

## Rancang Bangun Pencarian Kata Pada Kamus Istilah Komputer dan Informatika Menggunakan Algoritme Brute Force Berbasis Android

Rismayani<sup>1\*</sup>, Novita Sambo Layuk<sup>2</sup>, Sri Wahyuni<sup>3</sup>, Hartana Wali<sup>4</sup>, Ni Kadek Marselina<sup>5</sup>

<sup>1</sup>Program Studi Rekayasa Perangkat Lunak, STMIK Dipanegara Makassar

<sup>2,3,4,5</sup>Program Studi Teknik Informatika, STMIK Dipanegara Makassar

Jl. Perintis Kemerdekaan Km.9 Makassar, Makassar, Indonesia 90000

\*email:rismayani@dipanegara.ac.id

(Naskah masuk: 6 September 2020; diterima untuk diterbitkan: 2 April 2021)

**ABSTRAK** – Kamus istilah komputer berarti kamus yang memuat istilah dengan makna konsepnya dari bidang ilmu tertentu. Selama ini kamus istilah komputer hanya berbentuk buku secara fisik kemudian berkembang menjadi sebuah kamus digital berupa website ataupun berupa aplikasi berbasis Android yang tersedia di play Store atau pun App store. Adapun masalah dalam penelitian ini adalah selama ini kamus istilah masih banyak yang berbentuk buku untuk menemukan istilah yang ingin dicari masih harus membuka lembaran buku dan melakukan pencarian dengan melihat urutan berdasarkan abjad. Penelitian ini bertujuan untuk membuat aplikasi kamus pencarian istilah komputer dan informatika berbasis android. Metode yang digunakan adalah metode algoritme brute force sebagai metode pencarian istilah pada aplikasi yang dibuat, algoritme brute force adalah algoritme yang sifatnya straight, lurus atau bisa juga disebut sebagai algoritme yang lempeng. Algoritme brute force merupakan bentuk algoritme yang sangat kompleks, karena untuk dapat menyelesaikan masalah dengan teknik straight forward atau lempeng ini, dibutuhkan banyak masukan dan juga pertimbangan secara logis, sehingga dapat diperoleh sebuah keputusan pemecahan masalah yang langsung mengacu atau menuju kepada hasil yang diinginkan. Adapun hasil dari penelitian adalah untuk memudahkan user atau pengguna untuk mencari kata istilah dengan cepat dan hasil dari algoritme brute force dapat menampilkan hasil dari pencarian kata istilah komputer dan informatika.

**Kata Kunci** – Kamus Istilah, Komputer, Informatika, Algoritme Brute Force, Android

## Word Searching a Computer and Informatic Dictionary of Term Application Using Brute Force Algorithm Android-Based

**ABSTRACT**– A computer term dictionary means a dictionary that contains terms with their conceptual meanings from a particular field of science. So far, the term computer dictionary has only been in the form of a physical book and has developed into a digital dictionary in the form of a website or Android-based application available on the Play Store or App store. The problem in this research is that so far there are still many dictionary terms in the form of books. To find the term you want to search, you still have to open a book sheet and search by looking at alphabetical order. This study aims to create an Android-based computer and informatics search dictionary application. The method used is the brute force algorithm method as a method of searching for terms in the application being made, The brute force algorithm is a straight, straight algorithm or it can also be called a plate algorithm. The brute force algorithm is a very complex form of algorithm, because to be able to solve problems with the straight forward or plate technique requires a lot of input and also logical considerations, so that direct problem solving decisions can be obtained. refers to or leads to the desired result. The results of this study are to make it easier for users or users to search for terms quickly and the results of the brute force algorithm can display computer and informatics word search results.

