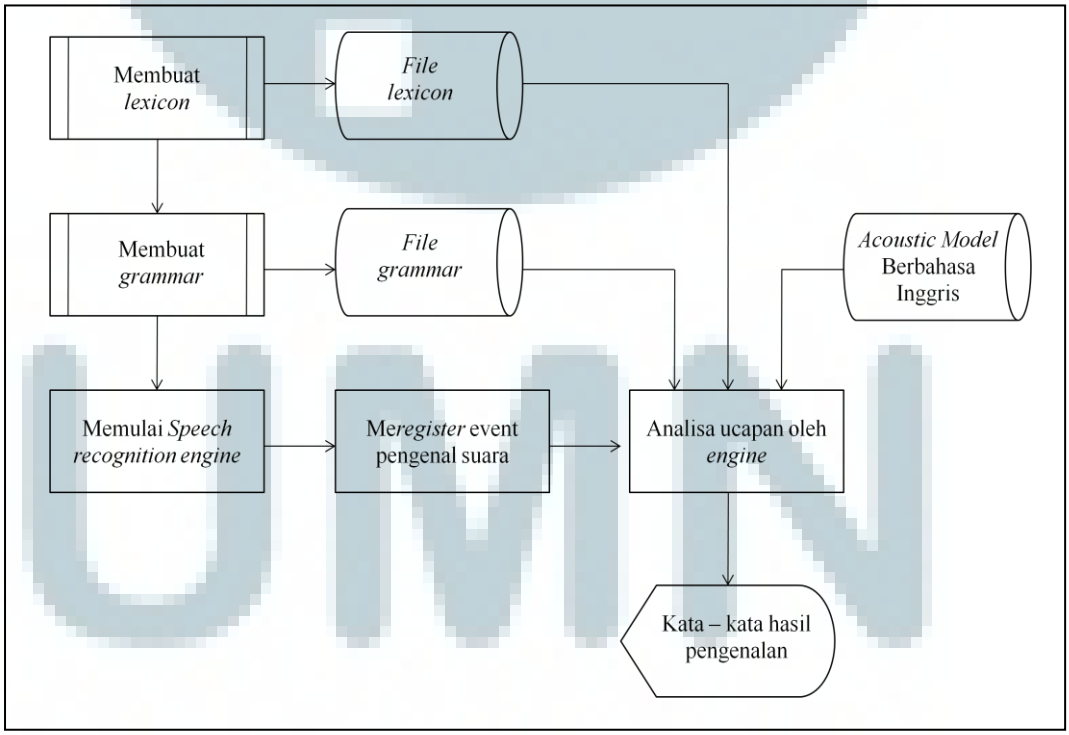
BAB III

ANALISIS DAN PERANCANGAN APLIKASI

3.1 Analisis Aplikasi

Pada penelitian ini, teknologi ASR yang digunakan adalah *engine* Microsoft *Speech Recognition*. *Engine* pengenal ucapan / *speech recognizer* ini telah dilengkapi oleh *acoustic model* / pemodelan akustik sesuai dengan jenis bahasa yang akan digunakan sebagai acuan pengenalan, sehingga *Finite Automata* akan berfokus pada dua tahapan lainnya, yaitu penyediaan *lexicon* dan *language model.* Berikut adalah *system flow* yang menggambarkan cara kerja aplikasi pendiktean berbasis ASR yang akan dibangun.

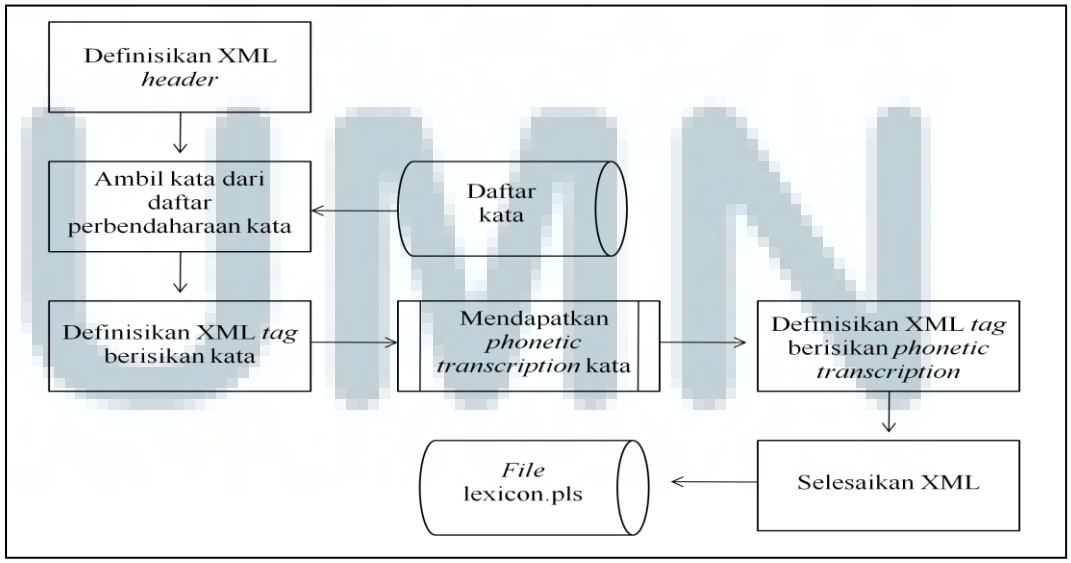


Gambar 3.1 *System Flow* Pendiktean Oleh Aplikasi

Dari *system flow* pada gambar 3.1, dapat dilihat bahwa proses pendiktean diawali dengan pembuatan *file lexicon* dan *file grammar* sebagai *language model* dari aplikasi ASR. *Engine* pengenal ucapan (Microsoft *Speech Recognition*) kemudian akan digunakan untuk mengenali suara yang diucapkan oleh pengguna, dengan berdasarkan pada *lexicon, grammar,* dan *acoustic model* berbahasa Inggris yang sudah tersedia dalam *engine*. Hasil pengenalan kata sebagai hasil pendiktean kemudian akan ditampilkan kepada pengguna. *Finite automata* sendiri akan digunakan dalam membuat *file lexicon* dan *file grammar* dengan penjelasan sebagai berikut.

3.1.1 *Finite Automata* Pada *Lexicon*

Lexicon mengacu pada dua hal utama, yaitu kata dan representasi fonem pelafalan dari kata tersebut dalam bentuk *phonetic transcription*. *Lexicon* sendiri pada *engine* ini merupakan sebuah *file* terpisah berformat XML dan memiliki ekstensi .pls. Berikut adalah diagram *system flow* yang menggambarkan proses pembuatan *lexicon* dalam penelitian ini.



Gambar 3.2 *System Flow* Pembuatan *File Lexicon* (lexicon.pls)

Proses pembuatan *file lexicon* diawali dengan pendefinisian XML *header* dan *tag* yang berisikan representasi kata secara utuh. Proses selanjutnya adalah mendapatkan *phonetic transcription* dari kata tersebut untuk kemudian ditambahkan ke dalam *tag* lain yang berfungsi sebagai pendefinisian pelafalan dari kata yang bersangkutan. Pada akhir proses, akan terbentuk suatu *file* bernama lexicon.pls yang akan berfungsi sebagai *lexicon* dari aplikasi.

Finite Automata akan diterapkan untuk mendapatkan *phonetic transcription* yang paling maksimal dari suatu kata. Simbol pelafalan yang digunakan adalah simbol UPS (*Universal Phone Set*), karena representasinya yang lebih mudah dibaca dan diimplementasikan dibandingkan dengan simbol IPA (*International Phonetic Alphabet*) dan simbol *Speech* API (SAPI) . Berdasarkan pada teori mengenai *phonetic transcription* yang ada pada bagian sebelumnya, maka terdapat beberapa hal yang perlu diperhatikan dalam menerapkan *Finite Automata* untuk membuat *phonetic transcription*.

1. Translasi Fonem

   Karena *engine* tidak menyediakan pemodelan akustik dalam Bahasa Indonesia, maka perlu dilakukan translasi fonem agar didapatkan suatu *phonetic transcription* yang paling sesuai terhadap suatu kata. Dengan mengacu pada translasi fonem yang telah dilakukan oleh Ferdiansyah (2012), alfabet penyusun kata berbahasa Indonesia oleh Kuntarto (2011), fonem Bahasa Indonesia oleh Lestari (2010), serta aturan simbol UPS (*Universal Phone Set*) oleh Microsoft (2011), didapatkanlah hasil analisis translasi

fonem yang dibutuhkan untuk ASR dalam penelitian ini. Hasil analisis translasi fonem oleh penulis disajikan dalam tabel berikut.

Tabel 3.1 Hasil Analisis Translasi Fonem ke Simbol UPS

Huruf Alfabet	Fonem Indonesia	Translasi Fonem Dalam ARPAbet	Translasi Fonem Dalam UPS
a	/a/	/ah/	**AH**
b	/b/	/b/	**B**
c	/c/	/ch/	**CH**
d	/d/	/d/	**D**
e	/e/	/eh/	**EH**
E	/E/	/ah/	**AX**
f	/f/	/f/	**F**
g	/g/	/g/	**G**
h	/h/	/hh/	**H**
i	/i/	/ih/	**IH**
j	/j/	/jh/	**JH**
k	/k/	/k/	**K**
l	/l/	/l/	**L**
m	/m/	/m/	**M**
n	/n/	/n/	**N**
o	/o/	/ow/	**O**
p	/p/	/p/	**P**

Tabel 3.1 Hasil Analisis Translasi Fonem ke Simbol UPS (Lanjutan)

Huruf Alfabet	Fonem Indonesia	Translasi Fonem Dalam ARPAbet	Translasi Fonem Dalam UPS
q	/k/	/k/	**K**
r	/r/	/r/	**R**
s	/s/	/s/	**S**
t	/t/	/t/	**T**
u	/u/	/uh/	**UH**
v	/f/	/f/	**V**
w	/w/	/w/	**W**
x	/k s/	/k s/	**K + S**
y	/y/	/y/	**J**
z	/z/	/z/	**ZH**
ai	/ai/	/ah ih/	**AH + IH**
au	/au/	/ah uh/	**AH + UH**
oi	/oi/	/ow ih/	**O + IH**
ei	/ei/	/eh ih/	**EH + IH**
kh	/kh/	/k hh/	**K + H**
ng	/ng/	/n g/	**NG**
ny	/ny/	/n y/	**N + J**
sy	/sy/	/s y/	**S + J**

Dengan penjelasan dari hasil analisis tersebut adalah sebagai berikut.

- Kolom “Huruf Alfabet” adalah semua kemungkinan alfabet penyusun suatu kata dalam Bahasa Indonesia, dengan penambahan diftong informal “ei”.
- Kolom “Fonem Indonesia” adalah representasi fonem dari alfabet tersebut. Isi kolom ini didapatkan dengan menggabungkan penelitian Ferdiansyah (2012) dan Lestari (2010). Khusus untuk alfabet x, penulis membuat pendekatan dengan menggabungkan huruf fonem k dan s.
- Kolom “Translasi Fonem Dalam ARPAbet” adalah hasil translasi fonem dari kolom fonem dari penelitian Ferdiansyah (2012). Untuk alfabet diftong dan gabungan konsonan (“au”, “oi”, dan “ng”), penulis membuat pendekatan dengan menggabungkan fonem masing – masing alfabet penyusunnya.
- Kolom “Translasi Fonem Dalam UPS” adalah hasil translasi fonem dengan memerhatikan kedua kolom sebelumnya. Apabila terdapat kesamaan antara simbol UPS dengan simbol ARPAbet, maka simbol UPS akan mengikuti simbol ARPAbet. Untuk simbol – simbol UPS yang tidak sama dengan ARPAbet, penulis menggunakan simbol UPS yang paling mendekati (misalnya /h/ menjadi H, /ow/ menjadi O, dan sebagainya). Untuk diftong dan gabungan konsonan (kecuali /ng/), penulis membuat pendekatan dengan menggunakan simbol UPS masing – masing huruf penyusunnya dan menambahkan tanda ‘+’ sebagai penanda bunyi komposit, sesuai dengan aturan pemberian simbol UPS pada *engine*.

