

IV HASIL DAN PEMBAHASAN

A. Perancangan Sistem Informasi Translate Indonesia – Lampung

Pada bab ini akan dibahas lebih lanjut tahapan-tahapan yang dilakukan dalam perancangan aplikasi Sistem Informasi Penerjemah *Online* Bahasa Lampung pada tahap implementasi menjadi sebuah aplikasi utuh. Nantinya Sistem Informasi Penerjemah *Online* Bahasa Lampung bisa digunakan sebagai sarana untuk menginformasikan bahasa Indonesia dalam bahasa Lampung dan sebaliknya. Pada tugas akhir ini tidak di bahas tentang kaidah tata bahasa Lampung, di karenakan kurangnya informasi tentang penulisan yang baku untuk kata-kata bahasa Lampung yang sebenarnya.

Penjelasan umum mengenai aplikasi Sistem Informasi Penerjemah *Online* Bahasa Lampung adalah sebagai berikut :

1. Aplikasi Sistem Informasi Penerjemah *Online* Bahasa Lampung adalah aplikasi berbasis web,
2. Sistem Informasi Penerjemah *Online* Bahasa Lampung mampu menampung data – data yang dimasukkan *administrator*,

Tools yang akan digunakan dalam pembuatan aplikasi Informasi Penerjemah *Online Bahasa Lampung* adalah:

1. Macromedia Dreamweaver 8 sebagai *editor* PHP dan HTML, yang memiliki fitur pembuatan *user interface* untuk HTML dan memiliki kemampuan *code completion* untuk CSS,
2. Photoshop sebagai *editor* desain tampilan grafis,
3. PHP adalah bahasa pemrograman *web* dinamis versi yang digunakan ialah PHP 5.
4. MySQL adalah DBMS *open source* yang digunakan ialah MySQL versi 5 versi MySQL 5.0.51a.
5. XAMPP adalah aplikasi *web server* yang umum digunakan dan *open source* versi yang digunakan ialah XAMPP versi 2.6

B. Penerapan metode *Waterfall* pada sistem yang di rancang

1. *User Requirement*

Pada tahap awal perancangan sistem informasi translate Indonesia Lampung dilakukan pencarian kebutuhan-kebutuhan *user*. Pencarian kebutuhan *user* mengambil beberapa contoh dari ketua adat di kampung Menggala, remaja suku Lampung yang tinggal di Bandar Lampung, remaja dari luar suku Lampung yang tinggal di Bandar Lampung, guru bahasa Lampung. Setelah dikumpulkan maka didapatkan *user requirement* sebagai berikut:

- a. Tampilan web sederhana agar memudahkan *user*
- b. Jangan terlalu banyak tombol navigasi

- c. Cepat menampilkan hasil terjemahan
- d. Ada pesan kesalahan jika terjadi *error*
- e. Bisa *copy paste* dokumen
- f. Sederhana tetapi menarik
- g. Semua kata berbahasa Lampung dapat ditemukan saat dilakukan *translate*
- h. Jika kata tersebut tidak terdapat di *database* untuk diterjemahkan maka ditampilkan kata tersebut
- i. Jika ada kata yang tidak diterjemahkan kata tersebut berubah warna jadi merah atau biru sehingga bisa langsung diketahui
- j. Tidak menyusahkan

Kemudian dilakukan analisa kebutuhan dan menghasilkan uraian sebagai berikut;

1. Kriteria pemakai sistem terdiri atas *Administrator* dan *User*. *Administrator* adalah orang yang bertanggung jawab dalam pengelolaan data baik *input*, *edit*, hapus dan *update* data keseluruhan. *User* adalah orang yang mengakses sistem untuk melihat dan mencari data yang dibutuhkan.
2. Kebutuhan *Administrator*, yaitu kebutuhan yang akan dibuat pada sistem agar *administrator* mampu menggunakan sistem sesuai kriteria antara lain
 - a. *Form login administrator*, agar hanya *administrator* yang bisa mengakses fasilitas *administrator* oleh karena itu perlu dibuat *form login* di mana nanti *administrator* akan memasukkan *login* dan *password* untuk bisa masuk.

- b. *Form input*, lihat, *edit* dan hapus *administrator*, memungkinkan *administrator* untuk menambah, melihat, meng-*edit* dan menghapus *administrator* yang terdaftar.
 - c. *Form input*, lihat, *edit* dan hapus kata, memungkinkan *administrator* untuk menambah, melihat, megedit dan menghapus kata yang terdaftar.
 - d. *Form* keluar, memungkinkan *administrator* untuk keluar secara aman dari sistem.
3. Kebutuhan *User*, yaitu kebutuhan yang akan dibuat pada sistem agar *user* mampu menggunakan sistem sesuai kriteria antara lain :
- a. Melakukan penerjemahan kata
 - b. Melakukan penerjemahan kalimat

2. Desain Sistem

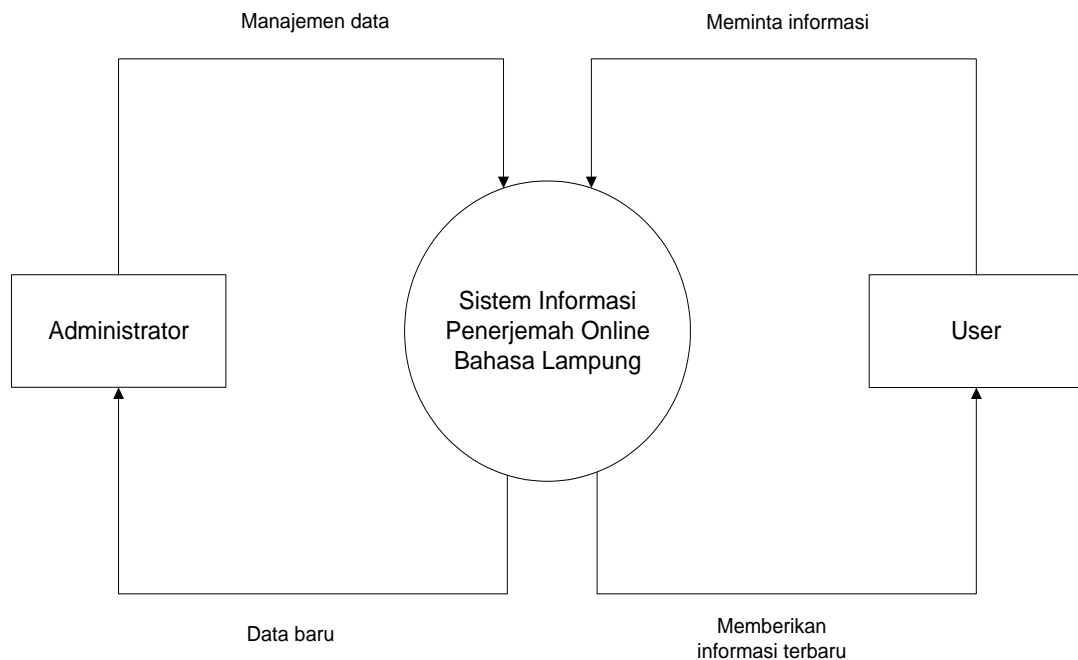
Setelah mendapatkan apa saja kebutuhan yang harus disediakan oleh Sistem Informasi Penerjemah *Online* Bahasa Lampung baik untuk *administrator* dan *user* dari tahap analisis kebutuhan, langkah selanjutnya ialah memodelkan sistem dalam, *Data Context Diagram* (DCD), *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD).

