

IMPLEMENTASI KAMUS KEDOKTERAN DENGAN METODE INTERPOLASI (*INTERPOLATION*) DAN MENCARI KEMIRIPAN KATA MENGGUNAKAN ALGORITMA LEVENSHEIN *DISTANCE* PADA PERANGKAT ANDROID

Irna Rahayu^{*1}, Bambang Pramono², Anita Puspita Dewi³

^{*1,2}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

³STMIK Catur Sakti, Kendari

e-mail: ^{*1}rahayu.nina66@gmail.com, ²bambangpramono09@gmail.com, ³uppiet77@yahoo.com

Abstrak

Kamus adalah kitab yang berisi kata-kata dan arti atau keterangan yang disusun secara alfabetik. Kedokteran adalah suatu ilmu dan seni yang mempelajari tentang penyakit dan cara-cara penyembuhannya. Tujuan dari penelitian ini adalah untuk membangun aplikasi kamus kedokteran berbasis Android. Aplikasi *mobile* dapat dijadikan sarana untuk membuat kamus terjemahan dari berbagai istilah kedokteran yang ada, karena dapat digunakan kapanpun dan dimanapun.

Algoritma dan metode yang digunakan adalah metode pencarian interpolasi untuk proses pencarian dan Algoritma Levenshtein *Distance*. Metode *Interpolation Search* hanya dapat dilakukan pada *list* yang telah berurut dan berada pada struktur *array* dan data yang dicari diperkirakan ada didalam *list*, teknik ini menemukan item dengan memperkirakan seberapa jauh kemungkinan item berada dari posisi saat itu dan pencarian berikutnya. Algoritma Levenshtein *Distance* menghitung jumlah operasi *string* paling sedikit yang diperlukan untuk mentransformasikan suatu *string* menjadi *string* lain.

Hasil dari penelitian ini berupa Aplikasi kamus kedokteran berbasis Android yang dapat menerjemahkan kata dengan cepat dan dapat menampilkan pilihan kata lain apabila kata yang dicari tidak ditemukan.

Kata Kunci : Kamus Kedokteran, *Interpolation Search*, *Levenshtein Distance* dan Android

Abstract

Dictionary is a book that contains the words and the meaning or information arranged alphabetically. Medicine is a science and an art that study about the disease and ways of healing. The purpose of this reSearch is to build Android-based medical dictionary application. The mobile application can be used as a means to create a dictionary translation of various medical terms that exist, because it can be used anytime and anywhere.

Algorithms and methods that used were Interpolation Search for the Search process and Levenshtein Distance algorithm. Interpolation Search method can only be done on a list that has sequential and located on the array structure and the data sought is thought to exist in the list, this technique find items by estimating how far the possibility of the items are from the current position and the next quest. Levenshtein Distance algorithm calculates the least number of string operations needed to transform one string into another string.

The results of this study in the form of Android-based medical dictionary application that can translate words quickly and can display options other words, if the word you are looking for was not found.

Keywords : Medical Dictionary, *Interpolation Search*, *Levenshtein Distance* and Android

1. PENDAHULUAN

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Kamus berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mempunyai pedoman sebutan, asal – usul (etimologi) suatu perkataan dan juga contoh penggunaan bagi suatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi dalam kamus. Kamus adalah kitab yang berisi kata-kata dan arti atau keterangan yang disusun secara alfabetik [1].

Kamus memiliki berbagai macam jenis, sesuai dengan isi yang terkandung di dalamnya. Ada kamus bahasa baik dwi bahasa atau tri bahasa, ada juga kamus istilah, misalnya kamus istilah populer yang menerangkan kata-kata ilmiah. Serta ada juga jenis kamus lain yang menjadi pedoman dan disiplin ilmu tertentu, misalnya kamus komputer dan kamus kedokteran seperti yang akan penulis paparkan yakni kamus kedokteran. Kamus kedokteran merupakan kamus yang mencakup istilah-istilah kedokteran, termasuk di dalamnya mengenai istilah penyakit, obat-obatan, istilah medis dan peralatan yang biasa dipakai untuk praktek kesehatan dan kedokteran.

