

[Pengiriman OKE]

Project Matakuliah Pemrograman Web

Program Studi Teknik Informatika, Genap 2016/2017

Diampu oleh:

Aryo Pinandito, ST, M.MT

Disusun oleh:

1. Guidita Suryani - 155150201111233 [DIT]
2. I Made Narendra Putra - 155150201111328 [NRN]
3. Salma Hanifah - 155150200111196 [SLM]
4. Galih Bhaktiar Candra - 155150207111182 [GBC]

Table of Contents

Latar Belakang.....	3
Analisis Kebutuhan.....	5
Desain dan Implementasi	6
Diagram Use Case	6
[Cek Tarif] [DIT]	7
Sequence Diagram Use Case Cek Tarif [DIT]	8
Class Diagram [Cek Tarif] [DIT].....	9
Rancangan Screen Flow [Cek Tarif] [DIT]	10
Rancangan Basis Data [Cek Tarif] [DIT]	11
Implementasi	12
[Lacak Kiriman Barang][NRN].....	26
Use Case Scenario [Lacak Kiriman Barang][NRN]	26
Sequence Diagram [Lacak Kiriman Barang][NRN]	26
Class Diagram [Lacak Kiriman Barang][NRN]	26
Rancangan Screen Flow [Lacak Kiriman Barang][NRN].....	27
Rancangan Basis Data [Lacak Kiriman Barang][NRN].....	27
Implementasi [Lacak Kiriman Barang][NRN].....	28
[Lihat Data][SLM]	36
Sequence Diagram [Lihat Data][SLM]	37
Rancangan Screen Flow [Lihat data][SLM]	37
Rancangan Basis Data [Lihat Data][SLM]	39
Implementasi	39
[Tambah Data][GBC]	45
Sequence Diagram [Tambah Data][GBC]	46
Class Diagram [Tambah Data][GBC].....	47
Rancangan Screen Flow [Tambah Data][GBC]	47
Rancangan Basis Data [Tambah Data][GBC]	50
Implementasi	51

Latar Belakang

Seiring dengan perkembangan zaman dan dengan adanya perkembangan ilmu pengetahuan dan teknologi yang pesat, kebutuhan manusia akan mengirimkan barang dari satu tempat ke tempat lain semakin tinggi. Dampak dari kebutuhan tersebut adalah semakin banyak bermunculannya jasa pengiriman barang dan kompetisi antara penyedia jasa tersebut. Setiap orang maupun perusahaan dapat mengirimkan barang dengan mudah dengan menggunakan jasa pengiriman barang. Banyak perusahaan besar yang menggunakan jasa pengiriman barang untuk mengirimkan barang ataupun dokumen, baik dalam jumlah yang kecil maupun besar. Kepercayaan terhadap penyedia jasa pengiriman barang telah membuat perkembangan yang cukup pesat dalam bidang jasa pengiriman barang. Persaingan yang timbul menyebabkan setiap perusahaan meningkatkan pelayanan mereka terhadap para pelanggan yang kini sebagaimana diketahuinya sudah sangat terbuka dengan adanya jaringan-jaringan internet sebagai sumber informasi.

Pengiriman barang dengan menggunakan penyedia jasa pengiriman barang telah memberikan kemudahan, akan tetapi kerap terjadi permasalahan salah satunya adalah Pelanggan membutuhkan informasi tentang biaya yang dibutuhkan untuk mengirim barang secara mudah dan cepat tanpa harus mendatangi kantor penyedia jasa pengiriman barang yang memakan waktu dan energi mereka. Informasi mengenai harga merupakan salah satu hal yang dibutuhkan bagi seseorang ketika ingin melakukan pengiriman barang menggunakan jasa pengiriman barang tertentu, agar terjadinya kesepakatan harga antara pengguna dan penyedia jasa. Maka untuk mengatasi permasalahan tersebut, diperlukan adanya rancangan sebuah web yang dinamis agar informasi harga yang ditawarkan dapat tersampaikan dengan baik, mudah dan cepat yang dilengkapi salah satu fitur dengan sebutan cek tarif untuk membangun suatu sistem yang mampu mendistribusikan data dan informasi secara sistematis dan teratur sehingga bekerja lebih efisien serta memberikan kemudahan kepada pelanggan untuk mengecek harga yang diperlukan untuk mengirimkan barang dari satu lokasi ke lokasi yang lainnya karena dapat diakses oleh semua orang dapat dilakukan di pc(personal computer) maupun pada gadget kapanpun dan dimanapun.[DIT]

Dalam suatu sistem ekspedisi, keberadaan fitur untuk melacak barang yang sudah dikirim sangatlah vital. Dengan fitur ini user dapat melacak barang dengan menggunakan nomor resi yang telah didapatkan sebelumnya saat melakukan proses pengiriman/submit barang di branch terkait. Pelacakan ini bersifat realtime dan user hanya perlu menginputkan nomor resi dan sebagai feedback user akan mendapatkan informasi dan status pengiriman barang.[NRN]

Dalam pembuatan sistem ini dibutuhkan suatu fungsi yang dapat menampilkan seluruh data pengiriman barang yang sebelumnya telah disimpan didalam database. Dengan fungsi ini, seorang admin perlu mengetahui keseluruhan rekaman data sehingga mampu untuk melihat data, menambahkan data, mengubah data jika terdapat kesalahan dari masukan yang diberikan sebelumnya, dan menghapus data pengiriman barang. Data pengiriman barang dapat dilihat secara langsung, kapanpun, dan dimanapun setelah admin melakukan login. Alhasil, admin mendapatkan informasi mengenai keseluruhan data pengiriman barang.[SLM]

Pencatatan pesanan secara manual dinilai tidak efisien dan juga membuang-buang kertas, kendala dalam estetika tulisan juga mempengaruhi orang yang membacanya, dan data yang sangat penting seperti nama , alamat, berat,dan nomor telepon berbahaya jika salah terbaca. Dengan aplikasi menambah data untuk mencatat pesanan informasi lebih mudah dibaca, data-data pemesanan dapat dicatat dan disimpan dalam database. [GBC]

Analisis Kebutuhan

Kebutuhan Fungsional:

1. Sistem harus menyediakan fungsi cek tarif dari Kota Asal ke Kota Tujuan
 - 1.1 Fungsi ini dapat diakses oleh user.
 - 1.2 Fungsi ini menampilkan inputan berupa kota asal, kotatujuan , dan berat.
 - 1.2.1 Tampilan pada inputan kota asal dan kota tujuan berupa dropdown yang berisikan lokasi yang datanya sesuai dengan yang ada pada database.
 - 1.2.2 Tampilan pada inputan berat berupa angka.
 - 1.3 Fungsi ini menampilkan tombol input dengan nama cek.
 - 1.4 Fungsi ini menampilkan halaman yang berisikan tabel tarif pengiriman.
 - 1.4.1 Tampilan tabel pengiriman berupa kota asal, kota tujuan, dan tarif
 - 1.4.2 Pada Tampilan kolom tarif melakukan perhitungan tarif berdasarkan data harga yang ada pada database dikalikan dengan inputan berat. [DIT]
2. Sistem harus menyediakan fungsi lacak kiriman
 - 2.1 Fungsi ini dapat diakses oleh user.
 - 2.2 Fungsi ini menerima inputan berupa no resi
 - 2.3 Fungsi ini menampilkan data barang dan status pengiriman. [NRN]
3. Sistem harus menyediakan fungsi lihat data barang
 - 3.1 Fungsi ini dapat diakses oleh admin.
 - 3.2 Fungsi ini menyediakan edit dan delete [SLM]
4. Sistem harus dapat menyediakan fungsi menambah data pengiriman
 - 4.1 Data pengiriman berupa nomor resi, nama pengirim, alamat pengirim, nomor telepon pengirim, nama penerima, alamat penerima, berat barang, status, kota asal dan kota tujuan bagi admin. [GBC]
5. Sistem harus memiliki fungsi login untuk admin
6. Sistem harus memiliki fungsi logout untuk admin

Kebutuhan Non-Fungsional

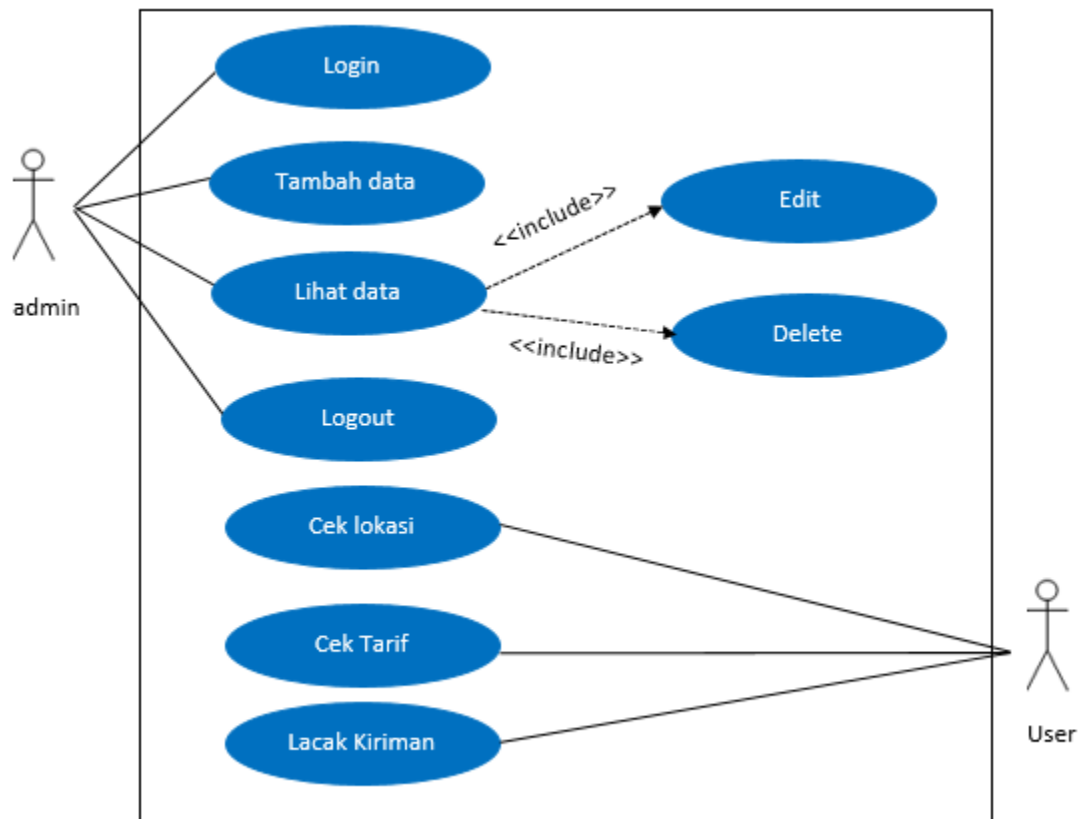
1. Sistem harus terhubung dengan suatu database

Kebutuhan Utility

1. Sistem dapat berjalan pada aplikasi browser (berbasis web).

Desain dan Implementasi

Diagram Use Case



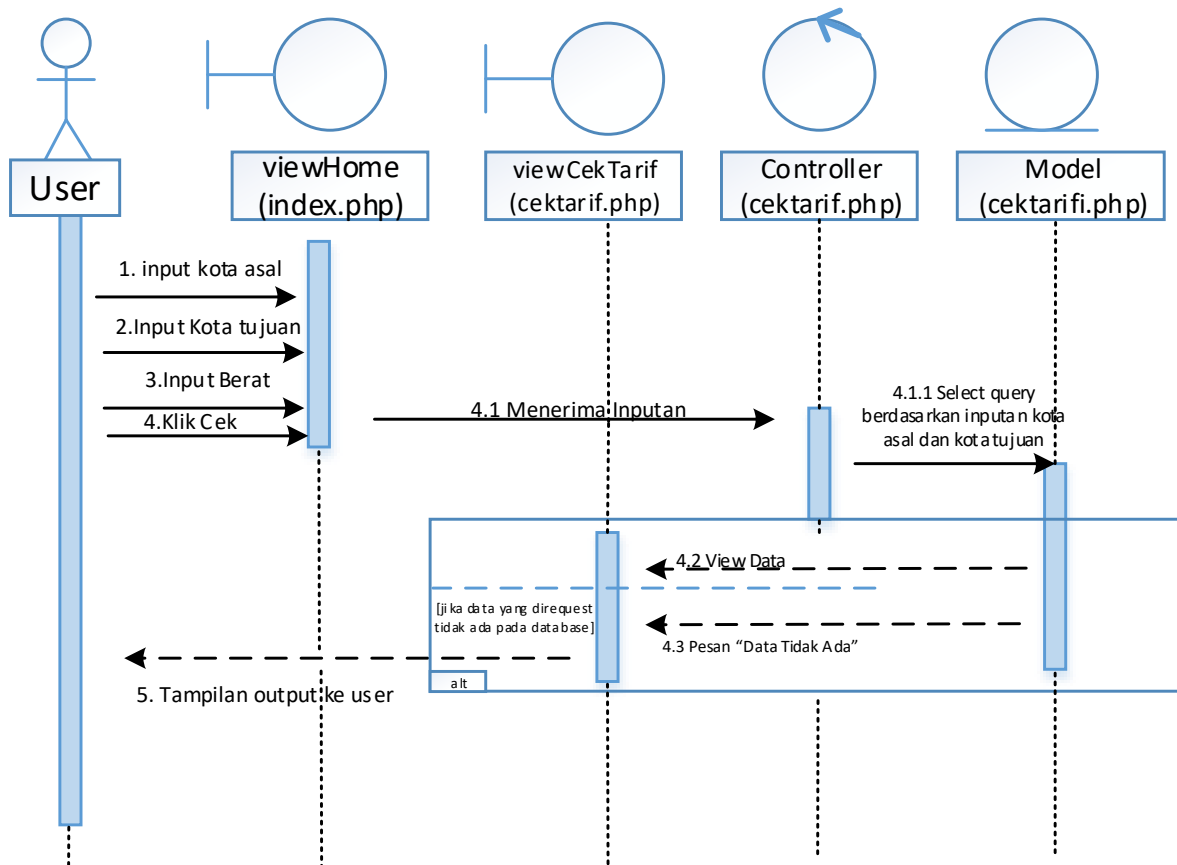
[Cek Tarif] [DIT]

Deskripsi use case : Use case Cek Tarif merupakan perilaku sistem dari tampak luar yang menyediakan fungsi untuk melihat data tarif berdasarkan inputan kota asal dan kota tujuan yang sebelumnya telah disimpan di dalam database bagi user sebagai aktor. Dan juga merupakan kebutuhan sistem terhadap fungsi cek tarif. Deskripsi use case cek tarif yang akan dirancang, diimplementasikan dalam sistem:

Proses / Komponen	Deskripsi	
Nama Use Case	Cek Tarif	
Deskripsi	Proses ini terjadi pada saat User mengakses website. User mengisi semua form input dan button yang tersedia dan menunggu data yang akan diberikan.	
Tujuan	Untuk Menampilkan Data Cek Tarif dari database sesuai dengan inputan User.	
Objective	Mengijinkan User untuk cek tarif.	
Actor	User	
Pre Condition	User mengisikan form input kota asal, kota tujuan, dan berat	
Action	Actor Action	System Action
Main flow	1.User mengakses halaman index	2.Sistem menampilkan halaman index.php
	3.User mengisikan inputan kota asal	
	4.User mengisikan inputan kota tujuan	
	5. User mengklik tombol cek	6.Sistem mengambil data dari database dan melakukan perhitungan berat dikali harga untuk mendapatkan nilai tarif.
		7.Sistem menampilkan halaman cektarif.php
Alternative flow	Jika Data yang diminta tidak ada pada database maka akan mengarah ke halaman cektarif.php namun, menampilkan tulisan "Data Belum Ada!"	