a . *Data Context Diagram* (DCD)

Data Context Diagram menggambarkan interaksi antara entitas dengan aplikasi yang dibuat dalam hal memberikan maupun mendapatkan data. *Data Context Diagram* adalah bagian dari *Data Flow Diagram* (DFD)

yang berfungsi memetakan model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

Berikut adalah *Data Context Diagram* (DCD) dari Sistem Informasi Penerjemah *Online* Bahasa Lampung:



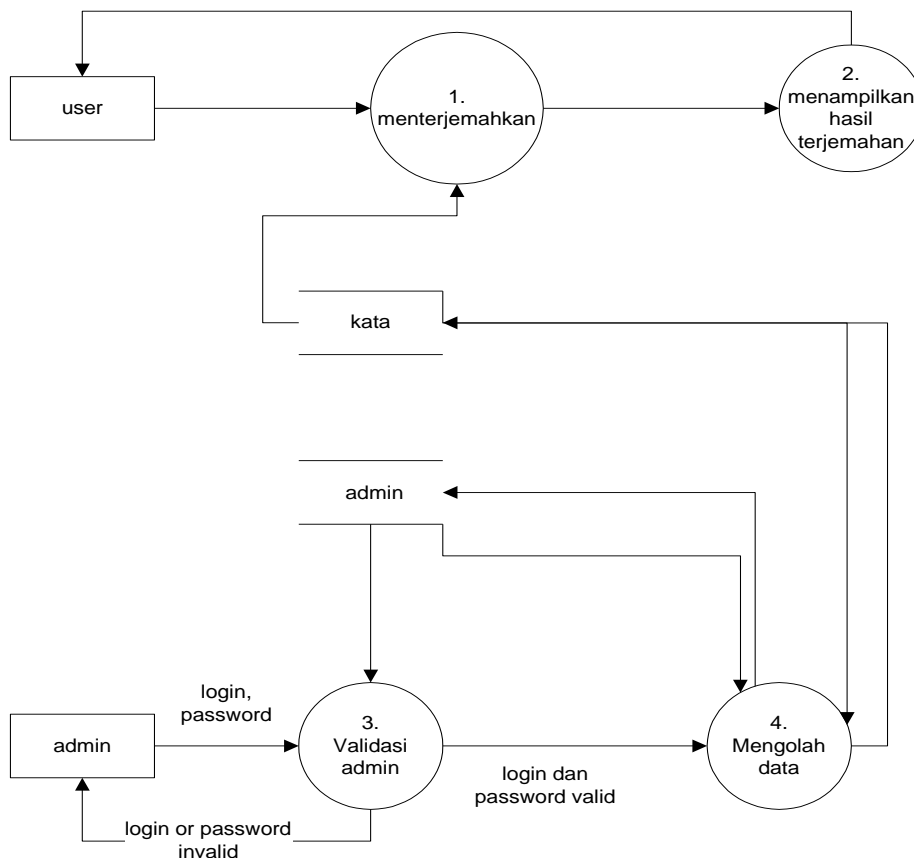
Gambar 26. Data Context Diagram

b. Data Flow Diagram

Data Flow Diagram (DFD) adalah sebuah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output*. DFD dapat digunakan untuk menyajikan sebuah

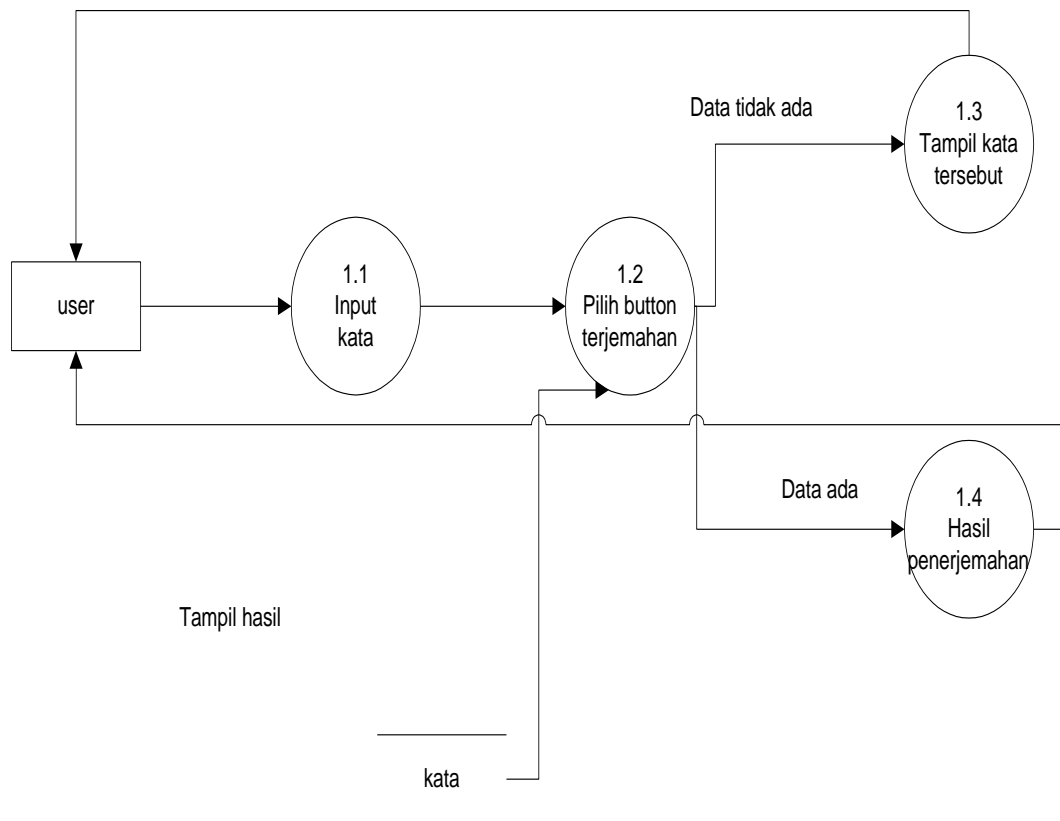
sistem atau perangkat lunak pada setiap tingkat abstraksi. Demikianlah, DFD memberikan suatu mekanisme bagi pemodelan fungsional dan pemodelan aliran informasi.

Pada DFD level 1 terdapat dua user. *User* biasa hanya bisa menerjemahkan kata atau kalimat dan melihat hasilnya. Proses menerjemahkan kata atau kalimat dengan mengambil terjemahan kata itu dari *database* kata. Sedangkan admin harus melakukan validasi terlebih dahulu untuk mengolah data. Jika *username* dan *password* sesuai *database* maka akan masuk ke sistem jika tidak akan diberikan pesan *error* untuk *login* kembali.



