BAB 3

ANALISIS DAN PERANCANGAN

3. 1 Analisis Sistem

Untuk memulai membangun suatu program aplikasi berupa aplikasi mengenai kamus, terlebih dahulu direncanakan alur kerja berdasarkan kebutuhan dari user yang akan menggunakan program aplikasi kamus dan aksara karo ini.

3. 1.1 Analis *Input*

Sebuah sistem harus memiliki masukan agar menghasilkan sesuatu. Masukan (*input*) dari *user* bagi aplikasi ini adalah karakter (huruf, angka, tanda baca dan operator) dalam tulisan latin. Untuk lebih jelasnya pembahasan masukkan dalam aplikasi ini adalah sebagai berikut.

Adapun karakter masukan dalam text tulisan latin tersebut adalah sebagai berikut:

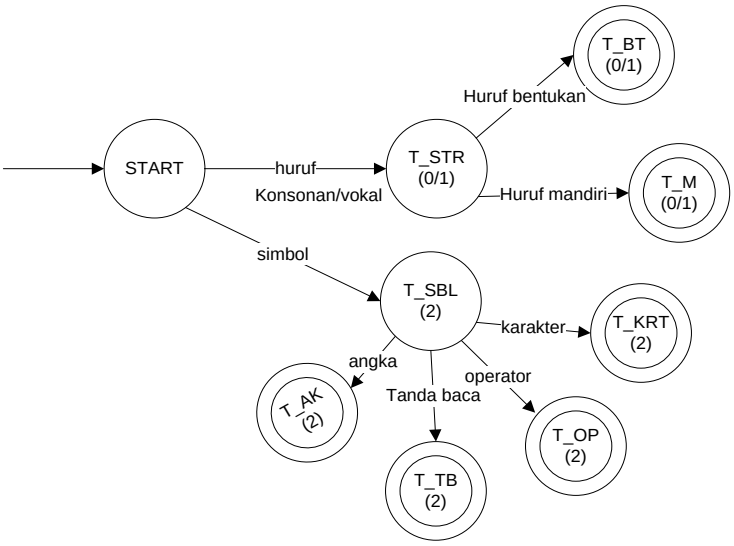
1. *Input* dalam aplikasi ini adalah huruf/ karakter A..Z, a..z, &, @, #, ~, %, $, _, \, |, (,{, [,], },).
2. *Input* angka dalam aplikasi ini adalah 0..9
3. *Input* tanda baca dalam aplikasi ini adalah ., ,, ?, !, ‘, “, ;, :.
4. *Input* operator dalam aplikasi ini adalah +, - , *, / , ^, =, <, >, ≤ , ≥.

3. 1.2 Analisis Leksikal

Analisis leksikal adalah mengidentifikasi semua besaran yang membangun suatu bahasan pada suatu sumber dalam analisi leksikal ini. *Input* teks yang dimasukkan adalah teks dalam tulisan latin seperti dijelaskan pada analisis *input*. Hal ini dimaksudkan agar sistem dapat mengenali atau membaca *input* teks tulisan.

Input fungsi teks tulisan latin memiliki himpunan token seperti t_STR,t_SBL, t_OP, t_AK, t_TB, t_SK. t_STR mengandung t_BT, t_M. t_BT mengandung semua huruf dalam teks tulisan latin A..Z, a..z. yang menghasilkan bunyi bentukan. , t_M mengandung semua huruf dalam teks tulisan latin A..Z, a..z. yang menghasilkan bunyi mandiri. t_SBL mengandung t_OP, t_TB, t_AK, t_KRT. t_OP mengandung semua operator +, - , *, / , ^, =, <, >, ≤ , ≥, t_AK mengandung semua angka 0..9, t_TB mengandung semua tanda baca dalam teks tulisan latin, t_KRT mengandung semua karakter &, @, #, ~, %, $, _, \, |, (,{, [,], },). Perlu ditekankan bahwa operator bukan untuk operasi perhitungan atau logika, melainkan dianggap sebagai karakter *input* saja.

State diagram atau diagram keadaan dari prosedur teks tulisan latin dapat dilihat pada Gambar 3.1 berikut.



Gambar 3. 1 State Diagram *Input* Teks

3. 1.3 Analisis Sintaksis

Analisis sintaksis merupakan kelanjutan dari analisis leksikal yang mengubah prosedur scan ke dalam bentuk token beserta mengelola token-token. Tahap ini di sebut juga dengan istilah *parsing*.

3. 1.6.1 Parsing

Sistem yang dibangun menggunakan metode *leftmost derivation* untuk menurunkan *input* tulisan teks. Berdasarkan metode *leftmost derivation*, *input* teks akan diperiksa dan diturunkan dari sebelah kiri ke sebelah kanan, *input* teks oleh pengguna harus sesuai dengan *input* teks tulisan latin dan analisis *input* teks tulisan latin.

Secara umum cara kerja algoritma untuk melakukan penurunan teks *input* adalah sebagai berikut :

1. Periksa apakah karakter pertama dalam *input* teks berupa huruf, angka, operator, tanda baca, atau karakter.
2. Jika angka, maka sistem akan mengenalinya sebagai angka dan akan menghasilkan token angka (t_AK), dan angka yang di *input*kan akan dicek ke dalam *array angka* sampai angka tersebut ditemukan. Jika angka dalam *array* sama dengan angka yang di*input*kan maka ganti angka dengan angka dalam aksara karo kemudian tampilkan angka dalam aksara Karo.
3. Jika tanda baca, maka sistem akan mengenalinya sebagai tanda baca dan akan menghasilkan t_TB. Dan tanda baca yang di*input*kan akan dicek ke dalam *array tandabaca* sampai tanda baca tersebut ditemukan. Jika tanda baca dalam *array* sama dengan tanda baca yang di*input*kan maka ganti tanda baca dengan tanda baca dalam aksara karo kemudian tampilkan tanda baca dalam aksara Karo.
4. Jika operator, maka sistem akan mengenalinya sebagai operator dan akan menghasilkan t_OP. Dan operator yang di *input* akan dicek ke dalam *array* operator sampai operator tersebut ditemukan. Jika operator dalam *array* sama dengan angka yang di *input* maka ganti operator dengan operator dalam aksara karo kemudian tampilkan operator dalam aksara Karo.
5. Jika karakter, maka sistem akan mengenalinya sebagai karakter dan akan menghasilkan t_KRT . Dan karakter yang di*input*kan akan dicek ke dalam *array karakter* sampai karakter tersebut ditemukan. Jika karakter dalam *array*

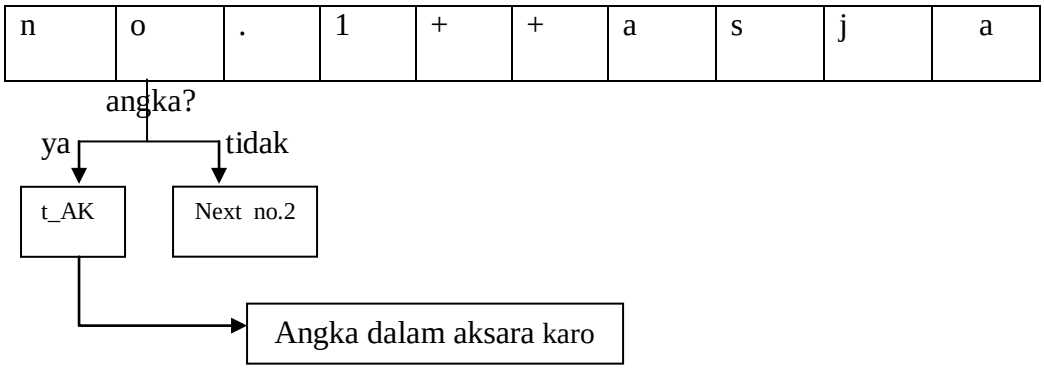
sama dengan karakter yang di*input*kan maka ganti karakter dengan karakter dalam aksara karo kemudian tampilkan karakter dalam aksara Karo.

6. Jika huruf A-Z atau a-z, maka sistem akan mengenalinya sebagai huruf dan menghasilkan token t_STR. Kemudian memeriksa karakter selanjutnya apakah membentuk suatu bunyi bentukan atau tidak?, jika huruf-huruf tersebut membentuk suatu bunyi bentukan, maka akan menghasilkan token t_BT. Dan bunyi bentukan yang di*input*kan akan dicek ke dalam *array dua* sampai bunyi bentukan tersebut ditemukan. Jika bunyi bentukan dalam *array* sama dengan bunyi bentukan yang di*input*kan maka ganti bunyi bentukan dengan bunyi bentukan dalam aksara karo kemudian tampilkan bunyi bentukan dalam aksara Karo. Jika tidak membentuk bunyi bentukan maka akan membentuk bunyi mandiri dan akan menghasilkan token t_M. Dan bunyi mandiri yang diinpukant akan dicek ke dalam *array satu* sampai bunyi mandiri tersebut ditemukan. Jika bunyi mandiri dalam *array* sama dengan bunyi mandiri yang di*input*kan maka ganti bunyi mandiri dengan bunyi mandiri dalam aksara karo kemudian tampilkan bunyi mandiri dalam aksara Karo.
7. Ulangi dari langkah ke dua untuk karakter-karakter selanjutnya smpai karakter terakhir
8. Konversikan setiap elemen di array dua menjadi aksara Karo.

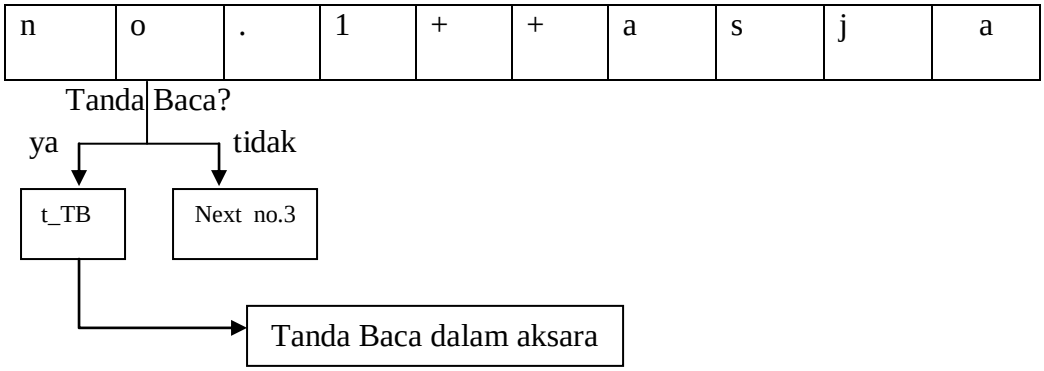
Agar lebih jelas coba perhatikan langkah-langkah berikut.

