

## BAB 4. IMPLEMENTASI SISTEM

Pada bab ini akan dibahas mengenai implementasi sistem. Adapun implementasi sistem dibuat berdasarkan rancangan sistem yang telah dibahas pada bab sebelumnya. Implementasi sistem meliputi pengaturan database *SQL Server 2008* beserta implementasi aplikasi.

### 4.1. Pengaturan Koneksi *Database*

Pengaturan koneksi *database* dilakukan untuk dapat menghubungkan antara aplikasi dengan *database*. Dengan terhubungnya aplikasi dengan *database*, maka aplikasi dapat menampilkan, menambah, mengganti dan mengelola data yang ada. Untuk dapat terkoneksi dengan *database*, maka pengguna harus melakukan konfigurasi pada *file* khusus yang disediakan oleh *framework Codeigniter*. *File* tersebut terdapat pada *application* → *config* → *database.php*. Adapun pengaturan yang dilakukan meliputi pengaturan *hostname*, *port*, *username*, *password*, dan *database* yang dipakai pada aplikasi. Pada Segmen 4.1 dijelaskan mengenai pengaturan koneksi *database* yang digunakan.

#### Segmen 4.1. Pengaturan Koneksi Aplikasi dengan *Database*

```
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'DESKTOP-A8MH6V3';
$db['default']['port'] = '1433';
$db['default']['username'] = 'skripsi';
$db['default']['password'] = 'skripsi';
$db['default']['database'] = 'master';
$db['default']['dbdriver'] = 'sqlsrv';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = FALSE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
```

#### 4.2. Implementasi Program pada Modul *Administrator*

Pada bagian ini dijelaskan mengenai implementasi program pada bagian *administrator* yang dibuat berdasarkan rancangan sistem pada bab sebelumnya. Implementasi program berupa aplikasi berbasis *website* dengan bahasa pemrograman PHP.

##### 4.2.1. *Model, View, dan Controller* Program pada Modul *Administrator*

Modul *Administrator* dibuat dengan menggunakan *framework Codeigniter versi 2.0.3*. *Framework* ini menerapkan pola pemrograman *model, view, dan controller (MVC)*. Komponen *model* merupakan suatu *class* yang mengatur setiap data yang ingin dikelola dari *database*. Pengaturan tersebut dilakukan dengan membuat fungsi dengan tujuannya masing-masing. Daftar *class model* pada modul *administrator* dapat dilihat pada Tabel 4.1 dan Tabel 4.2. Sedangkan komponen *view* merupakan komponen yang berisikan *code* dari *interface* program. *File view* dimasukan kedalam *directory* yang berbeda-beda sesuai dengan menu pada *navbar interface*. Daftar *file view* untuk modul *administrator* terdapat pada Tabel 4.3 dan Tabel 4.4. *Controller* merupakan *file* yang menghubungkan antara *view* dan *modul*. *Controller* memberikan perintah untuk menjalankan *view* tertentu dengan memberikan akses untuk data yang diakses melalui fungsi pada *model*. Daftar *class controller* dapat dilihat pada Tabel 4.5 dan Tabel 4.6.

Tabel 4.1. Daftar *File Model* dalam Modul *Administrator*

<i>Usecase</i>	<i>File</i>	Keterangan
<i>Manage Dashboard</i>	MDashboard	<i>Model</i> untuk mengatur <i>data</i> yang dibutuhkan pada pengaturan <i>dashboard</i>
<i>Manage Truck</i>	MTruck.php	<i>Model</i> untuk tabel <i>truck</i>
<i>Manage Driver</i>	MDriver.php	<i>Model</i> untuk tabel <i>driver</i>
<i>Manage Shipment</i>	MShipment.php	<i>Model</i> untuk tabel <i>shipment</i>
<i>Manage Agenda</i>	MAgenda.php	<i>Model</i> untuk mengatur <i>data</i> yang dibutuhkan pada pengaturan agenda

Tabel 4.2. Daftar *File Model* dalam Modul *Administrator* (lanjutan)

<i>Usecase</i>	<i>File</i>	<i>Keterangan</i>
<i>Manage Delivery Order</i>	MDeliveryOrder.php	<i>Model</i> untuk tabel <i>delivery order</i>
<i>Manage Detail Delivery Order</i>	MDetailDeliveryOrder.php	<i>Model</i> untuk tabel <i>detail delivery order</i>
<i>Processing Address</i>	MProcess.php	<i>Model</i> untuk mengatur data yang dibutuhkan dalam melakukan proses perhitungan rute
<i>Manage Report</i>	MReport.php	<i>Model</i> untuk mengatur data yang dibutuhkan pada proses <i>report</i>
<i>Search Delivery Order</i>	MEta.php	<i>Model</i> untuk mengatur data yang dibutuhkan pada proses pencarian data estimasi

Tabel 4.3. Daftar *File View* pada Modul *Administrator*

<i>Usecase</i>	<i>File</i>	<i>Keterangan</i>
<i>Manage Agenda</i>	Vcalendar.php	Tampilan untuk melihat agenda pengiriman
	Vcalendardetail.php	Tampilan untuk melihat urutan pengiriman barang berdasarkan armada
	Vcalendardetailbarang.php	Tampilan untuk melihat detail status pengiriman barang
<i>Manage Dashboard</i>	Vdashboard.php	Tampilan untuk melihat <i>progress</i> pengiriman barang
<i>Search Delivery Order</i>	VetaSearch.php	Tampilan untuk mencari suatu pengiriman barang beserta detail pengiriman
<i>Manage Detail Delivery Order</i>	Vlistdetdo.php	Tampilan untuk melihat dan menambah data <i>detail delivery order</i>
<i>Manage Delivery Order</i>	Vlistdo.php	Tampilan untuk melihat dan menambah data <i>delivery order</i>
<i>Manage Driver</i>	Vlistdriver.php	Tampilan untuk melihat data <i>driver</i>
<i>Manage Shipment</i>	Vlistshipment.php	Tampilan untuk melihat dan menambah data <i>shipment</i>

Tabel 4.4. Daftar *File View* pada Modul *Administrator* (lanjutan)

<i>Usecase</i>	<i>File</i>	<i>Keterangan</i>
<i>Manage Truck</i>	Vlisttruck.php	Tampilan untuk melihat data <i>truck</i>
<i>Processing Address</i>	Vprocess.php	Tampilan untuk melihat list semua pengiriman barang dengan keterangan status proses
	Vprocessdetail.php	Tampilan untuk menjalankan dan menampilkan hasil hitungan algoritma
	Veditaddressform.php	Tampilan untuk mengubah lokasi alamat pengiriman
<i>Manage Report</i>	Vreportdriver.php	Tampilan yang berisikan analisa <i>report</i> mengenai supir
	Vreportgoods.php	Tampilan yang berisikan analisa <i>report</i> mengenai barang pengiriman
	Vreportshipment.php	Tampilan yang berisikan analisa <i>report</i> mengenai <i>shipment</i>
	Vreporttruck.php	Tampilan yang berisikan analisa <i>report</i> mengenai armada pengiriman
-	Vheader.php	Berisikan <i>code</i> untuk <i>header file</i> seperti <i>file link</i> , <i>script</i> , <i>meta</i> , dan <i>title</i>
-	Vnav.php	Berisikan <i>code</i> untuk <i>navbar</i>
-	Vlogin.php	Berisikan <i>login page</i> program
-	Vtemplate.php	<i>File</i> untuk menggabungkan <i>vheader.php</i> , <i>vnav.php</i> , dan <i>interface file</i> yang diinginkan

Tabel 4.5. Daftar *File Controller* pada Modul *Administrator*

<i>Usecase</i>	<i>File</i>	<i>Keterangan</i>
<i>Manage Agenda</i>	CAgenda.php	<i>File</i> yang dipakai untuk mengelola segala kebutuhan pada proses <i>manage agenda</i>
<i>Manage Dashboard</i>	CDashboard.php	<i>File</i> yang dipakai untuk kebutuhan proses <i>manage dashboard</i>
<i>Searching Delivery Order</i>	CEta.php	<i>File</i> yang dipakai untuk mengelola proses <i>searching delivery order</i>

Tabel 4.6. Daftar *File Controller* pada Modul *Administrator* (lanjutan)

<i>Usecase</i>	<i>File</i>	<i>Keterangan</i>
<i>Manage Detail Delivery Order, Manage Delievery Order, Manage Driver, Manage Shipment, dan Manage Truck</i>	CManage.php	<i>Controller</i> untuk mengelola segala kebutuhan pada data supir, armada, <i>delivery order</i> , <i>shipment</i> , dan <i>detail delivery order</i>
<i>Processing Address</i>	CProcess.php	<i>Controller</i> yang membantu mengatur <i>view</i> dan <i>model</i> untuk proses perhitungan rute
<i>Manage Report</i>	CReport.php	<i>Controller</i> untuk mengelola segala kebutuhan pada proses <i>reporting</i>
-	CWelcome.php	<i>Controller</i> untuk menampilkan halaman <i>login</i>

#### 4.2.2. Pembuatan Fitur pada Modul *Administrator*

Pada Modul *Administrator* tersedia fitur-fitur yang memungkinkan pengguna untuk melakukan penambahan data, melakukan proses dari data yang ada, serta membuat *report*. Fitur-fitur tersebut akan dijelaskan pada bagian selanjutnya.

##### 4.2.2.1. Fitur *Manage Data*

Fitur ini digunakan untuk melihat dan menambahkan data supir, armada, *shipment*, *delivery order*, dan *detail delivery order*. Data disajikan dalam bentuk tabel yang dilengkapi dengan fitur *sorting* dan *searching*. Sedangkan untuk menambahkan data, pengguna dapat memilih menu sesuai dengan data yang ingin ditambahkan. Kemudian pilih menu *import* dan tekan tombol *import data*. Pengguna dapat memilih *file excel* yang diinginkan dan melakukan *import* pada bagian yang sudah disediakan. Setelah pengguna melakukan *import*, pengguna

dapat memastikan apakah data yang dimasukkan sudah benar, dengan cara melihat table yang muncul pada bagian dibawahnya. Jika data tersebut belum benar, pengguna dapat melakukan *import* data kembali. Daftar fungsi, prosedur, dan *event* pada fitur *manage* data dapat dilihat pada Tabel 4.7, Tabel 4.8, dan Tabel 4.9.