**Keywords** - Dictionary of Terms, Computers, Informatics, Brute Force Algorithm, Android

## 1. PENDAHULUAN

Kamus istilah komputer berarti kamus yang memuat istilah dengan makna konsepnya dari bidang ilmu tertentu. Kamus istilah komputer dan informatika yang di dalamnya berisi tentang istilah-istilah yang sering digunakan dalam bidang ilmu komputer dan informatika atau informasi dan teknologi. Selama ini kamus istilah komputer hanya berbentuk buku secara fisik kemudian berkembang menjadi sebuah kamus digital berupa *website* ataupun berupa aplikasi berbasis *Android* yang tersedia di *play Store* atau pun *App store*. Selama ini pula kamus istilah yang berbentuk buku untuk menemukan istilah yang ingin dicari masih harus membuka lembaran buku dan melakukan pencarian dengan melihat urutan berdasarkan abjad. Sedangkan kamus digital yang berupa *website* menampilkan kamus istilah dalam satu halaman blogsite sehingga proses pencarian istilah dilakukan dari halaman atas sampai halaman bawah, dan kamus dalam bentuk aplikasi *Android* sudah mempunyai proses pencarian istilah yang dapat menemukan istilah komputer yang diinginkan namun untuk melihat keterangan dari istilah harus mengklik istilahnya terlebih dahulu.

Tujuan dari penelitian adalah untuk membuat aplikasi pencarian kata pada kamus istilah komputer dan informatika berdasarkan kata yang akan dicari, menampilkan hasil pencarian, berupa istilah beserta dengan arti istilah dan mengimplementasikan algoritme *Brute Force* untuk pencarian kata kamus istilah pada *Android*.

Penggunaan algoritme *Brute Force* sebagai solusi karena algoritme tersebut dapat memecahkan masalah secara jelas dengan melalui banyak opini dan pilihan yang mampu untuk menyaring dari sekian banyak solusi yang ditawarkan sehingga pemecahan masalah yang dilakukan menjadi lebih baik dan lebih optimal.

Kamus adalah sebuah buku berisi kata-kata dari sebuah bahasa, biasanya disusun secara alfabetis, disertai keterangan akan artinya ucapannya, ejaannya, dan sebagainya [1][2]. Di dalam sebuah kamus yang ideal sepatutnya diberikan juga keterangan tentang pemenggalan kata, informasi tentang asal-usul kata, informasi tentang baku dan tidaknya sebuah kata, informasi tentang kata-kata arkhais dan juga klasik, informasi tentang area penggunaan kata, informasi tentang status sebuah kata, dan berbagai informasi lainnya.

Informatika merupakan salah satu cabang keilmuan yang cukup luas karena hampir mencakup dan terkait dengan berbagai hal tentang kehidupan ini, dimana kumpulan disiplin

informatika ini meliputi sains maupun teknik yang secara spesifik mengolah data menjadi informasi dengan memanfaatkan seoptimal mungkin teknologi informasi atau computer [3]. Komputer sebagai bahan utama dalam bidang keilmuan ini memiliki peranan yang sangat tinggi, sehingga informatika secara sederhana mengupas mulai dari bagaimana mesin tersebut bisa bekerja, bagaimana suatu data diolah dengan cara yang dimengerti oleh mesin sedangkan informasinya dimengerti juga oleh manusia, sampai bagaimana mesin tersebut mampu berkomunikasi dengan mesin lainnya.

*Android* merupakan sistem operasi berbasis *Linux* yang digunakan untuk perangkat mobile seperti *smartphone* [4]. Sejalan dengan hal tersebut, *Android* adalah sistem operasi untuk perangkat mobile berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi.

*Search engine* atau mesin pencari adalah sebuah sistem *software* atau sebuah aplikasi yang didesain dan disediakan oleh suatu badan komersial tertentu dan dibuat dengan tujuan untuk memudahkan pengguna internet mencari berbagai informasi di internet [5].

Algoritme *Brute Force* merupakan algoritme pencocokan string yang ditulis tanpa memikirkan peningkatan performa [6]. Algoritme ini sangat jarang dipakai dalam praktik, namun berguna dalam studi pembandingan dan studi-studi lainnya.

Kelebihan Algoritme *Brute Force* adalah dapat digunakan untuk memecahkan hampir sebagian besar masalah, sederhana dan mudah dimengerti, menghasilkan algoritme yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks dan menghasilkan algoritme baku (standar) untuk tugas-tugas komputasi seperti penjumlahan /perkalian N buah bilangan, menentukan elemen minimum atau maksimum ditabel [7].

Kelemahan Algoritme *Brute Force* adalah jarang menghasilkan algoritme yang mangkus/efektif, Lambat sehingga tidak dapat diterima dan Tidak sekreatif teknik pemecahan masalah lainnya[7].