Berikut adalah contoh *phonetic transcription* beberapa kata berbahasa Indonesia dalam simbol UPS dengan mengacu pada tabel hasil analisis penulis (tabel 3.1). Contoh kata – kata berbahasa Indonesia diambil dari tabel 2.7.

Tabel 3.2 Contoh *Phonetic Transcription* Kata Berbahasa Indonesia Dalam Simbol UPS

Kata Bahasa Indonesia	**Bentuk Representasi Fonem**	***Phonetic Transcription* Dalam UPS**
ABSOLUTISME	/a b s o l u t i s m E/	AH B S O L UH T IH S M AX
AKHMAD	/a kh m a d/	AH K + H M AH D
ANEKA	/a n e k a/	AH N EH K AH
ANTARANYA	/a n t a r a n y a/	AH N T AH R AH N + J A
BERLANTAI	/b E r l a n t a i/	B AX R L AH N T AH + IH
BERMASYARAKAT	/b E r m a sy a r a k a t/	B AX R M AH S + J AH R AH K AH T

2. Pemenggalan Suku Kata

Engine recognizer yang digunakan dalam penelitian ini mendukung pemenggalan suku kata dalam representasi pelafalannya, dengan menggunakan tanda '.' (titik) untuk menandakan pemisahan antar suku katanya. Karena bahasa utama dalam pokok penelitian ini adalah Bahasa Indonesia, maka pemenggalan antar suku katanya pun akan menggunakan aturan pemenggalan suku kata Bahasa Indonesia. Aturan pemenggalan suku kata yang menjadi acuan di sini adalah aturan pemenggalan suku kata

terhadap kata dasar, dikarenakan beberapa keterbatasan terhadap aturan lainnya sebagai berikut.

a. Untuk dapat melakukan pemenggalan suku kata pada kata – kata berimbuhan, diperlukan suatu algoritma khusus untuk mengetahui akar kata / kata dasarnya.
b. Untuk kata yang terdiri dari dua unsur atau lebih juga memiliki kasus yang sama dengan di atas, karena umumnya kata – kata tersebut dituliskan secara tersambung, sehingga tidak diketahui kata – kata yang menjadi unsur penyusunnya.

Dengan mengacu pada poin (1) dan (2), maka dilakukanlah perancangan *Finite Automata* yang dapat digunakan untuk memaksimalkan *lexicon* melalui *phonetic transcription*. Pendekatan yang digunakan adalah DFA (*Deterministic Finite Automata*), dengan deskripsi analisisnya adalah sebagai berikut.

1. Simbol – simbol inputnya adalah semua kemungkinan alfabet penyusun huruf suatu kata. Karena gabungan konsonan dan diftong dalam aturan pemenggalan suku kata tidak pernah dipisahkan, maka gabungan konsonan dan diftong dapat diperlakukan seperti satu huruf, dengan diftong adalah huruf vokal, dan gabungan konsonan adalah konsonan. Pada awal prosesnya, diftong dan gabungan konsonan diubah menjadi suatu angka sebagai berikut.

   ai → 1 au → 2 oi → 3 ei → 4

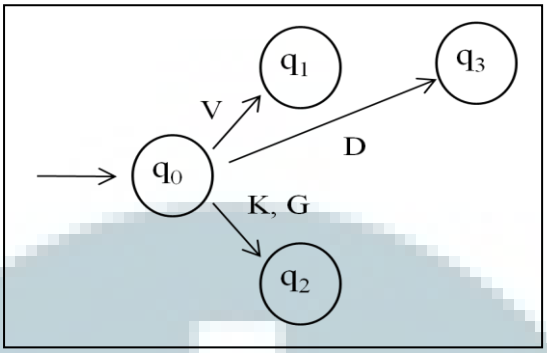
   kh → 5 ng → 6 ny → 7 sy → 8

2. Karena setiap alfabet bisa saja menjadi akhir dari suatu kata, maka dilakukan penambahan karakter ‘\0’ sebagai penanda akhir suatu kata.

3. Untuk memudahkan pembuatan DFA, maka simbol – simbol input digolongkan menjadi empat kategori, yaitu:
   - Vokal (V) → a, i, u, e, E, o
   - Konsonan (K) → b, c, d, f, g, h, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
   - Diftong (D) → ai, au, oi, ei
   - Gabungan Konsonan (G) → kh, ng, ny, sy
   - Tanda *Stop* (F) → \0
4. Diagram DFA yang dirancang tidak hanya berfungsi untuk menerima *string* kata hingga akhir, tetapi juga memiliki fungsi untuk mengembalikan hasil *phonetic transcription* dengan pemenggalan suku katanya. Fonem yang digunakan untuk menyusun *phonetic transcription* mengacu pada tabel 3.1.

Berikut adalah tahapan – tahapan analisis yang dilakukan oleh penulis dalam membuat DFA untuk *phonetic transcription*.

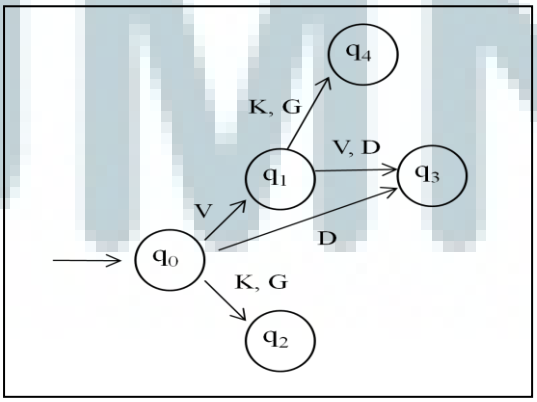
1. Simbol input pertama yang muncul pada kata dapat berupa salah satu dari vokal, konsonan, diftong, ataupun gabungan konsonan. Karena terdapat aturan bahwa tidak diperbolehkan adanya satu huruf vokal saja pada awal pemenggalan, maka dibuat dua *state* berbeda untuk mengakomodasi kemungkinan input berupa vokal dan diftong.



Gambar 3.3 Perancangan DFA Untuk *Phonetic Transcription* (1)

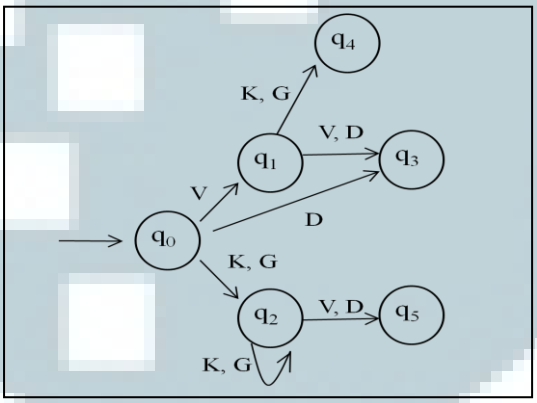
Analisis pada tahap – tahap berikutnya akan mengacu pada contoh rangkaian input pada *state* tersebut, disesuaikan dengan aturan pemenggalan suku kata dan simbol input selanjutnya.

2. Pada *state* q_1, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut.
   - Untuk vokal ataupun diftong, transisi akan mengarah pada *state* q_3, dikarenakan pada dasarnya diftong adalah dua vokal yang tidak dipisahkan.
   - Untuk konsonan ataupun gabungan konsonan, transisi akan mengarah pada *state* baru sebagai analisis awal (q_4).



Gambar 3.4 Perancangan DFA Untuk *Phonetic Transcription* (2)

3. Pada *state* q_2, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut.
   a. Untuk vokal ataupun diftong, transisi akan mengarah pada *state* baru sebagai analisis awal (q_5).
   b. Untuk konsonan ataupun gabungan konsonan, transisi akan mengarah kembali ke q_2, karena dalam *state* ini belum ada bunyi yang dihasilkan (dari vokal ataupun diftong), sehingga pemenggalan suku kata tidak bisa dilakukan.

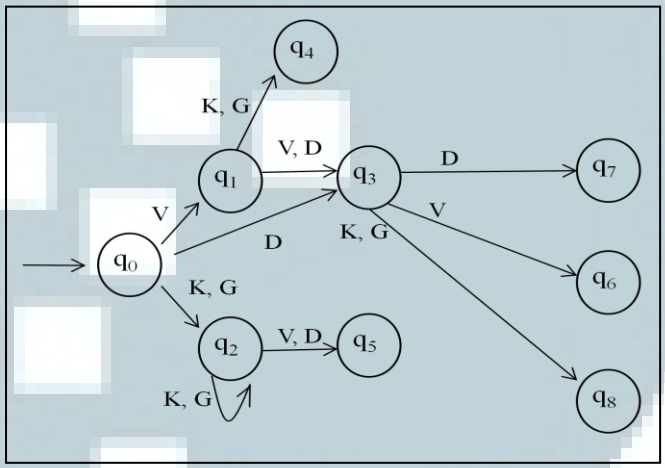


Gambar 3.5 Perancangan DFA Untuk *Phonetic Transcription* (3)

4. Pada *state* q_3, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "au").
   a. Untuk vokal, transisi akan mengarah kepada *state* baru (q_6) dimana dalam *state* ini akan dilakukan pemenggalan, sesuai dengan aturan pemenggalan kata poin 1a (misalkan "au-a").
   b. Untuk diftong, transisi akan mengarah kepada *state* yang berbeda dengan di atas (q_7) dimana dalam *state* ini juga akan dilakukan pemenggalan.

Pembedaan *state* mengacu pada aturan pemenggalan kata yang tidak memperbolehkan adanya satu huruf vokal di akhir kata.

c. Untuk konsonan dan gabungan konsonan, transisi akan mengarah kepada *state* baru sebagai analisis awal (q_8). *State* ini merupakan penanda suatu konsonan atau gabungan konsonan yang telah diawali oleh suatu vokal atau diftong.

