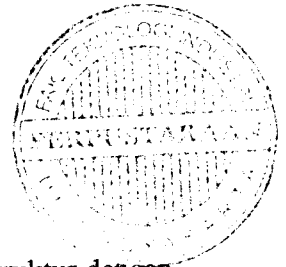


BAB III PERANCANGAN



3.1 Analisis Kebutuhan Perangkat Lunak

3.1.1 Metode Analisis

Metode analisis yang digunakan adalah metode analisis terstruktur dengan menggunakan teknik menentukan fakta, yaitu teknik mengumpulkan data dan menentukan fakta-fakta dalam mempelajari kondisi yang ada di lapangan serta dalam perancangan sistem menggunakan UML sebagai alat bantu.

Text editor penulisan sintaks menggunakan Macromedia Dreamweaver 8 karena dapat mendukung bahasa pemrograman PHP, CSS, Ajax, Java Script dan lain-lain. DBMS (*Database Management System*) yang digunakan sebagai pengolah data sistem adalah MySQL. Sedangkan untuk *web server* menggunakan xamp dan untuk aplikasi SMS Gateway menggunakan Gammu.

3.1.2 Hasil Analisis

Dari data hasil analisis selama penelitian dapat ditetapkan kebutuhan-kebutuhan yang akan digunakan dalam aplikasi ini. Kebutuhan tersebut meliputi: *Input* dan *output*, diantaranya yaitu:

3.1.2.1 Analisis Kebutuhan Proses

Kebutuhan proses Sistem Informasi Pengecekan Validasi Kendaraan Bermotor, meliputi:

a. Login

Proses login dilakukan admin sebelum memasuki Sistem Informasi Pengecekan Validasi Kendaraan Bermotor. Proses login ini bermanfaat untuk menjaga keamanan data. Proses login dimulai setelah admin memasukkan *username* dan *password*. Ketika admin sudah berhasil login maka admin berhak penuh untuk mengakses semua isi dari Sistem Informasi Pengecekan Validasi Kendaraan Bermotor ini.

b. Olah Data STNK

Proses data STNK meliputi: tambah, ubah, hapus data STNK. Data meliputi: nomor polisi, nomor rangka, nomor mesin dan masa berlaku pajak. Pengolahan data STNK ini diperuntukkan kepada admin dan pengolahan ini dapat dilakukan di halaman admin.

c. Manajemen SMS gateway

Proses manajemen SMS gateway meliputi : start service yang berguna untuk menyalakan service dari gammu, restart service yang berguna untuk menghentikan dan menyalakan kembali service dari gammu, stop service yang berguna untuk menghentikan service dari gammu dan start auto respond untuk menyalakan respond otomatis dalam proses mengirim sms ke user.

3.1.2.2 Analisis Kebutuhan Input

Input dari Sistem Informasi Pengecekan Validasi Kendaraan Bermotor terdiri dari 2 *user*, yaitu admin dan *user*. Admin memiliki wewenang penuh dalam pengolahan data yang ada di dalam *database* sedangkan *User* hanya bisa mengirimkan data kendaraan bermotor sesuai format yang telah ditentukan *via* SMS.

a. *Input* Admin

Input admin ini merupakan masukkan dari admin yaitu data STNK.

1. *Input* Data Login

Input data login berupa *username* dan *password*.

2. *Input* Data STNK

Input data STNK berupa segala sesuatu yang berhubungan dengan STNK seperti nomor polisi, nomor rangka dan nomor mesin dan masa berlaku pajak.

b. *Input* User

Input user Adalah masukkan data yang dimasukkan oleh *user* *via* SMS.

➤ *Input Data Nomor Polisi via SMS*

Input nomor polisi *via* SMS berupa nomor polisi kendaraan bermotor, nomor rangka kendaraan, dan nomor mesin yang ingin divalidasi ke Sistem Informasi Pengecekan Validasi Kendaraan Bermotor.

3.1.2.3 Analisis Kebutuhan Output

Data keluaran yang diperoleh oleh *user* hanya berupa data berdasarkan nomor polisi, nomor rangka dan nomor mesin yang di kirim melalui sms yaitu.

a. Informasi kesalahan format SMS

Output yang diberikan sistem yang berbasis SMS ini berupa SMS informasi bahwa format SMS yang dikirimkan oleh user salah atau tidak sesuai dengan format yang telah ditentukan dan sistem memberikan panduan SMS yang benar.

b. Informasi Valid Data Kendaraan

Output yang diberikan sistem informasi berbasis SMS ini berupa SMS informasi bahwa data yang dikirim oleh user valid dengan data yang ada pada database.

c. Informasi Tidak Valid Data Kendaraan

Output yang diberikan sistem informasi berbasis SMS ini berupa SMS informasi ketidak validan antara data yang dikirim oleh user dengan data yang ada pada database.

d. Informasi Data Kendaraan Tidak Ditemukan

Output yang diberikan sistem informasi berbasis SMS ini berupa SMS informasi bahwa data yang dikirim oleh user tidak ada / tidak ditemukan pada database.

3.2 Kebutuhan Antarmuka

Kebutuhan antarmuka dalam perancangan ini meliputi antarmuka / tampilan sebagai berikut :

a. Antarmuka Halaman Login

- b. Antarmuka Halaman Index
- c. Antarmuka Halaman Manajemen SMS
- d. Antarmuka Halaman Manajemen Gateway
- e. Antarmuka Halaman Manajemen Data STNK

3.3 Perancangan Perangkat Lunak

3.3.1 Metode Perancangan

Perancangan yang digunakan dalam penelitian ini dibagi dalam tiga kategori, yaitu perancangan sistem, perancangan *database*, dan perancangan *interface*. Untuk perancangan sistem, metode yang digunakan adalah metode perancangandengan menggunakan UML.

3.3.2 Hasil Perancangan

Berdasarkan analisis yang telah dilakukan maka dapat diketahui apa saja yang menjadi masukan sistem, proses yang digunakan sistem, keluaran sistem, serta antarmuka sistem yang dibuat, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan.

