

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Perancangan Sistem

Tahap awal penambahan sistem adalah tahap analisis. Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya. Langkah-langkah dalam tahap analisis sistem hampir sama dengan langkah-langkah yang dilakukan dalam mendefinisikan proyek-proyek sistem yang akan dikembangkan di tahap perancangan sistem.

Perancangan sistem merupakan tahap selanjutnya setelah analisis sistem, dengan perancangan sistem ini dapat digambarkan dengan jelas tentang apa yang dikerjakan pada analisis sistem seperti pembuatan *Data Flow Diagram*, *System Flow*, *Conceptual Data Model* dan *Physical Data Model*, maka dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut.

*Gammu* merupakan salah satu tool untuk mengembangkan aplikasi SMS gateway yang cukup mudah diimplementasikan dan gratis. *Gammu* bisa dikatakan sebagai aktor utama karena komponen inilah yang menjembatani pentransferan data-data SMS dari *mobile phone* ke komputer atau sebaliknya (Wahidin 2010).

Jadi *gammu* menghubungkan antara telepon seluler ataupun modem yang nantinya di koneksikan dengan tampilan komputer untuk menjalankan berbagai macam perintah. Untuk studi kasus ini hanya menggunakan fungsi *gammu* sebagai penerima dan pengirim SMS dengan bantuan modem sebagai nomor telepon seluler yang dipakai.

Perancangan sistem ini bertujuan untuk memberikan gambaran secara umum kepada pelanggan tentang sistem tambahan dan menghasilkan sistem yang sesuai dengan kebutuhan.

##### 4.1.1. *Data Flow Diagram*

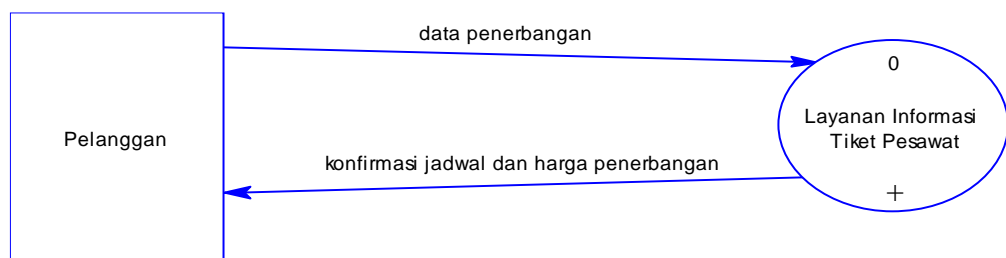
*Data Flow Diagram* (DFD) disebut juga dengan Diagram Arus Data (DAD). DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data, dan kemana tujuan data keluar dari sistem,

dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut (A. Kristanto, 2008).

Dalam pembuatan DFD ini menggunakan tool *Power Designer Process Analyst* sebagai media pembantu untuk kemudahan dalam perancangan dan penambahan sistem.

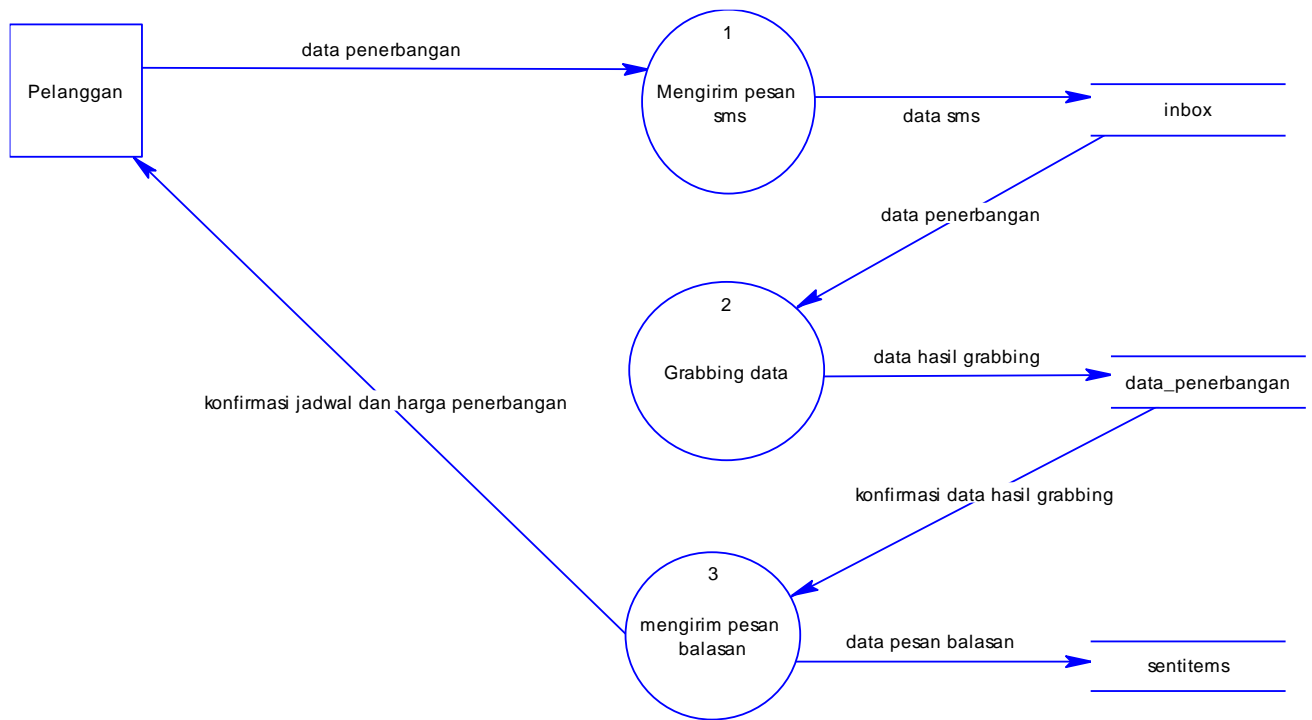
Sutabri (2003:163) menyatakan *Context Diagram* merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan system secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data menuju dan dari sistem diketahui menganalisis dari pelanggan dan sebagai hasil analisis dokumen.

Berikut adalah *context diagram* seperti ditunjukkan pada gambar 4.1 di bawah ini. Pelanggan mengirim SMS ke sistem, lalu sistem mengolah data yang masuk dan sistem melakukan balasan SMS kepada pelanggan yang bersangkutan.



Gambar 4.1 *Context Diagram*

Berikut adalah *data flow* diagram level 0 setelah *context diagram* dilakukan proses *decompose* dari Layanan Informasi Tiket Pesawat Berbasis SMS Gateway seperti ditunjukkan oleh gambar 4.2 di bawah ini.



Gambar 4.2 Data Flow Diagram level 0

#### 4.1.2. System Flow

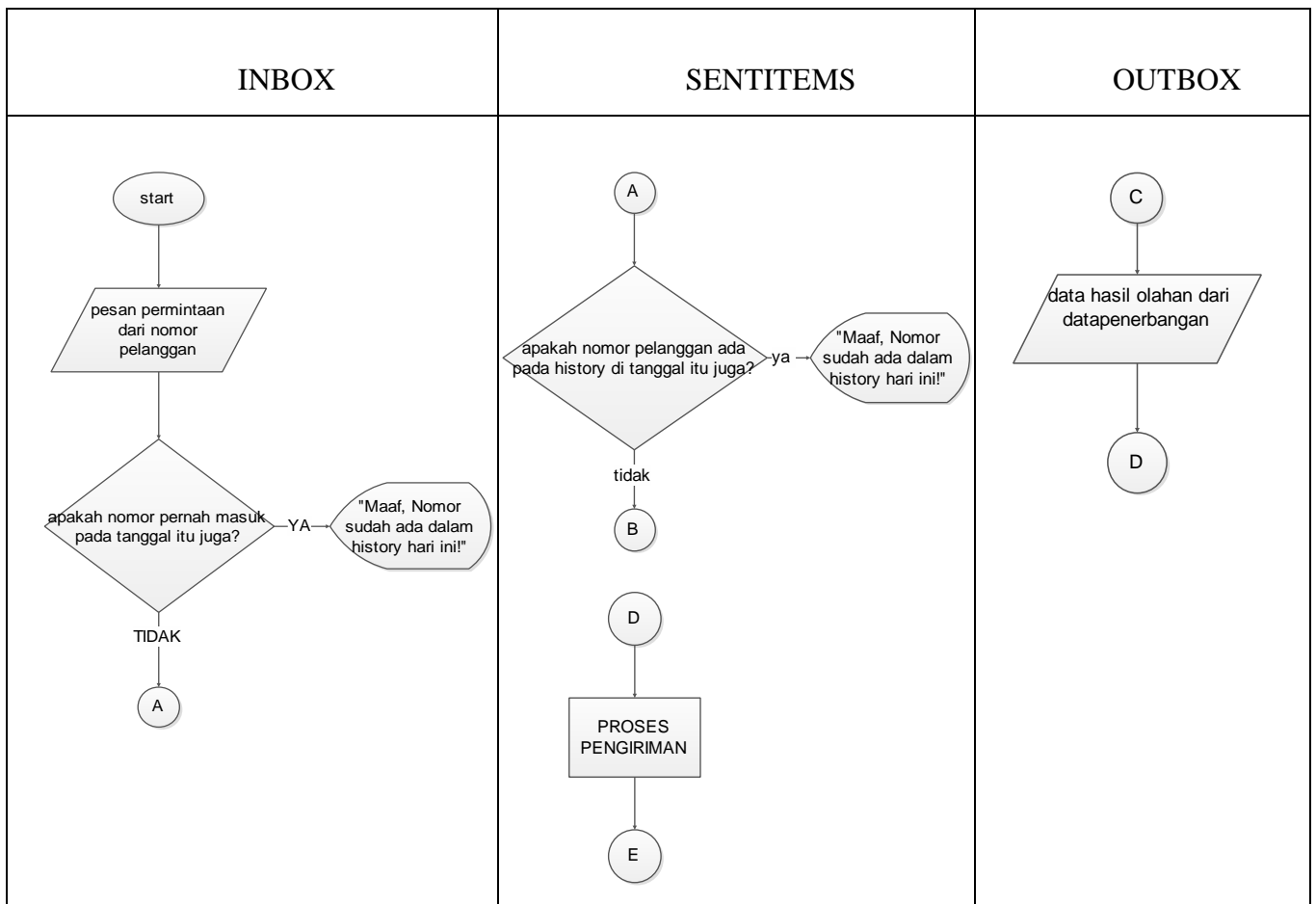
*System Flow* adalah lanjutan dari proses manual yang telah terkomputerisasi (B. Kristanto, 2008).