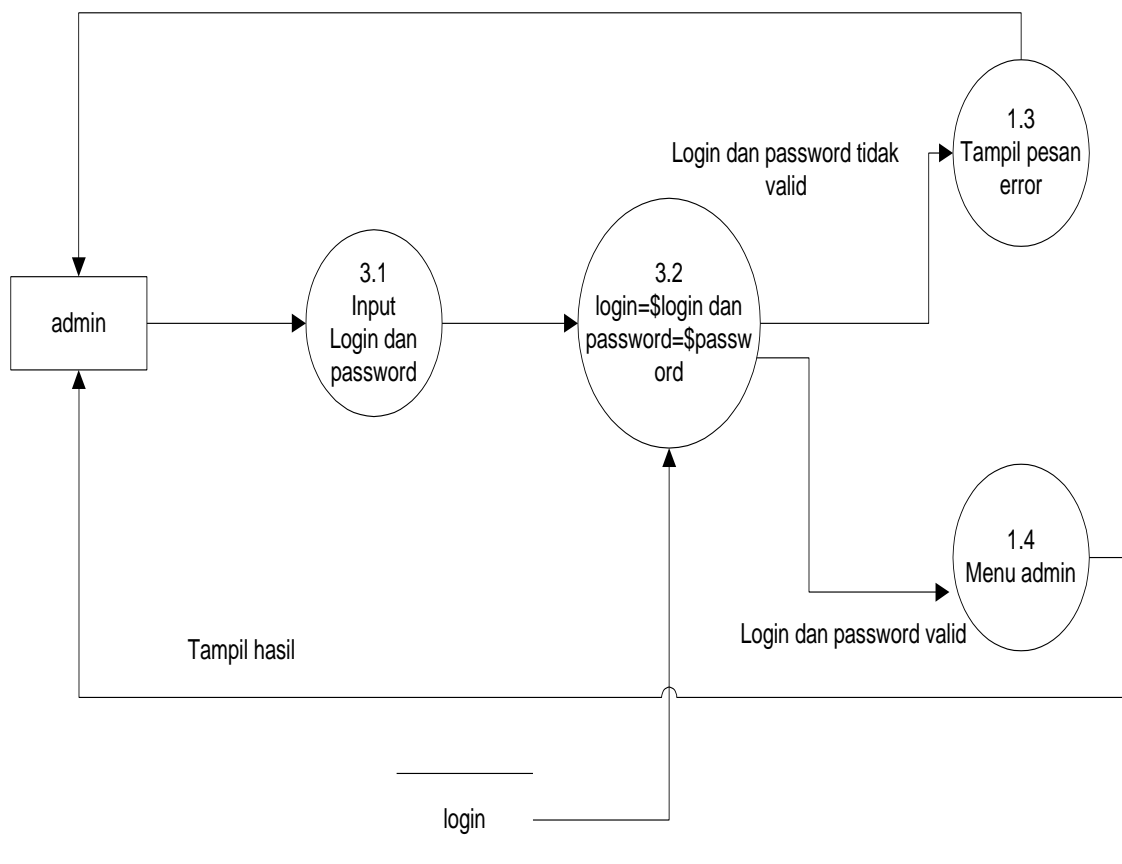
Gambar. 27. Data Flow Diagram Level 1

DFD level 2 proses 1 yaitu proses menerjemahkan kata bagi *user* biasa. *User* Melakukan *input* kata atau kalimat kemudian memilih *button* terjemahan yang terdiri atas enam macam pilihan. Setelah menekan tombol *translate* maka akan diberikan hasil terjemahannya. Jika kata yang dimaksud tidak terdapat dalam *database* maka akan ditampilkan kata yang dimaksud.



Gambar 28. DFD level 2 proses 1

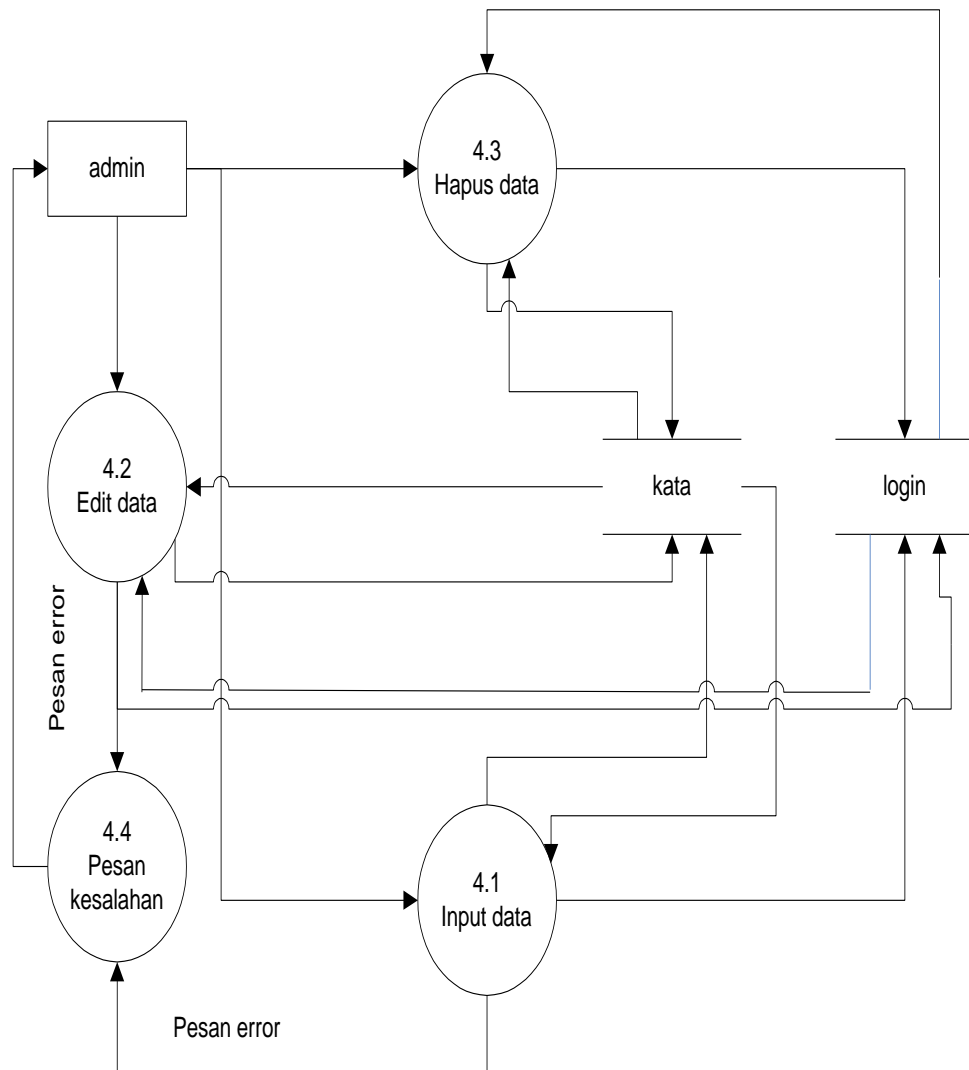
DFD level 2 proses 3 adalah proses validasi untuk admin. Pengguna ini melakukan proses memasukkan *login* dan *password* kemudian hasil itu dicocokkan dengan data yang ada di dalam *database login*. Jika sesuai masuk ke menu admin, jika tidak akan tampil pesan *error* dan kembali *login*.