Dengan semakin berkembangnya zaman, ilmu pengetahuan dan teknologi, beberapa jenis kamus ada di perpustakaan atau yang beredar di beberapa toko buku dirasa masih kurang memenuhi kebutuhan manusia. Kamus identik dengan bentuk buku tebal yang sulit dibawa dan kurang praktis dalam menggunakannya. Demikian juga yang dialami beberapa mahasiswa kedokteran atau praktisi pada bidang kedokteran yang lain, sangat kesulitan bila harus membawa kamus kedokteran kemanapun dia pergi, maka kiranya aplikasi ini sangat membantu para mahasiswa kedokteran, praktisi di bidang kedokteran ataupun masyarakat secara umum yang ingin mencari makna istilah kedokteran secara praktis dan efisien.

Hal ini dirasa perlu untuk mempermudah pencarian istilah kedokteran, disamping itu dengan dibuatnya aplikasi ini kamus yang dapat diakses kapan saja oleh mahasiswa kedokteran, praktisi bidang kedokteran ataupun masyarakat secara

umum hanya dengan menggunakan perangkat android. Dalam aplikasi kamus ini teknik pencarian menggunakan metode interpolasi dan proses pencocokan *string* dengan menggunakan algoritma Levenshtein *Distance*.

Kamus ini disusun menggunakan proses *Searching* didalamnya, yakni *Interpolation Search*. Sama seperti *Binary*, teknik ini hanya dapat dilakukan pada list yang telah terurut dan berada pada struktur array dan data yang dicari diperkirakan ada di dalam *list*. Teknik ini menemukan item dengan memperkirakan seberapa jauh kemungkinan item berada dari posisi saat itu dan pencarian berikutnya. Dalam aplikasi kamus ini digunakan juga algoritma untuk menghitung jumlah kemiripan antara dua *string* yaitu dengan menggunakan algoritma Levenshtein *Distance*.

2. METODE PENELITIAN

2.1 Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Kamus berfungsi untuk membantu seseorang mengenal perkataan baru. Kamus juga mempunyai pedoman sebutan, asal-usul (etimologi) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus. Kamus adalah kitab yang berisi kata-kata dan arti atau keterangan yang disusun secara alfabetik [1].

Kata kamus diserap dari bahasa Arab *qamus* (قاموس), dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *Ωκεανός* (*okeanos*) yang berarti “samudra”. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan, khususnya pengetahuan bahasa, yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan khazanah yang memuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya [2].

2.2 Kedokteran

Kedokteran adalah suatu ilmu dan seni yang mempelajari tentang penyakit dan cara-cara penyembuhannya. Ilmu kedokteran

adalah cabang ilmu kesehatan yang mempelajari tentang cara mempertahankan kesehatan manusia dan mengembalikan manusia pada keadaan sehat dengan memberikan pengobatan pada penyakit dan cedera. Ilmu ini meliputi pengetahuan tentang sistem tubuh manusia dan penyakit serta pengobatannya, dan penerapan dari pengetahuan tersebut [3].

2.3 Metode Interpolasi

Interpolation Search adalah algoritma pencarian yang lebih efisien daripada algoritma *Binary* dan *Sequential Search*. Hal ini dikarenakan algoritma ini tidak perlu menjelajahi setiap elemen dari tabel. Kerugiannya adalah algoritma ini hanya bisa digunakan pada tabel yang elemennya sudah terurut baik menaik maupun menurun. Sama seperti *Binary*, teknik ini hanya dapat dilakukan pada list yang telah terurut dan berada pada struktur array dan data yang dicari diperkirakan ada di dalam list. Teknik ini menemukan item dengan memperkirakan seberapa jauh kemungkinan item berada dari posisi saat itu dan pencarian berikutnya. Teknik ini juga dilakukan pada list yang sudah terurut. Misalkan *Kunci* : kunci yang akan dicari, $K[max]$: nilai kunci yang paling besar, $K[min]$: nilai kunci yang paling kecil, Max : nilai indeks paling besar dan Min : nilai indeks paling kecil, maka *interpolation Search* ditunjukkan oleh persamaan (1).

$$posisi = \frac{kunci - k[min]}{k[max] - k[min]}(max - min) + min \quad (1)$$

Adapun kegunaan lain dari interpolasi adalah untuk menaksir harga-harga tengah antara titik data yang sudah tepat. Interpolasi mempunyai orde atau derajat. Berikut adalah alur pencarian interpolasi :

1. Menentukan banyaknya *record array* (k)
2. Menentukan nilai awal,
 $min = 0 ; max = k - 1$
3. Menentukan data posisi tengah *mid*,

$$mid = min + \frac{(kunci - k[min]) * (max - min)}{k[max] - k[min]}$$

4. Membandingkan data yang dicari(kunci) dengan *mid*.
5. Jika Kunci yang didapat lebih kecil dari pada nilai tengah, proses dilanjutkan dengan posisi $max = mid - 1$.

