BAB III

METODOLOGI DAN PERANCANGAN APLIKASI

3.1 Metodologi Penelitian

Metode penelitian yang digunakan adalah sebagai berikut

1. Studi literatur

   Studi literatur dapat dilakukan dengan membaca jurnal atau *paper* serta buku yang berkaitan dengan algoritma Boyer Moore. Juga membaca artikel yang berhubungan dengan pembangunan aplikasi *mobile*, pemrograman Android

2. Perancangan Aplikasi

   Perancangan dapat dilakukan dengan membuat *flowchart diagram, Unified Modelling Language* atau disingkat UML, lalu pembuatan desain antarmuka aplikasi.

3. *Coding* dan Implementasi

   Melakukan *coding* terhadap aplikasi Kamusslang dengan basis pemrograman Android dengan proses pencarian *string* menggunakan algoritma Boyer Moore.

4. Uji coba

   Pada tahapan ini dilakukan pengujian terhadap kegunaan atau *usability* dari aplikasi Kamusslang, dengan mengamati proses jalannya aplikasi juga

dengan memberikan kuisioner kepada beberapa pengguna yang mencoba secara langsung aplikasi Kamusslang.

3.2 Perancangan Aplikasi

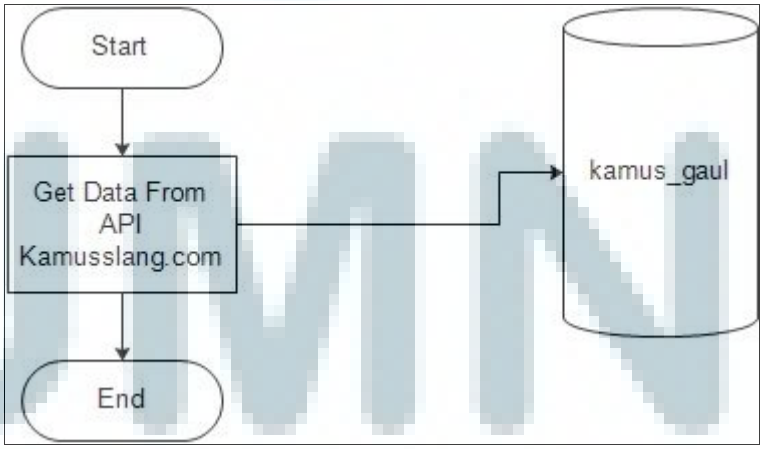
Dalam membangun sebuah sistem atau aplikasi yang baik dan benar perancangan sistem diharuskan spesifik, sehingga dapat mengurangi resiko kegagalan dari suatu aplikasi.

3.2.1 Flowchart Diagram

Dalam aplikasi Kamusslang memiliki beberapa *flowchart* yang dimana masing masing mempresentasikan alur kerja dari sistem yang ada.

A. Flowchart Diagram Aplikasi

Flowchart pada gambar menggambarkan alur kerja secara keseluruhan dari aplikasi Kamusslang. Pencarian dapat dilakukan apabila pengguna telah mengunduh data melalui API (*Application Programming Interface*). Dan data telah tersimpan di dalam basis data yang dimiliki oleh aplikasi Kamusslang.



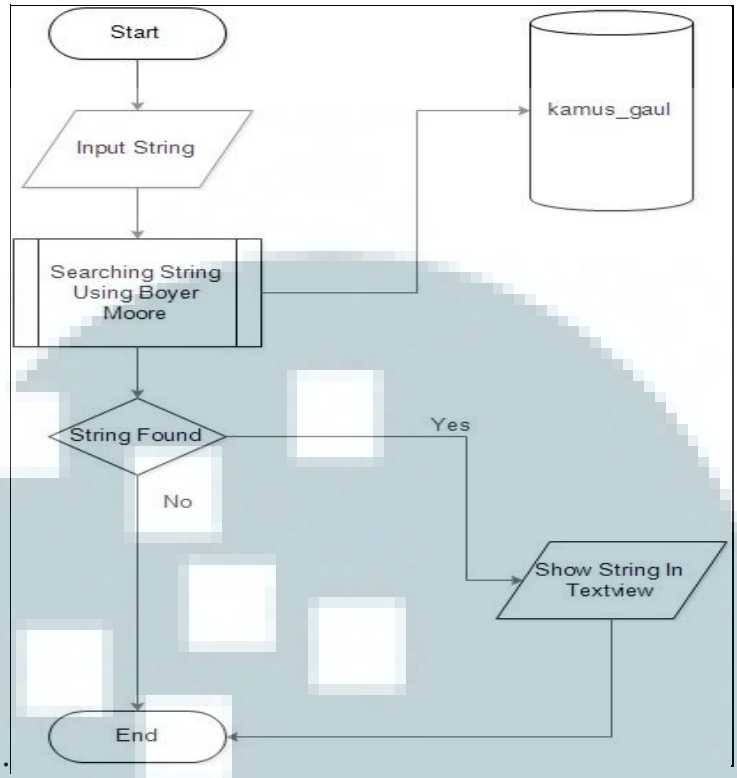
Gambar 3.1 Flowchart Aplikasi Pada Saat Menarik Data

Dari gambar 3.1 dapat dijabarkan sebagai berikut:

Flowchart ini akan dilakukan ketika pertama kali pengguna melakukan instalasi aplikasi ini. Dimulai dengan sistem meminta data yang berisikan kata-kata slang, berserta arti dan contoh penggunaannya melalui API Kamusslang.com. Sesudah melakukan permintaan sistem akan menyimpan data yang telah diambil ke dalam basis data yang bernama kamus_gaul. Selanjutnya pada sisi pengguna dapat memasukan kata yang ingin dicari definisinya. Proses penarikan data dilakukan hanya sekali pada saat pengguna sudah melakukan instalasi aplikasi Kamusslang pada perangkatnya.