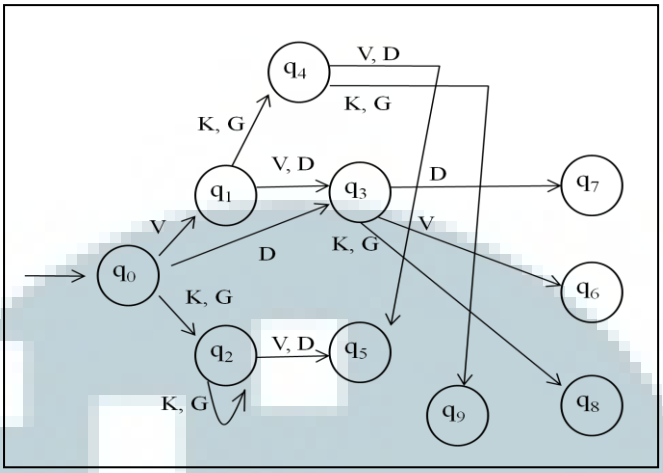


Gambar 3.6 Perancangan DFA Untuk *Phonetic Transcription* (4)

5. Pada *state* q_4, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "ak").

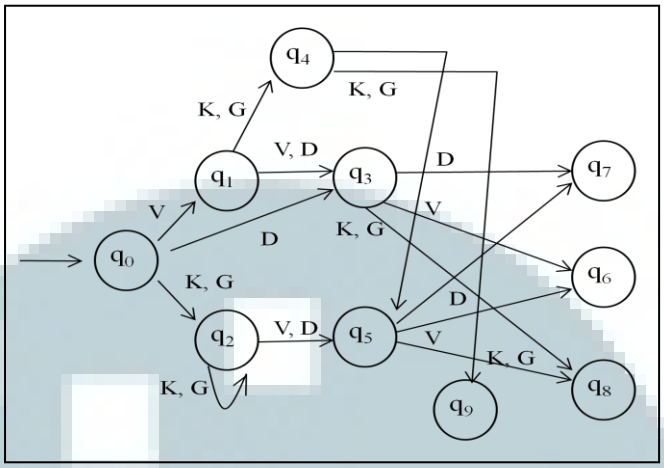
   a. Untuk vokal dan diftong tidak ada pemenggalan (misalnya "ak" dan "u" menjadi "aku", bukan "a-ku"). Hal ini serupa dengan yang terjadi pada *state* q_5, sehingga transisi akan mengarah pada *state* ini.

   b. Untuk konsonan dan gabungan konsonan, transisi akan mengarah pada *state* baru (q_9) dimana dalam *state* ini akan dilakukan pemenggalan karena terdapat dua konsonan berurutan.



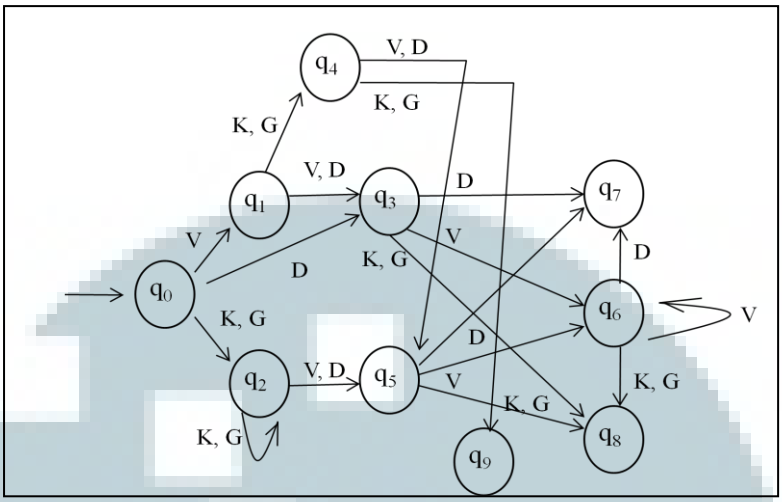
Gambar 3.7 Perancangan DFA Untuk *Phonetic Transcription* (5)

6. Pada *state* q_5, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "ki").
   a. Untuk vokal, maka akan dilakukan pemenggalan karena terdapat dua vokal berurutan (misalnya "ki-a", "ki-e", dan seterusnya). Maka transisi akan mengarah pada *state* q_6.
   b. Untuk diftong, maka juga akan dilakukan pemenggalan karena vokal diikuti diftong dapat diperlakukan sebagai vokal berurutan (misalnya "ki-au" dan seterusnya). Maka transisi akan mengarah pada *state* q_7.
   c. Untuk konsonan dan gabungan konsonan, transisi akan mengarah pada *state* dimana suatu konsonan dan gabungan konsonan telah diikuti oleh huruf vokal atau diftong, yaitu *state* q_8.



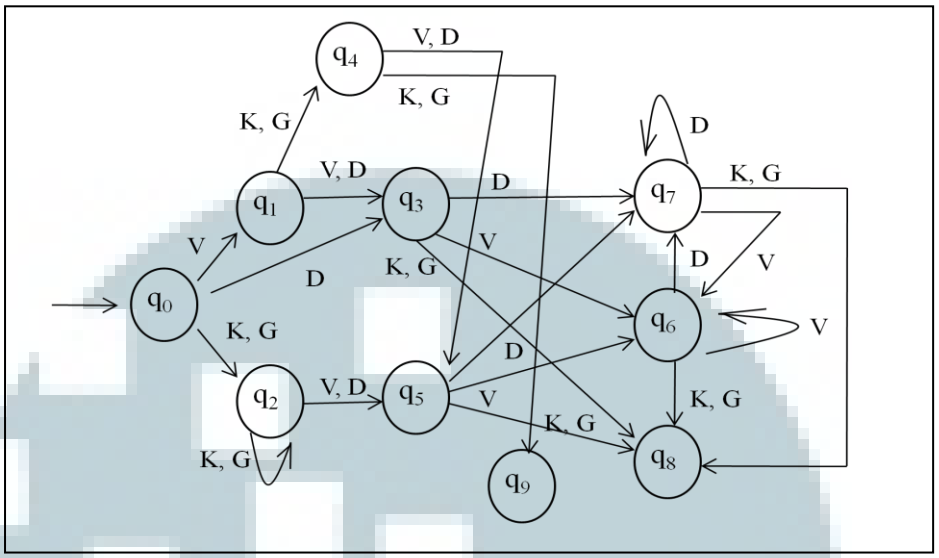
Gambar 3.8 Perancangan DFA Untuk *Phonetic Transcription* (6)

7. Pada *state* q_6, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "ki-a").
   a. Untuk vokal, akan dilakukan pemenggalan karena terjadinya vokal berurutan (misalnya "ki-a-e"). Maka transisi akan mengarah kembali ke *state* q_6.
   b. Untuk diftong, akan dilakukan pemenggalan karena terjadinya vokal berurutan (misalnya "ki-a-au"). Maka transisi akan mengarah ke *state* q_7.
   c. Untuk konsonan dan gabungan konsonan, transisi akan mengarah pada *state* dimana konsonan atau gabungan konsonan diikuti oleh vokal ataupun diftong, yaitu *state* q_8.



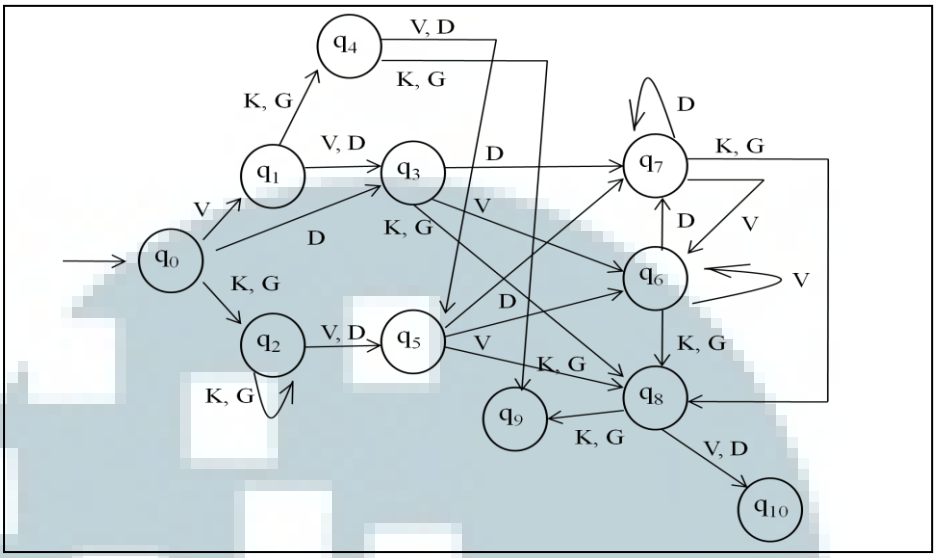
Gambar 3.9 Perancangan DFA Untuk *Phonetic Transcription* (7)

8. Pada *state* q_7, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "au-au").
   a. Untuk vokal, akan dilakukan pemenggalan karena terdapat vokal berurutan. Maka transisi akan mengarah pada *state* q_6.
   b. Untuk diftong, akan dilakukan pemenggalan karena terdapat vokal berurutan. Maka transisi akan mengarah kembali pada *state* q_7.
   c. Untuk konsonan dan gabungan konsonan, transisi akan mengarah pada *state* dimana konsonan dan gabungan konsonan diawali oleh vokal ataupun diftong, yaitu *state* q_8.



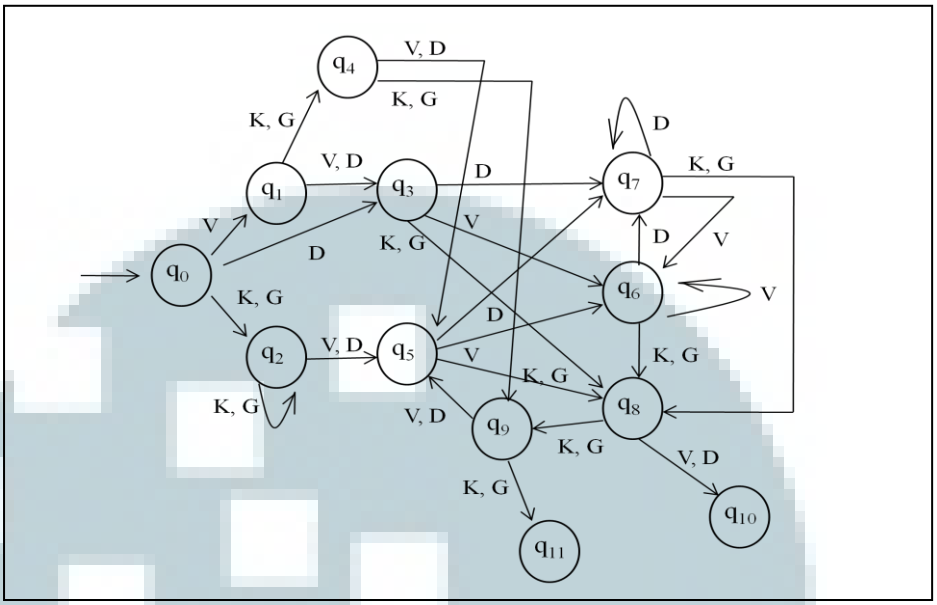
Gambar 3.10 Perancangan DFA Untuk *Phonetic Transcription* (8)

9. Pada *state* q_8, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "kit")
   a. Untuk vokal dan diftong, maka akan dilakukan pemenggalan pada konsonan atau gabungan konsonan tersebut (misalkan "kit" dan 'a', maka menjadi "ki-ta"). Transisi akan mengarah pada *state* baru (q_{10}).
   b. Untuk konsonan dan gabungan konsonan, akan dilakukan pemenggalan karena terdapat konsonan berurutan. Maka transisi akan mengarah pada *state* q_9.



