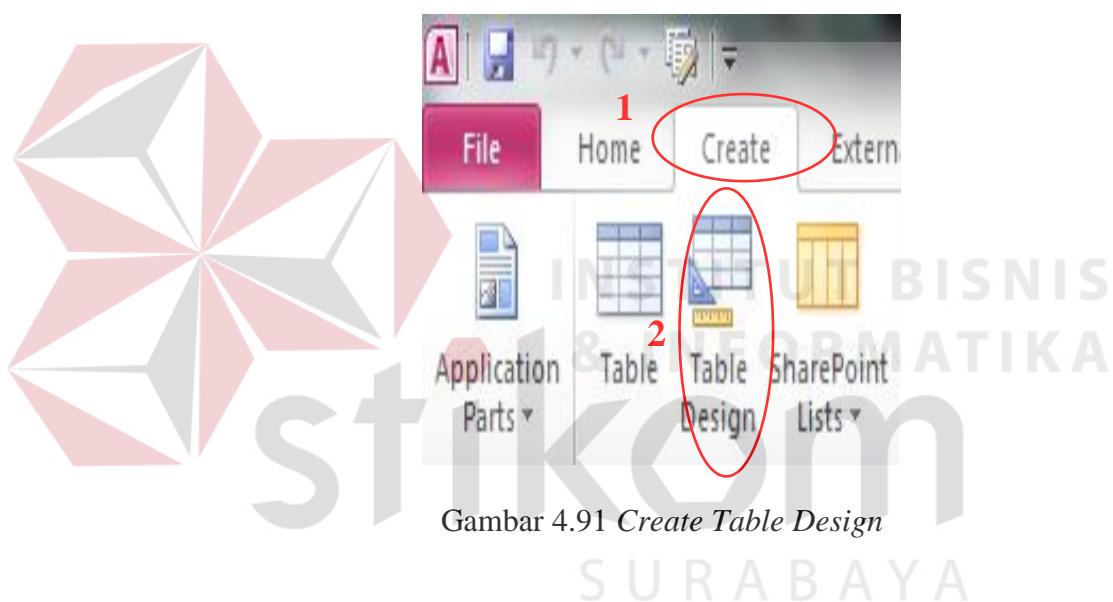


Gambar 4.90 *Create Database*

## B. *Create Table*

Pembuatan tabel untuk aplikasi *Training* hampir sama dengan pembuatan tabel pada aplikasi SP. Pada aplikasi *training* menggunakan 2 tabel, yaitu tabel Data Pegawai *Training* sebagai master *database* untuk menyimpan data *Training* karyawan perusahaan dan tabel *User Login* sebagai *database* *User* yang dapat mengakses aplikasi *Training*. Langkah pembuatan tabel aplikasi *Training* dapat dilihat dibawah ini.