<i>Post condition</i>	Data yang ditampilkan data yang ada dalam database dan sesuai dengan inputan user.
-----------------------	--

Sequence Diagram Use Case Cek Tarif [DIT]



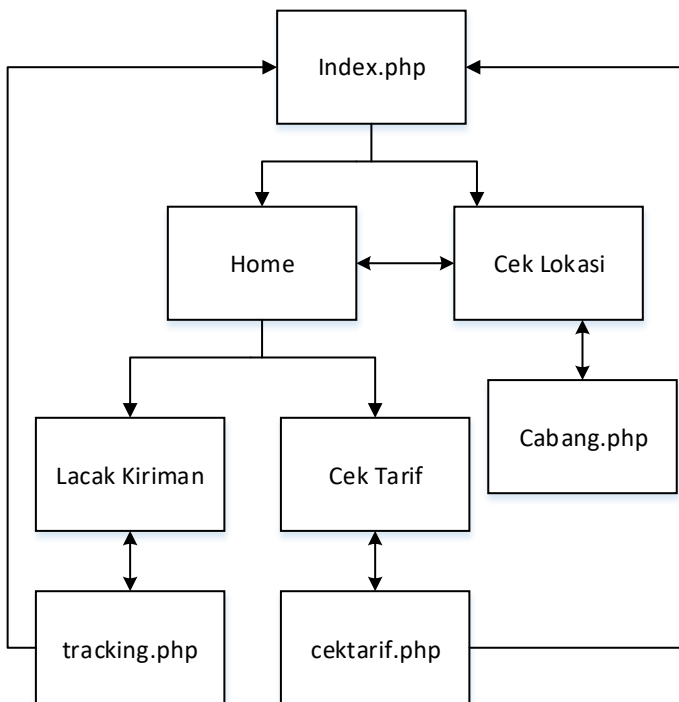
Class Diagram [Cek Tarif] [DIT]

Sistem yang dibangun adalah terstruktur sehingga tidak terdapat class diagram. Karena pada php terstruktur tidak ada class yang dipisah-pisah seperti pada Object Oriented.

Untuk diagram filenya:



Struktur Navigasi(Work Breakdown Structure)



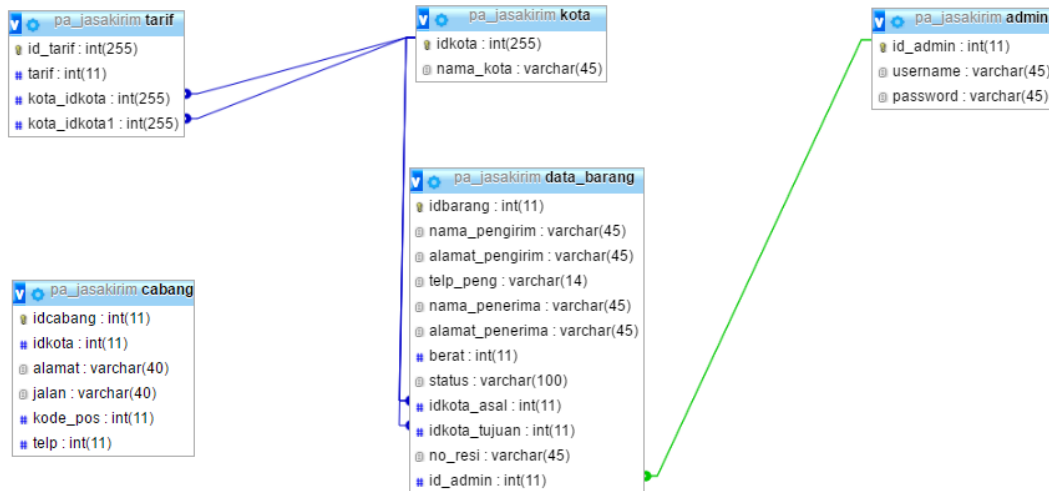
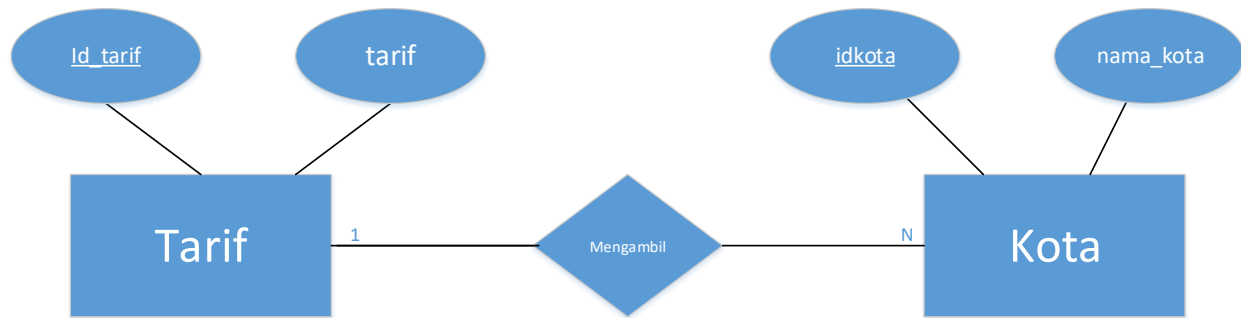
Rancangan Screen Flow [Cek Tarif] [DIT]

Gambar rancangan :

Tampilan awal untuk mengakses Use Case Cek Tarif:



Rancangan Basis Data [Cek Tarif] [DIT]



tarif

No	Nama Kolom	Tipe Data	Panjang Data	Contoh Data ¹
1	<i>Id_tarif (PK)</i>	INT	255	1
2	<i>tarif</i>	INT	11	3000
3	<i>kota_idkota (FK)</i>	INT	255	1
4	<i>kota_idkota1 (FK)</i>	INT	255	2

Nama tabel : tarif

Deskripsi : tabel tarif ini berisi informasi tarif pengiriman barang dari kota asal ke kota tujuan

Primary Key : id_tarif

Foreign key :kota_idkota
kota_idkota1

¹ Tuliskan minimal 2 contoh nilai data yang berbeda.

kota

No	Nama Kolom	Tipe Data	Panjang Data	Contoh Data
1	Idkota (PK)	INT	255	1
2	Nama_kota	VARCHAR	45	Ponorogo

Nama tabel : kota

Deskripsi : tabel tarif ini berisi informasi nama kota yang diberikan idnya sehingga pada saat tabel tarif mengakses primary key kota_idkota (untuk kota asal) dan kota_idkota2(untuk kota tujuan) bisa menampilkan nama kotanya sesuai dengan id yang sudah diberikan.

Primary Key : idkota

Implementasi

Implementasi PHP

Karena pemogramannya merupakan terstruktur sehingga tidak terdapat kelas ataupun method, begitu pula dengan jalannya program secara terstruktur hanya dari satu modul saja sehingga model dan controller menjadi satu dalam struktur program cektarif.php , dan bagian program yang menjadi controller adalah bagian yang mengatur jalannya data yaitu kode php yang mengatur data yang direquest saat di view pada index.php dan mengontrol mengambil data yang ada pada modelnya dari database. Untuk pengarahan tampilan ditentukan pada action form.

Kode program yang berfungsi sebagai controller yang mengatur(mengontrol) jalannya data atau memisahkan data-data yang masuk.

```
<?php
```

```
$kota_asal = $_POST['kota_idkota'];  
  
$kota_tujuan = $_POST['kota_idkota1'];  
  
$berat = $_POST['berat'];  
  
require_once('koneksi.php');
```

Kode php diatas adalah awalnya mengambil semua data dari method POST dan disimpan di dalam variabel yang dipisah-pisah berdasarkan inputan yaitu value kota_idkota disimpan dalam variabel \$kota_asal , value kota_idkota1 disimpan dalam variabel \$kota_tujuan dan berat disimpan dalam variabel \$berat. Untuk mendapatkan datanya Php membutuhkan koneksi ke database.

Bagian yang berfungsi sebagai “Model” yang bertindak sebagai Entity yang berurusan dengan data, mengambil data-data yang diperlukan dari database.

```
//untuk mencari nilai variabel harga

$sql_tarif = "SELECT * FROM tarif WHERE kota_idkota = '$kota_asal' AND kota_idkota1 = '$kota_tujuan' ";

if ($result = $conn->query($sql_tarif)) {
    while ($row = $result->fetch_row()) {
        $tarif = $row[1];
    }
}

$harga = $tarif * $berat;

//data yang diambil dari database

$sql = "SELECT id_tarif, tarif, dariKota.nama_kota as 'Dari Kota', tujuanKota.nama_kota as 'Tujuan Kota' FROM tarif LEFT JOIN kota AS dariKota ON dariKota.idkota = tarif.kota_idkota LEFT JOIN kota AS tujuanKota ON tujuanKota.idkota = tarif.kota_idkota1 WHERE kota_idkota = '$kota_asal' AND kota_idkota1 = '$kota_tujuan'";

$result = $conn->query($sql);
```

Untuk mencari nilai variabel harga dibutuhkan perintah query untuk memilih semua data dari tabel tarif dimana nilai kota_idkota sama dengan nilai variabel kota_idkota, kota_idkota1 sama dengan nilai variabel kota_idkota1 yang ada dari method post. Selanjutnya cara menampilkan data dari database MYSQL Dengan mysqli object style pada perintah if, dan pada perulangan while dengan argumen method fetch_row() dari objek \$result yang merupakan objek hasil pemanggilan method query(). Method fetch_row() ini tidak memerlukan argumen apapun, sehingga memanggilnya dengan cara \$result->fetch_row(). Sehingga langsung dapat diset nilai variabel tarif dengan mengambil index ke 1 dari row yaitu mengambil nilai tarif pada tabel tarif. Sehingga untuk mendapatkan variabel harga bisa langsung ditentukan dari nilai variabel tarif yang didapatkan dikalikan dengan nilai variabel berat yang diinputkan dan diambil dari method POST. Dan karena kedua tabel kota dan tarif berelasi dan untuk menampilkan nama kota bukan idnya diperlukan perintah query tabel tarif yang diambil nilai seluruh kolomnya dan diberikan inisial dari pengambilan foreign key idnya berupa Dari Kota dan Tujuan kota untuk memudahkan penamaanya saja yang dilakukan Left Join dengan tabel kota, dimana nilai dari kolom kota_idkota sama dengan variabel kota_asal dan kota_idkota1 sama dengan variabel tujuan yang maksudnya mengecek dari id yang sama. Dan query tersebut akan disimpan dalam variabel \$result

```
//data yang akan tampil pada view cektarif.php

    if ($result-> num_rows > 0) {

        while ($row = $result->fetch_assoc()) {

            <?php echo $row['Dari Kota'];?>

            <?php echo $row['Tujuan Kota'];?>

            <?php echo "Rp. ".$harga.".-";?>

        }

    } else {

        echo "Data Belum Ada!";

    }

?>
```

Selanjutnya nilai variabel \$result yang didapatkan pada kode program yang berindak sebagai model akan diformat untuk ditampilkan pada view output cektarif.php, Dengan fungsi fetch_assoc() mengambil data dari yang didapatkan pada variabel sql untuk menampilkan data dari kota, tujuan kota dan harganya dan bila datanya tidak ada maka akan menampilkan “Data Belum Ada!”.

Untuk tampilan view input pada index.php juga dilampirkan kode php karena tampilannya dibuat dinamis agar apabila terdapat perubahan nilai datanya pada database maka pada tampilan akan berubah juga.

```
<?php

$conn=mysqli_connect("localhost","root","","pa_jasakirim");

$sql_asal = "SELECT * FROM kota ";

$id1="SELECT * FROM kota";

$result = $conn->query($sql_asal);

foreach ($result as $key => $value) {

    ?><option value="<?php echo $value['idkota']?>"><?php
    echo $value["nama_kota"] ?></option><?php

    var_dump($value);

}

?>

</select>

</div>

</div>
```

```
<span class="glyphicon glyphicon-arrow-right"></span>
<div class="form-group">
    <div class="input-group">
        <select class="form-control" id="kota_idkota1" name="kota_idkota1"
required>

<?php
$conn=mysqli_connect("localhost","root","","pa_jasakirim");
$sql_asal = "SELECT * FROM kota ";
$id2="SELECT * FROM kota";
$result = $conn->query($sql_asal);
foreach ($result as $key => $value) {
    ?><option value="<?php echo $value['idkota']?>"><?php echo
$value["nama_kota"] ?></option><?php
    }
?>
```

Data-data yang diperlukan pada cek tarif adalah data nilai dari idkota dan nama kota data tabel kota, kode php diatas mengambil data-data nama dari tabel kota dan menyimpannya dalam variabel result dan menampilkan datanya berdasarkan nama_kota namun data yang akan diambil dan dikirimkan nanti adalah data berdasarkan idnya. Terdapat 2 kode php yang perintahnya sama karena pada input dibutuhkan 2 data yaitu data untuk kota asal dan kota tujuan yang mengambil nilai yang sama pada tabel kota di database.

Implementasi HTML5

Halaman input yang ada dari screenflow yaitu pada index.php. Berikut kerangka/susunan layout dari file HTML yang digunakan untuk input cektarif

```
<!DOCTYPE html>
<html>
  <head> Pengiriman Oke </head>
  <body>

<!-- cek tarif -->

<div class="panel panel-primary " style="margin-top: -140px; margin-left: 55%; width: 35%; height:50%">

<div class="panel-heading">

<h3 class="panel-title">Cek Tarif</h3>
```

```

</div>

<div class="panel-body" style="width: 100%">
<form class="form-inline" action="cektarif.php" method="post">
    <div class="form-group">
        <div class="input-group" style="width:100%">
            <select class="form-control" id="kota_idkota" name="kota_idkota"
            required >
                <option value=" ">
            </select>
        </div>
    </div>
    <div class="form-group">
        <div class="input-group">
            <select class="form-control" id="kota_idkota1"
            name="kota_idkota1" required>
                <option value=" ">
            </select>
        </div>
    </div>
    <div class="input-group">
        <input type="number" class="form-control" name="berat"
        placeholder="kg" style="width:70px" required>
    </div>
    <div class="input-group">
        <label class="control-label col-sm-2" for="save-button"></label>
        <div>
            <button class="btn btn-success" name="save-
            button">Cek</button>
        </div>
    </div>
</div>
</form>
</div>

```



```

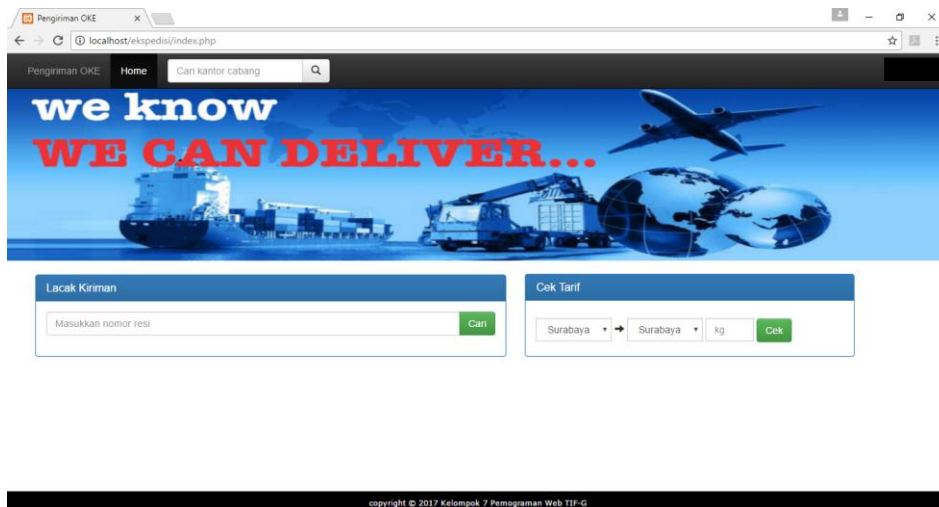
    </div>
</body>
</html>

```

Penjelasan: Kerangka diatas hanya digunakan untuk menampilkan layout input pada halaman index.php untuk membuat form input cek tarif saya menggunakan class-class yang ada pada bootstrap, untuk tampilan inputnya berada pada class panel panel-primary yang nantinya akan menampilkan kotak, dan untuk titlenya diberi tulisan cektarif, selajutnya pada isinya diberi 2(satu untuk kota asal, dan satu untuk kota tujuan) form-group dan input group untuk menampilkan inputan berupa dropdown namun untuk kedua option value yaitu pilihan pada dropdown ini tidak saya lampirkan karena saya mengambil datanya menggunakan perintah php untuk mengambil data yang ada pada database, selanjutnya diberikan glyphicon berupa panah ke kanan diantara dua form-group tersebut. Dan diberikan pula sebuah input dengan tipe number(angka) untuk berat dan diberikan placeholder “kg” untuk mejelaskan angka tersebut adalah berat perkg. Selanjutnya diberikan button disebelahnya dengan tulisan Cek. Form ini akan tampil sebaris dikarenakan diberikan format class pada bootstrap form-inline dan karena ini terstruktur aksi yang akan tampil setelah input button di klik akan menuju file cektarif.php dengan method yang digunakan adalah method post jadi dikirimkan ke halaman isi(body) dan tidak akan tampil pada URL. Tag-tag div diatas hanya digunakan untuk untuk membuat layer dengan cari memberi atribut id atau class pada tag div tersebut, yang akan memudahkan untuk membuat layout sesuai dengan design yang di inginkan.