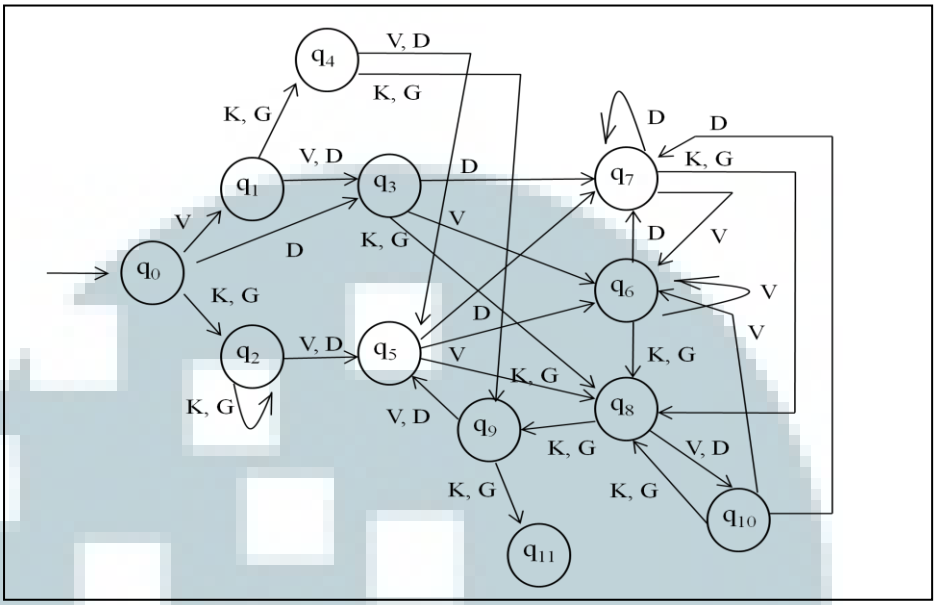
Gambar 3.11 Perancangan DFA Untuk *Phonetic Transcription* (9)

10. Pada *state* q_9, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "bak-s")
    a. Untuk vokal dan diftong, maka transisi akan mengarah pada *state* dimana vokal diawali oleh konsonan ataupun gabungan konsonan, yaitu *state* q_5.
    b. Untuk konsonan dan diftong, berdasarkan pada aturan pemenggalan kata poin 1e, maka dapat disimpulkan tidak akan terjadi pemenggalan selama tidak adanya vokal atau diftong yang melambangkan bunyi. Karena itu, transisi akan mengarah pada *state* baru yang mengakomodasi perulangan konsonan sampai dengan terdapat input berupa vokal atau diftong (q_{11}).



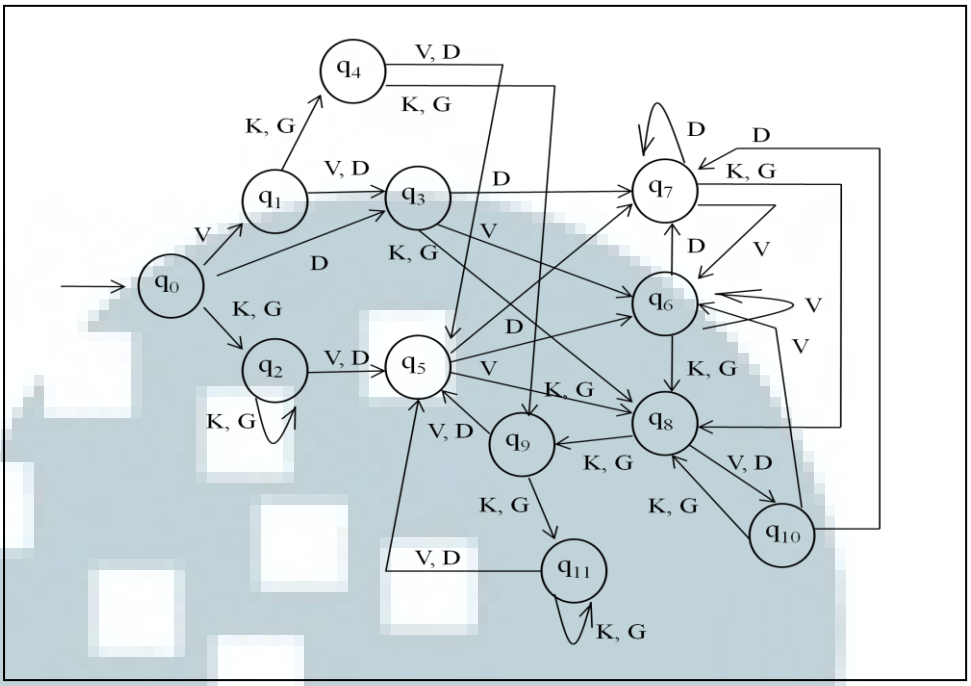
Gambar 3.12 Perancangan DFA Untuk *Phonetic Transcription* (10)

11. Pada *state* q_{10}, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "ki-ta")
    c. Untuk vokal, akan dilakukan pemenggalan karena terdapat dua vokal berurutan. Maka transisi akan mengarah pada *state* q_6.
    d. Untuk diftong, akan dilakukan pemenggalan karena terdapat dua vokal berurutan. Maka transisi akan mengarah pada *state* q_7.
    e. Untuk konsonan dan gabungan konsonan, transisi akan mengarah pada *state* dimana suatu konsonan atau gabungan konsonan diawali oleh suatu vokal atau diftong, yaitu *state* q_8.



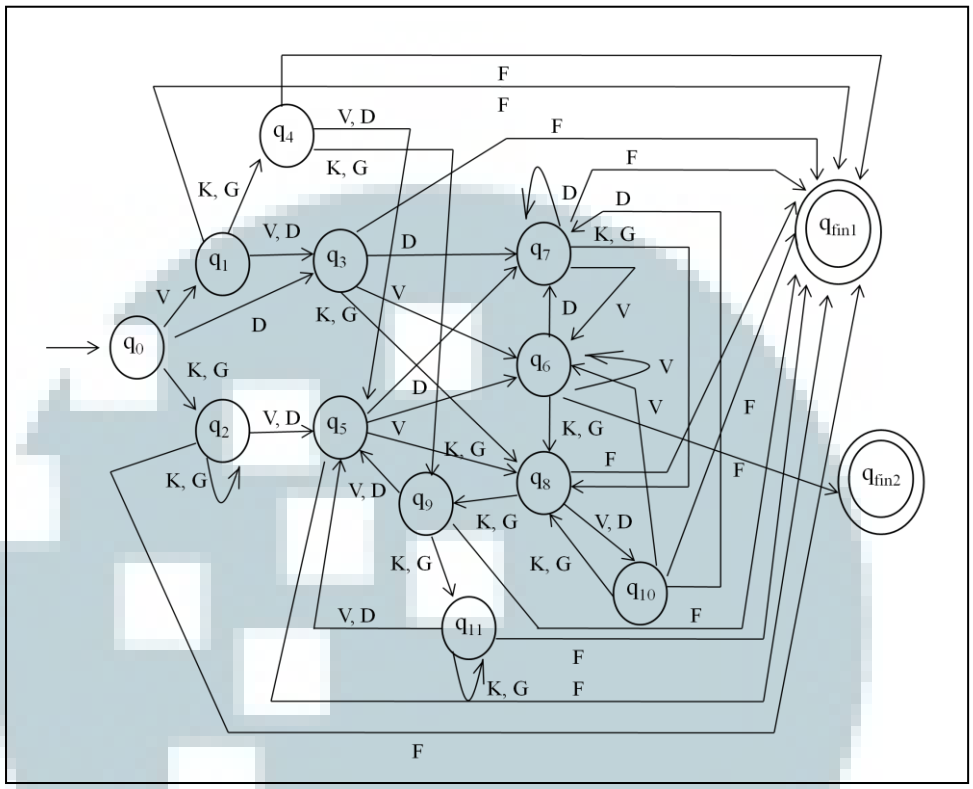
Gambar 3.13 Perancangan DFA Untuk *Phonetic Transcription* (11)

12. Pada *state* q_{11}, hasil analisis terhadap simbol input selanjutnya adalah sebagai berikut (misalkan rangkaian input adalah "ins-t")
    a. Untuk vokal dan diftong, transisi akan mengarah pada *state* dimana suatu vokal atau diftong diawali oleh konsonan atau gabungan konsonan, yaitu *state* q_5.
    b. Untuk konsonan dan gabungan konsonan, karena pada *state* ini pemenggalan tidak dilakukan selama tidak terdapat bunyi (diikuti oleh vokal atau diftong). Maka, transisi akan mengarah kembali pada q_{11}.



Gambar 3.14 Perancangan DFA Untuk *Phonetic Transcription* (12)

13. Karena masing – masing alfabet penyusun bisa saja menjadi akhir dari suatu kata (ditandai dengan adanya simbol input '\0'), maka setiap *state* (kecuali *state* q_0) memiliki satu simbol input lagi, yaitu F (penanda karakter '\0') yang mengarah kepada *state* akhir dengan ketentuan sebagai berikut.
    a. *State* akhir (q_{fin1}) adalah *state* yang berfungsi sebagai penanda penerima kata
    b. Khusus untuk *state* q_6, karena pada *state* ini huruf vokal berada paling akhir (misalnya "di-a"), maka *state* akhirnya dipisahkan terhadap *state* q_{fin} (menjadi q_{fin2}). Pada *state* ini dilakukan penggabungan kembali vokal terakhir yang sudah dipisahkan ("di-a" menjadi "dia", dan seterusnya).



