

**APLIKASI KAMUS BAHASA DAERAH SUKU SEKATAK BUJI
KALIMANTAN UTARA MENGGUNAKAN METODE BRUTE
FORCE BERBASIS ANDROID**

SKRIPSI



**Disusun oleh :
AFRIANSYAH RAMADANI
10.18.167**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2015**

LEMBAR PERSETUJUAN

**APLIKASI KAMUS BAHASA DAERAH SUKU SEKATAK BUJI KALIMANTAN
UTARA MENGGUNAKAN METODE BRUTE FORCE BERBASIS ANDROID**

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Informatika Strata Satu (S-1)*

Disusun oleh :


**AFRIANSYAH RAMADANI
NIM : 10.18.167**

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Ali Mahmudi, B.Eng. PhD
NIP: P.1031000429


Yosen Agus Pranoto, ST.MT.
NIP.P. 1031000432

**Mengetahui
Ketua Program Studi Teknik Informatika S-1**


Joseph Dedy Irawan, ST.MT.
NIP. 197404162005011002

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2015

**APLIKASI KAMUS BAHASA DAERAH SUKU SEKATAK BUJI KALIMANTAN
UTARA MENGGUNAKAN METODE BRUTE FORCE BERBASIS ANDROID**

AFRIANSYAH RAMADANI

Program Studi Teknik Informatika S-1

Fakultas Teknologi Industri, Institut Teknologi Nasional Malang

Jln. Raya Karanglo Km 2 Malang

Email : 4friansyah@gmail.com

Dosen Pembimbing :

- 1. Ali Mahmudi, B.Eng, PhD.**
- 2. Yosep Agus Pranoto, ST.MT.**

ABSTRAK

Sekatak Buji adalah salah satu desa yang ada di kecamatan Sekatak, Kabupaten Bulungan Kalimantan Utara, Indonesia. Bahasa yang digunakan disekatak buji antara lain bahasa Tidung, Bulungan dan Dayak Berusu. Karena mulai pudarnya pengetahuan masyarakat sekatak buji akan bahasa sukunya sendiri dalam hal ini adalah suku Bulungan, Tidung dan Dayak Berusu yang merupakan penduduk asli desa sekatak buji, penulis ingin membuat kamus bahasa daerah.

Brute force adalah sebuah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (problem statement) dan definisi konsep yang dilibatkan. Algoritma brute force memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (obvious way).

Aplikasi ini telah dilakukan pengujian terhadap 10 user dan 70% memberikan nilai baik atas keseluruhan dari aplikasi ini dan 30% memberikan nilai cukup. Untuk pengujian fungsional sistem dan pengujian OS dan Resolusi Layar mendapatkan hasil pengujian dengan nilai 100%. Para user juga memberikan beberapa masukan yang membangun kepada penulis, Banyak dari mereka berharap agar aplikasi ini bisa dikembangkan lebih jauh lagi.

Kata Kunci : Sekatak Buji, Brute Force, Hasil Pengujian.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT. yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul “APLIKASI KAMUS BAHASA DAERAH SUKU SEKATAK BUJI KALIMANTAN UTARA MENGGUNAKAN METODE BRUTE FORCE BERBASIS ANDROID” dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Jurusan Teknik Informatika S-1 ITN Malang dan untuk mencapai gelar Sarjana Komputer.

Keberhasilan penyelesaian laporan skripsi ini tidak terlepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST.MT selaku Ketua Jurusan Teknik Informatika S-1 yang selalu memberikan dukungan dan bantuan tak terhingga kepada penulis.
4. Ali Mahmudi, B.Eng, PhD. selaku Dosen Pembimbing I yang memberikan kemudahan dan jalan keluar yang terbaik.
5. Yosep Agus Pranoto, ST.MT. selaku Dosen Pembimbing II yang mau meluangkan waktu untuk membimbing penulis dan memberi kemudahan dalam pembuatan skripsi ini.
6. Ayah dan Ibu yang telah memberikan dorongan dan kebutuhan finansial.
7. Seluruh Dosen Teknik Informatika S1, yang telah banyak memberikan ilmu selama menjadi Mahasiswa ITN Malang.
8. Teman-teman Teknik Informatika 2010 yang membantu dalam pengerjaan dan penyelesaian skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan skripsi ini. Untuk itu penyusun mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan penulis sendiri maupun pembaca sekalian.

Malang, Agustus 2015

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi Pemecahan Masalah	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Android	5
2.1.1 Versi-versi Android	6
2.1.2 Karakteristik Android	6
2.1.3 Cara Kerja Sistem Operasi Android	7
2.2 Eclipse	8
2.3 Database	10
2.3.1 Database SQLite	10
2.4 Java	11
2.4.1 Java API	11
2.4.2 Java Virtual Machine	12
2.4.3 Sistem Operasi Java	12
2.4.4 Dasar Pemrograman Java	12
2.5 Algoritma Pencarian Kata	13
2.5.1 Algoritma Brute Force	13
BAB III PERANCANGAN SISTEM	17
3.1 Spesifikasi Aplikasi	17

3.2	Analisa Sistem	17
3.2.1	Analisa Perangkat Keras (Hardware)	17
3.2.2	Analisa Perangkat Lunak (Software)	18
3.3	Perancangan Aplikasi	18
3.3.1	Blok Diagram Pembuatan Aplikasi	18
3.3.2	Flowchart Sistem	19
3.3.3	Penerapan Algoritma Brute Force Pada Kamus Bahas Daerah Sekatak Buji	19
3.3.4	Perancangan Tabel Database.....	20
3.3.5	Perancangan Form Aplikasi	21
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM		23
4.1	Kebutuhan Sistem	23
4.2	Pengujian Aplikasi Pada Perangkat Android	24
4.2.1	Tampilan Awal Aplikasi Setelah di Instal	24
4.2.2	Tampilan Awal Aplikasi Pertama Kali	25
4.2.3	Tampilan Aplikasi Ketika di Lakukan Terjemahan	25
4.2.4	Tampilan Menu About	27
4.2.5	Tampilan Aplikasi Ketika Tidak Ditemukan Terjemahannya ...	28
4.3	Hasil Pengujian	30
4.3.1	Tabel Fungsional Program	30
4.3.2	Tabel Pengujian OS dan Resolusi Layar.....	31
4.3.3	Tabel Pengujian User	31
BAB V PENUTUP		33
5.1	Kesimpulan	33
5.2	Saran	33
DAFTAR PUSTAKA		34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Media elektronik pada saat ini berpengaruh sangat pesat dalam kehidupan manusia. Demikian juga dalam perkembangan dibidang teknologi *mobile phone* yang semakin pesat secara tidak langsung telah memberikan keinginan dalam pelestarian bahasa daerah khususnya bahasa suku daerah sekatak buji berupa aplikasi kamus bahasa Tidung, Bulungan dan Dayak Berusu. Dengan adanya kamus ini, setiap orang dapat belajar bahasa Suku yang terdapat di daerah sekatak buji.

Pada daerah sekatak buji terdapat tiga bahasa suku antara lain bahasa Tidung, Bulungan dan Dayak Berusu. Karena mulai pudarnya pengetahuan masyarakat sekatak buji akan bahasa sukunya sendiri dalam hal ini adalah suku Bulungan, Tidung dan Dayak Berusu yang merupakan penduduk asli desa sekatak buji, penulis ingin membuat kamus bahasa daerah. Dengan adanya kamus bahasa suku sekatak buji ini, kiranya dapat turut serta melestarikan bahasa yang ada didesa sekatak buji dan juga dapat membantu pengguna dari suku lain yang ingin belajar bahasa Bulungan, Tidung dan Dayak Berusu untuk memenuhi kosa kata dalam bahasa yang sering digunakan penduduk desa sekatak buji sehari-hari mengingat tidak adanya mata pelajaran ekstrakurikuler serta buku penunjang yang diajarkan di sekolah daerah sekatak buji.