Disini tidak terdapat proses manual dalam pembuatan Layanan Informasi Tiket Pesawat Berbasis SMS *Gateway*, dan langsung dilanjutkan ke *system flow* dimana data yang mengalir melalui komputer.

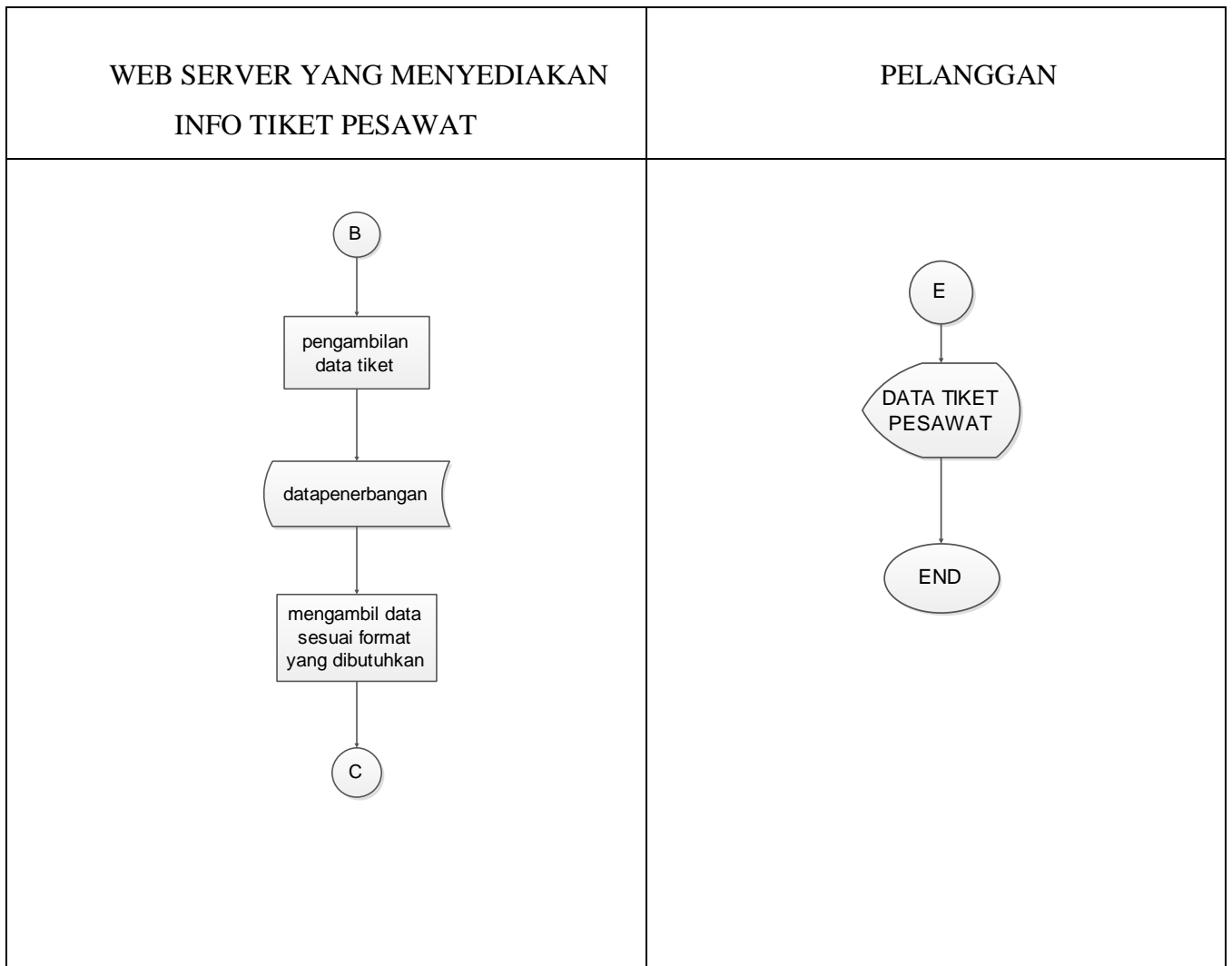
Berikut penjelasan mengenai proses terkomputerisasi dan dapat dilihat pada gambar 4.3 dan 4.4 :

1. Tabel *Inbox* menerima permintaan berupa SMS dari nomor pengirim
2. Proses pengecekan nomor pengirim yang ada di tabel *inbox* pada tanggal itu apakah sudah pernah melakukan permintaan atau belum
3. Pada saat (waktu) itu apakah nomor sudah pernah ada pada daftar nomor pengirim dalam tabel *sentitem*s atau belum

4. Jika nomor pengirim sudah ada pada daftar saat itu maka balasan untuk permintaan pengirim tidak akan terproses, jika nomor pengirim belum terdaftar maka proses pengiriman balasan akan dilanjutkan
5. Pengiriman balasan diawali dengan proses pengambilan data dari *website* yg menyediakan informasi tiket terlebih dahulu
6. Dan dari proses pengambilan data akan menghasilkan data informasi tiket yang tersedia
7. Data informasi tiket tersebut dimasukkan dalam tabel datapenerbangan
8. Kemudian data dari datapenerbangan diolah dan diambil sesuai format permintaan yang ada pada tabel *inbox*
9. Data olahan tersebut dimasukkan dalam tabel *outbox*
10. Proses pengiriman balasan SMS langsung dikirim ke nomor tujuan
11. Data yang ada pada *outbox* saat itu otomatis masuk ke dalam tabel *sentitems* yang artinya balasan sudah dipastikan sampai ke nomor tujuan



Gambar 4.3 *System Flow* Layanan Informasi Tiket Pesawat (I)



Gambar 4.4 System Flow Layanan Informasi Tiket Pesawat (II)

#### 4.1.3. Conceptual Data Model (CDM)

Model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu. Biasanya direpresentasikan dalam bentuk Entity Relationship Diagram.

##### Jenis-jenis objek dalam CDM :

1. *Entity*

*Entity* adalah obyek yang dapat dibedakan dalam dunia nyata.

2. *Relationship*

*Relationship* adalah hubungan yang terjadi antara satu atau lebih *entity*.