Gambar 29. DFD level 2 proses 3

DFD level 2 proses 4 yaitu proses mengelola data yang dilakukan admin. Proses hapus data, *edit* data, dan *input* data dilakukan pada *table login* dan

kata pada *database*. Jika terdapat kesalahan dalam proses *edit* data dan *input* data maka akan ditampilkan pesan kesalahan.



Gambar 30. DFD level 2 proses 4

c. Process Spesifikasi (P-Spec)

1.P-Spec *Administrator*

P-Spec 3

a. P-Spec proses autentifikasi *administrator*

Masukan : *Login dan Password*

Keluaran : Menu *administrator*

Algoritma : If (*Login dan Password = valid*)

Tampil menu *administrator*

Else

Tampil pesan kesalahan

b. P-Spec pengolahan data

1. P-Spec *input data*

P-Spec 4.1

Masukan : *input data baru*

Keluaran : Tampilan data baru

Algoritma : if (data baru=valid)

Tampil data baru

Else

Tampil pesan kesalahan

2. P-Spec *Edit* data

P-Spec 4.2

Masukan : *Edit* data baru

Keluaran : Tampilan data baru

Algoritma : if (data baru=valid)

Tampil data baru

Else

Tampil pesan kesalahan

3. P-Spec Hapus data

P-Spec 4.3

Masukan : Hapus data

Keluaran : Tampilan data baru

Algoritma : if (data baru=valid)

Tampil data baru

2. P-Spec *User*

P-spec 1.1 proses memasukkan data kata atau kalimat

Masukan : kata atau kalimat

Keluaran : Tampilan hasil terjemahan

Algoritma : if (kata=valid)

Tampil hasil pencarian

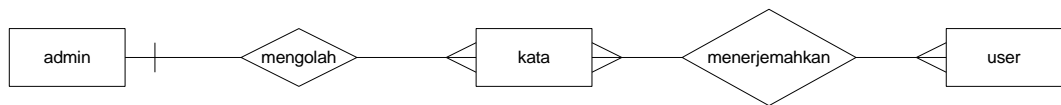
Else

Tampil pesan kesalahan

d. Entity Relationship Diagram (ERD)

ERD Yaitu diagram yang menggambarkan keterhubungan antar objek data.

Berikut adalah *Entity Relationship Diagram (ERD)*



Gambar 31. *Entity Relational Diagram*

Pada ERD diatas terlihat aktifitas yang dapat di kerjakan oleh admin dan *user* dimana admin berfungsi sebagai pengolah data, tiap admin bisa memasukan data dan kata – kata yang di inginkannya dan menterjemahkan kata, sedangkan untuk *user* hanya bias menerjemahkan kata yang di inginkan pada sistem ini.

e. Desain Database

Berdasarkan analisa kebutuhan di atas, dirancang basis data Sistem Informasi

Penerjemah *Online* Bahasa Lampung dengan tabel – tabel sebagai berikut:

a. Tabel 4. Login

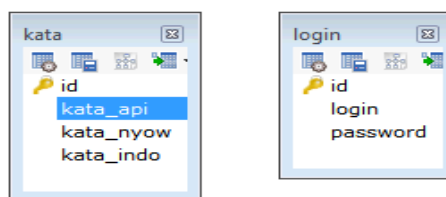
Field	Tipe	Null	Key	Ekstra
id	bigint (20)		PRI	auto_increment
login	varchar (20)			
password	varchar (50)			
Role	Enum('1','2','3')			

b. Tabel 5. Kata

Field	Tipe	Null	Key	Ekstra
Id	bigint (20)		PRI	auto_increment
kata_indo	varchar (255)			
kata_api	varchar (255)			
kata_nyow	varchar (255)			

Relasi antar tabel dalam basis data Sistem Informasi Penerjemah *Online*

Bahasa Lampung adalah sebagai berikut :



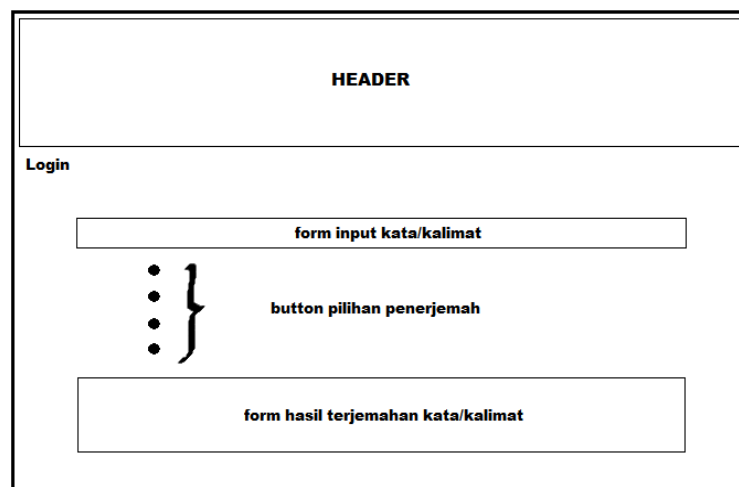
Gambar 32. Relasi antar table

Terlihat antar table tidak memiliki relasi secara langsung karena tidak adanya *foreign key* (kunci tamu) yaitu satu atau beberapa kolom pada table yang merupakan *primary key* (kunci utama) pada table satu namun diletakan pada table dimana tablenya berelasi dengan table dirinya. Pada table diatas terlihat

bahwa tiap table hanya berfungsi sebagai autentifikasi saja, Pada table *login* terdapat *field* “role” yang berfungsi untuk membedakan antara *user* admin dan *user* biasa, dalam hal ini *user* admin pada menu input kata bisa melakukan aksi tambah, edit, dan hapus, sedangkan untuk *user* biasa hanya bias tambah kata saja. Dalam penjelasan diatas relasi antar table yaitu bisa dikatakan derajat relasi dinyatakan dengan *one to many*, yaitu satu *user* bias memasukkan banyak kata.

f. Desain Antar Muka

Sesuai *user requirement* bahwa *interface* sederhana, mudah dipahami dan tidak membingungkan maka dirancanglah antarmuka web ini sebagai berikut



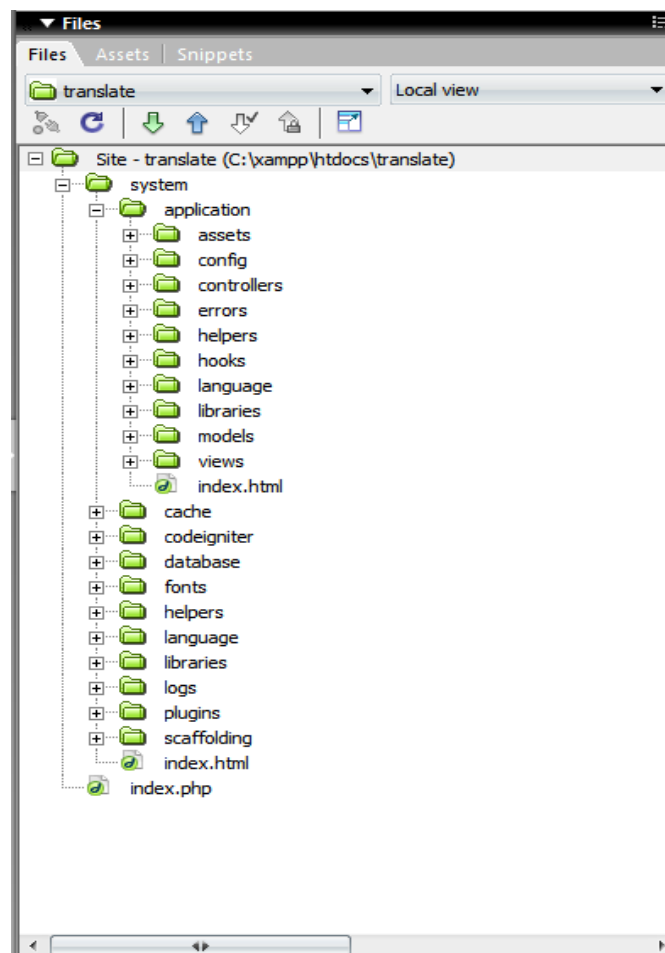
Gambar 33. Rancangan antarmuka *user*

3. Penulisan Program

Pembuatan aplikasi penerjemah *online* ini menggunakan bahasa pemrograman PHP dengan didukung oleh *Framework* Code Igniter. *Framework* ini sudah

