**PREDIKSI PENCARIAN KATA DENGAN *ALGORITMA LEVENSHTEIN DISTANCE* DI DALAM KAMUS BAHASA MANGGARAI, INDONESIA DAN INGGRIS BERBASIS *ANDROID***

**Darius Obi Januardi1, Alexius Endy Budianto, S.Kom.,MM2, Muhammad Priyono Tri S, M. Eng3**

Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Kanjuruhan Malang

Jl. S. Supriadi No.48 Malang, Telp (0341) 801488

Email : [obijanuaradi@gmail.com1](mailto:obijanuaradi@gmail.com1), [endybudianto@unikama.ac.id2](mailto:endybudianto@unikama.ac.id2), [m.priyono.ts@unikama.ac.id3](mailto:m.priyono.ts@unikama.ac.id3)

**Abstrak**

Perkembangan teknologi saat ini meningkat pesat, hal ini ditandai dengan semakin banyaknya fungsi pada perangkat *mobile* tersebut. Ini menjadi sebuah evolusi perangkat *mobile* dalam hal ini telepon genggam yang menggunakan teknologi *smartphone* yang kemampuanya hampir mirip dengan komputer.

Peran bahasa yang sangat penting untuk dipelajari sangat efektif jika diaplikasikan dalam sebuah sistem *android* oleh karena itu penulis menemukan sebuah masalah dalam membangun komunikasi antara masyarakat di Manggarai karena daerah ini menjadi pusat wisata lokal maupun manca Negara, untuk mengatasi masalaah ini penulis mencoba membangun sistem aplikasi Kamus tiga bahasa yaitu bahasa Manggarai, bahasa Indonesia dan Bahasa Inggris berbasis android, lengkap dengan Fitur *Autocomplete* dan Metode *levenshtein Distance* untuk membantu pengguna dalam melakukan pencarian kata yang lebih mudah dan efisien.

**Kata Kunci** : Prediksi Pencarian Kata; *Algoritma Levenshtein Distance*; Kamus.

1. **PENDAHULUAN**

Indonesia yang terdiri dari berbagai macam suku bangsa juga memiliki kebudayaan yang beraneka ragam. Keanekaragaman itu dapat dilihat dari beragamnya adat istiadat, bahasa, tarian dan aksara.

Bahasa daerah merupakan kekayaan budaya yang patut dilestarikan, Salah satunya adalah bahasa daerah Manggarai. Dengan Teknologi yang semakin canggih maka semakin terjamin pula kelancaran dalam proses pendidikan dan pendalaman Bahasa itu sendiri, asalkan kita sebagai generasi mudah punya niat untuk melestarikan bahasa yang kita miliki dengan tujuan yang baik dan tentunya mengadung unsur edukatif untuk generasi selanjutnya.

Bahasa Manggarai adalah salah satu bahasa yang digunakan oleh beberapa kelompok masyarakat dibelahan timur Indonesia tepatnya di Pulau Flores NTT. Beberapa faktor yang mendorong saya untuk membagun sebuah aplikasi kamus Bahasa Inggris­-Indonesia-Manggarai karena letak daerah Manggarai yang memiliki situs pariwasata warisan Dunia diantaranya Pulau

Komodo yang akhirnya resmi ditetapkan menjadi satu dari tujuh keajaiban dunia atau New 7 Wonders Of Nature pada tanggal 16 mei 2012

dan Wae Rebo yang menjadi salah satu warisan budaya yang diresmikan UNESCO tahun 2012.

Untuk lebih memudahkan pengguna dalam proses pencarian kata di dalam kamus tiga bahasa Manggarai-Indonesia-Inggris pengembangan aplikasi ini ditambahkan fitur autocomplete dalamnya. Autocomplete melibatkan program yang dapat melakukan prediksi terhadap sebuah kata atau frasa yang penggun ingin tulis tanpa harus menulis keseluruhan kata atau frasa secara lengkap (Banowosari, 2014). Fitur autocomplete yang berfungsi untuk memberikan pilihan saran kata ketika mengetik ini merupakan fitur yang disediakan baik itu di browser, search engine, pemrosesan kata, maupun pada baris perintah. Selain itu fitur ini mampu memberikan saran kata yang pengguna inputkan tanpa harus ditulis secara keseluruhan (Yao, 2013).