B. Flowchart Diagram Pencarian Kata

Setelah data yang diambil melalui API disimpan di dalam SQLite, selanjutnya dapat dilakukan proses pencarian kata atau *string*, dengan *flowchart diagram* sebagai berikut



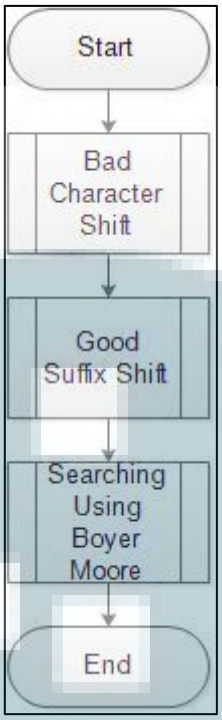
Gambar 3.2 Flowchart Pencarian Kata

Dari gambar 3.2 tersebut dapat dijabarkan sebagai berikut:

Flowchart dimulai ketika sistem sudah mendapatkan data melalui API, dan menerima informasi untuk mencari definisi dari suatu kata slang, selanjutnya dilakukan pengambilan data yang terdapat dalam basis data kamus_gaul, selanjutnya membaca teks atau *pattern* yang dimasukan oleh pengguna. Setelah itu dilakukan proses pencocokan dengan kata yang sudah ada di dalam bentuk *string*, dengan menggunakan algoritma Boyer Moore. Setelah proses pencarian dilakukan jika ditemukan hasil pencarian akan ditampilkan kepada pengguna dalam *listview*.

C. Flowchart Diagram Algoritma Boyer Moore

Alur proses kerja dari algoritma Boyer Moore dapat digambarkan seperti gambar 3.3

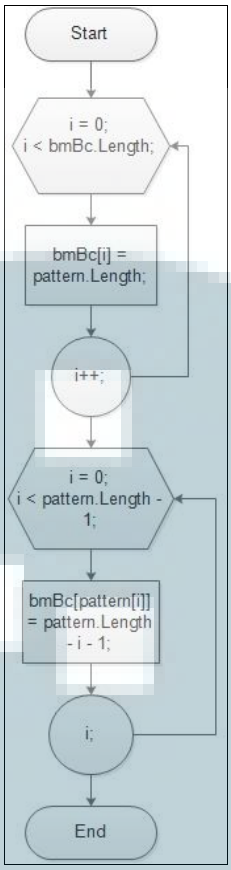


Gambar 3.3 Flowchart Garis Besar Algoritma Boyer Moore

Dalam proses pencarian *string* menggunakan algoritma Boyer Moore, diawali dengan proses *bad character shift*, lalu *good suffix shift*. Dari kedua proses ini dilakukan untuk menentukan banyaknya pola atau *pattern* dapat bergeser pada saat pencocokkan berlangsung. Setelah kedua proses ini dilakukan, algoritma akan mencocokkan karakter yang dimulai dari paling kanan karakter tersebut menuju ke kiri, dengan cakupan yang bergerak dari kiri menuju kanan. Ketika ditemukan karakter yang berbeda, maka pola atau *pattern* akan dilakukan penggeseran sesuai dengan nilai yang didapat dari proses *bad character shift* ataupun *good suffix shift.*

D. Flowchart Diagram Bad Character Shift

Flowchart pada gambar 3.4 akan menampilkan alur pada saat proses *bad character shift*