Berdasarkan permasalahan diatas, penulis berkeinginan untuk membuat aplikasi kamus bahasa daerah sekatak buji berbasis android, mengingat android adalah sebagai salah satu sistem operasi yang berbasis *mobile* dikarenakan perkembangan *mobile phone* di Indonesia berkembang sangat pesat.

Pada aplikasi kamus bahasa daerah dibuat menggunakan algoritma pencarian *string Brute Force*. Algoritma *Brute force* adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Algoritma *brute force* seringkali lebih mudah diimplementasikan dari pada algoritma yang lebih

BAB II

LANDASAN TEORI

2.1 Android

Android adalah sistem operasi yang berbasis Linux untuk mobile phone dengan cara ditanamkan di dalam suatu seperti *smartphone* dan komputer tablet. Android menyediakan platform terbuka bagi para *developer* untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services (GMS)* dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution (OHD)*.



Gambar 2.1 Lambang Android

2.1.1 Versi-versi Android

Berikut ini adalah versi-versi system operasi android yang beredar hingga sekarang:

1. Android 1.1 (*Bender*)
2. Android versi 1.5 (*Cupcake*)
3. Android Versi 1.6 (*Donut*)
4. Android versi 2.0/2.1 (*Eclair*)
5. Android versi 2.2 (*Frozen Youghurt (Froyo)*)
6. Android versi 2.3 (*Gingerbeard*)
7. Android versi 3.0 (*Honycomb*)
8. Android versi 4.0 (*ICS: Ice Cream Sandwich*)
9. Android versi 4.1 (*Jelly Bean*)
10. Android versi 4.2 (*A New Flafor of Jellly Bean*)

2.1.2 Karakteristik Android

Android memiliki karakteristik sebagai berikut:

1. Terbuka
Anroid dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti perangkat seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain.
 2. Semua aplikasi dibuat sama
Android tidak memberikan perbedaan terhadap aplikasi utama dari perangkat dan aplikasi yang dikembangkan oleh *user*. Semua aplikasi dibangun untuk memiliki akses yang sama terhadap kemampuan perangkat dalam menyediakan layanan dan aplikasi yang luas terhadap *user*.
 3. Pengembangan aplikasi yang cepat dan mudah
Android menyediakan akses yang sangat luas kepada *user* untuk menggunakan *library* yang diperlukan dalam *tools* yang dapat digunakan untuk membangun aplikasi yang semakin baik. Android memiliki
-

sekumpulan *tools* yang dapat digunakan sehingga membantu para *developer* dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Google Inc. Sepenuhnya menjadikan Android *open source* sehingga para *developer* dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari Google dan dapat membangun Android tanpa adanya batasan-batasan. Android *Software Developer*. Android *Software Development Kit* (SDK) menyediakan alat dan *Application Programming Interface* (API) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java dan XML.

2.1.3 Cara Kerja Sistem Operasi Android

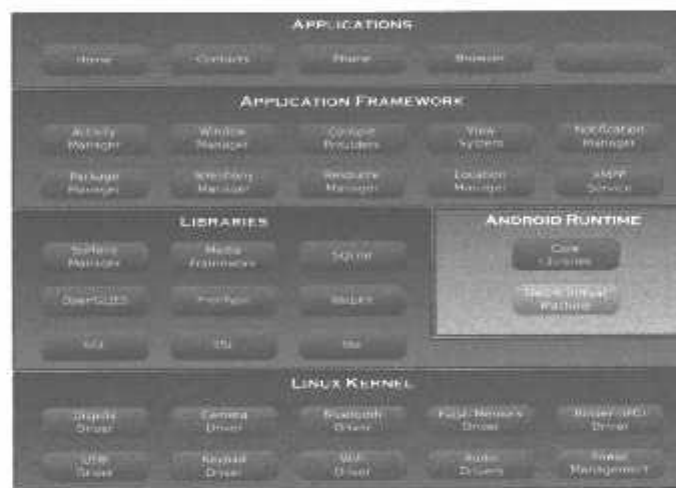
Google sebagai pencipta Android yang kemudian diasuh oleh *OpenHandset Alliance* mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Tumpukan paling bawah adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun Android, yang mencakup *memory management*, *security setting*, *power management*, dan beberapa *driver hardware*. Sebagai contoh, *device* android dilengkapi dengan kamera. Kernel Android terdapat *driver* kamera yang memungkinkan pengguna mengirimkan perintah kepada *hardware* kamera. Level berikutnya dari tumpukan ini adalah *library*, yakni serangkaian instruksi kepada *smartphone* yang berisi cara menangani data-data yang berbeda. Sebagai contoh, *media framework library* pada Android mendukung pemutaran dan perekaman berbagai format *audio*, video dan gambar.

Bertempat di level yang sama dengan *library* adalah lapisan *runtime* yang mencakup serangkaian inti *library Java*. Denganya para programmer dapat mengembangkan aplikasi untuk android menggunakan bahasa pemrograman *Java*. Lapisan selanjutnya adalah *application framework*, yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*.

Dilapisan teratas bercokol aplikasi itu sendiri. Dilapisan inilah pengguna menemukan fungsi-fungsi dasar *smartphone*, seperti menelpon dan mengirim

pesan singkat, menjalankan *web browser*, mengakses daftar kontak, dan lain-lain, bagi rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi-fungsi dasar tersebut melalui *user interface*.

Ada 4 hal yang harus dipahami dalam membangun aplikasi berbasis *Android Activity* adalah tampilan gratis yang kita lihat ketika menjalankan sebuah aplikasi. Sebuah aplikasi dapat mempunyai lebih dari satu *activity*. *Intent* adalah serangkaian *value* yang menunjukkan apa yang harus dilakukan ketika terjadi perpindahan layar. *Service* adalah layanan yang bekerja di *Backend*. *Content provider* memungkinkan sebuah aplikasi untuk dapat menyimpan dan menerima data dari database. Arsitektur android ditunjukkan pada gambar 2.2.



Gambar 2.2 Arsitektur Android (Arif Akbarul Huda 2013)

Dalam paket sistem operasi android terdiri dari beberapa unsur seperti tampak pada gambar dibawah. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.

2.2 Eclipse

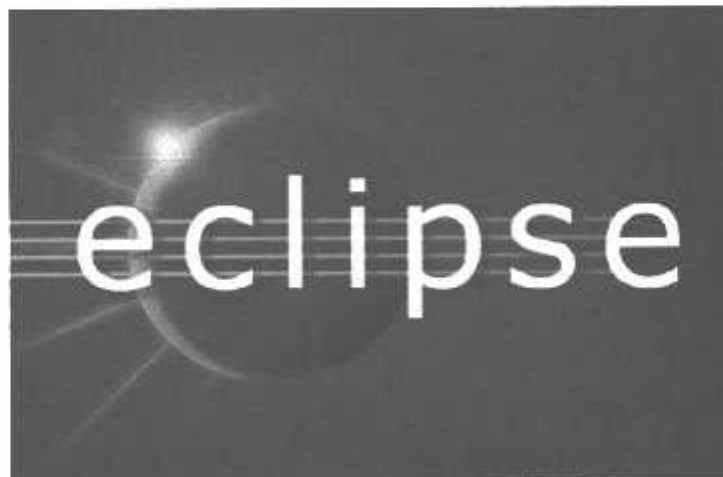
Eclipse adalah sebuah *Intergrade Development Environment* atau IDE yang bertujuan untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform (platform-independent)*.

Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Eclipse dapat digunakan dalam berbagai sistem operasi antara lain *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X*.
2. *Multi-role*: Digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.
3. *Multi-languange*: *eclipse* dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, PHP*, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *Open Source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Untuk keperluan programming eclipse menyediakan database secara default yang sudah ada di *library* yang disebut *SQLite*. Perangkat lunak Eclipse ditunjukkan pada gambar 2.3.



Gambar 2.3 Logo Perangkat Lunak Eclipse (Arif Akbarul Huda 2013)

2.3 Database

Database atau basisdata adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari database tersebut.

2.3.1 Database SQLite

Database SQLite merupakan paket *software* yang bersifat *public domain* yang menyediakan system manajemen basis data relasional atau *Relational Database Management System* (RDBMS). Kata "*Lite*" pada SQLite tidak menunjuk pada kemampuannya, melainkan pada sifat dari SQLite, yaitu ringan ketika dihubungkan dengan kompleksitas pengaturan, *administrasi overhead*, dan pemakaian sumber. SQLite tidak memiliki *username* atau *login*. SQLite memiliki fitur-fitur sebagai berikut:

1. Tidak memerlukan *server*
 2. *Single File Database*
 SQLite mengemas seluruh *database* ke dalam suatu *single file* yang berisi *layout database* dan data actual yang berada pada table dan indeks untuk memudahkan *user* membuat, menyalin, ataupun mem-*backup database*.
 3. *Zero Configuration*
 SQLite tidak membutuhkan apapun untuk melakukan instalasi dan konfigurasi.
 4. *Embedded Device Support*
 Ukuran *code* dari SQLite bersifat kecil membuatnya cocok digunakan untuk *embedded system* yang berjalan terbatas pada system operasi.
 5. Fitur-fitur yang unik
 SQLite memungkinkan *user* untuk memasukkan nilai ke dalam kolom tanpa memperhatikan tipe data. Selain itu dapat memanipulasi lebih dai satu *database* pada satu waktu.
 6. *Compatible license*
 SQLite dan SQLite *code* tidak memiliki lisensi dan tidak dilindungi oleh *GNU's Not Unix* (GNU) *General Public License* (GPL) atau *license open source* sejenisnya. Hal berarti *user* dapat memodifikasinya dengan berbagai cara dan mendistribusikan dengan berbagai cara.
-

2.4 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai 24opular24 termasuk telpon genggam. Bahasa ini awalnya di buat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin *Virtual Java* (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, java dikenal pula dengan sloganya, "Tulis sekali, jalankan di manapun". Saat ini java merupakan bahasa pemrograman yang paling 25opular digunakan., dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web dan *icon* java dapat dilihat pada gambar 2.4.



Gambar 2.4 Logo Java

2.4.1 Java API

Java API terdiri dari tiga bagian utama:

1. *Java Standart Edition (SE)*, sebuah standart API untuk merancang aplikasi desktop dan *applet* dengan bahasa dasar gratis.
2. *Java Eterprose Edition (EE)*, sebuah API yang dirancang untuk aplikasi serves dengan mendukung basis data/database.

3. *Java Macro Edition (ME)*, sebuah API untuk merancang aplikasi yang berjalan pada alat kecil seperti telepon genggam, komputer, dan pager.

2.4.2 Java Virtual Machine

Java Virtual Machine (JVM) adalah spesifikasi untuk sebuah computer yang terdiri dari sebuah kelas pemanggil dan sebuah *interpreter*.Java yang mengeksekusikan kode arsitektur netral. Kelas pemanggil memanggil file.class dari kedua program Java dan API yang dieksekusi oleh *interpreter Java*. Interpreter Java mungkin sejumlah *just-inteme (JIT)* komputer yang menurunkan *bycode* arsitektur netral ke dalam bahasa mesin untuk *host* komputer.

2.4.3 Sistem Operasi Java

Sistem operasi biasanya ditulis dalam sebuah kombinasi kode bahasa C dan assembly, terutama disebabkan oleh kelebihan performa dari bahasa tersebut dan memudahkan berinteraksi dengan perangkat keras.

Satu kesulitan dalam merancang sistem basis bahasa adalah dalam hal proteksi memori, yaitu memproteksi sistem operasi dari pemakai program yang sengaja memproteksi pemakai program lainnya. Sistem operasi tradisional mengharapakan pada tampilan perangkat keras untuk menyediakan proteksi dari bahasa, sebagai hasilnya sistem basis bahasa menginginkan pada saat perangkat keras kecil, yang memungkinkan kekurangan tampilan perangkat keras yang menyediakan proteksi *memory*.

2.4.4 Dasar Pemograman Java

Java adalah generasi kedua dari *Java Platform* (generasi awalnya adalah *Java Development*), dengan kata lain Java berdiri diatas sebuah mesin *interpreter* yang diberi nama JVM.

Platform Java terdiri *Library*, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah *computer, debugger*, dan perangkat yang lain yang dipaket dalam *Java Development Kit (JDK)*. Agar sebuah program Java dapat dijalankan maka ekstensi "Java" harus dikompilasi menjadi file *bycode* yang dibutuhkan *JRE (Java Runtime Enviroment)* yang memungkinkan program Java, hanya menjalankan dan tidak membuat kode baru lagi.

2.5 Algoritma Pencarian Kata

Algoritma pencarian *string* atau sering disebut juga pencocokan *string* adalah algoritma untuk melakukan semua kemunculan *string* pendek.

Pencocokan *string* merupakan permasalahan paling sederhana dari semua permasalahan *string* lainnya, dan dianggap sebagai bagian dari pemrosesan data, penkompresian data, dan temu balik informasi. Teknik untuk menyelesaikan permasalahan pencocokan *string* biasanya akan menghasilkan implikasi langsung ke aplikasi *string* lainnya.

Algoritma pencocokan *string* dikategorikan menjadi tiga bagian menurut arah pencariannya, tiga kategori tersebut adalah :

1. Dari arah yang paling alami, dari kiri ke kanan, yang merupakan arah untuk membaca, algoritma yang termasuk kategori ini adalah :
 - a. Algoritma Brute Force
 - b. Algoritma dari Morris dan Pratt, yang kemudian dikembangkan oleh Knuth, Morris dan Pratt (KMP)
2. Dari kanan ke kiri, arah yang biasanya menghasilkan hasil terbaik secara praktikal contohnya adalah :
 - a. Algoritma Boyer dan Moore, yang memungkinkan banyak menjadi algoritma turbo Boyer-Moore, algoritma tuned Boyer-Moore, dan algoritma Zhu-Takaoka.
3. Dan kategori terakhir, dari arah yang ditentukan secara spesifik oleh algoritma tersebut, arah ini menghasilkan hasil terbaik secara teoritis, algoritma yang termasuk algoritma ini adalah :
 - a. Algoritma Colusss.i
 - b. Algoritma Crochemore-Perrin

2.5.1 Algoritma Brute Force

Algoritma Brute force adalah sebuah pendekatan yang lempang (*straight forward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma brute force memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Algoritma brute force seringkali lebih mudah

diimplementasikan daripada algoritma yang lebih canggih, dan karena kesederhanaannya, kadang-kadang algoritma *brute force* dapat lebih mangkus.