Levenshtein distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Pada perhitungan edit distance dari metode ini didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string.

Pada penelitian yang dilakukan oleh Riya Mary Abraham (2014) algoritma Levenshtein Distance diimplementasikan dalam pencarian di cloud environment. Hasil yang dapat dibandingkan dengan pencarian tradisional, algoritma ini sedikit lebih unggul. Sehingga tidak salah jika algoritma ini efektif untuk digunakan dalam berbagai aplikasi atau kegunaan lain. Dari penelitian Riya Mary Abraham tersebut algoritma Levenshtein Distance memiliki nilai yang tinggi dalam memprediksi kemiripan katan yang dimasukan pengguna.

1. **LANDASAN TEORI**
   1. **Prediksi Pencarian Kata atau *Fitur* *Autocompleate***

*Autocomplete* adalah fungsi yang menampilkan perkiraan kata yang akan dimasukkan tanpa harus mengetikkan keseluruhan kata. Fitur ini juga merupakan salah satu bentuk *search* yang sering kita jumpai pada saat melakukan pencarian di google. *Autocomplete* atau *autocompletion* atau *word completion* adalah fasilitas yang disediakan oleh berbagai *web browser, email-programs, search engine interface, source code editors, database query tools, word Processor,* dan *command line interpreters. Autocomplete* juga tersedia untuk atau sudah terintegrasi di dalam *text editor.*.

*Autocomplete* bekerja ketika penulis menulis huruf pertama atau beberapa huruf/karakter dari sebuah kata. Program yang melakukan prediksi akan mencari satu atau lebih kemungkinan kata sebagai pilihan. Jika kata yang dimaksud ada dalam pilihan itu, maka penulis dapat memilih itu. Jika kata yang dimaksud tidak ada dalam pilihan prediksi maka penulis harus menulis huruf/karakter selanjutnya. Ketika penulis memilih pilihan kata yang ada dalam daftar pilihan kata prediksi maka kata yang dipilih tersebut akan disisipkan pada teks.

Dalam *source code editor*, *autocomplete* menyederhanakan struktur regular dari sebuah bahasa pemrograman. Biasanya hanya ada sejumlah kata yang berarti dalam konteks saat ini atau *namespace*, seperti nama dari sebuah variabel atau nama dari sebuah fungsi. Contoh dari *code completion* adalah desain *IntelliSense* dari *Microsoft*. Dia menampilkan sebuah *pop-up list* yang berisi prediksi yang mungkin dari karakter yang dimasukkan oleh pengguna untuk dipilih pengguna yang sesuai dengan apa yang diinginkan pengguna.Fitur ini sangat berguna dalam pemrograman berorientasi objek karena seringkali *programmer* tidak tahu secara benar *member* apa saja yang sebuah *class* punya (Banowosari, 2014).

*Autocomplete* melibatkan program yang dapat melakukan prediksi terhadap sebuah kata atau frasa yang pengguna ingin tulis tanpa harus menulis keseluruhan kata atau frasa secara lengkap (Banowosari, 2014). Keuntungan penggunaan *autocomplete* ini adalah komputer membantu pengguna untuk mengingat kata yang akan dimasukkan. Selain itu Fitur *autocomplete* ini pun sudah banyak digunakan diberbagai macam lingkup, seperti : mesin pencari, kamus digital, *browser*, bahkan dalam proses penulisan di aplikasi pemrograman seperti *dreamweaver*, *eclipse*, maupun yang lainnya sudah menerapkan fitur *autocomplete*. Hal ini tentu membantu bagi para *developer* pemula yang masih belum hapal dengan benar setiap sintak atau kode yang ada pada bahasa pemrograman tersebut.

*Autocomplete* sendiri ada beberapa jenis, yaitu : (1) *autocomplete suggestion* oleh Google. *Autocomplete* jenis ini digunakan oleh Google pada mesin pencariannya, di mana *autocomplete* ini akan menampilkan perkiraan kata atau frasa yang akan dimasukkan tanpa harus mengetikkan keseluruhan kata (Banowosari, 2014). *Autocomplete* ini melakukan pencarian sesuai dengan apa yang pengguna masukkan pertama kali, dan saran akan muncul berdasarkan kata ataupun huruf yang belum selesai diketik di kotak pencarian.