Teks	**No.1++asja**									
A	n	o	.	1	+	+	a	s	j	a

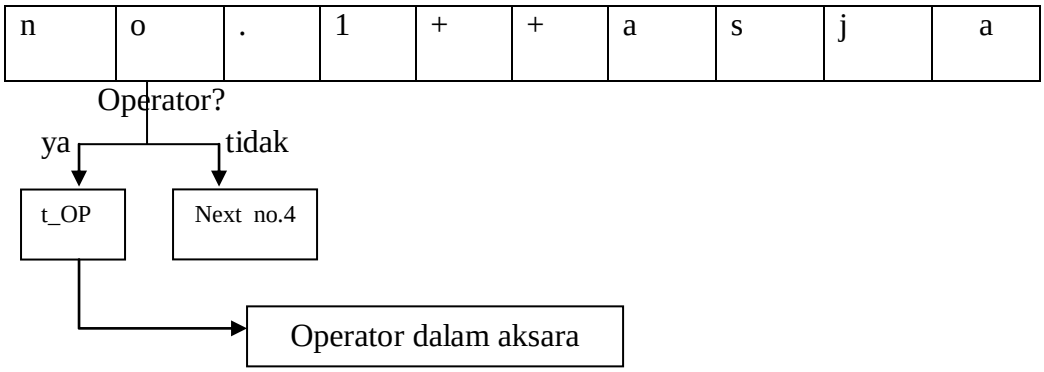
1. Periksa karakter pertama, apakah karakter pertama termasuk angka? Jika ya, maka sistem akan mengenalinya sebagai angka dan menghasilkan token t_AK. dan angka yang di*input*kan akan dicek ke dalam *array angka* sampai angka tersebut ditemukan. Jika angka dalam *array* sama dengan angka yang di*input*kan maka ganti angka dengan angka dalam aksara karo kemudian tampilkan angka dalam aksara Karo. Jika tidak sistem akan melanjutkan untuk memeriksa langkah selanjutnya.



2. Jika tanda baca, maka sistem akan mengenalinya sebagai tanda baca dan menghasilkan token t_TB. Dan tanda baca yang di*input*kan akan dicek ke dalam *array* tandabaca sampai tanda baca tersebut ditemukan. Jika tanda baca dalam *array* sama dengan tanda baca yang di*input*kan maka ganti tanda baca dengan tanda baca dalam aksara karo kemudian tampilkan tanda baca dalam aksara Karo. jika tidak sistem akan melanjutkan untuk memeriksa langkah selanjutnya.

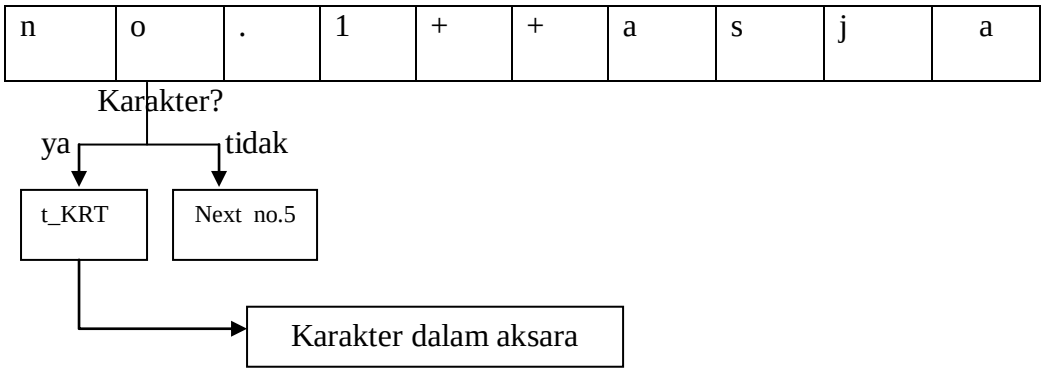


3. Jika operator, maka sistem akan mengenalinya sebagai operator dan menghasilkan token t_OP. Dan operator yang di*input*kan akan dicek ke dalam *array* operator sampai operator tersebut ditemukan. Jika operator dalam *array* sama dengan angka yang di*input*kan maka ganti operator dengan operator dalam aksara karo kemudian tampilkan operator dalam aksara Karo. jika tidak sistem akan melanjutkan untuk memeriksa langkah selanjutnya.



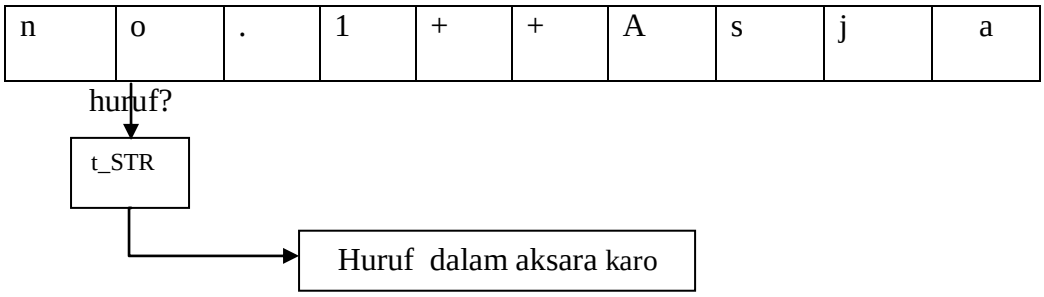
4. Jika karakter, maka sistem akan mengenalinya sebagai karakter dan menghasilkan token t_KRT. Dan karakter yang di*input*kan akan dicek ke dalam *array karakter* sampai karakter tersebut ditemukan. Jika karakter dalam *array* sama dengan karakter yang di*input*kan maka ganti karakter dengan

karakter dalam aksara karo kemudian tampilkan karakter dalam aksara Karo. jika tidak sistem akan melanjutkan untuk memeriksa langkah selanjutnya.

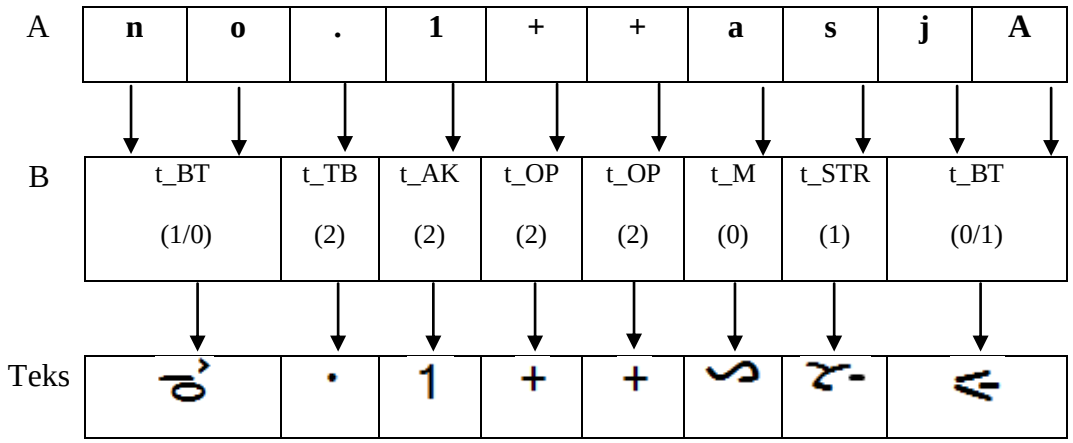


5. Dan terakhir pasti huruf, kemudian sistem akan mengenalinya sebagai huruf dan menghasilkan token t_STR. Jika termasuk dalam token t_STR maka sistem akan memeriksa apakan karakter tersebut membentuk suatu bunyi bentukan dengan karakter selanjutnya?, jika ia sistem akan menghasilkan t_BT. Dan bunyi bentukan yang di*input*kan akan dicek ke dalam *array dua* sampai bunyi bentukan tersebut ditemukan. Jika bunyi bentukan dalam *array* sama dengan bunyi bentukan yang di*input*kan maka ganti bunyi bentukan dengan bunyi bentukan dalam aksara karo kemudian tampilkan bunyi bentukan dalam aksara Karo. Jika tidak membentuk bunyi bentukan maka akan membentuk bunyi mandiri dan akan menghasilkan token t_M. Dan bunyi mandiri yang di*input*kan akan dicek ke dalam *array satu* sampai bunyi mandiri tersebut ditemukan. Jika bunyi mandiri dalam *array* sama dengan bunyi mandiri yang di*input*kan maka ganti bunyi mandiri dengan bunyi

mandiri dalam aksara karo kemudian tampilkan bunyi mandiri dalam aksara Karo.

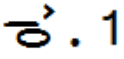
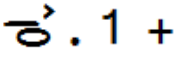
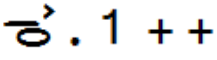
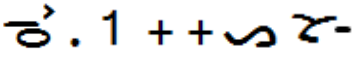
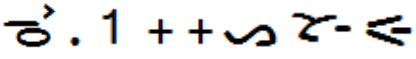


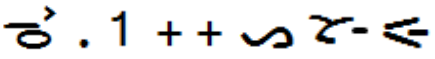
Dan langkah tersebut diulangi sampai semua karakter tereksekusi. Untuk lebih jelas coba perhatikan contoh berikut ini, misalnya akan menurunkan teks tulisan latin **no.1++asja** maka langkah penurunan teks tersebut adalah sebagai berikut.



3. 1.4 Analisis Output

Pada analisis output akan dijelaskan mengenai hasil keluaran yang diterima user. Keluaran yang dihasilkan oleh proses penurunan teks tulisan latin adalah aksara karo yang memiliki arti sesuai dengan *input* dari teks tulisan latin tersebut.

1. teks = 
2. teks = 
3. teks = 
4. teks = 
5. teks = 
6. teks = 
7. teks = 

Sehingga *output* yang diterima user adalah 

3. 1.5 Analisis Non Fungsional

Dalam analisis non fungsional, penulis membagi kedalam tiga bahasan, yaitu analisis pengguna, analisis perangkat keras dan analisis perangkat lunak.

3. 1.5.1 Analisis Pengguna

Pengguna dari aplikasi ini adalah Admin dan masyarakat umum yang terbiasa dengan jaringan internet dan aplikasi berbasis website, karena aplikasi ini menggunakan beberapa form sesuai dengan proses-proses yang dapat dilakukan pengguna pada setiap form tersebut

Admin adalah bagian pengguna aplikasi yang dapat menggunakan akses lebih banyak daripada para pengguna pada aplikasi yang akan dibangun. Admin pada aplikasi yang akan dibangun ini merupakan pengelola program aplikasi ini.

3. 1.5.2 Analisis Perangkat Keras

Analisis perangkat keras adalah analisis mengenai perangkat keras yang digunakan untuk mendukung aplikasi kamus dan aksara Karo ini. Kebutuhan sistem yang dibangun adalah spesifikasi perangkat keras yang dibutuhkan untuk membangun aplikasi kamus dan aksara karo. Spesifikasi perangkat keras tersebut adalah sebagai berikut.

Processor	Intel Pentium III
Monitor	CRT Standar
Hardisk	10 GB
Memory	128 MB
CD ROM	
Keybord + Mouse	

3. 1.5.3 Analisis Perangkat Lunak

Untuk membangun aplikasi kamus dan aksara karo penulis menggunakan spesifikasi perangkat lunak sebagai berikut.