Tabel 4.7. Daftar Fungsi/Prosedur/Event pada Fitur *Manage* Data

<i>Usecase</i>	Menu / Submenu	Fungsi/Procedure /Event	Keterangan	Segmen
<i>Manage Driver</i>	<i>List Driver</i>	Fungsi <i>get_all_driver_detail</i>	Fungsi yang digunakan untuk mengambil data detail <i>driver</i> pada <i>database</i>	Segmen 4.2
	<i>Import New Driver</i>	Event Penambahan Data pada <i>View</i>	<i>Code</i> pada bagian <i>view</i> yang digunakan untuk menyimpan data supir	Segmen 4.3
		Fungsi <i>add_driver_file</i>	Fungsi pada <i>model</i> yang bertugas untuk memasukan semua data pada <i>array</i> kedalam <i>database</i>	Segmen 4.4
		Fungsi <i>template_driver_download</i>	Fungsi yang digunakan untuk mengunduh <i>file template</i> data <i>driver</i>	Segmen 4.5 Segmen 4.6
<i>Manage Truck</i>	<i>List Truck</i>	Fungsi <i>get_all_truck_detail</i>	Fungsi yang digunakan untuk mengambil data detail <i>truck</i> pada <i>database</i>	-
	<i>Import New Truck</i>	Fungsi <i>add_truck_file</i>	Fungsi yang digunakan untuk menambahkan data <i>truck</i> yan didapat dari hasil baca <i>file excel</i>	-
		Fungsi <i>template_truck_download</i>	Fungsi yang digunakan untuk mengunduh <i>file template</i> data <i>truck</i>	-

Tabel 4.8. Daftar Fungsi/Prosedur/Event pada Fitur *Manage Data* (lanjutan)

<i>Usecase</i>	<i>Menu / Submenu</i>	<i>Fungsi/Procedure /Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Shipment</i>	<i>List Shipment</i>	Fungsi <i>get_all_shipment_detail</i>	Fungsi yang digunakan untuk mengambil data <i>shipment</i>	-
	<i>Import New Shipment</i>	Fungsi <i>add_shipment_file</i>	Fungsi yang digunakan untuk menambahkan data <i>shipment</i>	Segmen 4.7 Segmen 4.8
		Prosedur <i>sp_insert_shipment</i>	Prosedur untuk menambahkan data <i>shipment</i> pada <i>database</i>	Segmen 4.9 Segmen 4.10 Segmen 4.11
		Fungsi <i>template_shipment_download</i>	Fungsi yang digunakan untuk mengunduh <i>file template shipment</i>	-
<i>Manage Delivery Order</i>	<i>List Delivery Order</i>	Fungsi <i>get_do_title</i>	Fungsi yang digunakan untuk mengambil data judul tabel <i>delivery order</i>	Segmen 4.12
		Fungsi <i>get_all_do_detail</i>	Fungsi yang digunakan untuk mengambil data <i>delivery order</i>	-
	<i>Import New Delivery Order</i>	Fungsi <i>add_do_file</i>	Fungsi untuk menambahkan data <i>delivery order</i>	-
		Prosedur <i>sp_insert_deliveryorder</i>	Prosedur untuk menambahkan data <i>delivery order</i>	Segmen 4.13
		Fungsi <i>template_do_download</i>	Fungsi yang digunakan untuk mengunduh <i>file template do</i>	-

Tabel 4.9. Daftar Fungsi/Prosedur/Event pada Fitur *Manage Data* (lanjutan)

<i>Usecase</i>	<i>Menu / Submenu</i>	<i>Fungsi/Procedure /Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Detail Delivery Order</i>	<i>List Detail Delivery Order</i>	Fungsi <i>get_detdo_title</i>	Fungsi untuk mengambil data judul tabel pada tabel <i>detail delivery order</i>	-
		Fungsi <i>get_all_detdo_detail</i>	Fungsi yang digunakan untuk mengambil data <i>driver</i> pada <i>database</i>	-
	<i>Import New Detail Delivery Order</i>	Fungsi <i>save_data_detdelo</i>	Fungsi untuk menyimpan seluruh data pada <i>array detail delivery order</i> pada <i>database</i>	-

Pada menu *list driver* dibutuhkan fungsi *get\_all\_driver* yang bertugas untuk mengambil semua data supir. Fungsi yang dibuat dilengkapi dengan parameter dan nilai *default* parameter. Hal ini bertujuan agar fungsi dapat digunakan untuk mengambil data supir berdasarkan *employee id* supir. Kode fungsi *get\_all\_driver\_detail* dapat dilihat pada Segmen 4.2.

#### Segmen 4.2. Fungsi *get\_all\_driver\_detail*

```

public function get_all_driver_detail($id="", $num=0){
    $dfdb = $this->load->database('default', TRUE);
    $kondisi = "";
    if($num!=0){
        $kondisi = "where employee_id=" . $dbpp->escape($id);
    }

    $sqry = "SELECT * from DRIVER $kondisi order by 2";
    $qqry = $dfdb->query($sqry);
    $isi = $qqry->result_array();

    if($num != 0){
        $json = json_encode($isi);
        return $json; exit;
    }
    $hasil = json_encode($isi);
    return $hasil;
}

```

Pada bagian menu *import data driver*, dibutuhkan fungsi dengan bahasa pemrograman *javascript*. Fungsi ini akan mengambil setiap data hasil *import* yang dilakukan, kemudian menyimpannya kedalam suatu *array*. *Array* tersebut akan diberikan ke bagian *controller* untuk kemudian disimpan ke dalam *database* melalui *model* MDriver dengan fungsi *add\_driver\_file*. Segmen program untuk penambahan data dapat dilihat pada Segmen 4.3. Sedangkan fungsi *add\_driver\_file* terdapat pada Segmen 4.4.

### Segmen 4.3. Ajax Penambahan Data Driver

```

$( "#cmdUpload" ).click(function() {
    var isi      = [[]];
    var ii      = 0;

    $(".dataupload").each(function() {
        var sn      = $(this).find(".Shipment-Number").html();
        var sd      = $(this).find(".Shipment-Date").html();
        var nopol   = $(this).find(".Nopol").html();
        var did     = $(this).find(".DRIVER").html();
        var dname   = $(this).find(".Driver-Name").html();
        var aid     = $(this).find(".ASSISTANT").html();
        var aname   = $(this).find(".Assistant-Name").html();
        var vol     = $(this).find(".VOLUM").html();
        isi[ii]    = [sn, sd, nopol, did, dname, aid, aname, vol];
        ii++;
    });

    var lempar    = JSON.stringify(isi);

    $.ajax({
        url: '<?= base_url("CImport/save_data_shipment") ?>',
        type: 'POST',
        data: {
            datas   : lempar
        },
        success: function(data) {
            if(data=="false") window.location.reload();
            else if(data=="OK") {
                alert("Data Saved Successful !");
                window.location.replace('<?=
base_url("CManage/shipment") ?>');
            }
            else {
                alert("Data Saved Failed ! " + data);
            }
        }
    });
});

```

#### Segmen 4.4. Fungsi *add\_driver\_file*

```
public function add_driver_file($data){
    $dbf = $this->load->database('default', TRUE);
    if (!$adata = json_decode($data)) {
        return "Not JSON..."; exit;
    }

    foreach ($adata as $key => $fdata) {

        $datas = array(
            "EMPLOYEE_ID"      => $fdata[0],
            "NAME"              => $fdata[1],
            "PHONE_NUMBER"     => $fdata[2]
        );
        $dbf->insert("DRIVER", $datas);
    }
    print_r("OK");exit;
}
```

Pengguna yang ingin menambahkan data melalui fitur *import file excel* harus memperhatikan format urutan tabel pada *file excel* yang ingin di-*upload*. Format tabel dapat dilihat dengan mengunggah *file template* yang bersangkutan. Adapun untuk dapat menunggah *file* tersebut, maka dibutuhan fungsi *template\_driver\_download*. Segmen program *template\_driver\_download* tersedia pada Segmen 4.5 dan Segmen 4.6.

#### Segmen 4.5. Fungsi *template\_driver\_download*

```
public function template_driver_download(){
    $name      = "driver_template.xls";
    $path      =
    APPPPATH."libraries/template/template_driver_profile.xls";

    if(is_file($path)){
        if(ini_get('zlib.output_compression')) {
            ini_set('zlib.output_compression', 'Off'); }

        $this->load->helper('file');

        $mime = get_mime_by_extension($path);
        header('Pragma: public');
        header('Expires: 0');
        header('Cache-Control: must-revalidate, post-
        check=0, pre-check=0');
        header('Last-Modified: '.gmdate ('D, d M Y H:i:s',
        filemtime ($path)).' GMT');
```

#### Segmen 4.6. Fungsi *template\_driver\_download* (lanjutan)

```
        header('Cache-Control: private', false);
        header("Content-Type: application/xls");
        header('Content-Disposition: attachment;
filename="'.basename($name).'"');
        header('Content-Transfer-Encoding: binary');
        header('Content-Length: '.filesize($path));
        header('Connection: close');
        readfile($path);
    }
}
```

Pada menu *import new shipment*, terdapat fungsi *add\_shipment\_file*. Fungsi ini bertujuan untuk menambahkan data *shipment* kedalam *database*. Pada bagian ini terdapat dua bagian inti yaitu, memasukan data ke tabel *shipment* sementara dan menjalankan prosedur untuk menambahkan data. Fungsi *add\_shipment\_file* dapat dilihat pada Segmen 4.7 dan Segmen 4.8. Sedangkan isi dari prosedur *sp\_insert\_shipment* dapat dilihat pada Segmen 4.9, Segmen 4.10, dan Segmen 4.11.

#### Segmen 4.7. Fungsi *add\_shipment\_file*

```
public function add_shipment_file($data){
    $dbf = $this->load->database('default', TRUE);
    if (!$adata = json_decode($data)) {
        return "Not JSON...";
        exit; }
    $sqry = "DROP TABLE TEMPSHIPMENT";
    $qqry = $dbf->query($sqry);
    $sqry = "select * Into TEMPSHIPMENT From VSHIPMENTDETAIL
Where 1 = 2";
    $qqry = $dbf->query($sqry);

    foreach ($adata as $key => $fdata) {
        $datas = array(
            "NO_SHIPMENT"           => $fdata[0],
            "SHIPMENT_DATE"        => $fdata[1],
            "PLAT_NOMOR"           => $fdata[2],
            "EMPLOYEE_ID"          => $fdata[3],
            "DRIVER_NAME"          => $fdata[4],
            "ASSISTANT_ID"         => $fdata[5],
            "ASSISTANT_NAME"       => $fdata[6],
            "VOLUM"                => $fdata[7],
            "ID_TRUCK"             => "1" );
        $dbf->insert("TEMPSHIPMENT", $datas);
    }
    $strnya = "begin try
exec dbo.sp_insert_shipment end try" ;
```

#### Segmen 4.8. Fungsi *add\_shipment\_file* (lanjutan)

```
begin catch
select ERROR_MESSAGE() AS error
end catch";

$query    = $dbf->query($strnya);
$fetch    = $query->row();

print_r("OK");exit;
}
```

#### Segmen 4.9. Prosedur *sp\_insert\_shipment*

```
ALTER PROCEDURE [dbo].[SP_INSERT_SHIPMENT]
AS
BEGIN
SET NOCOUNT ON;

DECLARE
@newidtruck int,
@newiddriver varchar(50),@shipment_number int,
@date date, @nopol varchar(15),
@driver_id varchar(50),@driver_name varchar(50) ,
@assistant_id varchar(50),
@assistant_name varchar(50),@volum float,
@nErNo int, @sErrMsg varchar(255),
@insert_error int =0 ,@erName varchar(255)

DECLARE curr CURSOR FOR
SELECT NO_SHIPMENT, SHIPMENT_DATE, PLAT_NOMOR,
EMPLOYEE_ID, DRIVER_NAME,
ASSISTANT_ID, ASSISTANT_NAME,VOLUM FROM TEMPSHIPMENT

OPEN curr
FETCH NEXT FROM curr INTO @shipment_number, @date, @nopol,
@driver_id ,@driver_name,
@assistant_id ,@assistant_name , @volum

IF @@FETCH_STATUS <> 0
PRINT ' <<None>>'
WHILE @@FETCH_STATUS = 0
BEGIN

if not exists ( select * from SHIPMENT where
NO_SHIPMENT = @shipment_number )
begin

-----Check Data Truck sudah ada atau belum -----
if not exists ( select * from TRUCK where PLAT_NOMOR = @nopol
)
begin
```