6. Jika Kunci yang didapat lebih besar dari pada *mid*, proses dilanjutkan dengan posisi $min = mid + 1$.
7. Jika $mid = data$ yang dicari(kunci) , maka $index = mid$, selesai [4].

2.4 Algoritma Levenshtein Distance

Algoritma Levenshtein, atau sering disebut dengan *Levenshtein Distance* atau *Edit Distance* merupakan algoritma pencarian jumlah perbedaan *string* yang ditemukan oleh Vladimir Levenshtein, seorang ilmuwan Rusia, pada tahun 1965. Algoritma ini digunakan secara luas dalam berbagai bidang, misalnya mesin pencari, pengecek ejaan (*spell checking*), pengenalan pembicaraan (*speech recognition*), pengucapan dialek, analisis DNA, pendeteksi pemalsuan, dan lainlain. Algoritma ini menghitung jumlah operasi *string* paling sedikit yang diperlukan untuk mentransformasikan suatu *string* menjadi *string* yang lain [5].

Oleh [6], untuk menghitung *Levenshtein Distance* dari dua buah *string* dapat dilakukan secara rekursif. Formula dari algoritma rekursif untuk *string a* dan *string b* diberikan oleh $lev_{a,b}(|a|, |b|)$ pada persamaan (2).

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & ; \text{if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1 & a_i \neq b_j \end{cases} & ; \text{Untuk yang lain} \end{cases} \quad (2)$$

Misalkan i adalah indeks dari *string a* pada suatu tahap dan j adalah indeks *string b*. Formula yang diberikan akan membandingkan *string* dari a dan *string b* dari indeks paling akhir. Untuk setiap tahap, akan diputuskan operasi mana yang lebih optimal. Untuk mengecek nilai solusi dari operasi penghapusan, maka status i akan dikurangi sejumlah satu. Untuk mengecek nilai solusi dari operasi penyisipan, maka status j akan dikurangi dengan sejumlah satu. Untuk operasi penggantian, maka status i dan status j masing-masing akan dikurangi dengan sejumlah satu. Bila karakter ke i dari *string a* dan karakter ke j dari *string b* sama, maka operasi penggantian tidak perlu

diperhitungkan, tetapi status i dan status j dapat langsung dikurangi dengan jumlah satu. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma Levenshtein *Distance* yaitu :