Kamus *mobile* yang akan dirancang akan menggunakan algoritma *brute force* sebagai solusi dalam penyelesaian masalah pencarian kata, seperti halnya dalam penggunaan kamus berbentuk buku para pengguna memanfaatkan cara penyajian kata yang sesuai dengan urutan abjadnya, sehingga para pengguna sangat terbantu dalam mencari kata yang ingin diterjemahkan. Pada kamus *mobile* penyajian kata yang sesuai dengan urutan abjadnya sangat sulit untuk direalisasikan karena terbatasnya ukuran layar dari perangkat *mobile* seperti *Handphone*. Oleh karena itu pada perancangan kamus *mobile* ini algoritma *brute force* diharapkan dapat menyelesaikan permasalahan tersebut. Proses pencarian kata dengan menerapkan algoritma yang digunakan, merupakan hal penting dalam pembuatan kamus *mobile* ini karena proses tersebut merupakan bagian inti dari penggunaan kamus *mobile* tersebut.

Dasar dari pembuatan algoritma ini, adalah dengan menggunakan algoritma pencarian string dengan metode *brute force*, kemudian menambahkan beberapa fitur untuk merekam hasil pencarian *string* pada suatu saat, untuk kemudian dapat dipakai untuk melakukan pemilihan keputusan *search* yang akan diambil. Algoritma pencarian *string* dengan metode *brute-force* melakukan pencocokan karakter per karakter, dengan cara, pada pertama kali eksekusi program, algoritma akan mencocokkan karakter pertama *pattern* dengan karakter pertama teks, dan jika cocok, maka algoritma akan mencocokkan karakter kedua *pattern* dengan karakter kedua teks. Tetapi jika tidak terdapat kecocokan, maka pencocokan akan dimulai lagi dari karakter kedua karakter teks, dan dicocokkan dengan karakter pertama teks. berikut ditampilkan algoritma pencarian *string* dengan metode *brute-force*.

Ditinjau dari segi implementasi:

1. Mula-mula *pattern* dicocokkan pada awal teks.
 2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern* dengan karakter yang bersesuaian di dalam teks sampai:
 - a. semua karakter yang dibandingkan cocok atau sama (pencarian berhasil),
atau
-

BAB III

PERANCANGAN SISTEM

Bab ini berisi tentang perancangan dari sistem yang dibuat. Perancangan ini Meliputi pokok-pokok bahasan dari perancangan, baik secara perangkat lunak atau perangkat keras yang dibuat.

3.1 Spesifikasi Aplikasi

Disesuaikan dengan tujuan awal aplikasi, spesifikasi aplikasi ini adalah sebagai berikut :

1. Pembuatan aplikasi dibuat dengan menggunakan bahasa java. Bahasa pemograman java digunakan untuk mendukung pembuatan aplikasi kamus bahasa Suku sekatak Buji berbasis android menggunakan algoritma *Brute Force*.
2. IDE yang digunakan adalah Eclipse Luna perangkat lunak digunakan dalam pembuatan aplikasi android.
3. Aplikasi Berjalan pada Android versi 2.3 (Gingerbread) sampai ke versi ke 4.0 (ICS).
4. Aplikasi kamus ini berjalan pada Android versi 2.0 sampai ke 4.0 karena para *Developer* untuk mobile lebih banyak menggunakan versi-versi tersebut untuk handset yang mereka keluarkan.
5. Aplikasi ini tidak terhubung dengan internet.

Aplikasi kamus ini hanya menggunakan *Datubase* internal, dikarenakan data-data yang diinputkan hanya bersifat kata, dan database internal saja cukup memungkinkan untuk melakukan penyimpanan data.

3.2 Analisa Sistem

3.2.1 Analisa Perangkat Keras (*Hardware*)

Pembuat aplikasi kamus bahasa Indonesia-tidung, indonesia-bulungan dan indonesia-dayak berusu menggunakan *hardware* antara lain :

1. Laptop ASUS A43S
 - a. Intel Core-i3 2310M (2,1 GHz)

- b. VGA NVIDIA GFORCE GT450M
 - c. LCD 14" LED
 - d. 500 GB HDD
2. Modem Smartfreen
 3. Smartphone OPPO Neo K versi R831K_11_140911
 - a. *Android OS, v4.2.2(Jelly bean) Capacitive touchscreen*
 - b. *Dual core 1.3GHz*
 - c. *Internal 1 GB storage, 1GB RAM*
 - d. *Eksternal 8 GB microSD, up to 32GB*

3.2.2 Analisa Perangkat Lunak (*Software*)

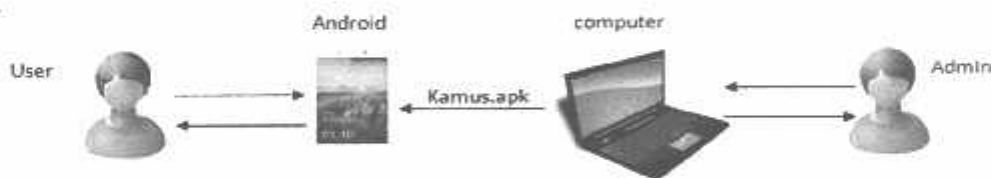
Selain *hardware*, dalam pembuatan aplikasi diperlukan beberapa *software* pendukung, yaitu :

1. Sistem Operasi Windows 7 Ultimate 64-bit
2. Eclipse Luna
3. JDK 6 Windows i586

3.3 Perancangan Aplikasi

3.3.1 Blok Diagram Pembuatan Aplikasi

Blok diagram pada pembuatan aplikasi ini berfungsi untuk menjelaskan alur dari sistem dalam pembuatan aplikasi kamus, seperti terdapat pada gambar 3.1.



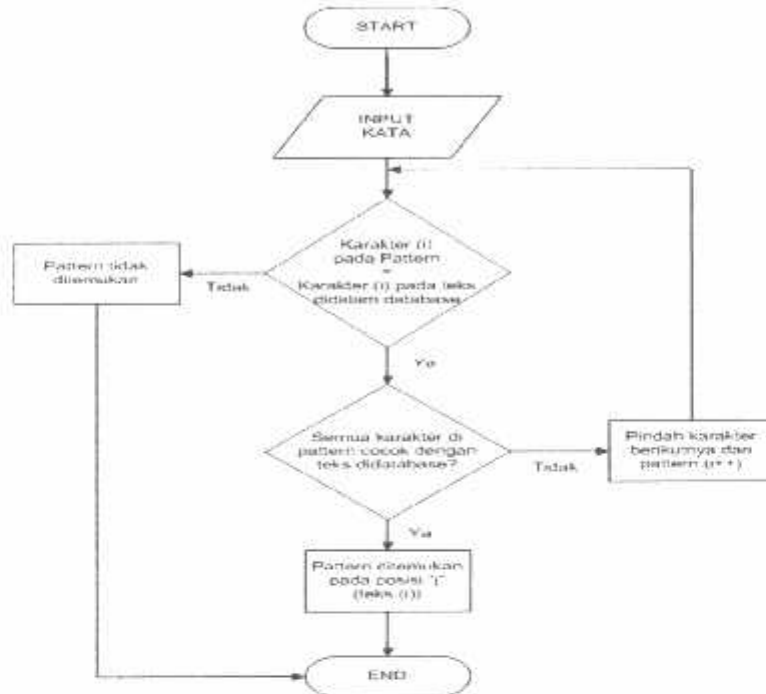
Gambar 3.1 Blok Diagram

Keterangan :

1. Admin membuat aplikasi Kamus dengan komputer menggunakan eclipse untuk membuat program android dan SQLite untuk perancangan database servernya.
2. Admin memperbaharui database pada aplikasi kamus.
3. Aplikasi dimasukkan kedalam hp android
4. User menggunakan aplikasi melalui hp android.
5. Aplikasi yang dijalankan *user* berupa aplikasi *offline*.