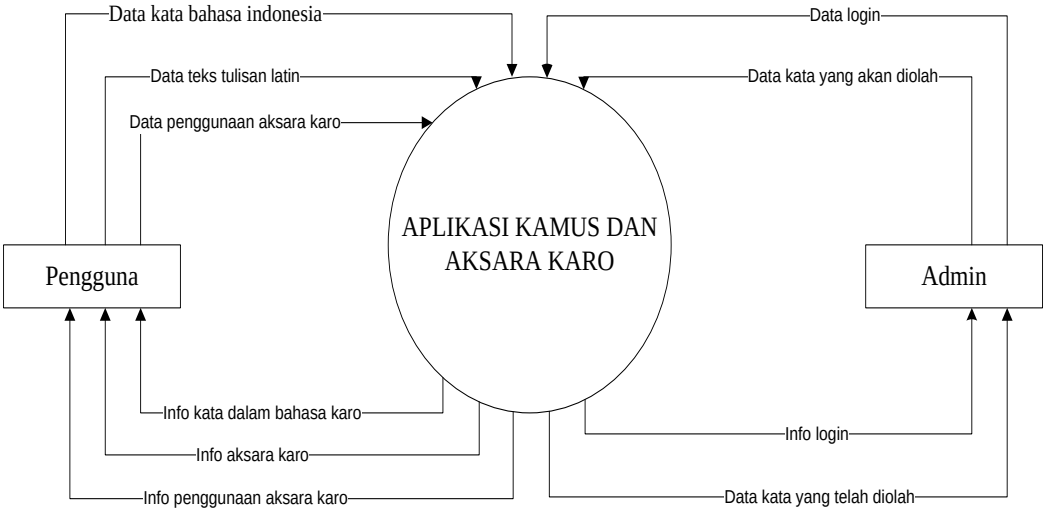
1. Sistem operasi adalah windows 7
2. MySQL sebagai basis data
3. Macromedia Dreamweaver sebagai tool pembangunan aplikasi
4. Bahasa pemrograman adalah mengunakan bahasa pemrograman PHP
5. *Browser* seperti Mozilla Firefox,opera, google chrome.

3. 1.6 Analisis Kebutuhan Fungsional

Analisis fungsional merupakan analisis yang digunakan untuk menggambarkan fungsi dari aplikasi penggambaran proses yang terjadi adalah sebagai berikut.

3. 1.6.1 Diagram Konteks (Contex Diagram)

Diagram konteks dari aplikasi kamus dan aksara karo adalah seperti pada gambar 3.2 berikut ini.



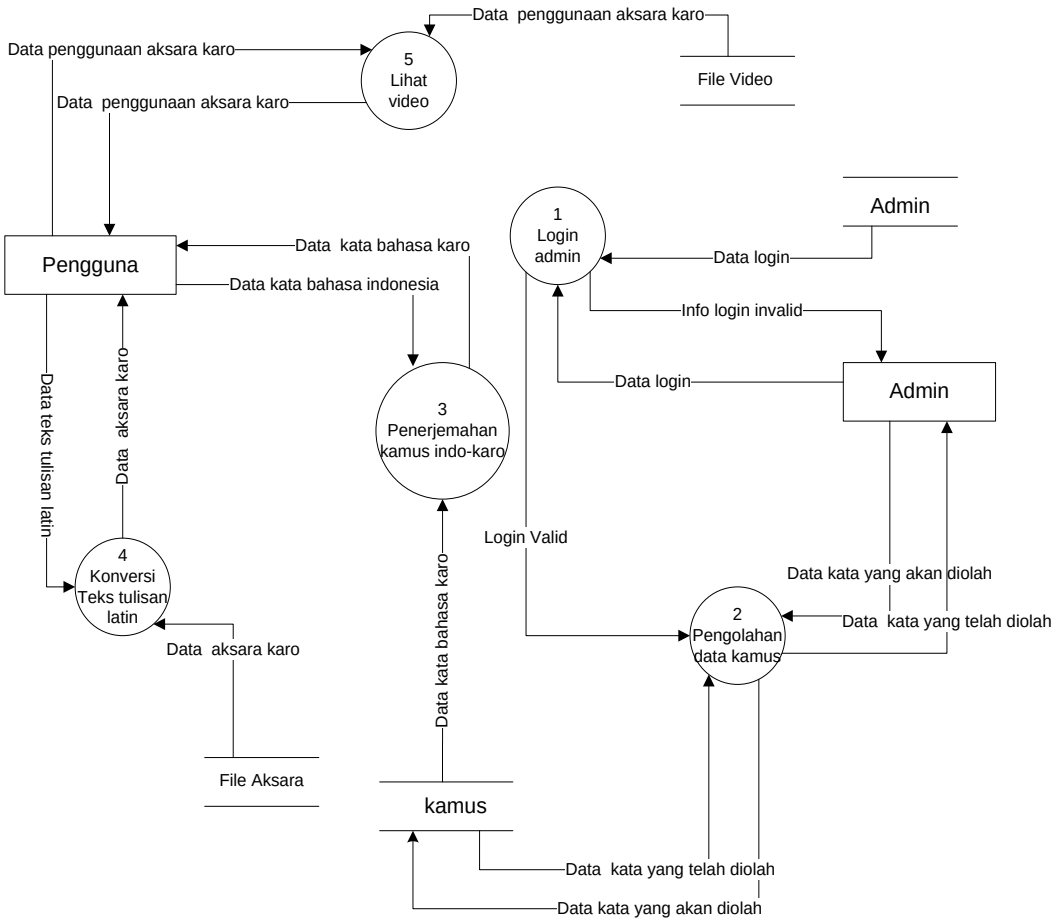
Gambar 3. 2 Diagram Konteks

3. 1.6.2 Data Flow Diagram (DFD)

DFD merupakan model dari sistem untuk menggambarkan pembangunan sistem ke model yang lebih kecil. Berikut ini adalah gambar DFD untuk aplikasi kamus dan aksara Karo.

3. 1.6.2.1 Data Flow Diagram level 1

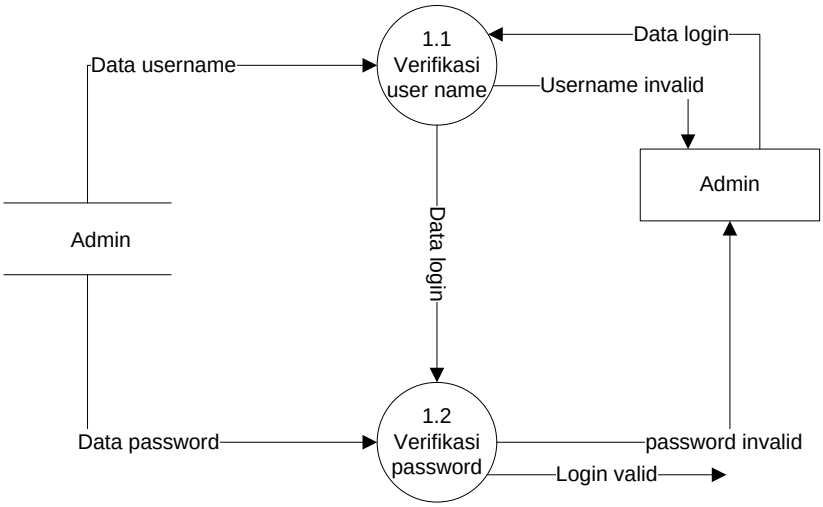
DFD Level 1 dibuat jika pada diagram Konteks masih terdapat proses yang harus dijelaskan lebih rinci, pada DFD Level 1 terdapat pada Gambar 3.3 berikut.



Gambar 3. 3 DFD Level 1

3. 1.6.2.2 Data Flow Diagram Level 2 Proses 1

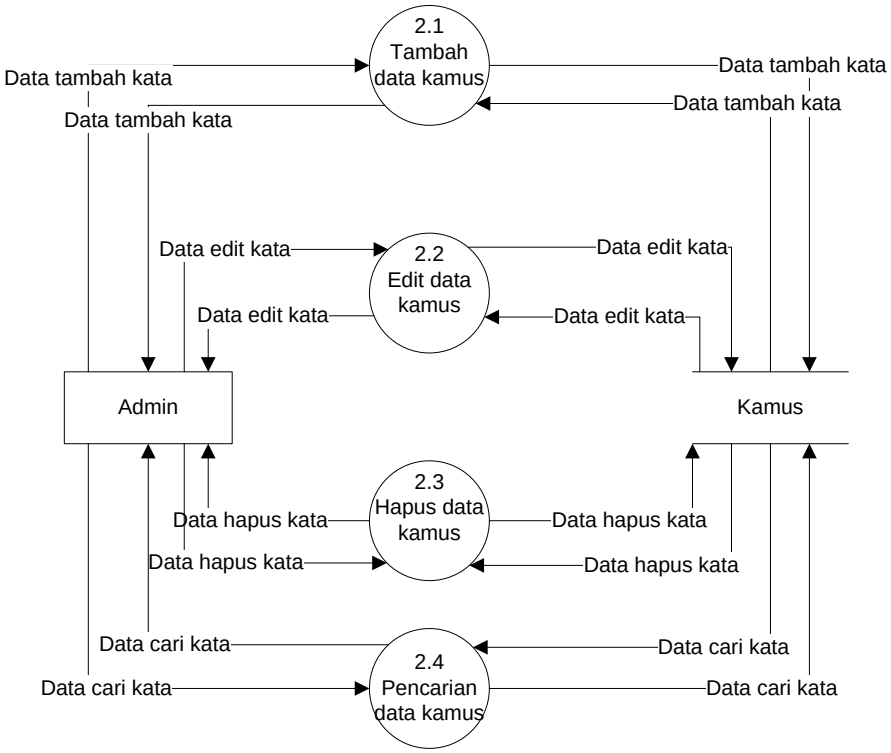
DFD Level 2 proses 1.0 merupakan pecahan dari proses *Login*, adapun proses yang terjadi yaitu : *Input* username, *input* password. Untuk lebih jelasnya dapat dilihat pada Gambar 3.4 berikut .