* 1. ***Algoritma Leventeins Distance***

Levenshtein distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Pada perhitungan edit distance dari metode ini didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B dengan menggunakan 3 macam operasi utama yang mencakupi operasi penyisipan (insertion), operasi penghapusan (deletion), operasi penggatian (subtitution) .

1. **Mekanisme Metode *Levenshtein Distance***

` Edit distance merupakan jumlah minimum point mutation yang diperlukan untuk merubah suatu string ke string yang lain. Point mutation tersebut adalah mengganti, menambah dan menghapus sebuah karakter. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini yaitu :

1. **Operasi Pengubahan Karakter**

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain. String ‘gimpunan’ menjadi ‘himpunan’, dalam kasus ini karakter ‘g’ yang terdapat pada awal string diganti dengan huruf ‘h’.

String 1 H i m p u n a n

String 2 **G** i m p u n a n

***Substitution***  **H**

1. **Operasi Penambahan Karakter**

Operasi penambahan karakter berarti menyisipkan karakter ke dalam suatu string. Contohnya string ‘disrit’ menjadi ‘diskrit’, dilakukan penyisipan karakter ‘k’ ditengah string. Penyisipan karakter tidak hanya dilakukan ditengah string, namun bisa disisipkan diawal maupun diakhir.

String 1 d i s k r i t

String 2 d i s **–** r i t

***Insertion*** **K**

1. **Operasi Penghapusan Karakter**

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu string. Contohnya string ‘matematikan’ karakter terakhir dihilangkan sehingga menjadi string ‘matematika’. Pada operasi ini dilakukan penghapusan karakter ‘n’.

String 1 m a t e m a t i k a **–**

String 2 m a t e m a t i k a n

***Deletion*** **N**

1. ***Dynamic Programming***

Pencarian edit distance dapat dilakukan dengan menggunakan beberapa metode, diantaranya algoritma dynamic programming dan hirschberg.

Dalam penelitian ini digunakan algoritma dynamic JUST TI, Volume 10 Nomor 1, Januari 2018: 58-62 programming. *Dynamic programming* merupakan suatu algoritma yang membagi problem menjadi sub-sub problem dimana solusi yang optimal dapat dicari dari sub problem tersebut. Sub-sub problem dipecahkan dan hasilnya kemudian disimpan dalam bentuk tabel. Penggunaan tabel dimana tiap sel berisi solusi yang dihitung berdasarkan algoritma tersebut. Algoritma ini dapat digunakan jika sub-sub problem tidak independent, artinya sub problem tersebut mempunyai irisan dengan sub problem yang lain. Pemecahan setiap sub-sub problem hanya dilakukan sekali dan kemudian disimpan dalam bentuk tabel. Dynamic programming biasanya diaplikasikan untuk problem optimasi, dimana problem-problem tersebut solusinya mempunyai banyak kemungkinan.

1. **Pengukuran *Similarity Value***

Setelah mendapatkan nilai edit distance dari dua dokumen yang dibandingkan selanjutnya menghitung presenstase kemiripan dari dua dokumen tersebut.

Untuk menghitung similarity value menggunakan persamaan sebagai berikut:

Diff adalah edit distance, terletak pada baris ke m dan kolom ke n.

CS adalah panjang string awal, ST adalah panjang  *string* target, dan Max(CS,ST) adalah panjang string terbesar antara string awal dan string target. Similarity value diasumsikan pada rentang 0 (nol) hingga 100 (seratus) persen, yang artinya nilai 100 adalah nilai maksimum yang menunjukan bahwa dua kata adalah sama identik. Pendekatan ini mampu digunakan untuk mengukur bobot similarity antar dua string berdasarkan susunan karakter.

1. **METODE PENELITIAN**
2. **Teknik Pengumpulan Data**

Metode pengumpulan data adalah sebuah metode tentang bagaimana dalam mengumpulkan data yang ada. Adapun metode pengumpulan yang digunakan adalah studi pustaka dimana pengumpulan dilakukan dengan cara meneliti literatur atau referensi yang mendukung dalam penyusunan Tugas Akhir ini.

