

# **Laporan Aplikasi Sistem Ekspedisi Barang Tugas Besar Basis Data**

**Disusun oleh:**

<b>Aulia Rahman Zulfi</b>	<b>119140110</b>
<b>Auriga Farhan Rizkiadhy</b>	<b>119140153</b>
<b>Fadhillah Azhar Alsani</b>	<b>119140217</b>
<b>Muhammad Daffa Massimiliano</b>	<b>119140139</b>
<b>Sayyid Muhammad Umar Al Haris</b>	<b>119140190</b>

**Kelas RC**



**Program Studi Teknik Informatika  
Jurusan Teknik Elektro, Informatika, dan Sistem Fisika  
Institut Teknologi Sumatera  
Lampung Selatan  
2021**

## DAFTAR ISI

DAFTAR ISI.....	2
BAB I .....	3
PENDAHULUAN .....	3
1.1 Latar Belakang.....	3
1.2 Rumusan Masalah.....	3
1.3 Tujuan Pengembangan Aplikasi .....	3
BAB II .....	4
RANCANGAN SISTEM .....	4
2.1 Deskripsi Sistem .....	4
2.2 Use Case Diagram .....	4
2.3 Rancangan Basis Data .....	5
BAB III .....	8
IMPLEMENTASI DAN PEMBAHASAN.....	8
3.1. Implementasi Basis Data.....	8
3.2 Penjelasan Fitur/Fungsi Aplikasi .....	10
BAB IV.....	24
KESIMPULAN DAN SARAN.....	24
4.1. Kesimpulan.....	24
4.2. Saran .....	24
DAFTAR PUSTAKA .....	25
LOG SHEET KEGIATAN .....	26
BIODATA DAN LINGKUP KERJA ANGGOTA.....	28

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Layanan ekspedisi merupakan salah satu layanan yang amat sering dimanfaatkan pada masa ini dimana sedang populernya belanja secara *online* atau dalam jaringan internet. Selain dari belanja *online*, layanan ekspedisi juga banyak digunakan untuk mengirimkan barang-barang untuk keperluan lain selain bagian dari transaksi *online*, contohnya adalah mengirimkan barang ke keluarga atau mengirimkan barang kepentingan pekerjaan.

Pada abad ke-21 ini, dimana internet sudah menjadi hal yang amat awam untuk diakses oleh kebanyakan orang, tentu saja banyak sekali aktivitas yang sebelumnya memerlukan kita untuk keluar rumah, sekarang dapat dilakukan dari mana saja dengan menggunakan akses internet. Hal ini terhubung dengan industri layanan ekspedisi dimana layanan ini menjadi salah satu layanan yang sering digunakan atau diperlukan pada masa ini. Dalam memastikan jalannya layanan dengan baik, dibutuhkan suatu cara untuk menyimpan informasi dari sistem layanan ekspedisi. Informasi dari sistem layanan ekspedisi ini dapat diintegrasikan ke dalam suatu basis data sehingga dapat diproses oleh komputer secara sistematis. Dengan menggunakan basis data, informasi-informasi terkait perjalanan dari sistem layanan ekspedisi dapat ditambahkan, diedit, dan dihapus dan terhubung ke dalam sebuah tempat yang sama yaitu suatu basis data, sehingga tidak terjadinya kesalahan informasi pada sistem layanan.

Untuk itu, demi menjaga sistem pelayanan ekspedisi dapat dijalankan secara lancar, dan mencegah atau mengurangi kemungkinan adanya salah informasi akibat ketidakkonsistenan data, dikembangkanlah aplikasi berbasis data yang dapat mengelola dan mengisi isi basis data dari sistem layanan ekspedisi.

### **1.2 Rumusan Masalah**

1. Bagaimana bentuk rancangan dari basis data pada sistem ekspedisi barang?
2. Bagaimana hubungan entitas-entitas yang terdapat pada basis data sistem?
3. Bagaimana mengimplementasikan operasi CRUD (*create, read, update, delete*) pada aplikasi yang dapat terhubung ke dalam basis data?

### **1.3 Tujuan Pengembangan Aplikasi**

1. Untuk mengetahui bentuk rancangan dari basis data pada sistem ekspedisi barang.
2. Untuk mengetahui hubungan dari entitas-entitas yang terdapat pada basis data.
3. Untuk dapat mengerti bagaimana pengimplementasian operasi CRUD pada aplikasi yang terhubung ke dalam suatu basis data.