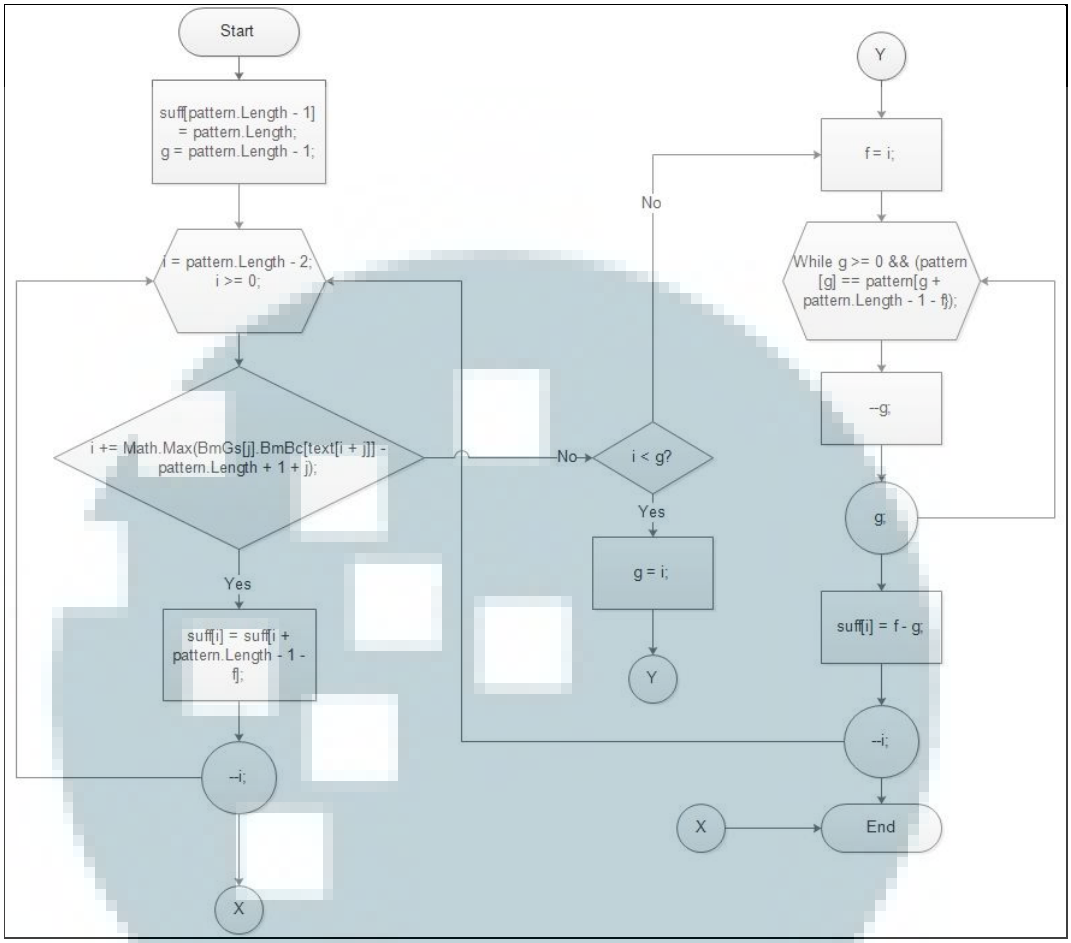


Gambar 3.4 Flowchart Perhitungan Bad Character Shift

Proses *bad character shift* memuat nilai pergeseran dari karakter huruf. Setiap karakter yang terdapat pada pola atau *pattern* akan diberikan nilai sebanyak jarak dari karakter paling kanan dari pola atau *pattern.* Pada karakter yang tidak ada dalam pola atau *pattern* akan diberikan nilai yang sama dengan banyaknya karakter yang dimasukan pada saat proses *input.*

E. Flowchart Diagram Perhitungan Suffix

Flowchart pada gambar 3.5 akan menampilkan alur pada saat proses perhitungan *suffix.*

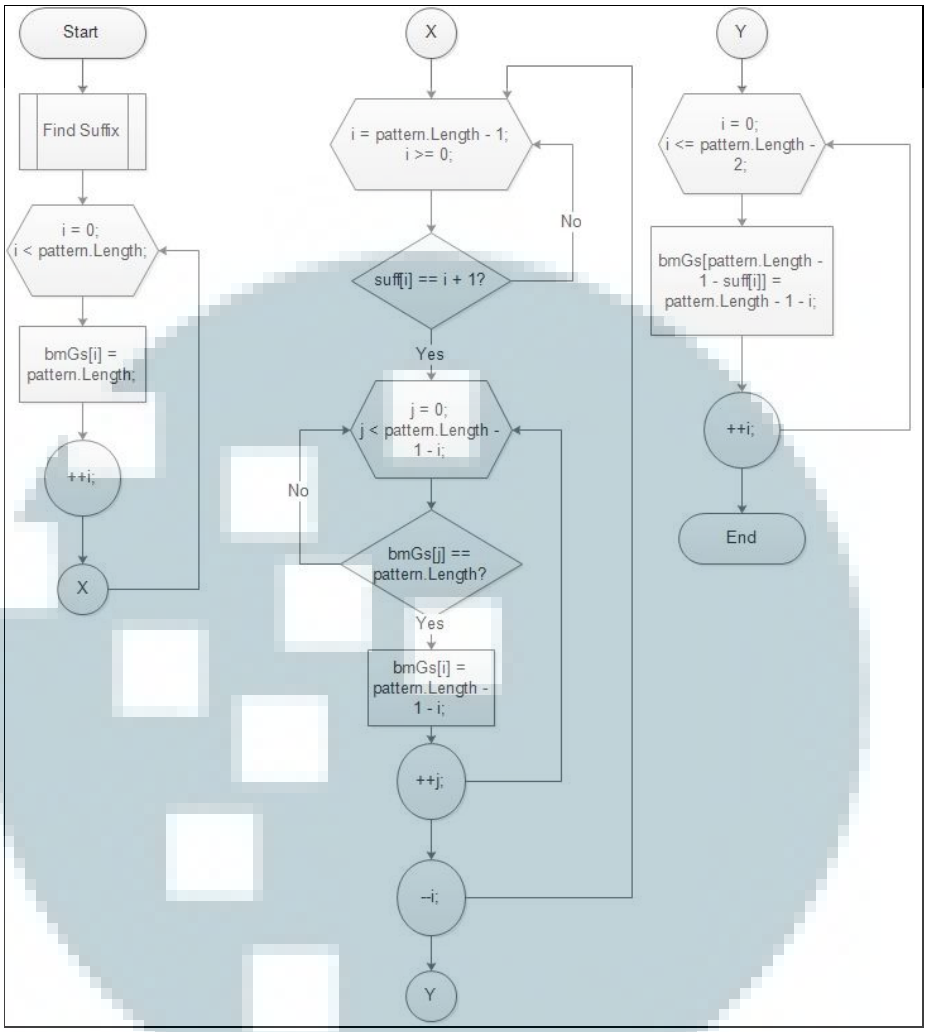


Gambar 3.5 Flowchart Perhitungan Suffix

Proses perhitungan *suffix* yang beguna untuk proses *good suffix shift* yang akan dilakukan setelah ini. Nilai-nilai dari setiap karakter yang terdapat pada pola atau *pattern* menunjukan adanya pengulangan dari akhiran yang dinamakan *suffix*. Perhitungan *suffix* juga dapat digunakan untuk mencari dimana letak pengulangan akhiran, sehingga pada saat proses *good suffix shift* dapat diketahui banyak karakter yang bergeser untuk pencocokan.