1. **Dokumen pendukung**

Dokumen pendukung dalam hal ini adalah sebuah buku atau dokumen penerjemah dari bahasa Manggarai, Indonesia Dan Inggris sebagai sebuah saran untuk mengumpulkan data

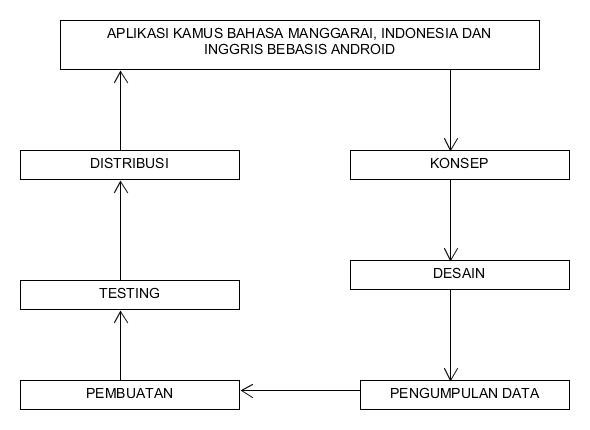
1. **Wawancara**

Wawancara dilakukan dengan pengumpulkan data melalui tanya jawab peneliti dengan narasumber. Wawancara pada penelitian ini digunakan untuk mengumpulkan data sebagai acuan analisis kebutuhan. Wawancara dilakukan kepada guru Bahasa Manggarai.

1. **Observasi**

Teknik pengumpulan data observasi dilakukan dengan mengamati langsung terhadap objek yang diteliti. Teknik observasi pada penelitian ini dilakukan untuk mengumpulkan data pengujian pada aspek performance efficiency, compatibility, dan reliability.

1. **Metode Perancangan**

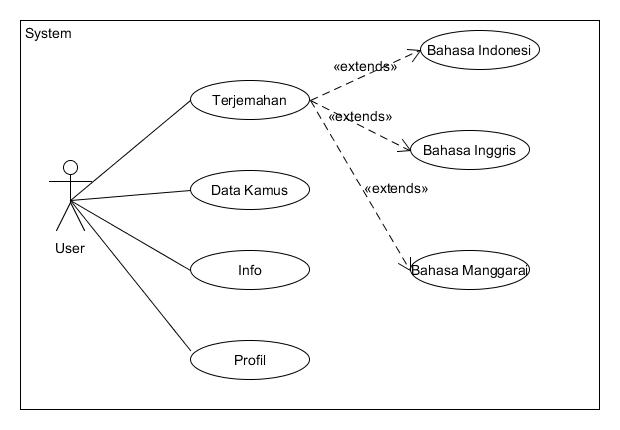


Gambar 3. 1Metode Perancangan System

1. **Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Aplikasi yang dibangun dimodelkan menggunakan *Unified Modeling Language* (UML), dan *tools* yang akan digunakan yaitu *use case diagram*, *activity diagram*, *sequence diagram,* dan *class diagram*.

1. ***Use Case Diagram***



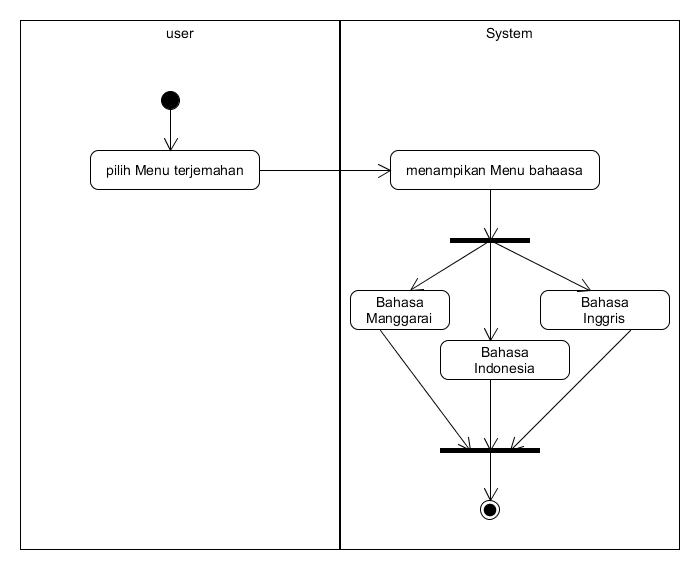
Gambar 4. 1 use case diagram pada aplikasi kamus tiga bahasa manggarai,Indonesia dan Inggris.