Adapun beberapa penelitian terkait adalah pencarian kumpulan rumus matematika yang bermanfaat sebagai pembelajaran bagi para pelajar untuk mempermudah pengguna dalam proses mencari rumus yang akan dilakukan dengan memanfaatkan algoritme *Knuth-Morris-danPratt* (KMP). Algoritme ini merupakan pencocokan string dianggap paling efisien pada pencarian tersebut [8]. Penelitian yang bertujuan untuk melakukan pencarian data maintenance secara cepat yang terkirim melalui email di penyimpanan komputer melalui bantuan metode pencarian *query* [9]. String *matching* adalah proses pencarian semua

kemunculan *query* yang selanjutnya disebut *pattern* ke dalam string yang lebih panjang yang digunakan pada penelitian pencarian istilah-istilah akuntansi pada jurusan akuntansi [10]. Penelitian yang mengimplementasikan algoritme KMP dalam proses pencarian teks pada surat dan ayat al quran yang berbasis *web* yang dapat memudahkan untuk menemukan kata ataupun pernyataan dalam al quran [11]. Penelitian yang dapat menyelesaikan dengan cara yang sistematis terhadap pencarian string atau teks dengan menerapkan *Brute Force* pada *search engine*, *search file* dan *search data* [12]. Penelitian yang berupa uji coba pencocokan string dengan algoritme *brute force* dengan studi kasus menggunakan mesin pencarian (*search engine*) dengan bahasa pemrograman PHP untuk pencocokan string yang akan ditemukan pada pola kalimat yang dicari [13]. Penelitian yang membuat aplikasi kamus tiga bahasa yaitu bahasa Jawa Ngoko, Jawa Krama dan Indonesia berbasis *Android* dengan menggunakan metode *Sequential Search*, dimana metode pencarian ini dengan cara pengurutan dari indeks awal hingga data yang ditemukan [14]. Penelitian yang membuat aplikasi untuk penyajian informasi buku yang ada di perpustakaan AMIK Stiekom Sumatera Utara menggunakan algoritme *brute force* [15]. Penelitian yang membuat aplikasi kamus Lintang-Indonesia berbasis *android* yang menerapkan algoritme pencocokan *string Horspool* dalam pencariannya dan database *SQLite* sebagai penyimpanan data bahasa Lintang [16]. Penelitian yang melakukan kombinasi algoritme *Brute Force* yang digunakan untuk pengamanan data yaitu algoritme *steganografi LSB*, menggunakan kombinasi algoritme *steganografi LSB* dapat meningkatkan pengamanan file gambar agar lebih terjaga keaslian datanya dan lebih efisien [17]. Penelitian yang membuat aplikasi penerjemah untuk Bahasa Indonesia ke Bahasa Inggris menggunakan algoritme *string mathing brute force*. Algoritme *brute force* mencocokkan setiap karakter dari kiri ke kanan [18]. Pencocokan string Algoritme merupakan suatu algoritme yang digunakan untuk memecahkan masalah kesalahan suatu *teks* terhadap *teks* lain yang menerapkan *Brute Force* untuk aplikasi kamus Fisika [19]. Penelitian yang berupa aplikasi kamus bahasa pemrograman PHP dengan menggunakan algoritme *brute force* pada proses pencariannya yang dapat membantu pengguna untuk mencari istilah kata menjadi lebih efisien dibandingkan kamus berbentuk buku pada umumnya [20]. Penelitian yang membuat aplikasi kamus Semboyan PT. Kereta Api Indonesia (Persero) bekerja dengan cara mencari kata yang dibutuhkan oleh petugas kereta api yang