### 3. Inheritance

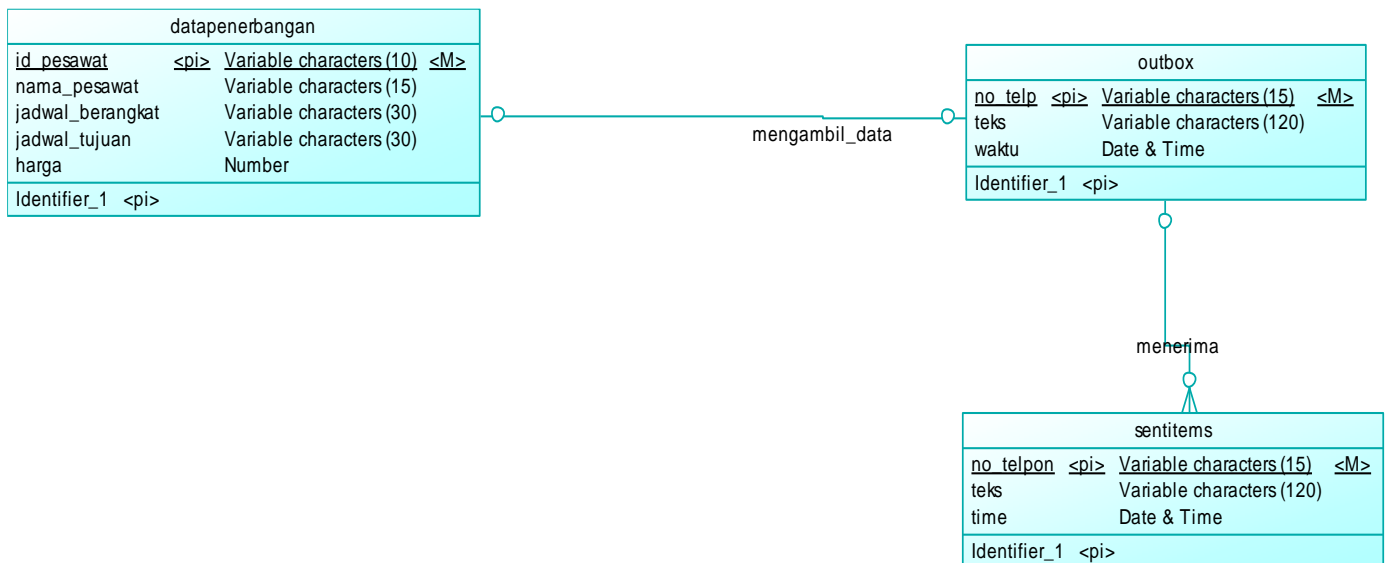
*Inheritance* adalah turunan atau pewarisan suatu *class* ke *class* baru dengan mewakili semua *function* yang ada di *class* utama (*parent class*).

### 4. Cardinalitas

*Cardinalitas* adalah banyaknya kejadian dalam suatu entitas untuk satu peristiwa tunggal dari entitas relasinya.

Tabel *datapenerbangan* memiliki relasi ke tabel *outbox* yang nantinya data dari *datapenerbangan* akan masuk ke dalam tabel *outbox*. Dengan relasi *one to one* dua tabel ini digabungkan. Lalu tabel *outbox* memiliki relasi *one to many* ke tabel *sentitems*. *Primary key* tabel *outbox* akan masuk ke tabel *sentitems*, yaitu *no\_telpon*.

Berikut CDM yang digunakan pada sistem ini dapat dilihat pada gambar 4.5.



Gambar 4.5 CDM Layanan Informasi Tiket Pesawat

#### 4.1.4. Physical Data Model (PDM)

PDM (Physical Data Model) merupakan hasil *generate* dari diagram CDM. Dari diagram ini dapat dilihat struktur tabel dari perancangan *system* yang sudah dibuat dan juga langsung me-generate tabel kedalam data basenya.

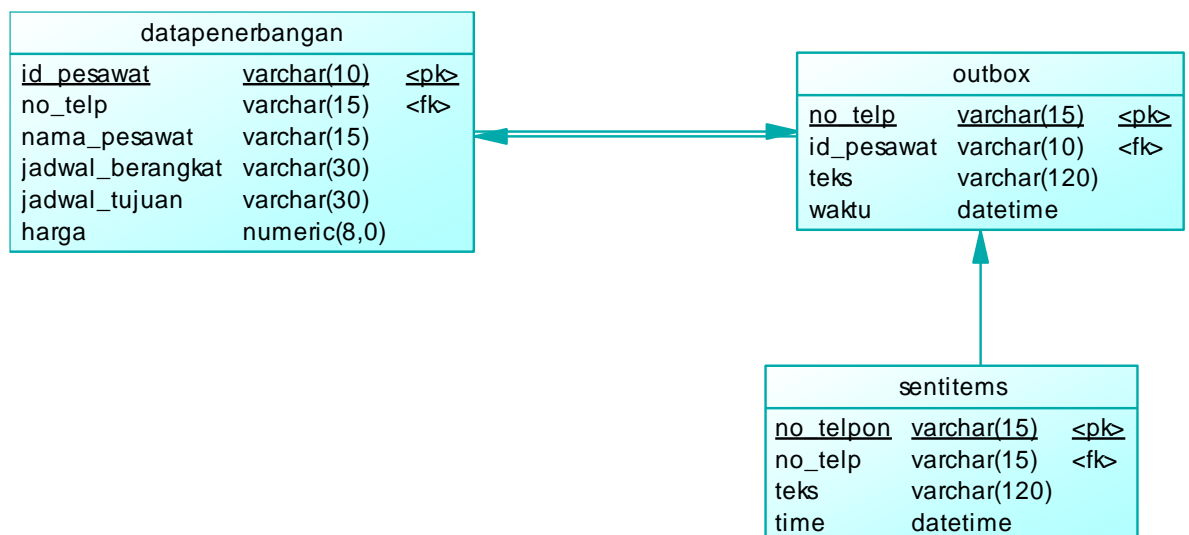
*Physical Data Model* Adalah *Tabel-Reference* diagram yang memodelkan sistem informasi termasuk detail implementasi fisiknya. Fungsi model ini adalah memodelkan struktur fisik dari suatu basis data. PDM bersifat spesifik.

Jenis – jenis object dalam PDM:

- Tabel
- View
- Reference

Tabel *datapenerbangan* yang memiliki *primary key* *id\_pesawat* mempunyai relasi dengan tabel *outbox*. Relasi dari kedua tabel tersebut adalah *one to one*. Lalu tabel *outbox* mempunyai relasi *one to many* dengan tabel *sentitems*. *Primary key* dari tabel *outbox* yaitu *no\_telp* akan masuk ke tabel *sentitems* sebagai *foreign key*.

Berikut PDM yang digunakan pada sistem ini dapat dilihat pada gambar 4.6:



Gambar 4.6 PDM Layanan Informasi Tiket Pesawat

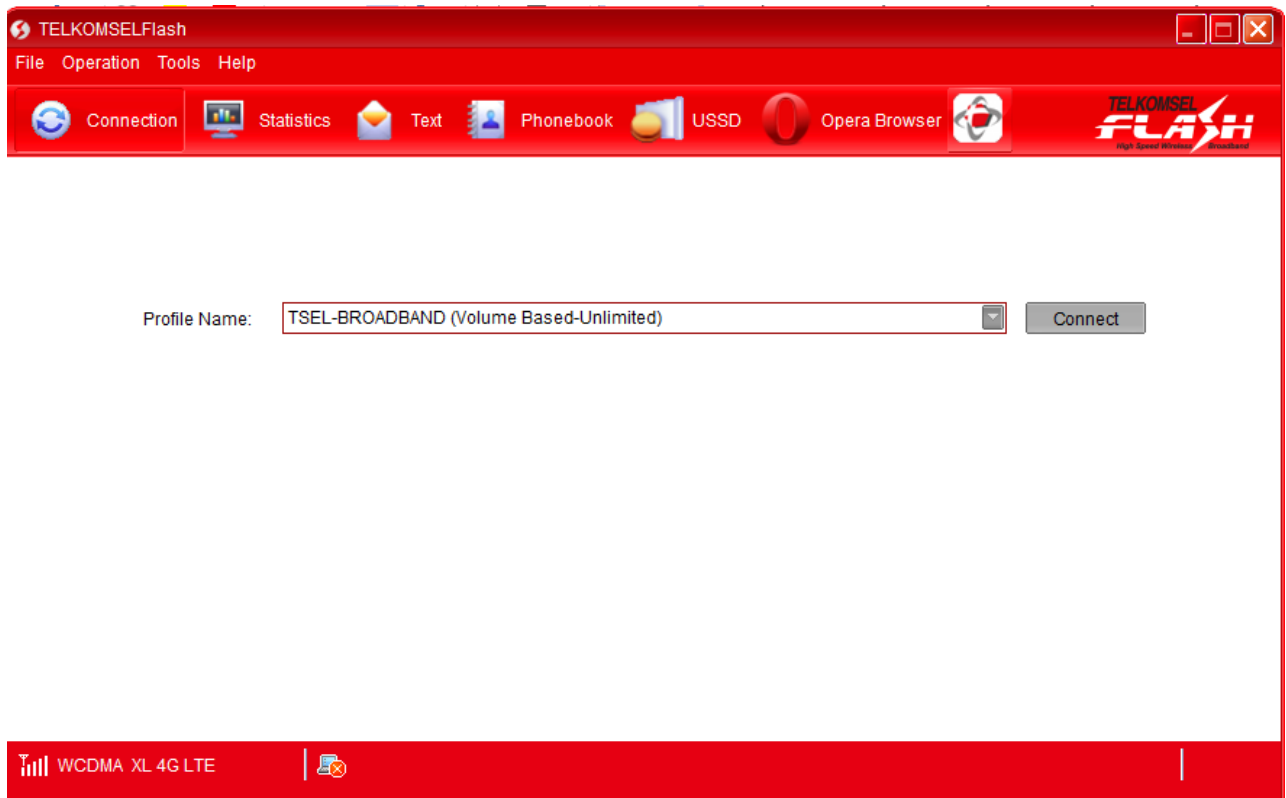
## 4.2. Proses Instalasi

Sebelum memulai membuat aplikasi SMS gateway ini, mula mula melakukan proses *install*. Berikut proses instalasi satu per satu.