Gambar 3. 4 DFD Level 2 Proses 1

3. 1.6.2.2.1 Data Flow Diagram Level 2 Proses 2

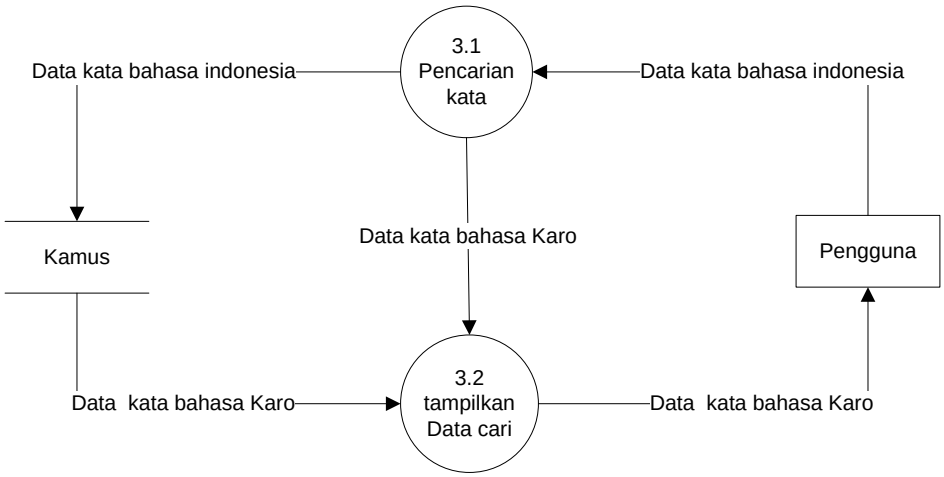
Data Flow Diagram level 2 proses 2 merupakan pecahan dari proses Pengolahan Data, adapun proses yang terjadi yaitu: Tambah Data, Edit Data, Hapus Data dan Cari data,. Untuk lebih jelasnya dapat dilihat pada Gambar 3.5 berikut :



Gambar 3. 5 DFD Level 2 Proses 2

3. 1.6.2.2.2 Data Flow Diagram Level 2 Proses 3

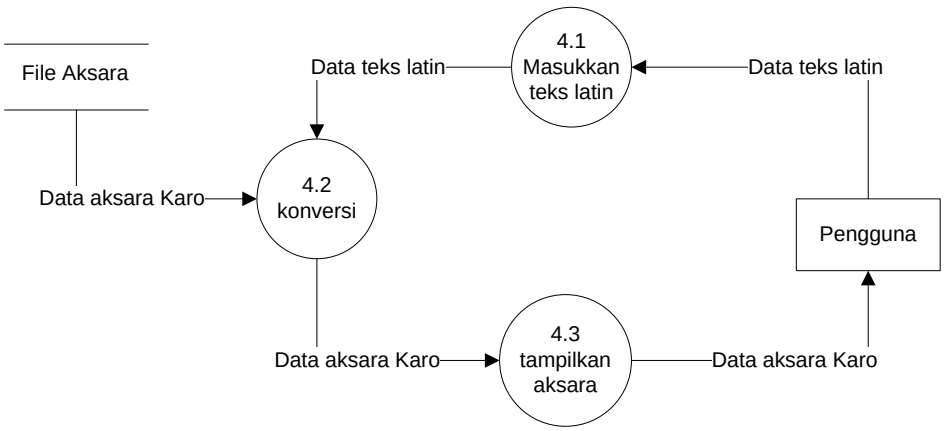
Data Flow Diagram level 2 proses 3 merupakan pecahan dari proses pecarian arti kata, adapun proses yang terjadi yaitu : *input* data cari, output data cari. Untuk lebih jelasnya dapat dilihat pada Gambar 3.6 berikut.



Gambar 3. 6 DFD Lvel 2 proses 3

3. 1.6.2.2.3 Data Flow Diagram Level 2 Proses 4

Data Flow Diagram level 2 proses 4 merupakan pecahan dari proses pengkonversian teks tulisan latin ke dalam aksara karo, adapun proses yang terjadi yaitu : *input* teks tulisan latin, konversi,output aksara karo. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7 berikut.



Gambar 3. 7 DFD Level 2 Proses 4

3. 1.6.3 Spesifikasi Proses

Spesifikasi proses bertujuan untuk untuk mendeskripsikan dari setiap fungsi yang disajikan pada diagram alir data. Spesifikasi proses pada alir data aplikasi kamus dan aksara karo adalah seperti tabel 3.1 berikut :

Tabel 3. 1 Tabel Spesifikasi Proses

No	Proses	Keterangan
1	No. Proses	1
	Nama	Login Admin
	Source	Admin
	Input	Data login
	Output	Info login invalid
	Destination	Admin
	Logika Proses	1. Admin memasukkan data login, klik login 2. Jika username dan password yang di*input*kan valid maka masuk ke pengolahan data kamus 3. jika username dan password yang diinputkan invalid maka sistem akan menampilkan pesan kesalahan
	No. Proses	2
	Nama	Pengolahan Data kamus
	Source	Admin

2	*Input*	Data kata yang akan diolah, data kata yang telah diolah
	Output	Data kata yang telah diolah, data kata yang akan diolah
	Destination	Admin
	Logika Proses	1. Admin melakukan login, jika valid sistem akan menampilkan form pengolahan data kamus 2. Admin akan memilih data kata yang akan diolah
3	No. Proses	3
	Nama	Penerjemahan indo-karo
	Source	Pengguna
	Input	Data kata bahasa Indonesia, Data kata bahasa karo
	Output	Data kata bahasa karo
	Destination	Pengguna
	Logika Proses	1. Pengguna memasukkan data kata bahasa Indonesia yang ingin diterjemahkan ke dalam bahasa Karo kemudian klik cari 2. Jika kata yang dicari valid, maka sistem akan menampilkan kata dalam bahasa Karo 3. jika kata invalid maka akan ditampilkan pesan bahwa kata tersebut belum ada di dalam database

4	No. Proses	4
	Nama	Konversi teks tulisan latin
	Source	Pengguna
	Input	Data teks tulisan latin , data aksara karo
	Output	Data aksara karo
	Destination	pengguna
	Logika Proses	1. Pengunjung memasukkan data teks latin yang ingin dikonversi ke dalam aksara karo 2. Teks yang diinputkan akan dicek satu persatu 3. Periksa karakter pertama, apakah termasuk angka?, jika ya maka sistem akan mengenalinya sebagai token t_AK kemudian akan dikonversi menjadi angka dalam aksara Karo 4. Apabila tandabaca maka sistem akan mengenalinya sebagai token t_TB kemudian akan dikonversi menjadi tandabaca dalam aksara Karo 5. Apabila operator maka sistem akan mengenalinya sebagai token t_OP kemudian akan dikonversi menjadi operator dalam aksara Karo 6. Apabila karakter maka sistem akan mengenalinya sebagai token t_KRT kemudian akan dikonversi menjadi karakter dalam aksara Karo 7. Apabila huruf maka sistem akan

		mengenalinya sebagai token t_STR kemudian memeriksa karakter selanjutnya, apakah membentuk bunyi bentukan atau tidak? Jika ya maka akan menghasilkan token t_BT dan akan dikonversi menjadi bunyi bentukan dalam aksara Karo. Jika tidak akan menghasilkan token t_M dan akan dikonversi menjadi bunyi mandiri dalam aksara Karo.
5	No. Proses	5
	Nama	Lihat video
	Source	Pengunjung
	Input	Data penggunaan aksara karo
	Output	Data penggunaan aksara karo
	Destination	Pengunjung
	Logika Proses	1. Pengunjung mengklik Video 2. System akan menampilkan Video
6	No. Proses	1.1
	Nama	Verifikasi username
	Source	Admin
	Input	Data login, data username
	Output	Username invalid, data login
	Destination	Admin

	Logika Proses	1. admin memasukkan data login 2. Jika username yang dimasukkan valid maka system akan menampilkan halaman utama admin 3. Jika username invalid maka akan ditampilkan pesan kesalahan
7	No. Proses	1.2
	Nama	Verifikasi password
	Source	Admin
	Input	Data login, data password
	Output	Info login invalid
	Destination	Admin
	Logika Proses	1. admin memasukkan data login 2. Jika password yang dimasukkan valid maka system akan menampilkan halaman utama admin 3. Jika password invalid maka akan ditampilkan pesan kesalahan
8	No. Proses	2.1
	Nama	Tambah data kamus
	Source	Admin
	Input	Data tambah kata
	Output	Data tambah kata

	Destination	admin
	Logika Proses	1. Sistem menampilkan form tambah data Kamus 2. Admin mengisi form tambah data kamus. 3. Jika admin mengklik simpan, maka data kamus baru disimpan ke *database* 4. Jika admin mengklik reset, maka data kata baru tidak ditambahkan ke *database*, dan form tambah data Kamus dikosongkan.
9	No. Proses	2.2
	Nama	Edit data kamus
	Source	Admin
	Input	Data edit kata
	Output	Data edit kata
	Destination	Admin
	Logika Proses	1. Tampilkan data kamus 2. Admin memilih kata yang akan diedit 3. System akan menampilkan data kamus yang akan diedit ke form pengeditan 4. Admin mengedit data kamus baru di form edit 5. Jika admin mengklik simpan, maka data kamus baru disimpan ke *database* 6. Jika admin mengklik reset, maka form edit kata akan dikosongkan dan data kamus baru tidak disimpan dalam *database*

11	No. Proses	2.3
	Nama	Hapus data kamus
	Source	Admin
	Input	Data hapus kata
	Output	Data hapus kata
	Destination	Admin
	Logika Proses	1. Tampilkan data kamus 2. Admin memilih kata yang akan dihapus 3. Admin mengklik hapus 4. System akan menampilkan pesan, apakah data tesebut akan dihapus 5. Jika admin memilih Ok, maka kata tersebut akan dihapus 6. Jika admin memilih Cancel, maka kata tersebut tidak dihapus
12	No. Proses	2.4
	Nama	Pencarian data kamus
	Source	Admin
	Input	Data cari kata
	Output	Data cari kata
	Destination	Admin
	Logika Proses	1. Admin memilih menu pencarian data kamus 2. System menampilkan form pencarian data kamus

		3. Admin memasukkan data kata yang ingin dicari 4. Admin mengklik cari, jika kata yang dicari ada maka system akan menampilkan kata tersebut. Jika kata yang dicari tidak ada maka system akan menampilkan pesan kesalahan
13	No. Proses	3.1
	Nama	Pencarian kata
	Source	Pengguna
	Input	Data kata bahasa Indonesia
	Output	Data kata bahasa Indonesia, data kata bahasa karo
	Destination	Pengguna
	Logika Proses	1. Pengunjung memasukkan data kata dalam bahasa Indonesia yang ingin diterjemahkan ke dalam bahasa Karo 2. Pengunjung mengklik terjemahkan 3. jika data kata yang dicari valid, maka tampilkan datakata dalam bahasa Karo 4. jika data kata invalid, maka tampil pesan kesalahan
	No. proses	3.2
	Nama	tampilkan data cari
	Source	Pengguna

14	*Input*	Data kata bahasa karo
	Output	Data kata bahasa karo
	Destination	Pengguna
	Logika proses	1. Data kata yang dimasukkan oleh pengguna di cek ke dalam database 2. Apabila data kata valid, tampilkan data kata yang dicari 3. Apabila data kata invalid, maka tampil pesan kesalahan
15	No. proses	4.1
	Nama	Masukkan teks latin
	Source	Pengguna
	Input	Data teks latin
	Output	Data teks latin
	Destination	Pengguna
	Logika proses	1. Tampilkan form aksara 2. Pengguna memasukkan data teks tulisan latin 3. Klik konversi
	No. proses	4.2
	Nama	Konversi
	Source	Pengguna
	Input	Data teks latin

16	*Output*	Data teks latin, data aksara Karo
	Destination	Pengguna
	Logika proses	1. Pengunjung memasukkan data teks latin yang ingin dikonversi ke dalam aksara karo 2. Teks yang diinputkan akan dicek satu persatu 3. Periksa karakter pertama, apakah termasuk angka?, jika ya maka sistem akan mengenalinya sebagai token t_AK kemudian akan dikonversi menjadi angka dalam aksara Karo 4. Apabila tandabaca maka sistem akan mengenalinya sebagai token t_TB kemudian akan dikonversi menjadi tandabaca dalam aksara Karo 5. Apabila operator maka sistem akan mengenalinya sebagai token t_OP kemudian akan dikonversi menjadi operator dalam aksara Karo 6. Apabila karakter maka sistem akan mengenalinya sebagai token t_KRT kemudian akan dikonversi menjadi karakter dalam aksara Karo 7. Apabila huruf maka sistem akan mengenalinya sebagai token t_STR kemudian memeriksa karakter selanjutnya, apakah membentuk bunyi bentukan atau tidak? Jika ya maka akan menghasilkan token t_BT dan akan dikonversi menjadi bunyi bentukan dalam aksara Karo. Jika tidak akan menghasilkan token t_M dan akan

		dikonversi menjadi bunyi mandiri dalam aksara Karo.
17	No. proses	4.3
	Nama	Tampilkan aksara
	Source	Pengguna
	Input	Data aksara karo
	Output	Data aksara karo
	Destination	Pengguna
	Logika proses	1. Data teks latin yang diinput dicek ke dalam array 2. Apabila angka, maka ditampilkan angka dalam aksara Karo 3. Apabila tandabaca, maka ditampilkan tandabaca dalam aksara Karo 4. Apabila operator, maka ditampilkan operator dalam aksara Karo 5. Apabila karakter, maka ditampilkan karakter dalam aksara Karo 6. Apabila huruf, maka diperiksa karakter selanjutnya apakah membentuk bunyi bentukan?, jika ya maka akan menghasilkan bunyi bentukan dan ditampilkan bunyi bentukan dalam aksara Karo, jika tidak maka akan menghasilkan bunyi mandiri dan ditampilkan bunyi mandiri dalam aksara Karo.