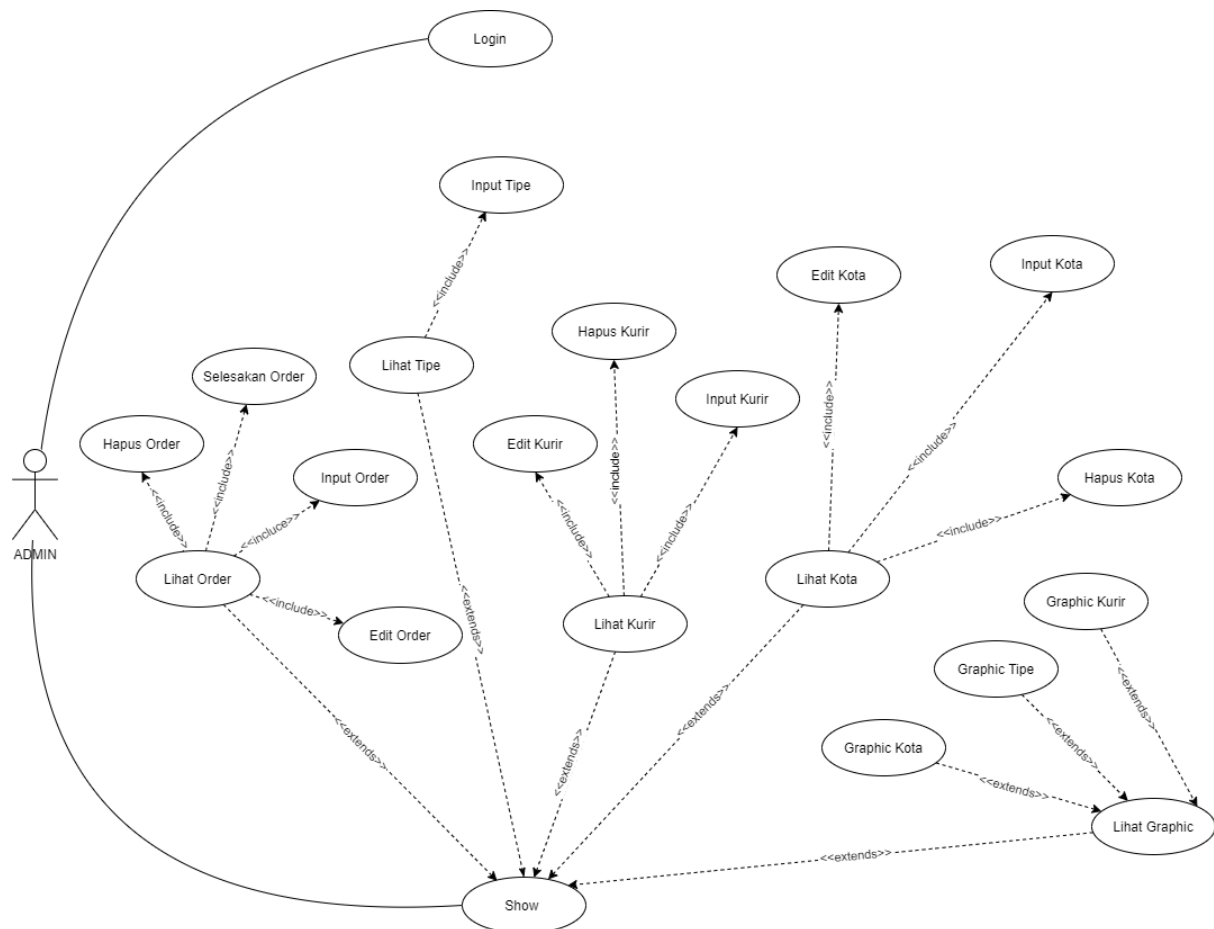
## BAB II

### RANCANGAN SISTEM

#### 2.1 Deskripsi Sistem

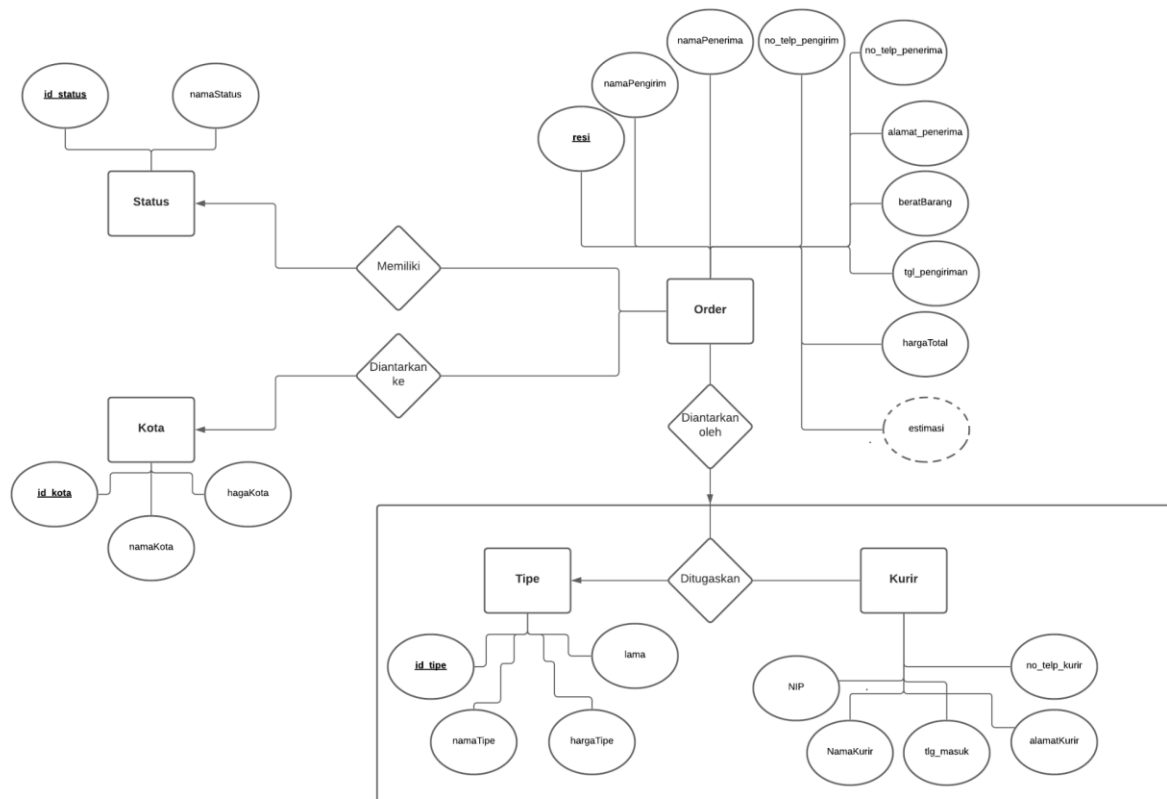
Aplikasi ini merupakan aplikasi dari sistem ekspedisi yang diperuntukan untuk admin atau pegawai dari perusahaan ekspedisi. Aplikasi ini ditulis dengan bahasa pemrograman utama *Python* dengan *framework*-nya *Flask*, dan dengan basis data *MySQL*. Aplikasi dapat menambahkan data order ekspedisi yang berisi data-data berkaitan dengan pengirim, penerima, dan barang, serta hubungannya dengan kota dan kurir yang ditugaskan untuk membawa barang tersebut. Aplikasi juga dapat mengedit dan menghapus data-data yang telah ditambahkan ke dalam basis data.

#### 2.2 Use Case Diagram

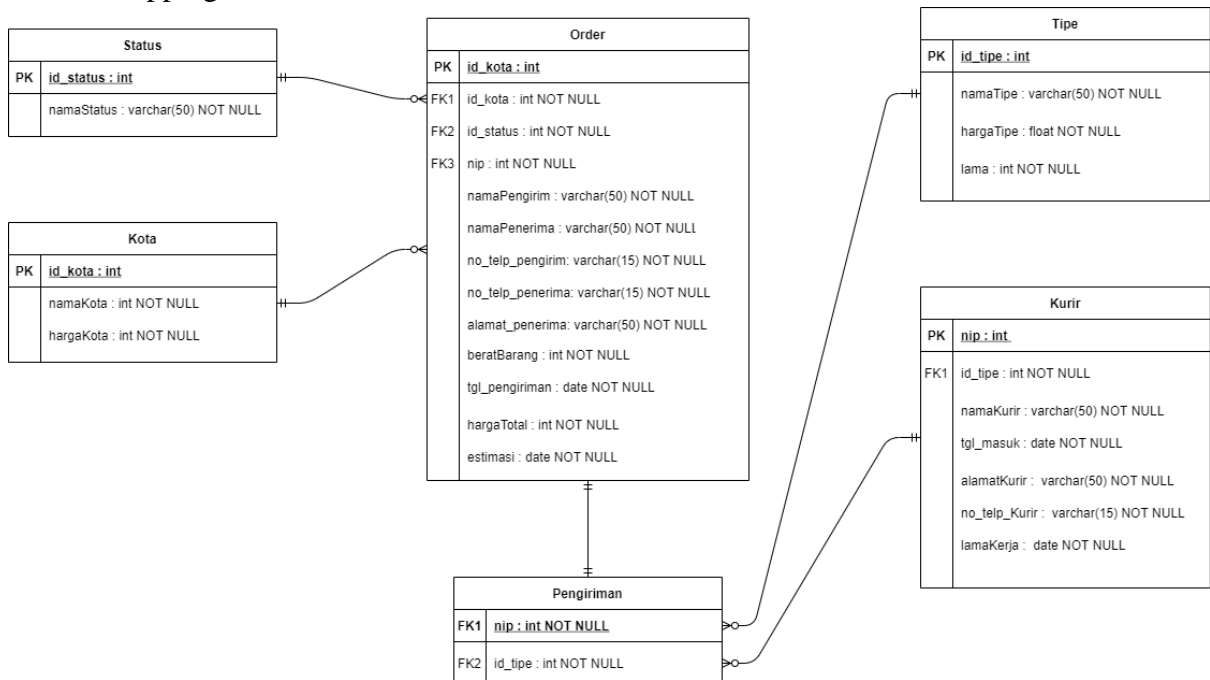


## 2.3 Rancangan Basis Data

### A. Rancangan ERD



### B. Mapping ERD ke Relational Model



### C. Perancangan Fisik (Integrity Constraint)

#### Admin

Name	Type	NULL	Default
nim (PK)	varchar(10)	No	None
name	varchar(50)	No	None
alamat	varchar(50)	No	None
password	varchar(50)	No	None

#### Kota

Name	Type	NULL	Default
id_kota (PK)	varchar(10)	No	None
namaKota	varchar(50)	No	None
hargaKota	int(20)	No	None

#### Kurir

Name	Type	NULL	Default
nip (PK)	int(10)	No	None
namaKurir	varchar(50)	No	None
tgl_masuk	date	No	None
alamatKurir	varchar(50)	No	None
no_telp_kurir	varchar(50)	No	None
id_tipe (FK)	int(10)	No	None

#### Order2

Name	Type	NULL	Default
resi (PK)	int(20)	No	None
namaPengirim	varchar(50)	No	None
namaPenerima	varchar(50)	No	None
no_telp_pengirim	varchar(50)	No	None
no_telp_penerima	varchar(50)	No	None

alamat_penerima	varchar(50)	No	None
beratBarang	int(10)	No	None
tgl_pengiriman	date	No	None
hargaTotal	int(10)	No	None
nip (FK)	int(10)	No	None
id_kota (FK)	int(10)	No	None
id_status (FK)	int(10)	No	None
estimasi	date	No	None