Gambar 3.15 Hasil Akhir Perancangan DFA Untuk *Phonetic Transcription*

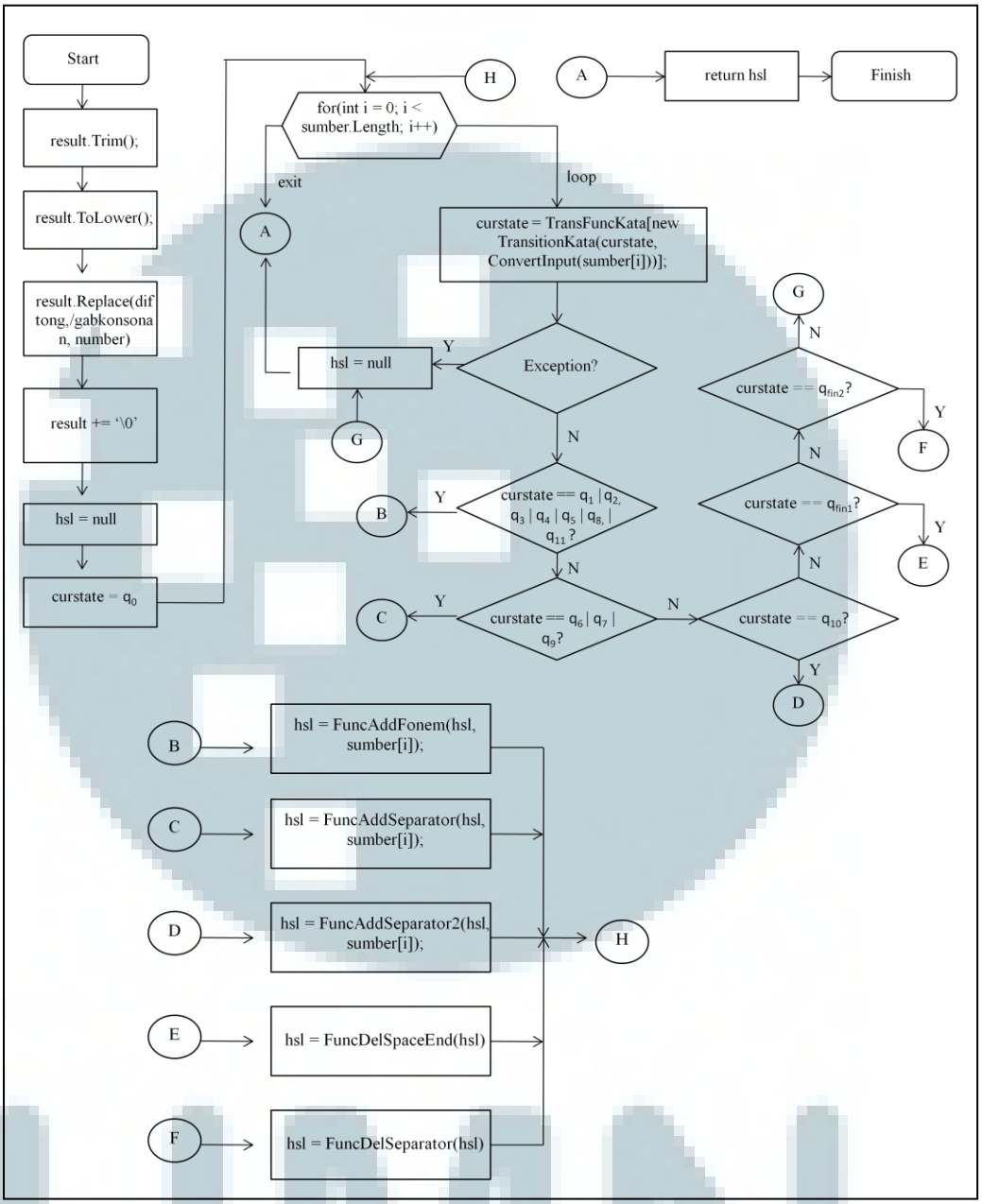
14. Masing – masing *state* (kecuali *state* q_0) akan menjalankan fungsi – fungsi yang akan menyusun *phonetic transcription* kata yang bersangkutan.
    a. q_1, q_2, q_3, q_4, q_5, q_8, q_{11} berfungsi untuk mengembalikan fonem dari simbol inputnya ditambahkan dengan karakter spasi (misal ‘a’ menjadi “AH “, ‘b’ menjadi “B “, dan seterusnya).
    b. q_6, q_7, q_9 berfungsi untuk mengembalikan separator “.” sebagai penanda pemenggalan suku kata, fonem dari simbol inputnya, dan ditambahkan dengan karakter spasi (misal ‘a’ menjadi “. AH “, ‘b’ menjadi “. B “, dan seterusnya).
    c. q_{10} berfungsi untuk menambahkan separator “.” sebelum huruf konsonan atau gabungan konsonan paling terakhir, fonem dari simbol input, dan

karakter spasi (misalnya pada kata “baca” diubah menjadi “B AH C”, lalu “B AH . C AH ”).

d. q_{fin1} berfungsi untuk menghapus spasi di awal dan akhir *phonetic transcription* yang didapatkan (misal “B AH . C AH “ menjadi “B AH . C AH”).

e. q_{fin2} berfungsi untuk menghapus separator paling terakhir dan spasi di awal dan akhir hasil *phonetic transcription* (misalnya “D IH . AH “ menjadi “D IH AH”).

Gambaran proses keseluruhan dalam mengubah suatu kata ke dalam representasi *phonetic transcription* nya diperlihatkan dalam *flowchart* sebagai berikut.



Gambar 3.16 *Flowchart* Proses Mendapatkan *Phonetic Transcription*

Flowchart tersebut menggambarkan proses pembuatan *phonetic transcription* dari suatu kata. Suatu kata akan mengalami serangkaian proses seperti penghilangan spasi di awal dan di akhir kata, pengubahan huruf penyusun kata menjadi huruf kecil (kecuali huruf 'E' sebagai pembeda pelafalan),

mengubah diftong dan gabungan konsonan ke dalam angka, lalu menambahkan karakter '\0' sebagai penanda akhir kata. *Phonetic transcription* akan diinisialisasi ke nilai *null* sebagai nilai awal, dan *state* DFA diinisialisasi ke *state* awal (q_0). Kemudian akan dilakukan pembacaan huruf – huruf penyusun secara berurutan sesuai dengan DFA yang telah dirancang, lalu menjalankan fungsi – fungsi tertentu sesuai dengan *state* hasil transisi suatu huruf. Apabila terdapat suatu huruf yang tidak menyebabkan transisi, maka akan dianggap pembacaan kata oleh DFA gagal dan mengembalikan *phonetic transcription* dengan nilai *null*. Apabila pembacaan dapat berlangsung hingga akhir kata (ditandai dengan karakter '\0'), maka hasil *phonetic transcription* akan didapatkan.

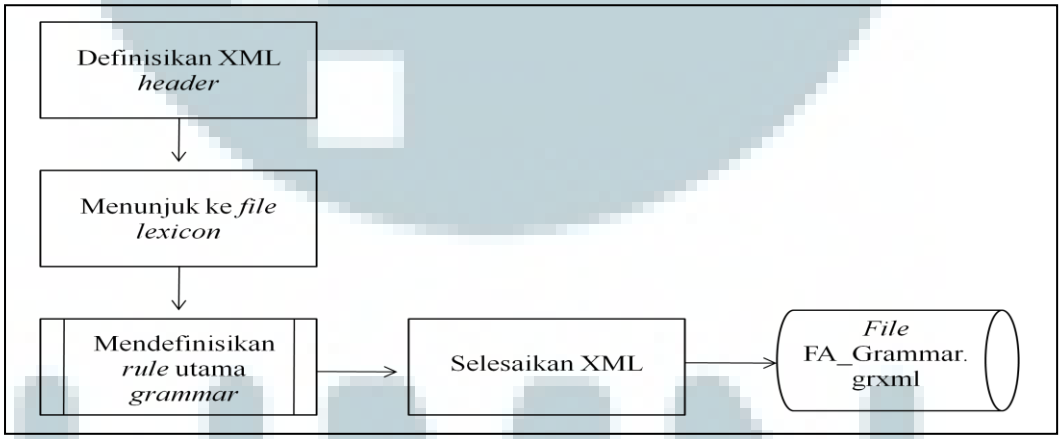
3.1.2 *Finite Automata* Pada *Language Model* (*Grammar*)

Language Model pada *engine* ini menggunakan suatu *file grammar* berformat XML dan memiliki ekstensi .grxml. *File grammar* ini nanti akan di*load* ke dalam *engine* dan akan dijadikan patokan dalam mengenali pola kata – kata yang akan diucapkan oleh pengguna, sehingga perancangannya akan disesuaikan dengan penggunaan aplikasi ASR yang akan dibangun. Karena penggunaan aplikasi ASR akan ditujukan untuk pendiktean kata – kata berbahasa Indonesia, maka terdapat beberapa analisis oleh penulis dalam merancang *grammar*.

1. Mengacu pada gambar 2.2, terlihat bahwa representasi *grammar* menyerupai suatu *Finite Automata*, dimana terdapat *state* awal dan akhir, simbol input yang direpresentasikan dengan kemungkinan kata yang diucapkan oleh pengguna, serta transisi antar *state* karena simbol input tersebut.

2. Agar dapat digunakan untuk pendiktean dengan arti apapun yang diucapkan oleh pengguna dapat dikenali oleh aplikasi (dengan catatan kata – kata tersebut bukanlah kata – kata OOV), maka dapat disimpulkan simbol input adalah semua alternatif kata yang terdapat pada *lexicon*. Hal ini mengacu pada peran *lexicon* sebagai pemuat semua daftar kata yang dapat dikenali oleh aplikasi.
3. Pendiktean kata – kata ditujukan agar pengguna dapat melakukan pendiktean lebih dari satu kata. Dengan demikian, pendekatan yang dilakukan dalam merancang *grammar* adalah jumlah kata yang mungkin diucapkan.

Berikut adalah *system flow* pembuatan *grammar* utama yang akan digunakan dalam aplikasi ASR penelitian ini.

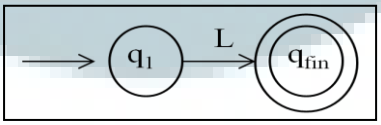


Gambar 3.17 *System Flow* Pembuatan *File Grammar* Utama

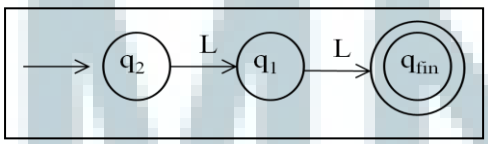
Proses pembuatan *file grammar* hampir sama dengan proses pembuatan *file lexicon*. Diawali dengan pendefinisian *header* XML, lalu pendefinisian *rule* utama dalam *grammar* melalui langkah – langkah *Finite Automata* sebagai penentu rangkaian kata yang akan diucapkan oleh pengguna. Penulisan dalam format XML kemudian akan diselesaikan, kemudian *file grammar* akan dibuat

dengan nama FA_Grammar.grxml. *File grammar* inilah yang akan di*load* ke dalam aplikasi sebagai patokan pengenalan kata. Nama FA_Grammar digunakan sebagai penunjuk bahwa *grammar* tersebut dirancang dengan menggunakan konsep *Finite Automata*. Berikut adalah hasil analisis penulis dalam menerapkan *Finite Automata* untuk perancangan *grammar*.

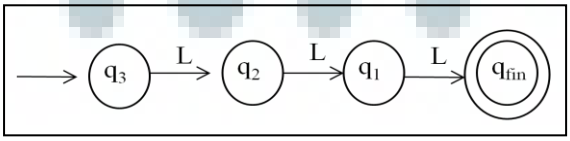
1. Karena pendekatan yang digunakan adalah jumlah kata yang mungkin diucapkan oleh pengguna, maka rancangan *Finite Automata* menggunakan pendekatan DFA. Hal ini dikarenakan jumlah kata yang diucapkan sudah pasti (misalkan satu kata, dua kata, tiga kata, dan seterusnya). Dengan simbol input adalah salah satu kata dari semua alternatif kata dalam *lexicon* (yang akan direpresentasikan dengan L), maka perancangan DFAnya adalah sebagai berikut. (dimisalkan nama *state* awal mengacu pada jumlah kata yang akan dikenali)



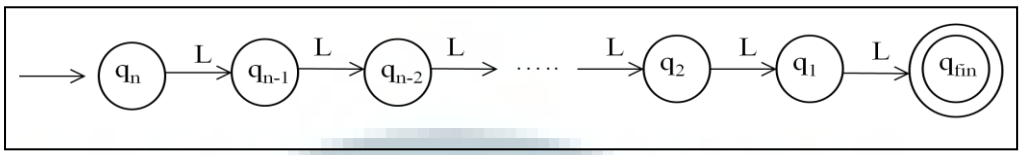
Gambar 3.18 Diagram DFA Untuk Menerima Satu Kata