1. Pilih Tab *create* kemudian pilih *toolbar table design*.

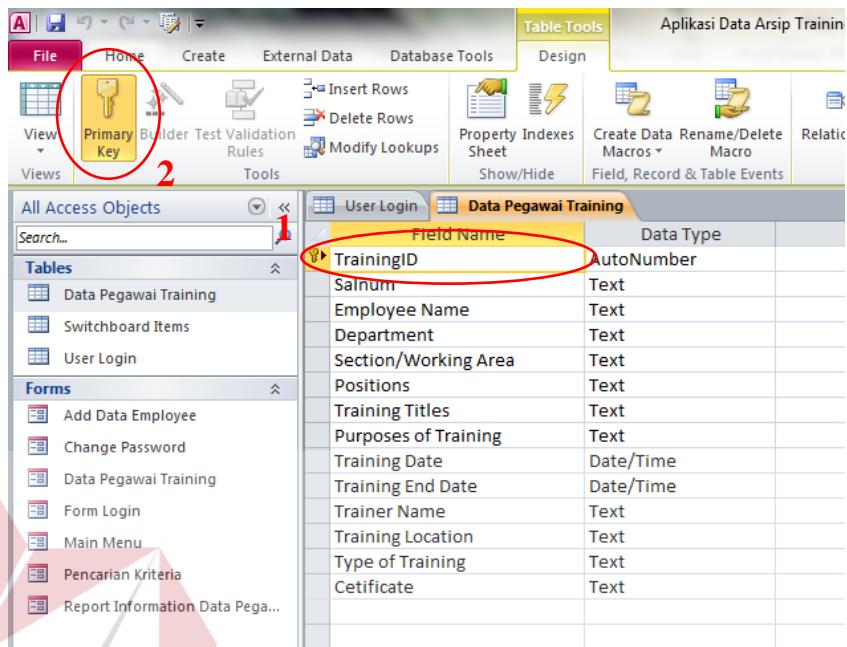


2. Isikan *field name* dan *data type*

Field Name	Data Type
TrainingID	AutoNumber
Salnum	Text
Employee Name	Text
Department	Text
Section/Working Area	Text
Positions	Text
Training Titles	Text
Purposes of Training	Text
Training Date	Date/Time
Training End Date	Date/Time
Trainer Name	Text
Training Location	Text
Type of Training	Text
Certificate	Text

Gambar 4.92 *Field Name* dan *Data Type* Tabel data Pegawai *Training*

3. Klik *Primary Key* untuk menjadikan *field TrainingID* sebagai *Primary key*.

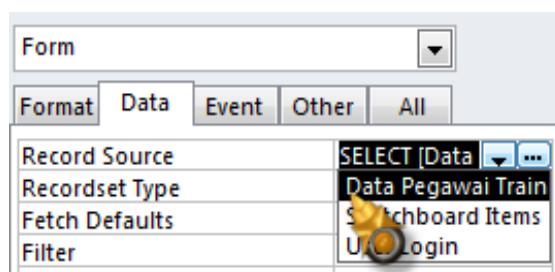


Gambar 4.93 Primary Key

### C. Create Form

Pembuatan *form* pada aplikasi *Training* hampir sama dengan pembuatan *Form* pada aplikasi SP. Dengan menggunakan *toolbar blank form*. Langkah pembuatan formulir *add training form* dapat dilihat dibawah ini.

1. Pilih tab *Create* kemudian pilih *tollbar blank form*
2. *Setting form* dengan tabel data pegawai *training*.



Gambar 4.94 Setting Form

3. Setelah *setting form* isikan formulir dengan *text box* dan *combo box* sesuai dengan jumlah *field* yang ada pada tabel data pegawai *Training*. Seperti gambar dibawah ini.

**ZCCO LEATHER**

**TRAINING FORM**

**Data Employee**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
Salnum				Employee Name		Section/Working Area	Positions										
					Department												
						Section/Working Area											
							Positions										

**Data Training**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
Training Titles										Training Date	Training End Date						
												Trainer Name					
													Training Location				
														Type of Training			
														Certificate			