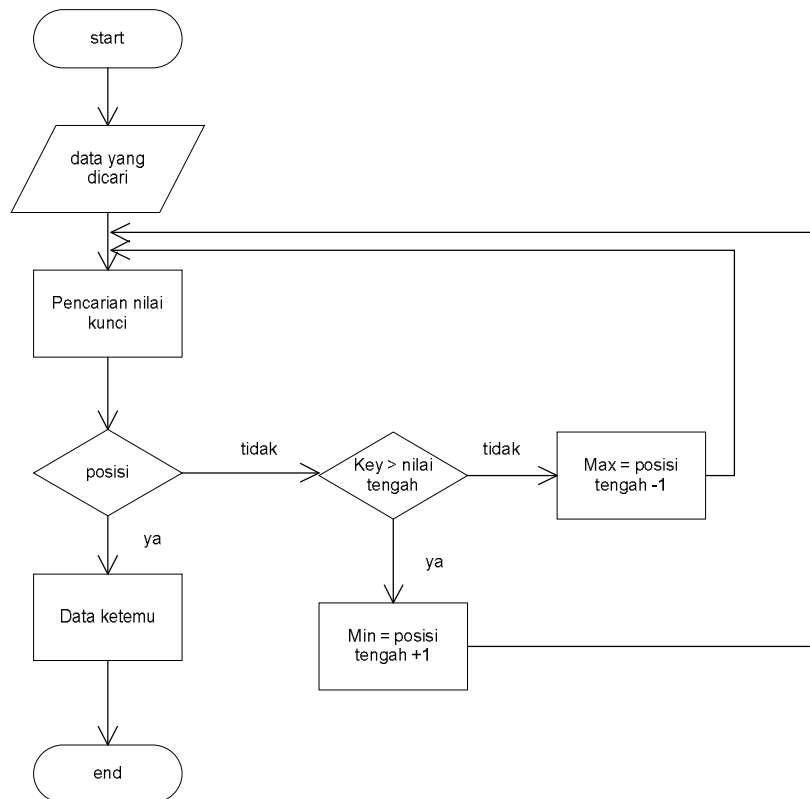
1. Operasi Pengubahan Karakter

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan *string* “yang” menjadi “yang”.

Dalam kasus ini karakter “m” diganti dengan huruf “n”.

2. Operasi Penambahan Karakter

Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu *string*. Contohnya *string* “kepad” menjadi *string* “kepada”, dilakukan penambahan karakter “a” di akhir *string*. Penambahan karakter tidak hanya dilakukan di akhir kata, namun bisa ditambahkan di awal maupun disisipkan di tengah *string*.



Gambar 1 *Flowchart* metode Interpolasi [4]

3. Operasi Penghapusan Karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu *string*. Contohnya *string* “baru” karakter terakhir dihilangkan sehingga menjadi *string* “baru”. Pada operasi ini dilakukan penghapusan karakter “r” [6].

Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter string awal dan *string* target dan diberikan nilai *cost*. Nilai *cost* pada ujung kanan bawah menjadi

nilai *edit Distance* yang menggambarkan jumlah perbedaan dua *string* [5].

Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter string awal dan *string* target dan diberikan nilai *cost*. Nilai *cost* pada ujung kanan bawah menjadi nilai *edit Distance* yang menggambarkan jumlah perbedaan dua *string* [6].

Gambar 1 adalah gambar yang menjelaskan tentang *flowchart* metode pencarian interpolasi, dimulai dari menginput kata yang akan dicari, lalu

pencarian nilai kunci dengan menggunakan rumus interpolasi, jika kunci yang dicari sama dengan posisi maka data ketemu dan proses terhenti, dan apa bila kunci yang didapat lebih besar dari nilai tengah maka nilai $min = posisi + 1$, dan apa bila kunci yang didapat lebih kecil dari nilai tengah maka nilai $max = posisi - 1$, lalu diulangi proses pencarian nilai kunci dengan menggunakan rumus interpolasi sampai kata kunci ditemukan [4].

Tabel 1 menunjukkan contoh beberapa istilah dalam kamus kedokteran.

Tabel 1 Beberapa istilah dalam kamus kedokteran

Indeks	Konversi	Kata	Arti Kata
[0]	0	abatement	kekurangan derajat nyeri atau gejala penyakit
[1]	1	abdomen	1)Bagian diantara sekat rongga badan dan pinggul. 2) bagian tubuh yang terletak diantara toraks dan pelvis dan di dalamnya terdapat rongga abdomen dan visera. 3) perut; rongga tubuh dibawah diafragma berisi alat pencernaan, pemuahan dan pembiakan
[2]	2	abiotropy	kehilangan vitalitas dari jaringan tertentu secara progresif, yang ke arah hereditif degeneratif
[3]	3	Ablation	1)

			pembuangan penghancuran, pemisahan atau pelepasan ekstirpasi. 2) terutama dengan cara memotong
[4]	4	abnormal	tidak normal, atau menyimpang dari yang lazim

a. Kunci pencarian (konversi) ? 1

$$\begin{aligned} \text{Posisi [index]} &= \frac{kunci-k[\min]}{k[\max]-k[\min]}(max - min) + min \\ &= \frac{1-0}{9-0}(9 - 0) + 0 \\ &= 1 \end{aligned}$$

Kode [1] == kunci?

Ya => terdapat dalam *database* indeks ke [1] yaitu abdomen.

b. Kunci pencarian (konversi) ? 2

$$\begin{aligned} \text{Posisi [index]} &= \frac{kunci-k[\min]}{k[\max]-k[\min]}(max - min) + min \\ &= \frac{2-0}{9-0}(9 - 0) + 0 \\ &= 2 \end{aligned}$$

Kode [2] == kunci?