3.3.2 Flowchart Sistem

Untuk memperjelas alur dari sistem, maka akan digambarkan dalam bentuk flowchart. *Flowchart* menjelaskan rancangan urutan proses yang terjadi pada aplikasi seperti ditunjukkan pada gambar 3.3.



Gambar 3.3 Flowchart Sistem

Penjelasan dari alur sistem *flowchart* adalah sebagai berikut :

1. *User* atau pengguna menginput kata yang akan dicari artinya.
2. Selanjutnya kata yang diinputkan oleh user tadi akan diterjemahkan kemudian dicocokkan ke database, dalam hal ini proses pencocokan kata dilakukan menggunakan algoritma *BRUTE FORCE* pada proses terjemahan.
3. Apabila karakter yang diinputkan terpenuhi maka pencarian atau proses terjemahan akan berhasil.

3.3.3 Penerapan Algoritma Brute Force Pada Kamus Bahasa Daerah Sekatak Buji.

Berikut ini merupakan algoritma *brute force* yang diterapkan pada kamus bahasa daerah sekatak buji. Pada metode ini dilakukan pencocokan jumlah karakter yang dicari sama *pattern* nya, kemudian dicocokkan jumlahnya satu persatu. Jika sudah habis dan sesuai maka hasilnya akan ditampilkan dan nilainya

sama dengan 1, jika tidak ditemukan maka hasilnya sama dengan 0. Seperti pada gambar 2.7 berikut ini :

```
public int brute_force(String pattern, String kata)
{
    char[] p;
    char[] k;
    int x, j;
    int flag = 0;
    int length_pattern = pattern.length();
    int length_kata = kata.length();

    p = pattern.toCharArray();
    k = kata.toCharArray();

    for(x=0; x<length_pattern;){
        j = 1;
        while((j<length_kata)&&(k[x+j] == p[j])){
            j = j+1;
        }
        if(j>=length_kata){
            flag = 1;
            break;
        }
    }
}
```

Gambar 3.4 Algoritma Brute Force pada Kamus Daerah Sekatak Buji.

3.3.4 Perancangan Tabel Database

Berikut perancangan tabel-tabel yang digunakan pada aplikasi, pada tabel 3.1 merupakan tabel dari database aplikasi dimana semua data-data tersimpan di database internal.

Tabel 3.1 Tabel Database Aplikasi

Field	Type	Length	Keterangan
Id	Int	10	<i>primary</i>
Indonesia	Text	100	
Tidung	Text	100	
Bulungan	Text	100	
Berusu	Text	100	
Terjemahan	Text	100	

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Tahap implementasi dan pengujian terhadap aplikasi Kamus Daerah Sekatak Buji dengan metode *Brute force*, dilakukan setelah seluruh proses pengumpulan data berupa bahasa daerah suku yang ada di Sekatak Buji yang ditujukan untuk melakukan pembuatan kamus dianggap telah mencukupi untuk dilanjutkan ke proses pembuatan aplikasi dengan mengkombinasikan data-data pendukung aplikasi atau elemen-elemen multimedia dengan menggunakan perangkat lunak *Eclipse Luna* guna mempermudah proses pembuatan aplikasi.

Perangkat pendukung lainnya juga diperlukan, disini penulis menggunakan perangkat tambahan berupa handphone dengan *Operating System Android 4.4.2*, perangkat pendukung untuk menjalankan file *Apk* dari aplikasi yang sudah dibuat.

4.1 Kebutuhan Sistem

Kebutuhan yang digunakan untuk pembuatan aplikasi Kamus dengan metode *Brute Force*, meliputi perangkat lunak dan perangkat keras yang dihubungkan dengan system operasi. Perangkat lunak yang dibutuhkan untuk membuat aplikasi Kamus Bahasa Daerah Sekatak Buji menggunakan metode *Brute Force*, meliputi:

1. Perangkat lunak Eclipse Luna, berfungsi sebagai editor guna membuat source code dan juga penggabungan semua fungsi.
2. Perangkat pendukung Android SDK, untuk memberikan dukungan Operating System yang berjalan pada handphone android nantinya.

Sedangkan perangkat keras yang digunakan dalam membuat aplikasi kamus bahasa daerah, meliputi:

1. Mainboard Intel Core i3.
2. Processor Intel Core i3-2310M CPU@ 2.10 GHz.
3. Memory 4,00 GB.
4. VGA NVIDIA GEFORCE GT 540 1 GB.
5. Handphone OPPO android KitKat 4.4.2.

4.2 Pengujian Aplikasi Pada Perangkat Android

Perangkat android yang digunakan untuk pengujian aplikasi ini adalah OPPO Neo K versi R831K_11_140911 yang menggunakan android versi 4.2.2 dengan ukuran layar 4.45 inch (Jelly Bean). Pengujian aplikasi dapat dilihat pada Gambar 4.1.



Gambar 4.1 Perangkat Android OPPO Neo K

4.2.1 Tampilan Awal Aplikasi Setelah di Instal

Tampilan awal aplikasi ini merupakan tampilan yang telah terinstal pada perangkat android. Dalam hal ini penulis menampilkan aplikasi yang telah terinstall pada perangkat dengan Operating System versi 4.2.2. Jika aplikasi berhasil diinstall menandakan bahwa aplikasi yang dibuat dapat berjalan pada perangkat android. Tampilan pada perangkat android yang telah terinstall dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan Aplikasi Setelah di Instal Pada Perangkat Android

4.2.2 Tampilan Awal Aplikasi Pertama Kali

Tampilan awal merupakan tampilan keseluruhan aplikasi yang meliputi menu-menu pada aplikasi dan teks Kamus Bahasa Daerah Suku Sekatak Buij. Menu yang terdapat pada aplikasi berupa Menu pilihan KAMUS, ABOUT dan KELUAR. Tampilan awal dapat dilihat pada Gambar 4.3.

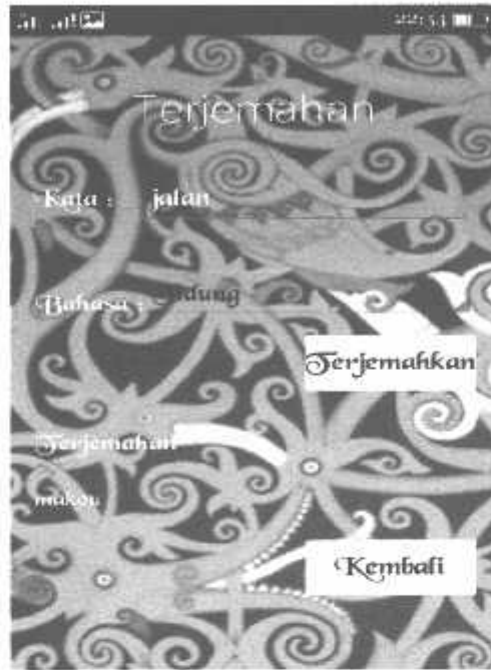


Gambar 4.3 Tampilan Awal Aplikasi

4.2.3 Tampilan Aplikasi Ketika di Lakukan Terjemahan.

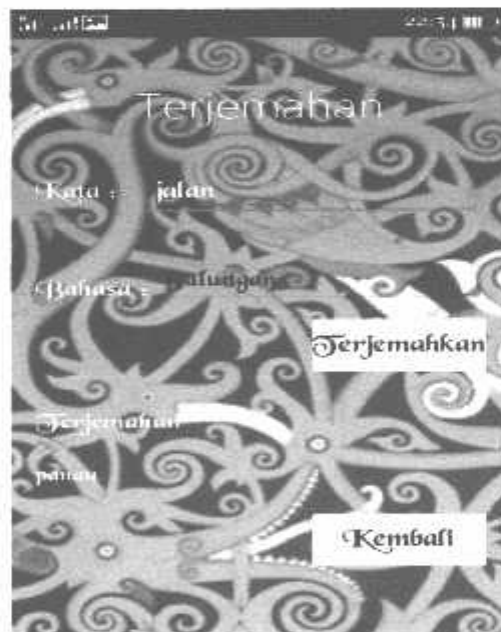
Menjelaskan proses Terjemahan dengan cara menginputkan kata dari bahasa Indonesia ke bahasa daerah suku Tidung, Bulungan dan Dayak Berusu yang dicari terjemahannya dimana tampilan Output adalah hasil terjemahan dari kata yang tadi diinputkan. Setelah menginputkan kata kita tinggal memilih apakah akan diterjemahkan kedalam bahasa suku Tidung, Bulungan atau Dayak Berusu.