**Add Data** **Save** **Cancel** **Back**

Gambar 4.95 Isi Form dengan Text Box dan Combo Box

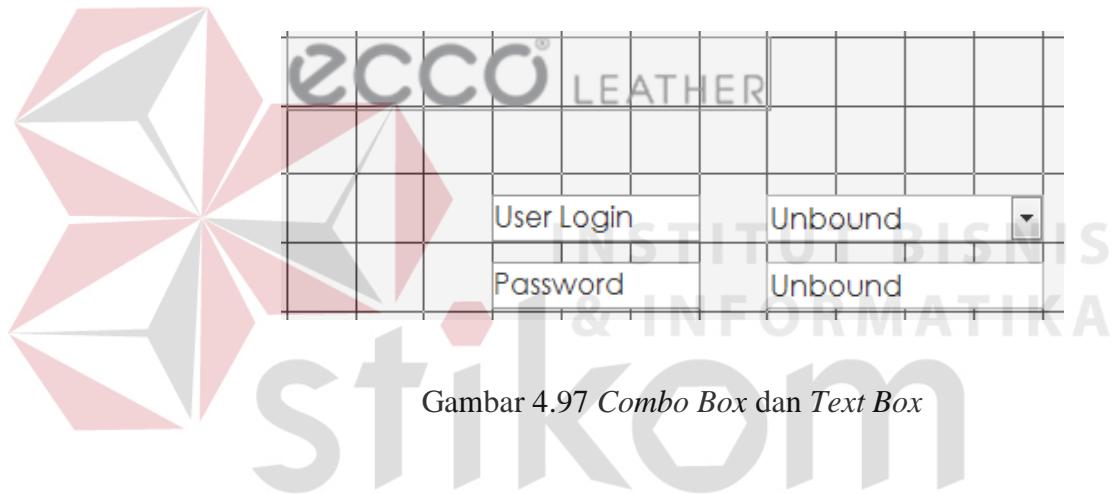
4. Langkah terakhir pembuatan *form* yaitu memberikan tombol *button*.



Gambar 4.96 Tombol *Button*

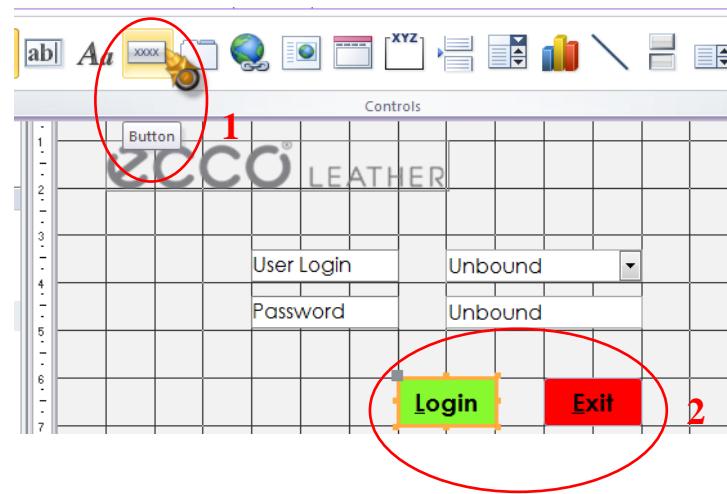
#### D. Create Form Login

1. Pilih tab *create* kemudian klik *toolbar blank form*.
2. Masukkan *combo box* dan *text box*. *Combo box* sebagai *user login*, *text box* sebagai *Password*.