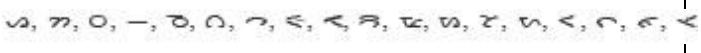
3. 1.6.4 Kamus Data

Kamus data (*data dictionary*) merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi daari suatu sistem. Kamus data dibuat berdasarkan arus data yang ada pada diagram alir data. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir pada sistem secara lengkap. Kamus data digunakan untuk merancang *input*, merancang laporan dan database. Kamus data yang dibutuhkan untuk membuat aplikasi kamus dan aksara karo dapat dilihat pada table 3.2 berikut.

Tabel 3. 2 Kamus Data

Nama	Data Login
Where used / how used	Proses 1.0 Login Admin Proses 1.1 verifikasi username Proses 1.2 verifikasi password
Deskripsi	Berisi seluruh data admin
Struktur data	Username + password
Username Password	{A-Z | a-z} {A-Z | a-z | 0-9 | simbol }
Nama	Data kata

Where used / how used	Proses 2.0 pengolahan data kamus Proses 3.0 penerjemahan kamus Indo-Karo Proses 2.1 tambah data kamus Proses 2.2 edit data kamus Proses 2.3 hapus data kamus Proses 2.4 pencarian data kamus Proses 3.1 pencarian kata Proses 3.2 tampilkan data cari
Deskripsi	Berisi seluruh data kata kamus
Struktur data	Bahasa Indonesia + bahasa karo
Bahasa Indonesia Bahasa Karo	{A-Z | a-z | 0-9 | tanda_baca | operator } {A-Z | a-z | 0-9 | tanda_baca | operator }
Nama	Data teks latin
Where used / how used	Proses 4.0 konversi teks tulisan latin Proses 4.1 masukkan teks latin Proses 4.2 konversi
Deskripsi	Berisi seluruh data teks yang di*input*kkan ke dalam aplikasi yang ingin di konversi ke dalam aksara Karo.

Struktur data	Teks tulisan latin
Teks tulisan latin	{A-Z | a-z | tanda_baca | operator }
Nama	Data aksara Karo
Where used / how used	Proses 4.0 Proses 4.2
Deskripsi	Berisi seluruh data output dalam aksara Karo
Struktur data	Aksara Karo
Aksara Karo	{  | 0-9 | &, @, #, ~, %, $, _, \, |, (,{, [,],},)| ., ,, ?,!, ‘, “, ;| +, - , *, / , ^, = ,< >, ≤ , ≥ }

3. 2 Perancangan Sistem

Perancangan sistem merupakan suatu proses yang mengaplikasikan berbagai teknik dan prinsip yang bertujuan mendefinisikan kebutuhan suatu sistem sedemikian detail sehingga mendekati keadaan fisik yang sebenarnya dan untuk menghasilkan suatu yang baru, atau memperbaharui sistem yang ada untuk meningkatkan efektifitas kerja yang sesuai dengan teknologi dan fasilitas yang tersedia. Dimana suatu perancangan sistem informasi merupakan suatu langkah

awal dalam pembuatan suatu sistem sehingga perlu diperhatikan kebutuhan dari sistem atau *software* yang diharapkan.

3. 2.1 Perancangan Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam pembangunan aplikasi kamus dan aksara karo adalah sebagai berikut:

Tabel 3. 3 Tabel Admin

Nama Tabel : Admin			
Primary Key : id_admin			
Nama Field	**Tipe Data**	**Ukuran**	**Keterangan**
Id_admin	Integer	6	PK
Username	Varchar	30	Username_admin
Password	Varchar	25	Password_admin

Tabel 3. 4 Tabel Kamus

Nama Tabel : Kamus			
Nama Field	**Tipe Data**	**Ukuran**	**Keterangan**
Indonesia	Varchar	20	Kata bahasa

			Indonesia
Karo	Text		Kata bahasa karo

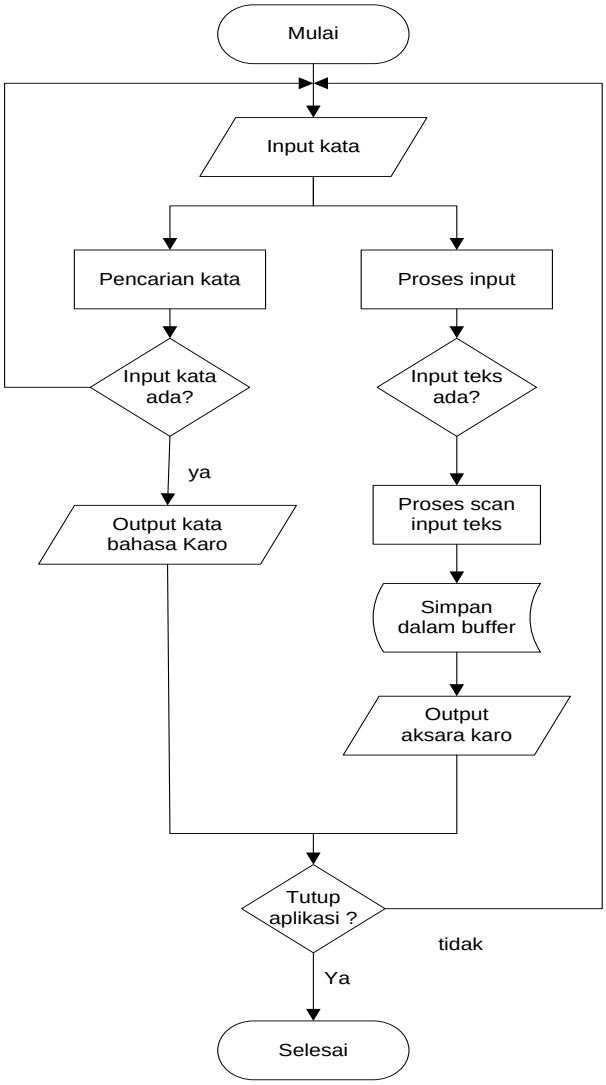
3. 2.2 Perancangan File

File-file yang digunakan dalam pembangunan aplikasi kamus dan aksara karo adalah sebagai berikut:

1. Untuk pembangunan Aksara digunakan file gambar (.png) yang disimpan dalam satu folder.
2. Untuk pembangunan Video digunakan file video (.mp4) yang disimpan dalam satu folder.

3. 2.3 Perancangan Prosedural

Untuk menggambarkan proses yang dapat dilakukan pengguna dalam aplikasi kamus dan aksara karo yang dimulai dari proses *input* sampai hasil output, penulis menggunakan model proses prosedural yaitu dengan menggunakan *flowchart,* adapun proses umum dalam sistem tampak seperti gambar 3.8 berikut.



Gambar 3. 8 Flowchart Umum Sistem

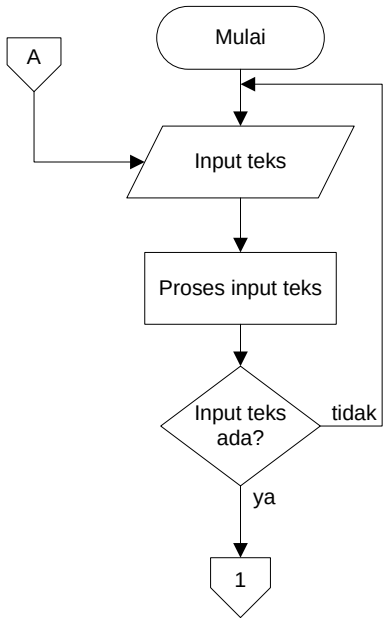
Dalam aplikasi kamus dan aksara karo terdapat dua proses utama yaitu kamus (terjemahan kata dalam bahasa Indonesia ke bahasa karo) dan pengkonversian teks tulisan latin ke dalam aksara karo.

3.2.3.1 Pengkonversian Teks menjadi Aksara Karo

Flowchart pengkonversian teks menjadi aksara karo dibagi menjadi beberapa bagian yaitu sebagai berikut.

1. *Input* teks tulisan latin

Flowchart dari *input* dari *input* teks tulisan latin adalah seperti pada gambar 3.9 berikut ini.