1. ***Actifity Digram***

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem proses atau proses bisnis. (Hendini, A, 2016.

Gambar-gambar berikut ini akan memberikan sedikit penjelasan dari peroses kerja atau sistem kerja pada aplikasi kamus tiga Bahasa manggaraai, Indonesia Dan Inggris.

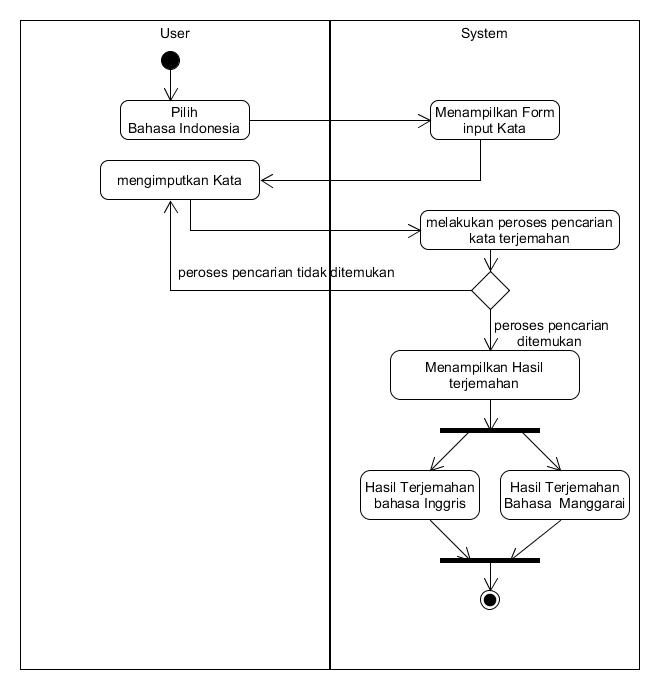
1. ***Actifity Diagram Menu* Terjemahan**



*Gambar 4. 2 Actifity Diagram pada menu Terjemahan*

1. ***Actifity Diagram* *Menu* Bahasa Indonesia**

Gambar 4.5 menunjukan *actifity Diagram* aplikasi kamus tiga bahasa Manggarai, Inggris dan Indonesia pada menu Bahasa Indonesia hingga berlanjut pada peroses berikutnya,dimana *system* akan menampilkan *layout* yang akan memberikan akses kepada *user* untuk mengetik kata terjemahan bahasa Indonesia, setelah itu *system* aplikasi akan menampilkan hasil terjemaahan.

****

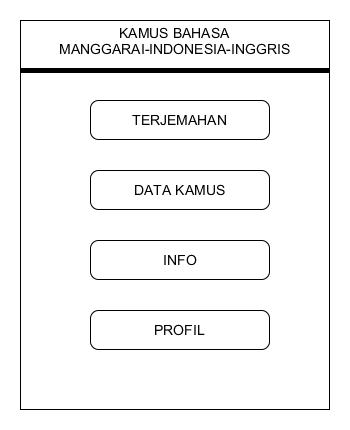
Gambar 4. 3 Actifity Diagram pada menu bahasa Indonesia

1. **Perancangan Antarmuka**

Perancangan antarmuka adalah tahapan pembuatan rancangan antarmuka untuk digunakan pada pembangunan aplikasi telepon yang dibagi menjadi empat bagian yaitu perancangan *form* menu utama, *form* terjemhan, *form* Data kamus , *form* Info dan *form* Profil.

1. *Form* Menu Utama

G*ambar* 4.12 menggambarkan *user* memilih menu yang akan dibuka sesuai kebutuhannya, terdapat empat menu, yaitu: Terjemahan, Data Kamus, Info dan Profil.