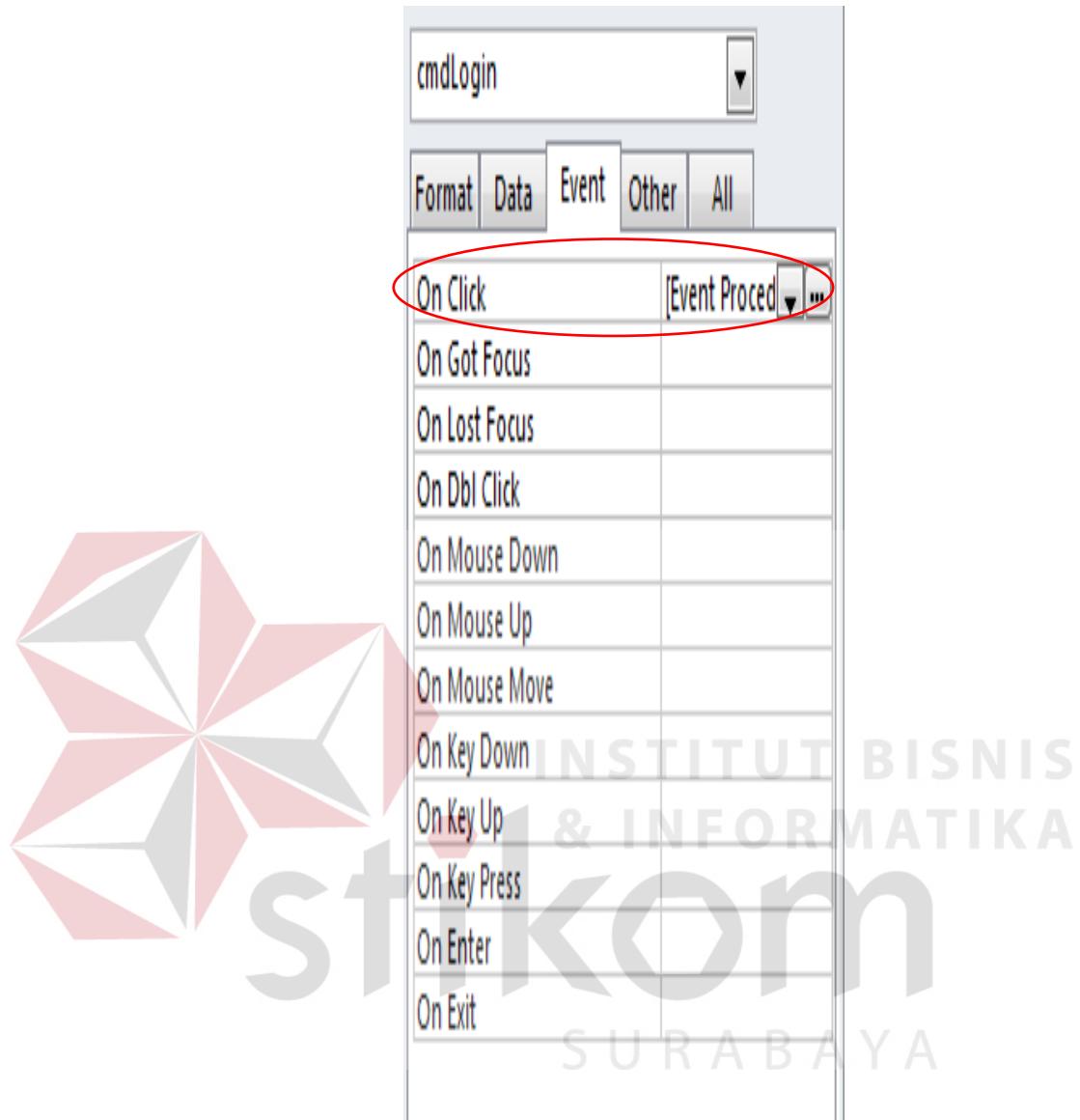
Gambar 4.97 *Combo Box* dan *Text Box*

3. Buat tombol *button* dengan nama *Login* dan *Exit*.



Gambar 4.98 Tombol *Button*

4. Pada *Button Login* berikan *event on click*.



Gambar 4.99 *Event On Click*

5. Berikan *code* dibawah ini.

```

Private Sub cmdLogin_Click()

If IsNull(Me.Cboname) Or Me.Cboname = "" Then

    MsgBox "Please Enter User Login", vbOKOnly, "LoginID
Required"

    Me.Cboname.SetFocus

    Exit Sub

End If

If IsNull(Me.txtpwd) Or Me.txtpwd = "" Then

    MsgBox "Please Enter Password", vbOKOnly, "Password
Required"

    Me.txtpwd.SetFocus

    Exit Sub