### 4.2.1. Instalasi *Gammu* dan Modem

Dalam pembuatan aplikasi SMS gateway, membutuhkan *gammu* sebagai emulator untuk mengirim dan menerima SMS. Berikut adalah tata cara dalam proses instalasi *gammu*

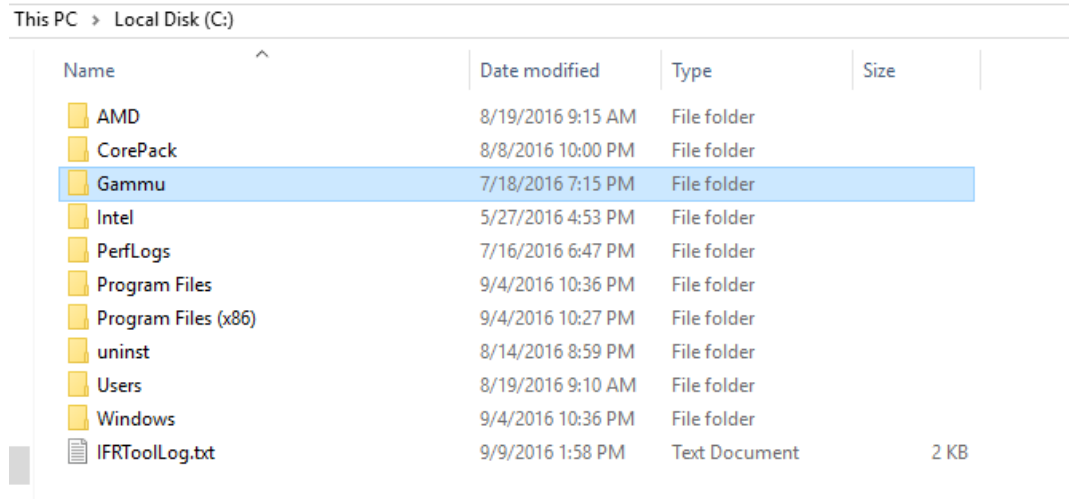
1. *Instal driver* modem yang sudah satu paket dengan modem seperti ditunjukkan pada gambar 4.7 berikut ini.



Gambar 4.7 Instalasi modem berhasil

2. *Download* dan ekstrak *gammu* dari *google*
3. Taruh hasil ekstrak tadi di *local disk* C:\ seperti pada gambar 4.8 berikut ini





Gambar 4.8 letak *gammu* di *local disk C:\*

4. Lakukan konfigurasi *file gammurc* dan *smsdrc* seperti pada gambar 4.9. Masuk ke direktori *C:\gammu\share\doc\gammu\examples\config*. *Copy* kedua *file* tersebut ke *C:\gammu\bin*.
5. Pertama lakukan konfigurasi *gammurc*. Untuk membuka file ini gunakan aplikasi notepad++. Untuk file *gammurc* hanya mengubah konfigurasi pada bagian *device* dan *connection* seperti pada gambar 4.9.

```

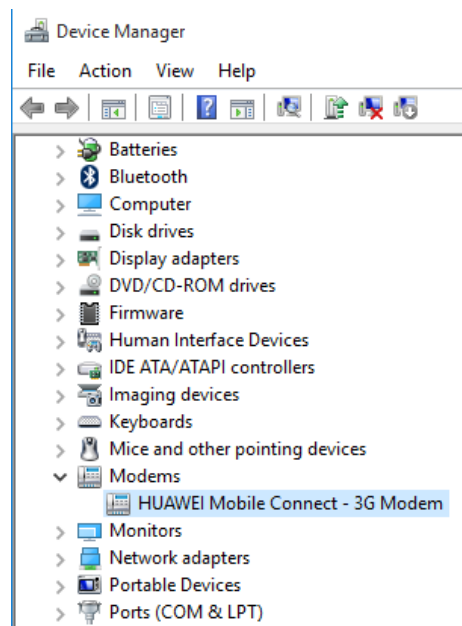
9  [gammu]
10
11  device = com5:
12  connection = at115200
13  ; Do not use model config
14  ;model = 6110
15  ;synchronizetime = yes
16  ;logfile = gammulog
17  ;logformat = textall
18  ;use_locking = yes
19  ;gammuloc = locfile
20  ;startinfo = yes
21  ;gammucoding = utf8
22  ;usephonedb = yes
23

```

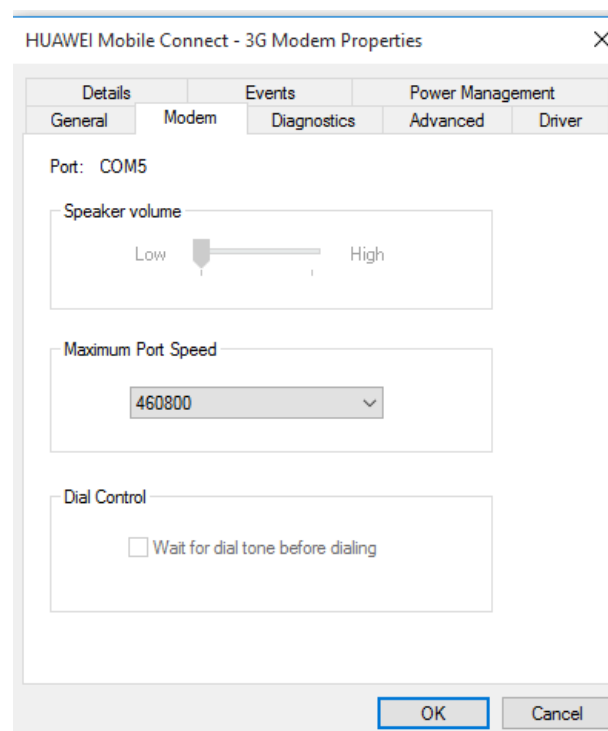
Gambar 4.9 Konfigurasi *gammurc*

*Device* sesuaikan dengan *port* modem pada komputer. Untuk melihat *port* modem, masuk ke *Device Manager* seperti terlihat pada gambar 4.10. Lalu

sesuaikan *port* dengan modem (com5) seperti pada gambar 4.11. Sedangkan untuk *connection*, menggunakan at115200



Gambar 4.10 *Device Manager*

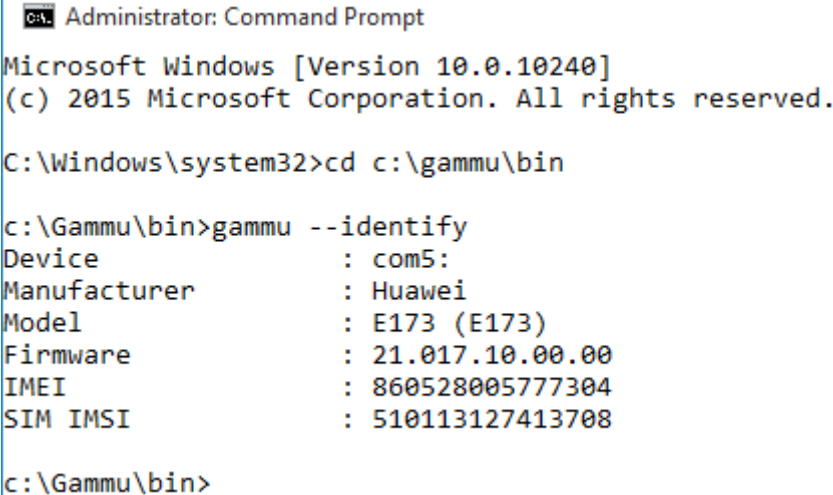


Gambar 4.11 Port modem

6. Langkah berikutnya yaitu melakukan pengecekan apakah konfigurasi modem sudah sesuai. Jalankan *Command Prompt (run as administrator)*,

kemudian ketikkan perintah berikut untuk masuk ke direktori `C:\gammu\bin` seperti terlihat pada gambar 4.12

```
cd c:\gammu\bin
gammu --identify
```