#### Segmen 4.10. Prosedur *sp\_insert\_shipment* (lanjutan)

```

insert into TRUCK (PLAT_NOMOR) values (@nopol)
      SET @newidtruck = (select top 1 ID_TRUCK from
TRUCK where PLAT_NOMOR = @nopol)
      print @newidtruck
end
else
begin
      SET @newidtruck = (select top 1 ID_TRUCK from
TRUCK where PLAT_NOMOR = @nopol )
      print @newidtruck
end
DECLARE @a varchar(MAX), @pass varchar(MAX)
SET @a = CONVERT(NVARCHAR(32) , (select
SUBSTRING(master.dbo.fn_varbintohexstr(HashBytes('MD5',
'123')), 3, 32)), 2)
SET @pass = master.dbo.fn_varbintohexsubstring(0,
HashBytes('SHA1', (@a)), 1, 0)

-----Check Data Driver sudah ada atau belum -----

      if not exists ( select * from DRIVER where
EMPLOYEE_ID = @driver_id )
      begin
            insert into DRIVER (EMPLOYEE_ID, NAME, PASSWORD)
values (@driver_id, @driver_name, @pass)
            print 'Data driver created' ;
      end

-----Check Data Assistant sudah ada atau belum -----

      if not exists ( select * from DRIVER where
EMPLOYEE_ID = @assistant_id )
      begin
            insert into DRIVER (EMPLOYEE_ID, NAME, PASSWORD)
values (@assistant_id, @assistant_name, @pass)
            print 'Data Assistant created' ;
      end

      insert into
SHIPMENT (NO_SHIPMENT, ID_TRUCK, EMPLOYEE_ID, ASSISTANT_ID, SHIPMEN
T_DATE, VOLUM)

      values (@shipment_number, @newidtruck, @driver_id, @assistant_
id, @date, @volum)

end
else
BEGIN
      SET @nErNo          = 207000
      SET @sErrMsg        = 'This shipment number is
already exist !'
      SET @erName         = 'Insert Shipment Number '
+ CONVERT(nvarchar, @shipment_number)

```

#### Segmen 4.11. Prosedur *sp\_insert\_shipment* (lanjutan)

```
                INSERT INTO
upload_log (NAME, ERROR_CODE, ERROR_MSG) SELECT
@erName, @nErNo, @sErrMsg

                END

                FETCH NEXT FROM curr INTO @shipment_number, @date, @nopol,
@driver_id ,@driver_name,
@assistant_id ,@assistant_name, @volum

                end
                CLOSE curr
                DEALLOCATE curr

                insert into group_shipment (ID_TRUCK, EMPLOYEE_ID,
ASSISTANT_ID, SHIPMENT_DATE)
                select distinct ID_TRUCK, EMPLOYEE_ID,
ASSISTANT_ID, SHIPMENT_DATE from SHIPMENT GROUP BY ID_TRUCK,
EMPLOYEE_ID, ASSISTANT_ID, SHIPMENT_DATE
                HAVING SHIPMENT_DATE =@date
                order by SHIPMENT_DATE

                update a set a.group_shipment_id = b.group_shipment_id
                from SHIPMENT a join group_shipment b on
a.ID_TRUCK=b.id_truck and a.EMPLOYEE_ID=b.employee_id and
a.ASSISTANT_ID = b.assistant_id and
a.SHIPMENT_DATE=b.shipment_date
                where a.SHIPMENT_DATE =@date

                select * from GROUP_SHIPMENT

END
```

Menu pada *list delivery order*, memuat banyak kolom pada tabel. Untuk memudahkan dalam menampilkan setiap kolom yang ada, maka dibutuhkan suatu fungsi yang bertugas untuk mendapatkan semua judul kolom yang ada pada tabel tersebut. Segmen fungsi tersebut terdapat pada Segmen 4.12.

#### Segmen 4.12. Fungsi *get\_do\_title*

```
public function get_do_title(){
    $dfdb = $this->load->database('default', TRUE);
    $sqry = "SELECT column_name FROM
INFORMATION_SCHEMA.COLUMNS where TABLE_NAME =
'DELIVERY_ORDER'";
    $qqry = $dfdb->query($sqry);
    $isi = $qqry->result_array();
    $hasil = json_encode($isi);
    return $hasil; }
```

Data *delivery order* memuat data yang cukup kompleks. Oleh karena itu dibutuhkan suatu prosedur pada *database* yang berfungsi untuk mengolah dan menyimpan data *delivery order*. Isi dari prosedur dapat dilihat pada Segmen 4.13.

#### Segmen 4.13. Prosedur *sp\_insert\_deliveryorder*

```

ALTER PROCEDURE [dbo].[SP_INSERT_DELIVERYORDER]
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE
        @datenow datetime;

    set @datenow = (select
convert(char(8),CURRENT_TIMESTAMP,112))

    ---INSERT DO -----
    INSERT      DELIVERY_ORDER (
    DELIVERY_ORDER,NO_SHIPMENT,FAKTUR_NUMBER,SHIP_TO,
SHIP_TO_NAME,ADDRESS,TO_CITY,TO_REGION,TO_ADDRESS,TO_TELEPHONE_
NUMBER,
    PRIORITY,VOLUM,INFORMATION,CREATED_AT,US_UPDATE
    )
    SELECT DELIVERY_ORDER,NO_SHIPMENT,FAKTUR_NUMBER,SHIP_TO,
SHIP_TO_NAME,ADDRESS,TO_CITY,TO_REGION,TO_ADDRESS,TO_TELEPHONE_
NUMBER,
    PRIORITY,VOLUM,INFORMATION,CREATED_AT,US_UPDATE FROM TEMPDO
WHERE convert(char(8),CREATED_AT,112) =@datenow

    ----- UPDATE WAREHOUSE INFORMATION -----
    update a set
    WH_ADDRESS = b.WH_ADDRESS,
    WH_CITY      = b.WH_CITY,
    WH_TELP     = b.WH_TELP
    from SHIPMENT a
    RIGHT JOIN (SELECT NO_SHIPMENT, WH_ADDRESS , WH_CITY ,
WH_TELP FROM TEMPDO WHERE convert(char(8),CREATED_AT,112)
=@datenow
    GROUP BY no_shipment, WH_ADDRESS , WH_CITY , WH_TELP ) b on
a.NO_SHIPMENT= b.NO_SHIPMENT

END

```

#### 4.2.2.2. Fitur *Processing Address*

Fitur ini digunakan untuk menghitung dan mengurutkan setiap alamat pada suatu pengiriman. Untuk dapat mengurutkan alamat tersebut, terdapat empat proses besar yang dilakukan. Keempat proses tersebut dilakukan pada menu yang

sama, yakni menu *detail process*. Tabel *list* fungsi yang dipakai dapat dilihat pada Tabel 4.10.

Tabel 4.10. Daftar fungsi/prosedur/event pada Fitur *Processing Address*

<i>Usecase</i>	<i>Flowchart</i>	<i>Menu / Submenu</i>	<i>Fungsi / Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Process</i>	<i>Proses 1 Process Address</i>	<i>Detail Process</i>	Penambahan alamat dari <i>google maps API</i>	Fungsi digunakan untuk mencari alamat yang ada pada <i>google maps</i> .	Segmen 4.14
			Fungsi <i>add_location</i>	Fungsi yang digunakan untuk menyimpan alamat yang didapat dari <i>google maps API</i> .	Segmen 4.15
	<i>Proses 2 K-means Clustering</i>		Fungsi Pemanggilan Algoritma <i>K-means Clustering</i>	Fungsi digunakan untuk menjalankan algoritma <i>clustering</i>	Segmen 4.16 Segmen 4.17
	<i>Proses 3 Calculate Route</i>		Fungsi untuk menghitung rute	Barisan program yang dijalankan untuk menghitung rute	Segmen 4.18 Segmen 4.19 Segmen 4.20 Segmen 4.21 Segmen 4.22 Segmen 4.23
	<i>Proses 4 Insert Priority DO</i>		Fungsi untuk memasukan DO prioritas	Fungsi yang dijalankan untuk menggabungkan pengiriman yang memiliki permintaan khusus	Segmen 4.24 Segmen 4.25 Segmen 4.26

Penambahan alamat dari *google maps API* dilakukan ketika data belum memiliki alamat yang berasal dari *response google maps API*. Hal ini ditandai dengan nilai *null* atau 0 pada data *address\_added* alamat yang bersangkutan.

Segmen program untuk pemanggilan *API* tersebut terdapat pada Segmen 4.14. Untuk setiap *response* yang didapat, disimpan dengan menggunakan fungsi *add\_location*. Kode untuk fungsi *add\_location* terdapat pada Segmen 4.15.

#### Segmen 4.14. Penambahan alamat dari *google maps API*

```

if($fdata['ADDRESS_ADDED'] == "" || $fdata['ADDRESS_ADDED'] ==
0 ) {
    $startpos1 = str_replace(" ", "+", $fdata['ADDRESS']);

    $json1="https://maps.googleapis.com/maps/api/geocode/json?
address=".$startpos1."&key=[YOUR_KEY]";
    $data1 = json_decode(file_get_contents($json1), true);

    if($data1['status'] != "ZERO_RESULTS"){
        foreach ($data1['results'] as $key => $value) {
            $alamat = $value['formatted_address'] ;
            $lat = $value['geometry']['location']['lat'];
            $lng = $value['geometry']['location']['lng'];
        }

        echo '<script type="text/javascript">',
        'addlocation('.$no.', '.$lat.', '.$lng.', '.$fdata['DELIVERY_
ORDER'].', "'.$fdata['WH_CITY'].', '.$alamat.', '.$fdata['
GROUP_SHIPMENT_ID'].');',
        '</script>'
        echo '
        <td class="address">'.$alamat .'</td>
        <td class="lat">'.$lat .'</td>
        <td class="lng"> '.$lng .'</td>;
        }
    }

```

#### Segmen 4.15. Fungsi *Add\_location*

```

function addlocation(number, lat, lng, nodo, city, address, group){
    $.ajax({url: '<?=' . base_url("CProcess/save_address") . '?>',
type: 'POST',
    data: {
        dlat : lat , dlng : lng
        , nodo : nodo , dcity : city
        , address : address , gr : group
    },
    success: function(data){
        if(data=="false")
        window.location.reload();
    } });
    number += ' ' ;
    loc = new Array(number, lat, lng);
    locations.push(loc); tot++;
    totlat += lat; totlng += lng;
}

```

Setelah setiap alamat memiliki alamat yang berasal dari *response google maps API*, maka data tersebut siap untuk di *clustering*. Metode *cluster* yang dipakai yaitu metode *K-means*. Pada Segmen 4.16 dan Segmen 4.17 terdapat segmen program untuk menjalankan pemanggilan algoritma *clustering k-means*.

#### Segmen 4.16. Fungsi Pemanggilan Algoritma *K-means Clustering*

```

$points = array();
$myarrays = json_decode($nodetagenda,true);
$desArray = array();

$alloriginsno = '';
foreach($myarrays as $rdata => $fdata){

    array_push($desArray,$fdata['ID_DESTINATION']);

    //Insert to kmeans array
    $coord =
    array(($fdata['LAT']*100)*($fdata['LAT']*100),($fdata['LNG
    ']*100)*($fdata['LNG']*100));
    Array_push($points,$coord);
    $alloriginsno =
    $alloriginsno.'|'.$fdata['LAT'].'|'.$fdata['LNG'];
}
// create a 2-dimensions space
$space = new KMeans\Space(2);

// add points to space
foreach ($points as $coordinates)
    $space->addPoint($coordinates);

// display the cluster centers and attached points
$tot = 8;
$count =0 ;
$dx = 0;
$dt = new DateTime;
$dt->setTime(0, 0);

$dt->add(new DateInterval('P0Y1DT7H00M'));

$_SESSION['epoch']= date_format($dt, 'U');

while($tot > 8 - ($no/2) + $dx)
{
    $count ++;
    if($count % 10 == 0 ) $dx++;
    $clusters = $space->solve(8);
    $tot=8;
    foreach ($clusters as $i => $cluster)
    {
        if(count($cluster)!=0)$tot--;
    }
}

```

#### Segmen 4.17. Fungsi Pemanggilan Algoritma *K-means Clustering* (lanjutan)

```
}
echo "<br>Banyak pengulangan : " . $count . "<br>";
$clusterarray = '';

$c= 0;
foreach ($clusters as $i => $cluster)
{
    if(count($cluster)!=0)
    {
        $tot--;
        $clusterarray =
        $clusterarray.'|'.(sqrt($cluster[0]) /-100).'|'.
        (sqrt($cluster[1])/100);
    }
    printf("<br>Cluster %s [%.4f,%.4f]: %d points\n",
    $i,sqrt($cluster[0])/-100,sqrt($cluster[1])/100,
    count($cluster));
    foreach ($cluster as $point)
        printf(' [%.4f,%.4f]', sqrt($point[0])/-100,
        sqrt($point[1])/100);
    }
}
```

Setelah data ter-*cluster*, setiap titik *centroid* dari *cluster* digunakan sebagai data untuk perhitungan rute. Selanjutnya, setiap destinasi dalam *cluster* akan dihitung kembali. Segmen dari algoritma terdapat pada Segmen 4.18, Segmen 4.19, Segmen 4.20, Segmen 4.21, Segmen 4.22, dan Segmen 4.23.