### Status

Name	Type	NULL	Default
id_status (PK)	int(10)	No	None
namaStatus	varchar(50)	No	None

### Tipe

Name	Type	NULL	Default
id_tipe (FK)	int(10)	No	None
namaTipe	int(50)	No	None
hargaTipe	int(5)	No	None
lama	int(10)	No	None

## BAB III

### IMPLEMENTASI DAN PEMBAHASAN

#### 3.1. Implementasi Basis Data

##### Basis Data Ekspedisi

Pada basis data ini bernama ekspedisi. Basis Data ini memiliki tabel admin, kota, kurir, order2, status dan tipe.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin	Browse  Structure  Search  Insert  Empty  Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> kota	Browse  Structure  Search  Insert  Empty  Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> kurir	Browse  Structure  Search  Insert  Empty  Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> order2	Browse  Structure  Search  Insert  Empty  Drop	2	InnoDB	utf8mb4_general_ci	64.0 KiB	-
<input type="checkbox"/> status	Browse  Structure  Search  Insert  Empty  Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> tipe	Browse  Structure  Search  Insert  Empty  Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
6 tables	Sum	15	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

##### 3.1.1. Tabel admin

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	nim	varchar(10)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 2	name	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	alamat	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	password	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More

Pada tabel ini, nim menjadi primary key.

##### 3.1.2. Tabel kota

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id_kota	int(10)			No	None			Change  Drop  More
<input type="checkbox"/> 2	namaKota	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	hargaKota	int(20)			No	None			Change  Drop  More

Pada tabel ini, id\_kota menjadi primary key.

##### 3.1.3. Tabel kurir

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	nip	int(10)			No	None			Change  Drop  More
<input type="checkbox"/> 2	namaKurir	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	tgl_masuk	date			No	None			Change  Drop  More
<input type="checkbox"/> 4	alamatKurir	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	no_telp_Kurir	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 6	id_tipe	int(10)			No	None			Change  Drop  More



Pada tabel ini, nip menjadi primary key. id\_tipe disini merupakan foreign key yang referensi merupakan dari tabel tipe pada kolom id\_tipe.

### 3.1.4 Tabel order2

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 resi	int(20)			No	None			Change  Drop  More
<input type="checkbox"/>	2 namaPengirim	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 namaPenerima	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 no_telp_pengirim	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5 no_telp_penerima	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6 alamat_penerima	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7 beratBarang	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	8 tgl_pengiriman	date			No	None			Change  Drop  More
<input type="checkbox"/>	9 hargaTotal	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	10 nip	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	11 id_kota	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	12 id_status	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	13 estimasi	date			No	None			Change  Drop  More

Pada tabel ini, resi menjadi primary key. Kemudian nip, id\_kota, id\_status merupakan foreign key. Foreign key nip berasal dari tabel kurir, id\_kota berasal dari tabel kota, dan id\_status berasal dari tabel status.

### 3.1.5 Tabel status

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_status	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	2 namaStatus	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More

Pada tabel ini, id\_status merupakan primary key.

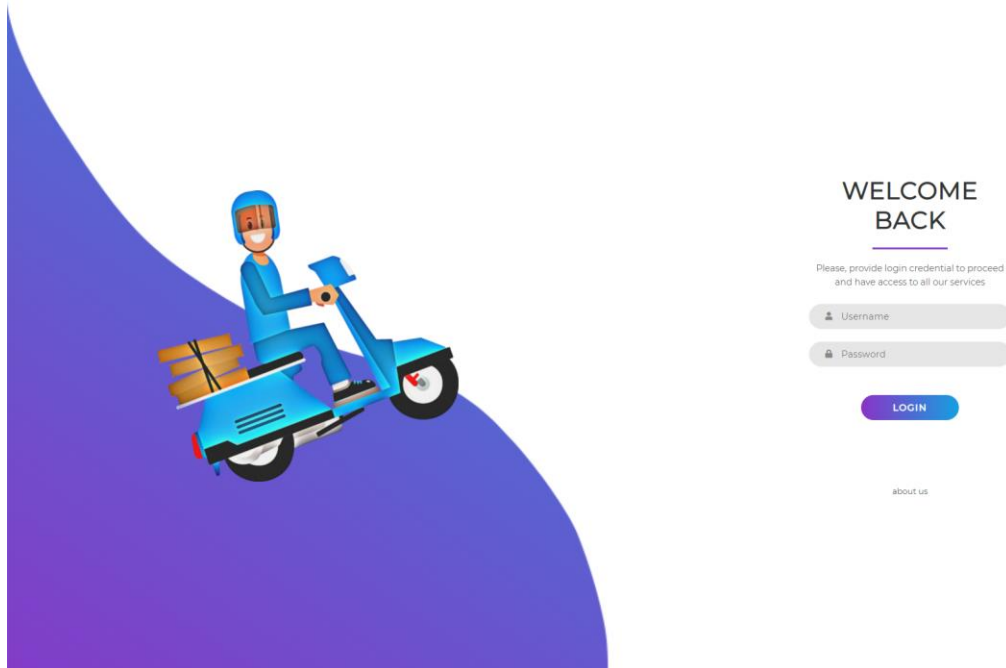
### 3.1.6 Tabel tipe

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_tipe	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	2 namaTipe	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 hargaTipe	int(5)			No	None			Change  Drop  More
<input type="checkbox"/>	4 lama	int(10)			No	None			Change  Drop  More

Pada tabel ini, id\_tipe merupakan primary key.

## 3.2 Penjelasan Fitur/Fungsi Aplikasi

### 3.2.1 Login



Landing Page akan menampilkan Login Page yang akan meminta user untuk memasukkan username dan password.

```
userid = request.form['nim']
password = request.form['password']

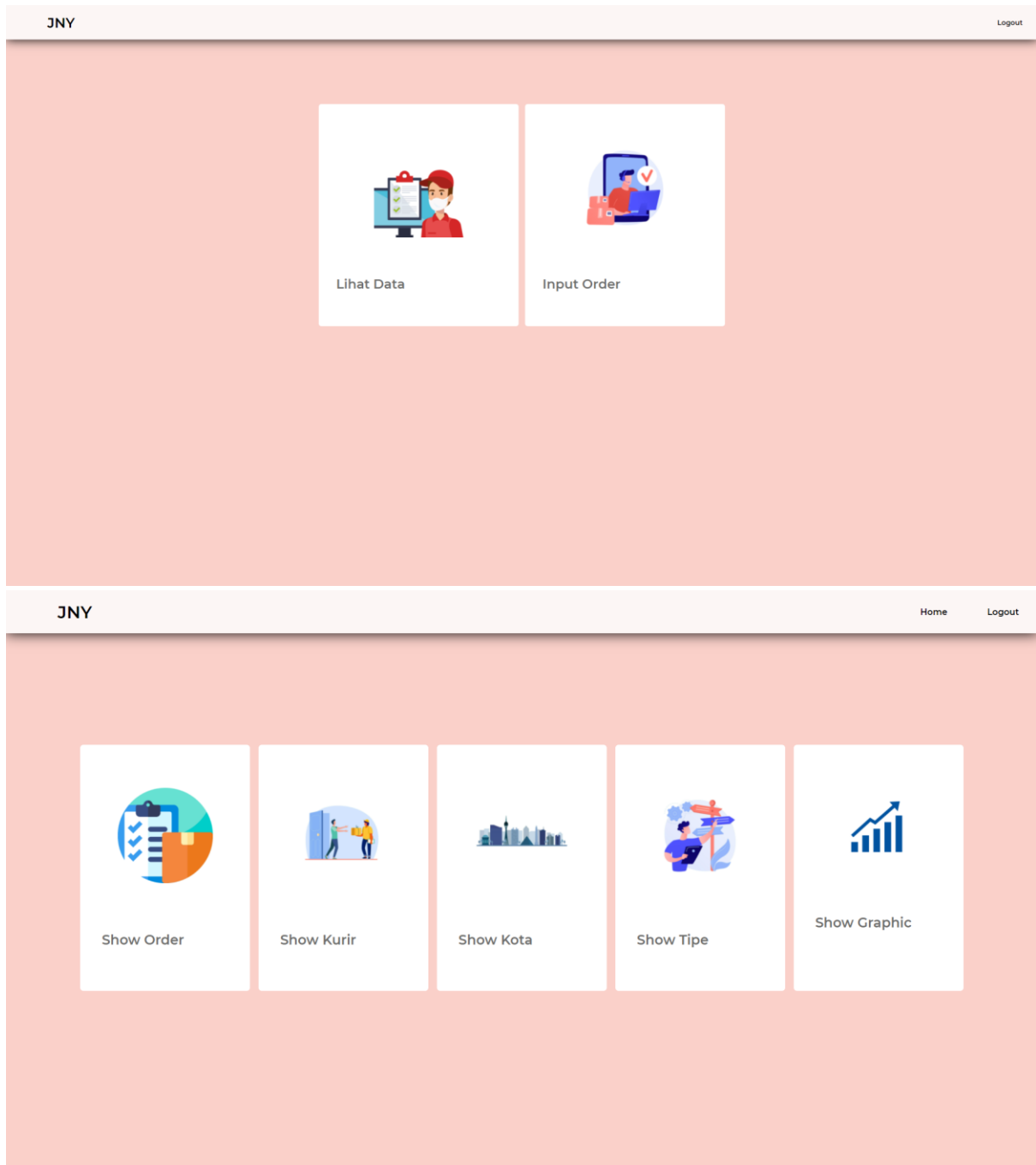
cur = mysql.connection.cursor()
cur.execute('select * from admin')
admin = cur.fetchall()

for i in range(len(admin)):
    if (str(admin[i][0]) == str(userid)) and (str(admin[i][3]) == str(password)):

        loguser = user(admin[i][0],admin[i][1],admin[i][2])
        return redirect(url_for('menu'))
```