Screenshot halaman input:

The screenshot shows a web browser window with the URL 'localhost/ekspedisi/index.php'. The website has a header with 'Pengiriman OKE', 'Home', and a search bar. Below the header is a large banner with the text 'we know WE CAN DELIVER...' and images of a ship, a truck, and a globe. The main content area contains two forms: 'Lacak Kiriman' (Track Shipment) and 'Cek Tarif' (Check Rate). The 'Lacak Kiriman' form has a text input field labeled 'Masukkan nomor resi' and a green 'Cari' button. The 'Cek Tarif' form has two dropdown menus for 'Malang' and 'Ponorogo', a text input field for weight with a placeholder '5', and a green 'Cek' button. The footer contains the text 'copyright © 2017 Kelompok 7 Pemrograman Web TIF-G'.



Bagian yang saya jelaskan hanya kerangka html 5 untuk menampilkan kotak cek tarif saja

Halaman output yang akan menampilkan data yang diterima dari input dan diproses sehingga akan menampilkan data yang sesuai dengan yang diinputkan yang ada dari screenflow ceftarif.php . Berikut kerangka/susunan layout dari file HTML yang digunakan untuk outputnya:

```
<!DOCTYPE html>
<html>
  <head> cek tarif</head>
  <body>

    <!-- NAVBAR -->
    <nav class="navbar navbar-inverse navbar-fixed-top">
      <div class="container-fluid">
        <div class="navbar-header">
          <p class="navbar-text">Pengiriman OKE </p>
        </div>
        <ul class="nav navbar-nav">
          <li><a href="index.php"
            onclick="javascript()">Home</a></li>
        </ul>

        <form class="navbar-form navbar-left" action="cabang.php"
        method="post">
          <div class="input-group">
            <input type="text" class="form-control" placeholder="Cari kantor
            cabang " name="cabang">
            <div class="input-group-btn">
```

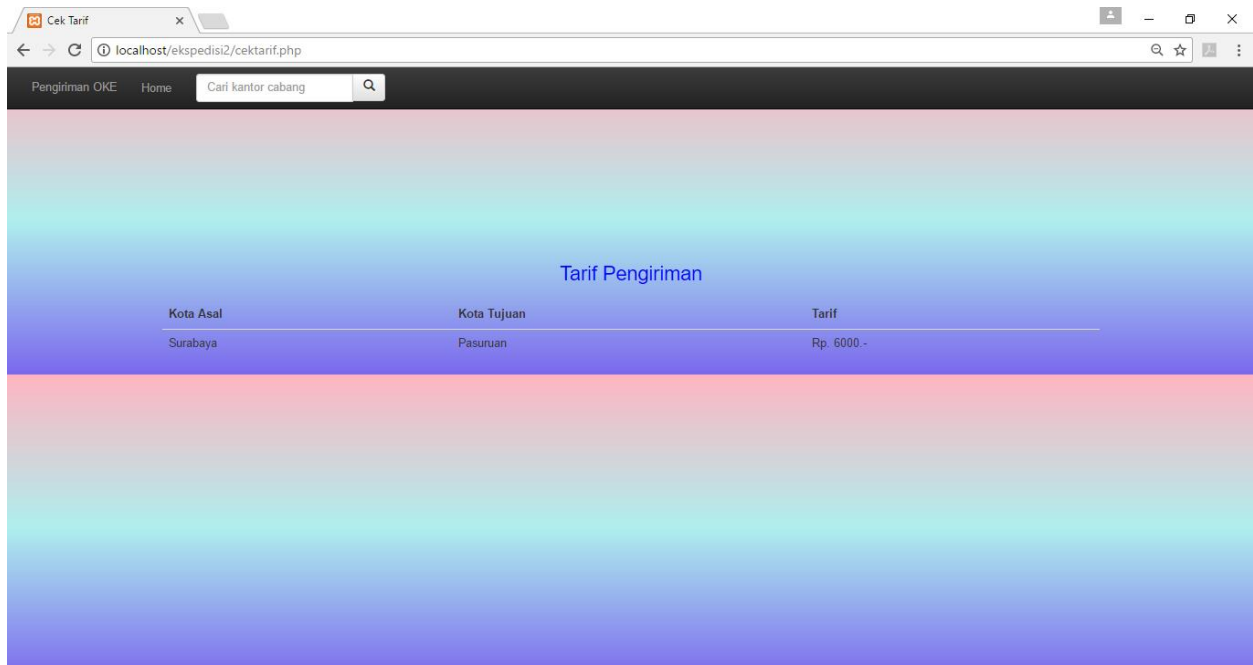
```

        <button class="btn btn-default" type="submit">
            <i class="glyphicon glyphicon-search"></i>
        </button>
    </div>
</div>
</div>
</form>
</div>
</nav>
<div class="container">
    <h3 style="margin-top: 200px;text-align: center;color:blue">Tarif Pengiriman</h3>
    <table class="table table-hover" style="margin-top: 20px;">
        <thead>
            <th>Kota Asal</th>
            <th>Kota Tujuan</th>
            <th>Tarif </th>
        </thead>
        <tbody>
            <tr>
                <td><?php echo $row['Dari Kota'];?></td>
                <td><?php echo $row['Tujuan Kota'];?></td>
                <td><?php echo "Rp. ".$harga.".-";?></td>
            </tr>
        </tbody>
    </table>
</div>
</html>

```

Kerangka html diatas hanya akan menampilkan output dari cektarif yang berada pada file cektarif.php sama juga dengan sebelumnya ini menggunakan bootstrap dan memakai class container dan dibuat sebuah tabel yang diambil dari class table table-hover pada bootstrap, yang pada tabel tersebut di heading(judul) tabelnya diberi Kota Asal, Kota Tujuan, Tarif, selanjutnya pada body(isinya) diisi nilai-nilai dari mengambil data menggunakan php ke dalam kolom-kolom tersebut.

Screenshot halaman output:



Implementasi CSS3

Pada implementasi CSS hanya sedikit yang digunakan karena saya menggunakan style bootstrap yang sudah menyediakan tampilan-tampilan yang bagus.

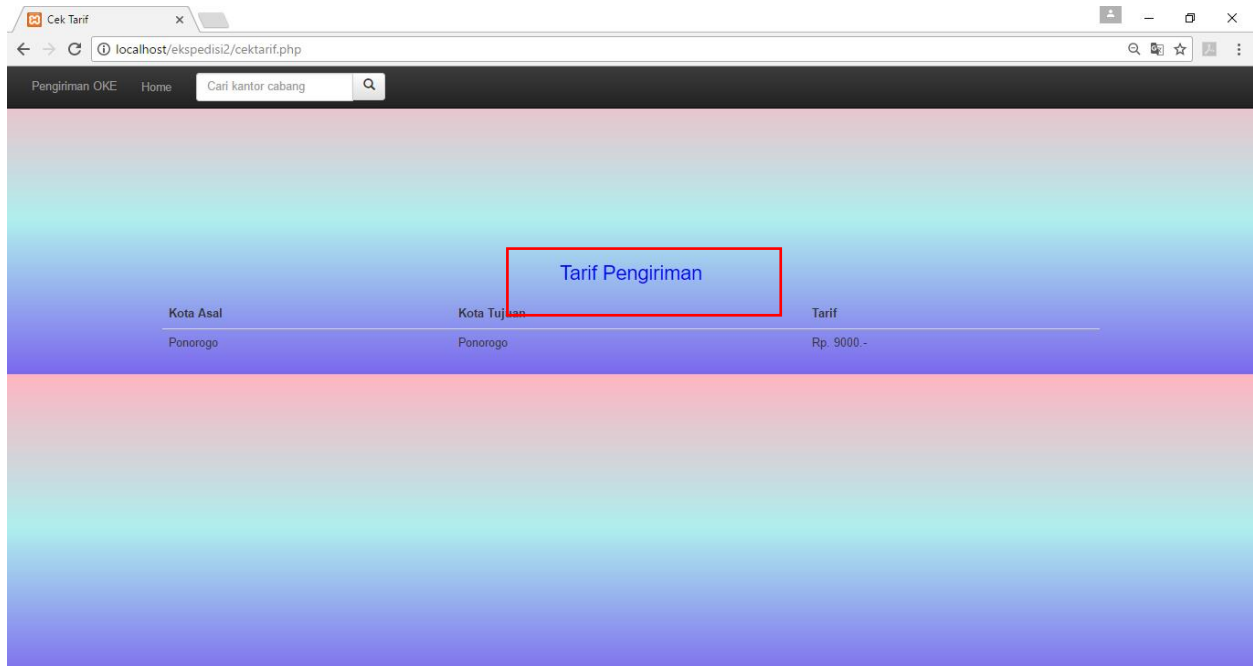
Ada 2 cara yang saya gunakan untuk menerapkan css yaitu:

1. Inline style, yaitu berada pada tag(satu elemen HTML) tertentu yang diberi style css

```
<h3 style="margin-top: 200px;text-align: center;color:blue">Tarif Pengiriman</h3>
```

Saya meletakkan style css pada tulisan “Tarif Pengiriman” yang diberikan heading h3, style yang diberikan adalah margin-top untuk mengatur jarak margin atasnya sebesar 200px dampaknya tulisan “Tarif Pengiriman” akan turun(memiliki margin atas) sebesar 200px, text-align:center untuk mengarahkan tulisan menjadi berada di posisi tengah. Dan color:blue untuk merubah warna tulisan menjadi biru.

Sehingga efeknya tulisan tersebut akan berada ditengah dengan posisi berada setelah margin 200px dan warnanya biru.



2. Embedded style/Internal Styles, di letakkan di dalam elemen(tag) head , yang diberi elemen(tag) style

```
<style type="text/css">
    .form-inline {
        margin-top: 10px;
        margin-bottom: 5px;
    }
</style>
```

Memberikan efek ukuran margin atas sebesar 10px dan bawah sebesar 5 px saja terhadap kelas form-inline



*bagian yang diberi tanda kotak merah yang diimplementasikan css didalamnya.

Serta:

```
<style type="text/css">
```

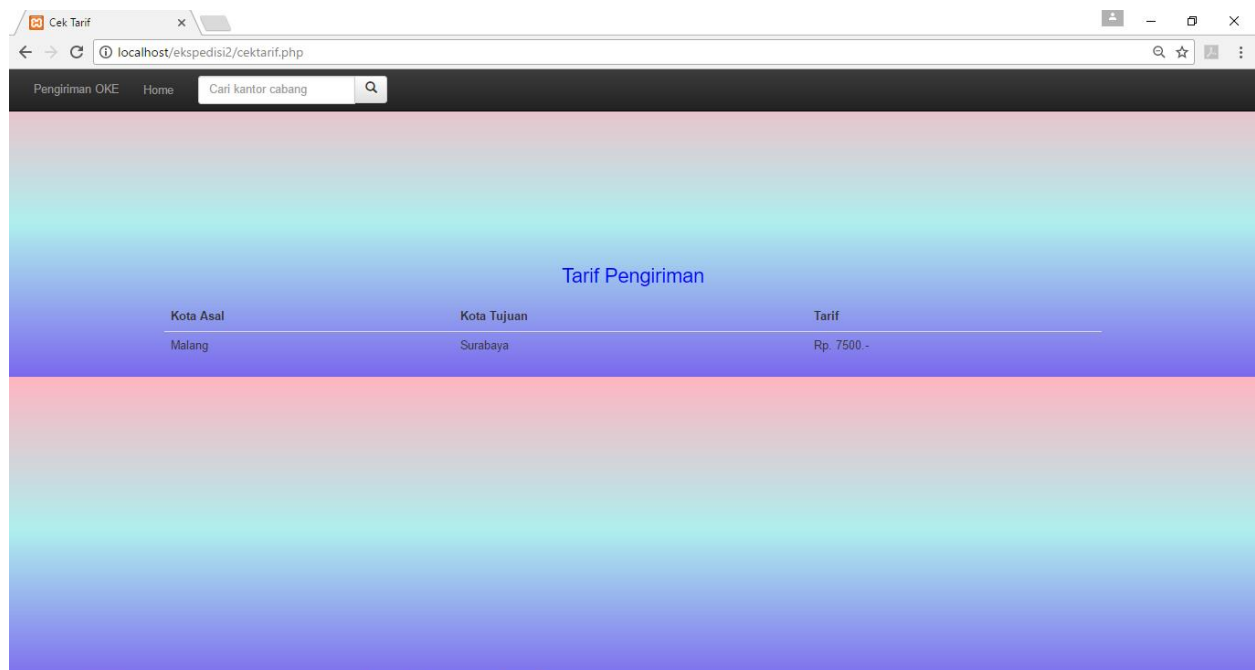
```
body{
```

```
    background: linear-gradient(#FFB6C1,#AFEEEE,#7B68EE);
```

```
}
```

```
</style>
```

Yang memberikan efek warna background gradien dari warna ungu,pink,dan tosca.

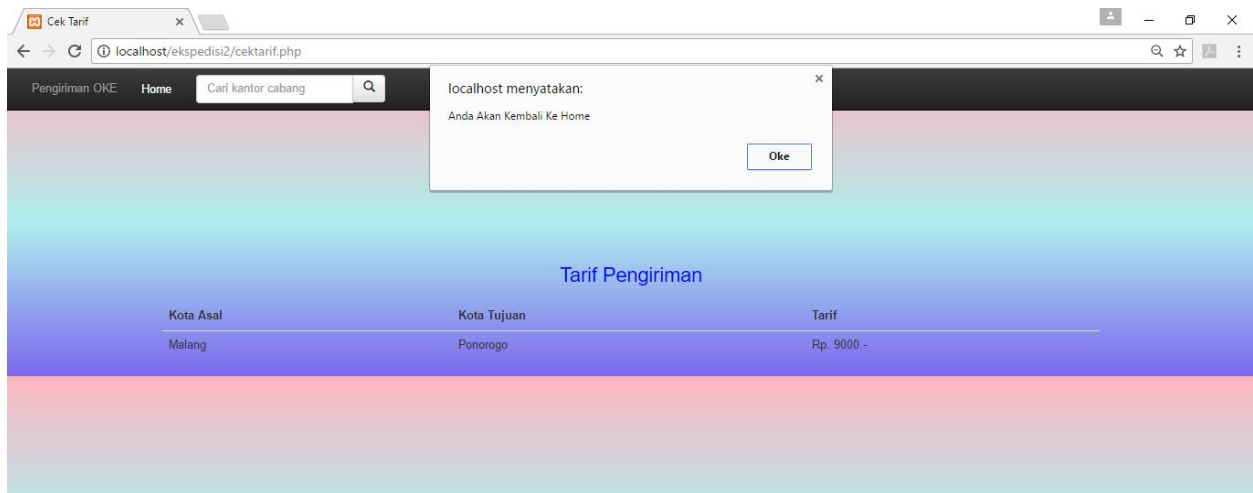


Implementasi Javascript

```
<script> function javascript(){  
    alert('Anda Akan Kembali Ke Home');  
}</script>
```

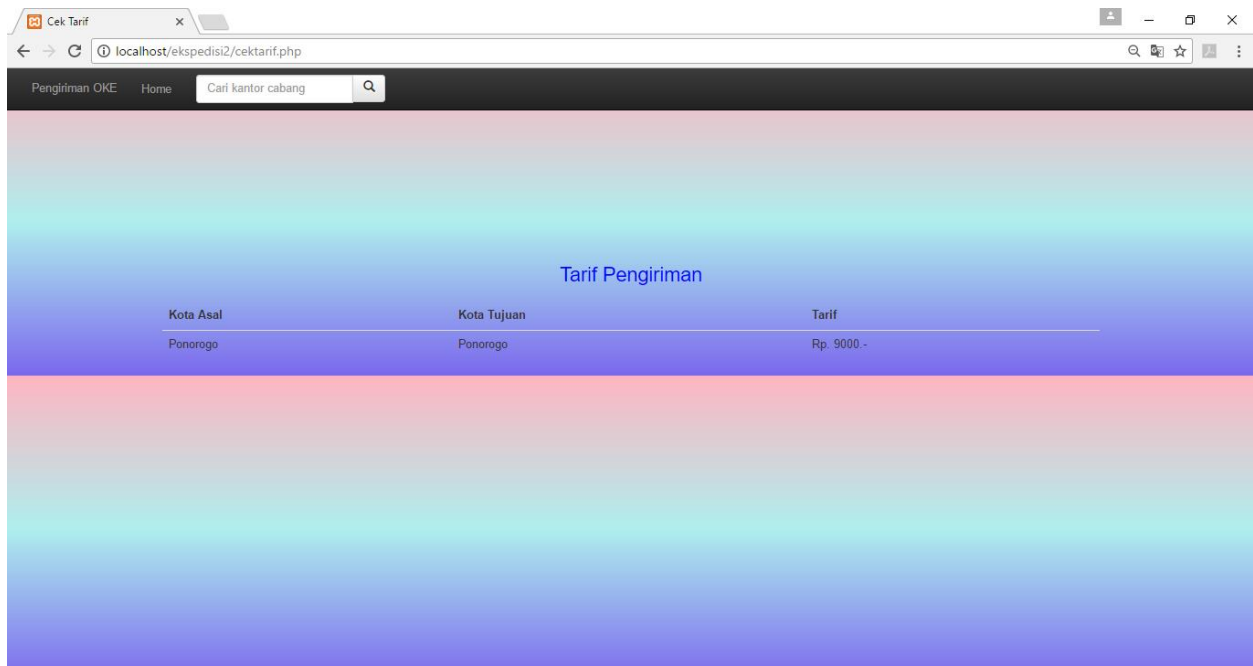
Cara kerja javascript ini adalah diletakkan di dalam fungsi dengan nama javascript(), ketika fungsi ini dipanggil maka akan muncul javascript alert yaitu berupa notifikasi berupa popup box dengan pesan “Anda Akan Kembali Ke Home” di mana fungsi tersebut diletakkan(aksi yang akan memanggil fungsi ini).