Ya => terdapat dalam *database* indeks ke [2] yaitu abiotropy.

c. Kunci pencarian (konversi) ? 3

$$\begin{aligned} \text{Posisi [index]} &= \frac{kunci-k[\min]}{k[\max]-k[\min]}(max - min) + min \\ &= \frac{3-0}{9-0}(9 - 0) + 0 \\ &= 3 \end{aligned}$$

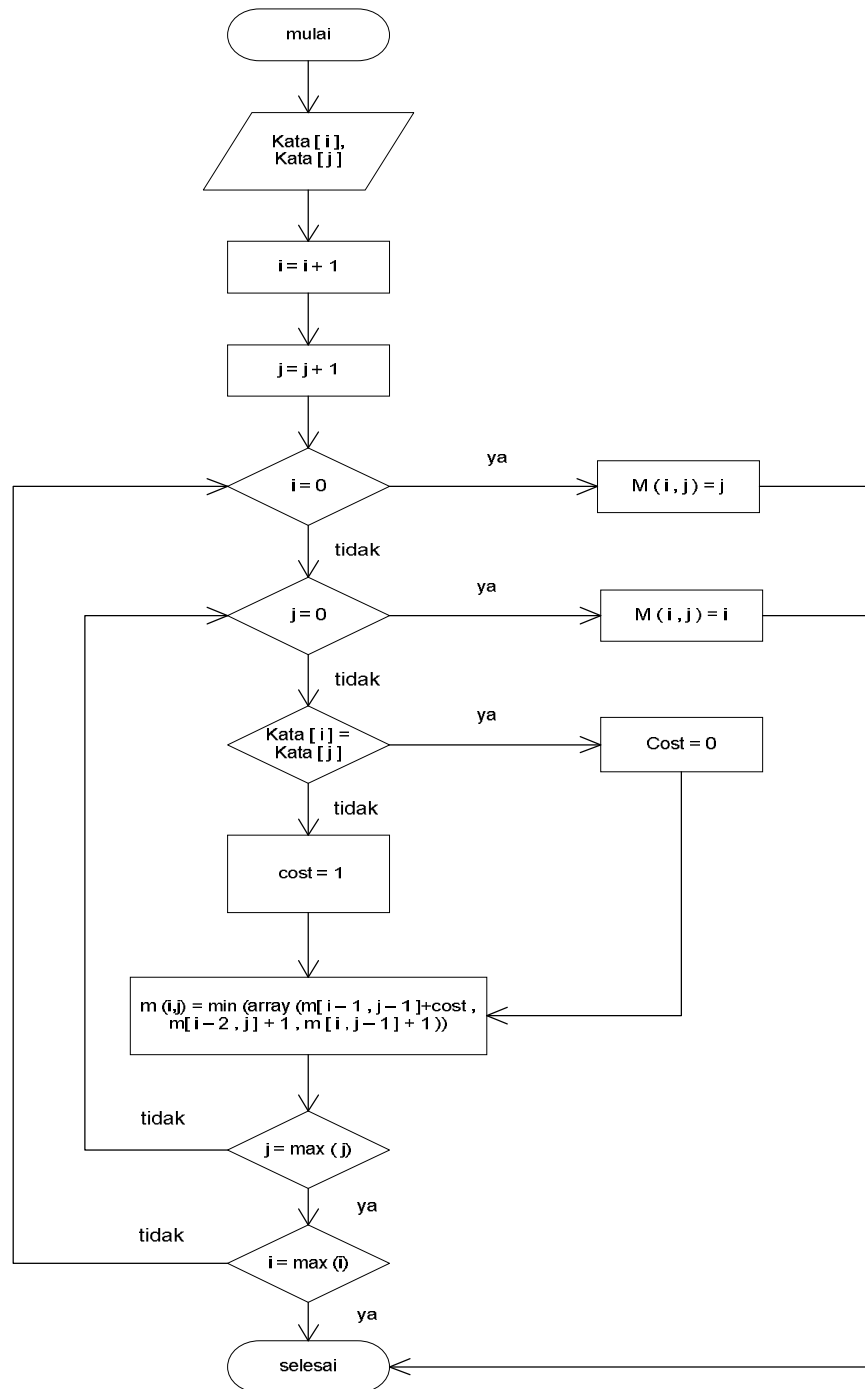
Kode [3] == kunci?