#### Segmen 4.18. Fungsi untuk menghitung rute

```
<?php
echo "<br>";
$reqs =
'https://maps.googleapis.com/maps/api/directions/json?origin='
.substr($whadr, 0, -10).'&destination='.substr($whadr, 0, -
10).'&waypoints=optimize:true|'.substr($clusterarray,1,-
1).'&key=[YOUR_KEY]';
echo $reqs;

$data2 = json_decode(file_get_contents($reqs),true);

$totBest =0;
echo "<br>";
echo "<br>";
echo "<br>";
```

#### Segmen 4.19. Fungsi untuk menghitung rute (lanjutan)

```
$prevroutelat = '';
$nextroutelat = '';
$nextroutelng = '';
$clustid = '';

$routeResult = array(array());

foreach ($data2['routes'] as $key2 => $value2)
{
    for($i =0 ; $i < count($value2["waypoint_order"]) ;$i++)
    {
        $counter = -1;
        Echo '<br><br>'.$value2["waypoint_order"][$i].'<br>';
        $ctot = 0;
        foreach ($clusters as $c => $cluster)
        {
            $counter++;
            if(count($cluster)==0)
            {
                if($value2["waypoint_order"][$i] == $ctot)
                {
                    foreach ($cluster as $point)
                        printf('[[%.4f,%.4f]]',
sqrt($point[0])/-100, sqrt($point[1])/100);
                }
            }
            else
            {
                if($value2["waypoint_order"][$i] == $ctot)
                {
                    echo "COUNTER ".$counter .'\<BR>';
                    $prevlat = '';
                    $prevlng = '';
                    $destroute = '';
                    $diff =0;
                    $totrealdiff = 0;
                    foreach ($cluster as $point)
                    {
                        $totrealdiff++;
                        if($prevlat == $point[0] && $prevlng
== $point[1]){
                            else{
                                $diff++;
                                $prevlat = $point[0];
                                $prevlng = $point[1];
                                $destroute =
$destroute.'|'.(sqrt($point[0])/-
100).','.(sqrt($point[1])/100);
                            }
                        printf('[[%.4f,%.4f]]',sqrt($point[0])/-
100, sqrt($point[1])/100);
                    }
                }
            }
        }
    }
}

// cari titik centroid berikutnya
```

## Segmen 4.20. Fungsi untuk menghitung rute (lanjutan)

```
$dy =0;
$temp =0;

for($j =0 ; $j < 8 ;$j++){

    if(count($clusters[$j])==0){}

    else {
        if(($i+1)<count($value2["waypoint_order"]))
        {
            if($value2["waypoint_order"][$i+1] == $temp )
            {
                $nextrouatelat = sqrt($clusters[$j][0])/100;
                $nextrouatelng = sqrt($clusters[$j][1])/100;
                if($i == 0)
                {
                    $prevrouatelat = substr($whadr, 0, -10);
                    $prevrouatelng = '';
                    echo "<br>PREV : ".$whadr;
                }
                else {
                    echo "<br>PREV".$prevrouatelat. ',' .
$prevrouatelng;
                }
                echo "<br>NEXT".$nextrouatelat. ',' .
$nextrouatelng;
                break;
            }
            $temp++;
        }
        else {
            echo "<br>PREV".$prevrouatelat. ',' .
$prevrouatelng;
            echo "<br>NEXT : ".$whadr;
            $nextrouatelat = substr($whadr, 0, -10);
            $nextrouatelng = '';
            break;
        }
    }
}
if($diff>1 )
{
    if($ctot ==0)
    {
        $reqs1 =
'https://maps.googleapis.com/maps/api/directions/json?origin='
.$prevrouatelat.','.$prevrouatelng.'&destination='.$nextrouatelat
.','.$nextrouatelng.'&waypoints=optimize:true|'.substr($destrout
e,1,-1).'&key=[YOUR_KEY]';

        $data21 =
json_decode(file_get_contents($reqs1),true);
    }
}
```

## Segmen 4.21. Fungsi untuk menghitung rute (lanjutan)

```
foreach ($data21['routes'] as $key21 => $value21)
{
    echo "<br>";
    for($i = 0 ; $i < count($value21["waypoint_order"])
; $i++)
    {
        echo $value21["waypoint_order"][$i] . '-';
        $expp = explode('|', substr($destroute,1,-1));
        $expget = $expp[$value21["waypoint_order"][$i]] ;
        $expp1 = explode(',', $expget);

        if(($i) == count($value21["waypoint_order"])-1 )
        {
            $expp = explode('|', substr($destroute,1,-1));
            $expget = $expp[$value21["waypoint_order"][$i]] ;
            $expp1 = explode(',', $expget);
            $prevroutelat = $expp1[0];
            $prevroutelng = $expp1[1];
        }

        $sids = 0;
        $array = json_decode($nodetagenda,true);

        foreach($array as $rdata => $vdata){
            echo round($vdata['LNG'],4) . '-'. round($expp1[1],4)
.'<br>';

            if(round($vdata['LAT'],4) == round($expp1[0],4) &&
round($vdata['LNG'],5) == round($expp1[1],4))
            {
                $sids= $vdata['ID_DESTINATION'] ;
                break;
            }
        }
        echo $sids . '-';
        $newroute= array($expp1[0], $expp1[1],$sids) ;
        array_push($routeResult, $newroute);
    }
}
else {
    $reqs1 =
'https://maps.googleapis.com/maps/api/directions/json?origin='
.$prevroutelat.'.'. $prevroutelng.'&destination='.$nextroutelat
.'.'. $nextroutelng.'&waypoints=optimize:true|'.substr($destrou
te,1,-1).'&key=[YOUR_KEY]';
    $data21 = json_decode(file_get_contents($reqs1),true);
    echo '<br>'. $reqs1;

    foreach ($data21['routes'] as $key21 => $value21)
    {
        echo "<br>";

        for($i = 0 ; $i < count($value21["waypoint_order"]) ; $i++){
```

## Segmen 4.22. Fungsi untuk menghitung rute (lanjutan)

```
echo $value21["waypoint_order"][$ii] . '-';
$expp = explode('|', substr($destroute, 1, -1));
$expget = $expp[$value21["waypoint_order"][$ii]] ;
$expp1 = explode(',', $expget);

if((($ii) == count($value21["waypoint_order"])-1 )
{
    $prevroutelat = $expp1[0];
    $prevroutelng = $expp1[1];
}
$idds = 0;
$array = json_decode($nodetagenda, true);

foreach($array as $rdata => $vdata){
echo round($vdata['LNG'], 4) . '-'. round($expp1[1], 4) . '<br>';

if(round($vdata['LAT'], 4) == round($expp1[0], 4) &&
round($vdata['LNG'], 4) == round($expp1[1], 4))
{
    $idds= $vdata['ID_DESTINATION'] ;
    break;
}
}
echo $idds . '-';
$newroute= array($expp1[0], $expp1[1], $idds) ;

    array_push($routeResult, $newroute);
}
}
}
else {
    //Cluster tidak mempunyai data
    foreach ($clusters[$counter] as $points)
    {
        printf('【%.4f,%.4f】', sqrt($points[0])/-100,
sqrt($points[1])/100);
        $idds = 0;
        $array = json_decode($nodetagenda, true);

        foreach($array as $rdata => $vdata){
        if(round($vdata['LAT'], 4) == round(sqrt($points[0])/-100, 4) &&
round($vdata['LNG'], 4) == round(sqrt($points[1])/100, 4))
        {
            $idds= $vdata['ID_DESTINATION'] ;
            break;
        }
        }
        $newroute= array(sqrt($points[0])/-100, sqrt($points[1])/100
, $idds) ;
        array_push($routeResult, $newroute);
        break;
    }
    echo $clusters[$counter][0][0];
    echo "<BR>". $totrealdiff. '<BR>';
}
```

### Segmen 4.23. Fungsi untuk menghitung rute (lanjutan)

```
$prevroutelat = sqrt($point[0])/-100;
$prevroutelng = sqrt($point[1])/100;
    }
  }
  $ctot++;
}
}
}
```

Tahap terakhir yaitu, memasukan data destinasi yang memiliki permintaan khusus kedalam *array* destinasi yang sudah terurut sebelumnya.

Fungsi untuk memasukan data tersebut terdapat pada Segmen 4.24, Segmen 4.25, dan Segmen 4.26.

### Segmen 4.24. Fungsi untuk memasukan DO prioritas

```
startupdate :

$dt = new DateTime;
$dt->setTime(0, 0);

$dt->add(new DateInterval('POY1DT7H00M'));
$_SESSION['epoch']= date_format($dt, 'U');
$_SESSION['plusEpoch'] =60 ;
$routeTime = array(array());
$counting = 0 ;

for($vv= 0 ; $vv< count($routeResult) ;$vv++)
{
    $stringdate = $dt->format('H:i:s');
    $_SESSION['epoch'] =date_format($dt, 'U');
    if($vv == 0 ){
        $newroutetime=
array((float)0, (float)$routeResult[$vv][2], (float)0, (float)0, (
float)$routeResult[$vv][0], (float)$routeResult[$vv][1], (float)
geteta(substr($whadr, 0, -
10), $routeResult[$vv][0].','. $routeResult[$vv][1],
$_SESSION['epoch'], substr($whadr, 0, -
10)), $stringdate, $_SESSION['epoch']) ;
    }
    else {
        $newroutetime= array((float)$routeResult[$vv-
1][2], (float)$routeResult[$vv][2], (float)$routeResult[$vv-
1][0], (float)$routeResult[$vv-
1][1], (float)$routeResult[$vv][0], (float)$routeResult[$vv][1],
(float)geteta($routeResult[$vv-1][0].','. $routeResult[$vv-
1][1], $routeResult[$vv][0].','. $routeResult[$vv][1],
```

#### Segmen 4.25. Fungsi untuk memasukan DO prioritas (lanjutan)

```
$_SESSION['epoch'],substr($whadr, 0, -
10)), $stringdate, $_SESSION['epoch']) ;
}
$dt->add(new DateInterval('P0Y0DT0H'.
$_SESSION['plusEpoch'].'M'));
array_push($routeTime, $newroutetime);
}
$stringdate = $dt->format('H:i:s');
$newroutetime= array((float)$routeResult[count($routeResult) -
1][2], (float) -1, (float)$routeResult[count($routeResult) -
1][0], (float)$routeResult[count($routeResult) -1][1], (float)-
1, (float)-1, (float)geteta($routeResult[count($routeResult) -
1][0].','. $routeResult[count($routeResult) -
1][1],substr($whadr, 0, -10),
$_SESSION['epoch'],substr($whadr, 0, -
10)), $stringdate, $_SESSION['epoch']) ;

array_push($routeTime, $newroutetime);
array_splice($routeTime, 0, 1);

$arrcons = json_decode($withdetagenda,true);
for($nd =0 ; $nd < count($arrcons) ;$nd++)
{
    $hasilArray = array(array());
    $donebefore = false;
    $jumpUpdate = false;
    $trysearch = false;
    $co =0;

    for( $ndd =0 ;$ndd < count($routeTime);$ndd ++)
    {
        if((float)$routeTime[$ndd][0] ==
$arrcons[$nd]['ID_DESTINATION'])
        {
            $donebefore = true;
            break;
        }
    }
    if($donebefore == false)
    {
        for( $nds =0 ;$nds < count($routeTime);$nds ++)
        {
            $dteta = new DateTime;
            $dteta->setTime(0, $routeTime[$nds][6]);

            $val =
is_null($arrcons[$nd]['START_AVAIL_DELIVERY_TIME']) ?
'00:00:00' : $arrcons[$nd]['START_AVAIL_DELIVERY_TIME'];
            echo 'VAL:'. $val.'  
';

            if($routeTime[count($routeTime) -1][7] <
$arrcons[$nd]['START_AVAIL_DELIVERY_TIME'] &&
$arrcons[$nd]['START_AVAIL_DELIVERY_TIME'] != null)
            {
                //Masukan ke destinasi paling akhir
            }
        }
    }
}
```