banyak digunakan oleh *developer* karena memberikan kemudahan dalam membangun sebuah sistem berbasis *web* menggunakan PHP, di antaranya *developer* dapat menggunakan konsep *model view controller*. Entitas yang sudah diketahui dibuat *model* dan *controller* nya. Lalu *user* berinteraksi melalui *view*, di mana di dalam *view* ada fungsi yang memanggil *controller* untuk me-load data dari *model*. Kemudian hasilnya diberikan kembali ke *view* sehingga *user* dapat melihat hasilnya.

Berikut adalah struktur dari Code Igniter :



Gambar 34. Struktur Code Igniter

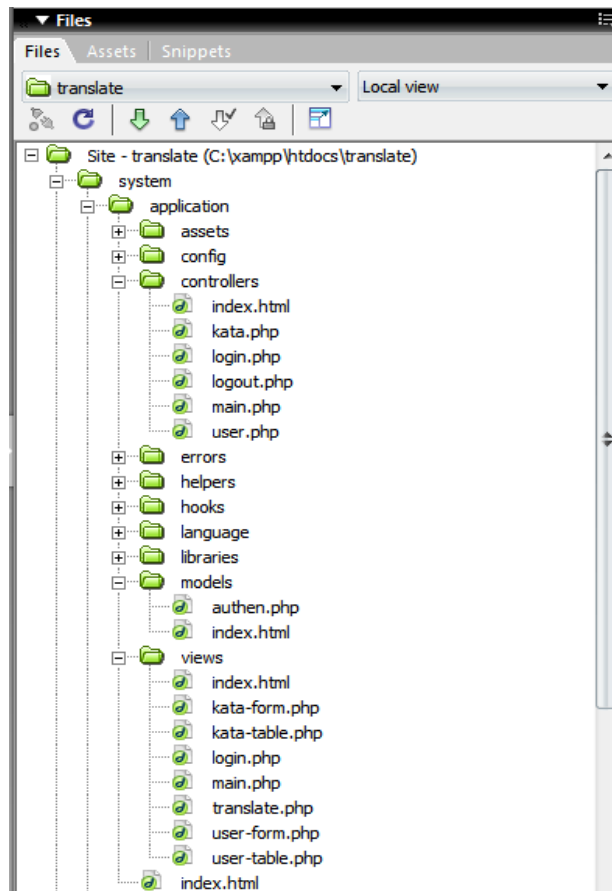
Keuntungan yang lain adalah konfigurasi koneksi *database* yang cukup sekali dilakukan di *folder* `/system/application/config/database.php`, ada lima poin yang penting untuk setting pada konfigurasi *database*, yaitu *hostname*, *username*, *password*, *database* (nama *database* pada tugas akhir ini nama *database*-nya adalah *translate*) dan *dbdriver*. Konfigurasinya dapat dilihat pada gambar berikut ini