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

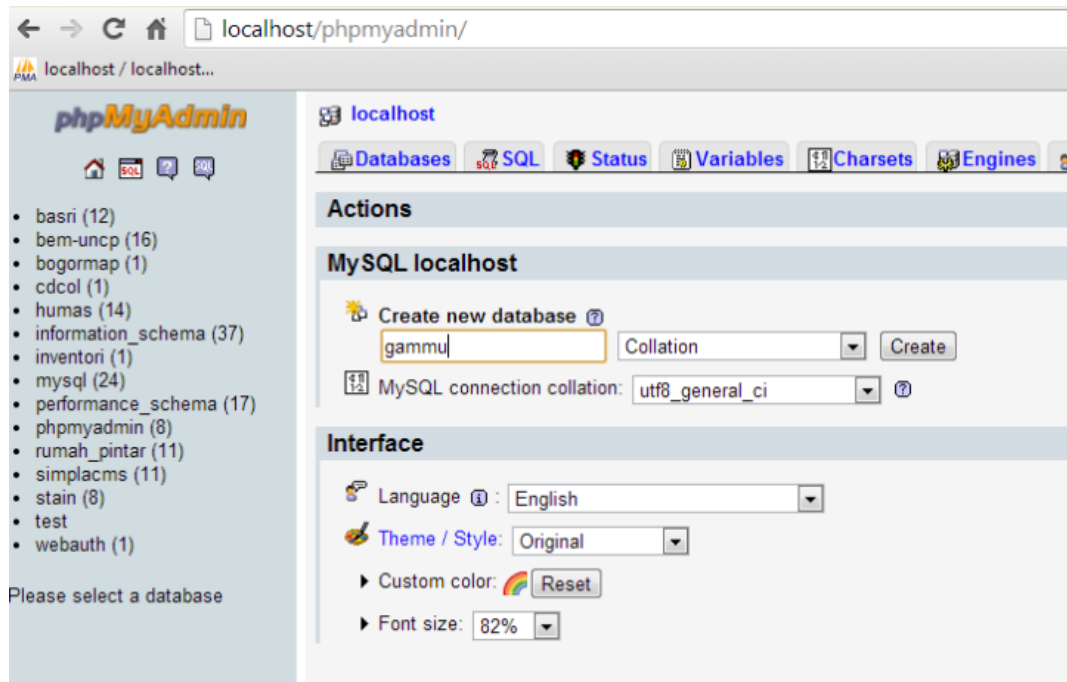
C:\Windows\system32>cd c:\gammu\bin

c:\Gammu\bin>gammu --identify
Device           : com5:
Manufacturer     : Huawei
Model            : E173 (E173)
Firmware         : 21.017.10.00.00
IMEI             : 860528005777304
SIM IMSI         : 510113127413708

c:\Gammu\bin>
```

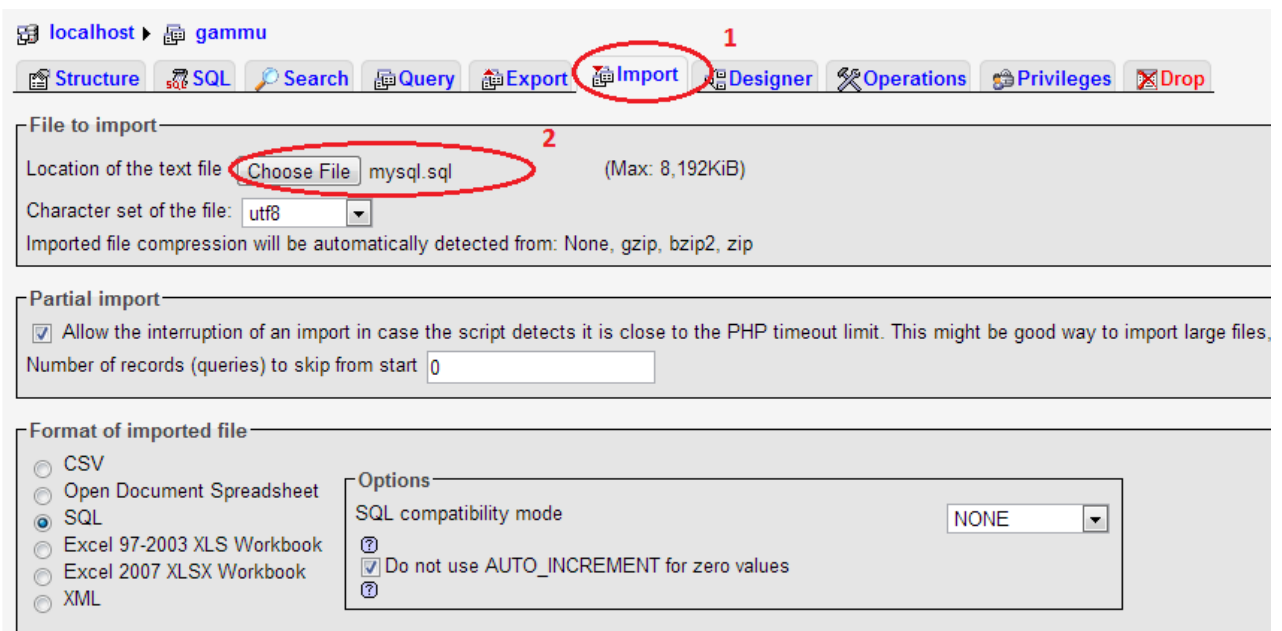
Gambar 4.12 Konfigurasi *gammu* di command prompt

7. Setelah modem sudah terkoneksi, maka selanjutnya tahap menyambungkan dengan *database*. Buat *database* baru di *localhost/phpmyadmin*, untuk contoh kali ini buat *database* dengan nama “*gammu*” seperti terlihat pada gambar 4.13

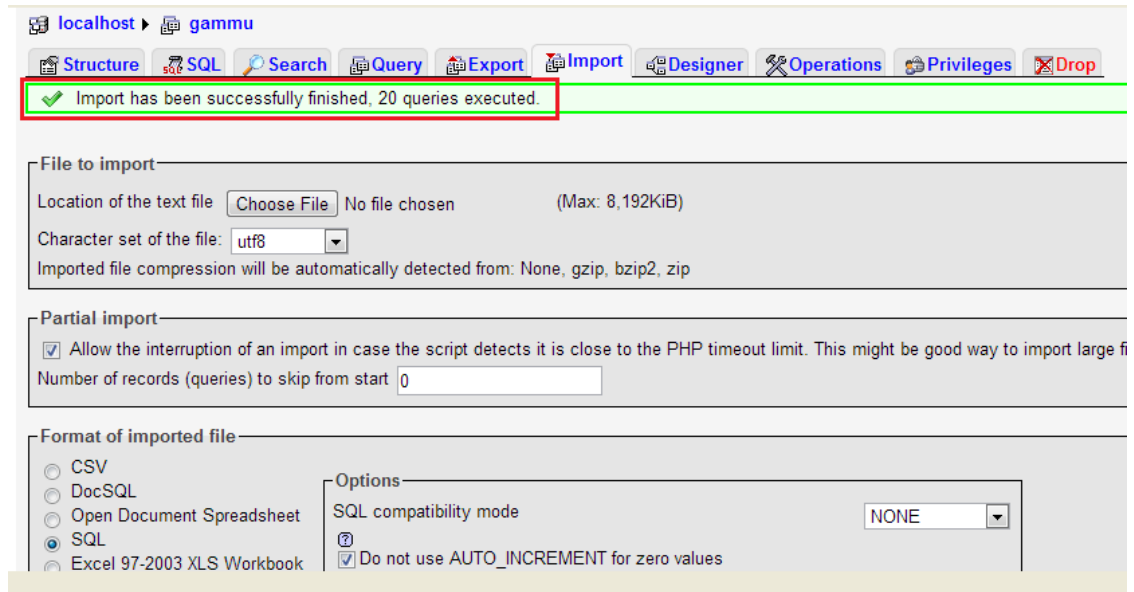


Gambar 4.13 Buat database baru

Kemudian *import database mysql.sql* dari folder *C:\gammu\share\doc\gammu\examples\sql* seperti terlihat pada gambar 4.14

Gambar 4.14 *Import database gammu*

Jika berhasil, maka akan tampil seperti gambar 4.15 di bawah ini



Gambar 4.15 *Import database gammu* berhasil

8. Selanjutnya melakukan konfigurasi pada *file smsdrc* seperti terlihat pada kode program 4.1. Jangan lupa hilangkan tanda # pada bagian yang dikonfigurasi

```
[gammu]
device = com5:
#model = 6110
connection = at115200
```

Kode program 4.1 konfigurasi *smsdrc* untuk *device*

Untuk bagian PIN diberi tanda # jika kartu sim anda tidak menggunakan pin seperti terlihat pada kode program 4.2

```
[smsd]
# SMSD service to use, one of
FILES, MYSQL, PGSQL, DBI
service = mysql
# PIN for SIM card
# PIN = 1234
# File (or stderr, syslog,
eventlog) where information will be
logged
logfile = smsdlog
```

Kode program 4.2 konfigurasi *smsdrc* untuk kartu sim

Isikan informasi untuk databasenya seperti terlihat pada kode program 4.3.