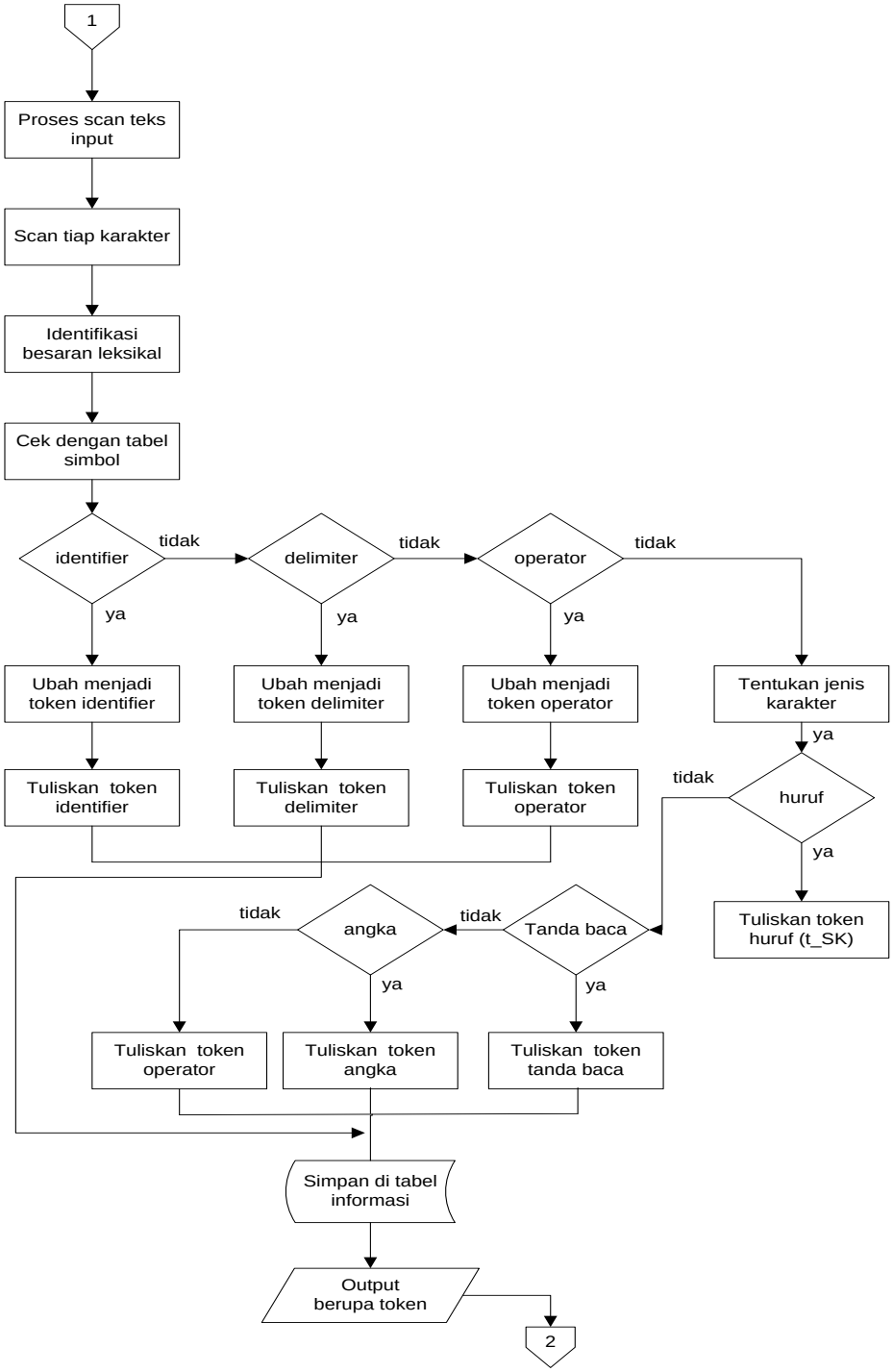
Gambar 3. 9 Flowchart *Input* Teks Tulisan Latin

2.Analisis leksikal

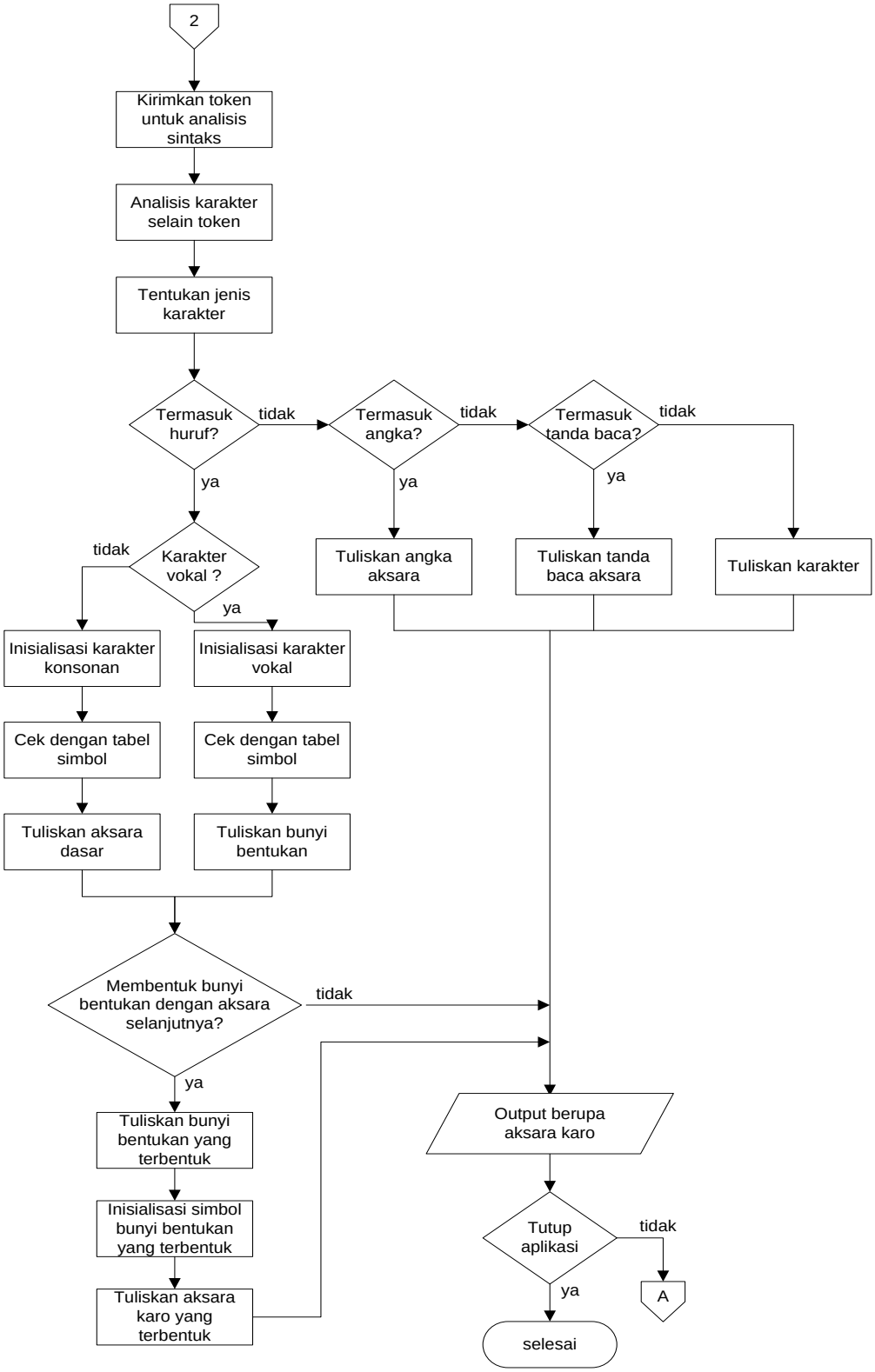
Flowchart dari bagian penganalisis leksikal kamus dan aksara karo adalah seperti pada gambar 3.10.

3.Analisis sintaksis

Flowchart dari bagian penganalisis sintaksis kamus dan aksara karo adalah seperti pada gambar 3. 11.



Gambar 3. 10 Flowchart Analisis Leksikal

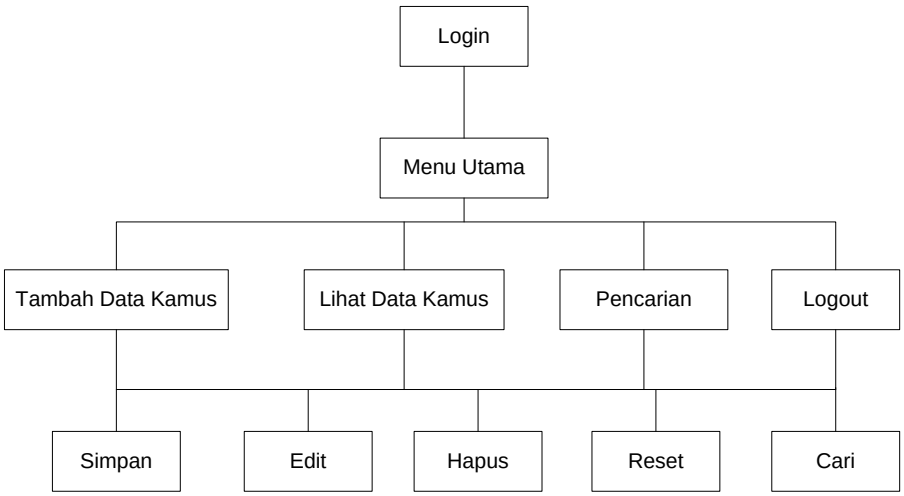


Gambar 3. 11 Flowchart Analisis Sintaks

3. 2.4 Perancangan Menu Aplikasi

Perancangan menu aplikasi digunakan untuk menjelaskan menu, sub menu yang terdapat didalam aplikasi kamus dan aksara karo agar pengguna dapat lebih mudah dalam menggunakannya. Pengguna aplikasi kamus dan aksara karo yaitu admin dan pengguna.

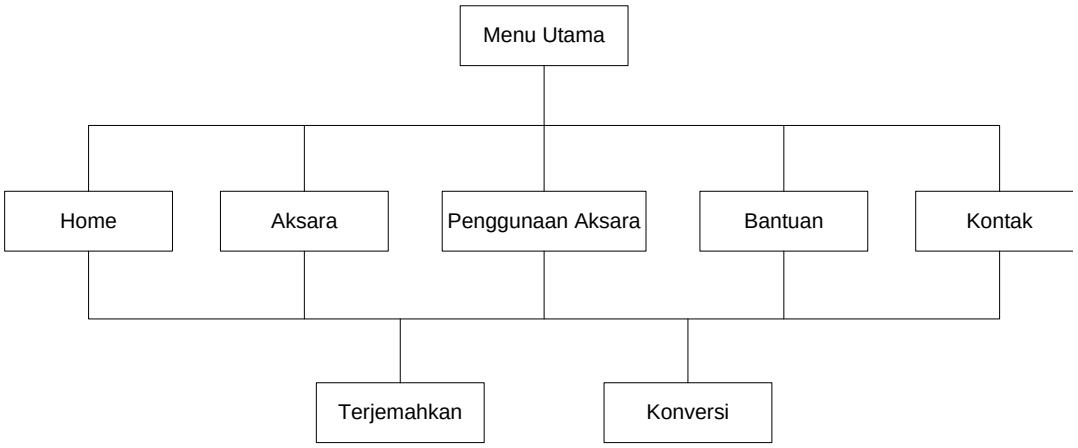
1. Menu Aplikasi Admin



Gambar 3. 12 Perancangan Menu Admin

Menu ini muncul ketika halaman admin dibuka oleh admin, form ini mempunyai fasilitas tambah, edit hapus dan cari yang dikelola oleh admin.