Gambar 4. 4 Form menu utama

1. *Form* Menu Bahasa

Gambar 4.13 yaitu *form* yang menunjukan tampilan dari sistem aplikasi ketika user menekan tombol terjemahan pada menu utama kemudian sistem akan melanjutkan peroses kemenu Pilih bahasa setelah sistem menampikan menu pilih bahasa, user menekan tombol bahasa mana yang ingin diterjemahkan maka sistem akan menanmpilkan form yang mengizikan user untuk mengetik kata terjemahan setelah itu sistem akan menampilkan hasil terjemahan.



1. **IMPLEMENTASI**

Implementasi merupakan tahap pengembangan rancangan menjadi kode program. Pada awal bagian ini dijabarkan spesifikasi implementasi yang dilakukan berdasarkan hasil pada tahap perancangan. Implementasi perancangan antarmuka terbagi menjadi beberapa bagian utama yaitu:

1. Form Menu Utama

Gambar 4.18 yaitu form menu utama, user memilih menu yang akan dibuka sesuai kebutuhannya, terdapat 4 menu, yaitu form Terjemahan , Data Kamus, Info, dan Profil.



Gambar 4. 5 Form menu utama

1. Form Menu Terjemahan

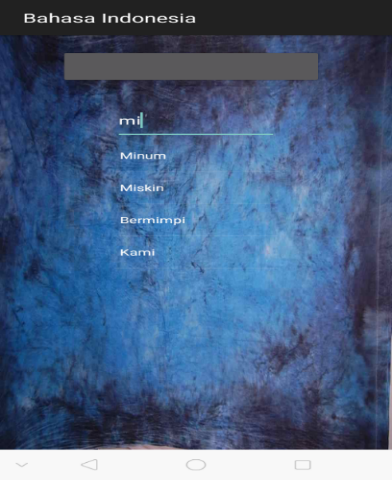
Gambar 4.19 menunjukan form menu Terjemahan, setelah user menekan tombol terjemahan sistem akan menampilkan *layout* baru berupa *Butto* Bahasa Indinesia, *Button* Bahasa Inggris Dan Button Bahasa Manggarai.



Gambar 4. 6 Menu Bahasa

1. *Fitur Auto Complete Text Dengan Levenstein Distance*

`Gambar 4.20 yaitu *menu fitur auto complete text,*  ketika user hendak memasukkan kata yang akan diterjemahkan, maka fitur auto completebtext akan berfungsi. Fitur ini menampilkan kata pencarian bisa berdasarkan Huruf Inputan awal dan seterusnya sedangkan peran *Levenshtein Distance* Menyisipkan kata dari Kiri dan kanan atau menampilkan kata yang terdapat di dalam database. Misalkan kita memasukkan huruf “mi”, maka pada tampilan layar akan muncul kata yang memiliki huruf “mi” .



Gambar 4. 7 hasil fitur auto complete text dan levenshtein Distance.

1. Form Hasil Terjemahan

Gambar 4.21 menunjukan form hasi terjemahan kata, user akan melihat hasil terjemahan dari kata yang telah dinputkan. Misalkan mencari terjemahan kata “Handsome” pada form bahasa inggris maka dengan sendirinya sistem akan menampikan hasil terjemahan dari bahasa Manggarai Dan Bahasa Indonesia.



Gambar 4. 8 Form Hasil Terjemahan

* **Pengujian Sistem Aplikasi**

Setelah dilakukan pengujian terhadap aplikasi Kamus Tiga Bahasa, maka dibawah ini merupakan hasil yang didapat dari proses pengujian dengan metode *black box*.

1. Pengujian Aplikasi Kamus Tiga Bahasa

Hasil pengujian yang dilakukan dengan metode *black box* pada sistem aplikasi Kamus Tiga Bahasa Ditunjukkan pada Tabe

Tabel 4. 1 Pengujian Aplikasi Kamus Tiga Bahasa

|  |  |  |
| --- | --- | --- |
| **Hal Yang Diuji** | **Butir Uji** | **Hasil Pengujian** |
| Menu Terjemahan aplikasi kamus tiga Bahasa | Menampilkan menu bahasa yang akan diterjemahkan | Valid |
| Menu Data Kamus | Menampikan Data yang ada didalam Kamus | Valid |
| Menu Info | Menampilkan Deskripsi singkat tentang ketiga bhasa yaitu Manggarai, Inggris Dan Indonesia | Valid |
| Menu Profil | Menampilkan Profil kamus | Valid |
| *Autocomplete* | Dapat menampilkan *autocomplete* | Valid |
| Pemberian saran | Dapat menampilkan saran | Valid |