#### Segmen 4.26. Fungsi untuk memasukan DO prioritas (lanjutan)

```

        $newroute=
array((float)$arrcons[$nd]['LAT'],(float)$arrcons[$nd]['LNG']
,(float)$arrcons[$nd]['ID_DESTINATION']);
        array_push($routeResult, $newroute);
        $jumpUpdate = true;
        break;
    }
    else if($val < $routeTime[$nds][7]    &&
$routeTime[$nds][7] < $arrcons[$nd]['END_AVAIL_DELIVERY_TIME']
)
    {
        $trysearch = true;
        $secs = strtotime("00:00:00")-
strtotime("00:00:00");
        echo '<br>'. $val . '<<<' . date("H:i:s", (
strtotime($routeTime[$nds][7]) + $secs)). 'XX' .
$arrcons[$nd]['END_AVAIL_DELIVERY_TIME']. ' --  ';

        $tott =
geteta($routeTime[$nds][2].','. $routeTime[$nds][3],
$arrcons[$nd]['LAT'].','.
$arrcons[$nd]['LNG'], $routeTime[$nds][8], substr($whadr, 0, -
10), substr($whadr, 0, -10))+ geteta($arrcons[$nd]['LAT'].','.
$arrcons[$nd]['LNG'], $routeTime[$nds][4].','. $routeTime[$nds][
5], $routeTime[$nds][8] , substr($whadr, 0, -10));

        echo $tott. '<br>';
        $newroute=
array((float)$arrcons[$nd]['LAT'],(float)$arrcons[$nd]['LNG']
,(float)$arrcons[$nd]['ID_DESTINATION'], $tott
,$routeTime[$nds][0], $nds);

        array_push($hasilArray, $newroute );
    }
    if($trysearch == true && $nds == ( count($routeTime)-1))
    {
        array_splice($hasilArray, 0, 1);
        $newwAr = array(array());
        $newwAr = sort_2d_array($hasilArray, 3, "ASC");

        $newroute= array((float)$newwAr[0][0],(float)$newwAr[0][1]
,(float)$newwAr[0][2]);
        $neww = array($newroute);
        ECHO $newwAr[0][5] .'URUTAN E DE';
        array_insert($routeResult, $newwAr[0][5], $neww);

        $jumpUpdate = true;
    }
}
if($jumpUpdate)
    goto startupdate;
}
$co++;
}

```

### 4.2.2.3. Fitur Agenda

Fitur agenda digunakan untuk melihat detail agenda pengiriman. Pada fitur ini terdapat menu untuk melihat *list* agenda, *detail* dari agenda yang dipilih, serta *list* barang yang dikirimkan pada alamat yang dipilih. Daftar *event* yang terjadi pada fitur agenda dapat dilihat pada Tabel 4.11.

Tabel 4.11. Daftar Fungsi/Prosedur/Event pada Fitur Agenda

<i>Usecase</i>	Menu / Submenu	Fungsi/ <i>Procedure</i> / <i>Event</i>	Keterangan	Segmen
<i>Manage Agenda</i>	<i>List Agenda</i>	Fungsi <i>get_agenda</i>	Fungsi untuk melihat semua agenda	Segmen 4.27
		<i>Query view</i> tabel vagenda	<i>Query</i> untuk membuat tabel <i>view</i> vagenda	Segmen 4.28
	<i>List Detail Agenda</i>	Fungsi <i>get_detail_agenda</i>	Fungsi untuk melihat detail agenda yang dipilih	Segmen 4.29
		<i>Query view</i> tabel vdodetail	<i>Query</i> untuk membuat tabel <i>view</i> vdodetail	-
	<i>List Detail Barang</i>	Fungsi <i>get_all_detdo_detail</i>	Fungsi untuk mendapatkan data <i>detail delivery order</i> secara <i>detail</i>	-
		Fungsi <i>get_all_do_detail</i>	Fungsi untuk mendapatkan data <i>detail delivery order</i>	-

Menu agenda merupakan menu yang disediakan untuk menampilkan daftar agenda yang ada. Data yang didapat dihasilkan dari fungsi *get\_agenda* pada Segmen 4.27. Sedangkan *query* pada *database* untuk vagenda dapat dilihat pada Segmen 4.28 dan vdodetail pada Segmen 4.29.

#### Segmen 4.27. Fungsi *get\_agenda*

```
public function get_agenda($nopol="", $eid="", $sdate="") {
    $dfdb = $this->load->database('default', TRUE);
    $kondisi = "";
    if($nopol!=""){
        $kondisi = "where ID_TRUCK='". $nopol . "' and EMPLOYEE_ID
='". $eid ."' and SHIPMENT_DATE= '". $sdate. "'";
    }
    $sqry = "select * from vagenda $kondisi order by 1";
    $qqry = $dfdb->query($sqry);
    $isi = $qqry->result_array();
    $hasil = json_encode($isi);
    return $hasil;
}
```

#### Segmen 4.28. *Query view* tabel *vagenda*

```
SELECT      a.GROUP_SHIPMENT_ID, b.ID_TRUCK, c.PLAT_NOMOR,
c.SHIPMENT_DATE, c.EMPLOYEE_ID, c.ASSISTANT_ID, c.WH_ADDRESS,
c.WH_CITY, c.WH_TELP, c.volum, c.total, c.SENT, c.VISITED,
c.DEPARTED, c.CANCEL, c.priority, c.address, c.is_priority,
c.DRIVER_NAME, c.ASSISTANT_NAME, c.urutan_rute, c.max_shift,
c.min_shift, c.QTY_TOTAL, c.QTY_SEND, c.QTY_RETURN, CASE WHEN
(SENT + CANCEL = total) THEN 1 ELSE 0 END AS finished
FROM dbo.GROUP_SHIPMENT AS a INNER JOIN
dbo.TRUCK AS b ON a.ID_TRUCK = b.ID_TRUCK INNER JOIN
(SELECT      MAX(time_departure) AS max_shift,
MIN(time_departure) AS min_shift, ISNULL(PLAT_NOMOR,
'default') AS PLAT_NOMOR, SHIPMENT_DATE, EMPLOYEE_ID,
ASSISTANT_ID, WH_ADDRESS, WH_CITY, WH_TELP, SUM(VOLUM) AS
volum, COUNT(SHIPMENT_DATE) AS total, SUM(ISNULL(IS_SENT, 0))
AS SENT, SUM(ISNULL(IS_VISITED, 0)) AS VISITED,
SUM(ISNULL(IS_DEPARTED, 0)) AS DEPARTED, SUM(ISNULL(qty_total,
0)) AS QTY_TOTAL, SUM(ISNULL(qty_send, 0)) AS QTY_SEND,
SUM(ISNULL(qty_return, 0)) AS QTY_RETURN,
SUM(ISNULL(IS_CANCELED, 0)) AS CANCEL, SUM(PRIORITY) AS
priority, SUM(ISNULL(ADDRESS_ADDED, 0)) AS address, CASE WHEN
SUM(priority) > COUNT(volum)
THEN 1 ELSE 0 END AS is_priority, SUM(CASE WHEN URUTAN_RUTE IS
NULL THEN 1 ELSE 0 END) AS urutan_rute, DRIVER_NAME,
ASSISTANT_NAME

FROM dbo.VDODETAIL

GROUP BY WH_ADDRESS, WH_CITY, WH_TELP, ID_TRUCK, PLAT_NOMOR,
EMPLOYEE_ID, ASSISTANT_ID, SHIPMENT_DATE, DRIVER_NAME,
ASSISTANT_NAME) AS c ON
a.ASSISTANT_ID = c.ASSISTANT_ID AND a.EMPLOYEE_ID =
c.EMPLOYEE_ID AND b.PLAT_NOMOR = c.PLAT_NOMOR AND
a.SHIPMENT_DATE = c.SHIPMENT_DATE
```

#### Segmen 4.29. Query view tabel vdodetail

```
SELECT a.DELIVERY_ORDER, a.NO_SHIPMENT, a.ID_ROUTE,
a.START_TIME, a.END_TIME, a.EFFICIENCY, a.URUTAN_RUTE,
a.ETA_TIME, a.USER_REQUEST, a.STATUS, a.MARGIN_LATE,
a.VALIDATE_POINT, a.IS_BREAKTIME, a.BREAKTIME_INFO,
a.FAKTUR_NUMBER, a.SHIP_TO, a.SHIP_TO_NAME, a.ADDRESS,
a.TO_CITY, a.TO_REGION, a.TO_ADDRESS,
a.TO_TELPHONE_NUMBER, a.PRIORITY, a.VOLUM, a.INFORMATION,
a.CREATED_AT, a.US_UPDATE, a.LAST_UPDATE_DATE,
a.START_AVAIL_DELIVERY_TIME, a.END_AVAIL_DELIVERY_TIME,
a.CONFIRM_BEFORE_DELIVERY, b.ID_TRUCK, b.EMPLOYEE_ID,
b.ASSISTANT_ID, b.SHIPMENT_DATE, b.TOT_CANCELED_DELIVERY,
b.TOT_ROUTE, b.KILOMETER_START, b.KILOMETER_END, b.WH_ADDRESS,
b.WH_CITY, b.WH_TELP, c.PLAT_NOMOR, c.TYPE, c.WIDTH, c.HEIGHT,
c.LENGTH, d.DELIVERY_PRIORITY, d.DESCRPTION, b.DRIVER_NAME,
b.ASSISTANT_NAME, a.ADDRESS_ADDED, b.GROUP_SHIPMENT_ID,
a.IS_VISITED, a.IS_DEPARTED, a.IS_CANCELED, a.IS_SENT,
a.CANCEL_REASON, dbo.route.time_departure, dbo.route.eta_time
AS Expr1, e.qty_total, e.qty_send, e.qty_return, e.do,
e.TOTPASANG

FROM dbo.DELIVERY_ORDER AS a
LEFT OUTER JOIN dbo.route ON a.ID_ROUTE = dbo.route.id_route
LEFT OUTER JOIN dbo.VSHIPMENTDETAIL AS b ON a.NO_SHIPMENT =
b.NO_SHIPMENT
LEFT OUTER JOIN dbo.TRUCK AS c ON b.ID_TRUCK = c.ID_TRUCK
LEFT OUTER JOIN dbo.LPRIO AS d ON a.PRIORITY =
d.DELIVERY_PRIORITY
LEFT OUTER JOIN

(
SELECT      SUM(QUANTITY) AS qty_total,
ISNULL(SUM(QUANTITY_SEND), 0) AS qty_send,
ISNULL(SUM(QUANTITY_RETURN), 0) AS
qty_return, DELIVERY_ORDER AS do, SUM(PASANG) AS TOTPASANG

FROM dbo.DO_DETAIL

GROUP BY DELIVERY_ORDER
) AS e

ON e.do = a.DELIVERY_ORDER
```

#### 4.2.2.4. Fitur Dashboard

Fitur *dashboard* merupakan fitur yang dibuat untuk melihat status pengiriman secara keseluruhan pada hari itu. Menu ini dilengkapi dengan grafik untuk memudahkan pengguna dalam melihat *progress* pengiriman barang. Daftar fungsi yang digunakan pada fitur ini terdapat pada Tabel 4.12.