Fungsi javascript diatas diletakkan pada menu “Home” dengan perintah onclick pada tampilan cektarif.php .

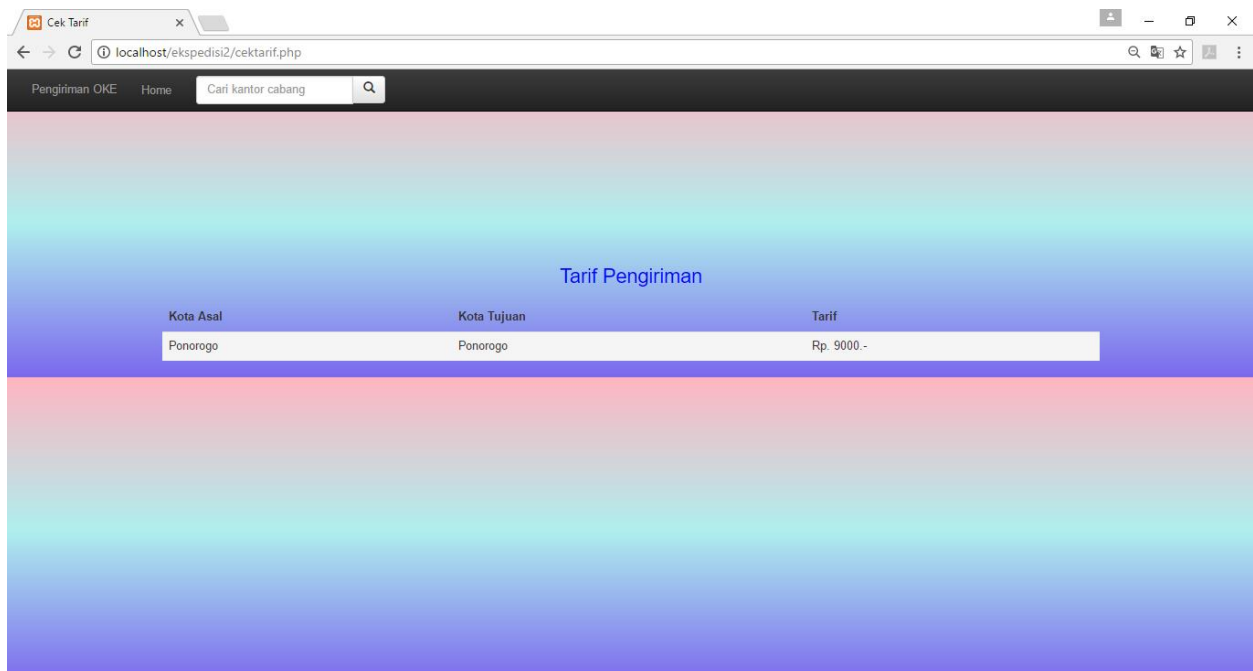


Untuk style sendiri saya menggunakan bootstrap yang juga mengimplementasikan javascript secara otomatis misalnya untuk tampilan tabel cektarif saya menggunakan class table table-hover dimana setiap mouse diarahkan pada baris datanya akan berubah warna

Dari yang sebelumnya tidak ada efek



Ketika diarahkan mouse barisnya akan berubah warna menjadi putih:



Implementasi Database

Tabel tarif:

--struktur tabel tarif yang digunakan--

```
CREATE TABLE `tarif` (  
  `id_tarif` int(255) NOT NULL AUTO_INCREMENT,  
  `tarif` int(11) DEFAULT NULL,  
  `kota_idkota` int(255) NOT NULL,  
  `kota_idkota1` int(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--merelasikan tabel--  
  
ALTER TABLE `tarif`  
  
  ADD CONSTRAINT `fk_Tarif_kota` FOREIGN KEY (`kota_idkota`) REFERENCES `kota` (`idkota`) ON  
DELETE NO ACTION ON UPDATE NO ACTION,  
  
  ADD CONSTRAINT `fk_Tarif_kota1` FOREIGN KEY (`kota_idkota1`) REFERENCES `kota` (`idkota`) ON  
DELETE NO ACTION ON UPDATE NO ACTION;
```

--menambah kolom foreignkey kedalam tabel tarif--

```
ALTER TABLE `tarif`  
  
  ADD PRIMARY KEY (`id_tarif`),
```



```
ADD KEY `fk_Tarif_kota_idx` (`kota_idkota`),  
ADD KEY `fk_Tarif_kota1_idx` (`kota_idkota1`);
```

- Isi tabel--

```
INSERT INTO `tarif` (`id_tarif`, `tarif`, `kota_idkota`, `kota_idkota1`) VALUES  
(1, 3500, 1, 2),  
(2, 4000, 1, 3),  
(3, 4500, 1, 4),  
(4, 2500, 2, 3),  
(5, 2500, 2, 4),  
(6, 3000, 3, 4),  
(7, 3000, 1, 1),  
(8, 3000, 2, 2),  
(9, 3000, 3, 3),  
(10, 3000, 4, 4),  
(11, 4500, 2, 1),  
(12, 5000, 3, 2),  
(13, 5500, 3, 1),  
(14, 5000, 4, 1),  
(15, 5500, 4, 2),  
(16, 6000, 4, 3);
```

Pada SQL- DDL tabel tarif diatas, untuk perintah create table digunakan untuk membuat tabel yang memiliki 2 kolom(atribut) yaitu id_tarif dan tarif sesuai pada ER diatas , selanjutnya dilakukan perubahan pada tabel tersebut dengan perintah alter untuk merelaskannya dengan tabel kota dimana idkota pada tabel kota akan menjadi foreign key(constraint) pada tabel tarif. Sehingga dilakukan juga perubahan pada struktur tabel untuk menambahkan atribut relasi foreignkey ke dalam tabel tarif.

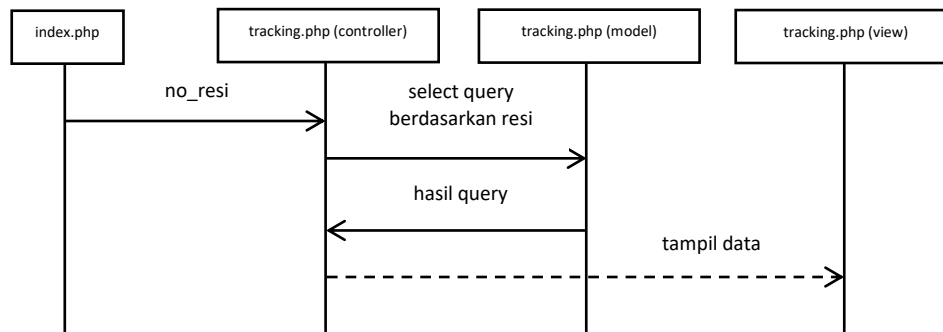
[Lacak Kiriman Barang][NRN]

Dalam suatu sistem ekspedisi, keberadaan fitur untuk melacak barang yang sudah dikirim sangatlah vital. Dengan fitur ini user dapat melacak barang dengan menggunakan nomor resi yang telah didapatkan sebelumnya saat melakukan proses pengiriman/submit barang di branch terkait. Pelacakan ini bersifat realtime dan user hanya perlu memasukkan nomor resi dan sebagai feedback user akan mendapatkan informasi dan status pengiriman barang.

Use Case Scenario [Lacak Kiriman Barang][NRN]

UCS: Melacak kiriman barang	
Objective	User dapat melakukan pengecekan terhadap barang kiriman berdasarkan nomor resi
Actors	User
Pre-Condition	User telah membuka halaman utama (index.php)
Main flow	<ol style="list-style-type: none">1. User mengisi field nomor resi dengan nomor resi barang yang ingin dilacak2. User mengklik tombol cari
Alternative flow	Jika nomor resi yang dimasukkan tidak valid/tidak cocok dengan data yang ada di database, user akan diberi peringatan berupa alert bahwa nomor resi yang dimasukkan salah
Goal state	User ditampilkan halaman yang berisi informasi barang sesuai dengan nomor resi yang diinputkan

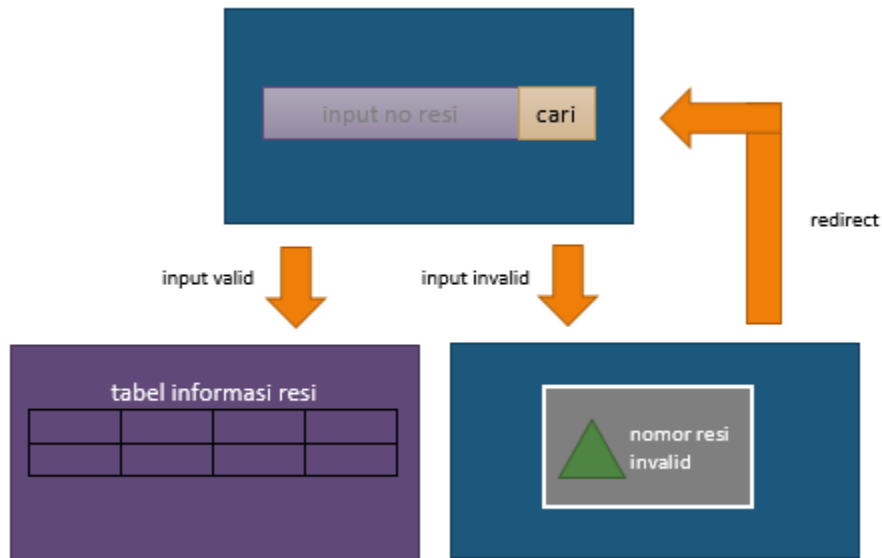
Sequence Diagram [Lacak Kiriman Barang][NRN]



Class Diagram [Lacak Kiriman Barang][NRN]



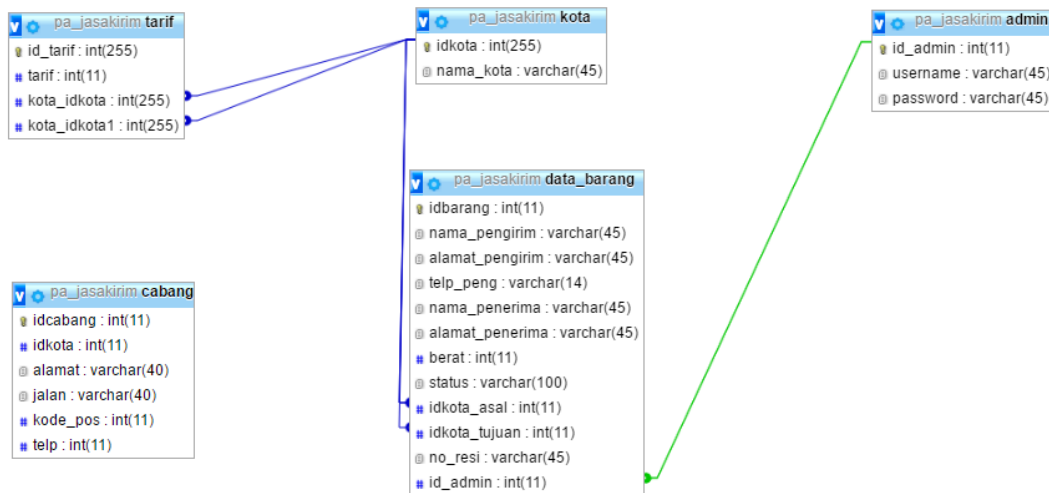
Rancangan Screen Flow [Lacak Kiriman Barang][NRN]



Rancangan Basis Data [Lacak Kiriman Barang][NRN]

data_barang

No	Nama Kolom	Tipe Data	Panjang Data	Contoh Data
1	<i>idbarang</i>	<i>int</i>	11	1, 141
2	<i>nama_pengirim</i>	<i>varchar</i>	45	Narendra, John
3	<i>alamat_pengirim</i>	<i>varchar</i>	45	Jalan x, Jalan y
4	<i>telp_peng</i>	<i>varchar</i>	14	0124234, 872613
5	<i>nama_penerima</i>	<i>varchar</i>	45	Cena, Rocky
6	<i>alamat_penerima</i>	<i>varchar</i>	45	Jalan a, Jalan b
7	<i>berat</i>	<i>int</i>	11	75, 10
8	<i>status</i>	<i>varchar</i>	100	Packing, Diterima
9	<i>idkota_asal</i>	<i>int</i>	11	11, 6
10	<i>idkota_tujuan</i>	<i>int</i>	11	23, 15
11	<i>no_resi</i>	<i>varchar</i>	45	SGS-01, PW-04
12	<i>id_admin</i>	<i>int</i>	11	5, 2



Implementasi [Lacak Kiriman Barang][NRN]

Implementasi PHP

```

<?php

    $no_resi = $_POST['no-resi'];
    require_once("koneksi.php");

    $sql = "SELECT * FROM data_barang INNER JOIN kota ON
    data_barang.idkota_asal=kota.idkota WHERE data_barang.no_resi='$no_resi'";

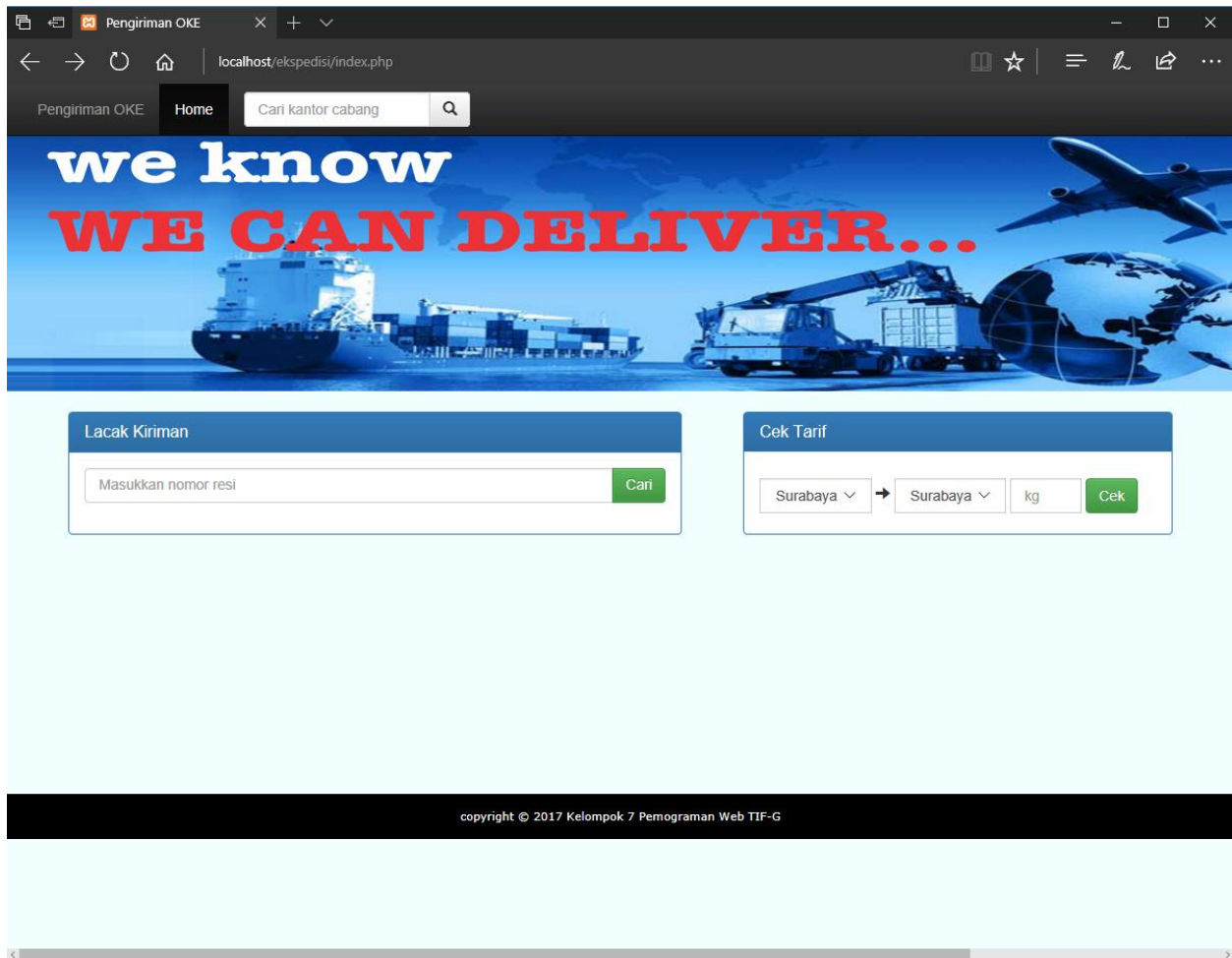
    $sql2 = "SELECT * FROM data_barang INNER JOIN kota ON
    data_barang.idkota_tujuan=kota.idkota WHERE data_barang.no_resi='$no_resi'";

    $result = $conn->query($sql);
    $result2 = $conn->query($sql2);
  
```

Model dan controller menjadi satu dalam struktur program. Untuk mendapatkan nomor resi hasil inputan user, digunakan method post pada halaman input yang value nya disimpan kedalam variable \$no_resi (dan \$no_resi2). Penyimpanan ke dua variabel yang berbeda diperlukan karena data yang akan ditampilkan berada pada dua tabel berbeda. Lalu php akan melakukan koneksi ke database dan melakukan query untuk informasi pengiriman yang no_resi nya sesuai dengan value dari \$no_resi. Kemudian hasil dari query nya disimpan kedalam variabel \$result untuk kemudian diformat untuk ditampilkan pada view nya.