Berikut ini dilakukan pengujian dengan menginputkan kata “ Jalan “ untuk diterjemahkan kedalam bahasa Suku Tidung dimana hasil dari terjemahan kata tersebut adalah “ Makou “ seperti pada Gambar 4.4.



Gambar 4.4 Hasil Terjemahan Kata Kedalam Bahasa Tidung

Selanjutnya dilakukan pengujian dengan menginputkan kata “Jalan” untuk diterjemahkan kedalam bahasa Suku Bulungan dimana hasil dari terjemahan kata tersebut adalah “Panau” seperti pada Gambar 4.5.



Gambar 4.5 Hasil Terjemahan Kata Kedalam Bahasa Bulungan

Selanjutnya dilakukan pengujian dengan menginputkan kata “Jalan” untuk diterjemahkan kedalam bahasa Suku Berusu dimana hasil dari terjemahan kata tersebut adalah “Mokou” seperti pada Gambar 4.6.



Gambar 4.6 Hasil Terjemahan Kata Kedalam Bahasa Dayak Berusu

4.2.4 Tampilan Menu About

Tampilan menu about yaitu tampilan yang berisikan tentang Pembuat, Alamat, *Email* serta No *Handphone* dan photo pembuat Kamus Bahasa Daerah. Tampilan menu about dapat dilihat pada Gambar 4.7.



Gambar 4.7 Tampilan Menu *About*

4.2.5 Tampilan Aplikasi Ketika Tidak Ditemukan Terjemahannya

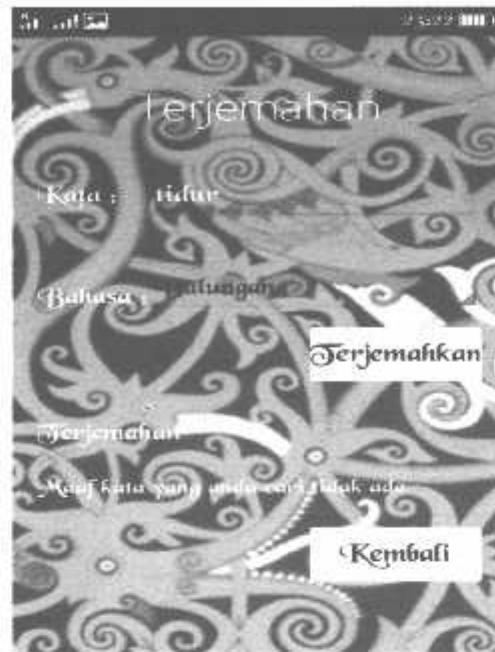
Menjelaskan proses Terjemahan dengan cara menginputkan kata dari bahasa Indonesia ke bahasa daerah suku Tidung, Bulungan dan Dayak Berusu yang dicari terjemahannya dimana jika kata yang diinputkan tidak ditemukan atau kata yang dicari tidak terdapat didalam *database* pada kamus bahasa daerah sekatak buji setelah dilakukan pencocokan string dengan menggunakan metode *brute force*.

Berikut ini dilakukan pengujian dengan menginputkan kata “ Tidur “ untuk diterjemahkan kedalam bahasa Suku Tidung dimana hasil dari terjemahan kata tersebut tidak ditemukan seperti pada Gambar 4.8.



Gambar 4.8 Hasil Terjemahan Kata kedalam Bahasa Tidung

Berikut ini dilakukan pengujian dengan menginputkan kata “ Tidur “ untuk diterjemahkan kedalam bahasa Suku Bulungan dimana hasil dari terjemahan kata tersebut tidak ditemukan seperti pada Gambar 4.9.



Gambar 4.9 Hasil Terjemahan Kata kedalam Bahasa Bulungan

Berikut ini dilakukan pengujian dengan menginputkan kata “ Tidur “ untuk diterjemahkan kedalam bahasa Suku Dayak Berusu dimana hasil dari terjemahan kata tersebut tidak ditemukan seperti pada Gambar 4.10.



Gambar 4.10 Hasil Terjemahan Kata kedalam Dayak Berusu

4.3 Hasil Pengujian

4.3.1 Tabel Fungsional Program

Peroses ini dilakukan pada device OPPO Neo K versi R831K_11_140911 yang menggunakan android versi 4.2.2 dengan ukuran layar 4.45 inch Asus Zenfone 4 dengan ukuran layar 4 inch, Lenovo 389i dengan ukuran layar 4 inch, dan device Lenovo S850 android versi 4.4.2 dengan ukuran layar 5 inch.

Tabel 4.1 Hasil Pengujian Fungsional

Prosedur	Hasil	
	Sesuai	Tidak Sesuai
Menampilkan Menu Utama Aplikasi	✓	
Melakukan Terjemahan kata dari Bahasa Indonesia ke dalam Bahasa Tidung	✓	
Melakukan Terjemahan kata dari Bahasa Indonesia ke dalam Bahasa Bulungan	✓	
Melakukan Terjemahan kata dari Bahasa Indonesia ke dalam Bahasa Berusu	✓	
Tampilan Menu About	✓	
Total	100 %	0 %

Pada Tabel 4.1 Merupakan tabel dari pengujian aplikasi pada perangkat android yang digunakan oleh penulis, dimana pada hasil percobaan yang dilakukan oleh penulis aplikasi dapat berjalan dengan baik 100 %. Semua prosedur yang telah ditetapkan oleh penulis dapat berjalan dengan baik pada perangkat android yang digunakan oleh penulis.

4.3.2 Tabel Pengujian OS dan Resolusi Layar

Berikut ini merupakan hasil pengujian pada beberapa jenis handphone yang memiliki spesifikasi OS dan Resolusi layar yang berbeda dengan hasil 100% berjalan dengan baik, seperti yang ditunjukkan pada tabel 4.2 berikut :

Tabel 4.2 Tabel Pengujian OS dan Resolusi Layar.