Tabel 4.12. Daftar fungsi/prosedur/event pada Fitur *Dashboard*

<i>Usecase</i>	<i>Menu / Submenu</i>	<i>Fungsi/Procedure /Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Dashboard</i>	<i>View Dashboard</i>	<i>Fungsi get_overview</i>	Fungsi untuk melihat <i>overview</i> pengiriman	-
		<i>Fungsi get_fasttrack_status</i>	Fungsi untuk melihat status pengiriman barang <i>fast track</i>	-
		<i>Query view</i> tabel <i>vstatusfasttrack</i>	<i>Query</i> untuk membuat tabel <i>vstatusfasttrack</i>	Segmen 4.30
		<i>Fungsi get_chart_overall</i>	Fungsi untuk melihat <i>chart progress</i> pengiriman barang	-
		<i>Fungsi get_chart_percentage</i>	Fungsi untuk melihat persentase pengiriman barang	-

Pada menu fitur *dashboard* terdapat fitur untuk dapat melihat status pengiriman barang *fast track*. Adapun fungsi tersebut menggunakan tabel *vstatusfasttrack* sebagai sumber data. *Query* untuk membuat *view vstatusfasttrack* terdapat pada Segmen 4.30.

#### Segmen 4.30. *Query view* tabel *vstatusfasttrack*

```

SELECT      a.DELIVERY_ORDER, a.NO_SHIPMENT, a.SHIPMENT_DATE,
a.PLAT_NOMOR, a.DRIVER_NAME, a.PRIORITY, b.DESCRPTION, CASE
WHEN IS_SENT IS NOT NULL
                THEN 'Ter kirim' ELSE CASE WHEN
IS_CANCELED IS NOT NULL THEN 'Pengiriman di batalkan' ELSE
'Belum Ter kirim' END END AS Status, CASE WHEN IS_SENT IS NOT
NULL
                THEN 0 ELSE CASE WHEN IS_CANCELED IS NOT
NULL THEN 2 ELSE 1 END END AS StatusLevel
FROM        dbo.VDESTINATIONDO AS a LEFT OUTER JOIN
                dbo.LPRIO AS b ON a.PRIORITY =
b.DELIVERY_PRIORITY
WHERE      (a.PRIORITY > 1)
    
```

#### 4.2.2.5. Fitur Report

Pada fitur ini, pengguna dapat melihat *report* dari data pengiriman barang yang ada. Fitur *report* terbagi menjadi empat menu yaitu, *report driver*, *report truck*, *report goods*, dan *report shipment*. Daftar fungsi tersebut terdapat pada Tabel 4.13, Tabel 4.14, dan Tabel 4.15.

Tabel 4.13. Daftar fungsi/prosedur/event pada Fitur Report

<i>Usecase</i>	Menu / Submenu	Fungsi/Procedure /Event	Keterangan	Segmen
<i>Manage Report</i>	<i>Report Driver</i>	Fungsi <i>get_driver_report_full</i>	Fungsi ini digunakan untuk mengambil data <i>report</i> supir	-
		<i>Query view</i> tabel <i>vdriverreport</i>	<i>Query</i> untuk membuat tabel <i>vdriverreport</i>	Segmen 4.31
		<i>Query view</i> tabel <i>vdriverreportfull</i>	<i>Query</i> untuk membuat tabel <i>vdriverreportfull</i>	Segmen 4.32
		Fungsi <i>get_driver_charts</i>	Fungsi ini digunakan untuk membuat grafik <i>report driver</i>	Segmen 4.33
		Fungsi <i>get_chart_driver_workingdays</i>	Fungsi ini digunakan untuk menampilkan grafik <i>report</i> hari kerja supir	-
		Fungsi <i>get_chart_driver_workinghours</i>	Fungsi ini digunakan untuk menampilkan grafik <i>report</i> jam kerja supir	Segmen 4.34
		Event untuk menampilkan grafik <i>driver workinghours</i>	Kode untuk menampilkan grafik halaman <i>report driver</i>	-
		Event untuk memberikan fitur unduh pada tabel	Kode untuk menambahkan fitur <i>download</i> pada tabel <i>datatables</i>	Segmen 4.35

Tabel 4.14. Daftar fungsi/prosedur/event pada Fitur Report (lanjutan)

<i>Usecase</i>	<i>Menu / Submenu</i>	<i>Fungsi/Procedure /Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Report</i>	<i>Report Truck</i>	Fungsi <i>get_truck_report</i>	Fungsi ini digunakan untuk mengambil data <i>report</i> armada	Segmen 4.36 Segmen 4.37
		Prosedur <i>sp_view_report_truck</i>	<i>Query</i> untuk melihat <i>report truck</i>	Segmen 4.38
		Fungsi <i>get_truck_report_bbm</i>	Fungsi yang digunakan untuk mendapatkan <i>report</i> pemakaian BBM	-
		<i>Query view</i> tabel <i>vtruckreportbbm</i>	<i>Query</i> untuk membuat tabel <i>vtruckreportbbm</i>	Segmen 4.39
		Fungsi <i>get_truck_area</i>	Fungsi yang dipakai untuk melihat <i>report</i> mengenai jumlah armada pada suatu area	-
	<i>Report Goods</i>	Event untuk menampilkan grafik <i>world cloud</i>	Kode untuk menampilkan grafik pada halaman <i>report</i>	Segmen 4.40 Segmen 4.41
		Fungsi <i>get_goods_send_report</i>	Fungsi yang dipakai untuk mendapatkan <i>report</i> barang yang terkirim	-
		Fungsi <i>get_return_reason</i>	Fungsi untuk mendapatkan <i>report</i> alasan barang gagal terkirim	Segmen 4.42
		Fungsi <i>get_return_reason_summary</i>	Fungsi untuk mendapatkan <i>report</i> secara keseluruhan alasan me-retur barang	-

Tabel 4.15. Daftar fungsi/prosedur/event pada Fitur *Report* (lanjutan)

<i>Usecase</i>	<i>Menu / Submenu</i>	<i>Fungsi/Procedure /Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Report</i>	<i>Report Shipment</i>	Fungsi <i>get_shipment_area</i>	Fungsi untuk mendapatkan <i>report</i> daerah mana saja yang sering dikunjungi	Segmen 4.43

Pada menu *report driver* dibutuhkan *query* khusus untuk dapat menampilkan data *report* supir. *Query* tersebut disimpan pada *view* *vdriverreport* dan *vdriverreportfull*. Adapun isi dari *query* tersebut dapat dilihat pada Segmen 4.31 dan Segmen 4.32.

#### Segmen 4.31. *Query view* tabel *vdriverreport*

```

SELECT a.totalHrs, a.totShipment, a.avghours, a.monthyear,
b.avgDayperMonth
FROM      (SELECT      SUM(jamkerja / 60) AS totalHrs,
COUNT(SHIPMENT_DATE) AS totShipment, ROUND(AVG(CAST(jamkerja
AS float) / CAST(60 AS float)), 2) AS avghours,
CONVERT(varchar(6), SHIPMENT_DATE, 112) AS monthyear

FROM  dbo.temp_driverreport
WHERE (jamkerja <> 0)
GROUP BY CONVERT(varchar(6), SHIPMENT_DATE, 112)) AS a
LEFT OUTER JOIN

(
SELECT  monthyear, AVG(totDayperMonth) AS avgDayperMonth
FROM
(
SELECT  CONVERT(varchar(6), SHIPMENT_DATE, 112) AS monthyear,
SUM(CASE WHEN isnull(jamkerja, 0) != 0 THEN 1 ELSE 0 END) AS
totDayperMonth, EMPLOYEE_ID
FROM    dbo.temp_driverreport AS temp_driverreport_1
GROUP BY CONVERT(varchar(6), SHIPMENT_DATE, 112), EMPLOYEE_ID
) AS z
GROUP BY monthyear

) AS b

ON a.monthyear = b.monthyear
ORDER BY a.monthyear DESC

```

### Segmen 4.32. Query view tabel vdriverreportfull

```
SELECT a.periode, a.EMPLOYEE_ID, a.NAME, a.workingdays,
a.AverageWorkingHours,
CASE WHEN a.AverageWorkingHours < b.avghours THEN 'Below
Average Hours' ELSE 'Ok' END AS StatusworkingHours,
CASE WHEN a.workingdays < b.avgDaypermonth THEN 'Below Average
Day' ELSE 'Ok' END AS StatusworkingDays
FROM      (SELECT      a_1.periode, a_1.EMPLOYEE_ID, b.NAME,
a_1.workingdays, a_1.AverageWorkingHours

FROM (SELECT      CONVERT(varchar(6), SHIPMENT_DATE, 112) AS
periode, EMPLOYEE_ID, SUM(CASE WHEN isnull(jamkerja, 0) != 0
THEN 1 ELSE 0 END) AS workingdays,
ROUND(AVG(CAST(jamkerja AS float) / CAST(60 AS float)), 2) AS
AverageWorkingHours

FROM dbo.temp_driverreport
GROUP BY EMPLOYEE_ID, CONVERT(varchar(6), SHIPMENT_DATE, 112))
AS a_1 LEFT OUTER JOIN
dbo.DRIVER AS b ON a_1.EMPLOYEE_ID = b.EMPLOYEE_ID) AS a LEFT
OUTER JOIN
dbo.VDRIVERREPORT AS b ON a.periode = b.monthyear
ORDER BY a.periode DESC, a.EMPLOYEE_ID
```

Pada menu *driver report* terdapat grafik yang menampilkan jam kerja seorang supir pada suatu hari. Fungsi yang digunakan untuk mengambil data tersebut yaitu fungsi *get\_driver\_charts*. Potongan kode dari fungsi tersebut terdapat pada Segmen 4.33.

### Segmen 4.33. Fungsi *get\_driver\_charts*

```
public function get_driver_charts($id){
    $dfdb    = $this->load->database('default', TRUE);
    $sqry    = " select SHIPMENT_date date, round(CAST(jamKerja
AS float) / CAST(60 AS float),2) value from temp_driverreport
where replace(EMPLOYEE_ID, ' ', '')= ' ".$id."' and
jamKerja/60 > 1 order by 1 ";

    $qqry    = $dfdb->query($sqry);
    $isi     = $qqry->result_array();
    $hasil   = json_encode($isi);
    return  $hasil;
}
```

Untuk dapat menampilkan suatu grafik, dibutuhkan suatu fungsi bertugas untuk menggambarkan grafik tersebut. Beberapa pengaturan perlu dilakukan

terhadap grafik yang diinginkan. Pengaturan tersebut dapat dilihat pada Segmen 4.34.

#### Segmen 4.34. Grafik *Average Working Hours*

```
<script>
  var chart = AmCharts.makeChart( "chartdiv2", {
    "type": "serial",
    "addClassNames": true,
    "theme": "light",
    "autoMargins": false,
    "marginLeft": 30,
    "marginRight": 8,
    "marginTop": 10,
    "marginBottom": 26,
    "balloon": {
      "adjustBorderColor": false,
      "horizontalPadding": 10,
      "verticalPadding": 8,
      "color": "#ffffff"
    },
    "dataLoader": {
      "url": "<?="
base_url("CReport/get_chart_driver_workinghours") ">",
      "format": "json"
    }, "valueAxes": [ {
      "axisAlpha": 0,
      "position": "left"
    } ], "startDuration": 1,
    "graphs": [ {
      "alphaField": "alpha",
      "balloonText": "<span style='font-size:12px;'>[[title]] in [[category]]:<br><span style='font-size:20px;'>[[value]]</span> [[additional]]</span>",
      "fillAlphas": 1,
      "title": "Average Working Hours",
      "type": "column",
      "valueField": "value",
      "dashLengthField": "dashLengthColumn",
      "fillColors" : "#FF6600",
      "lineColor" : "#FF6600"
    } ], "categoryField": "year",
    "categoryAxis": {
      "gridPosition": "start",
      "axisAlpha": 0,
      "tickLength": 0
    }, "export": {
      "enabled": true
    }
  } );
</script>
```

Selain fitur menggambar grafik, terdapat juga beberapa *report* yang disajikan dalam bentuk tabel. Untuk mempermudah pengguna dalam melakukan *report*, maka disediakan fitur untuk dapat mengunduh data *report* yang ada. Pengaturan untuk penambahan fitur tersebut dapat dilihat pada Segmen 4.35.