Implementasi HTML5

```
<div class="panel panel-primary" style="margin-top: 200px; margin-left: 9%;  
width: 83%;">  
    <div class="panel-heading">  
        <h3 class="panel-title">Lacak Kiriman</h3>  
    </div>  
    <div class="panel-body">  
        <form action="tracking.php" method="post" >  
            <div class="form-group">  
                <div class="input-group">  
                    <input type="text" class="form-  
control" name="no-resi" placeholder="Masukkan nomor resi">  
                    <span class="input-group-  
btn"><button class="btn btn-success" type="submit">Cari</button></span>  
                </div>  
            </div>  
        </form>  
    </div>  
</div>
```



Bagian input dari pelacakan barang yaitu field input nomor resi menggunakan template dari bootstrap. Untuk melacak informasi barang, user menginputkan nomor resi pada field yang telah disediakan. Setelah melakukan input, user kemudian mengklik cari dan kemudian hasil inputan dari user akan dikirim ke file tracking.php untuk kemudian di proses sehingga menghasilkan data informasi barang.

```
if ($result-> num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $row2 = $result2->fetch_assoc();
        ?>
        <div class="container">
        <h3 style="margin-top: 200px;">Status Pengiriman</h3>
        <table class="table table-hover" style="margin-top: 20px;">
            <thead>
                <tr>
```

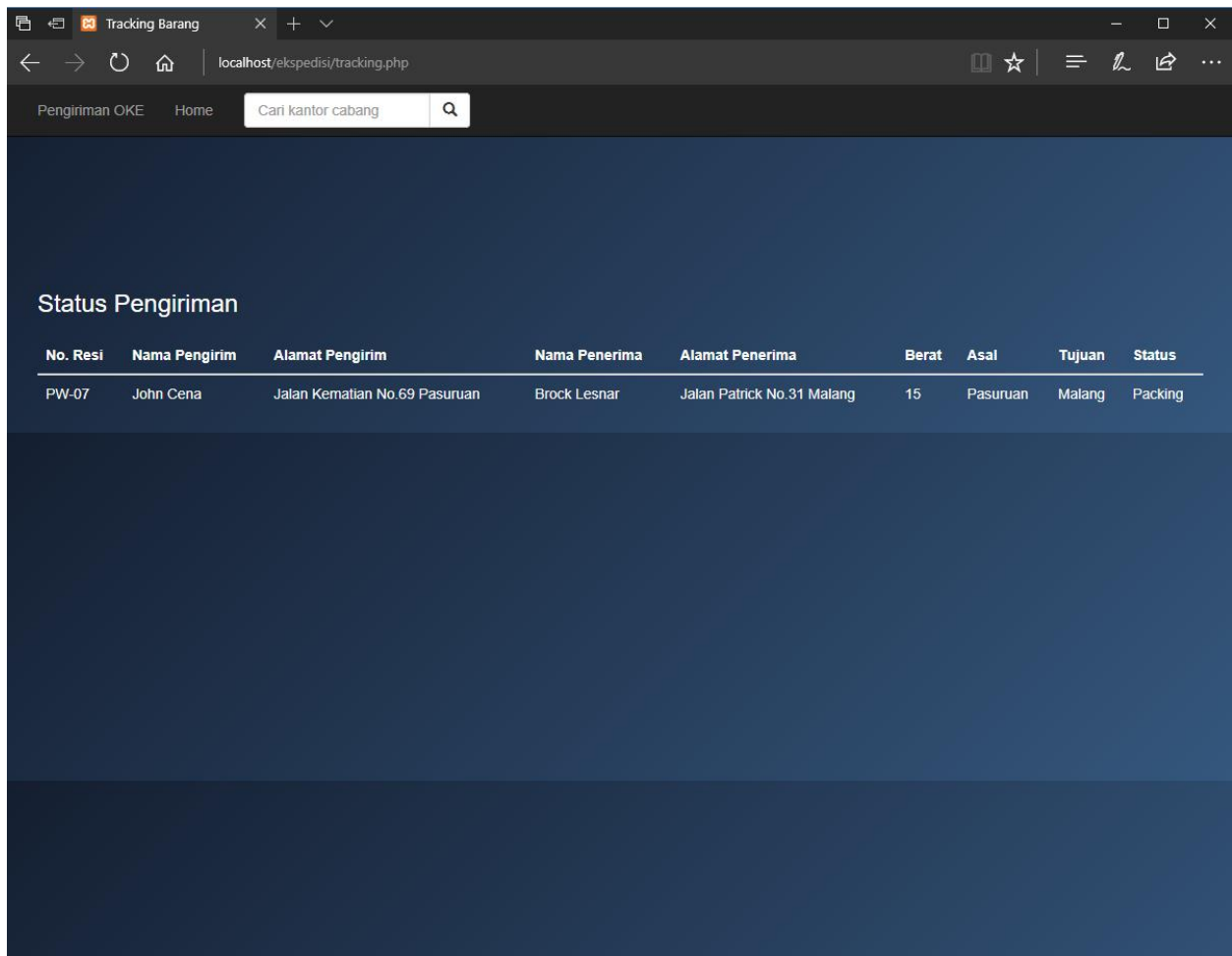
```

        <th>No. Resi</th>
        <th>Nama Pengirim</th>
        <th>Alamat Pengirim</th>
        <th>Nama Penerima</th>
        <th>Alamat Penerima</th>
        <th>Berat</th>
        <th>Asal</th>
        <th>Tujuan</th>
        <th>Status</th>
    </tr>
</thead>
<tbody>
<tr>
    <td><?php echo $row['no_resi'];?></td>
    <td><?php echo $row['nama_pengirim'];?></td>
    <td><?php echo $row['alamat_pengirim'];?></td>
    <td><?php echo $row['nama_penerima'];?></td>
    <td><?php echo $row['alamat_penerima'];?></td>
    <td><?php echo $row['berat'];?></td>
    <td><?php echo $row['nama_kota'];?></td>
    <td><?php echo $row2['nama_kota'];?></td>
    <td><?php echo $row['status'];?></td>

    </tr>
</tbody>
</table>
</div>

</body>
</html>

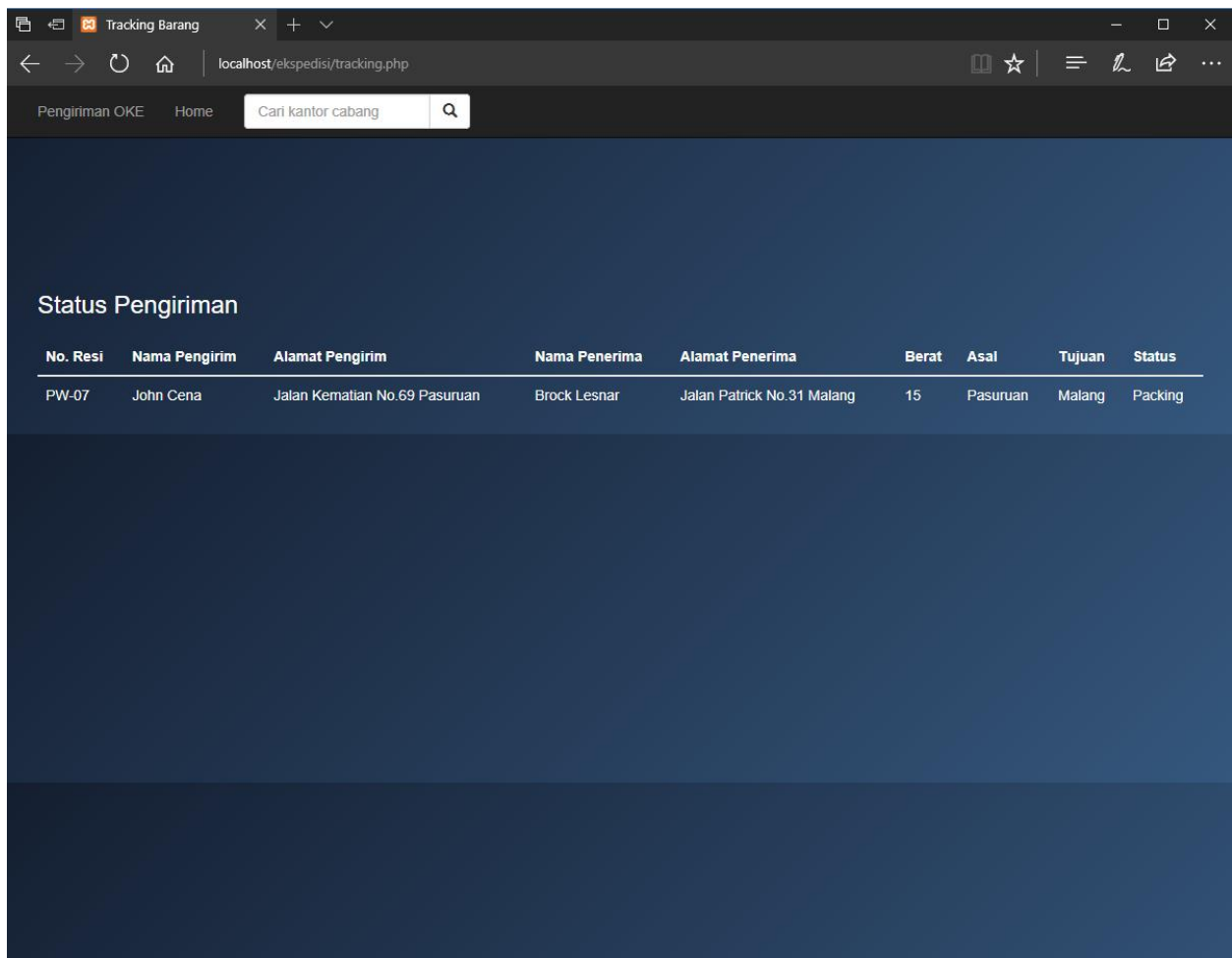
```



Setelah melakukan input nomor resi, dan jika nomor resi yang dimasukkan benar, user kemudian diarahkan ke halaman tracking.php yang dimana akan ditampilkan data informasi barang yang ingin dilacak. Kode program akan membuat header tabel dan isi yang isinya merupakan hasil fetching array dari hasil query yang sebelumnya telah didapatkan dan disimpan kedalam variabel \$row (dan \$row2) dan di panggil melalui index array nya sesuai dengan data mana yang ingin ditampilkan.

Implementasi CSS3

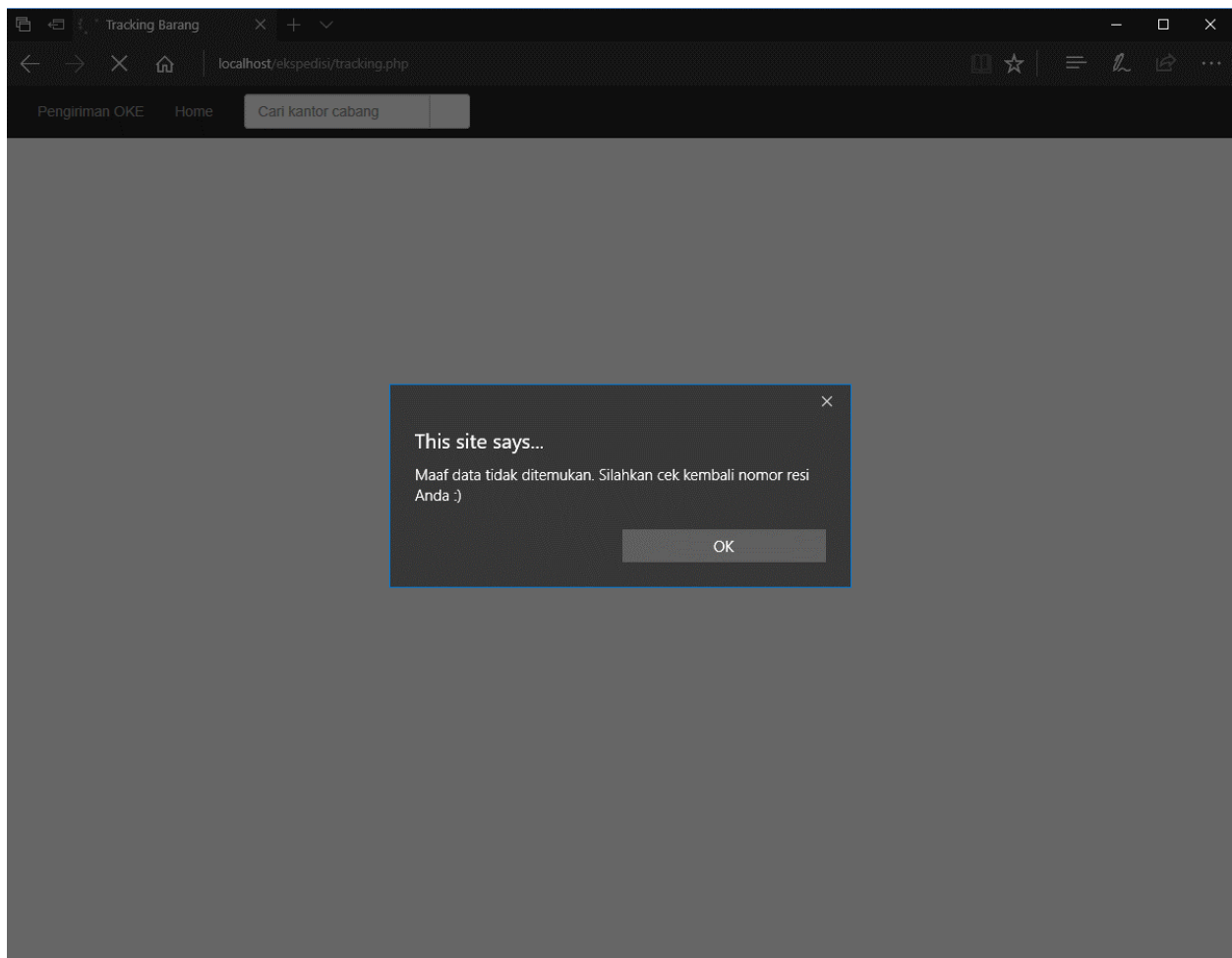
```
body {
    background: linear-gradient(-45deg, #35577D, #141E30);
    margin: 0;
    padding: 0;
    color: white;
}
```

Pada halaman tracking.php, digunakan stylesheet untuk mengatur warna background, margin, padding, dan warna font. Warna background menggunakan gradasi yang value nya diambil dari template yang sudah ada, margin dan padding di set 0, dan kemudian warna font di set ke putih sehingga mudah dibaca.