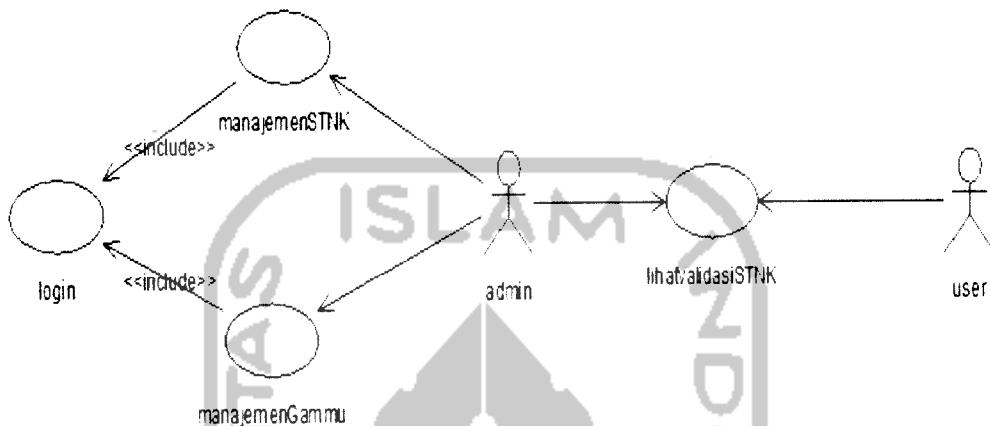
Perancangan sistem ini akan dibagi menjadi beberapa subsistem yaitu:

- a. Use Case Diagram
- b. Class Diagram
- c. Sequence Diagram
- d. Activity Diagram
- e. Perancangan Tabel basisdata.
- f. Perancangan Antamuka.
- g. Perancangan SMS.

3.3.2.1 Perancangan Use Case Diagram

Perancangan *Use Case Diagram* berfungsi untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari prespektif pengguna diluar sistem. Sebuah *Use Case* diagram mempresentasikan interaksi yang terjadi antara aktor dengan proses atau sistem yang dibuat.

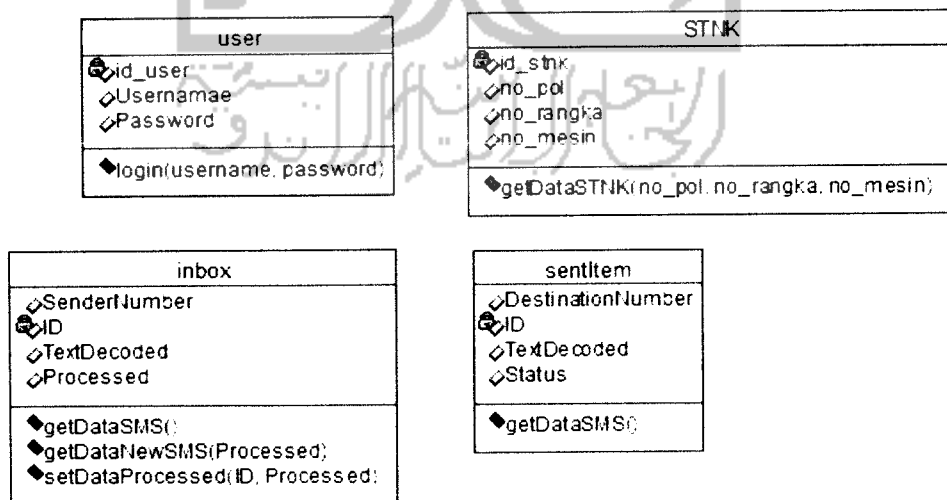
Untuk menggambarkan Aplikasi yang akan dibangun ini, *Use Case* diagram tersebut dapat dilihat pada gambar 3.1



Gambar 3.1 Use Case Diagram

3.3.2.2 Perancangan Class Diagram

Class Diagram merupakan gambaran struktur kelas, objek beserta hubungan antar kelas, pewarisan kelas dan lain-lain. Pada sistem yang dibangun, terdapat 4 kelas yaitu kelas *User*, *STNK*, *Inbox*, *Outbox*. *Class diagram* tersebut dapat dilihat pada gambar 3.2:



Gambar 3.2 Class Diagram

3.3.2.3 Perancangan Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi *horizontal* (objek – objek yang terkait) (Dharwiyanti, 2003).

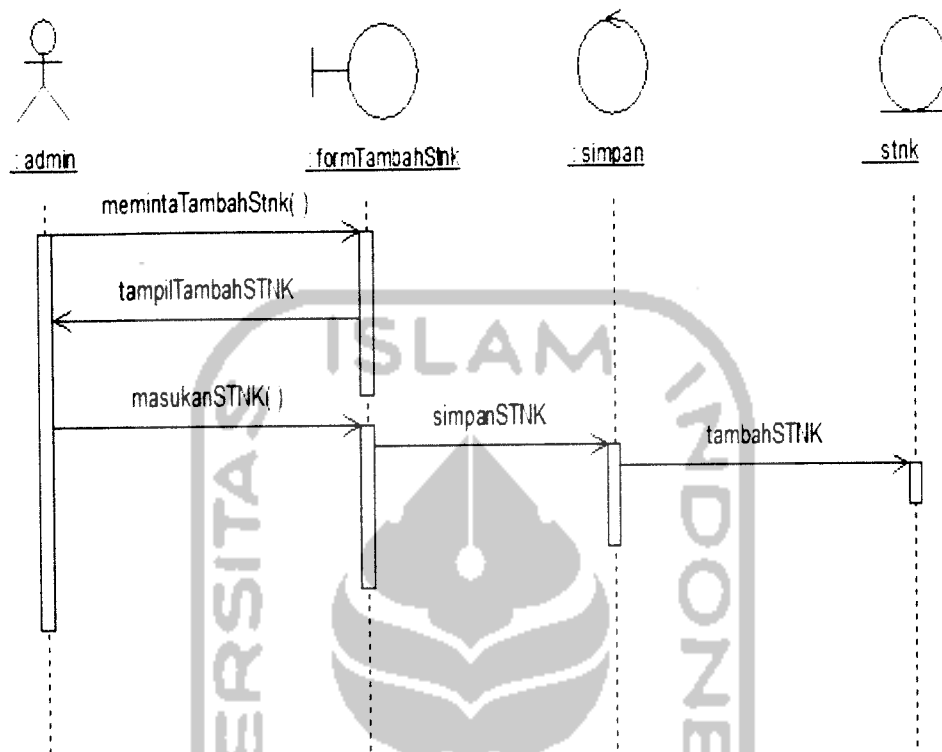
Sequence diagram digunakan untuk menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan use case. Sequence diagram juga dapat memperlihatkan tahap demi tahap proses yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case.