#### Segmen 4.35. *Event* fitur *export* data dalam tabel

```
$('#datatables').dataTable( {  
  Responsive      : true,  
  "aLengthMenu": [5, 10, 15, 20],  
  Dom              : 'B<"clear">lfrtip',  
  buttons: [  
    {  
      extend: 'excel',  
      title: 'Average Working Day per Month'  
    }, {  
      extend: 'print',  
      title: 'Average Working Day per Month'  
    }  
  ]  
} );
```

Pada menu *report truck* dibutuhkan fungsi untuk membantu menampilkan data *report* yang ada. Fungsi tersebut dapat dilihat pada Segmen 4.36. Selain itu, fungsi ini membutuhkan suatu prosedur, yang bernama prosedur *sp\_view\_report\_truck*, untuk memudahkan dalam mendapatkan data *report*. *Isi query* prosedur terdapat pada Segmen 4.38 dan Segmen 4.37.

#### Segmen 4.36. Fungsi *get\_truck\_report*

```
public function get_truck_report($type)  
{  
  $dfdb = $this->load->database('default', TRUE);  
  if($type=='KM')  
  {  
    $type1 = "'round(AVG(km_end- km_start),2) AverageKM'";  
    $type2 = "'sum(AverageKM) '";  
  }  
  else if($type== 'used')  
  {  
    $type1= "'count(id_truck)used'";  
    $type2= "'sum(used) '";  
  }  
}
```

### Segmen 4.37. Fungsi *get\_truck\_report* (lanjutan)

```
$gsid = '[' + ".date("Y")." + "], [" + ".date("Y").(date("m")-1)." + "], [" + ".date("Y").(date("m"))." + "]'";

$sqry = "begin
exec dbo.SP_VIEW_REPORT_TRUCK $type1,$type2, $gsid
end ";

$qqry = $dfdb->query($sqry);
$isi = $qqry->result_array();

$hasil = json_encode($isi);
return $hasil;
}
```

### Segmen 4.38. Prosedur *sp\_view\_report\_truck*

```
ALTER PROCEDURE [dbo].[SP_VIEW_REPORT_TRUCK] @type1
varchar(40),@type2 varchar(40),@str varchar(100)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @query NVARCHAR(MAX)
    SET @query = '
select y.PLAT_NOMOR , x.* from (
select * from (
select id_truck,'+@type1+', CONVERT(varchar(6), DATE, 112)
Periode from truck_usage
where CONVERT(varchar(6), DATE, 112) >= CONVERT(varchar(6),
GETDATE(), 112) -1
group by id_truck, CONVERT(varchar(6), DATE, 112) UNION ALL
select id_truck , '+@type1+', CONVERT(varchar(4), DATE, 112)
Periode
from truck_usage group by id_truck, CONVERT(varchar(4), DATE,
112)
) as c
PIVOT
(
'+@type2+'
FOR Periode IN (' + @str + ')
) AS PivotTable )
x left join TRUCK y on x.id_truck = y.ID_TRUCK
order by 2';

execute(@query) ;
END
```

Selain fitur untuk menampilkan data penggunaan armada, juga terdapat fitur untuk menampilkan data pemakaian Bahan Bakar Minyak (BBM). Tetapi

untuk menampilkan hasil *report* tersebut tidaklah mudah. Oleh karena itu, dibuatlah *query view* *vtruckreportbbm* untuk membantu dalam menampilkan data *report*. Isi *query* tersebut terdapat pada Segmen 4.39.

#### Segmen 4.39. *Query view* tabel *vtruckreportbbm*

```
SELECT TOP (100) PERCENT i.ID_BBM, i.hari,
i.averageVolume, i.kmtempuh, j.LITRE, ROUND(i.kmtempuh /
j.LITRE, 2) AS averageKmperLitre
FROM
  (SELECT      ID_BBM, COUNT(ID_BBM) AS hari, AVG(VOLUME)
AS averageVolume, SUM(tempuh) AS kmtempuh
  FROM
    (SELECT      a.ID_BBM, a.ID_TRUCK, a.EMPLOYEE_ID,
a.DATE_START, a.DATE_END, a.LITRE, b.KM_END, b.KM_START,
b.KM_END - b.KM_START AS tempuh, b.DATE, b.VOLUME
  FROM
    dbo.BBM AS a INNER JOIN
    dbo.TRUCK_USAGE AS b ON a.ID_TRUCK = b.ID_TRUCK AND
CONVERT(varchar(10), a.DATE_END, 112) >= CONVERT(varchar(10),
b.DATE, 112) AND CONVERT(varchar(10),
a.DATE_START, 112) <= CONVERT(varchar(10), b.DATE, 112))
  AS x
  GROUP BY ID_BBM) AS i LEFT OUTER JOIN
  dbo.BBM AS j ON j.ID_BBM = i.ID_BBM
ORDER BY averageKmperLitre
```

Submenu berikutnya yaitu *report goods*. Pada bagian ini akan ditampilkan grafik dengan model *world cloud*. Adapun segmen dari program tersebut terdapat pada Segmen 4.40 dan Segmen 4.41. Sedangkan *report* untuk melihat *trend* alasan mengapa barang dikembalikan dapat dilihat dengan bantuan dari fungsi *get\_return\_reason* yang terdapat pada Segmen 4.42.

#### Segmen 4.40. *World Cloud trend Goods*

```
$(document).ready(function() {
  $.ajax({
    url: '<?= base_url("CReport/get_wc") ?>',
    type: 'POST',

    success: function(data) {

      var text = data;
```

#### Segmen 4.41. *World Cloud trend Goods* (lanjutan)

```
var lines = text.split(/[,\. ]+/g),
data = Highcharts.reduce(lines, function (arr, word) {
  var obj = Highcharts.find(arr, function (obj) {
    return obj.name === word;
  });
  if (obj) {
    obj.weight += 1;
  } else {
    obj = {
      name: word,
      weight: 1
    };
    arr.push(obj);
  }
  return arr;
}, []);

Highcharts.chart('container', {
  series: [{
    type: 'wordcloud',
    data: data,
    name: 'Occurrences'
  }],
  title: {
    text: 'Wordcloud of bought Items'
  }
});
});
});
```

#### Segmen 4.42. Fungsi *get\_return\_reason*

```
public function get_return_reason()
{
  $dfdb = $this->load->database('default', TRUE);
  $sqry = "select COUNT(article) total, return_reason from
(select ARTICLE, ARTICLE_DESCRIPTION, b.SHIPMENT_DATE
,a.RETURN_REASON , CONVERT(varchar(6), SHIPMENT_DATE, 112)
periode
from DO_DETAIL a left join VDESTINATIONDO b on
a.DELIVERY_ORDER = b.DELIVERY_ORDER where QUANTITY_RETURN is
not null and CONVERT(varchar(6), SHIPMENT_DATE, 112) =
CONVERT(varchar(6), GETDATE(), 112)) a
group by return_reason , periode";

  $sqry = $dfdb->query($sqry);
  $isi = $sqry->result_array();

  $hasil = json_encode($isi);
  return $hasil;
}
```

Pada menu *report shipment* terdapat *report* yang menunjukkan daerah mana saja yang sering dikunjungi. Untuk dapat melihat *report* ini dibutuhkan fungsi *get\_shipment\_area*. Potongan dari kode tersebut dapat dilihat pada Segmen 4.43.

#### Segmen 4.43. Fungsi *get\_shipment\_area*

```
public function get_shipment_area()
{
    $result=array();
    $this->load->helper('url');
    $this->load->model("MReport", "mr");
    $city = $this->mr->get_shipment_city();
    $region = $this->mr->get_shipment_region('city');

    foreach ($city as $key => $value) {
        $region = $this->mr->
        >get_shipment_region($value['TO_CITY']);
        $reg = array();

        for($i =0 ; $i < count($region); $i++)
        {
            $reg[] = array(
                "type"      => $region[$i]['TO_ADDRESS'],
                "percent"   => $region[$i]['total']
            );
        }

        $result[] = array(
            "type"      => $value['TO_CITY'],
            "value"     => $value['totalShipment'],
            "subs"     => $reg
        );
    }

    echo json_encode($result);
}
```

#### 4.2.2.6. Fitur *Searching*

Fitur *searching* digunakan untuk mencari data pengiriman. Pencarian dapat dilakukan dengan memasukan kata dengan *keyword* apapun. Kemudian fungsi *get\_veta\_data* akan mencari setiap isi dari *field* yang memiliki data serupa. Daftar fungsi yang dipakai pada fitur ini terdapat pada Tabel 4.16. Sedangkan *query* yang dijalankan untuk dapat menghasilkan data tersebut terdapat pada Segmen 4.44.

Tabel 4.16. Daftar Fungsi dan *Query* pada Fitur *Searching*

<i>Usecase</i>	Menu / Submenu	Fungsi/ <i>Procedure</i> / <i>Event</i>	Keterangan	Segmen
<i>Search Delivery Order</i>	<i>ETA Search</i>	Fungsi <i>get_veta_data</i>	Fungsi untuk melakukan pencarian data pengiriman	-
		<i>Query view</i> tabel <i>vsearch</i>	<i>Query</i> untuk membuat tabel <i>vsearch</i>	Segmen 4.44

Segmen 4.44. *Query view* tabel *vsearch*

```

SELECT      b.SHIPMENT_DATE, b.NO_SHIPMENT, b.DELIVERY_ORDER,
b.FAKTUR_NUMBER, a.ARTICLE, a.ARTICLE_DESCRIPTION,
b.SHIP_TO_NAME, b.ADDRESS, b.TO_TELPHONE_NUMBER, a.QUANTITY,
           b.PRIORITY, b.DESCRPTION,
b.DRIVER_NAME, b.ASSISTANT_NAME, b.PLAT_NOMOR,
d.time_departure
FROM        dbo.DO_DETAIL AS a LEFT OUTER JOIN
           dbo.VDESTINATIONDO AS b ON
a.DELIVERY_ORDER = b.DELIVERY_ORDER LEFT OUTER JOIN
           dbo.SHIPMENT AS c ON b.NO_SHIPMENT =
c.NO_SHIPMENT LEFT OUTER JOIN
           dbo.route AS d ON b.ID_ROUTE =
d.id_route
    
```

4.3. **Implementasi Program Pada Modul *Driver***

Pada bagian ini akan dijelaskan mengenai implementasi fitur, fungsi, atau *event* yang terjadi pada modul *driver*. Penjelasan diawali dengan pemaparan mengenai *framework* modul, kemudian penjelasan mengenai fitur yang terdapat pada modul.

4.3.1. ***Model, View, dan Controller* Program Pada Modul *Driver***

Pada modul *driver*, *framework* yang dipakai untuk pembuatan program sama dengan modul *Administrator*, yakni *framework codeigniter*. Adapun daftar *file model* yang digunakan pada modul *driver* terdapat pada Tabel 4.17. Sedangkan *file controller* modul *driver* dapat dilihat pada Tabel 4.18. Untuk tampilan *user interface*, penulis menggunakan *template bootstrap* khusus yang

*responsive* untuk digunakan pada *smartphone*. Adapun daftar *file view* tersebut terdapat pada Tabel 4.19.