Implementasi Javascript

```
<script>  
alert("Maaf data tidak ditemukan. Silahkan cek kembali nomor resi Anda :)") );  
window.location.href = 'index.php';  
</script>
```



Ketika nomor resi yang dimasukkan user tidak valid/tidak cocok dengan data yang ada di database, maka secara otomatis halaman akan mengeluarkan peringatan berupa alert bahwa nomor resi yang telah diinputkan salah, dan kemudian user akan di redirect kembali ke halaman penginputan nomor resi.

Implementasi Database

```
CREATE TABLE `data_barang` (  
  `idbarang` int(11) NOT NULL,  
  `nama_pengirim` varchar(45) NOT NULL,  
  `alamat_pengirim` varchar(45) NOT NULL,  
  `telp_peng` varchar(14) NOT NULL,  
  `nama_penerima` varchar(45) NOT NULL,  
  `alamat_penerima` varchar(45) NOT NULL,  
  `berat` int(11) NOT NULL,  
  `status` varchar(100) NOT NULL,
```

```

`idkota_asal` int(11) NOT NULL,
`idkota_tujuan` int(11) NOT NULL,
`no_resi` varchar(45) NOT NULL,
`id_admin` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `data_barang` (`idbarang`, `nama_pengirim`, `alamat_pengirim`,
`telp_peng`, `nama_penerima`, `alamat_penerima`, `berat`, `status`,
`idkota_asal`, `idkota_tujuan`, `no_resi`, `id_admin`) VALUES
(6, 'Narendra', 'Jalan Kehidupan No.1 Malang', '081987999234', 'Jason',
'Jalan Kenanga No.4B Surabaya', 4, 'Proses Pengiriman', 2, 1, 'PW-02', 1),
(7, 'John Cena', 'Jalan Kematian No.69 Pasuruan', '081773364823', 'Brock
Lesnar', 'Jalan Patrick No.31 Malang', 15, 'Packing', 3, 2, 'PW-07', 1);

```

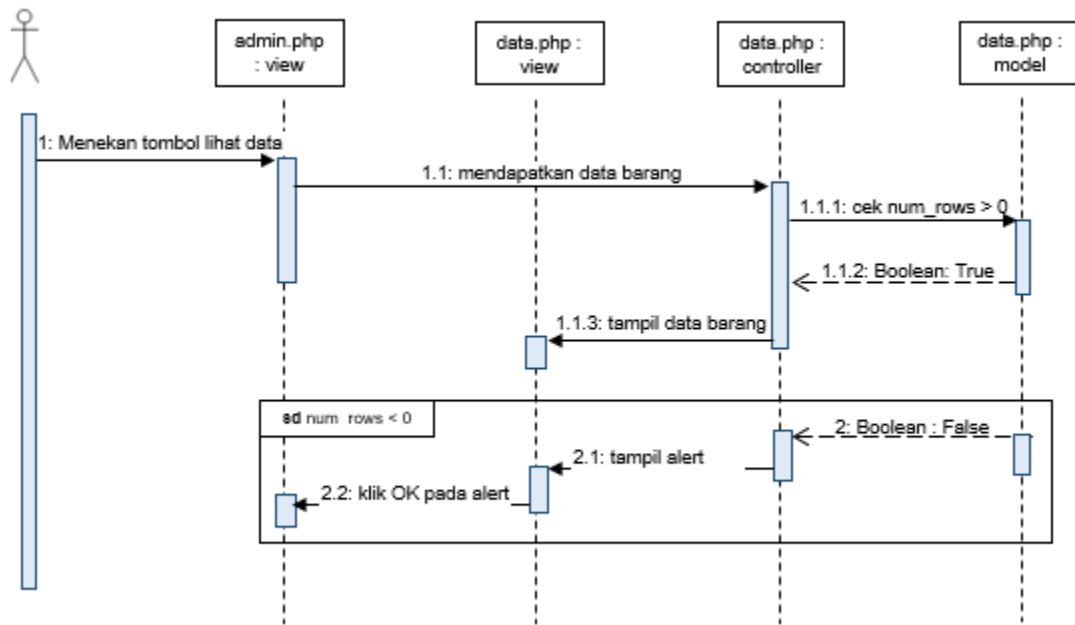
[Lihat Data][SLM]

Deskripsi use case : use case “lihat data” merupakan perilaku sistem dari tampak luar yang menyediakan fungsi untuk melihat data pengiriman yang sebelumnya telah disimpan di dalam database bagi admin sebagai aktor. Use case ini memiliki hubungan <<include>> dengan use case “edit” dan use case “delete” karena jika di dalam halaman yang menampilkan use case “lihat data” terdapat pula use case “delete” dan use case “edit”, sehingga dapat dikatakan bahwa use case “lihat data” merupakan base use case, sedangkan use case “edit” dan use case “delete” merupakan supplier use case.

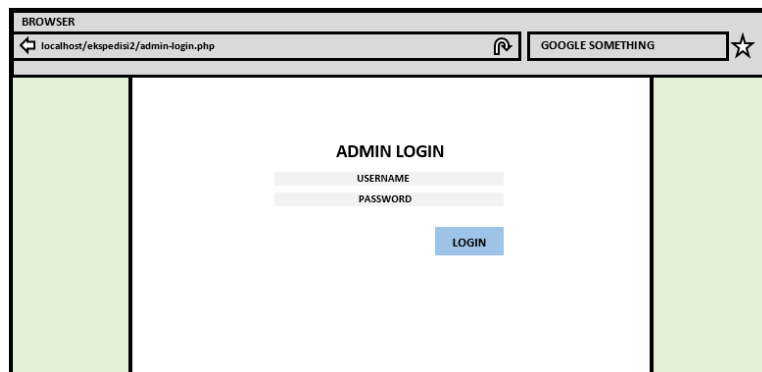
USE-CASE SCRENARIO

Flow of events dari use case “Lihat Data”	
<i>Objective</i>	Memperbolehkan admin untuk melihat data pengiriman barang yang sebelumnya telah disimpan di dalam database
<i>Actors</i>	Admin
<i>Pre-condition</i>	Database dan tabel sudah ada dan/ sudah ada data di dalam tabel
<i>Main Flow</i>	<ol style="list-style-type: none">1. Admin memilih tab bernama “Lihat Data” pada halaman web2. Halaman web menampilkan data pengiriman barang yang sebelumnya telah disimpan di dalam database
<i>Alternative flows</i>	<ol style="list-style-type: none">1. Jika data pengiriman barang belum ada, akan muncul sebuah <i>pop-up/alert</i> yang berisi “Tidak ada data pengiriman ! Silahkan tambah Data !”
<i>Post-condition</i>	Admin dapat mengetahui rekaman data pengiriman barang yang ditampilkan pada halaman web

Sequence Diagram [Lihat Data][SLM]



Rancangan Screen Flow [Lihat data][SLM]



Admin-login.php



BROWSER

localhost/ekspedisi2/admin.php

GOOGLE SOMETHING

TAMBAH DATA LIHAT DATA

Admin.php



BROWSER

localhost/ekspedisi2/data.php

GOOGLE SOMETHING

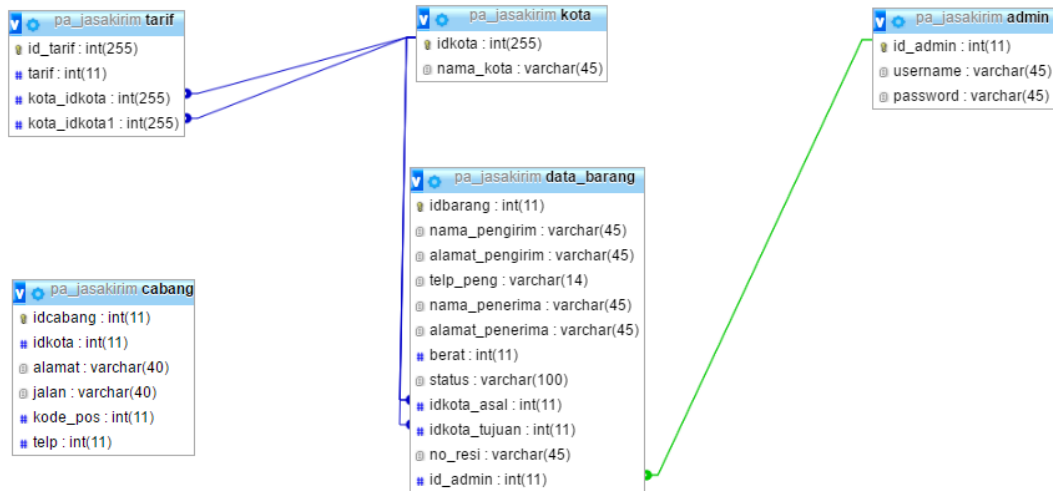
TAMBAH DATA LIHAT DATA

DATA PENGIRIMAN

Kolom1	Kolom2	Kolom3	Kolom4	Kolom5	Kolom6	Kolom7	Kolom8	Kolom9

Data.php

Rancangan Basis Data [Lihat Data][SLM]



Tabel data_barang

No	Nama Kolom	Tipe Data	Panjang Data	Contoh Data
1	idbarang	int	11	1, 12, 13
2	nama_pengirim	varchar	45	Salma, Hanifah
3	alamat_pengirim	varchar	45	Jalan x, Jalan y
4	telp_peng	varchar	14	0124234, 872613
5	nama_penerima	varchar	45	Silmi
6	alamat_penerima	varchar	45	Jalan a, Jalan b
7	berat	int	11	75, 10
8	status	varchar	100	Packing, Diterima
9	idkota_asal	int	11	11, 6
10	idkota_tujuan	int	11	23, 15
11	no_resi	varchar	45	SGS-01, PW-04
12	id_admin	int	11	5, 2

Implementasi

Implementasi PHP

```
<?php
```

```
require_once("koneksi.php");
```

```
$sql = "SELECT * FROM data_barang INNER JOIN kota ON
data_barang.idkota_asal=kota.idkota INNER JOIN admin ON
data_barang.id_admin=admin.id_admin";
```

```
$sql2 = "SELECT * FROM data_barang INNER JOIN kota ON
data_barang.idkota_tujuan=kota.idkota";
```

```
$result = $conn->query($sql);
```

```
$result2 = $conn->query($sql2);
```

Pada implementasi php diatas, model dan controller berada dalam satu file. Untuk dapat menampilkan data barang, pertama dilakukan koneksi ke database terlebih dahulu kemudian menuliskan query untuk menampilkan seluruh data dari database data_barang. Query disimpan dalam variable \$sql dan \$sql2. Penyimpanan dilakukan pada 2 variable karena data berasal dari tabel yang berbeda. Kemudian php akan melakukan koneksi ke database dan melakukan query untuk mendapatkan seluruh data pengiriman barang yang berasal dari gabungan beberapa tabel. Hasil query disimpan dalam variable \$result dan \$result2 yang nantinya akan dipanggil lagi untuk dilakukan fetching agar dapat ditampilkan pada halaman web sesuai dengan yang diharapkan.

Implementasi HTML5

```
<?php
```

```
if($result -> num_rows > 0) {
```

```
?>
```

```
</form>
```

```
<table class="table" style="width: 60%;margin-left: 10%; margin-
top:100px;">
```

```
<thead>
```

```
<th>Admin</th>
```

```
<th>ID Barang</th>
```

```
<th>No.Resi</th>
```

```
<th>Nama Pengirim</th>
```

```
<th>Alamat Pengirim</th>
```

```
<th>Telepon Pengirim</th>
```

```
<th>Nama Penerima</th>
```

```
<th>Alamat Penerima</th>
```

```
<th>Berat</th>
```

```
<th>Asal</th>
```

```
<th>Tujuan</th>
```

```
<th>Status</th>
```



```

        <th></th>
        <th></th>
    </thead>
    <tbody>
    <?php
        $row2;
        while($row = $result->fetch_assoc()) {
            $row2 = $result2->fetch_assoc();
        ?>
    <tr>

        <td><?php echo $row['username'];?></td>
        <td><?php echo $row['idbarang'];?></td>
        <td><?php echo $row['no_resi'];?></td>
        <td><?php echo $row['nama_pengirim'];?></td>
        <td><?php echo $row['alamat_pengirim'];?></td>
        <td><?php echo $row['telp_peng'];?></td>
        <td><?php echo $row['nama_penerima'];?></td>
        <td><?php echo $row['alamat_penerima'];?></td>
        <td><?php echo $row['berat'];?></td>
        <td><?php echo $row['nama_kota'];?></td>
        <td><?php echo $row2['nama_kota'];?></td>
        <td><?php echo $row['status'];?></td>

        <?php echo "<td> <a
href=\"crud.php?ID=$row[idbarang]&code=1\">Edit</a></t
d>"; ?>

        <?php echo "<td> <a
href=\"crud.php?ID=$row[idbarang]&code=2\">delete</a><
/td>"; ?>

    </tr>

    <?php

```

```

    }
    ?>
</tbody>
</table>

```

Setelah diambil data pengiriman barang melalui database, akan di cek apakah jumlah row atau baris lebih dari 0. Jika benar, maka kode program akan menjalankan header table yang akan menampilkan sebuah tabel dengan kolom-kolom yang telah ditentukan dan berisikan hasil fetching array dari query sebelumnya yang telah di dapatkan yang akan disimpan ke dalam variable \$row untuk menampilkan semua kolom kecuali kolom Tujuan dan variable \$row2 untuk menampilkan isi kolom Tujuan sehingga akan menghasilkan data mengenai informasi pengiriman barang.

Pengiriman OKE	Tambah Data	Lihat Data	Logout
----------------	-------------	------------	--------

Admin	ID Barang	No.Resi	Nama Pengirim	Alamat Pengirim	Telepon Pengirim	Nama Penerima	Alamat Penerima	Berat	Asal	Tujuan	Status		
admin	5	PW-01	Galih	Ponorogo	082	Galih	malang	10	Ponorogo	Ponorogo	Packing	Edit	delete
admin	6	PW-02	Narendra	Jalan Kehidupan No.1 Malang	081987999234	Jason	Jalan Kenanga No.4B Surabaya	4	Malang	Malang	Proses Pengiriman	Edit	delete
admin	7	PW-07	John Cena	Jalan Kematian No.69 Pasuruan	081773364823	Brock Lesnar	Jalan Patrick No.31 Malang	15	Surabaya	Malang	Packing	Edit	delete

Implementasi CSS3

```
<table class="table" style="width: 60%;margin-left: 10%; margin-top: 100px;" border="2"> //pada header tabel secara langsung.
```

```

table {
    background-color: transparent;
    border-collapse: collapse;
}
body {
    font-size: 14px;
    color: #333;
}

```

}

Pada data.php terdapat CSS untuk tabel yang dijabarkan secara langsung. Terdapat width dan margin kiri (left), dan margin atas (top) untuk menentukan letak tabel pada halaman web. Digunakan pula stylesheet untuk mengatur warna background tabel, border tabel, ukuran font dan warna font untuk keseluruhan halaman. Background tabel di set transparent sehingga menyatu dengan warna background halaman, ukuran font di set 14px agar terlihat lebih besar dan mudah dibaca dengan warna hitam #333.