Gambar 3.19 Diagram DFA Untuk Menerima Dua Kata

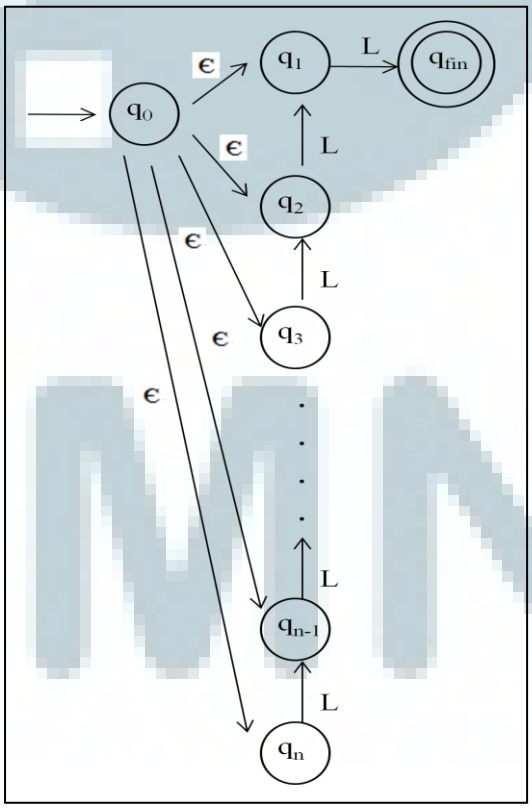


Gambar 3.20 Diagram DFA Untuk Menerima Tiga Kata



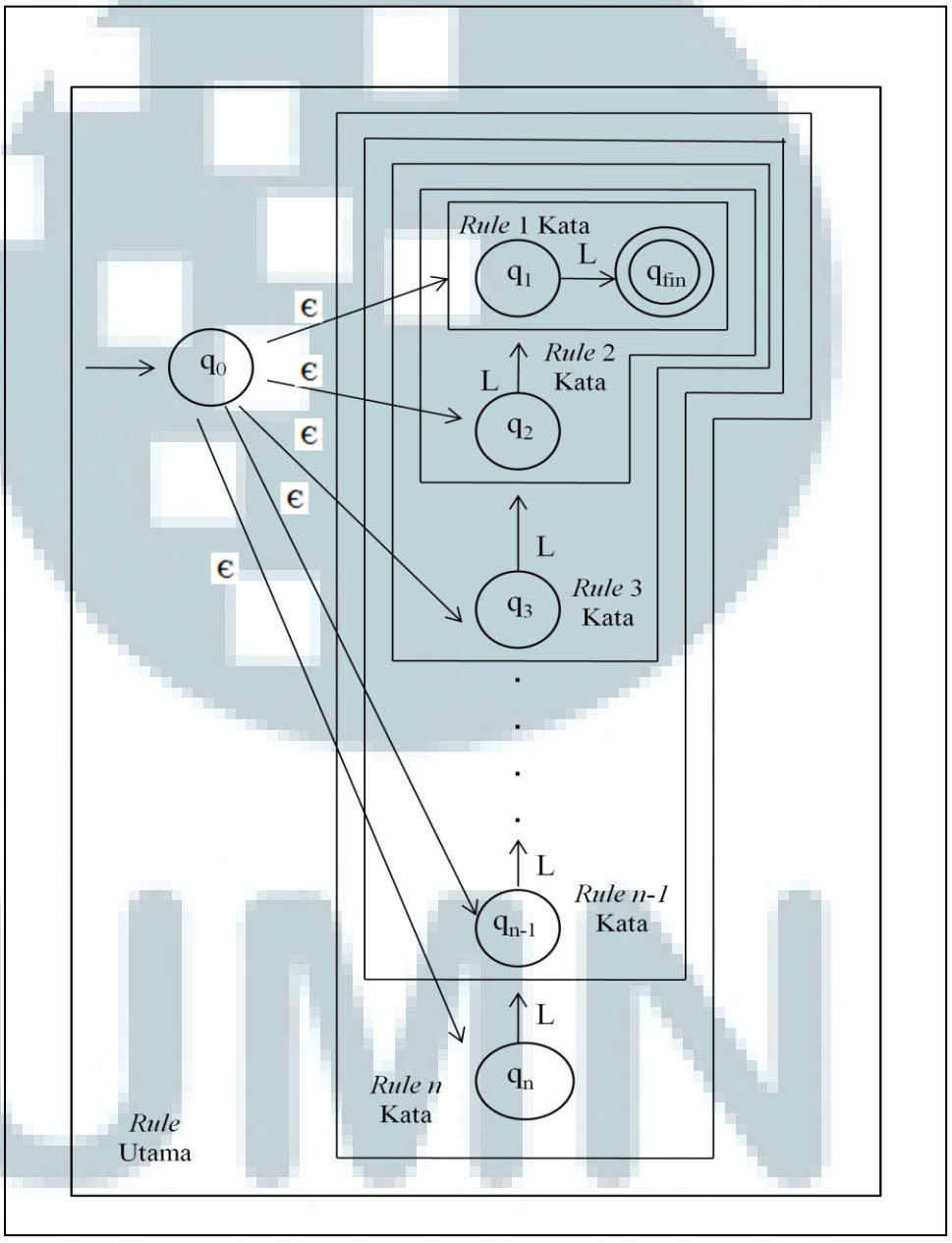
Gambar 3.21 Diagram DFA Untuk Menerima *n* Kata

2. Dalam prakteknya, tentu tidak dapat ditentukan jumlah kata pasti yang akan diucapkan oleh pengguna (bisa saja satu kata, dua kata, tiga kata, sampai dengan *n* kata). Dengan demikian, pendekatan yang dapat dilakukan adalah dengan menerapkan NFA. Untuk mempermudah representasi, maka digunakanlah transisi *epsilon* sebagai penanda simbol input pertama sehingga hasil analisis perancangan NFA (dengan *epsilon* atau sebagai ϵ - NFA) yang didapatkan adalah sebagai berikut.



Gambar 3.22 Diagram ϵ - NFA Untuk Menerima Satu Sampai Dengan *n* Kata

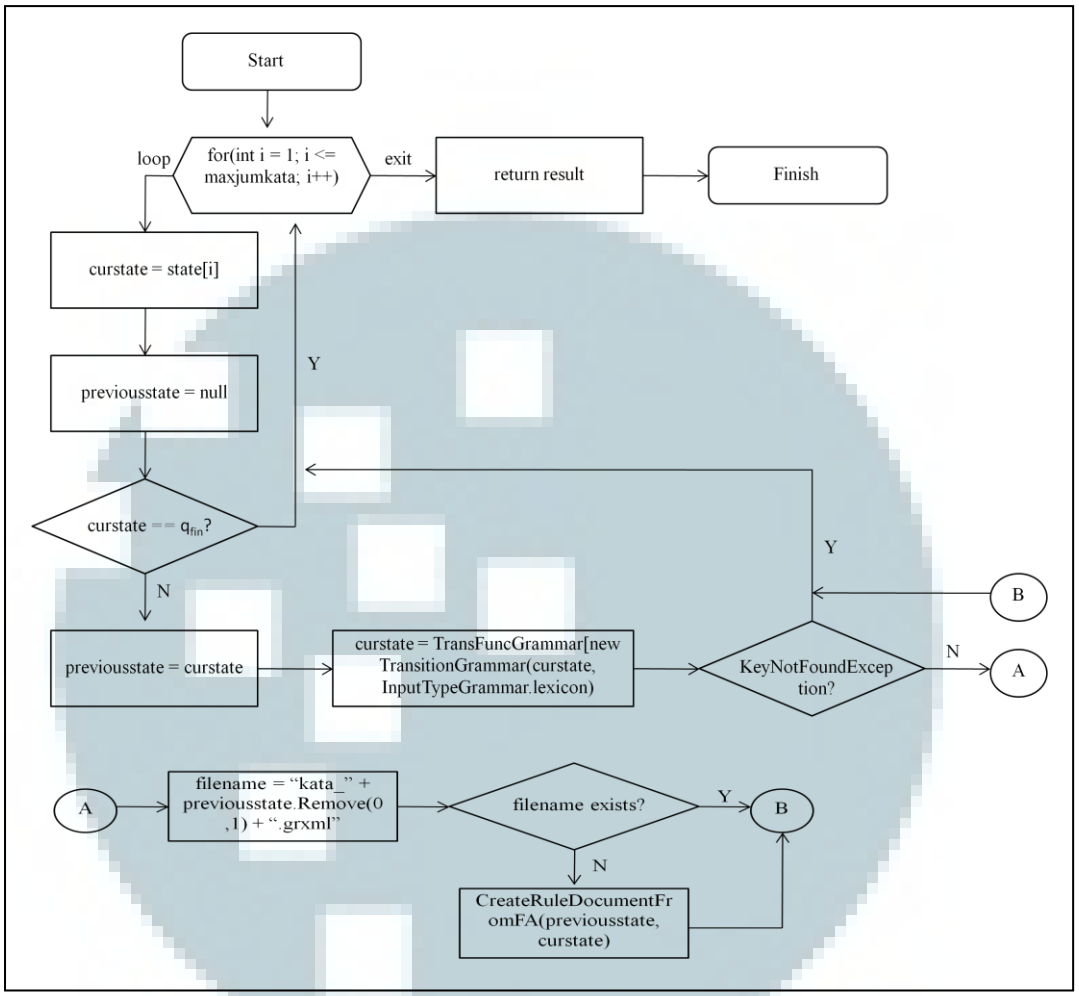
3. Mengacu pada aturan penulisan *grammar*, dimana rangkaian suatu *state* awal dengan beberapa *state* lainnya sampai dengan *state* akhir dapat dinyatakan dengan suatu *rule*, maka representasi diagram ϵ - NFA tersebut dapat dinyatakan sebagai suatu kumpulan *rule* seperti berikut.



Gambar 3.23 Pengelompokkan *Rule* Pada *Grammar* Berdasarkan Diagram ϵ - NFA

4. Dengan melihat pola diagram ϵ - NFA dengan pengelompokkan *rule* tersebut, maka didapatkan beberapa analisis sebagai berikut.
   a. Pada awal hasil transisi (transisi *epsilon* dari q_0 ke masing – masing *state* berbeda, mulai dari q_1, q_2, q_3, dan seterusnya hingga q_n), akan menunjuk pada suatu *rule* baru, yang nantinya akan mendefinisikan jumlah kata yang dapat diucapkan dalam *rule* tersebut. Isi dari *rule* tersebut didapatkan dengan menganalisis transisi yang terjadi di dalam *rule*.
   b. Untuk setiap rangkaian transisi akibat L sampai dengan *state* akhir dari suatu *state* awal (misalnya q_2-q_1-q_{fin}, q_3-q_2-q_1-q_{fin}, sampai dengan q_n-q_{n-1}-…-q_3-q_2-q_1-q_{fin}), pada setiap transisinya akan menunjuk kepada *rule* berisikan jumlah kata yang telah dikurangi satu kata dari *rule* sekarang (misalnya *rule* tiga kata berisikan salah satu L dan *rule* dua kata, *rule* dua kata berisikan salah satu L dan *rule* satu kata). Hal tersebut berlaku kecuali untuk *state* q_1 yang hanya berisikan salah satu L.

Berikut adalah *flowchart* proses pembuatan aturan *rule* dengan berdasarkan diagram ϵ - NFA dan hasil analisis pada poin (4).