F. Flowchart Good Suffix Shift

Flowchart pada gambar 3.6 akan menampilkan alur pada saat proses *good suffix shift*

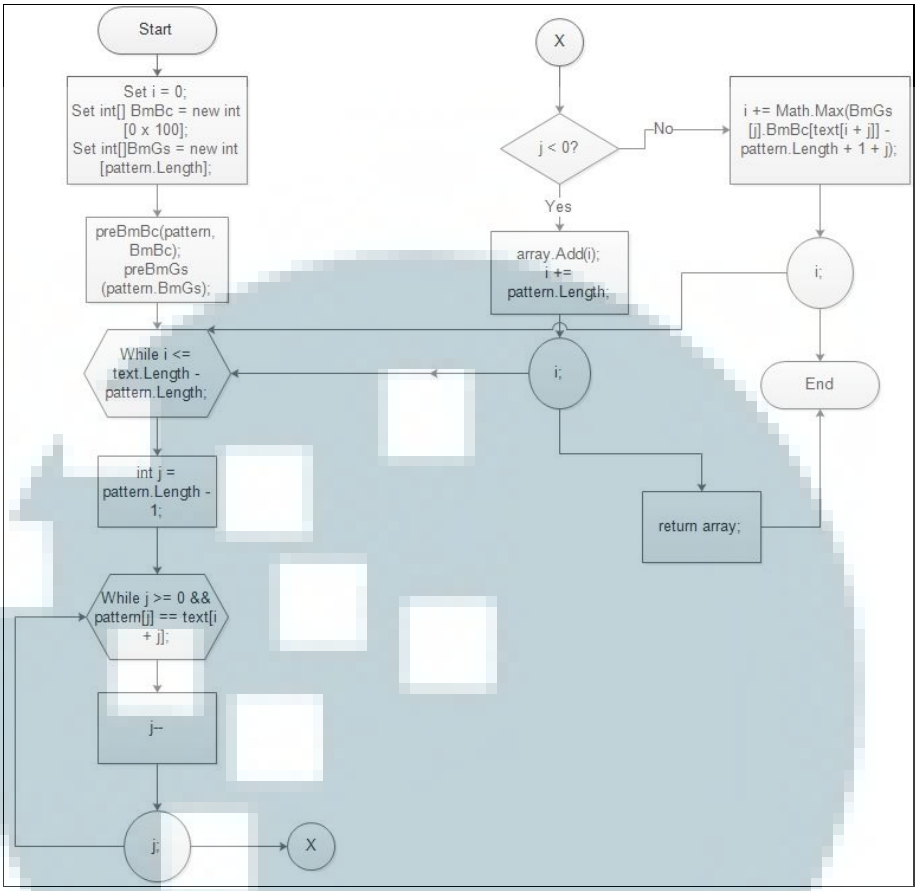


Gambar 3.6 Flowchart Perhitungan Good Suffix Shift

Proses *good suffix shift* terkandung nilai yang digunakan untuk melakukan pergeseran ketika terjadi perbedaan ditemukan berdasarkan letak karakter pada posisi mana yang mengakibatkan perbedaan. Nilai dari setiap karakter yang terdapat didalam pola atau *pattern* bergantung pada ada atau tidaknya pengulangan akhiran atau disebut *suffix*. Makin banyaknya terjadi pengulangan akhiran, maka semakin kecil pula nilai pergeseran yang didapat.

G. Flowchart Diagram Pencarian String Dengan Algortima Boyer Moore

Flowchart pada gambar 3.7 akan menampilkan alur pada saat proses pencarian *string* dengan menggunakan algoritma Boyer Moore

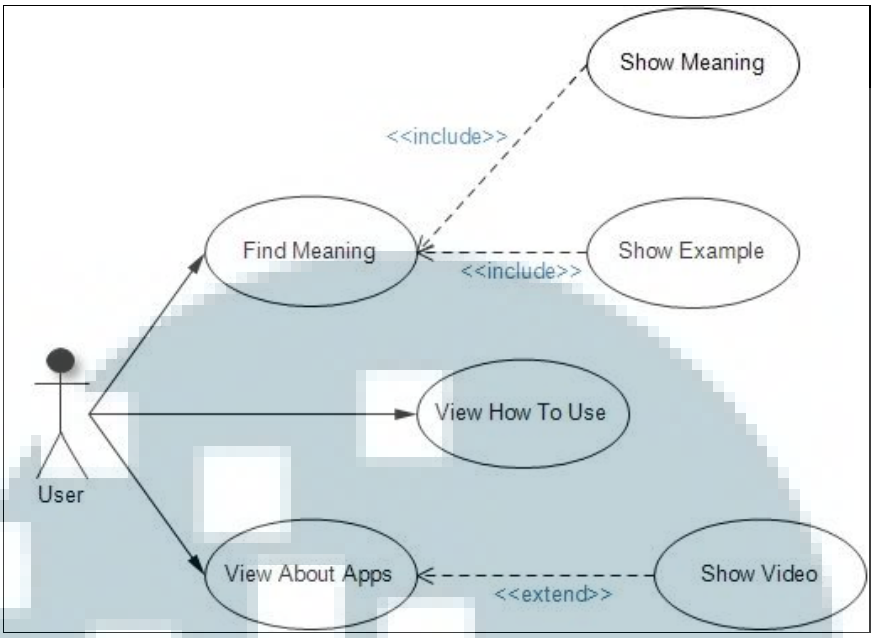


Gambar 3.7 Flowchart Proses Pencarian Dengan Algoritma Boyer Moore

Proses diawali dengan penempatan cakupan dari pola atau *pattern.* Selanjutnya proses pencarian dimulai dari karakter yang paling kanan pola atau *pattern*, lalu setiap karakter akan dilakukan proses pembandingan pada masing-masing karakter sampai pada karakter terakhir. Jika terjadi perbedaan maka akan dilakukan pengecekan dengan melihat pada proses *bad character shift* dan *good suffix shift.* Nilai yang paling besar didapatkan dari kedua proses tersebut akan dijadikan nilai pergeseran cakupan yang dilakukan.