Pengiriman OKE

Tambah Data

Lihat Data

Logout

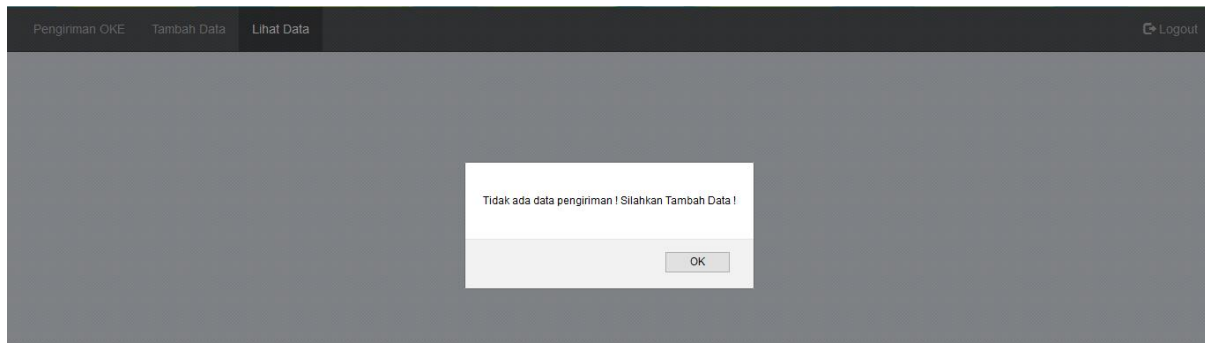
Admin	ID Barang	No.Resi	Nama Pengirim	Alamat Pengirim	Telepon Pengirim	Nama Penerima	Alamat Penerima	Berat	Asal	Tujuan	Status		
admin	5	PW-01	Galih	Ponorogo	082	Galih	malang	10	Ponorogo	Ponorogo	Packing	Edit	delete
admin	6	PW-02	Narendra	Jalan Kehidupan No.1 Malang	081987999234	Jason	Jalan Kenanga No.4B Surabaya	4	Malang	Malang	Proses Pengiriman	Edit	delete
admin	7	PW-07	John Cena	Jalan Kematian No.69 Pasuruan	081773364823	Brock Lesnar	Jalan Patrick No.31 Malang	15	Surabaya	Malang	Packing	Edit	delete

Implementasi Javascript

```
<script>
alert("Tidak ada data pengiriman ! Silahkan Tambah Data !" );
    window.location.href = 'admin.php';
</script>
```

Kode javascript diatas bertujuan untuk memberikan peringatan (*alert*) kepada admin untuk mengisi data pengiriman terlebih dahulu karena didalam database belum ada data pengiriman yang telah disimpan sebelumnya, dan setelah tampilan *alert* tersebut muncul dan di klik "OK" maka akan direferensikan ke halaman admin.php

Cara kerjanya yaitu di dalam halaman data.php yang memuat kode javascript ini, terdapat sintaks if else yang akan memutuskan apakah kode javascript ini dijalankan atau tidak. Jika kondisi if terpenuhi yaitu di dalam database terdapat data pengiriman, maka kode javascript tidak akan dijalankan. Begitu juga ketika kondisi if tidak terpenuhi, maka akan dijalankan blok selanjutnya yaitu else, jalankan kode javascript. Sehingga, tampil halaman seperti di bawah ini.



Implementasi Database

//membuat tabel data_barang

```
CREATE TABLE `data_barang` (  
  `idbarang` int(11) NOT NULL,  
  `nama_pengirim` varchar(45) NOT NULL,  
  `alamat_pengirim` varchar(45) NOT NULL,  
  `telp_peng` varchar(14) NOT NULL,  
  `nama_penerima` varchar(45) NOT NULL,  
  `alamat_penerima` varchar(45) NOT NULL,  
  `berat` int(11) NOT NULL,  
  `status` varchar(100) NOT NULL,  
  `idkota_asal` int(11) NOT NULL,  
  `idkota_tujuan` int(11) NOT NULL,  
  `no_resi` varchar(45) NOT NULL,  
  `id_admin` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

//memasukkan data pengiriman barang

```
INSERT INTO `data_barang` (`idbarang`, `nama_pengirim`, `alamat_pengirim`,  
  `telp_peng`, `nama_penerima`, `alamat_penerima`, `berat`, `status`,  
  `idkota_asal`, `idkota_tujuan`, `no_resi`, `id_admin`) VALUES  
(5, 'Galih ', 'Ponorogo', '082', 'Galih ', 'malang', 10, 'Packing', 1, 2,  
  'PW-01', 1),
```

```
//untuk melihat data pengiriman barang
SELECT * FROM data_barang INNER JOIN kota ON
data_barang.idkota_asal=kota.idkota INNER JOIN admin ON
data_barang.id_admin=admin.id_admin

SELECT * FROM data_barang INNER JOIN kota ON
data_barang.idkota_tujuan=kota.idkota
```

[Tambah Data][GBC]

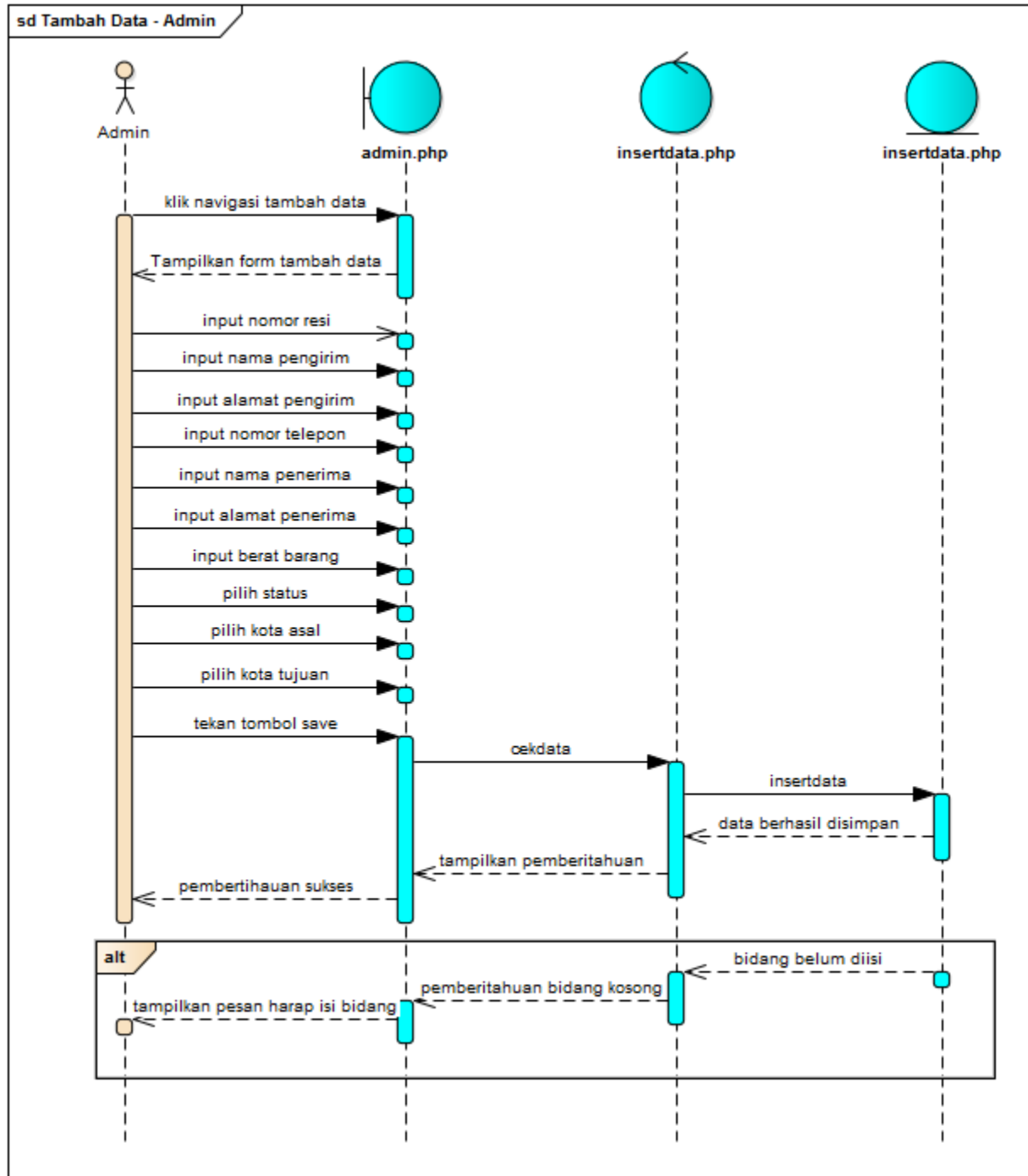
Use case Tambah Data bertujuan untuk menambahkan data pengiriman barang ketika terdapat barang baru yang akan dikirim yang hanya dapat dilakukan oleh admin yang telah login. Data yang dimasukkan kedalam database berupa nomor resi, nama pengirim, alamat pengirim, nomor telepon pengirim, nama penerima, alamat penerima, berat barang, status, kota asal, dan kota tujuan yang dapat dilakukan oleh admin. Data berupa nomor resi, nama pengirim, alamat pengirim, nomor telepon pengirim, nama penerima, alamat penerima, berat barang dapat diinputkan oleh admin secara langsung dengan menulis kedalam kolom, sedangkan data status, kota asal, dan kota tujuan diisi dengan memilih dari data yang sudah disediakan. Data data tersebut penting karena berkaitan dengan identitas dari pengirim barang dan penerima barang. Jika admin lupa tidak mengisi data pada kolom atau ada kolom yang belum diisi maka sistem akan memunculkan pemberitahuan bahwa kolom tersebut harus diisi. Setelah itu admin menekan tombol save. Ketika data benar maka data akan tersimpan didalam database.

Usecase Scenario

Flow of events dari use case "Tambah data "	
Objective	Memperbolehkan admin untuk menambahkan data pengiriman barang ke dalam database
Actors	Admin
Pre-condition	Admin telah login ke dalam sistem
Main Flow	<ol style="list-style-type: none"> 3. Admin klik tab navigasi "Tambah Data" pada halaman web 4. Sistem menampilkan halaman Tambah Data dan form untuk mengisi data pengiriman 5. Admin mengisi kolom Nomor Resi 6. Admin mengisi kolom Nama Pengirim 7. Admin mengisi kolom Alamat Pengirim 8. Admin mengisi kolom Nomor Telepon 9. Admin mengisi kolom Nama Penerima 10. Admin mengisi kolom Alamat Penerima 11. Admin mengisi kolom Berat Barang 12. Admin memilih pilihan pada kolom Status 13. Admin memilih pilihan pada kolom Kota Asal 14. Admin memilih pilihan pada kolom Kota Tujuan 15. Admin menekan tombol Save 16. Sistem memvalidasi data dan menyimpan didalam database

Alternative flows	2. Jika kolom tidak terisi atau ada kolom yang belum diisi, akan muncul sebuah peringatan yang berisi "Harap isi bidang ini."
Post-condition	Admin telah menambahkan data pengiriman barang kedalam database

Sequence Diagram [Tambah Data][GBC]

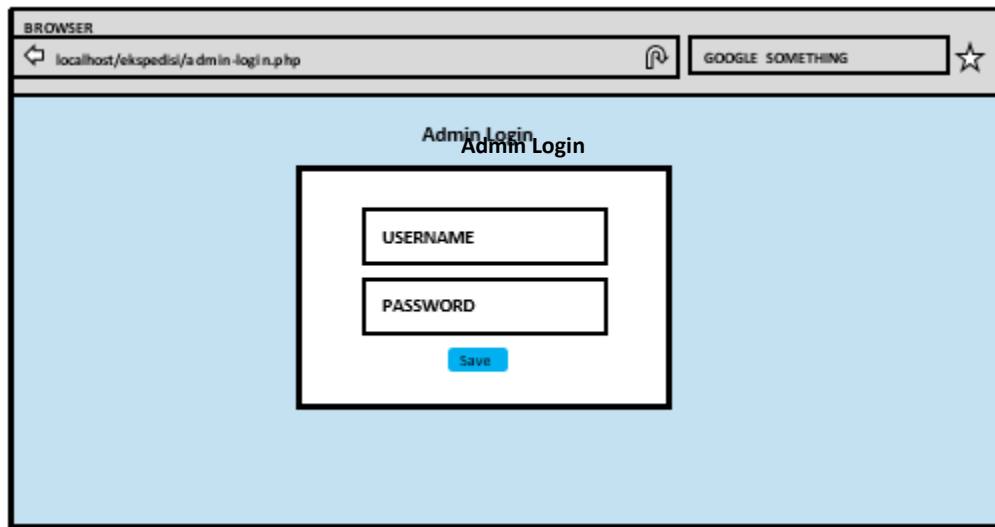


Class Diagram [Tambah Data][GBC]



Rancangan Screen Flow [Tambah Data][GBC]

admin-login.php



A screenshot of a web browser window. The address bar shows 'localhost/ekspedisi/admin-login.php'. The page title is 'Admin Login'. The main content area has a light blue background. In the center, there is a white rectangular box containing a login form. The form has two input fields: 'USERNAME' and 'PASSWORD'. Below these fields is a blue button labeled 'Save'.



admin.php

BROWSER

localhost/ekspedisi/admin.php

GOOGLE SOMETHING

Pengiriman OKE TAMBAH DATA LIHAT DATA Logout

Tambahkan data pengiriman

Nomor Resi

Nama Pengirim

Alamat Pengirim

Nomor Telepon

Nama penerima

Alamat Penerima

Berat Barang

Status

Kota Asal

Kota Tujuan

Save



admin.php

BROWSER

localhost/ekspedisi/admin.php

GOOGLE SOMETHING

Pengiriman OKE TAMBAH DATA LIHAT DATA Logout

Tambahkan data pengiriman

Nomor Resi

Nama Pengirim

Alamat Pengirim

Nomor Telepon

Nama penerima

Alamat Penerima

Berat Barang

Status

Kota Asal

Kota Tujuan

Save



insertdata.php



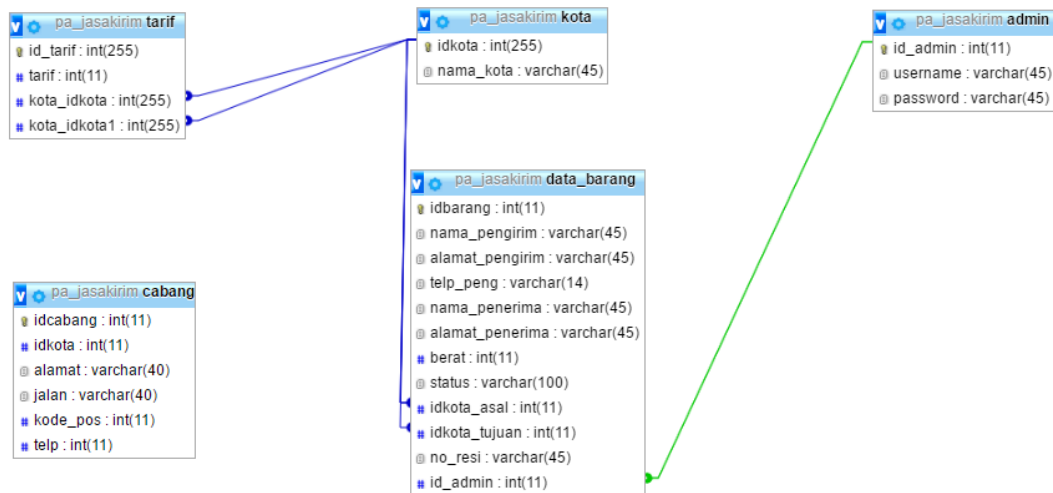
admin.php

The screenshot shows a web browser window with the address bar displaying 'localhost/ekspedisi/admin.php'. The page has a navigation bar with buttons: 'Pengiriman OKE', 'TAMBAH DATA', 'LIHAT DATA', and 'Logout'. The main content area is titled 'Tambahkan data pengiriman' and contains a form with the following fields:

Nomor Resi	<input type="text"/>
Nama Pengirim	<input type="text"/>
Alamat Pengirim	<input type="text"/>
Nomor Telepon	<input type="text"/>
Nama penerima	<input type="text"/>
Alamat Penerima	<input type="text"/>
Berat Barang	<input type="text"/>
Status	<input type="text"/>
Kota Asal	<input type="text"/>
Kota Tujuan	<input type="text"/>

At the bottom of the form is a blue 'Save' button.

Rancangan Basis Data [Tambah Data][GBC]



data_barang

No	Nama Kolom	Tipe Data	Panjang Data	Contoh Data ²
1	idadmin	Int	11	1,2
2	idbarang	Int	11	5,6
3	nama_pengirim	Varchar	45	Galih, Narendra
4	alamat_pengirim	Varchar	45	Ponorogo, Jalan Kehidupan No.1 Malang
5	telp_peng	Varchar	14	085725251777, 081987999234
6	nama_penerima	Varchar	45	Bhaktiar, Jason
7	alamat_penerima	Varchar	45	Malang, Jalan Kenanga No.4B Surabaya
8	berat	Int	11	10,4
9	status	Varchar	100	Packing, Proses Pengiriman
10	idkota_asal	Int	11	1,2
11	idkota_tujuan	Int	11	2,1
12	no_resi	Varchar	45	PW-01, PW-02

² Tuliskan minimal 2 contoh nilai data yang berbeda.

Implementasi

Implementasi PHP

```
<?php
session_start();

if(!isset($_SESSION['username'])) {
    header("location:admin-login.php");