Sequence diagram biasanya digunakan untuk menggambarkan skenario atau rangkaian-rangkaian yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Dalam sistem ini terdapat sequence diagram yang menggambarkan aliran proses terhadap waktu dari pertama pengguna memulai sampai menghasilkan output yang diharapkan.

Dalam perancangan SMS gateway pengecekan validasi kendaraan bermotor ini, sequence diagram dibagi berdasarkan pengelompokkan class diagram diatas. Dimana setiap proses dalam sequence diagram ini memiliki interaksi antar objek yang berurutan.

1. Sequence Diagram Tambah STNK

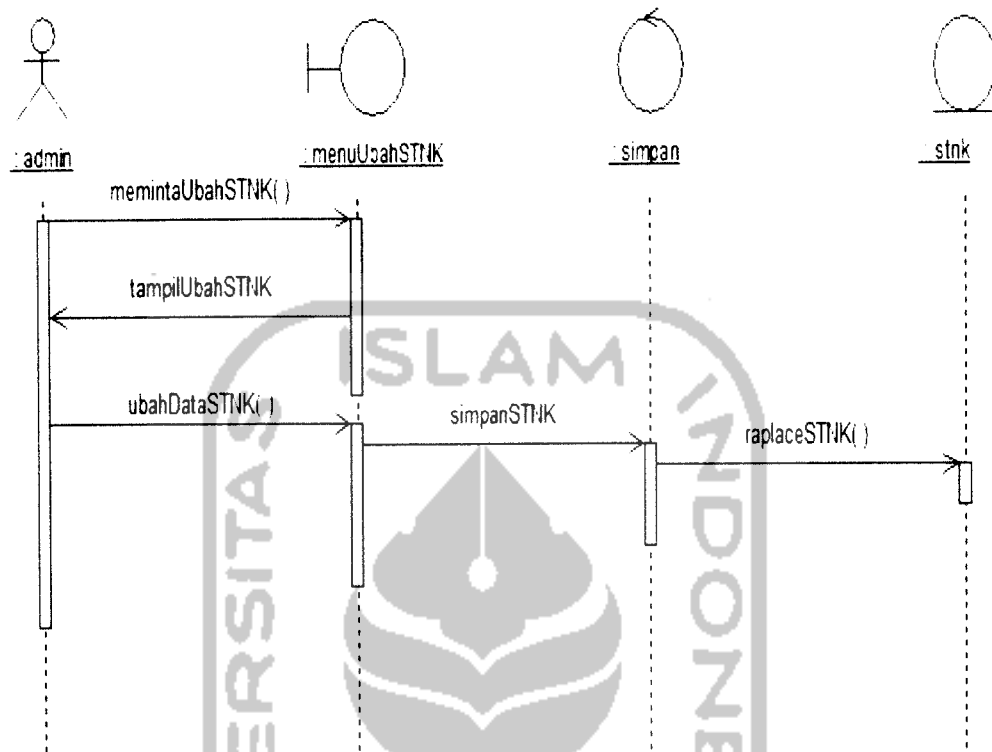
Aktor admin mengawali sequence ini dengan memanggil method `formTambahSTNK()` yang berisikan tampilan form penambahan data STNK, kemudian sistem menampilkan form `tampilTambahSTNK`, dan admin masukan data STNK yang berupa nomor polisi, nomor rangka, dan nomor mesin. Kemudian class `FormTambahSTNK` melakukan instansiasi ke class `simpan` dan menyimpan data STNK, dari method class `simpan` menyimpan method `tambahSTNK()` ke database tabel STNK. Sequence diagram tambah STNK dapat di lihat pada gambar 3.3:



Gambar 3.3 Sequence Diagram Tambah Data STNK

2. Sequence Diagram Ubah STNK

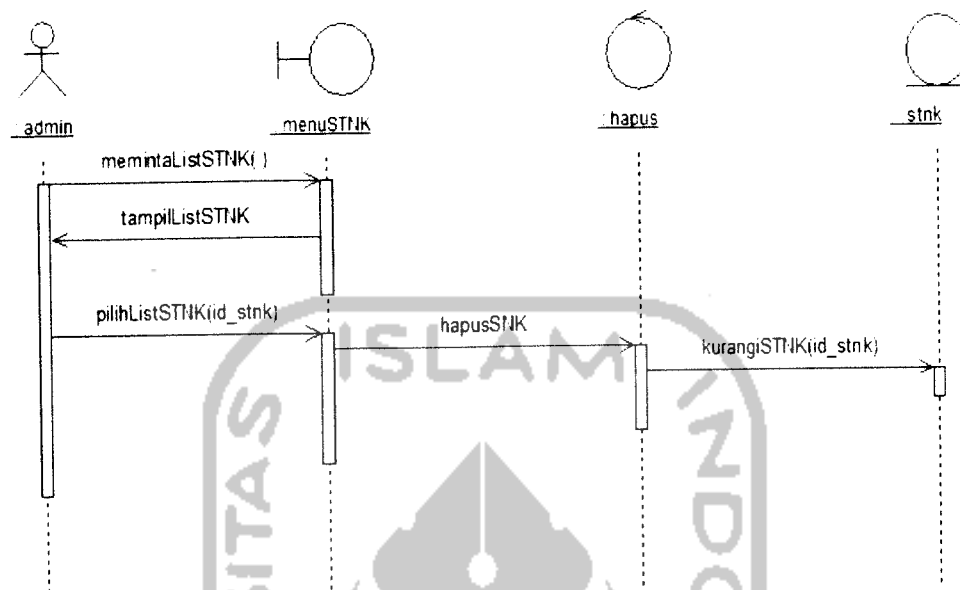
Aktor admin mengawali sequence ini dengan memanggil method `memintaUbahSTNK()`, kemudian class `menuUbahSTNK()` menampilkan `tampilUbahSTNK`, dan admin `ubahDataSTNK()` ke class `menuUbahSTNK`. Kemudian class `menuUbahSTNK` melakukan instansiasi ke class `simpan` dan menyimpan data STNK, dari method class `simpan` menyimpan method `replaceSTNK()` ke database tabel STNK. Sequence diagram ubah STNK dapat dilihat pada gambar 3.4:



Gambar 3.4 Sequence Diagram Ubah STNK

3. Sequece Diagram Hapus STNK

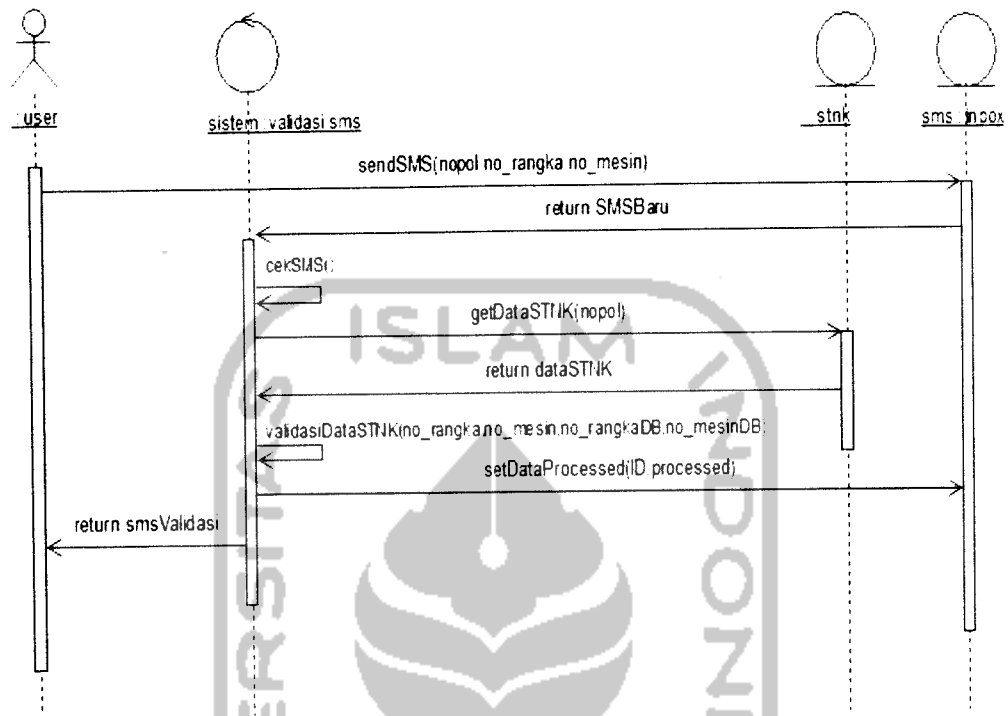
Aktor **admin** mengawali sequence ini dengan memanggil method `memintaListSTNK()`, kemudian class `menuListSTNK()` menampilkan `tampilListSTNK`, dan `admin pilihListSTNK()` ke class `menuSTNK`. Kemudian class `menuSTNK` melakukan instansiasi ke class `hapus` dan `hapus data STNK`, dari metod class `hapus` menghapus metod `kurangiSTNK(id_stnk)` ke database tabel `STNK`. Sequence Diagram `STNK` dapat di lihat pada gambar 3.5:



Gambar 3.5 Sequence Diagram Hapus STNK

4. Sequece Diagram Respon SMS

Aktor user biasa mengirimkan SMS pada sistem yang kemudian divalidasi format penulisan SMS lalu dilakukan pengecekan data. Hal yang dilakukan pertama adalah mengambil data STNK pada database berdasarkan nomor polisi yang telah di SMS oleh user. Setelah itu dilakukan validasi data SMS menggunakan nomor rangka dan nomor mesin kendaraan. Lalu dilakukan pengesetan data SMS mejadi telah diproses. Apabila data sesuai akan direspon valid oleh sistem, apabila data tidak diketemukan akan direspon tidak diketemukan dan apabila data tidak valid akan direspon data kendaraan tidak valid. Sequence Diagram proses SMS dapat di lihat pada gambar 3.6:



Gambar 3.6 Sequence Diagram Proses SMS

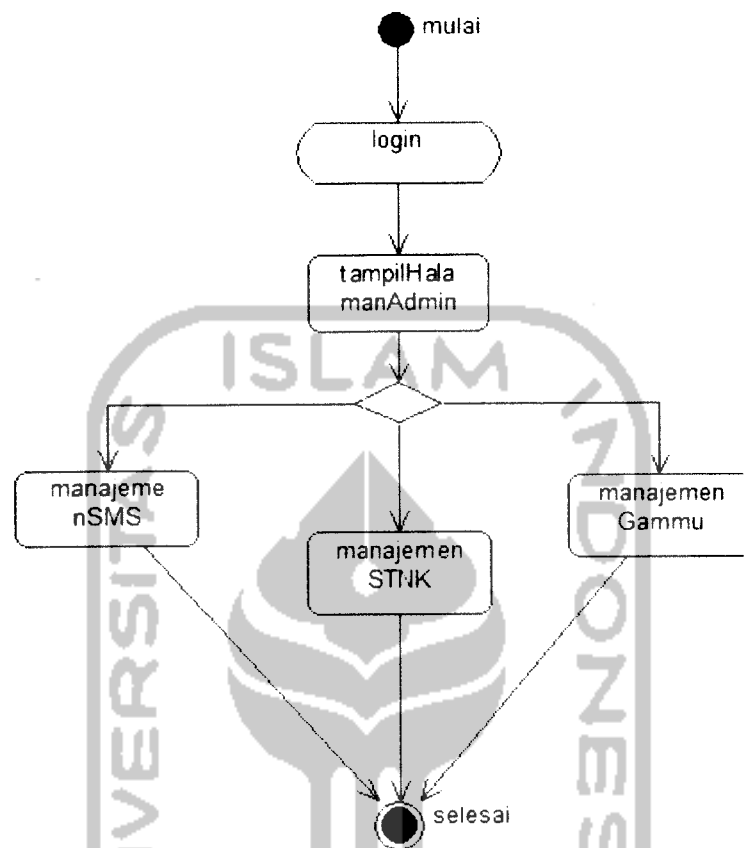
3.3.2.4 Perancangan Activity Diagram

Untuk menggambarkan aktivitas yang terjadi di dalam sistem digunakan *activity diagram*. Didalam *activity diagram* akan digambarkan berbagai aliran aktifitas dalam sistem yang akan dibangun, bagaimana aliran aktifitas dalam sistem, bagaimana aliran aktifitas berawal, *decision* yang mungkin terjadi dan bagaimana aktifitas itu berakhir.

Activity diagram umumnya tidak menggambarkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan proses yang terjadi.

1. *Activity Diagram* Admin

Admin memulai *activity* dengan login menampilkan *tampilHalamanAdmin* dan admin dapat memilih manajemen STNK dan *Activity Diagram* Admin dapat dilihat pada gambar 3.7:



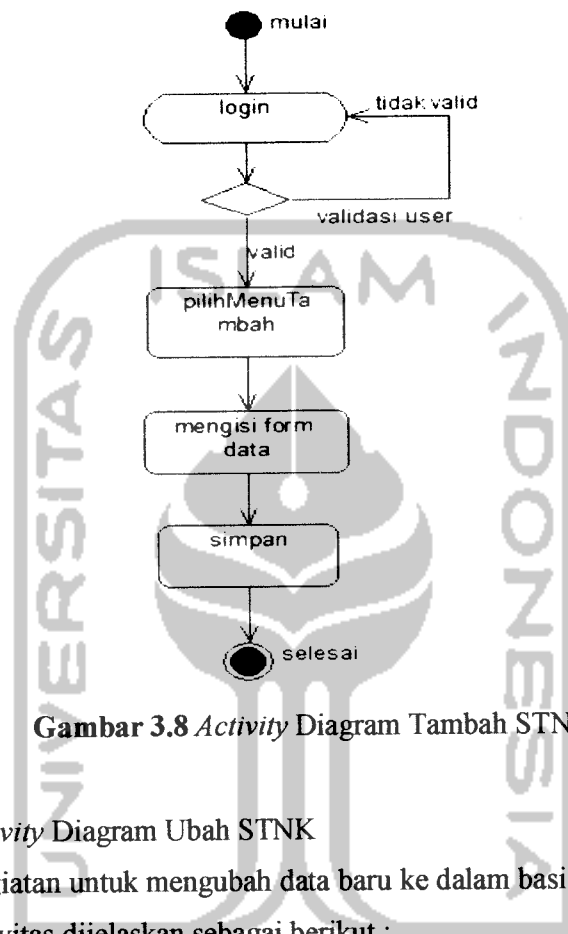
Gambar 3.7 Activity Diagram Admin

2. Activity Diagram Tambah STNK

Kegiatan untuk menambahkan data baru ke dalam basis data. Untuk urutan aktivitas dijelaskan sebagai berikut :

1. Admin memilih menu login, dan kemudian sistem akan memeriksa apakah username dan password yang dimasukan benar.
2. Admin memilih menu tambah STNK.
3. Selanjutnya user mengisikan data pada form dan menekan tombol simpan untuk mengirimkan data.
4. Sistem menyimpan data yang berhasil diisikan ke basisdata
5. Kemudian sistem mengakhiri kegiatan ini.

Activity Diagram Tambah STNK dapat dilihat pada gambar 3.8:



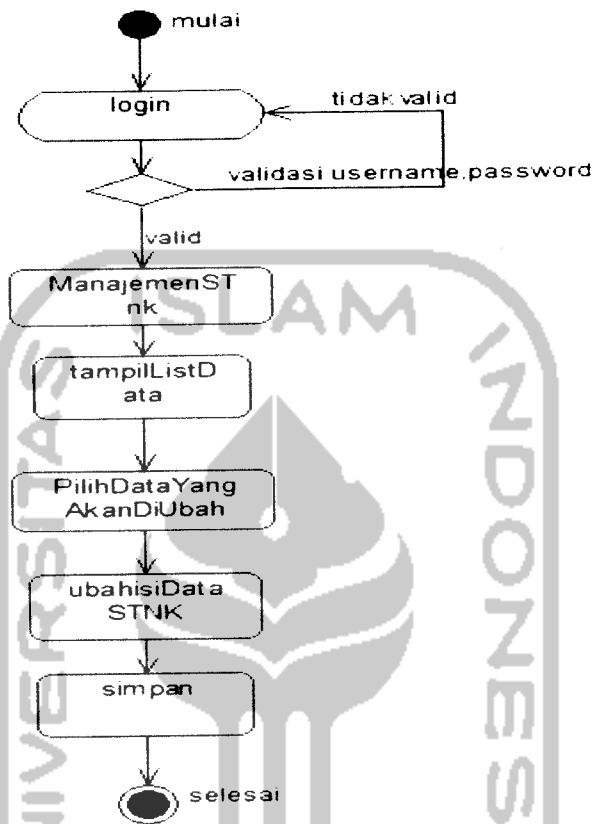
Gambar 3.8 *Activity Diagram* Tambah STNK

3. *Activity Diagram* Ubah STNK

Kegiatan untuk mengubah data baru ke dalam basis data. Untuk urutan aktivitas dijelaskan sebagai berikut :

1. Admin memilih menu login, dan kemudian sistem akan memeriksa apakah username dan password yang dimasukan benar.
2. admin memilih menu manajemen STNK.
3. Selanjutnya sistem menampilkan listDataSTNK.
4. Admin pilihDataYangAkanDiubah.
5. Sistem menampilkan data yang akan diubah, dan admin ubahDataSTNK.
6. Sistem menyimpan data yang berhasil diisikan ke basisdata
7. Kemudian sistem mengakhiri kegiatan ini.