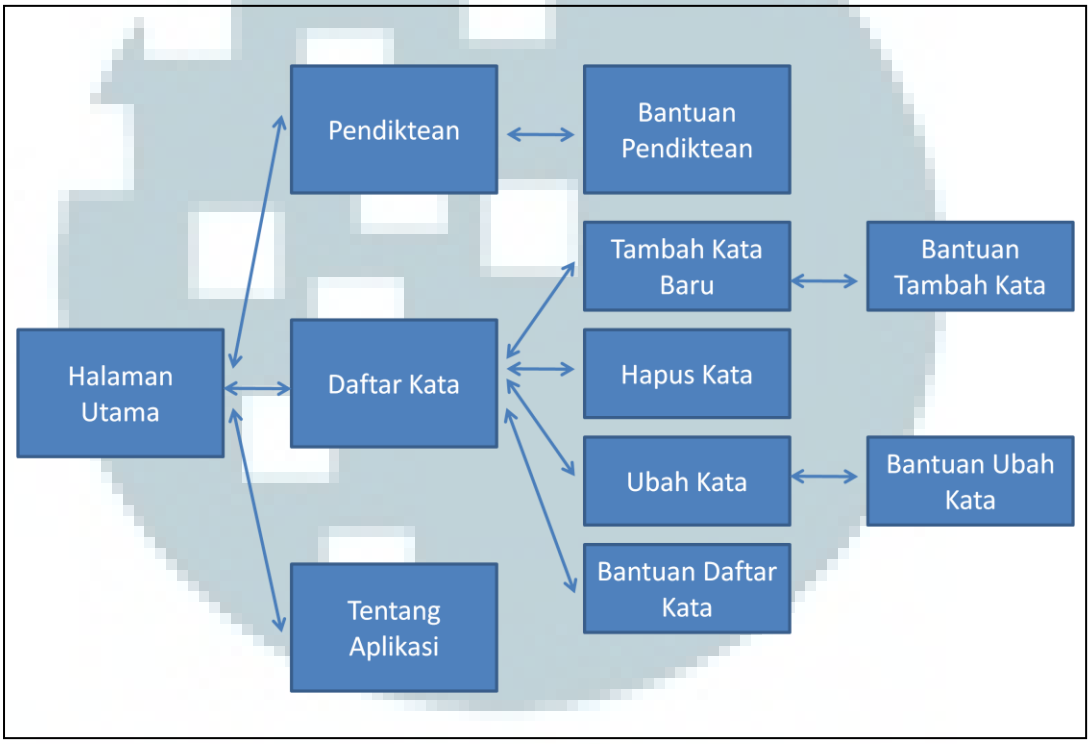
Gambar 3.24 *Flowchart* Pembuatan Aturan *Grammar* Berdasarkan Diagram ϵ - NFA

Inkremental variabel i dari nilai satu sampa i dengan variabel maxjumkata menggambarkan aturan *grammar* untuk mengakomodasi *rule – rule* yang berfungsi menerima satu sampai dengan *n* kata. Pada setiap inkremen akan dilakukan transisi NFA (menghasilkan curstate) dengan *state* sebelumnya disimpan sebagai previousstate. Apabila transisi berhasil, akan dilakukan pengecekan terlebih dahulu apakah *file grammar* untuk mengakomodasi jumlah kata sesuai dengan previousstate sudah dibuat. Apabila belum, suatu *file grammar* berisikan aturan penerima kata sebanyak previousstate akan dibuat.

3.2 Perancangan Aplikasi

3.2.1 Struktur Navigasi Menu

Berikut adalah struktur navigasi menu yang dapat dipilih oleh pengguna saat menjalankan aplikasi.



Gambar 3.25 Struktur Navigasi Menu Aplikasi

Pada halaman utama aplikasi (halaman pertama yang tampil saat pengguna menjalankan aplikasi) terdapat tiga menu utama dengan rincian masing – masing menu adalah sebagai berikut.

1. Pendiktean

   Pada menu ini, akan ditampilkan halaman yang memungkinkan pengguna untuk melakukan pendiktean kata – kata berbahasa Indonesia. Apabila kata – kata yang diucapkan oleh pengguna mampu dikenali oleh aplikasi, maka

representasi tekstual dari kata tersebut akan ditampilkan. Pada halaman ini terdapat satu submenu, yaitu Bantuan Pendiktean yang akan menampilkan halaman berisikan bantuan mengenai cara – cara melakukan pendiktean dengan benar.

2. Daftar Kata

   Pada menu ini, akan ditampilkan halaman yang berisikan seluruh daftar kata dalam aplikasi yang akan menjadi *lexicon* bagi aplikasi ASR. Pada halaman ini, terdapat dua submenu lain sebagai berikut.

   a. Submenu Tambah Kata Baru, dimana akan ditampilkan suatu halaman yang memungkinkan pengguna untuk menginput suatu kata berbahasa Indonesia baru ke dalam daftar kata aplikasi
   b. Submenu Hapus Kata, dimana pengguna dapat menghapus salah satu kata tertentu dari daftar kata yang ditampilkan. Penghapusan tersebut juga akan menyebabkan terjadinya perubahan terhadap daftar kata dalam aplikasi.

   Selain kedua submenu tersebut, terdapat satu submenu lainnya yaitu Bantuan Daftar Kata. Submenu ini akan menampilkan halaman berisikan bantuan mengenai cara – cara melihat daftar kata serta mekanisme penghapusan suatu kata tertentu dari daftar kata yang ditampilkan.

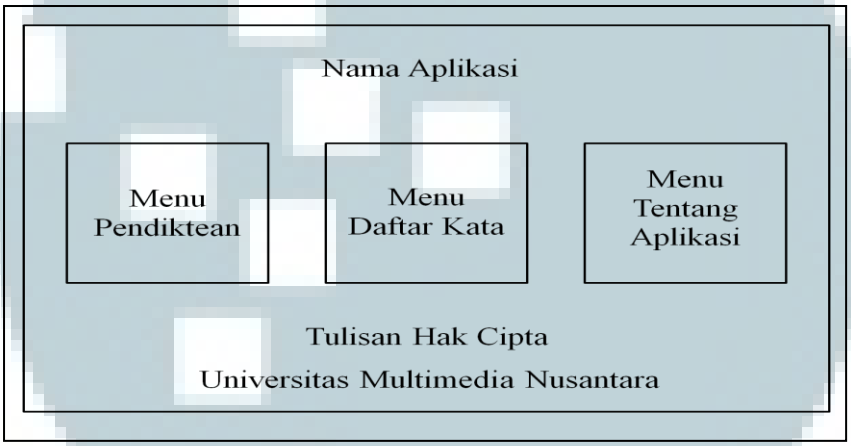
3. Tentang Aplikasi

   Menu ini akan menampilkan halaman berisikan keterangan / informasi singkat mengenai aplikasi. Informasi – informasi yang ditampilkan antara lain adalah kegunaan aplikasi, menu – menu utama dari aplikasi, serta informasi mengenai pembuat aplikasi.

3.2.2 Sketsa *Interface* Aplikasi

Sketsa *interface* aplikasi akan digunakan sebagai pedoman dalam membagun *interface* yang akan dilihat dan digunakan oleh pengguna untuk menggunakan aplikasi. Berikut adalah hasil perancangan sketsa *interface* aplikasi dalam penelitian ini.

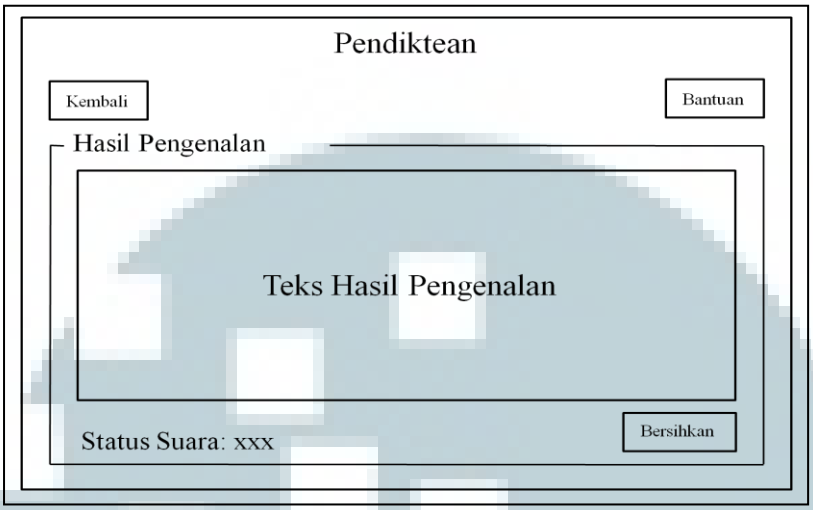
1. Halaman Utama



Gambar 3.26 Sketsa *Interface* Halaman Utama

Halaman utama adalah halaman yang pertama kali ditampilkan kepada pengguna saat memulai aplikasi. Pada halaman utama, terdapat tiga pilihan menu utama yang masing – masing dapat dipilih oleh pengguna, yaitu menu pendiktean, menu untuk melihat daftar kata, dan menu untuk melihat informasi tentang aplikasi. Di bagian bawah terdapat tulisan hak cipta aplikasi dan Universitas Multimedia Nusantara.

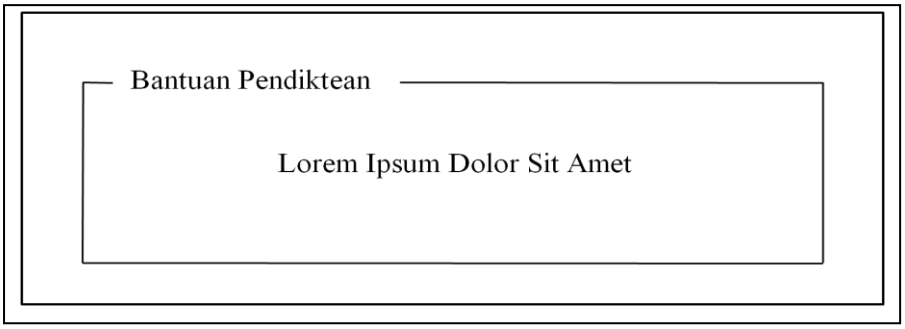
2. Halaman Pendiktean



Gambar 3.27 Sketsa *Interface* Halaman Pendiktean

Halaman ini adalah halaman yang ditampilkan ketika pengguna memilih menu pendiktean yang terdapat pada halaman utama. Pada halaman ini, pengguna dapat mendiktekan kata – kata dalam Bahasa Indonesia untuk kemudian ditampilkan representasi tekstualnya. Terdapat keterangan mengenai status suara yang dapat dijadikan petunjuk oleh pengguna dalam mengetahui apakah suaranya telah terdeteksi selama melakukan pendiktean. Pengguna juga dapat memilih pilihan lain seperti "Kembali" untuk kembali ke halaman utama, pilihan "Bersihkan" untuk mengosongkan kembali hasil pengenalan kata oleh aplikasi, dan pilihan Bantuan untuk melihat informasi mengenai cara – cara melakukan pendiktean di halaman ini.

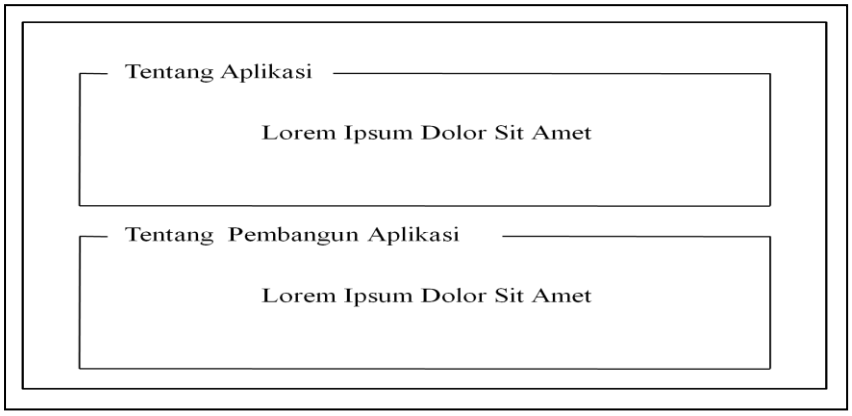
3. Halaman Bantuan Pendiktean



Gambar 3.28 Sketsa *Interface* Halaman Bantuan Pendiktean

Halaman ini adalah halaman yang ditampilkan ketika pengguna memilih menu "Bantuan" yang terdapat pada halaman Pendiktean. Pada halaman ini, akan ditampilkan informasi mengenai cara – cara melakukan pendiktean pada halaman Pendiktean.