Ya => terdapat dalam *database* indeks ke [3] yaitu ablation.

Gambar 2 menunjukkan *flowchart* algoritma Levenshtein Distance. Untuk membandingkan antara *string* satu dengan *string* yang lainnya dimulai dengan menginisialisasikan matriks *i* dan *j* = 0, jika *string* yang diinput sama dengan *string* yang dibandingkan maka nilai *cost*-nya = 0, jika tidak sama maka nilai *cost*-nya = 1, kemudian mencari nilai minimum matriks

$[i, j]$ dengan beberapa operasi yakni *substitution*, *deletion* dan *insertion*. Jika sudah ditemukan nilai terkecil dari

beberapa operasi tersebut maka akan dimasukkan pada matriks $[i, j]$ [6].



Gambar 2 Flowchart algoritma Levenshtein Distance [6]

Pseudocode Algoritma Levenshtein Distance ditunjukkan oleh Gambar 3.

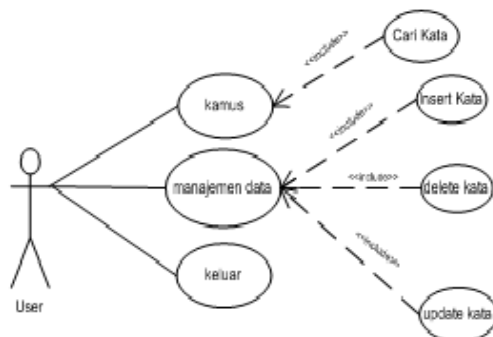
```

Procedure LevenshteinDistance;
Kamus
i, j, s1, s2, m, cost: integer
algoritma
for i ← 0 to s1.length do
for j ← 0 to s2.length do
    if i = 0 then
        m [i, j] ← j
    {perbandingan dengan kosong}
    else if j = 0 then
        m [i, j] ← i
    {perbandingan dengan kosong}
    else
        if s1 [i] = s2 [j] then
            cost ← 0
        else
            cost ← 1
        m [i, j] = minimum (
        m [i-1, j-1+cost], {substitusi}
        m [i-1, j+1], {penghapusan}
        m [i, j-1+1], {penambahan} )
    return m [s1.length, s2.length]
    
```

Gambar 3 *Pseudocode* algoritma Levenstein Distance [7]

2.5 Use Case Aplikasi

Use case diagram menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. *Use case diagram* dekat kaitannya dengan kejadian-kejadian. Kejadian (*Scenario*) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. Diagram *Use Case* berguna dalam beberapa hal seperti menjelaskan yang ada (*requirements*). *Use case* baru selalu menghasilkan fasilitas baru ketika sistem dianalisa, dan *design* menjadi lebih jelas. Gambar 4 menunjukkan *use case* diagram dalam pembuatan kamus kedokteran berbasis android.



Gambar 4 *Use case* diagram kamus kedokteran berbasis android

3. HASIL DAN PEMBAHASAN

Implementasi merupakan tahap pengembangan rancangan menjadi kode program. Pada awal bagian ini dijabarkan spesifikasi perangkat keras dan perangkat lunak serta implementasi yang dilakukan berdasarkan hasil pada tahap perancangan. Penjelasan pada subbab ini meliputi lingkungan pengembangan yang digunakan, batasan implementasi dan proses, serta hasil implementasi dari aplikasi.

Implementasi *interface* dari perangkat lunak dilakukan berdasarkan perancangan *user interface* yang telah dijelaskan, namun disesuaikan dengan komponen-komponen *view* yang tersedia pada *framework* GUI android. Implementasi antarmuka ditampilkan dalam bentuk *screen shoot* dari *handphone* android.

Gambar 5 menunjukkan *interface* menu dari aplikasi, dimana di dalamnya terdapat menu manajemen data, menu kamus dan menu keluar. Di dalam menu manajemen data terdapat beberapa inputan yaitu inputan untuk menghapus data, menambah data dan mengubah data. Di dalam menu kamus terdapat beberapa kata dan arti kata dari istilah-istilah kedokteran.