2. Menu Aplikasi Pengguna



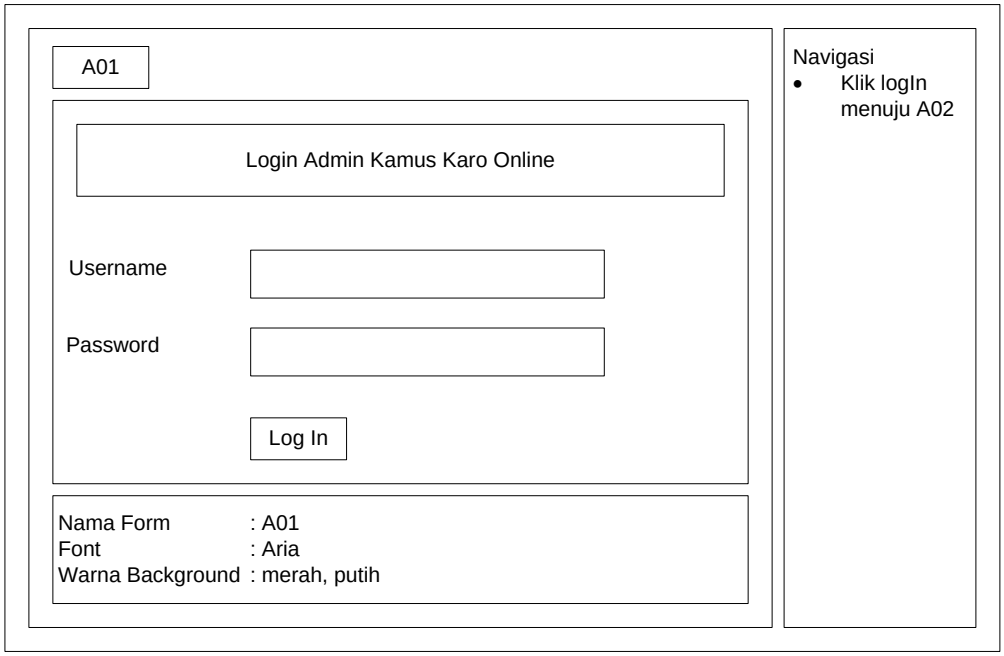
Gambar 3. 13 Perancangan Menu Pengguna

Menu ini muncul ketika aplikasi dibuka, sistem ini mempunyai fasilitas kamus, konversi teks tulisan latin ke aksara karo, Penggunaan Aksara Karo, Bantuan dan Kontak.

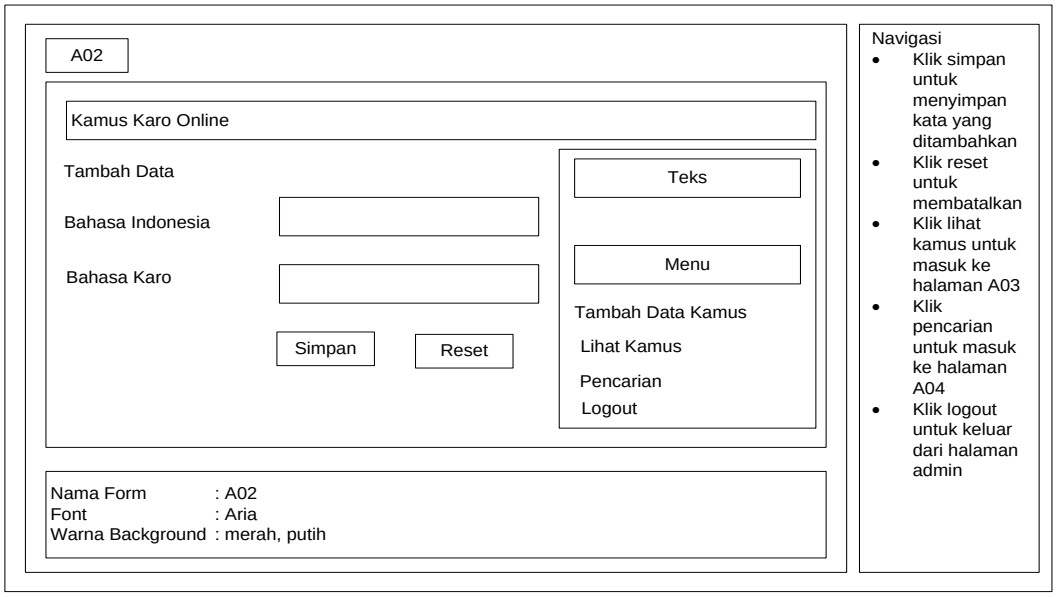
3. 2.5 Perancangan Antar Muka

Perancangan Antar muka merupakan media komunikasi antara pemakai dengan sistem komputer. Hasil perancangan antarmuka pemakai akan memudahkan proses implementasi

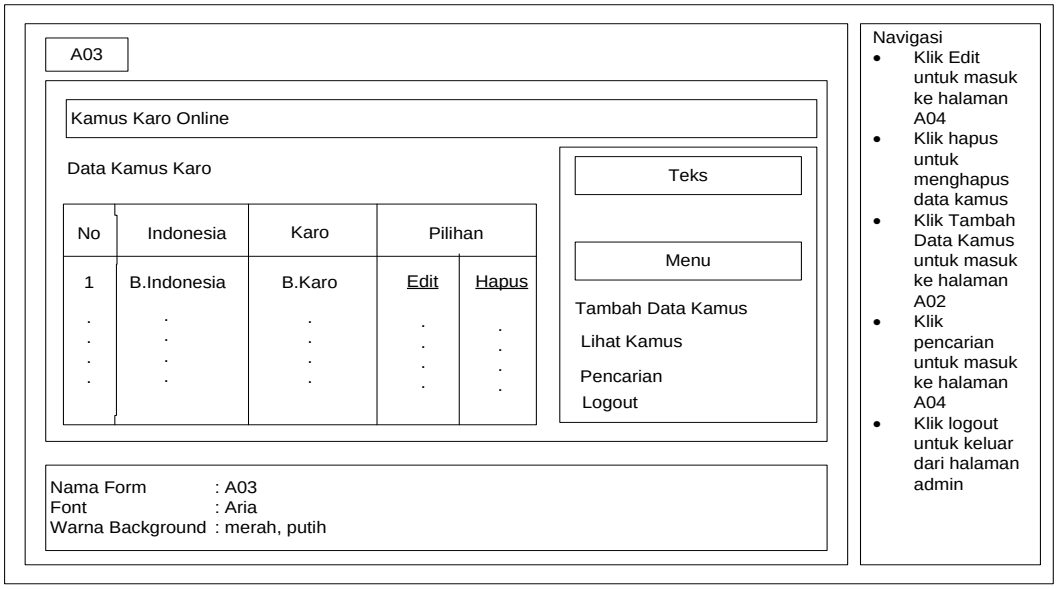
3.2.5.1 Perancangan Antar Muka Admin



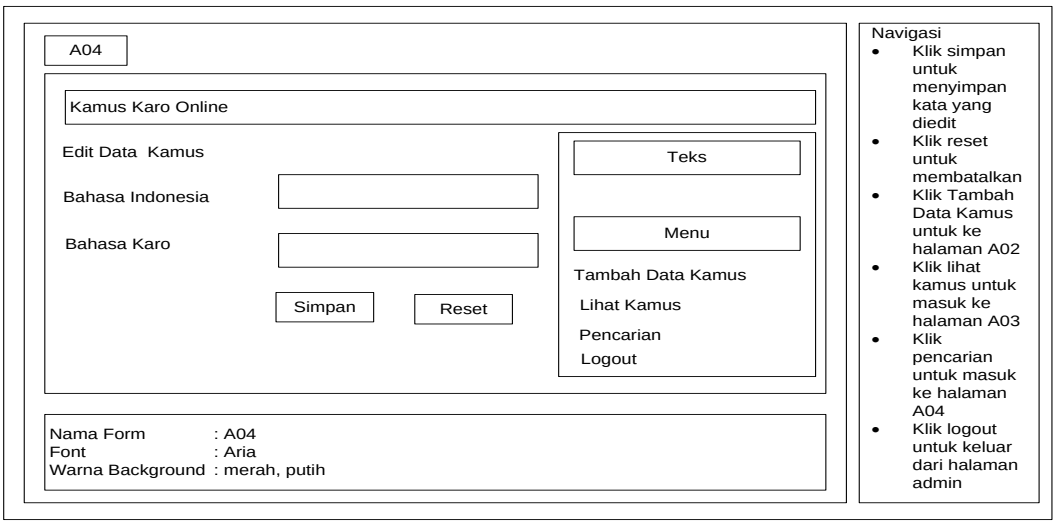
Gambar 3. 14 Perancangan Antar Muka Login Admin



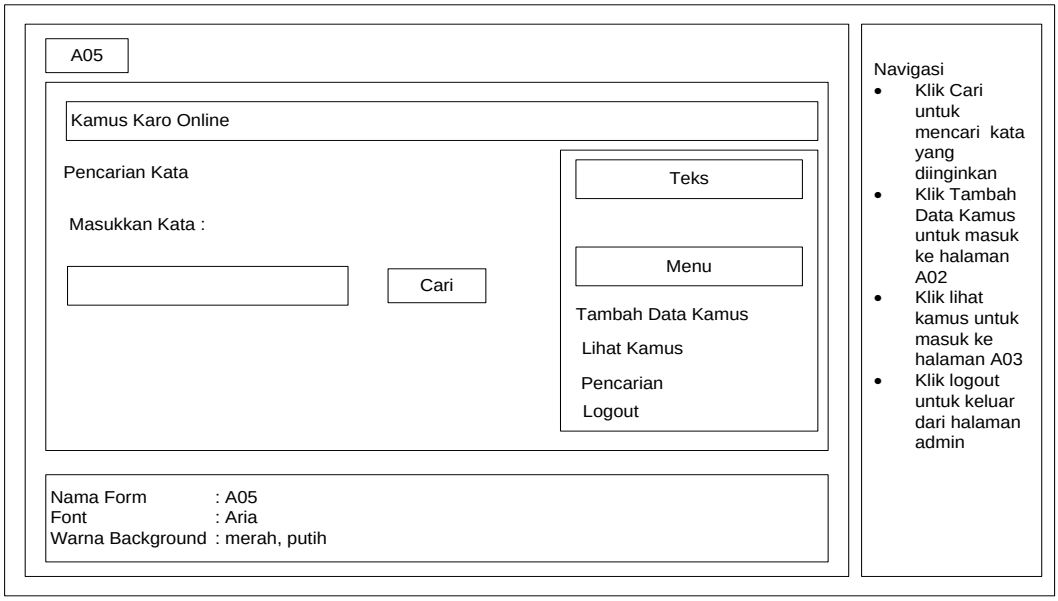
Gambar 3. 15 Perancangan Antar Muka Admin Tambah Data Kamus