No	Handphone	OS	Resolusi Layar	Berhasil	Gagal
1.	Lenovo 389i	4.2.2	4 inch 800 x 480	✓	
2.	Oppo Neo K	4.2.2	4.45 inch 854 x 480	✓	
3.	Asus Zenfone 4	4.4.2	4 inch 800 x 480	✓	
4.	Lenovo S850	4.4.2	5 inch 1280 x 720	✓	

4.3.3 Tabel Pengujian User

Pada bagian ini penulis memasukan data ke tabel dari hasil pengujian ke 10 user dengan random sebanyak sepuluh sample. *Sample* yang diujikan meliputi *user interface*, tingkat kemudahan, fitur-fitur didalam aplikasi, tingkat kepuasan pengguna dan juga aplikasi mampu memberikan hasil. Adapun hasil dari beberapa kriteria pertanyaan yang diajukan kepada user dengan penilaian seperti berikut:

Table 4.3 Pengujian User

No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	7	3	0
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	10	0	0
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	6	3	1
4	Apakah aplikasi ini sudah cukup	6	3	1

DAFTAR PUSTAKA

- [1] Live Coding 9 Aplikasi Android Buatan Sendiri Diterbitkan Oleh Arif Akbarul Huda 2013
 - [2] Cepat dan Mudah Membuat aplikasi Android Diterbitkan oleh Fadjar Hamidi F. (26 Juli 2014)
 - [3] Implementasi Algoritma Brute Force dalam Pencocokan Teks oleh May Aprina Saragih (20 Desember 2014)
 - [4] Membangun Database oleh Sulastri (Desember 2014)
 - [5] Wawancara Masyarakat Bulungan, Tidung dan Dayak Berusu (28 Agustus-20 September 2014)
-

LAMPIRAN



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG**


**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Afriansyah Romadani
NIM : 10.18.167
Jurusan : Teknik Informatika S-1
Judul : Aplikasi Kamus Bahasa Daerah Suku Sekatak Buji
Kalimantan Utara Menggunakan Metode Brute Force

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)
pada:

Hari : Rabu
Tanggal : 25 Februari 2015

**Panitia Ujian Skripsi
Ketua Majelis Penguji**


Joseph Dedy Irawan, ST, MT
NIP.19740416 2005011022

Anggota Penguji :

Penguji Pertama

Penguji Kedua


Febriana Santi W, S.Kom, M.Kom
NIP. 1031000425


Ahmad Faisol, ST, MT
NIP.P. 1031000431



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

BUKTI BIMBINGAN SKRIPSI

Nama : Afriansyah Romadani
Nim : 10.18.167
Program Studi : Teknik Informatika S-1
Judul : Aplikasi Kamus Bahasa Daerah Suku Sekatak Buji Kalimantan Utara Menggunakan Metode Brute Force Berbasis Android

No.	Tanggal	Uraian	Paraf Pembimbing
1.	2/12/2014	Draft Bab 1,2,3	
2	3/12/2015	Revisi bab 1, 2	
3	9/2/2015	Makalah semhas	
4	20/2/2015	Laporan progres	

Malang, 8 Juni 2015

Dosen Pembimbing I

Ali Mahmudi, B.Eng.P.hd

NIP.P.1031000429



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

BUKTI BIMBINGAN SKRIPSI

Nama : Afriansyah Romadani
Nim : 10.18.167
Program Studi : Teknik Informatika S-1
Judul : Aplikasi Kamus Bahasa Daerah Suku Sekatak Buji
Kalimantan Utara Menggunakan Metode Brute
Force Berbasis Android

No.	Tanggal	Uraian	Paraf Pembimbing
1	18/11/2014	Revisi Bab 1,2,3	
2.	20/11/2014	Revisi Bab 1,2,3	
3	3/12/2014	Demo Progres Program	
4	3/12/2014	Revisi Bab-1	
5	4/12/2014	Revisi Blok Diagram	

Malang, 8 Juni 2015

Dosen Pembimbing II

Yosep Agus Franoto, ST,MT

NIP.P.1031000432



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

FORMULIR PERBAIKAN UJIAN SKRIPSI

Nama : Afriansyah Romadani
NIM : 10.18.167
Jurusan : Teknik Informatika S-1
Judul : Aplikasi Kamus Bahasa Daerah Suku Sekatak Buji Kalimantan Utara Menggunakan Metode Brute Force.

Dosen Penguji	Revisi	Paraf
Dosen Penguji 1	1. Revisi Laporan	
Dosen Penguji 2	1. Revisi Program : Tambahkan Tombol Back di setiap Sub Menu 2. Revisi Program : Tambahkan Background sesuai dengan Ciri Khas Suku Sekatak.	

Anggota Penguji :

Penguji Pertama

Febriana Santi W. S.Kom, M.Kom
NIP. 1031000425

Penguji Kedua

Ahmad Faisol, ST, MT
NIP.P. 1031000431

Mengetahui

Dosen Pembimbing I

Ali Mahmudi, B.Eng, PhD
NIP.P. 1031000429

Dosen Pembimbing II

Yosep Agus Pranoto, ST, MT.
NIP: P.103100432

DB Helper.java :

```
package com.example.database;

import java.util.ArrayList;
import java.util.List;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;
import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;

public class DBHelper extends SQLiteAssetHelper{

    private static final String DATABASE_NAME = "kamus";
    private static final int DB_VER = 1;
    private static final String TABLE_NAME = "kamus";
    private static final String ID = "_id";

    private static final String INDONESIA = "indonesia";
    private static final String TIDUNG = "tidung";
    private static final String BULUNGAN = "bulungan";
    private static final String BERUSU = "berusu";

    private static DBHelper dbInstance = null;
    private static SQLiteDatabase db;
    public DBHelper(Context context){
        super(context, DATABASE_NAME, null, DB_VER);
    }

    public static DBHelper getInstance(Context context){
        if (dbInstance == null){
```

```

        dbInstance = new DBHelper(context);
        db = dbInstance.getWritableDatabase();
    }
    return dbInstance;
}
@Override
public synchronized void close(){
    super.close();
    if(dbInstance != null){
        dbInstance.close();
    }
}

public List<Bahasa> getBahasa(String indonesia){
    List<Bahasa> listBahasa = new ArrayList<Bahasa>();
    Cursor cursor = null;

        cursor = db.query(TABLE_NAME, new String[]{ID, INDONESIA, TIDUNG,
        BULUNGAN, BERUSU}, INDONESIA + "=?", new
        String[]{String.valueOf(indonesia)}, null, null, null, null);
        if (cursor.getCount() == 0){

return listBahasa;
}
else {
    Log.d("Cursor rows count size: ", String.valueOf(cursor.getCount()));
    cursor.moveToFirst();
    do {

        Bahasa data = new Bahasa();
        data.setId(cursor.getInt(cursor.getColumnIndexOrThrow(ID)));
        data.setIndonesia(cursor.getString(cursor.getColumnIndexOrThrow(INDONESIA)));

```

```
data.setTidung(cursor.getString(cursor.getColumnIndexOrThrow(TIDUNG)));  
data.setBulungan(cursor.getString(cursor.getColumnIndexOrThrow(BULUNGAN)));  
data.setBerusu(cursor.getString(cursor.getColumnIndexOrThrow(BERUSU)));
```

```
listBahasa.add(data);
```

```
}while(cursor.moveToNext());
```

```
return listBahasa;
```

```
}
```

```
}
```

```
}
```

Bahasa.java :

```
package com.example.database;
```

```
public class Bahasa {
```

```
    private int id;
```

```
    private String indonesia;
```

```
    private String tidung;
```

```
    private String bulungan;
```

```
    private String berusu;
```

```
    public Bahasa(){
```

```
}
```

```
public Bahasa(int id, String indonesia, String tidung, String bulungan, String berusu)
```

```
{
```

```
    this.id = id;
```

```
        this.indonesia = indonesia;
```

```
        this.tidung = tidung;
```

```
        this.bulungan = bulungan;
```

```
        this.berusu = berusu;
```

```
}
```

```
    public Bahasa(String indonesia, String tidung, String bulungan, String berusu){
```

```
        this.indonesia = indonesia;
```

```
        this.tidung = tidung;
```

```
        this.bulungan = bulungan;
```

```
        this.berusu = berusu;
```

```
}
```

```
    public int getId(){
```

```
        return id;
```

```
}
```

```
    public void setId(int id){
```

```
        this.id = id;
```

```
}  
    public String getIndonesia(){  
        return indonesia;  
    }  
public void setIndonesia(String indonesia){  
    this.indonesia = indonesia;  
}  
    public String getTidung(){  
        return tidung;  
    }  
public void setTidung(String tidung){  
    this.tidung = tidung;  
}  
    public String getBulungan(){  
        return bulungan;  
    }  
public void setBulungan(String bulungan){  
    this.bulungan = bulungan;  
}  
    public String getBerusu(){  
        return berusu;  
    }  
public void setBerusu(String berusu){  
    this.berusu = berusu;  
}  
}
```