```
# Database backends congfiguration
user = root
password =
pc = localhost
# pc can also contain port or socket path after
colon (eg. localhost:/path/to/socket)
database = gammu
```

Kode program 4.3 konfigurasi *smsdrc* untuk *database*

- Setelah konfigurasi *smsdrc*, selanjutnya install *service gammu*, buka *Command Prompt* lalu ketik seperti terlihat pada kode program 4.4 dan gambar 4.16

```
gammu-smsd -i -c smsdrc -n gammuSMSD
```

Kode program 4.4 install *service gammu*

```
c:\gammu\bin>gammu-smsd -i -c smsdrc -n gammuSMSD
Service gammuSMSD installed sucessfully
```

Gambar 4.16 *Gammu* telah terinstal

Kemudian jalankan *service gammu* dengan perintah seperti terlihat pada kode program 4.5 dan gambar 4.17

```
gammu-smsd -s -c smsdrc -n gammuSMSD
```

Kode program 4.5 menjalankan *service gammu*

```
c:\gammu\bin>gammu-smsd -s -c smsdrc -n gammuSMSD
Service gammuSMSD started sucessfully
```

Gambar 4.17 *Gammu* started

#### 4.2.2. Instalasi *PHP* dan *cURL*

1. Buka folder C:\xampp\php
2. kemudian cari *php.ini*  
cari ;*extension=php\_curl.dll*
3. Hapus tanda ; sehingga menjadi  
*extension=php\_curl.dll*

```
extension=php_bz2.dll
extension=php_curl.dll
extension=php_mbstring.dll
extension=php_exif.dll
```

4. Save
5. Sesudah itu *restart apache* (stop lalu start kembali).

#### 4.3. Teknik *Grabbing*

Konsep *grabbing* ini berbeda dengan konsep *web service* karena data teks yang diambil melalui *grabbing* ini tidak disajikan dalam bentuk format tertentu seperti halnya *XML* atau *JSON*. Namun teks yang diambil berada di halaman *website*, atau dengan kata lain terselip di dalam kode-kode *html*.


Untuk melakukan *grabbing*, dapat menggunakan *cURL*. Fungsi *cURL* adalah dapat membaca semua kode *html* dari suatu situs lalu diambil informasi yang diinginkan dan kemudian informasi tersebut dapat ditampilkan di *website*.

Pada contoh ini, dilakukan proses *grabbing* pada sebuah *website* yaitu mengambil informasi tiket pesawat yang ada di salah satu situs penyedia layanan informasi tiket pesawat, perhatikan gambar 4.18 di bawah ini

Filter: Airlines ▼ Time ▼ Price ▼ Stops ▼ All Without Stop

Jakarta (JKT) to Denpasar, Bali (DPS) 26-08-2016

Flights	Depart	Arrival	Transit	Facility	Price
<b>Citilink</b> Citilink QG-946 JT-568 Lion Multiple Carriers	<b>13:05</b> Jakarta (CGK)	<b>23:00</b> Denpasar (DPS)	1 Stop	Detail ▼	<b>IDR 1.509.160</b> <a href="#">Book Now &gt;&gt;</a>
<b>Citilink</b> Citilink QG-801 QG-644	<b>07:45</b> Jakarta (CGK)	<b>15:05</b> Denpasar (DPS)	1 Stop ▼	20 Kg Airport Tax	<b>IDR 1.523.000</b> <a href="#">Book Now &gt;&gt;</a>
<b>AirAsia</b> Airasia QZ-7550 JT-568 Lion Multiple Carriers	<b>14:30</b> Jakarta (CGK)	<b>23:00</b> Denpasar (DPS)	1 Stop	Detail ▼	<b>IDR 1.565.800</b> <a href="#">Book Now &gt;&gt;</a>
<b>Sriwijaya Air</b> Sriwijaya SJ-230 JT-568	<b>12:40</b>	<b>23:00</b>	1 Stop	Detail ▼	<b>IDR 1.520.800</b> <a href="#">Book Now &gt;&gt;</a>

Butuh Bantuan? LIVE CHAT 

Gambar 4.18 Website penyedia layanan tiket pesawat

Langkah pertama, buka halaman *website* yang akan di lakukan proses *grabbing* dan pelajari *source codenya* (Klik kanan -> *view page source*). Lalu cari bagian dalam kode *html source* yang terdapat informasi tiket pesawat. Informasi tiket pesawat terdapat pada gambar 4.19 berikut ini.

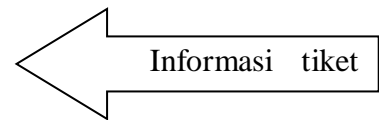


```

</div>
</h2>
<table>
<thead>
<tr class="topsorting">
<th class="td1" colspan="2"><a class="sort" data-field="airlines">Flights <i></i></a></th>
<th class="td2"><a class="sort" data-field="depart">Depart <i></i></a></th>
<th class="td3"><a class="sort" data-field="arrival">Arrival <i></i></a></th>
<th class="td4"><a class="sort" data-field="stop">Transit <i></i></a></th>
<th class="td6"><a class="sort" data-field="baggage">Facility <i></i></a></th>
<th class="td7"><a class="sort" data-field="price">Price <i></i></a></th>
<th class="td8">&nbsp;</th>
</tr>
</thead>
<tbody id="tbody_depart">
<tr id="flight_236588729"
class="flight-rows flight-BATIK flight-rows-depart-HLPDPS flight-item"
data-airlines="ID-7271/JT-365/JT-910"
data-airlinesname="BATIK"
data-flightid="236588729"
data-depart="05:55"
data-arrival="1472199600"
data-specialsubsidy=""
data-stop="2"
data-stoptext="2 Stops"
data-baggage="20 Kg"
data-priceconvert="2361800"
data-price="2361800"
data-multiplier = "1">
<td class="td1">
<div class="flight_mark">


ID-7271<br />JT-365<br />JT-910 </td>
<td class="td2">
<h4>05:55</h4>
<h5>Jakarta Halim (HLP)</h5>

```



Gambar 4.19 View page source halaman website tiket

Berikut adalah informasi satu data tiket pesawat. Kemudian, pilih kode *tag* yang terletak sebelum bagian informasi pesawat. Sebaiknya pilih beberapa *tag* sekaligus supaya unik (tidak ada yang sama) dalam satu halaman. Misalnya dalam contoh ini diambil *tag* yang terletak sebelum informasi pesawat seperti di tag `</h2>` ini.

Tag di atas terletak di atas bagian yang diberi panah di atas. Selanjutnya dengan tag tersebut, akan dilakukan proses *exploding* (pemecahan string).

Sehingga scriptnya menjadi seperti kode program 4.6 berikut ini.