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

$db['default']['hostname'] = "localhost";
$db['default']['username'] = "root";
$db['default']['password'] = "";
$db['default']['database'] = "translate";
$db['default']['dbdriver'] = "mysql";
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = "utf8";
$db['default']['dbcollat'] = "utf8_general_ci";
```

Gambar 35. Konfigurasi *database*

Berikut adalah direktori *folder* untuk *controller* dan *view*. Pada pembuatan aplikasi ini, model dan *controller* dibuat menjadi satu di *folder controller*. Hal ini untuk memudahkan pengerjaan dengan tidak terlalu banyak memisahkan antara model dan *controller*.



Gambar 36. Direktori *controller* dan *view*

Untuk kode program dapat dilihat pada lampiran.

4. Pengujian

Pengujian dilakukan untuk membuktikan kesesuaian aplikasi dengan spesifikasi sistem pada bagian analisis, perancangan dan implementasi. Melalui pengujian ini juga akan dapat diketahui apakah sistem telah memenuhi spesifikasi yang dibutuhkan.

Metode yang digunakan adalah metode struktural. Pengujian yang dilakukan bertujuan untuk memastikan sistem bekerja sesuai dengan kode yang diimplementasikan.

Pengujian dimulai dengan menjalankan XAMPP *web server*, MySQL *database server*, dan *browser* Mozilla Firefox.

Pengujian dilakukan terhadap sistem pertama kali dilakukan dengan mengakses halaman indeks dengan alamat <http://localhost/translate/>, akan muncul halaman sebagai berikut :

The screenshot shows the initial interface of the online translation system. It features a blue header with the title "Sistem Informasi Penerjemah Online Bahasa Lampung" and a "Login" link. The main content area is titled "Translate" and contains an "Input kata" text box, a "Pilihan" section with radio buttons for selecting the source and target dialects, and a "Hasil" text box. Below the form are "Translate" and "Reset" buttons. The footer bar contains the copyright notice "© 2010 By Noval".

Gambar 37. Halaman awal Sistem Informasi Penerjemah *Online* Bahasa Lampung

a. Pengujian Menu *User*

User memasukkan kata atau kalimat yang ingin diterjemahkan. Kemudian memilih tombol navigasi untuk arah penerjemahan diantaranya;

1. Indonesia – Dialek Api
2. Indonesia – Dialek Nyow
3. Dialek Api – Indonesia
4. Dialek Api – Dialek Nyow
5. Dialek Nyow – Indonesia
6. Dialek Nyow – Dialek Api

Kemudian pilih tombol *translate* dan hasilnya akan ditampilkan di *text area*.

Berikut adalah hasil pengujiannya:

The screenshot displays the user interface of an online translation system. At the top, a blue banner reads 'Sistem Informasi Penerjemah Online Bahasa Lampung'. Below this, the 'Input Kata' section contains a text box with the word 'mau'. A 'Pilihan' (Selection) section follows, featuring six radio button options: 'Indonesia - Dialek Api' (selected), 'Indonesia - Dialek Nyow', 'Dialek Api - Indonesia', 'Dialek Nyow - Indonesia', 'Dialek Api - Dialek Nyow', and 'Dialek Nyow- Dialek Api'. The 'Hasil' (Result) section shows the translated word 'haga' in a text box. At the bottom of the form are two buttons: 'Translate' and 'Reset'. The footer of the page contains the copyright notice '© 2010 By Noval'.

Gambar 38. Pengujian Indonesia – Dialek Api

Sistem Informasi Penerjemah Online Bahasa Lampung

Input Kata

Translate

Input kata : mau

Pilihan : Indonesia - Dialek Api
 Indonesia - Dialek Nyow
 Dialek Api - Indonesia
 Dialek Nyow - Indonesia
 Dialek Api - Dialek Nyow
 Dialek Nyow- Dialek Api

Hasil : hagow

© 2010 By Noval

Gambar 39. Pengujian Indonesia – Dialek Nyow

b. Pengujian Menu Admin

Admin harus *login* terlebih dahulu. Jika *login* dan *password* salah maka akan tampil pesan *error*. Ini dikarena pada login yang dimasukan akan di cek di *database login* dengan sebuah perintah "SELECT *FROM login WHERE login='\$login' AND password=Password('\$password')", apakah *login* sesuai atau tidak, bila sesuai maka akan muncul halaman selanjutnya dan bila tidak akan muncul pesan *error* 'Login/Password anda salah!'. Berikut ini adalah tampilan menu *login*.

Sistem Informasi Penerjemah Online Bahasa Lampung

Translate

Login

Login :

Pasword :

© 2010 By Noval

Gambar 40. Halaman *Login*

Pada halaman awal admin ditampilkan tabel daftar kata yang terdiri dari kata Indonesia, kata dialek api, dialek nyow dan kolom aksi. Penambahan, penghapusan dan pengeditan kata akan dijelaskan di gambar berikutnya.

Sistem Informasi Penerjemah Online Bahasa Lampung

Translate | User | Logout

Daftar Kata

No.	Indonesia	Dialek Api	Dialek Nyow	Aksi	
1	mau	haga	hagow	<input type="button" value="Edit"/>	<input type="button" value="Hapus"/>