Pada kode program userid dan password dapat dari username dan password dari masukkan pengguna, setelah itu pemanggilan dari database select \* from admin. Kemudian userid dan password dicocokkan dengan username dan password pada database. Jika benar akan melanjutkan pada menu.

### 3.2.2 Menu



Pada menu utama, terdapat pilihan untuk lihat data atau input order yang dapat di pilih oleh user. Pada halaman lihat data, terdapat pilihan untuk melihat *order*,

### 3.2.3 Show

JNY

HomeMenuLogout

List Order

RESI	NAMA PENGIRIM	NAMA PENERIMA	NO TELP PENGIRIM	NO TELP PENERIMA	ALAMAT PENERIMA	BERAT BARANG	TGL PENGIRIMAN	HARGA TOTAL	NIP	ID KOTA	STATUS	ESTIMASI	
543676575	Parjo	Deka	085435343426	085435345654	Jalan Denpasar	8	2021-05-19	160000	1	3	Terkirim	2021-05-24	<a href="#">EDIT</a> <a href="#">hapus</a>
543676576	Sayyid	Haris	08953485023	081231827122	Jalan Sepuncak	3	2021-05-19	96000	5	1	Dalam Proses	2021-05-20	<a href="#">SELESAI</a> <a href="#">EDIT</a> <a href="#">hapus</a>
543676577	Sims	Maung	08123126382	08187236182	Jalan Cambuk Kemayoran	4	2021-05-19	52000	2	2	Dalam Proses	2021-05-24	<a href="#">SELESAI</a> <a href="#">EDIT</a> <a href="#">hapus</a>
543676578	Auau	Padil	085346577564	082178696789	Jalan Spens Bersatu	11	2021-05-19	176000	1	1	Dalam Proses	2021-05-24	<a href="#">SELESAI</a> <a href="#">EDIT</a> <a href="#">hapus</a>
543676579	Kaith	Ky	081221122112	082112211221	Tengkluyu Foreva Street	1	2021-05-19	139000	2	6	Dalam Proses	2021-05-24	<a href="#">SELESAI</a> <a href="#">EDIT</a> <a href="#">hapus</a>

[INPUT ORDER](#)  
[Kembali](#)

```
cur = mysql.connection.cursor()
cur.execute(f'select * from order2')
order2 = cur.fetchall()
cur.close()

return render_template('/show/order.html',order2=order2)
```

Pada kode program ini akan menampilkan semua kolom pada tabel order2 dengan sintaks select \* from order2.

JNY

HomeMenuLogout

List Kurir

NIP	NAMA KURIR	TGL MASUK	ALAMAT KURIR	NO TELP KURIR	ID TIPE	BANYAK ORDER	
1	Joni	2021-01-13	Jalan Sepuncak	081285671346	1	2	<a href="#">edit</a> <a href="#">hapus</a>
2	Odin	2011-03-16	Kemanggisan	084435234342	1	2	<a href="#">edit</a> <a href="#">hapus</a>
3	Dina	2017-01-11	Cisalak	084324321323	2	0	<a href="#">edit</a> <a href="#">hapus</a>
4	Khloe	2021-05-05	Beverly Hills	081100001111	2	0	<a href="#">edit</a> <a href="#">hapus</a>
5	Kurbar	2021-05-18	Jalan Semanggis	081234567890	3	1	<a href="#">edit</a> <a href="#">hapus</a>

[INPUT Kurir](#)  
[Kembali](#)

```

cur = mysql.connection.cursor()
cur.execute(f'select * from kurir')
kurir = cur.fetchall()
banyakOrder = []
for i in kurir:
    cur.execute(f'select count(resi) from order2 where nip={i[0]}')
    temp = cur.fetchall()
    banyakOrder.append(temp[0][0])
cur.close()
return render_template('/show/kurir.html',kurir=kurir, banyakOrder=banyakOrder)

```

Pada kode program ini akan menampilkan semua kolom pada tabel kurir dengan sintaks `select * from kurir`. Kemudian, untuk menampilkan banyakorder dengan menggunakan `count` pada kolom menggunakan sintaks `select count(resi) from order2 where nip=<setiap-nip-kurir>`.

JNY

Home Menu Logout

List Kota

ID KOTA	NAMA KOTA	HARGA KOTA	BANYAK ORDER	
1	Palembang	16000	2	edit hapus
2	Jakarta	13000	1	edit hapus
3	Bali	20000	1	edit hapus
4	Bandung	14000	0	edit hapus
5	Manado	35000	0	edit hapus
6	New Delhi	139000	1	edit hapus

INPUT Kota  
Kembali

```

cur = mysql.connection.cursor()
cur.execute(f'select * from kota')
kota = cur.fetchall()
banyakOrder = []
for i in kota:
    cur.execute(f'select count(resi) from order2 where id_kota={i[0]}')
    temp = cur.fetchall()
    banyakOrder.append(temp[0][0])
cur.close()
return render_template('/show/kota.html',kota=kota, banyakOrder=banyakOrder)

```

Kode program ini akan menampilkan semua kolom pada tabel kota menggunakan sintaks `select * from kota`. Kemudian untuk menghitung banyak order menggunakan `select count(resi) from order2 where id_kota=<setiap-id-kota>`.

## Tipe Pengiriman

ID TIPE	NAMA TIPE	HARGA TIPE	LAMA PENGIRIMAN	BANYAK	DETAILS
1	REG	1.0	5	4	<a href="#">SHOW</a>
2	FLASH	1.5	3	0	<a href="#">SHOW</a>
3	INSTANT	2.0	1	1	<a href="#">SHOW</a>

[Kembali](#)

```
cur = mysql.connection.cursor()
cur.execute(f'select * from kurir')
kurir = cur.fetchall()
banyakOrder = []
for i in kurir:
    cur.execute(f'select count(resi) from order2 where nip={i[0]}')
    temp = cur.fetchall()
    banyakOrder.append(temp[0][0])
banyakOrder2 = []
cur.execute(f'select * from tipe')
tipe= cur.fetchall()
for i in tipe:
    banyaktemp = 0
    for j in range(0, len(kurir)):
        if(i[0]==kurir[j][5]):
            banyaktemp += banyakOrder[j]
    banyakOrder2.append(banyaktemp)
cur.close()

return render_template('/show/tipe.html',tipe=tipe, banyakOrder=banyakOrder2)
```

Pada kode program ini akan menampilkan semua kolom yang ada pada tabel kurir menggunakan sintaks `select * from kurir`. Kemudian untuk menghitung banyak menggunakan sintaks `select count(resi) from order2 where nip=<setiap-nip-kurir>`.