3.2.2 Use Case Diagram

Gambar 3.8 merupakan *use case diagram* dari aplikasi kamusslang



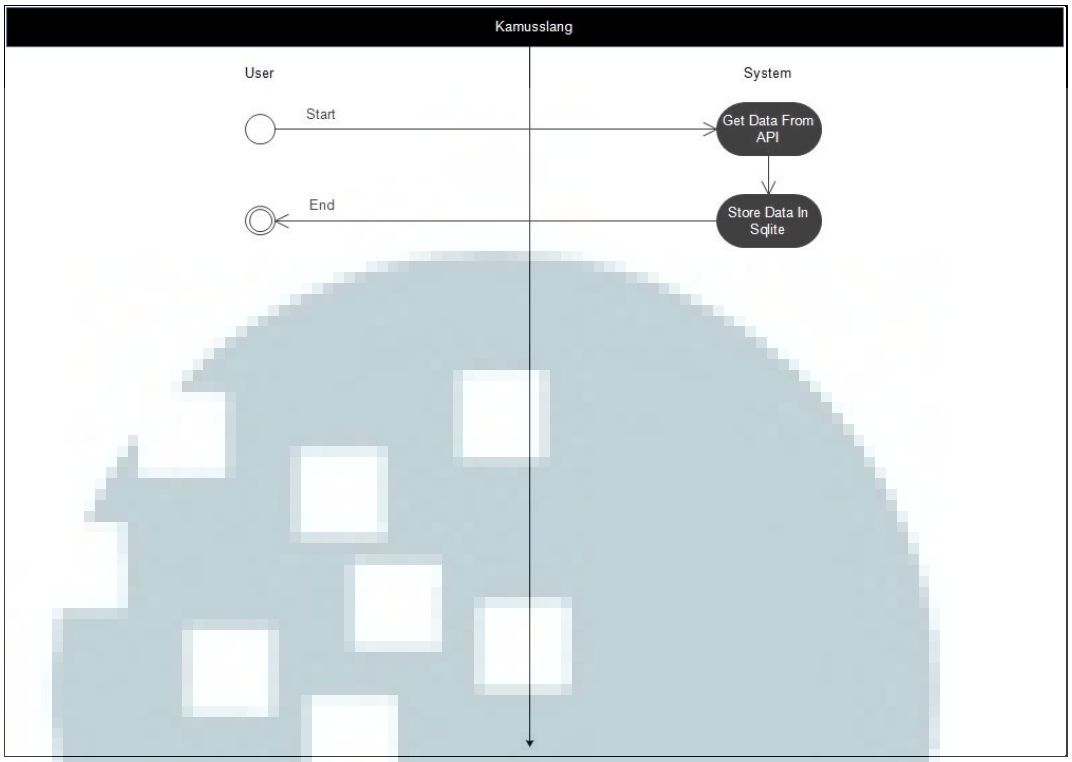
Gambar 3.8 Use Case Diagram Aplikasi Kamusslang

Dengan penjelasan sebagai berikut:

Aplikasi Kamusslang dapat diakses oleh pengguna melalui perangkat Androidnya. Aktivitas yang dapat dilakukan oleh pengguna, yaitu mencari arti kata slang. Dalam mencari arti dari kata slang, pengguna dapat melihat definisi dari kata slang. Pengguna juga dapat melihat contoh dari penggunaan kata slang yang dicarinya. Pengguna dapat melihat cara penggunaan dari aplikasi ini. Pengguna juga dapat melihat tentang aplikasi ini dengan menghubungkan langsung ke tautan video di Youtube