Activity diagram ubah STNK dapat dilihat pada gambar 3.9:



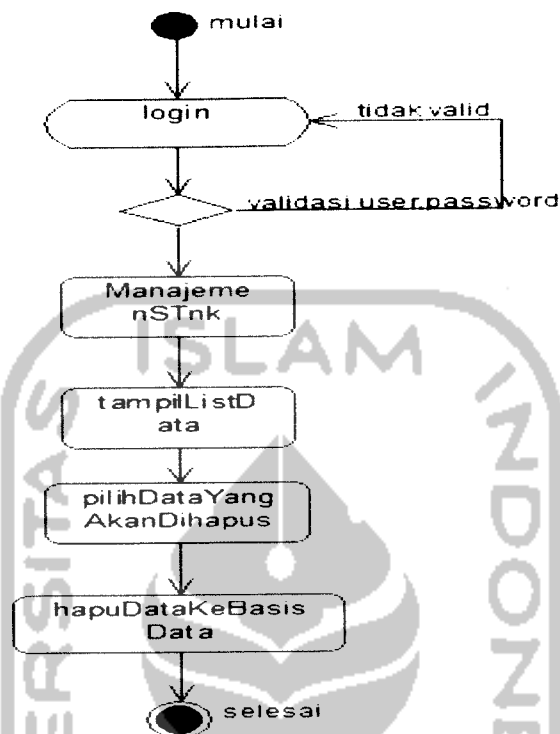
Gambar 3.9 Activity Diagram Ubah STNK

4. Activity Diagram Hapus STNK

Kegiatan untuk menghapus data pada basis data. Untuk urutan aktivitas dijelaskan sebagai berikut :

1. Admin memilih menu login, dan kemudian sistem akan memeriksa apakah username dan password yang dimasukan benar.
2. admin memilih menu manajemen STNK.
3. Selanjutnya sistem menampilkan listDataSTNK.
4. Admin pilihData YanAkanDihapus.
5. Hapus data yang sudah dipilih ke basisdata.
6. Kemudian sistem mengakhiri kegiatan ini.

Activity diagram hapus STNK dapat dilihat pada gambar 3.10:



Gambar 3.10 Activity Diagram Hapus STNK

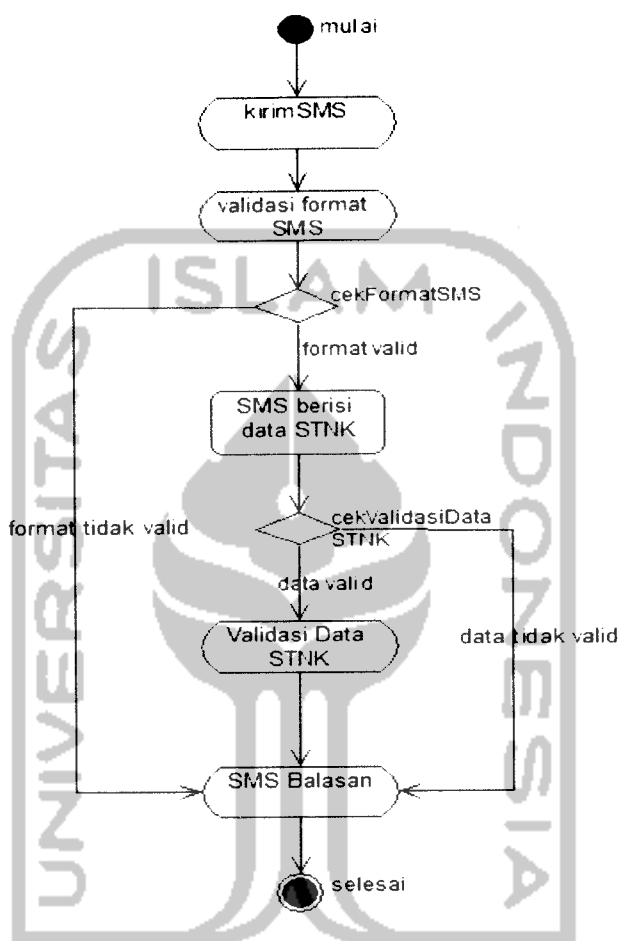


5. Activity Diagram Proses SMS STNK

Kegiatan untuk proses SMS, Untuk urutan aktivitas dijelaskan sebagai berikut :

1. User mengirim SMS.
2. SMS yang baru masuk dalam database akan divalidasi format SMS, jika format SMS tidak valid sistem akan membalas, tetapi jika format SMS benar sistem akan memproses data SMS STNK.
3. Setelah diproses SMS akan dicek validasi data STNK jika valid sistem akan memberi informasi valid sesuai didatabase STNK dan akan sistem akan membalas SMS tetapi jika tidak valid sistem akan langsung membalas dan memberikan contoh SMS yang benar.
4. Kemudian sistem mengakhiri kegiatan ini.

Activity diagram respon SMS STNK dapat dilihat pada gambar 3.11:



Gambar 3.11 Activity Respon SMS

3.3.2.5 Perancangan Tabel Basisdata

Perancangan tabel basis data merupakan proses pembangunan sistem yang meliputi perancangan struktur tabel.

3.3.2.5.1 Struktur Tabel

Sistem yang dibangun ini menggunakan tabel *default* yang dimiliki oleh Gammu, tabel yang digunakan adalah tabel inbox, outbox dan sentitems. Selain menggunakan tabel *default* dari Gammu sistem ini juga menggunakan beberapa penambahan tabel yang digunakan untuk aplikasi Sistem Informasi Pengecekan Validasi Kendaraan Bermotor ini. Berikut struktur tabel dari masing-masing tabel:

1. Tabel *Default* Gammu

a. Tabel Inbox

Tabel 3.1 merupakan tabel inbox yang digunakan untuk menerima SMS yang masuk. SMS yang masuk ini akan ditampung kedalam *field* TextDecoded beserta nomor pengirim ke dalam *field* SenderNumber dan waktu SMS diterima ke dalam *field* ReceivingDateTime. *Field* ID pada tabel inbox bersifat *unique*.