End Sub

End If

If Me.txtpwd.Value = DLookup("Password", "User Login",
"[LoginID] = " & Me.Cboname.Value) Then

    MyUserID = Me.Cboname.Value

    MsgBox "Login Succes", vbOKOnly, "Welcome"

    DoCmd.Close acForm, "form LOGIN", acSaveNo

    DoCmd.OpenForm "main menu"

Else

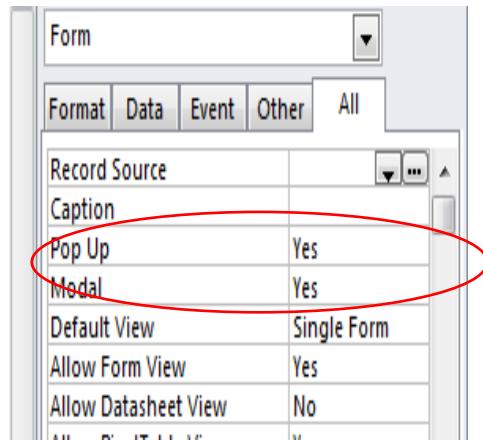
    MsgBox "Cannot Login", vbCritical, "Try Again"

    Me.txtpwd.SetFocus

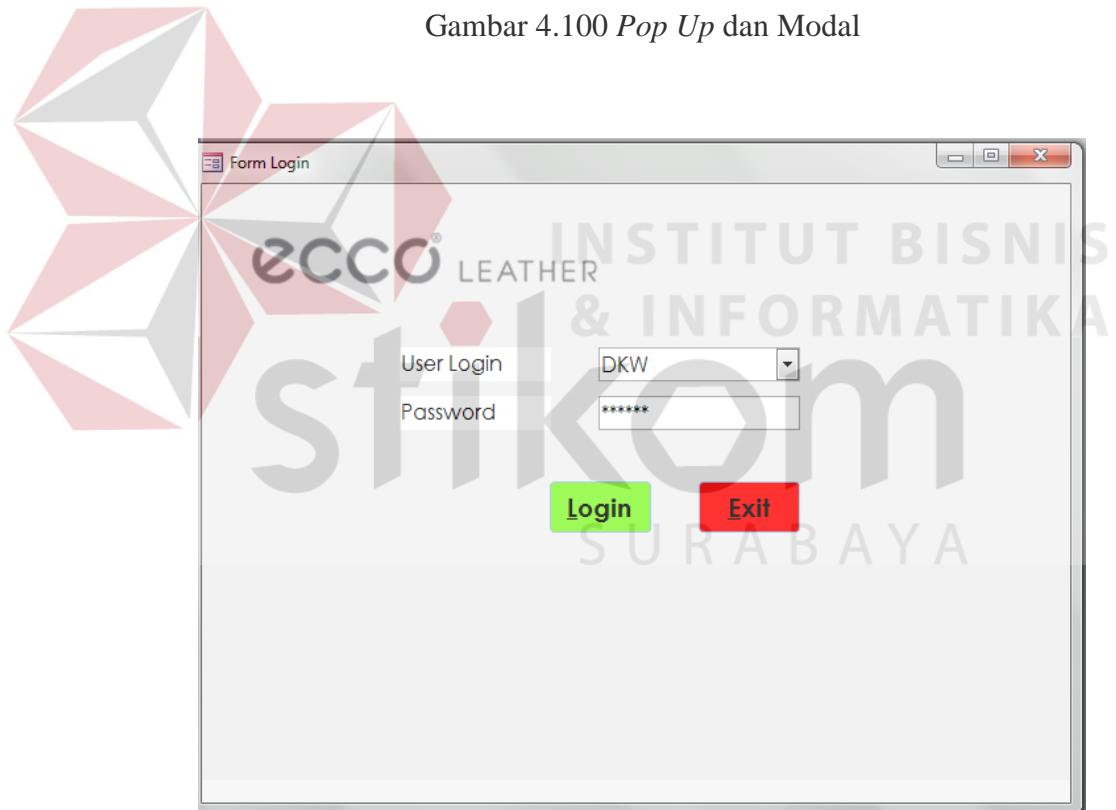
End If

```

6. Ubah *pop up* dan modal dari “NO” menjadi “YES” kemudian tampilkan *form login* menjadi *form view*.



Gambar 4.100 *Pop Up* dan *Modal*



Gambar 4.101 *Form Login*