### 3.2.4. Input

JNY

HomeLogout

INPUT ORDER

Resi

Nama Pengirim

Nama Penerima

Telepon Pengirim

Telepon Penerima

Alamat Penerima

Berat Barang

Kurir

REG - Joni

Kota

Palembang

Submit

```
cur = mysql.connection.cursor()
cur.execute(f"select id_tipe from kurir where nip='{j}'")
id_tipe= cur.fetchall()
id_tipe= id_tipe[0][0]

cur.execute(f"select hargaTipe,lama from tipe where id_tipe='{id_tipe}'")
hargaTipeLama = cur.fetchall()

hargaTipe = hargaTipeLama[0][0]
lama= hargaTipeLama[0][1]
hargaTipe = float(hargaTipe)/2

cur.execute(f"select hargaKota from kota where id_kota='{k}'")
KotaLama = cur.fetchall()
Kota=KotaLama[0][0]

hargaTipe = hargaTipe * int(g) * int(Kota)

x = datetime.datetime.now()
tgl = (f'{x.year}-{x.month}-{int(x.day)+lama}')

cur.close()
cur = mysql.connection.cursor()

cur.execute("""INSERT INTO order2 (resi,namaPengirim,namaPenerima,no_telp_pengirim,
no_telp_penerima,alamat_penerima,beratBarang,tgl_pengiriman,hargaTotal,nip,id_kota,id_status,estimasi)
VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)""", (a,b,c,d,e,f,g,datetime.date.today(),hargaTipe,j,k,'0',tgl))
mysql.connection.commit()

cur.close()

return redirect('/show/order')
```

Pada kode program ini, karena harga tidak masukkan secara langsung, maka untuk mencari harga melalui pencarian nip kurir menggunakan `select id_tipe from kurir where nip={input_nip_kurir}`. Setelah mendapatkan `id_tipe`, lalu dapat dicari `hargaTipe` menggunakan sintaks `select hargaTipe, lama from tipe where id_tipe={id_tipe_yang_dicari}`. Kemudian, untuk mencari `hargaKota` dapat digunakan sintaks `select hargaKota from kota where id_kota={input_kota}`. Setelah melakukan perhitungan untuk harga, kemudian dapat memasukkan kedalam database menggunakan `insert into order2` yang valuenya menerima masukkan dari pengguna dan perhitungan harga dari sistem.

JNYHomeLogout

### INPUT KURIR

NIP

Nama

Tanggal Masuk  
mm/dd/yyyy

Alamat

No. Telepon

Tipe : REG

Submit

```
cur = mysql.connection.cursor()
cur.execute(f"insert into kurir values ('{a}','{b}','{c}','{d}','{e}','{g}')"
cur.connection.commit()
cur.close()

return redirect('/show/kurir')
```

Pada kode program ini akan memasukkan nilai pada tabel kurir. Values disini bersumber dari input user yaitu nip, nama, tanggal masuk, alamat, no telpon dan tipe.

JNYHomeLogout

### INPUT KOTA

ID\_Kota

Nama Kota

Harga/kg

Submit



```

cur = mysql.connection.cursor()
cur.execute(f"insert into kota values ('{a}','{b}','{c}')"")
cur.connection.commit()
cur.close()

return redirect('/show/kota')

```

Pada kode program ini akan memasukkan nilai pada tabel kota. Values disini bersumber dari input user yaitu id\_kota, nama kota dan harga/kg.

### 3.2.5 Edit

JNY
Home
Menu
Logout

Detail Order

Label	Isi
RESI	543676575
Nama Pengirim	Parjo
Nama Penerima	Delia
No Telp Pengirim	085435343426
No Telp Penerima	085435345654
Alamat Tujuan	Jalan Dempasar
Berat Barang	8
Tanggal pengiriman	2021-05-19
Harga Bayar Total	160000
NIP Kurir	1
ID Tipe	3
Status	1
Estimasi Sampai	2021-05-24
Nama Kurir	Joni
Nama Kota	Ball
No Telp Kurir	081285670348
Harga Kota (/kg)	20000
Tipe Pengiriman	REG
Harga Tipe	1.0X

EDIT DATA  
Nama Penerima Baru:

No. telp penerima baru:

Alamat penerima baru (kota yang sama):

Ganti Data

EDIT DATA  
Nama Penerima Baru:

No. telp penerima baru:

Alamat penerima baru (kota yang sama):

Ganti Data

```

addrbaru = request.form['addrbaru']
cur = mysql.connection.cursor()
if(namabaru != ''):
    cur.execute(f"update order2 set namaPenerima = '{namabaru}' where resi = '{id}'")
    cur.connection.commit()
if(nobaru != ''):
    cur.execute(f"update order2 set no_telp_penerima = '{str(nobaru)}' where resi = '{id}'")
    cur.connection.commit()
if(addrbaru != ''):
    cur.execute(f"update order2 set alamat_penerima = '{str(addrbaru)}' where resi = '{id}'")
    cur.connection.commit()
cur.close()

```

Pada kode program ini, jika user memasukkan sesuatu pada masing-masing tabel input maka program akan mengubah nilai pada kolom yang terkena dampak. Mengganti data pada order dapat menggunakan sintaks `update order2 set {nama_kolom} = {input_user} where resi = {no_resi}`.

```

cur = mysql.connection.cursor()
cur.execute(f"""SELECT order2.*, namaKurir, no_telp_Kurir, hargaKota,
namaTipe, hargaTipe
FROM order2 NATURAL JOIN kurir NATURAL JOIN tipe NATURAL JOIN kota where resi='{id}'
""")
orderDetail = cur.fetchall()
orderDetail = orderDetail[0]
cur.close()

```

Pada kode program ini sistem akan menampilkan semua kolom yang ada pada order2, ditambah dengan kolom namaKurir, no\_telp\_Kurir, hargaKota, namaTipe, dan hargaTipe dari tabel kurir dengan natural join.

JNY
Home
Menu
Logout

### Tipe Pengiriman

Label	Isi
NIP	1
ID Tipe	1
Nama Kurir	Joni
Tanggal Masuk	2021-01-13
Alamat Kurir	Jalan Sepuncak
No Telp	081285671346
Tipe Pengiriman	REG
Harga Tipe	2
Estimasi Lama Pengiriman	5
Banyak Order	2

EDIT DATA  
No. telp kurir baru:   
Alamat kurir baru:

```

cur = mysql.connection.cursor()
if(nobaru != ''):
    cur.execute(f"update kurir set no_telp_Kurir = '{str(nobaru)}' where nip = '{id}'")
    cur.connection.commit()
if(addrbaru != ''):
    cur.execute(f"update kurir set alamatKurir = '{str(addrbaru)}' where nip = '{id}'")
    cur.connection.commit()
cur.close()

```

Pada kode program ini akan mengubah nilai pada tabel kurir. Jika user memasukkan sesuatu pada tabel input, maka tabel yang terkena dampak akan berubah nilainya pada database dengan sintaks update kurir set {kolom} = {input\_user} where nip = {no\_resi}.

```
cur = mysql.connection.cursor()
cur.execute(f'select * from kurir natural join tipe where nip={id}')
kurir = cur.fetchall()
for i in kurir:
    cur.execute(f'select count(resi) from order2 where nip={i[0]}')
    temp = cur.fetchall()
    banyakOrder = temp[0][0]
cur.close()
```

Pada kode program ini akan menampilkan semua kolom pada tabel kurir dan tabel tipe. Kemudian untuk menampilkan kolom banyak order dapat melakukan perhitungan menggunakan sintaks select count(resi) from order2 where nip={id\_kurir}.