```

<?php
function grabbing($url){
    $data = curl_init();
        curl_setopt($data, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($data, CURLOPT_URL, $url);
    $output = curl_exec($data);
    curl_close($data);
    return $output;
}
    $ambilhtml=grabbing('http://en.██████████.com/pesawat/cari?d=JK
T&a=DPS&date=2016-07-22&ret_date=&flexDeparting=2016-
07&adult=1&child=0&infant=0');
    $filter = explode('</h2>', $ambilhtml);
    echo $filter[2];
?>

```

#### Kode program 4.6 *Grabbing*

Selanjutnya, perhatikan kembali *source code* dari halaman depan *website* tersebut, bahwa ternyata terdapat 2 buah *tag* yang telah dipilih tadi (</h2>). Cara untuk mengetahui bawah terdapat 2 buah *tag* yang telah dipilih adalah tekan ctrl + F, lalu ketik *tag* yang telah dipilih tadi. Nanti akan terlihat ada berapa buah tag disitu.

Hasil *script* di atas tampak seperti gambar 4.20 di bawah ini.

Pesawat	Pergi	Tiba	Transit	Fasilitas	Harga
 IW-1710 IW-1211 Multiple Airlines JT-960	07:50 Jakarta Halim (HLP)	14:45 Denpasar (DPS)	HLP - TKG (07:50 - 08:50) TKG - BDO (09:25 - 10:25) BDO - DPS (12:00 - 14:45) 2 Transit	10 Kg Pajak Bandara	IDR 2.447.600 Pesan Sekarang
 IW-1710 IW-1211 Multiple Airlines JT-902	07:50 Jakarta Halim (HLP)	19:00 Denpasar (DPS)	HLP - TKG (07:50 - 08:50) TKG - BDO (09:25 - 10:25) BDO - DPS (16:15 - 19:00) 2 Transit	10 Kg Pajak Bandara	IDR 2.370.600 Pesan Sekarang
 ID-7271 JT-365 Multiple Airlines JT-910	05:55 Jakarta Halim (HLP)	15:20 Denpasar (DPS)	HLP - BPN (05:55 - 09:00) BPN - SUB (10:55 - 11:35) SUB - DPS (13:30 - 15:20) 2 Transit	20 Kg Pajak Bandara Makanan Gratis	IDR 2.361.800 Pesan Sekarang
 ID-7271 JT-675 Multiple Airlines JT-745	05:55 Jakarta Halim (HLP)	18:00 Denpasar (DPS)	HLP - BPN (05:55 - 09:00) BPN - UPG (14:40 - 15:50) UPG - DPS (16:40 - 18:00) 2 Transit	20 Kg Pajak Bandara Makanan Gratis	IDR 2.529.000 Pesan Sekarang
 ID-7271 JT-679 Multiple Airlines JT-745	05:55 Jakarta Halim (HLP)	18:00 Denpasar (DPS)	HLP - BPN (05:55 - 09:00) BPN - UPG (11:00 - 12:10) UPG - DPS (16:40 - 18:00) 2 Transit	20 Kg Pajak Bandara Makanan Gratis	IDR 2.529.000 Pesan Sekarang

Gambar 4.20 Hasil contoh *grabbing*

#### 4.4. Desain Database *Gammu*

*Gammu* telah memiliki *database* sendiri dan hanya melakukan *import database*. Terdapat 9 tabel di dalamnya di antaranya tabel *daemons*, *gammu*, *inbox*, *outbox*, *outbox\_multipart*, *pbk*, *pbk\_group*, *phone*, dan *sentitems*. Lalu tabel yang akan digunakan hanya tabel *inbox* dan *sentitems* serta *outbox* seperti pada gambar 4.21 , gambar 4.22 , gambar 4.23. Namun untuk implementasinya juga membutuhkan satu tabel tambahan untuk digunakan menampung data penerbangan yang telah berhasil dilakukan proses *grabbing*, yaitu tabel tambahan dengan nama *datapenerbangan* seperti terlihat pada gambar 4.24 di bawah ini.

UpdatedInDB	ReceivingDateTime	Text	SenderNumber	Coding	UDH	SMSCNumber	Class	TextDecoded	ID
2016-08-08 10:22:23	2016-08-08 10:17:09	005400690...	96B +6281252408574	Default...	0B	+62811078801	-1	Tiket SUB LOP 20...	24B 27
2016-08-08 10:22:23	2016-08-08 10:16:38	005400690...	96B +6289675674641	Default...	0B	+628964011134	-1	Tiket SUB LOP 20...	24B 26
2016-08-08 10:22:23	2016-08-08 10:16:18	005400690...	96B +6289675674641	Default...	0B	+628964011131	-1	Tiket SUB LOP 20...	24B 25
2016-08-08 10:11:38	2016-08-08 10:11:33	005400690...	96B +6289675674641	Default...	0B	+628964011131	-1	Tiket JKT DPS 20...	24B 24
2016-08-08 09:34:25	2016-08-08 09:33:19	005400690...	96B +6289675674641	Default...	0B	+628964011134	-1	Tiket JKT DPS 20...	24B 22
2016-08-08 09:34:25	2016-08-08 09:33:00	005400690...	96B +6289675674641	Default...	0B	+628964011131	-1	Tiket JKT DPS 20...	24B 21
2016-08-08 09:34:25	2016-08-08 09:31:43	005400690...	96B +6281252408574	Default...	0B	+62811078801	-1	Tiket SUB DPS 20...	24B 20
2016-08-08 09:46:26	2016-08-08 09:31:43	005400690...	96B +6281252408574	Default...	0B	+62811078801	-1	Tiket SUB DPS 20...	24B 23

Gambar 4.21 Tabel *inbox*

UpdatedInDB	InsertIntoDB	SendingDateTime	DeliveryDateTime	Text	DestinationNumber	Coding
2016-08-07 18:30:35	2016-08-06 21:46:37	2016-08-07 18:30:35	(NULL)	00530055004200200044005000...	500B	+6289675674641
2016-07-21 11:05:50	2016-07-21 10:43:38	2016-07-21 11:05:50	(NULL)	006F006B0065	12B	+6289675674641
2016-08-06 20:42:47	2016-08-06 20:42:29	2016-08-06 20:42:47	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-07-23 14:53:35	2016-07-23 14:53:07	2016-07-23 14:53:35	(NULL)	006400680069006B0061	20B	+6282231370145
2016-07-23 14:53:39	2016-07-23 14:53:32	2016-07-23 14:53:39	(NULL)	00720065006E006400720061	24B	+6289675674641
2016-08-06 20:35:00	2016-08-03 15:20:02	2016-08-06 20:35:00	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:35:07	2016-08-03 15:25:06	2016-08-06 20:35:07	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:35:14	2016-08-03 15:27:54	2016-08-06 20:35:14	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:35:21	2016-08-03 15:33:10	2016-08-06 20:35:21	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:35:34	2016-08-03 15:36:25	2016-08-06 20:35:34	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:45:02	2016-08-06 20:44:34	2016-08-06 20:45:02	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:52:09	2016-08-06 20:51:31	2016-08-06 20:52:09	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:54:14	2016-08-06 20:53:26	2016-08-06 20:54:14	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-06 20:58:26	2016-08-06 20:58:10	2016-08-06 20:58:26	(NULL)	00440050005300200053005500...	492B	+6289675674641
2016-08-07 18:30:40	2016-08-06 22:19:31	2016-08-07 18:30:40	(NULL)	00530055004200200044005000...	500B	+6289675674641
2016-08-07 18:51:28	2016-08-07 18:51:00	2016-08-07 18:51:28	(NULL)	00430047004B00200044005000...	540B	+6281252408574
2016-08-08 09:43:18	2016-08-08 09:42:54	2016-08-08 09:43:18	(NULL)	00430047004B00200044005000...	492B	+6289675674641
2016-08-08 10:29:11	2016-08-08 10:28:45	2016-08-08 10:29:11	(NULL)	0053005500420020004C004F00...	492B	+6281252408574

Gambar 4.22 Tabel *sentitem*s

UpdatedInDB	InsertIntoDB	SendingDateTime	SendBefore	SendAfter	Text	DestinationNumber	Coding	UDH	Clas
* RENT_TIMESTAMP	-00-00 00:00:00	000-00-00 00:00:00	23:59:59	00:00:00	(NULL)	OK	Def...	(NULL)	OK

Gambar 4.23 Tabel *outbox*

id_penerbangan	pesawat	keberangkatan	kedatangan	jamberangkat	jamdatang	harga
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 4.24 Tabel *datapenerbangan*

## 4.5. Implementasi Aplikasi

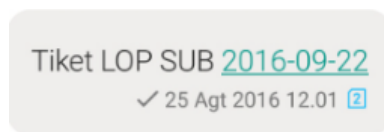
Setelah semua proses instalasi telah selesai dilakukan dan berhasil, maka langkah selanjutnya masuk ke dalam implementasi aplikasi.

### 4.5.1. Format Pengiriman SMS

Fungsi SMS ini diperuntukkan kepada pelanggan setia atau *member* karena asumsinya adalah pelanggan tersebut telah sering berpergian dan pelanggan telah mengetahui singkatan tujuan bandara tertentu atau *The International Air Transport Association (IATA)* untuk memudahkan proses dalam SMS ini.

Format pengiriman SMS disini untuk memudahkan aplikasi berjalan dengan lancar. Formatnya sederhana dan mudah untuk diingat, cukup ketik : Tiket [spasi] Berangkat [spasi] Tujuan [spasi] yyyy-mm-dd. Contohnya sebagai berikut : *Tiket SUB DPS 2016-08-29*. SMS ini dikirim ke nomer modem yaitu 0878-5529-

9389. SMS yang diterima oleh sistem kemudian secara otomatis masuk ke dalam tabel *inbox*. Format SMS tersebut seperti pada gambar 4.25 di bawah ini.



Gambar 4.25 Format pengiriman SMS

## KOTAK MASUK

No.	Tgl. Masuk	Pengirim	Isi	Action
1	2016-08-25 12:01:22	+6281252408574	Tiket LOP SUB 2016-09-22	<a href="#">Hapus</a>
2	2016-08-24 23:11:31	+6289675674641	Tiket DPS SUB 2016-09-29	<a href="#">Hapus</a>
3	2016-08-24 23:02:54	+6281216532857	Tiket DPS LOP 2016-09-29	<a href="#">Hapus</a>
4	2016-08-08 10:17:09	+6281252408574	Tiket SUB LOP 2016-08-29	<a href="#">Hapus</a>
5	2016-08-08 10:16:38	+6289675674641	Tiket SUB LOP 2016-08-29	<a href="#">Hapus</a>
6	2016-08-07 19:10:04	+6289675674641	Tiket JKT DPS 2016-08-29	<a href="#">Hapus</a>
7	2016-08-07 18:41:12	+6281252408574	Tiket CGK DPS 2016-08-29	<a href="#">Hapus</a>
8	2016-08-07 18:28:56	+6281252408574	Tiket CGK DPS 2016-08-29	<a href="#">Hapus</a>

Gambar 4.26 Tampilan *inbox* di *localhost*

Gambar 4.26 di atas menunjukkan inputan SMS masuk di dalam tabel *inbox*. Dilakukan pemecahan *string* di dalam atribut isi. Kata kunci yang dipakai yaitu ketik : Tiket [spasi] Berangkat [spasi] Tujuan [spasi] yyyy-mm-dd. Contohnya sebagai berikut : *Tiket SUB DPS 2016-08-29*

### 4.5.2. File Load.php

Sistem melakukan *select* data dari *database* sebagai inputan untuk *url* yang akan dilakukan proses *grabbing*. Terdapat beberapa proses yaitu mengambil kota keberangkatan, kota tujuan, dan mengambil tanggal penerbangan yang terlihat pada kode program 4.7 , kode program 4.8 , kode program 4.9.