} else {
    $username = $_SESSION['username'];
}

$no_resi = $_POST['no-resi'];
$nama_pengirim = $_POST['nama-pengirim'];
$almt_pengirim = $_POST['almt-pengirim'];
$telp_pengirim = $_POST['telp-pengirim'];
$nama_penerima = $_POST['nama-pengirim'];
$almt_penerima = $_POST['almt-penerima'];
$berat = $_POST['berat'];
$kota_asal = $_POST['kota-asal'];
$kota_tujuan = $_POST['kota-tujuan'];
$status = $_POST['status'];
$id_admin = $_SESSION['id_admin'];

require_once("koneksi.php");

$link = mysqli_connect($namahost, $username , $password , $database);

if ($link) {
    $sql = "INSERT INTO data_barang "
        . " (no_resi, nama_pengirim, alamat_pengirim, telp_peng,
nama_penerima, alamat_penerima, berat, idkota_asal, idkota_tujuan, status, id_admin) VALUES "
        . " ('$no_resi', '$nama_pengirim', '$almt_pengirim',
'$telp_pengirim', '$nama_penerima', '$almt_penerima', '$berat', '$kota_asal', '$kota_tujuan',
'$status', '$id_admin');"

    $result = mysqli_query($link, $sql);
    if($result) {
        ?>
        <script>
            alert("Data telah berhasil ditambahkan");
            window.location.href = 'admin.php';
        </script>
        <?php
        exit;
    } else {
        die("Query gagal dieksekusi. " .
            mysqli_error($link));
    }
}
```

```
}
?>
```

Model dan controller jadi satu file pada insertbarang.php. Untuk melakukan tambah data diperlukan login admin yang sesuai dengan database. Kemudian jika login berhasil maka nilai username akan disimpan menjadi session dan disimpan pada variabel username. Jika variabel username tidak ada isinya, berarti session tidak ada isinya juga dan tidak bisa membuka admin.php atau bisa dikatakan tidak bisa melakukan tambah data. Setelah itu data yang dimasukkan didalam form pada admin.php, maka data akan dikirim ke insertbarang.php. Data dikirim menggunakan method POST sehingga datanya tidak dapat dilihat. Setelah data dikirim ke insertbarang.php maka data akan diambil oleh insertbarang.php dengan menggunakan method POST dan disimpan ke dalam masing-masing variabel yang berbeda. Selain mengambil data yang dimasukkan di form ketika menambahkan data, juga mengambil nilai dari id_admin yang digunakan untuk login yang telah disimpan pada variabel username. Ini berfungsi sebagai data, admin yang mana yang telah menambahkan data barang ke dalam database. Kemudian setelah itu, dilakukan koneksi dengan database dikarenakan data yang diambil akan disimpan didatabase. Jika koneksi berhasil maka data akan dimasukkan ke dalam database data_barang sesuai dengan nilainya masing-masing. Data ditambahkan menggunakan query insert. Setelah penambahan data kedalam databse berhasil, maka akan muncul alert menggunakan javascript. Alert memberitahukan bahwa data berhasil ditambahkan dan kemudian akan me-redirect halaman ke admin.php dan formnya akan kembali kosong.

Implementasi HTML5

```
<div class="container" style="margin-top: 60px">
    <form class="form-horizontal" action="insertbarang.php" method="post">
        <h3> Tambahkan Data Pengiriman </h3>

        <div class="form-group">
            <label class="control-label col-sm-2" for="nama-pengirim">Nomor
Resi</label>

            <div class="col-xs-5">
                <input class="form-control " type="text" name="no-resi" id="nama-
pengirim" required placeholder="Nomor Resi"/>
            </div>
        </div>
        <div class="form-group">
            <label class="control-label col-sm-2" for="nama-pengirim">Nama
Pengirim</label>

            <div class="col-xs-5">
                <input class="form-control " type="text" name="nama-pengirim"
id="nama-pengirim" required placeholder="Nama Pengirim"/>
            </div>
        </div>
        <div class="form-group">
            <label class="control-label col-sm-2" for="almt-pengirim">Alamat
Pengirim</label>

            <div class="col-xs-5">
                <input class="form-control " type="text" name="almt-pengirim"
id="almt-pengirim" required placeholder="Alamat Pengirim"/>
            </div>
        </div>
    </div>
</form>
```

```

        <label class="control-label col-sm-2" for="tlp-pengirim">Nomor
Telepon</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="tlp-pengirim"
id="tlp-pengirim" required placeholder="Nomor Telepon Pengirim"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="nama-penerima">Nama
Penerima</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="nama-penerima"
id="nama-penerima" required placeholder="Nama Penerima"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="almt-penerima">Alamat
Penerima</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="almt-penerima"
id="almt-penerima" required placeholder="Alamat Penerima"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="berat">Berat Barang</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="berat" id="berat"
required placeholder="Berat Barang"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="status">Status</label>
        <div class="col-xs-5">
            <select class="form-control" id="status" name="status" >
                <option value="">--pilih--</option>
                <option value="Packing">Packing</option>
                <option value="Proses Pengiriman">Proses
Pengiriman</option>
                <option value="Sampai Tujuan">Sampai Tujuan</option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="kota-asal">Kota Asal</label>
        <div class="col-xs-5">
            <select class="form-control" id="kota-asal" name="kota-asal" >
                <option value="">--pilih--</option>
                <option value="1">Surabaya</option>
                <option value="2">Malang</option>
                <option value="3">Pasuruan</option>
                <option value="4">Ponorogo</option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="kota-tujuan">Kota
Tujuan</label>

```

```

        <div class="col-xs-5">
            <select class="form-control" id="kota-tujuan" name="kota-tujuan"
>
                <option value="">--pilih--</option>
                <option value="1">Surabaya</option>
                <option value="2">Malang</option>
                <option value="3">Pasuruan</option>
                <option value="4">Ponorogo</option>
            </select>
        </div>
    </div>

    <div class="form-group">
        <label class="control-label col-sm-2" for="save-button"></label>
        <div class="col-xs-5">
            <button class="btn btn-primary" name="save-button">Save</button>
        </div>
    </div>
</form>
</div>

```

HTML diatas adalah digunakan untuk form input pada saat akan menambahkan data. Form berisi beberapa jenis inputan. Pertama terdapat inputan yang berjenis text saja, yaitu inputan nomor resi, nama pengirim, alamat pengirim, telpon pengirim, nama penerima, alamat penerima dan status. Kemudian ada inputan yang membuat pengguna untuk memilih data yang sudah ada pada form menggunakan drop-down list, yaitu status, kota asal, dan kota tujuan. Form diatas dibuat dengan memanfaatkan bootstrap. Form dibuat dengan metode horizontal, artinya setiap form akan dibuat satu persatu per baris. Seperti pada hasil screenshoot, inputan nama pengirim berada dibawah inputan nomor resi, tidak disampingnya. Ini adalah fungsi dari class dari bootstrap yaitu form-horizontal. Dalam penulisan label juga memanfaatkan bootstrap, yaitu dengan control-label col-sm-2. Hal ini berarti label ditentukan ukurannya dengan aturan sm dan menggunakan lebar kolom tipe 2. Kemudian setiap tempat inputan akan di class-kan lagi dengan fungsi form-group yang berguna untuk mengkatogerikan setiap tempat inputan. Setiap inputan juga diatur lebarnya dengan menggunakan fungsi class grid pada bootstrap. Hal tersebut bisa dilihat pada class yang bernilai "col-xs-2" yaitu berarti tempat inputan adalah kolom yang ukurannya adalah xs atau berarti digunakan untuk telepon dan angka 2 menunjukkan jenis pengaturan yang digunakan adalah tipe 2. Dalam button juga menerapkan fungsi dari bootstrap yaitu dengan menggunakan jenis button primary yang lebarnya juga diatur dengan ketentuan col-xs-5.

The screenshot shows a web application for managing shipping data. The interface includes a top navigation bar with 'Admin Login', 'Tambah Data', and 'Lihat Data' tabs. The main content area is titled 'Tambahkan Data Pengiriman' and contains a form with the following fields:

- Nomor Resi (text input)
- Nama Pengirim (text input)
- Alamat Pengirim (text input)
- Nomor Telepon (text input)
- Nama Penerima (text input)
- Alamat Penerima (text input)
- Berat Barang (text input)
- Status (dropdown menu)
- Kota Asal (dropdown menu)
- Kota Tujuan (dropdown menu)

A 'Save' button is located at the bottom of the form.

```
<div class="container" style="margin-top: 60px">
    <form class="form-horizontal" action="insertbarang.php" method="post">
        <h3> Tambahkan Data Pengiriman </h3>

        <div class="form-group">
            <label class="control-label col-sm-2" for="nama-pengirim">Nomor
Resi</label>

            <div class="col-xs-5">
                <input class="form-control " type="text" name="no-resi" id="nama-
pengirim" required placeholder="Nomor Resi"/>
            </div>
        </div>
        <div class="form-group">
            <label class="control-label col-sm-2" for="nama-pengirim">Nama
Pengirim</label>

            <div class="col-xs-5">
                <input class="form-control " type="text" name="nama-pengirim"
id="nama-pengirim" required placeholder="Nama Pengirim"/>
            </div>
        </div>
        <div class="form-group">
            <label class="control-label col-sm-2" for="almt-pengirim">Alamat
Pengirim</label>

            <div class="col-xs-5">
                <input class="form-control " type="text" name="almt-pengirim"
id="almt-pengirim" required placeholder="Alamat Pengirim"/>
            </div>
        </div>
        <div class="form-group">
            <label class="control-label col-sm-2" for="tlp-pengirim">Nomor
Telepon</label>
```

```

        <div class="col-xs-5">
            <input class="form-control " type="text" name="tlp-pengirim"
id="tlp-pengirim" required placeholder="Nomor Telepon Pengirim"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="nama-penerima">Nama
Penerima</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="nama-penerima"
id="nama-penerima" required placeholder="Nama Penerima"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="almt-penerima">Alamat
Penerima</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="almt-penerima"
id="almt-penerima" required placeholder="Alamat Penerima"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="berat">Berat Barang</label>
        <div class="col-xs-5">
            <input class="form-control " type="text" name="berat" id="berat"
required placeholder="Berat Barang"/>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="status">Status</label>
        <div class="col-xs-5">
            <select class="form-control" id="status" name="status" >
                <option value="">--pilih--</option>
                <option value="Packing">Packing</option>
                <option value="Proses Pengiriman">Proses
Pengiriman</option>
                <option value="Sampai Tujuan">Sampai Tujuan</option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="kota-asal">Kota Asal</label>
        <div class="col-xs-5">
            <select class="form-control" id="kota-asal" name="kota-asal" >
                <option value="">--pilih--</option>
                <option value="1">Surabaya</option>
                <option value="2">Malang</option>
                <option value="3">Pasuruan</option>
                <option value="4">Ponorogo</option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2" for="kota-tujuan">Kota
Tujuan</label>
        <div class="col-xs-5">

```



```

        <select class="form-control" id="kota-tujuan" name="kota-tujuan"
>
            <option value="">--pilih--</option>
            <option value="1">Surabaya</option>
            <option value="2">Malang</option>
            <option value="3">Pasuruan</option>
            <option value="4">Ponorogo</option>
        </select>
    </div>
</div>

    <div class="form-group">
        <label class="control-label col-sm-2" for="save-button"></label>
        <div class="col-xs-5">
            <button class="btn btn-primary" name="save-button">Save</button>
        </div>
    </div>
</form>
</div>

```

Untuk halaman output akan kembali ke halaman admin.php yang menampilkan halaman untuk input data barang. Jadi susunan kerangka pada halaman html output sama dengan input. Tetapi meskipun kembali ke halaman admin.php semua data yang telah diinputkan tidak akan tampil, dikarenakan form akan kembali dikosongi. Untuk penjelasan susunannya kerangkanya adalah sebagai berikut. HTML diatas adalah digunakan untuk form input pada saat akan menambahkan data. Form berisi beberapa jenis inputan. Pertama terdapat inputan yang berjenis text saja, yaitu inputan nomor resi, nama pengirim, alamat pengirim, telpon pengirim, nama penerima, alamat penerima dan status. Kemudian ada inputan yang membuat pengguna untuk memilih data yang sudah ada pada form menggunakan drop-down list, yaitu status, kota asal, dan kota tujuan. Form diatas dibuat dengan memanfaatkan bootstrap. Form dibuat dengan metode horizontal, artinya setiap form akan dibuat satu persatu per baris. Seperti pada hasil screenshoot, inputan nama pengirim berada dibawah inputan nomor resi, tidak disampingnya. Ini adalah fungsi dari class dari bootstrap yaitu form-horizontal. Dalam penulisan label juga memanfaatkan bootstrap, yaitu dengan control-label col-sm-2. Hal ini berarti label ditentukan ukurannya dengan aturan sm dan menggunakan lebar kolom tipe 2. Kemudian setiap tempat inputan akan di class-kan lagi dengan fungsi form-group yang berguna untuk mengkatogerikan setiap tempat inputan. Setiap inputan juga diatur lebarnya dengan menggunakan fungsi class grid pada bootstrap. Hal tersebut bisa dilihat pada class yang bernilai "col-xs-2" yaitu berarti tempat inputan adalah kolom yang ukurannya adalah xs atau berarti digunakan untuk telepon dan angka 2 menunjukkan jenis pengaturan yang digunakan adalah tipe 2. Dalam button juga menerapkan fungsi dari bootstrap yaitu dengan menggunakan jenis button primary yang lebarnya juga diatur dengan ketentuan col-xs-5.

Implementasi CSS3

```
<style>
  body{
    font-size: 14px;
  }
  h3{
    font-family: comic;
    font-size: 24px;
    color: blue;
  }
</style>
```

CSS yang saya implementasikan hampir semuanya menggunakan bootstrap sehingga saya hanya sedikit menggunakan CSS yang murni saya buat sendiri. CSS yang saya buat tersebut terletak diantara tag <head> pada admin.php dan mengatur penampilan dari body dan heading 3. Dampak yang diberikan adalah mengubah ukuran font pada body html menjadi ukuran 14 px. Selain itu, juga merubah style dari tag h3. Heading dirubah font stylenya menjadi comic, ukuran font-nya 24 px dan warna dari font tersebut berubah menjadi biru.

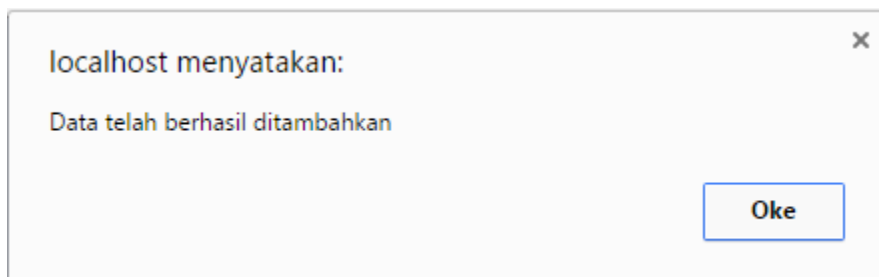
Tambahkan Data Pengiriman

Nomor Resi	<input type="text" value="Nomor Resi"/>
Nama Pengirim	<input type="text" value="Nama Pengirim"/>
Alamat Pengirim	<input type="text" value="Alamat Pengirim"/>
Nomor Telepon	<input type="text" value="Nomor Telepon Pengirim"/>
Nama Penerima	<input type="text" value="Nama Penerima"/>
Alamat Penerima	<input type="text" value="Alamat Penerima"/>
Berat Barang	<input type="text" value="Berat Barang"/>
Status	<input type="text" value="--pilih--"/>
Kota Asal	<input type="text" value="--pilih--"/>
Kota Tujuan	<input type="text" value="--pilih--"/>
<input type="button" value="Save"/>	

Implementasi Javascript

```
<script>
    alert("Data telah berhasil ditambahkan");
    window.location.href = 'admin.php';
</script>
```

Javascript yang saya terapkan adalah menampilkan sebuah alert atau peringatan ketika sebuah data berhasil ditambahkan. Kemudian setelah itu ketika pengguna menekan tombol Oke maka akan me-redirect halaman kembali ke admin.php.



Implementasi Database

```
CREATE TABLE `data_barang` (  
  `idbarang` int(11) NOT NULL,  
  `nama_pengirim` varchar(45) NOT NULL,  
  `alamat_pengirim` varchar(45) NOT NULL,  
  `telp_peng` varchar(14) NOT NULL,  
  `nama_penerima` varchar(45) NOT NULL,  
  `alamat_penerima` varchar(45) NOT NULL,  
  `berat` int(11) NOT NULL,  
  `status` varchar(100) NOT NULL,  
  `idkota_asal` int(11) NOT NULL,  
  `idkota_tujuan` int(11) NOT NULL,  
  `no_resi` varchar(45) NOT NULL,  
  `id_admin` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```