JNY
Home
Menu
Logout

Detail Kota

Label	Isi
ID Kota	1
Nama Kota	Palembang
Harga Kota	16000
Banyak Order	2

EDIT DATA  
Harga kota baru:

ganti data

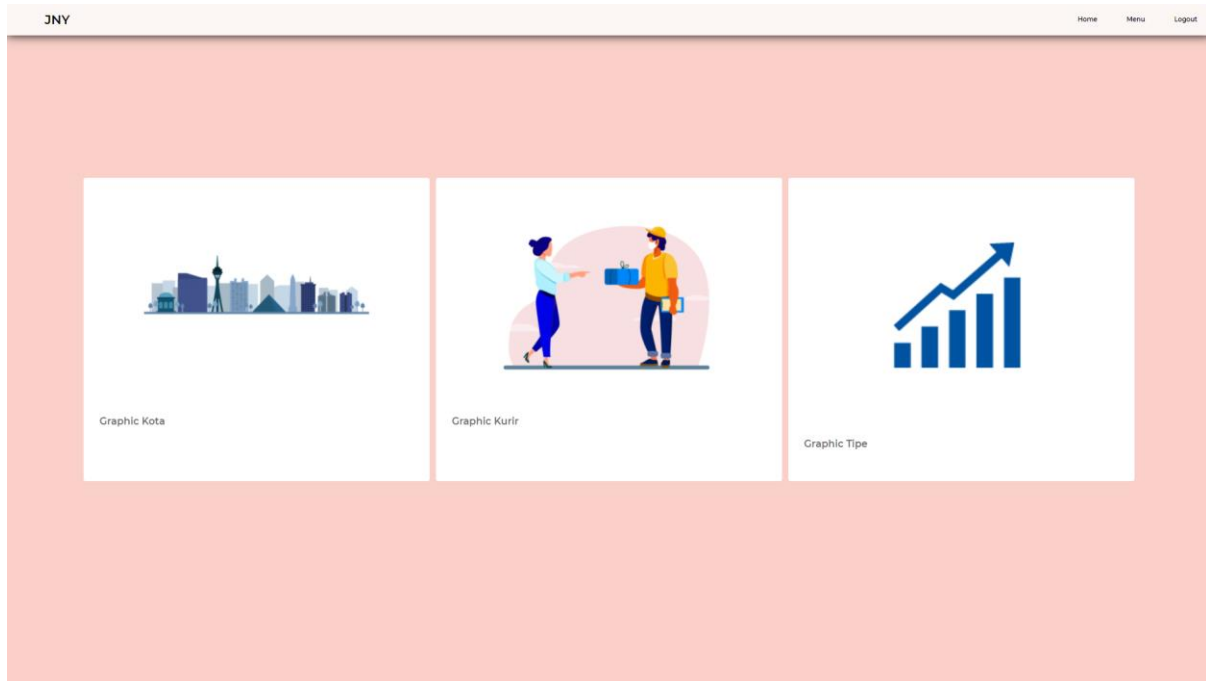
```
cur = mysql.connection.cursor()
if(hargabaru != ''):
    cur.execute(f"update kota set hargaKota = '{str(hargabaru)}' where id_kota = '{id}'")
    cur.connection.commit()
cur.close()
repres = '/editKota/' + str(id)
return redirect(repres)
```

Pada kode program ini akan mengubah harga pada tabel kota jika user memasukkan nilai pada tabel input. Sintaks yang digunakan update kota set hargaKota = {harga\_baru} where id\_kota= {id}.

```
cur.execute(f'select * from kota where id_kota={id}')
kota = cur.fetchall()
for i in kota:
    cur.execute(f'select count(resi) from order2 where id_kota={i[0]}')
    temp = cur.fetchall()
    banyakOrder = temp[0][0]
cur.close()
```

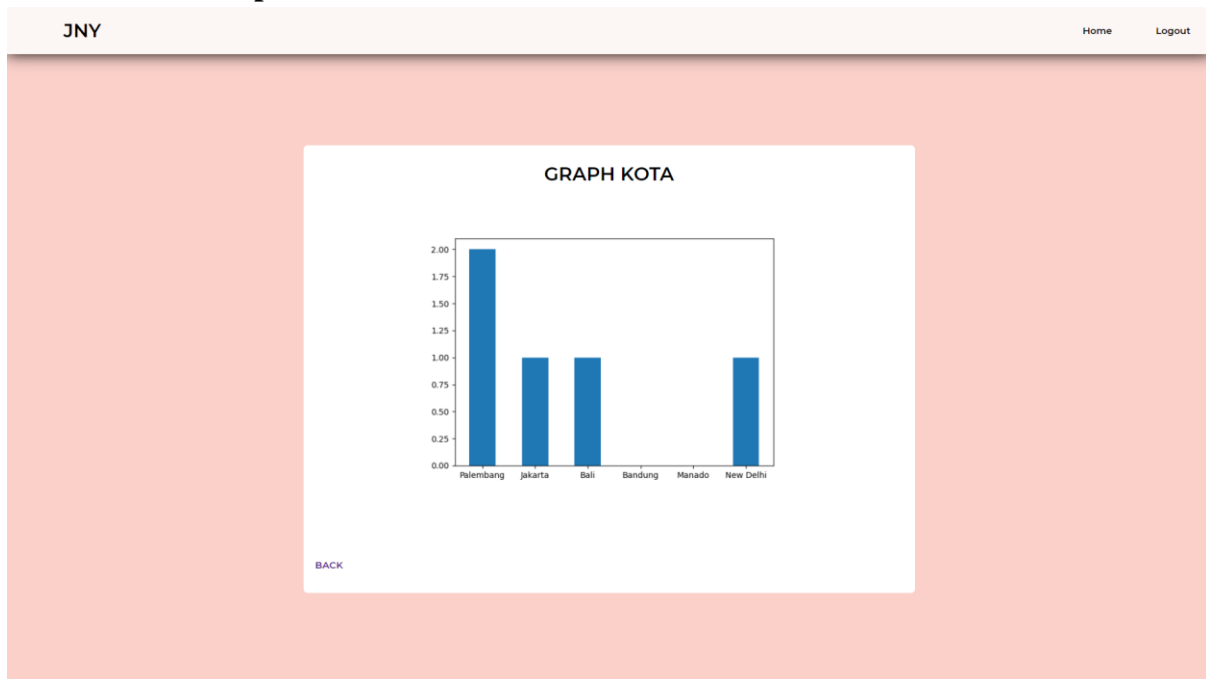
Pada kode program ini akan menampilkan semua kolom yang ada pada tabel kota ketika `id_kota={id}`. Kemudian untuk menampilkan banyak order tiap kota dengan sintaks `select count(resi) from order2 where id_kota={semua_id_kota}`.

### 3.2.6 Graphic



Pada Program ini terdapat show graphic. Graphic yang dapat ditampilkan yaitu graphic kota, graphic tipe dan graphic kurir.