Gambar 3. 16 Perancangan Antar Muka Admin Lihat Data Kamus

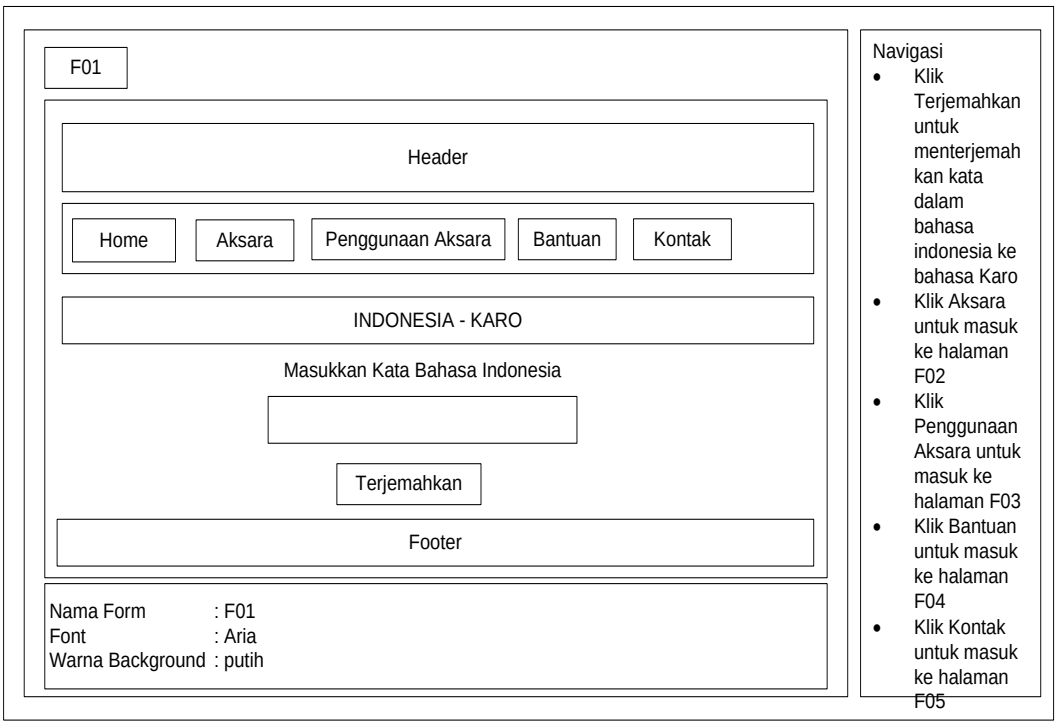


Gambar 3. 17 Perancangan Antar Muka Admin Edit Data Kamus

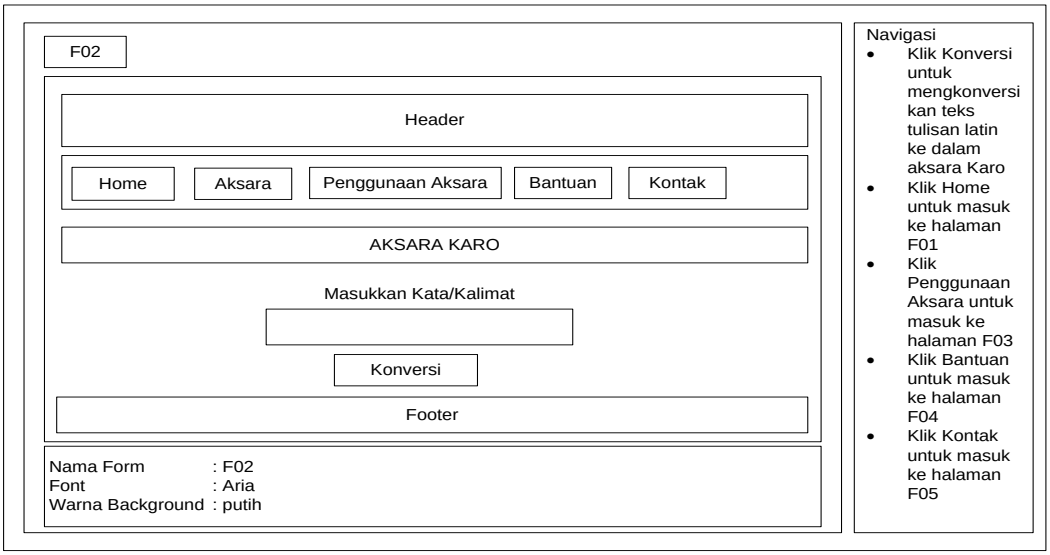


Gambar 3. 18 Perancangan Antar Muka Admin Pencarian Data Kamus

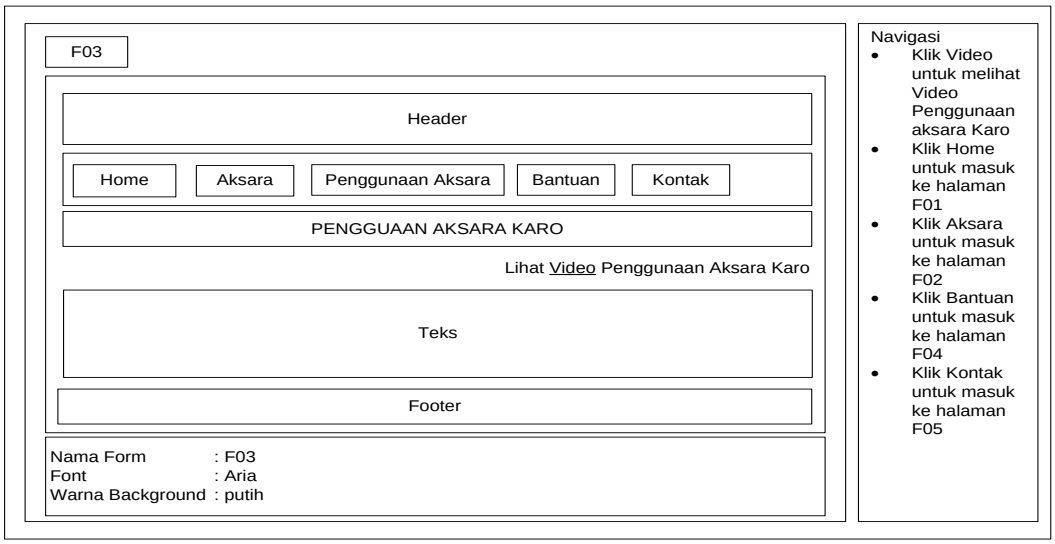
3.2.5.2 Perancangan Antar Muka Pengunjung



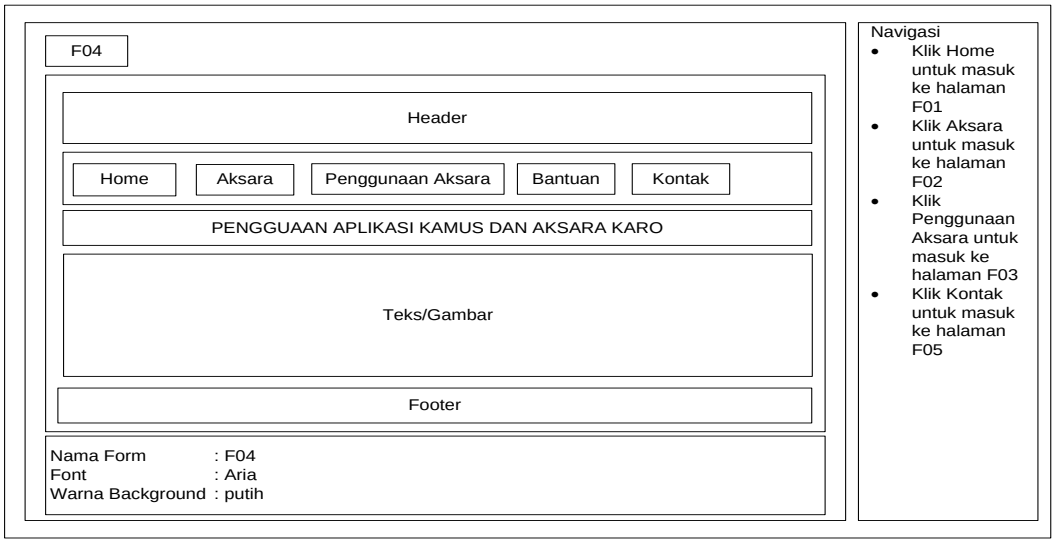
Gambar 3. 19 Perancangan Antar Muka Pengunjung Menu Home



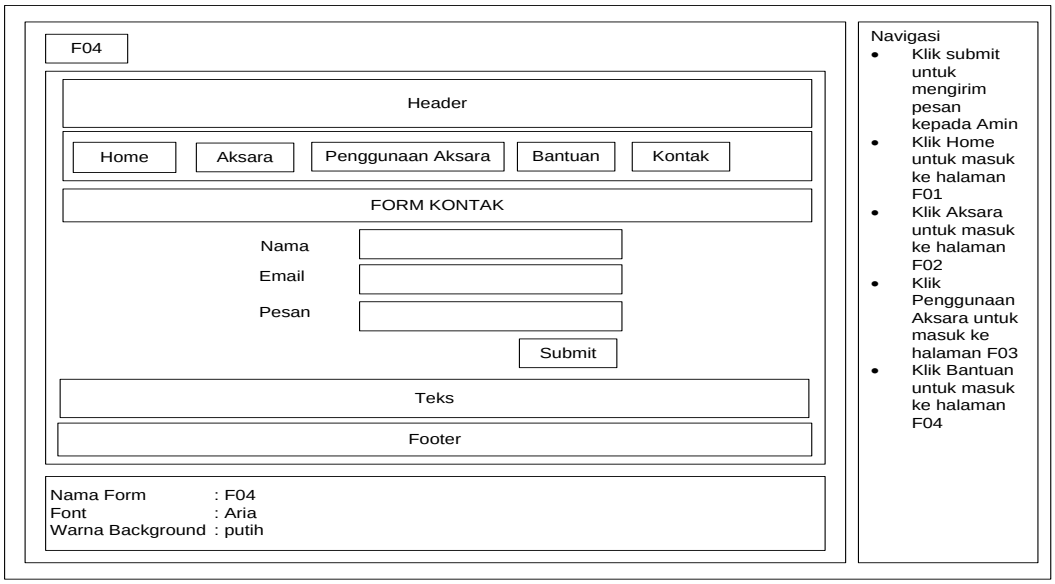
Gambar 3. 20 Perancangan Antar Muka Pengunjung Menu Aksara Karo



Gambar 3. 21 Perancangan Antar Muka Pengunjung Menu Penggunaan Aksra Karo



Gambar 3. 22 Perancangan Antar Muka Pengunjung Menu Bantuan

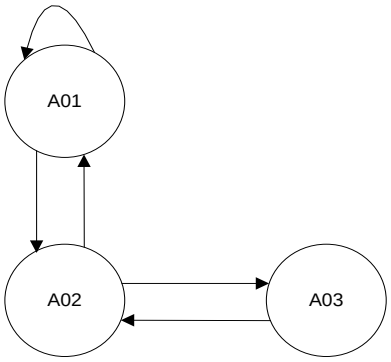


Gambar 3. 23 Perancangan Antar Muka Pengunjung Menu Kontak

3. 2.6 Jaringan Semantik

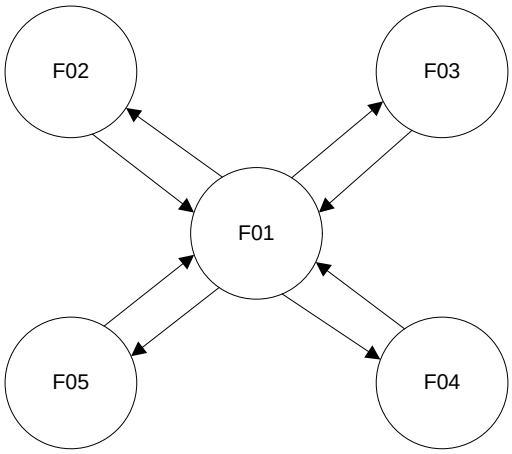
Jaringan semantik merupakan keterhubungan navigasi menu dari satu form dengan form lainnya. Berikut adalah jaringan semantik untuk Kamus dan Aksara Karo.

3.2.6.1 Jaringan Semantik Admin



Gambar 3. 24 Jaringan Semantik Admin

3.2.6.2 Jaringan Semantik Pengunjung



Gambar 3. 25 Jaringan Semantik Pengunjung