3.2.3 Activity Diagram

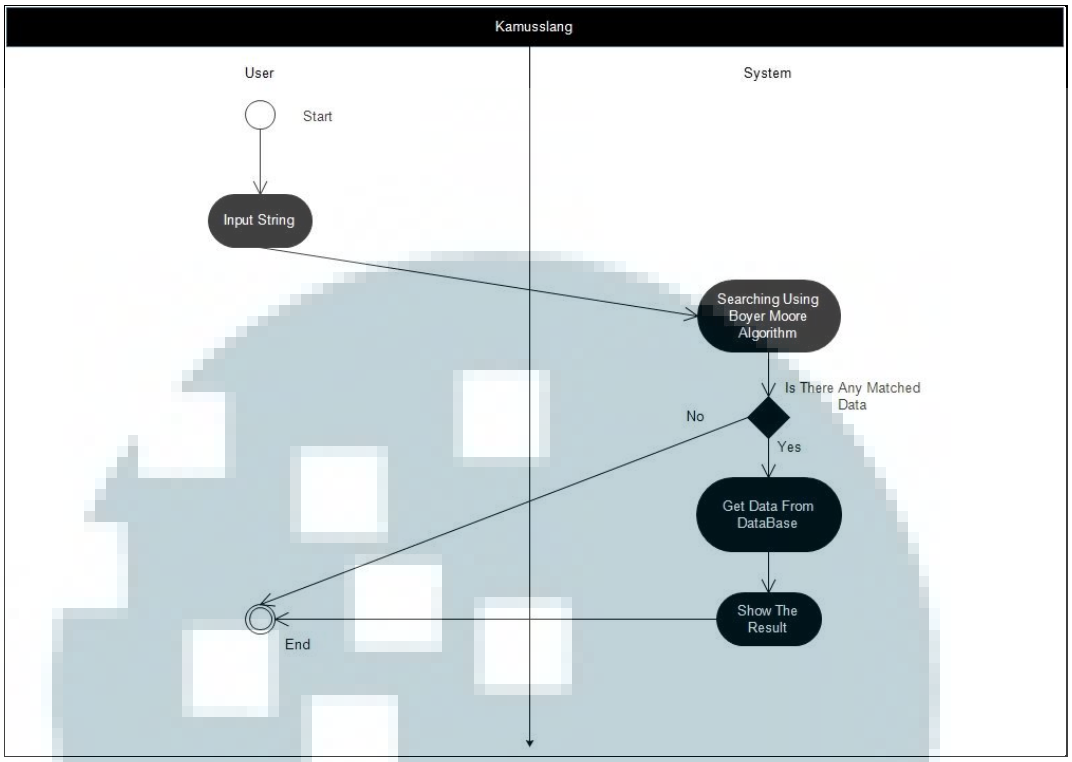
Aplikasi ini membutuhkan data yang berasal dari basis data situs Kamusslang.com, atau dalam kata lain, mengacu kepada Kamusslang.com, sehingga membutuhkan API (*Application Programming Interface*) sebagai perantara komunikasi dari aplikasi *mobile* dan situs kamusslang.com. Gambar 3.9 merupakan *activity* diagram yang terjadi di dalam pencarian kata slang.



Gambar 3.9 Activity Diagram Proses Penarikan Data

Dengan penjelasan sebagai berikut:

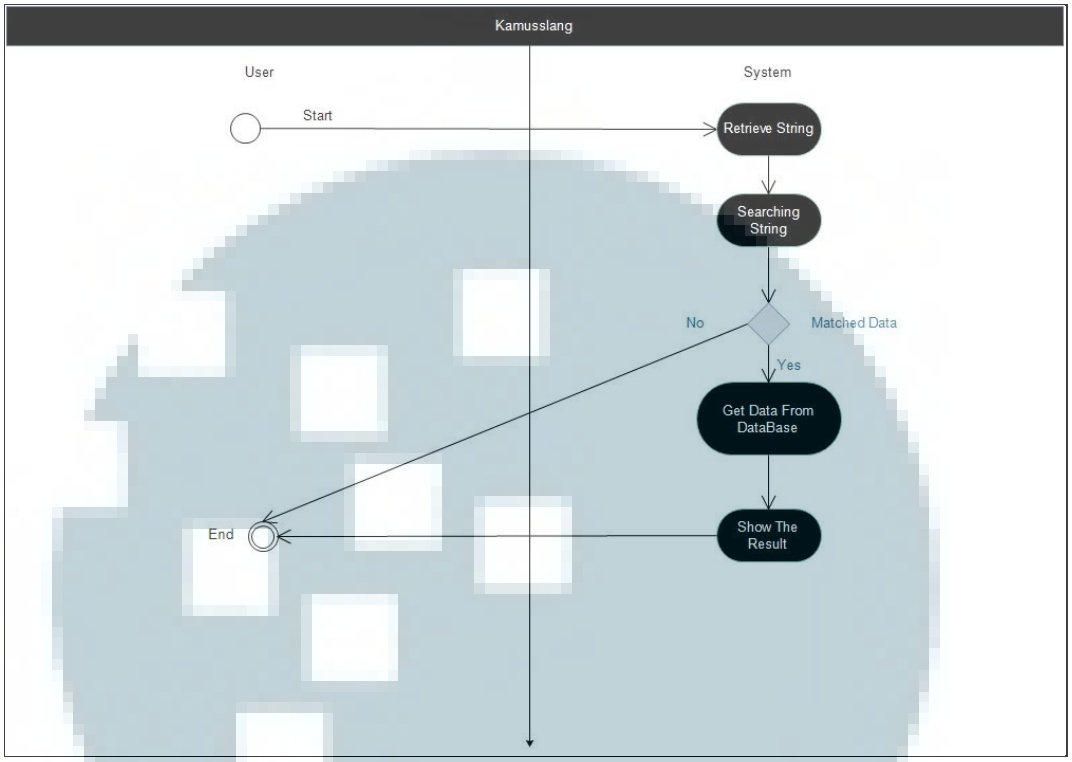
Aktivitas pertama terdapat pada sisi pengguna dimana pengguna memulai aplikasi, ketika aplikasi baru dipasangkan pada perangkat pengguna, sistem langsung melakukan penarikan data melalui API, dan menyimpannya pada basis data SQLite. Aplikasi Kamusslang, hanya memerlukan satu kali proses penarikan data yang berisikan informasi kata slang, arti dan penggunaannya. Selanjutnya hanya melihat kepada basis data aplikasi ini.



Gambar 3.10 Activity Diagram Proses Pencarian String

Dengan penjelasan sebagai berikut:

Setelah semua data telah disimpan dalam basis data SQLite, pengguna dapat memasukan kata slang yang ingin dicari arti katanya. Setelah memasukan kata, sistem akan mencari *string* yang didapatkan dari masukan tersebut dengan menggunakan algoritma Boyer Moore. Setelah ditemukan dalam basis data, lalu sistem menampilkan arti dari kata slang yang dicari, berserta dengan contoh penggunaannya.