1. Pengujian *Autocomplete* Dengan *Algoritma Levenstein Distance*

Pengujian *autocomplete* merupakan pengujian untuk mengetahui apakah *autocomplete* dapat memunculkan pilihan kata atau tidak. Hasil pengujian dari *autocomplete* di aplikasi Kamus Tiga Bahasa ditunjukkan pada Tabel 4.5

*Tabel 4. 2 Pengujian Autocomplete Dengan Algoritma Levenstein Distance*

|  |  |  |
| --- | --- | --- |
| **Kata inputa** | **Muncul/ Tidak** | **Saran Yang Muncul** |
| Ma | muncul | makan, marah, manis, matahari, memaafkan, teman, Memasak, Rumah dan Dimana. |
| So | Muncul | Sombong dan Sore |
| ha | muncul | Hair, Handsome dan Hand |
| Ki | muncul | Tawa, Tanggal, Tadang dan Retang |
| Ba | Muncul | Ba, Bakok, Kreba dan Reba |
| Si | Muncul | Sit, Sing, Beside |
| Ga | Muncul | Garden, Arogant |
| Mo | Muncul | Molas, Hemong |
| Ko | Muncul | Kolang, Toko, Lako, Hanang Koe, Bakok |
| Ge | Muncul | Gereng, Gelang, Senget, Lengge |

1. **PENUTUP**
   1. **Kesimpulan**
2. Fitur *autocomplete* text pada aplikasi Kamus tiga bahasa Manggarai, Indonesia Dan Inggris berbasis Android dapat berjalan sesuai dengan yang diharapkan oleh pembuat aplikasi.
3. Aplikasi kamus tiga bahasa ini mempermudah pengguna dalam menggunakanya, karena aplikasi ini dibangun dengan metode *Levenshtein Distance* yang berfungsi sebagai pemberian saran kata.
4. Dapat menciptakan suatu aplikasi yang bisa dipergunakan oleh orang banyak orang karena aplikasi ini diterapkan untuk smartphon.
5. **Saran**
6. Fitur pada aplikasi ini diharapkan bisa dikembangkan lagi, sehingga lebih mengefisienkan user ketika menggunakannya.
7. Untuk penelitian selanjutnya, aplikasi ini dapat dikembangkan di seluruh daerah.
8. Algoritma yang digunakan pada aplikasi Kamus tiga bahasa ini dapat dikembangkan dengan algoritma atau metode yang lain. Kemudian bandingkan antara *Levenshtein Distance* dengan algoritma yang lain.

**DAFTAR PUSTAKA**

Anonim. Tanpa Tahun. *AutocompleteTextView. http://developer.android.com/*. Diakses 19 Oktober 2015.

Abraham, Riya Mary. 2014. Use Of Edit Distance Algorithm To Search A Keyword In CloudEnvironment. International Journal Of Database Theory And Application. (7) 6 : 223-232.

Banowosari, Lintang Y., dkk. 2014. Analisis pada Fitur Autocomplete Suggestion dan Semantic pada Pencarian di Mesin Pencari Google. Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2014) Vol. 8 Oktober 2014. Diakses 6 Maret 2014.

Bagus Ida.2018.Optimasi Pencarian Kata Pada Kamus Aneka Bahasa Menggunakan Algoritma Levenstein Distance.Jurnal Sistem Dan Informatika.Vol. 12,NO.

Chen, Yoke Yie.,dkk. 2014. E-mail Hoax Detection System Using Levenshtein distance Method. Journal Of Computer. (9) 2 : 445-446.

Darni Yulia, 2018. Aplikasi Sistem Pencarian Lagu menggunakan Algoritma Levenshtein Distance, JTIS, Volume 1 Nomor 3, Desember 2018, ISSN : 2614 – 3070, E-ISSN : 2614 – 3089

Yao, Zhiqiang. 2013. Implementation Of The Autocomplete Feature Of The Textbox Based On Ajax And Web Service. Journal of Computers. (8) 9 : 21992201.