```

//menampilkan daftar SMS masuk
include ("config.php");
//ambil data keberangkatan
$query1 = "SELECT SUBSTR(TextDecoded, 7, 3) FROM inbox ORDER BY
ReceivingDateTime DESC LIMIT 1";
    $data1 = mysql_query($query1);
    $berangkat = mysql_fetch_array($data1);

```

#### Kode program 4.7 Mengambil kota keberangkatan

```

//ambil data kedatangan
$query2 = "SELECT SUBSTR(TextDecoded, 11, 3) FROM inbox ORDER BY
ReceivingDateTime DESC LIMIT 1";
    $data2 = mysql_query($query2);
    $datang = mysql_fetch_array($data2);

```

#### Kode program 4.8 Mengambil kota tujuan

```

//ambil data tanggal
$query3 = "SELECT SUBSTR(TextDecoded, 15, 10) FROM inbox ORDER BY
ReceivingDateTime DESC LIMIT 1";
    $data3 = mysql_query($query3);
    $tanggal = mysql_fetch_array($data3);

```

#### Kode program 4.9 Mengambil tanggal penerbangan

Lalu semua data yang telah di *input* tadi dimasukkan ke dalam *url* sebagai *input*. Data yang telah di ambil dari teknik *grabbing* adalah data tiket dari halaman *website* yang telah dibuka tadi seperti pada kode program 4.10.

```

//AMBIL DARI HTTP
$url =
bacaHTML('http://en.tiket.com/pesawat/cari?d='.$berangkat[0].
'a='
.$datang[0].&date='.$tanggal[0].&ret_date=&flexDepartin
g=2016-
07&adult=1&child=0&infant=0');
$kodeHTML = $url;

```

**Kode program 4.10 *Insert* alamat web**

Setelah data semua penerbangan pada hari tertentu, pada bandara awal dan tujuan tertentu berhasil di lakukan proses *grabbing*, data pada kode program 4.10 dimasukkan ke *database* pada tabel datapenerbangan untuk mempermudah mengelolanya.

#### **4.5.3. Pada Pengiriman SMS**

Setelah data masuk ke *database*, sistem akan mengarahkan ke bagian pengiriman SMS. Di dalam sistem ini terjadi proses pengiriman SMS dan data yang didapat berasal dari *database*. Terjadi proses *select* pada form ini dari tabel datapenerbangan dan *inbox*.

```

//UNTUK KIRIM SMS
//BUKA DATABASE
include ("config.php");
//AMBIL NOMOR HP
$kode = "SELECT SenderNumber FROM inbox order by
ReceivingdateTime DESC LIMIT 1";

//AMBIL DATA UTK ISI PESAN
$kode2 = "SELECT pesawat, jamberangkat, jamdatang,
harga FROM datapenerbangan ORDER BY harga ASC LIMIT 4";
$kode3 = "SELECT substr(keberangkatan, -4, 3) FROM
datapenerbangan ORDER BY harga ASC LIMIT 4";
$kode4 = "SELECT substr(kedatangan, -4, 3) FROM
datapenerbangan ORDER BY harga ASC LIMIT 4";

```

```

//EKSEKUSI
$result = mysql_query($kode);
$result2 = mysql_query($kode2);
$result3 = mysql_query($kode3);
$result4 = mysql_query($kode4);

$hasil1 = mysql_fetch_row($result);//NOMOR HP TUJUAN
$hasil3 = mysql_fetch_row($result3);//KEBERANGKATAN
$hasil4 = mysql_fetch_row($result4);//KEDATANGAN
?>

<!--FORM UNTUK KIRIM SMS -->
<?php
$teks = "$hasil3[0] $hasil4[0]";
?>
<?php
$tanda = 1; ?>
<?php while ($hasil2=mysql_fetch_row($result2)) { ?>
<?php $tanda++;
$teks2[$tanda] = "$hasil2[0] $hasil2[1]-$hasil2[2] $hasil2[3]"; } ?>
<?php

$msg = $teks.'
'.$teks2[2].'
'.$teks2[3].'
'.$teks2[4].'
'.$teks2[5];

$sintak = "INSERT INTO outbox (DestinationNumber, TextDecoded,
CreatorID) VALUES ('$hasil1[0]', '$msg', 'Gammu')";
$sintak2 = "DELETE FROM datapenerbangan";
$sakhir = mysql_query($sintak);
$sakhir2 = mysql_query($sintak2);
?>

```

Kode Program 4.11 Bagian pengiriman SMS



#### 4.6. Testing

Testing adalah pengujian yang dilakukan terhadap keseluruhan sistem (secara lengkap) dan sistem yang telah terintegrasi untuk mengevaluasi apakah sistem yang dibuat telah sesuai dengan kebutuhan pengguna.

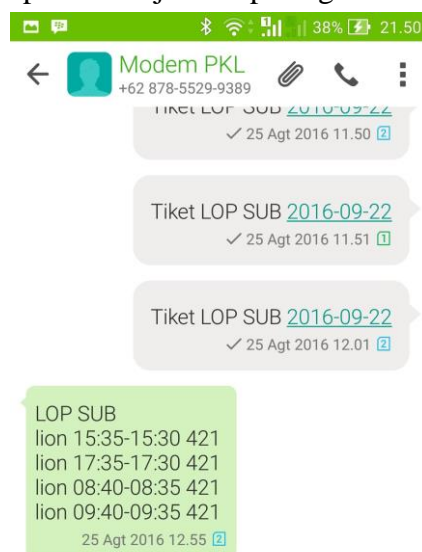
*Black box* testing adalah bagian dari testing itu sendiri. Cara pengujian *black box* sendiri tanpa membutuhkan pengetahuan tentang coding dan logika pemrograman. Pada tabel 4.1 ini ditunjukkan tentang bagaimana pengujian Layanan Informasi Tiket Pesawat, apakah telah memenuhi kriteria atau belum.

Tabel 4.1 *Black box* testing Layanan Informasi Tiket Pesawat

No	Data Masukkan	Kondisi Awal	Yang Diharapkan	Hasil Uji	Kesimpulan
1	Pelanggan melakukan sms dengan data yang benar	Program berjalan	Program dapat membalas pesan sesuai prosedur	Program dapat membalas pesan sesuai prosedur	[ X ] diterima [ ] ditolak
2	Pelanggan melakukan sms sekali dalam sehari dengan data benar	Program berjalan	Program dapat membalas pesan sesuai prosedur	Program dapat membalas pesan sesuai prosedur	[ X ] diterima [ ] ditolak
3	Pelanggan melakukan lebih dari satu kali sms	Program berjalan	Program tidak akan berjalan	Program tidak berjalan	[ X ] diterima [ ] ditolak
4	Pelanggan melakukan sms sekali dalam sehari namun data tidak benar	Program berjalan	Program dapat berjalan dan membalas dengan pesan "Maaf"	Program membalas dengan pesan "Maaf"	[ X ] diterima [ ] ditolak
5	Pelanggan melakukan	Program	Program	Program	[ X ] diterima

	sms berkali kali dalam sehari namun dengan data yang benar	berjalan	tidak akan berjalan meskipun data yang dimasukkan benar	tidak berjalan	[ ] ditolak
6	Pelanggan melakukan sms sekali dalam sehari dengan data yang benar namun tidak ada jadwal penerbangan pada hari itu	Program berjalan	Program akan membalas dengan pesan "Maaf"	Program membalas dengan pesan "Maaf"	[ X ] diterima [ ] ditolak
7	Pelanggan sms dengan data yang tidak benar	Program berjalan	Program dapat membalas pesan "Maaf"	Program membalas dengan pesan "Maaf"	[ X ] diterima [ ] ditolak

Berikut adalah gambar yang menunjukkan ketika pelanggan mengirimkan SMS dan mendapat balasan seperti ditunjukkan pada gambar 4.27



Gambar 4.27 Pelanggan SMS dan sistem membalas

Dan gambar 4.28 ini menunjukkan ketika sistem membalas dengan pesan “Maaf”



Gambar 4.28 Sistem membalas dengan pesan “Maaf”

#### 4.7. Kebutuhan Peralatan

Agar aplikasi dapat berjalan dengan lancar, maka peralatan minimal yang harus dimiliki *user* adalah :

1. Satu set *Personal Computer (PC)* / Laptop
2. Satu set modem
3. *Processor Intel Pentium 4*
4. 1000 Mb RAM

#### 4.8. Kebutuhan Sistem

Aplikasi hanya dapat berjalan di *Laptop* atau *PC* dengan Sistem Operasi *WINDOWS XP*, *WINDOWS 7*, *WINDOWS 8*, atau *WINDOWS 10* dan didalamnya telah terinstall *database* yang menggunakan *SQLyog Ultimate*