2	saya	nyak/ikam	nyak	<input type="button" value="Edit"/>	<input type="button" value="Hapus"/>
3	tidur	pedom	pedom	<input type="button" value="Edit"/>	<input type="button" value="Hapus"/>

© 2010 By Noval

Gambar 41. Halaman awal *administrator*

1. Pengujian tambah kata

Untuk menambah kata, pada kolom aksi pilih tambah maka akan ditampilkan form *input* kata. Masing-masing kata harus di-*input*, jika salah satu field kata tidak diisi maka akan ditampilkan pesan *error*.



The screenshot shows a web application interface for a Lampung language translation system. At the top, there is a blue header with the text 'Sistem Informasi Penerjemah Online Bahasa Lampung'. Below the header, there is a navigation bar with links for 'Translate', 'User', and 'Logout'. The main content area is titled 'Input Kata' and contains three input fields labeled 'Indonesia', 'Dialek Api', and 'Dialek Nyow'. Below these fields are three buttons: 'Simpan', 'Reset', and 'Batal'. At the bottom of the page, there is a footer with the text '© 2010 By Noval'.

Gambar 42. *Form* tambah kata

Pada menu penambahan kata bila ketika menginput kata ternyata kata tersebut telah ada dalam *database* maka akan muncul pesan “data sudah ada silahkan cek isian anda” pesan ini menunjukkan bahwa kata yang di input telah ada dalam *database* sebelumnya. Bila salah satu kolom tidak diisi maka sistem akan ada pesan sesuai data mana yang belum terisi, pesan tersebut yaitu, 'Kata Indonesia tidak boleh kosong!', 'Kata Dialek Api tidak boleh kosong!', dan 'Kata Dialek Nyow tidak boleh kosong!'

2. Pengujian ubah kata

Pengubahan kata dilakukan dengan memilih aksi ubah pada daftar kata halaman awal admin dan akan ditampilkan form ubah kata dimana masing-masing field akan muncul kata yang akan diubah.

Sistem Informasi Penerjemah Online Bahasa Lampung

Translate | User | Logout

Kelola Kata

Indonesia :

Dialek Api :

Dialek Nyow :

© 2010 By Noval

Gambar 43. *Form* ubah kata

3. Pengujian tambah *user*

Administrator bisa menambah *user* yang berhak mengakses sistem seperti menambahkan kata dalam *database*.. Berikut tampilan tabel *user*.

Sistem Informasi Penerjemah Online Bahasa Lampung

Translate | Input Kata | Logout

Daftar User

Login	Aksi
admin	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

© 2010 By Noval

Gambar 44. Tabel *user*

Untuk menambah *user* dilakukan dengan memilih aksi tambah dan akan dilanjutkan ke *form input user*. Pada form ini disediakan *field login*, *password*, dan konfirmasi *password*. Bila penambahan *user* ternyata *login* tersebut sudah terdapat di database maka timbul pesan 'data sudah ada silahkan cek isian anda'. Berikut ini adalah tampilan menu tambah *user*.



The screenshot shows the 'Input User' form within a web application titled 'Sistem Informasi Penerjemah Online Bahasa Lampung'. The navigation bar includes 'Translate | Input Kata | Logout'. The form contains three input fields: 'Login', 'Password', and 'Konfirmasi'. Below the fields are three buttons: 'Simpan', 'Reset', and 'Batal'. A copyright notice '© 2010 By Noval' is visible at the bottom of the page.

Gambar 45. *Form tambah user*

4. Pengujian ubah *user*



The screenshot shows the 'Input User' form within the same web application. The 'Login' field is now filled with the text 'admin'. The 'Password' and 'Konfirmasi' fields are empty. The 'Simpan', 'Reset', and 'Batal' buttons remain. The copyright notice '© 2010 By Noval' is still present at the bottom.

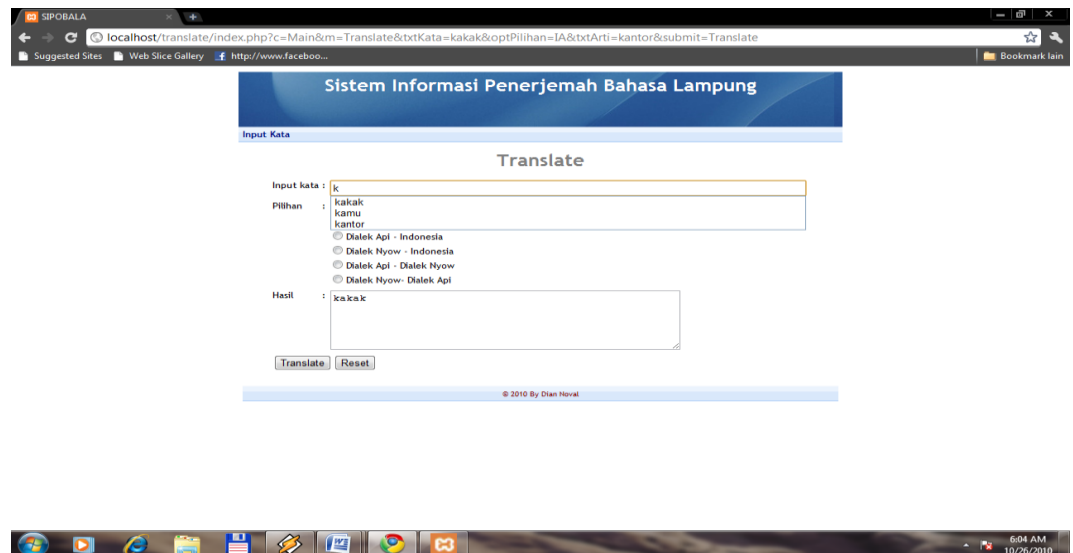
Gambar 46. *Form ubah user*

Pengujian ubah user dengan memilih aksi ubah dan akan dilanjutkan ke form ubah user. Pada form ini *field login* sudah terisi dan *field password* kosong untuk memasukkan *password* baru.

c. Pengujian Sistem dengan *Browser*.

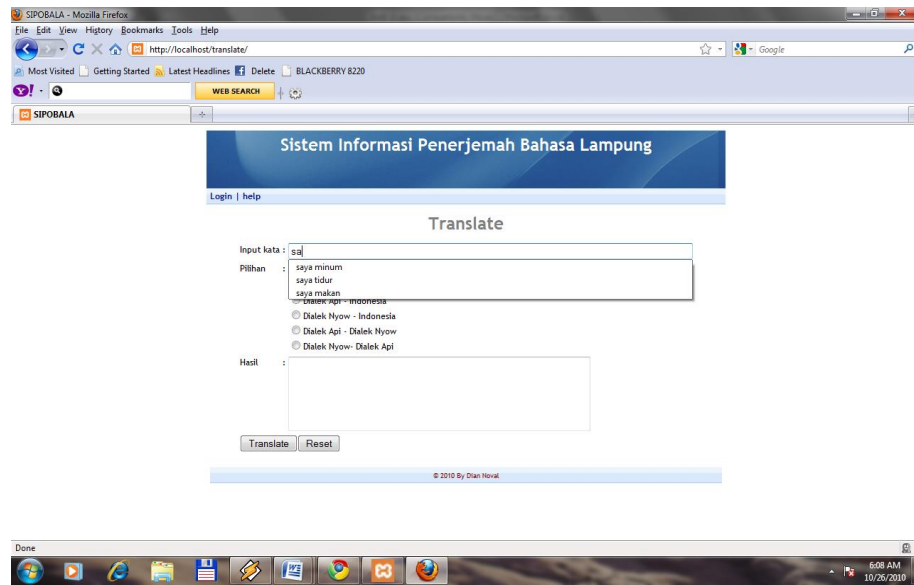
Pada pengujian sistem ini di uji dengan tiga buah *browser* yaitu Google Chrome, Mozilla Firefox dan dan Opera dari ketiga *browser* yang di gunakan ternyata ketika di uji dengan *browser* Opera sistem tidak berkerja secara optimal dan stabil, terlihat dari tampilan menu yang tidak teratur dan fungsi session yang tidak muncul. Berikut ini ada tampilan sistem dari beberapa *browser*.

1. Tampilan sistem pada Google Chrome



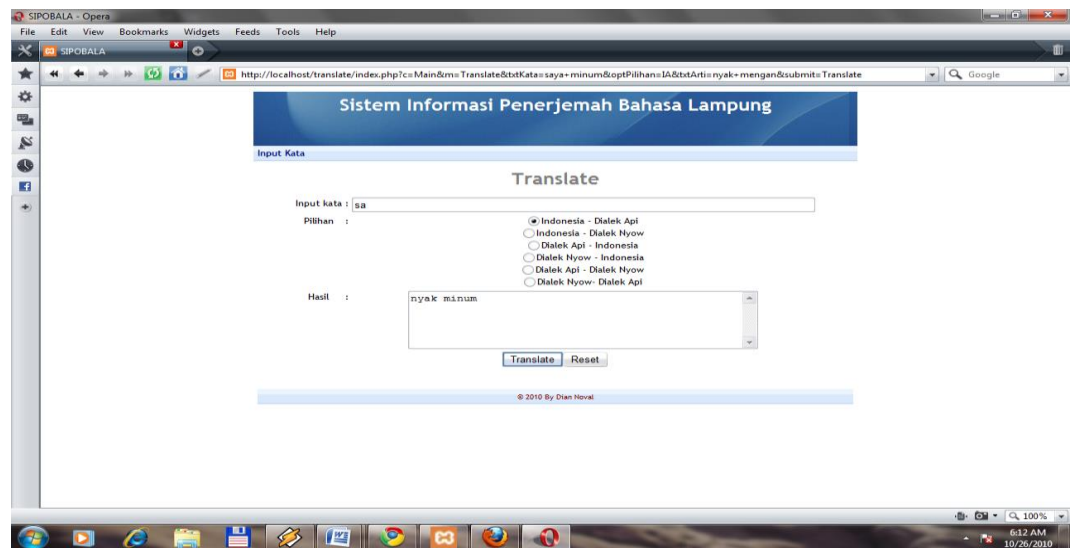
Gambar 47. Sistem di *browser* Google Chrome

2. Tampilan sistem pada *browser* Mozilla Firefox



Gambar 48. Sistem pada Mozilla Firefox

3. Tampilan sistem pada *browser* Opera



Gambar 49. Sistem pada Opera

d. Pengujian Menterjemahkan Kata

Sistem Informasi Penerjemah Online Bahasa Lampung

Input Kata

Translate

Input kata :

Pilihan : Indonesia - Dialek Api
 Indonesia - Dialek Nyow
 Dialek Api - Indonesia