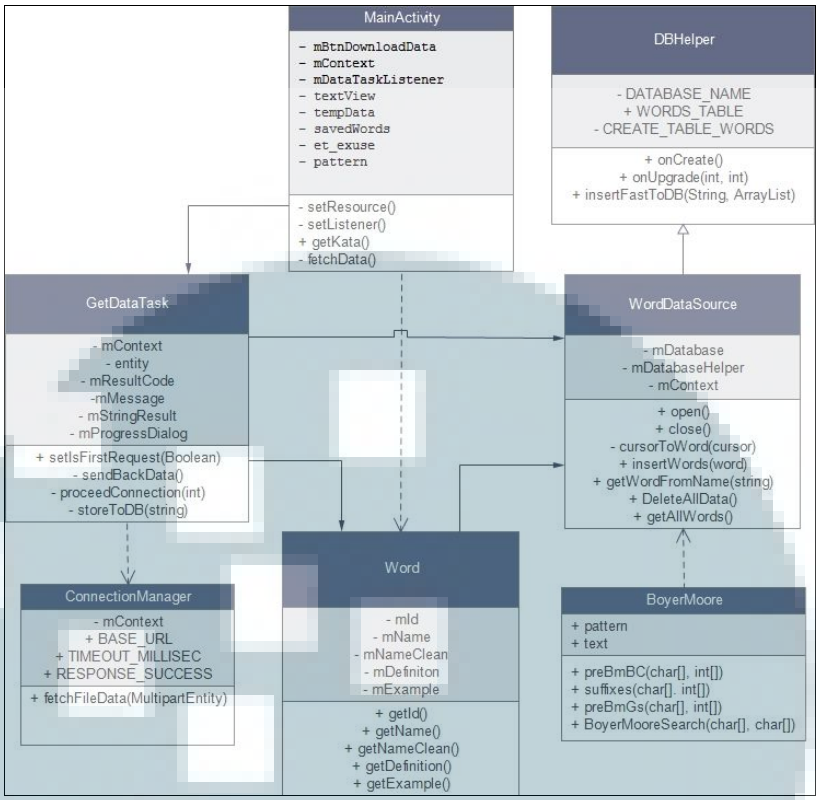
Gambar 3.11 Activity Diagram Proses Penampil Hasil Pencarian

Dengan penjelasan sebagai berikut:

Setelah sistem menerima kata yang dimasukan pengguna, sistem akan melakukan pencarian *string*, ketika ditemukan lalu, sistem akan menampilkan ke antarmuka pengguna dengan bentuk *listview*.

3.2.4 Class Diagram

Hubungan antara kelas dari suatu aplikasi secara umum dapat dijabarkan menggunakan berbagai bentuk diagram, berikut gambar 3.11 merupakan *class* diagram dari aplikasi kamusslang.



Gambar 3.12 Class Diagram Aplikasi Kamusslang

Proses pencarian kata pada aplikasi Kamusslang diawali dengan penarikan data yang dilakukan oleh sistem melalui API Kamusslang.com, dimana situs Kamusslang.com menjadi acuan bagi basis data pada aplikasi ini. Pada *class* diagram, *class* ConnectionManager berfungsi sebagai pembuka koneksi dengan API menggunakan JSON. Lalu *class* GetDataTask berfungsi untuk penarik data dengan menterjemahkan JSON yang kemudian dijadikan bentuk *array*. Kemudian pengguna memasukan kata atau istilah bahasa slang yang akan dicari. Sesudah pengguna memasukan kata atau istilah, sistem melakukan proses pencarian kata dengan menggunakan algoritma Boyer Moore. Pada *class* Word akan membagi-bagi data *string* tersebut kedalam beberapa variabel yang nantinya akan disimpan didalam basis data SQLite. *Class* DBHelper berfungsi untuk membuat basis data SQLite, membuat tabel, serta memasukan *string* kedalam

basis data. *Class* WordDataSource berfungsi untuk mendapatkan semua kata yang ada di dalam basis data kepada *class* selanjutnya untuk proses pencarian *string. Class* BoyerMoore berfungsi untuk mencari kata berdasarkan masukan teks yang dilakukan oleh pengguna.

3.2.5 Struktur Tabel

Dalam pembangunan aplikasi Kamusslang ini, hanya terdapat 1 (satu) tabel yang digunakan. Berikut penjelasan dari tabel yang digunakan.

Nama *Database* : kamus_gaul

Nama Tabel : WORDINGS

Fungsi : Menyimpan kata slang, arti, dan penggunaan kata slang

Tabel 3.1 Tabel Struktur Database Aplikasi Kamusslang

Nama Kolom	**Tipe**	**Keterangan**
id	integer primary key	id dari masing-masing istilah
name	Text	nama dari sebuah istilah
name_clean	Text	
definition	Text	definisi dari sebuah istilah
example	Text	contoh penggunaan dari sebuah kata slang

3.2.6 Rancangan Antarmuka

Rancangan dari antarmuka aplikasi ini dapat dijabarkan sebagai berikut

A. Rancangan Antarmuka Splashscreen

Pada rancangan antarmuka pada pertama kali aplikasi dibuka akan tampil di layar yaitu *splashscreen,* yang akan tampil logo dari Kamusslang.com selama beberapa detik, seperti pada gambar 3.13

Gambar 3.13 Tampilan Antarmuka *Splashscreen*

B. Rancangan Antarmuka Menu Utama Kamus

Penjelasan untuk gambar 3.14 pada laman menu kamus akan terdapat sebuah *text field* yang digunakan untuk menerima masukan kata yang dilakukan oleh pengguna yang digunakan sebagai pola atau *pattern* yang akan dilanjutkan ke dalam proses pencarian kata. Selain sebuah *text field* terdapat pula dua tombol, pada tombol yang terletak di sebelah kiri berguna untuk memberikan arti dari kata yang terlah dimasukkan oleh pengguna. Lalu tombol yang di sebelah kanan berfungsi untuk menarik data yang berasal dari API.