Tabel 4.17. Daftar *File Model* dalam Modul *Driver*

<i>Usecase</i>	<i>File</i>	Keterangan
-	MLogin.php	<i>File model</i> yang digunakan untuk mengambil setiap data untuk keperluan <i>login</i>
<i>Manage Agenda, Review Route</i>	Magenda.php	<i>Model</i> untuk mengatur <i>data</i> yang dibutuhkan pada fitur <i>manage agenda</i>
<i>Manage Route</i>	MRoute.php	<i>Model</i> untuk mengatur <i>data</i> yang dibutuhkan untuk mengelola rute pengiriman barang

Tabel 4.18. Daftar *File View* dalam Modul *Driver*

<i>Usecase</i>	<i>File</i>	Keterangan
<i>Manage Agenda</i>	VCalendar.php	<i>View</i> untuk agenda pengiriman barang
<i>Review Route</i>	VListreviewroute.php	Tampilan untuk melihat <i>review</i> rute pengiriman barang
<i>Manage Route</i>	Vlistroute.php	Tampilan untuk melihat urutan rute pengiriman barang
	VMaproute.php	<i>View</i> untuk menampilkan rute suatu destinasi pengiriman
<i>Manage Profile</i>	VProfilepage.php	<i>View</i> untuk melihat halaman profil supir. Bagian ini juga meliputi halaman untuk melakukan <i>manage profile driver</i>
-	VHeader.php	<i>File</i> yang berisikan <i>header</i> program seperti <i>tag script, meta, title, link</i> , dan sebagainya.
-	VLogin.php	<i>File view</i> untuk proses <i>login</i>
-	VMain.php	Kode untuk menggabungkan file <i>vheader.php, vnav.php</i> , dengan isi suatu <i>view</i> yang diperintahkan pada <i>controller</i>
-	VNav.php	<i>View</i> untuk menu navigasi / <i>navbar</i>

Tabel 4.19. Daftar *File Controller* dalam Modul *Driver*

<i>Usecase</i>	<i>File</i>	<i>Keterangan</i>
<i>Manage Agenda</i>	CAgenda.php	<i>Controller</i> untuk mengatur segala kebutuhan proses <i>manage agenda</i>
<i>Review Route</i>	Creviewroutes.php	<i>Controller</i> untuk mengatur segala kebutuhan proses <i>review rute</i>
<i>Manage Route</i>	Croutes.php	<i>Controller</i> yang mengatur segala keperluan untuk mengelola rute pengiriman
<i>Manage Profile</i>	CProfile.php	<i>File</i> untuk mengatur proses <i>manage profil</i>
-	CDashboard.php	<i>File</i> yang digunakan untuk mengelola <i>dashboard</i>
-	CLogin.php	<i>Controller</i> untuk proses <i>login</i>

#### 4.3.2. Pembuatan Fitur pada Modul *Driver*

##### 4.3.2.1. Fitur Login dan Logout

Fitur ini dibutuhkan agar pengguna dapat masuk kedalam program. Fitur ini terbagi menjadi dua komponen utama, yaitu *login* dan *logout*. *Login* merupakan halaman pengaman agar tidak sembarang orang dapat mengakses program. Sedangkan fitur *logout* merupakan fitur yang seakan-akan mengeluarkan pengguna dari program. Seseorang yang telah melakukan *logout*, harus melakukan *login* kembali untuk dapat mengakses program. Daftar fungsi yang dipakai pada fitur ini terdapat pada Tabel 4.20.

Tabel 4.20. Daftar Fungsi/Prosedur untuk Fitur *Login* dan *Logout*

<i>Usecase</i>	<i>Menu / Submenu</i>	<i>Fungsi/Procedure /Event</i>	<i>Keterangan</i>	<i>Segmen</i>
<i>Manage Profile</i>	<i>Login</i>	Fungsi <i>attempts</i>	Fungsi untuk masuk kedalam program	Segmen 4.45
		Fungsi <i>setting session</i>	Fungsi untuk pengaturan <i>session</i>	Segmen 4.46 Segmen 4.47
	<i>Logout</i>	Fungsi <i>Logout</i>	Fungsi untuk membuang data <i>session</i>	-

Pengguna yang ingin memasuki program, terlebih dahulu harus sukses dalam melewati fungsi *attempts*. Pada fungsi ini, terdapat enkripsi data dengan menggunakan sha dan md5 untuk meningkatkan keamanan program. Potongan program dapat dilihat pada Segmen 4.45. Ketika melakukan *login*, program juga akan menyimpan data *session* dengan cara memanggil fungsi *setting\_session*. Segmen program dapat dilihat pada Segmen 4.46 dan Segmen 4.47.

#### Segmen 4.45. Fungsi *attempts*

```
public function attempts(){
if($this->input->post() && $this->input->is_ajax_request()){
$userid = $this->input->post('userid');
$pass = $this->input->post('pass');
$cuser = $this->ml->getuser($userid,1);
if ($cuser == 0) {
echo "TETOT"; exit;
}elseif($cuser == 1){
$strpass = $this->ml->getuser($userid,2);
$qq = sha1(md5($pass));

if ($strpass == $qq) {
$this->ml->ses_set($userid);
$setime = $this->ml->set_time($userid);
if($setime){
echo "OK"; exit;
}else{
echo $setime; exit;
}
}else{
echo "Login failed : Invalid User ID or Password";
exit;
}
}
}else{
redirect(base_url());
}
}
```

#### Segmen 4.46. Fungsi *Setting Session*

```
public function ses_set($userid)
{
$db = $this->load->database("default", TRUE);
$sql = "SELECT top 1 c.EMPLOYEE_ID as
driverid,c.DRIVER_NAME as drivername ,b.ASSISTANT_ID as
assistantid,b.ASSISTANT_NAME as assistantname,a.* from VAGENDA
a left join VAGENDA b on a.GROUP_SHIPMENT_ID =
b.GROUP_SHIPMENT_ID and a.ID_TRUCK = b.ID_TRUCK and
```

### Segmen 4.47. Fungsi *Setting Session* (lanjutan)

```

a.SHIPMENT_DATE=b.SHIPMENT_DATE and REPLACE(b.ASSISTANT_ID, '
','')=REPLACE('".$userid."' ',' ','')
left join VAGENDA c on a.GROUP_SHIPMENT_ID =
c.GROUP_SHIPMENT_ID and a.ID_TRUCK = c.ID_TRUCK and
a.SHIPMENT_DATE=c.SHIPMENT_DATE and REPLACE(c.EMPLOYEE_ID, '
','')=REPLACE('".$userid."' ',' ','')
where REPLACE(a.EMPLOYEE_ID, ' ','')=REPLACE('".$userid."' ',' '
','') OR REPLACE(a.ASSISTANT_ID, ' ','')=REPLACE('".$userid."'
',' ','')
order by SHIPMENT_DATE desc

";

$query = $db->query($sql, array($userid));
$res = $query->row();

$user = ($res->driverid ?: $res->assistantid);
$username = ($res->drivername ?: $res->assistantname);

$newdata = array(
    'us_id' => $user
    , 'us_name' => $username
    , 'has_logged' => TRUE
    , 'group_shipment_id' => $res->GROUP_SHIPMENT_ID
    , 'plat_nomor' => $res->PLAT_NOMOR
    , 'wh_address' => $res->WH_ADDRESS
    , 'wh_city' => $res->WH_CITY
    , 'wh_telp' => $res->WH_TELP
);

$this->session->set_userdata($newdata);
}

```

#### 4.3.2.2. Fitur Agenda

Fitur ini digunakan untuk melihat setiap agenda pengiriman barang yang ada. Daftar agenda yang ada berbeda antara satu supir dengan yang lainnya. Hal ini dikarenakan agenda yang ditampilkan merupakan agenda khusus untuk supir yang ditugaskan. Daftar fungsi pada fitur ini dapat dilihat pada Tabel 4.21.

Tabel 4.21. Daftar Fungsi pada Fitur Agenda

<i>Usecase</i>	Menu / Submenu	Fungsi/ <i>Procedure</i> / <i>Event</i>	Keterangan	Segmen
<i>Manage Agenda</i>	Agenda	Fungsi <i>get_agenda</i>	Fungsi yang dibutuhkan untuk mendapatkan data agenda	Segmen 4.48

Untuk dapat menampilkan data agenda, dibutuhkan fungsi *get\_agenda* dengan parameter *employee id* dari supir yang bersangkutan. Isi dari fungsi tersebut dapat dilihat pada Segmen 4.48.

Segmen 4.48. Fungsi *get\_agenda*

```
public function get_agenda($id){
    $dfdb = $this->load->database('default', TRUE);

    $sqry = "select * from VAGENDA where
REPLACE (EMPLOYEE_ID, ' ', '')=REPLACE('".$id."', ' ', '') OR
REPLACE (ASSISTANT_ID, ' ', '')=REPLACE('".$id."', ' ', '')";
    $qqry = $dfdb->query($sqry);
    $isi = $qqry->result_array();

    $hasil = json_encode($isi);
    return $hasil;
}
```

#### 4.3.2.3. Fitur *Manage Route*

Fitur ini digunakan untuk mengelola rute pengiriman barang. Pada bagian ini tersedia beberapa fitur yang memudahkan supir dalam menentukan rute pengiriman barang. Daftar fungsi yang dipakai pada fitur ini dapat dilihat pada Tabel 4.22 dan Tabel 4.23.

Tabel 4.22. Daftar Fungsi dan *Query* pada Fitur *Manage Route*

<i>Usecase</i>	Menu / Submenu	Fungsi/Procedure /Event	Keterangan	Segmen
<i>Manage Route</i>	<i>Manage Route</i>	Fungsi <i>get_route</i>	Fungsi ini digunakan untuk menampilkan urutan rute pengiriman	-
		Fungsi <i>get_maproute</i>	Fungsi ini digunakan untuk menampilkan rute perjalanan pada suatu destinasi	-
		<i>Query View</i> vroute	<i>Query</i> untuk membuat <i>view</i> vroute	Segmen 4.49

Tabel 4.23. Daftar Fungsi dan *Query* pada Fitur *Manage Route* (lanjutan)

<i>Usecase</i>	Menu / Submenu	Fungsi/Procedure /Event	Keterangan	Segmen
<i>Manage Route</i>	<i>Manage Route</i>	Fungsi <i>get_article</i>	Fungsi untuk mendapatkan data barang pengiriman	-
		Fungsi <i>updateStatusDO</i>	Fungsi untuk melakukan <i>update</i> pada status <i>delivery order</i>	-

Untuk dapat menampilkan rute pengiriman barang yang terurut, maka dibutuhkan adanya *query* yang membantu untuk mempermudah pemanggilan data pengiriman yang terurut tersebut. *Query* tersebut dapat dilihat pada Segmen 4.49.

Segmen 4.49. *Query View* *vroute*

```

SELECT a.eta_time, a.time_departure, a.id_route,
b.ID_DESTINATION AS bid, c.ID_DESTINATION AS cid, x.ADDRESS AS
badr, y.ADDRESS AS cadr, b.LAT AS blat, b.LNG AS blng, c.LAT
AS clat, c.LNG AS clng
FROM      dbo.route AS a LEFT OUTER JOIN dbo.DESTINATION AS
b ON a.id_destination1 = b.ID_DESTINATION LEFT OUTER JOIN
dbo.DELIVERY_ORDER AS x ON x.DELIVERY_ORDER = b.DELIVERY_ORDER
LEFT OUTER JOIN dbo.DESTINATION AS c ON a.id_destination2 =
c.ID_DESTINATION LEFT OUTER JOIN dbo.DELIVERY_ORDER AS y ON
y.DELIVERY_ORDER = c.DELIVERY_ORDER
    
```

#### 4.3.2.4. Fitur *Review Route*

Pada bagian ini, *driver* dapat melihat dan mengelola status setiap rute pengiriman barang yang ada. Daftar fungsi yang dipakai pada fitur ini dapat dilihat pada Tabel 4.24

Tabel 4.24. Daftar Fungsi pada Fitur *Review Route*

<i>Usecase</i>	Menu / Submenu	Fungsi/Procedure/Event	Keterangan	Segmen
<i>Review Route</i>	<i>Review Route</i>	Fungsi <i>get_route</i>	Fungsi untuk menampilkan status rute pengiriman barang	-