Tabel 3.1 Tabel Inbox

Field	Type	Null	Default
UpdateInDB	Timestamp	Yes	CURRENT_TIMESTAMP
ReceivingDateTime	Timestamp	Yes	0000-00-00 00:00:00
Text	Text	No	
SenderNumber	varchar(20)	No	
Coding	enum('Default_No_Compression')	No	Default_No_Compression
UDH	Text	No	
SMSCNumber	varchar(20)	No	
Class	int(11)	No	-1
TextDecoded	varchar(160)	No	
ID	int(10)	No	
RecipientID	Text	No	
Processed	enum('false','true')	No	False

b. Tabel Outbox

Tabel 3.2 merupakan tabel outbox yang digunakan untuk mengirimkan SMS dengan cara memasukkan teks yang akan dikirim ke dalam *field* TextDecoded beserta nomor tujuan kedalam *field* DestinationNumber. *Field* ID pada tabel outbox bersifat *unique*.

Tabel 3.2 Tabel Outbox

Field	Type	Null	Default
UpdateInDB	Timestamp	Yes	CURRENT_TIMESTAMP
InsertIntoDB	Timestamp	Yes	0000-00-00 00:00:00
SendingDateTime	Timestamp	Yes	0000-00-00 00:00:00
Text	Text	Yes	NULL
DestinationNumber	varchar(20)	No	
Coding	enum('Default_No_Compression')	No	Default_No_Compression
UDH	Text	Yes	NULL
Class	int(11)	Yes	-1
TextDecoded	varchar(160)	No	
ID	int(10)	No	
MultiPart	enum('false','true')	Yes	False
RelativeValidity	int(11)	Yes	-1
SenderID	varchar(255)	Yes	NULL
SendingTimeOut	Timestamp	Yes	0000-00-00 00:00:00
DeliveryReport	enum('default','yes','no')	Yes	Default
CreatorID	Text	No	

c. Tabel Sentitems

Tabel 3.3 merupakan tabel sentitems yang digunakan untuk menyimpan SMS yang telah dikirimkan. *Field* ID pada tabel sentitems bersifat *unique*.

Tabel 3.3 Tabel Sentitems

Field	Type	Null	Default
UpdateInDB	Timestamp	Yes	CURRENT_TIMESTAMP
InsertIntoDB	Timestamp	Yes	0000-00-00 00:00:00
SendingDateTime	Timestamp	Yes	0000-00-00 00:00:00
DeliveryDateTime	Timestamp	Yes	NULL
Text	Text	No	
DestiantionNumber	varchar(20)	No	
Coding	enum('Default_No_Compression')	No	Default_No_Compression
UDH	Text	No	
SMSCNumber	varchar(20)	No	
Class	int(11)	No	-1
TextDecoded	varchar(160)	No	
ID	int(10)	No	0
SenderID	varchar(255)	No	
SequencePosition	int(11)	No	1
Status	enum('SendingOK', 'SendingOKNoReport', 'SendingError', 'DeliveryOK', 'DeliveryFailed', 'DeliveryPending', 'DeliveryUnknown', 'Error')	No	SendingOK

Field	Type	Null	Default
Statuserror	int(11)	No	-1
TPMR	int(11)	No	-1
RelativeValidity	int(11)	No	-1
CreatorID	Text	No	

2. Tabel Sistem Informasi Pengecekan Validasi Kendaraan Bermotor

a. Tabel user

Tabel 3.4 merupakan tabel user yang digunakan untuk menyimpan user yang berhak masuk ke Sistem Informasi Pengecekan Validasi Kendaraan Bermotor. *Field id_user* pada tabel user bersifat *unique*.

Tabel 3.4 Tabel User

Field	Type	Null	Default
id_user	Int(5)	No	
Username	varchar(100)	No	
Password	varchar(32)	No	

b. Tabel STNK

Tabel 3.5 merupakan tabel STNK yang digunakan untuk menyimpan data-data STNK Sistem Informasi Pengecekan Validasi Kendaraan Bermotor. *Field id_stnk* pada tabel STNK bersifat *unique*.

Tabel 3.5 Tabel STNK

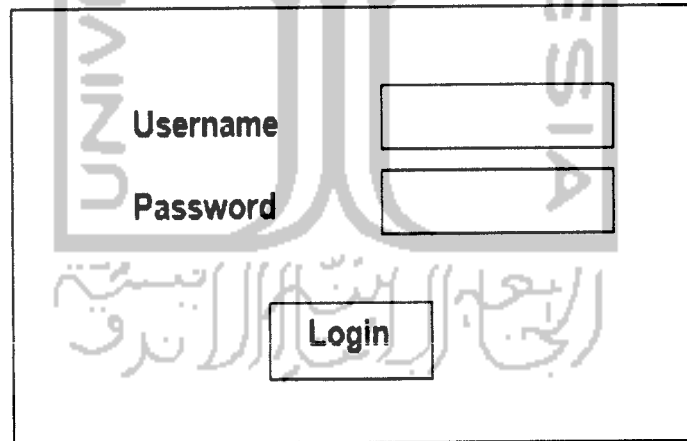
Field	Type	Null	Default
id_stnk	Int(11)	No	
Nopol	varchar(10)	No	
No_rangka	varchar(20)	No	
No_mesin	varchar(20)	No	

3.3.2.6 Perancangan Halaman Antarmuka

Rancangan antarmuka pada Sistem Informasi Pengecekan Validasi Kendaraan Bermotor ini dibuat dengan sederhana sehingga diharapkan mampu menciptakan suatu Sistem Informasi Manajemen yang mudah digunakan oleh pengguna. Perancangan antarmuka aplikasi ini terdiri dari lima halaman sebagai berikut:

a. Antarmuka Halaman Login

Halaman login merupakan halaman yang akan dijumpai oleh admin ketika pertama kali membuka aplikasi ini. Halaman ini berisi *form* login yang akan diisi oleh admin yang akan mengakses sistem, sehingga hanya admin yang mempunyai hak akses yang berhak mengakses sistem. Rancangan antarmuka halaman login dapat dilihat pada Gambar 3.12:



The image shows a login form with a white background and a black border. It contains three main elements: a label 'Username' followed by a rectangular input field, a label 'Password' followed by another rectangular input field, and a 'Login' button centered below the input fields. The background of the image is a watermark of the logo of Universitas Islam Indonesia, featuring a stylized 'U' and 'I' with the word 'ISLAM' and 'INDONESIA' around it, and Arabic calligraphy at the bottom.