 Dialek Nyow - Indonesia
 Dialek Api - Dialek Nyow
 Dialek Nyow - Dialek Api

Hasil : nyak mengan nasi lalu pedom

© 2010 By Noval

Gambar 50. Sitem Penerjemah

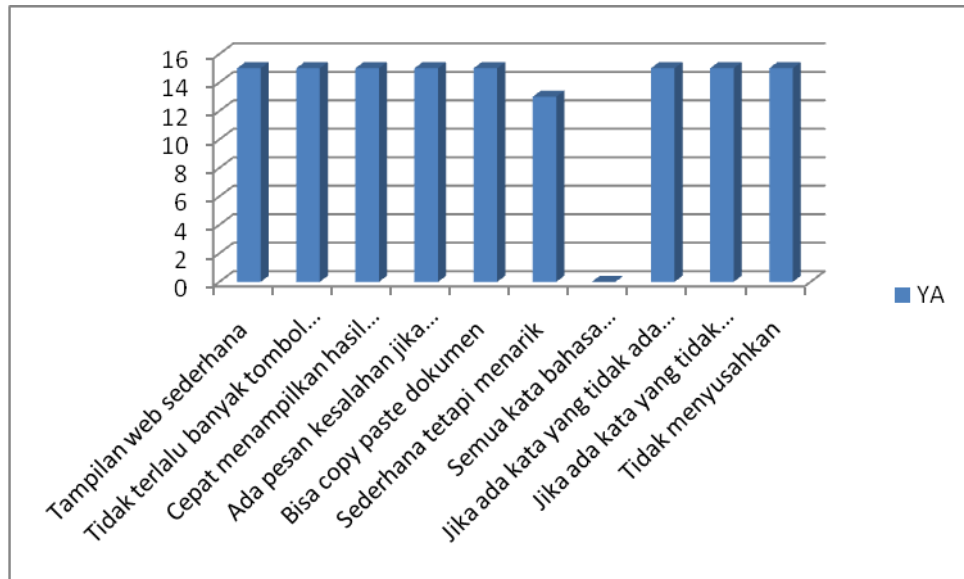
Terlihat ketika menerjemahkan kata, bila kata tersebut tidak terdapat di dalam *database* maka data pada hasil terjemahan akan muncul kata yang sama seperti yang di input dengan tampilan berwarna merah ini dikarena pada di listening program di masukkan perintah `$ret = '' . $key . '';` yang memerintahkan bila kata tersebut tidak terdapat di *database* maka di kolom hasil akan muncul kata dengan warna merah. Pada pengujian ini maksimal kata yang bias di terjemahkan sebanyak 1082 kata maka web akan *error* karena terlalu banyak data yang di *load*.

1. Hasil Pengujian *user requirement*

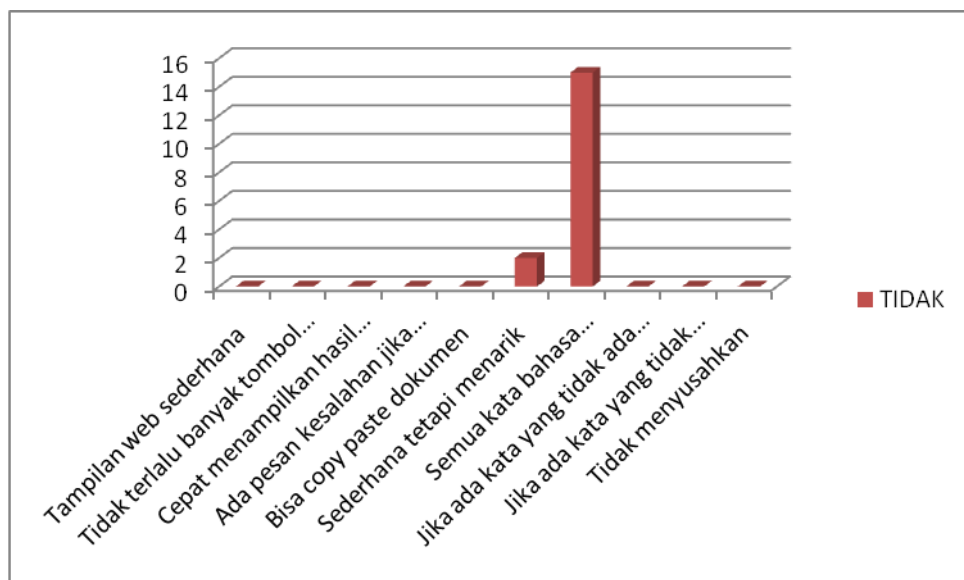
Pengujian ini dilakukan oleh 15 *user* yang kemudian setelah melakukan pengujian sistem ini, para *user* mengisi kuesioner yang telah dibuat sesuai dengan *user requirement* sebelum sistem di buat, pengujian dilakukan dengan menggunakan localhost. Untuk hasilnya dapat di lihat pada table berikut:

Tabel 6. Hasil Pengujian Sistem oleh *User*.

NO	Pertanyaan / Pengujian	Jumlah <i>user</i>	
		Ya	Tidak
1.	Tampilan web sederhana	15	0
2.	Tidak terlalu banyak tombol navigasi	15	0
3.	Cepat menampilkan hasil terjemahan	15	0
4.	Ada pesan kesalahan jika terjadi <i>error</i>	15	0
5.	Bisa <i>copy paste</i> dokumen	15	0
6.	Sederhana tetapi menarik	13	2
7.	Semua kata bahasa Lampung dapat ditemukan saat dilakukan <i>translate</i>	0	15
8.	Jika ada kata yang tidak ada di <i>database</i> diterjemahkan maka ditampilkan kata tersebut	15	0
9.	Jika ada kata yang tidak diterjemahkan berubah warna jadi merah atau biru sehingga bisa langsung ketahuan	15	0
10.	Tidak menyusahkan	15	0



Gambar 51. Diagram pengujian *user* yang mengatakan “ya”



Gambar 52. Diagram pengujian *user* yang mengatakan “Tidak”

Dari diagram 51 dan 52 diatas terlihat bahwa hampir semua pertanyaan yang sesuai dengan permintaan *user* pada tahap sebelumnya, *user* mengatakan “ya” yang menandakan sistem ini telah memenuhi kebutuhan *user*, dari diagram diatas pada pertanyaan “semua kata bahasa lampung dapat di translate”, semua *user* menjawab tidak ini di karenakan *database* yang dimiliki kurang lengkap dan

membutuhkan *database* yang lebih banyak lagi agar dalam penerjemahan kata, kata yang dicari ada dalam *database* dan kata tersebut dapat di tampilkan sesuai dengan permintaan *user*, sedangkan untuk pertanyaan yang lain rata-rata semua menjawab “iya”, dari hasil pengujian ini dapat di simpulkan bahwa sistem yang di rancang hampir memunihi kebutuhan *user*.

2. Hasil Pengujian *Online*.

a. Pengujian 1 *user*

Pada pengujian ini menggunakan koneksi tekonselflash, ketika *user* membuka web membutuhkan waktu kurang lebih 3 detik sampai web memunculkan semua content, selain itu juga dilakukan pengujian penerjemahan kata pada sistem, setelah pengujian dilakukan ternyata sistem membutuhkan waktu kurang lebih 5 detik untuk menampilkan hasilnya.

b. Pengujian dengan 5 *User*.

Pada pengujian ini dilakukan penghitungan waktu yang di butuhkan untuk membuka website dengan menggunakan koneksi internet yang berbeda-beda, kemudian mengukur kecepatan menerjemahkan kalimat pada waktu bersamaan, untuk hasilnya dapat di lihat pada table berikut

Table 7. Hasil waktu membuka web

No	Jenis Koneksi	Kecepatan Membuka web
1	TelkonselFlash	3 detik
2	Flexy	12 detik
3	Smart	4 detik
4	TelkonselFlash	3 detik
5	Speedy	2 detik

Tabel 8. Hasil waktu penerjemahan

No	Jenis Koneksi	Kecepatan menterjemah kata
1	TelkomselFlash	2 detik
2	Flexy	3 detik
3	Smart	2 detik
4	TelkomselFlash	3 detik
5	Speedy	2 detik

Pada pengujian ini di gunakan kata sebanyak 254 kata. Pada tabel dapat terlihat bahwa *website translate* ini bisa dibuka dengan cepat dalam hitungan detik. Hal ini dikarenakan *content* dari *website translate* sangat sederhana, tidak ada fitur yang memberatkan sehingga tidak membutuhkan waktu lama bagi *server* untuk *me-load content website*. Waktu *loading* juga di pengaruhi oleh koneksi internet yang dipakai. Pada pengujian menerjemahkan kata secara bersamaan, ternyata waktu yang di butuhkan untuk menerjemahkan kata ketika di pakai satu *user* dan lima *user* tidak begitu berbeda jauh bahkan hampir sama ketika di pakai dengan satu *user* yaitu sekitar 2-3 detik untuk setiap koneksi dari setiap *user*.

5. Penerapan Program

Penerapan program di lakukan dengan menerapkan sistem yang telah selesai dibuat dan diuji ke dalam lingkungan internet sehingga dapat di gunakan oleh pengguna yang membutuhkan informasi, sistem ini dapet di akses di www.kobala.web.id.