4. Halaman Tentang Aplikasi

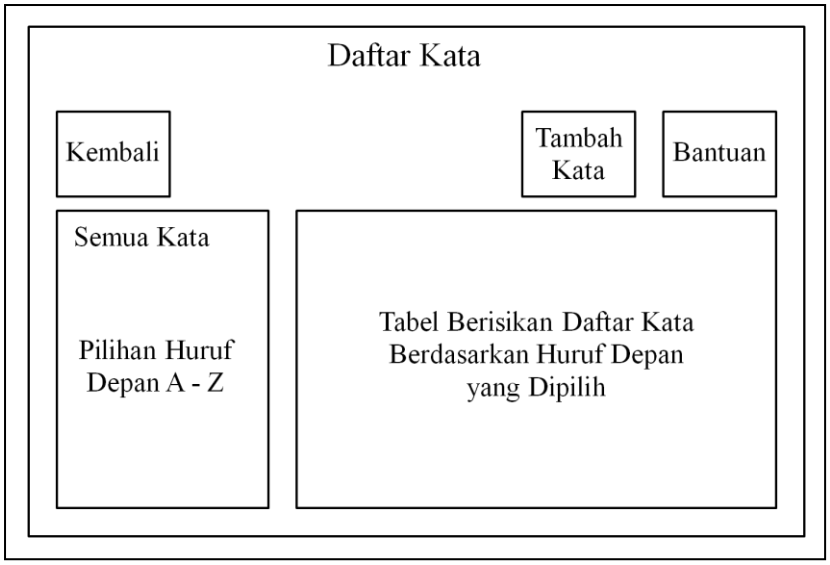


Gambar 3.29 Sketsa *Interface* Halaman Tentang Aplikasi

Halaman ini akan ditampilkan ketika pengguna memilih menu "Tentang Aplikasi" yang terdapat pada halaman utama. Halaman ini berisikan dua informasi utama, yaitu tentang aplikasi (berisikan cara menggunakan aplikasi

beserta menu – menunya) dan tentang pembangun aplikasi (yang dalam hal ini adalah penulis sendiri).

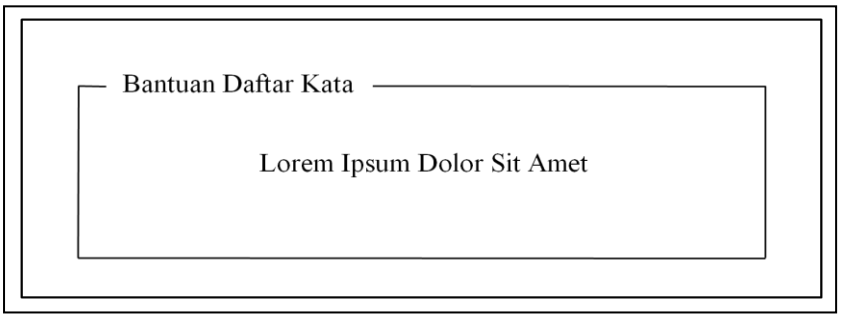
5. Halaman Daftar Kata



Gambar 3.30 Sketsa *Interface* Halaman Daftar Kata

Halaman ini adalah halaman yang ditampilkan ketika pengguna memilih menu "Daftar Kata" yang terdapat di halaman utama. Halaman ini akan menampilkan daftar kata pada tabel berdasarkan pilihan huruf depan yang dapat dipilih pengguna di bagian kiri halaman. Pada halaman ini, pengguna dapat kembali ke halaman utama dengan memilih pilihan kembali yang terletak di pojok kiri atas halaman, melihat bantuan, ataupun menambahkan suatu kata baru dengan memilih menu tambah kata.

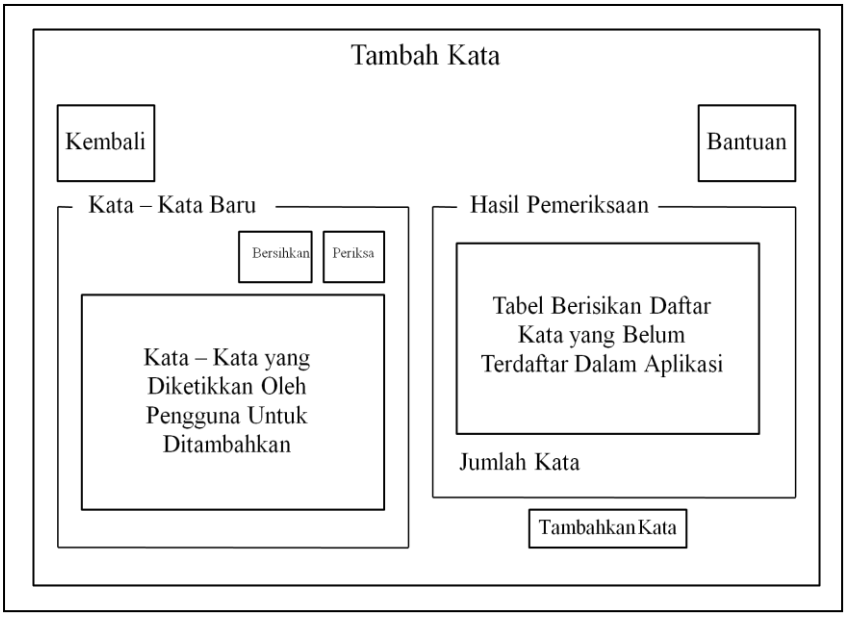
6. Halaman Bantuan Daftar Kata



Gambar 3.31 Sketsa *Interface* Halaman Bantuan Daftar Kata

Halaman ini adalah halaman yang ditampilkan ketika pengguna memilih menu "Bantuan" yang terdapat di halaman Daftar Kata. Halaman ini berisikan bantuan mengenai cara menggunakan halaman daftar kata beserta menu – menu yang dapat dipilih di halaman tersebut.

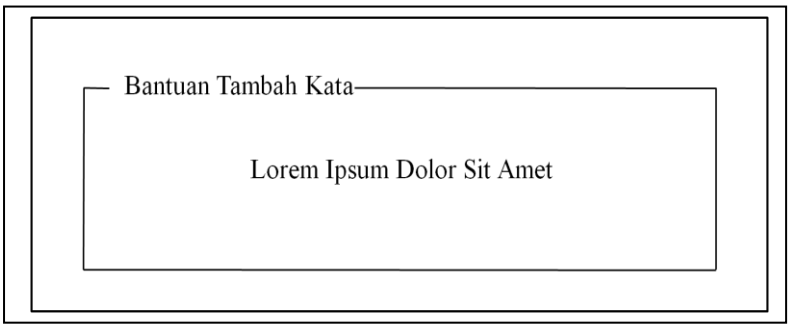
7. Halaman Tambah Kata



Gambar 3.32 Sketsa *Interface* Halaman Tambah Kata

Halaman ini adalah halaman yang akan ditampilkan ketika pengguna memilih pilihan "Tambah Kata" pada halaman Daftar Kata. Pada halaman ini, pengguna dapat mengetikkan sejumlah kata pada bagian kiri halaman, lalu mengklik tombol Periksa untuk melihat kata – kata yang belum terdaftar dalam aplikasi pada tabel di sebelah kanan. Kemudian pengguna dapat menambahkan kata – kata tersebut dengan mengklik tombol "Tambahkan Kata" yang terdapat di bawah tabel. Pada halaman ini, juga terdapat beberapa pilihan lain, seperti pilihan Pilihan "Kembali" untuk kembali ke halaman daftar kata, pilihan "Bantuan" untuk melihat bantuan mengenai cara – cara menambahkan kata, serta tombol "Bersihkan" untuk membersihkan kembali tabel dan kata – kata yang telah diketikkan.

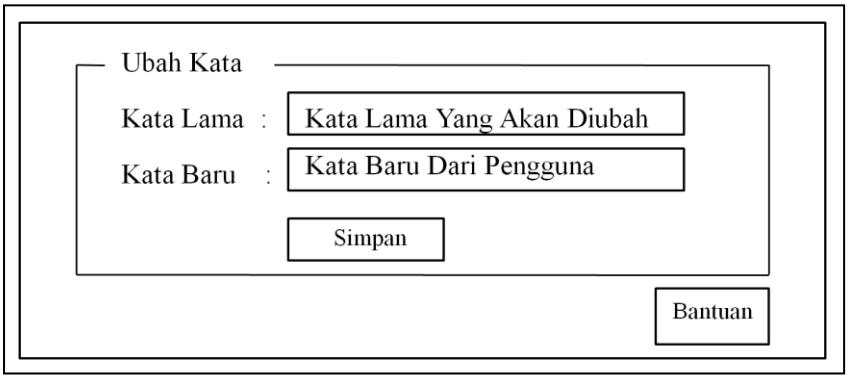
8. Halaman Bantuan Tambah Kata



Gambar 3.33 Sketsa *Interface* Halaman Bantuan Tambah Kata

Halaman ini adalah halaman yang akan ditampilkan ketika pengguna memilih menu "Bantuan" pada halaman Tambah Kata. Halaman ini berisikan informasi mengenai cara – cara menambahkan kata pada halaman Tambah Kata, beserta menu – menu yang dapat dipilih pada halaman tersebut.

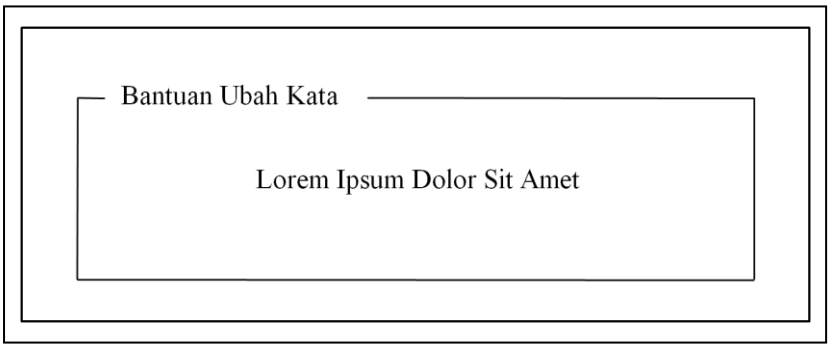
9. Halaman Ubah Kata



Gambar 3.34 Sketsa *Interface* Halaman Ubah Kata

Halaman ini akan ditampilkan ketika pengguna melakukan klik dua kali terhadap salah satu kata yang ditampilkan pada tabel di halaman Daftar Kata. Pada halaman ini, pengguna dapat mengganti suatu kata dengan kata lain yang belum terdaftar dalam aplikasi. Selain itu, pengguna juga dapat melihat bantuan mengenai cara pengubahan kata dengan memilih menu "Bantuan" pada bagian pojok kanan bawah halaman.

10. Halaman Bantuan Ubah Kata



Gambar 3.35 Sketsa *Interface* Halaman Bantuan Ubah Kata

Halaman ini adalah halaman yang ditampilkan ketika pengguna memilih menu “Bantuan” pada halaman ubah kata. Halaman ini berisikan informasi mengenai cara – cara mengubah suatu kata pada halaman Ubah Kata.