Fungsi Button Terjemahan :

```
public void terjemah(){

    Button terjemahan = (Button) findViewById(R.id.button_terjemah);
    db = DBHelper.getInstance(this);
    terjemahan.setOnClickListener(new OnClickListener()
    {
        @Override

        public void onClick(View v) {

            EditText Ekata = (EditText) findViewById(R.id.edit_terjemahan);
            Spinner S_bahasa = (Spinner) findViewById(R.id.bahasa_spinner);
            String kata = Ekata.getText().toString();
            String kata2 = kata.toLowerCase();
            String bhsa = S_bahasa.getSelectedItem().toString();

            int check ;

            listBahasa = db.getBahasa(kata2);
            Bundle bun = new Bundle();
            if (listBahasa.isEmpty()){
                bun.putString("hasil", "Maaf kata yang anda cari tidak ada");
                bun.putString("indonesia", kata2);
            }
            else{
                bahasa = listBahasa.get(gBahasa);
                check = brute_force(bahasa.getIndonesia(), kata2);
                if (check == 1){
                    if (bhsa.matches("Berusu")){
```

```
        bun.putString("hasil", bahasa.getBerusu());
    }
    else if (bhsa.matches("Bulungan")){
    bun.putString("hasil", bahasa.getBulungan());
    }
        else if (bhsa.matches("Tidung")){
        bun.putString("hasil", bahasa.getTidung());
        }
    bun.putString("indonesia", bahasa.getIndonesia());
    }
    else{
        bun.putString("hasil", "Maaf kata yang anda cari tidak ada");
        bun.putString("indonesia", kata2);
    }
    }
    Intent i = new Intent(Terjemahan.this, Terjemahan.class);
    i.putExtras(bun);
    startActivity(i);
    finish();
}
});
}
```

Fungsi Algoritma Brute_Forcenya :

```
public int brute_force(String pattern, String kata)

{
    char[] p;
    char[] k;
    int x, j;
    int flag = 0;

    int length_pattern = pattern.length();
    int length_kata = kata.length();

    p = pattern.toCharArray();
    k = kata.toCharArray();

    for(x=0; x<length_pattern;){
        j = 1;
        while((j<length_kata)&&(k[x+j] == p[j])){
            j = j+1;
        }
        if(j>=length_kata){
            flag = 1;
            break;
        }
    }
    return flag;
}
```

Fungsi klik Tombol Menu Terjemahan :

```
public void terjemahan(){  
  
    Button terjemahan = (Button) findViewById(R.id.button_kamus);  
    terjemahan.setOnClickListener(new OnClickListener() {  
  
        @Override  
        public void onClick(View v) {  
            // TODO Auto-generated method stub  
            Intent intent = new Intent(MainActivity.this, Terjemahan.class);  
            startActivity(intent);  
            finish();  
        }  
    });  
}
```

Fungsi Klik Tombol About ke Menu About :

```
public void about(){  
  
    Button about = (Button) findViewById(R.id.button_about);  
    about.setOnClickListener(new OnClickListener() {  
  
        @Override  
        public void onClick(View v) {  
            // TODO Auto-generated method stub  
            Intent intent = new Intent(MainActivity.this, About.class);  
            startActivity(intent);  
            finish();  
        }  
    });  
}
```

Fungsi Klik Tombol Keluar untuk Keluar Aplikasi :

```
public void btn_keluar(){  
  
    Button btn_keluar = (Button) findViewById(R.id.button_exit);  
    btn_keluar.setOnClickListener(new OnClickListener() {  
  
        @Override  
        public void onClick(View arg0) {  
            // TODO Auto-generated method stub  
            System.exit(0);  
            finish();  
        }  
    });  
}
```

Fungsi Tombol Back di Halaman Terjemahan :

```
public void back(){

    Button about = (Button) findViewById(R.id.button_back);
    about.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(Terjemahan.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
    });
}
```

No. Kuisisioner : 1
Nama : BARETA P.L.K.
Alamat : JOMBANG
Email : BARETA.KUSUMIA@yahoo.com



No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?			✓
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?	✓		
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?		✓	

No. Kuisisioner : 2

Nama : Lufti PRADINA

Alamat : Jl. Perumahan no. 7

Email : LuftiPradina@gmail.com




No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?		✓	
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?	✓		
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?		✓	

No. Kuisioner : 3

Nama : Sigit Pangestu

Alamat : Jl. Bend. Sigara-gara

Email : SigitPangestu27@gmail.com



No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?		✓	
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	✓		
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?	✓		
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?		✓	

No. Kuisisioner : A

Nama : Setyawan Fajar Rizki

Alamat : Kediri. Jalan Koranyanyar Nganygo #

Email : ~~Setyawan~~ rizki.setyawan91@gmail.com



No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?		✓	
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?	✓		
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		

No. Kuisisioner : 5

Nama : Didik horiyanto

Alamat : Malang

Email : Mldidik@gmail.com



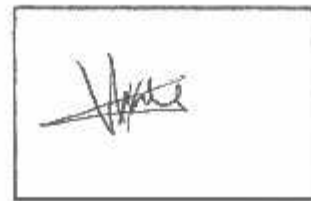
No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?		✓	
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	✓		
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?	✓		
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		

No. Kuisisioner : 6

Nama : Verbyan Arizona

Alamat : Jln-Suasembada barat 19 no.64

Email : Verbyan-phantom@ymail.com



No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	✓		
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?		✓	
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		

No. Kuisisioner : 7
Nama : Andika Alhairil A.Hz
Alamat : Lumajang
Email : Andika0n@gmail.com



No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	✓		
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?	✓		
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		

No. Kuisisioner : 8
Nama : Lulu Rarnaldi
Alamat : Jln Hockey No. 40.
Email : lrcos.008@gmail.com



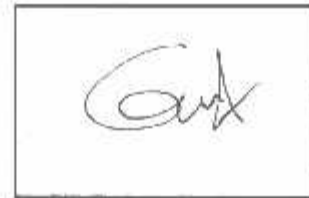
No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?		✓	
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?		✓	
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		

No. Kuisisioner : 9

Nama : Muhammad Saifudin Zuhri

Alamat : Jl. Hoki no 18

Email : zuhri91@gmail.com



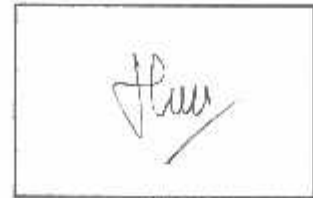
No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?	✓		
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	✓		
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?		✓	
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		

No. Kuisisioner : 10

Nama : Dhani Swara . Prayoga

Alamat : Jl Wijaya No 64 Singosari

Email : dhani swara prayoga @ gmail . com



No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Menurut Anda apakah tampilan dari aplikasi ini sudah baik ?		✓	
2	Bagaimana penilaian anda, apakah kemampuan aplikasi dalam melakukan pencarian kata ?	✓		
3	Bagaimana menurut Anda, terhadap fitur – fitur yang ada ?	✓		
4	Apakah aplikasi ini sudah cukup memudahkan pengguna dalam belajar bahasa daerah sekatak buji ?			✓
5	Apakah secara keseluruhan aplikasi ini sudah cukup baik menurut anda ?	✓		