Gambar 5 *Inteface* awal aplikasi kamus kedokteran berbasis android

Gambar 6 menunjukkan *interface* menu dari manajemen data, di dalam menu manajemen data terdapat beberapa inputan yaitu inputan untuk menghapus data, menambah data dan mengubah data. Untuk menghapus data kita harus menginput kata

dan menerjemahkannya terlebih dahulu, kemudian tekan tombol hapus. Untuk menambah data tinggal memasukan kata dan terjemahan lalu tekan tambah.



Gambar 6 Interface menu manajemen data

Gambar 7 menunjukkan interface menu pencarian dari aplikasi yang menggunakan Metode *Interpolation Search* dan Algoritma *Levenshtein Distance*. User tinggal memasukan kata yang ingin dicari di kolom pencarian dan otomatis pencarian akan menampilkan kata yang berada di dalam *database*.



Gambar 7 Interface menu pencarian

Gambar 8 menunjukkan interface Screenshot kata yang mendekati bobot,

dimana *user* ingin mencari kata "abdoment" tetapi tidak terdapat dalam *database* dengan otomatis aplikasi akan memberikan alternatif pilihan kata untuk *user* yang paling mendekati sesuai dengan bobot kata yang dicari.



Gambar 8 Hasil pencarian kata

4. KESIMPULAN

Dari hasil pengembangan aplikasi kamus kedokteran menggunakan *Smartphone* berbasis android dapat diambil beberapa kesimpulan, yaitu :

1. Aplikasi kamus kedokteran ini telah diterapkan pada *Smartphone* berbasis android .
2. Aplikasi ini dilengkapi dengan metode pencarian yang menggunakan metode *Interpolation Search* dan algoritma *Levenshtein Distance* yang memudahkan *user* untuk mencari kata yang diinginkan dengan mudah dan cepat.
3. Aplikasi ini dapat menambah kata, menghapus kata, dan mengubah kata yang sudah ada.

5. SARAN

Beberapa saran untuk pengembangan lebih lanjut dari aplikasi penerjemah ini antara lain:

1. Aplikasi ini hanya melakukan proses penerjemahan dalam bentuk kata saja. Oleh karena itu, untuk pengembangan selanjutnya diharapkan aplikasi ini sudah bisa dikembangkan dengan

- menampilkan gambar dan suara dari beberapa istilah kedokteran tersebut.
2. Jumlah kosa kata yang disimpan dalam *database* berjumlah 1000 kata, tentunya ini masih harus ditingkatkan. Oleh karena itu, pada pengembangan selanjutnya diharapkan jumlah kata yang ada bisa ditambah sehingga akan memperkaya data dari aplikasi ini sehingga dapat melakukan proses pencarian kata yang diinginkan bisa bekerja secara maksimal.
 3. Diharapkan pengembangan selanjutnya bisa diatas versi Android yang telah ditentukan.

Pelita Informatika Budi Darma,
Volume : VII, Nomor : 2.

- [7] Ardiyanto, R.I., 2008, Dynamic Programming dalam Levenshtein *Distance* untuk Mengetahui Keterbedaan Dua String, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2007-2008/Makalah2008/MakalahIF2251-2008-035.pdf>, diakses 8 juni 2014.

DAFTAR PUSTAKA

- [1] Pius, A.P. dan Al Barry, M.D., 1994, *Kamus Ilmiah Populer*, Surabaya, Arkola.
- [2] Pratama, R. A., 2013, Implementasi Kamus 5 Bahasa untuk Smartphone Berbasis Android dengan menggunakan Algoritma Binary Search dan Metode Aproximate String Matching, *Skripsi*, Universitas Halu Oleo, Kendari.
- [3] Bali, C.C., 2014, Pengertian Kedokteran, <https://cybercornerbali.wordpress.com/tag/pengertian-kedokteran/>, diakses 8 juni 2014
- [4] Nurhikmah, J., 2014, Perancangan Pencarian Arsip Data Siswa Menggunakan Metode Interpolation Search (Studi Kasus : SMP Negeri 2 Medang Deras). *Medan : Pelita Informatika Budi Darma*, Volume : VII, Nomor : 1.
- [5] Bernardino, A., 2009, *Algoritma Levenshtein Dalam Pendekatan Approximate String Matching*, Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [6] Junaedy, R. A., 2014, Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks Dengan Menggunakan Metode Levenshtein *Distance*, *Medan :*

,