#### 3.2.6.1 Graph Kota



```

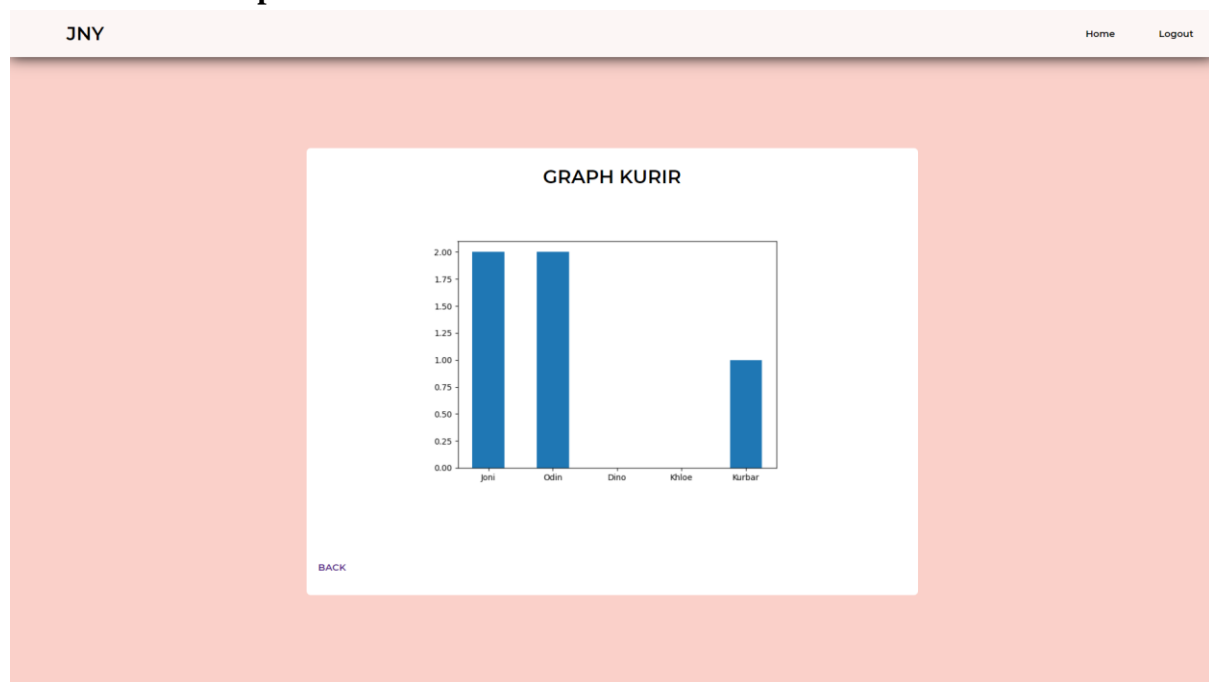
cur = mysql.connection.cursor()
cur.execute(f'select * from kota')
kota = cur.fetchall()
banyakOrder = []
for i in kota:
    cur.execute(f'select count(resi) from order2 where id_kota={i[0]}')
    temp = cur.fetchall()
    banyakOrder.append(temp[0][0])

cur.execute('select namaKota from kota')
basing = cur.fetchall()

```

Pada kode program ini digunakan untuk membuat graphic kota. Graphic kota memerlukan x= basing dan y= banyakorder. Untuk mendapatkan kota dapat dari sintaks select \* from kota. Kemudian banyakorder diambil dari count(resi) yang dihitung dari semua id\_kota yang ada. Setelah itu untuk mendapatkan variabel basing dapat dari select namaKota from kota

### 3.2.6.2 Graph Kurir



```

cur = mysql.connection.cursor()
cur.execute(f'select * from kurir')
kurir = cur.fetchall()
banyakOrder = []
for i in kurir:
    cur.execute(f'select count(resi) from order2 where nip={i[0]}')
    temp = cur.fetchall()
    banyakOrder.append(temp[0][0])

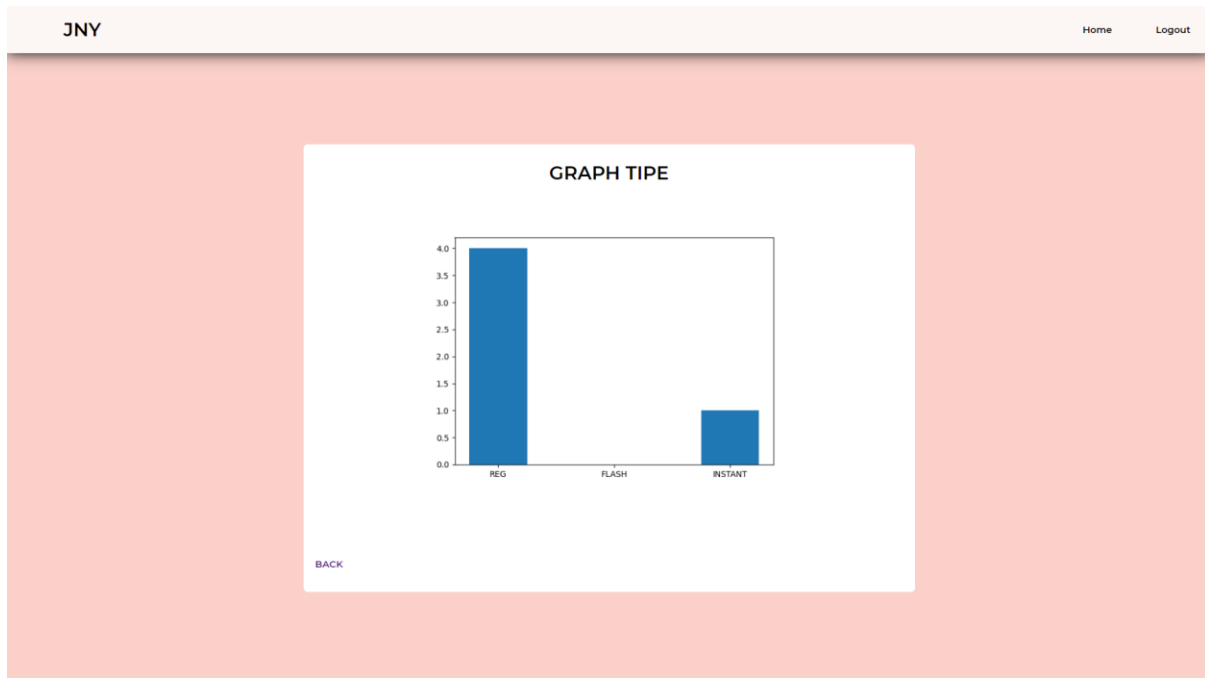
cur.execute('select namaKurir from kurir')
basing = cur.fetchall()

```

Pada kode program ini digunakan untuk membuat graphic kurir. Graphic kurir memerlukan x= basing dan y= banyakorder. Untuk mendapatkan kurir dapat dari sintaks select \* from kurir. Kemudian banyakorder diambil dari select count(resi) from order2 yang diambil

dari semua banyak nip yang ada. Setelah itu, untuk mendapatkan namaKurir didapatkan dari `select namaKurir from kurir`.

### 3.2.6.3 Graph Tipe



```
cur = mysql.connection.cursor()
cur.execute(f'select * from kurir')
kurir = cur.fetchall()
banyakOrder = []
for i in kurir:
    cur.execute(f'select count(resi) from order2 where nip={i[0]}')
    temp = cur.fetchall()
    banyakOrder.append(temp[0][0])
banyakOrder2 = []
cur.execute(f'select * from tipe')
tipe= cur.fetchall()
for i in tipe:
    banyaktemp = 0
    for j in range(0, len(kurir)):
        if(i[0]==kurir[j][5]):
            banyaktemp += banyakOrder[j]
    banyakOrder2.append(banyaktemp)

cur.execute('select namaTipe from tipe')
basing = cur.fetchall()
```

Pada kode program ini, digunakan untuk membuat graphic tipe. Graphic tipe memerlukan  $x = \text{graphtipe}$  dan  $y = \text{banyakOrder2}$ . Untuk mendapatkan banyakOrder2, perlu mendapatkan semua kolom pada tabel kurir terlebih dahulu. Setelah itu, dihitung banyak order yang dimiliki oleh setiap kurir yang nantinya akan disimpan ke dalam list banyakOrder. Setelah itu dilakukan *nested loop* untuk menjumlahkan semua banyakOrder dari kurir yang memiliki tipe pengiriman yang sama, untuk setiap jenis tipe yang tersedia, hasilnya akan disimpan ke dalam list banyakOrder2 yang menjadi *value Y* pada grafik tipe ini.

### 3.2.7 Delete

```
cur = mysql.connection.cursor()
cur.execute(f"DELETE FROM order2 WHERE resi='{id}'")
cur.connection.commit()
cur.close()
return redirect('/show/order')
```

```
cur = mysql.connection.cursor()
cur.execute(f'DELETE FROM kota WHERE id_kota={id}')
cur.connection.commit()
cur.close()
return redirect('/show/kota')
```

```
cur = mysql.connection.cursor()
cur.execute(f'DELETE FROM kurir WHERE nip={id}')
cur.connection.commit()
cur.close()
return redirect('/show/kurir')
```

```
cur = mysql.connection.cursor()
cur.execute(f'DELETE FROM tipe WHERE id_tipe={id}')
cur.connection.commit()
cur.close()
return redirect('/show/tipe')
```

Semua sintaks menggunakan delete from, disini tabel yang dapat dihapus yaitu order2, kota, kurir, tipe. Kemudian kondisi yang dimasukkan berdasarkan routing dari masing-masing nama kolom yang disimpan dalam variabel id.