Gambar 3.12 Rancangan Antarmuka Halaman Login

b. Antarmuka Halaman Index

Halaman index merupakan halaman yang berisi mengenai halaman admin. Rancangan antarmuka halaman admin dapat dilihat pada Gambar 3.13.

Beranda	SMS	SMS Gateway	Database STNK	LogOut
Sistem Informasi Pengecekan Validasi Kendaraan Bermotor				
Beranda				
Selamat Datang di SI Pengecekan				

Gambar 3.13 Rancangan Antarmuka Halaman Index

c. Antar Muka Halaman Manajemen SMS

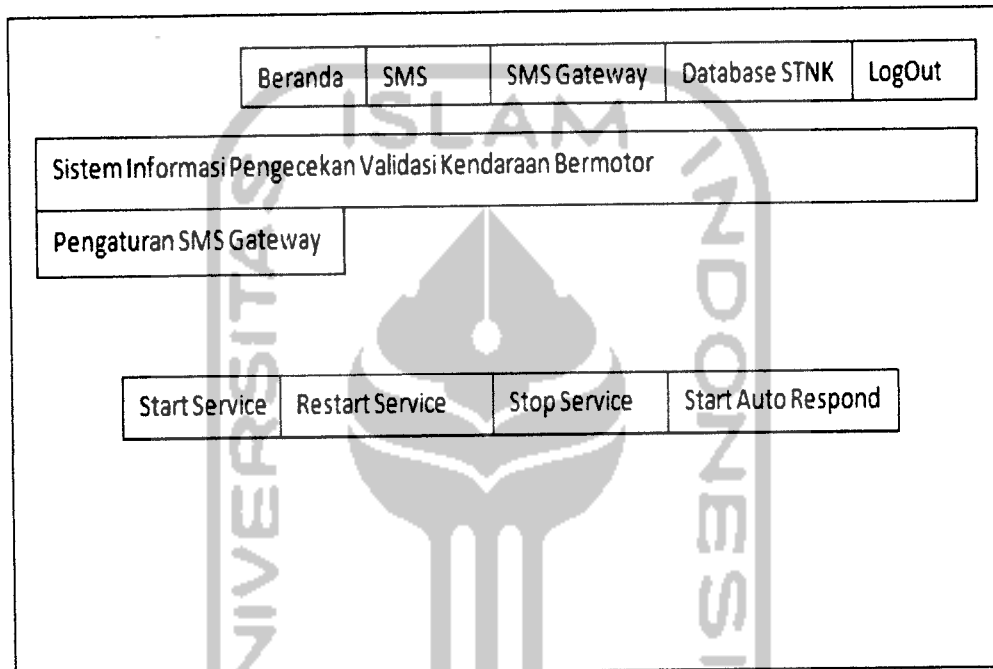
Halaman manajemen SMS merupakan halaman yang menampilkan tulis SMS, pesan masuk, dan pesan keluar. Rancangan antarmuka manajemen gateway dapat dilihat pada Gambar 3.14.

Beranda	SMS	SMS Gateway	Database STNK	LogOut
Sistem Informasi Pengecekan Validasi Kendaraan Bermotor				
Pengaturan SMS				
Tulis SMS	Pesan Masuk	Pesan Keluar		
No Tujuan :	<input type="text"/>			
Pesan :	<input type="text"/>			
				Kirim

Gambar 3.14 Rancangan Atarmuka Halaman Manjemen SMS

d. Antarmuka Halaman Manajemen Gateway

Halaman manajemen gateway merupakan halaman yang menampilkan pengaturan dari gammunya. Rancangan antarmuka manajemen gateway dapat dilihat pada Gambar 3.15.



Gambar 3.15 Rancangan Antarmuka Halaman Manajemen Gateway

e. Antarmuka Halaman Manajemen Data STNK

Halaman Manajemen data STNK ini berisi mengenai informasi manajemen data-data STNK yang terdapat dalam Sistem Informasi Pengecekan Validasi Kendaraan Bermotor. Rancangan antarmuka halaman data STNK dapat dilihat pada Gambar 3.16.

Gambar 3.16 Rancangan Antarmuka Halaman Manajemen Database STNK

3.3.2.6.1 Perancangan Format SMS

Adapun format SMS yang dikirimkan oleh *user* atau pengguna dan akan diterima aplikasi adalah sebagai berikut.

Tabel 3.6 Format Pesan Masuk

No	Format SMS	Keterangan
1	cek<spasi>nopol<spasi>no.rangka<spasi>no.mesin>	Untuk melakukan pengecekan validasi nomor kendaraan bermotor

Tabel 3.7 Format Balasan SMS

No	Format SMS balasan	Keterangan
1	Format sms salah. Silahkan ketik "cek<spasi>nopol tanpa spasi<spasi>no.rangka tanpa spasi<spasi>no.mesin tanpa spasi"	Jika sms salah tidak sesuai balasan
2	Data kendaraan nomor polisi: nopol (VALID)	Jika isi SMS yang dikirim user valid
3	Data kendaraan nomor polisi : nopol (TIDAK VALID)	Jika isi SMS yang dikirim user tidak valid
4	Data kendaraan nomor polisi : nopol (DATA TIDAK DITEMUKAN)	Jika isis SMS yang dikirim user tidak ada dalam data base