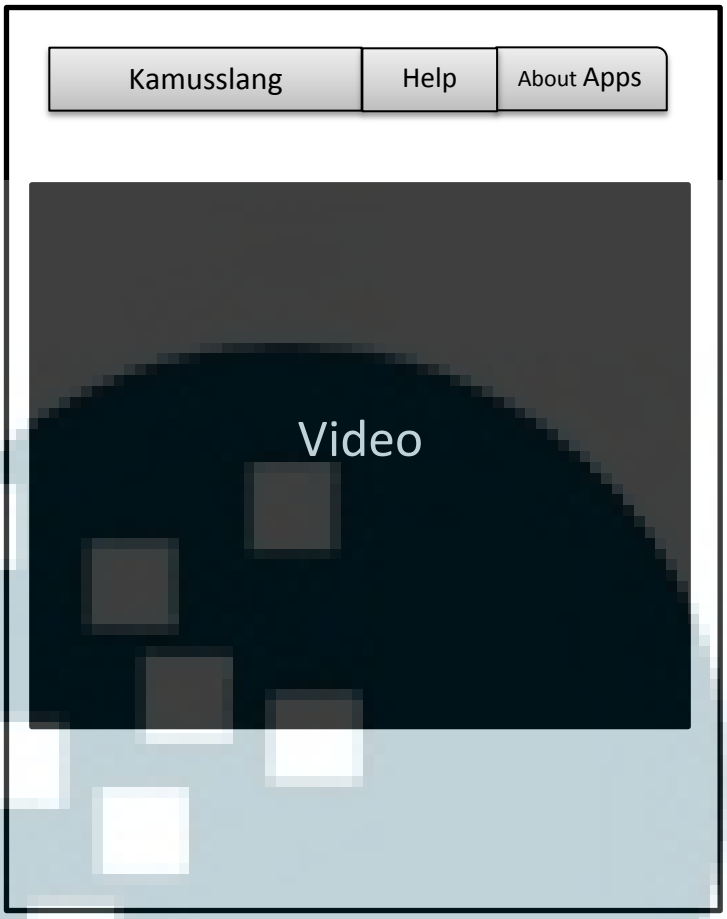
Di bawah kedua tombol tersebut terdapat *text field* yang menampilkan hasil arti dari kata yang dimasukkan oleh pengguna. Di bawah text tersebut *terdapat text view* yang akan menampilkan contoh dari penggunaan kata slang

tersebut. Pada Kanan atas terdapat menu yang menampilkan pilihan About Apps dan How to Use Apps.

Gambar 3.14 Tampilan Antarmuka Menu Kamus

C. Rancangan Antarmuka Menu About Apps

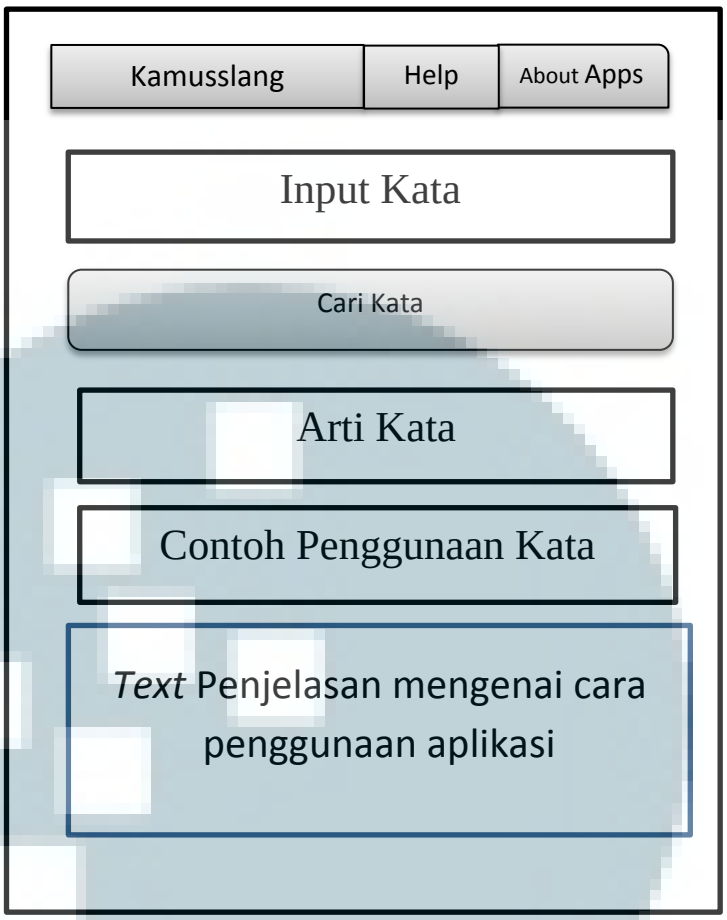
Pada gambar 3.15 menggambarkan antarmuka dari menu About Apps, dimana akan menampilkan penjelasan dari Kamusslang.com dalam bentuk video, yang berasal dari Youtube.



Gambar 3.15 Tampilan Antarmuka Menu About Apps

D. Rancangan Antarmuka Menu How To Use Apps

Pada gambar 3.16 menggambarkan antarmuka dari menu How To Use Apps, dimana akan menampilkan penjelasan dari cara penggunakan aplikasi Kamusslang ini



Gambar 3.16 Tampilan Antarmuka Menu How To Use Apps