## BAB IV

### KESIMPULAN DAN SARAN

#### 4.1. Kesimpulan

1. Rancangan basis data dari sistem ekspedisi barang terdiri dari entitas-entitas seperti *order*, *admin*, *status*, *kota*, *tipe*, dan *kurir*. Setiap entitas memiliki *primary key* dan atribut-atributnya.
2. Entitas yang terdapat pada sistem basis data ekspedisi barang memiliki hubungan-hubungan (contoh suatu *order* diantarkan oleh suatu *kurir*) yang dihubungkan melalui *foreign key*.
3. *Create, Read, Update, Delete (CRUD)* dapat dilakukan pada aplikasi dengan menghubungkan aplikasi tersebut dengan satu *database* untuk mencegah ketidakkonsistenan data pada basis data sistem ekspedisi barang.

#### 4.2. Saran

1. Penambahan notifikasi pada bagian *UI* ketika suatu aksi *CRUD* dilakukan dapat membantu pengguna untuk dapat lebih yakin jika aksi yang dilakukan telah disimpan ke dalam basis data.
2. Keamanan basis data perlu ditingkatkan lagi.



## DAFTAR PUSTAKA

Hunter, J., Dale, D., Firing, E., & Droettboom, M. (2021). Release 3.4.2.  
<https://matplotlib.org/stable/contents.html>

Ronacher, A. (2019). Flask Documentation. 293. <http://flask.pocoo.org/docs/0.10/>.  
<https://flask.palletsprojects.com/en/2.0.x/>

Manual, M. R. (2013). MySQL 5 . 0 Reference Manual. MySQL 5.0 Reference Manual, 1,  
1692. dev.mysql.com <https://dev.mysql.com/doc/refman/8.0/en/>

### LOG SHEET KEGIATAN

No	Tanggal	Nama	NIM	Kegiatan
1	16 April 2021	Aulia Rahman Zulfi Auriga Farhan R Fadhillah Azhar Alsani M.Daffa Massimiliano A Sayyid Muhammad Umar Al Haris	119140110 119140153 119140217 119140139 119140190	Diskusi Topik
2	18 April 2021	Aulia Rahman Zulfi Auriga Farhan R Fadhillah Azhar Alsani M.Daffa Massimiliano A Sayyid Muhammad Umar Al Haris	119140110 119140153 119140217 119140139 119140190	Membuat ERD
3	20 April 2021	Aulia Rahman Zulfi Auriga Farhan R Fadhillah Azhar Alsani M.Daffa Massimiliano A Sayyid Muhammad Umar Al Haris	119140110 119140153 119140217 119140139 119140190	Membuat Relational Model
4	22 April 2021	Aulia Rahman Zulfi Auriga Farhan R Fadhillah Azhar Alsani M.Daffa Massimiliano A Sayyid Muhammad Umar Al Haris	119140110 119140153 119140217 119140139 119140190	Merancang Database
5	27 April 2021	Sayyid Muhammad Umar Al Haris	119140190	Membuat Laporan BAB 1
6	30 April 2021	Aulia Rahman Zulfi Sayyid Muhammad Umar Al Haris	119140110 119140190	Membuat Routing
7	1 Mei 2021	Aulia Rahman Zulfi	119140110	Implementasi Database ke PHP
8	2 Mei 2021	Sayyid Muhammad Umar Al Haris	119140190	Penerapan Foreign Key Dalam Database
9	2 Mei 2021	Auriga Farhan R	119140153	Membuat Algoritma Login
10	3 Mei 2021	Fadhillah Azhar Alsani	119140217	Membuat Algoritma Menampilkan Data kurir dan Tipe Dari Database
11	4 Mei 2021	M.Daffa Massimiliano A	119140139	Membuat Algoritma Menampilkan Kota
12	5 Mei 2021	Aulia Rahman Zulfi	119140110	Membuat Algoritma Menampilkan Order
13	5 Mei 2021	Sayyid Muhammad Umar Al Haris	119140190	Membuat Fitur Edit dan Delete Data
14	6 Mei 2021	Aulia Rahman Zulfi	119140110	Membuat Grafik Kurir, Kota, Tipe

15	6 Mei 2021	Aulia Rahman Zulfi Sayyid Muhammad Umar Al Haris	119140110 119140190	Debugging Backend
16	7 Mei 2021	M.Daffa Massimiliano A	119140139	Membuat Algoritma Logout
17	7 Mei 2021	Aulia Rahman Zulfi	119140110	Membuat Algoritma Pengecekan User yang Aktif
18	8 Mei 2021	Sayyid Muhammad Umar Al Haris	119140190	
19	8 Mei 2021	Aulia Rahman Zulfi Auriga Farhan R Fadhillah Azhar Alsani M.Daffa Massimiliano A Sayyid Muhammad Umar Al Haris	119140110 119140153 119140217 119140139 119140190	Diskusi Desain UI
20	8 Mei 2021	Auriga Farhan R Fadhillah Azhar Alsani	119140153 119140217	Merancang Desain UI
21	9 Mei 2021	Auriga Farhan R	119140153	Membuat UI Login dan Menu
22	9 Mei 2021	Fadhillah Azhar Alsani	119140217	Membuat UI Input Order
23	10 Mei 2021	Auriga Farhan R	119140153	Membuat UI Tabel Order, Kurir, Tipe, Kota
24	10 Mei 2021	Fadhillah Azhar Alsani	119140217	Membuat UI Menu Grafik dan Tampilan Grafik Kota, Tipe, Kurir
25	13 Mei 2021	Fadhillah Azhar Alsani	119140217	Membuat UI Input Kota dan Kurir
26	13 Mei 2021	Auriga Farhan R	119140153	Membuat UI Edit, Order, Kurir, Kota
27	15 Mei 2021	M.Daffa Massimiliano A	119140139	Testing Fungsionalitas UI
28	16 Mei 2021	Auriga Farhan R Fadhillah Azhar Alsani	119140153 119140217	Membuat Laporan BAB 2
29	16 Mei 2021	Aulia Rahman Zulfi Sayyid Muhammad Umar Al Haris	119140110 119140190	Membuat Laporan BAB 3
30	17 Mei 2021	M.Daffa Massimiliano A	119140139	Membuat Lampiran Laporan
31	18 Mei 2021	Aulia Rahman Zulfi Auriga Farhan R Fadhillah Azhar Alsani M.Daffa Massimiliano A Sayyid Muhammad Umar Al Haris	119140110 119140153 119140217 119140139 119140190	Membuat Kesimpulan dan Finalisasi Laporan

## BIODATA DAN LINGKUP KERJA ANGGOTA

Nama : Sayyid Muhammad Umar Al Haris  
NIM : 119140190  
Email : sayyid.119140190@student.itera.ac.id  
Lingkup Kerja : *Project manager, back end main, debugging, edit dan hapus data*

Nama : Aulia Rahman Zulfi  
NIM : 119140110  
Email : aulia.119140110@student.itera.ac.id  
Lingkup Kerja : *Back end lead, main routes, data graph*

Nama : Auriga Farhan Rizkiandy  
NIM : 119140153  
Email : auriga.119140153@student.itera.ac.id  
Lingkup Kerja : *Front end lead*

Nama : Fadhillah Azhar Alsani  
NIM : 119140217  
Email : fadhillah.119140217@student.itera.ac.id  
Lingkup Kerja : *Front end main*

Nama : M. Daffa Massimiliano A  
NIM : 119140139  
Email : muhammad.119140139@student.itera.ac.id  
Lingkup Kerja : *Desain, mapping, tester*