menggunakan *smartphone* berbasis *android* dngan menerapkan algoritme *Brute Force* yang dapat memecahkan masalah [21]. Dibutuhkan kamus yang dapat diakses melalui internet dan dapat digunakan melalui *handphone* atau komputer dengan menerapkan algoritme pencarian *sequential* untuk bahasa Indonesia, Jawa dan Banten [22]. Penelitian yang menunjukkan sebuah *jpeg/exif file fingerprint* tersusun atas tiga *hash value*. *SOI (Start of Image)* segmen *hash value* digunakan untuk mendeteksi terjadinya modifikasi berkas dalam bentuk perubahan tipe berkas dan penambahan objek pada konten gambar yang bertujuan untuk menyusun file fingerprint dari berkas *jpeg/exif* menggunakan *SHA256* dan algoritme *Brute Force string matching* untuk verifikasi integritas berkas *jpeg/exif* [23]. Sebuah aplikasi kamus istilah psikologi berbasis mobile yang mampu melakukan pencarian istilah psikologi dan menampilkan hasil berupa arti istilah psikologi dengan menerapkan algoritme *sequential search* dan algoritme *zhu-takaoka*. Algoritme *zhu-takaoka* merupakan salah satu algoritme *string matching* yang dapat melakukan pencarian kata dengan pergeseran yang banyak yaitu dengan array 2 (dua) dimensi [24]. Penelitian yang membuat aplikasi statistika dalam menggunakan bahasa pemrograman *visual basic net 2008*, dalam aplikasi statistika yang dirancang menggunakan algoritme *string matching* yaitu *brute force* untuk memperoleh hasil pencarian yang cepat dan tepat [25]. Algoritme *Brute Force* dan *Stemming* digabungkan dalam sistem pencarian kata *mashdar*, dalam menemukan kata dasar dari sebuah kata beserta perubahannya berdasarkan review dari kata dasar ke *mashdar* [26].

## 2. METODE DAN BAHAN

Adapun lokasi dari penelitian adalah STMIK Dipanegara Makassar di Jl. Perintis Kemerdekaan Km.9 Makassar.

Adapun alat yang digunakan dalam penelitian yaitu :

- A. Perangkat Keras :
  1. Laptop Acer VX15 Spesifikasi :
    - a. Processor Intel i7 Gen 7
    - b. GPU Nvidia Gefore GTX 1050
    - c. Ram 16GB DDR4
  2. Mouse Logitec K150
  3. Smartphone Android Xiaomi Mi 5X spesifikasi :
    - a. Qualkomm Snapdragon 625
    - b. Gpu Andreno 506
    - c. Ram 4 Gb Intern
- B. Perangkat Lunak :
  1. Windows 10 pro

2. Microsoft Word 2010
3. Android Studio 2017
4. Visual Studio 2016
5. StarUML
6. Mozilla Firefox
7. Os 7.1 Nougat

Adapun bahan penelitian sebagai berikut :

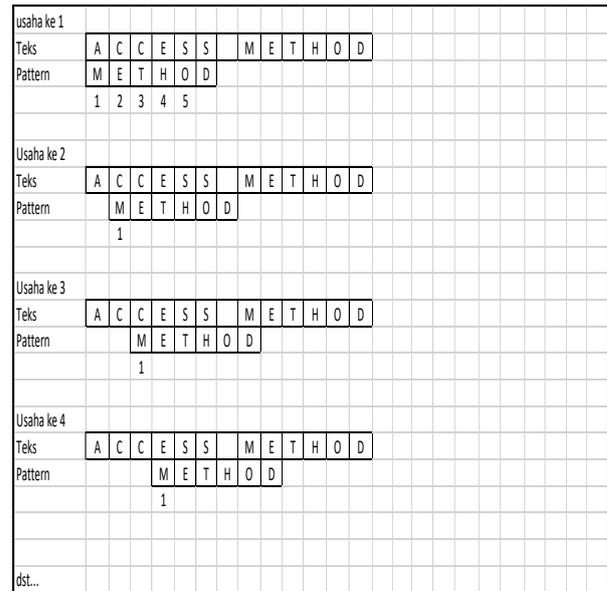
- a. Data - data kata istilah Komputer dan Informatika
- b. Kamus Manual Istilah Komputer dan Informatika.

Tahapan penelitian yang dilakukan sebagai berikut:

- a. Pengumpulan Data : mengumpulkan informasi dari lokasi penelitian.
- b. Analisis Sistem : penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehinggadapat diusulkan perbaikannya.
- c. Merancang Sistem : proses merancang sistem sebelum membuat coding dari sistem informasi yang dibuat.
- d. Pembuatan sistem informasi : merupakan strategi untuk memecahkan masalah dan mengembangkan solusi terbaik bagi permasalahan.
- e. Pengujian Sistem : mengetahui cara kerja dari aplikasi yang dirancang secara terperinci sesuai spesifikasi dan menilai apakah setiap fungsi atau prosedur yang dirancang sudah bebas dari kesalahan logika.
- f. Implementasi : yaitu penerapan pemodelan sistem yang telah dibuat untuk diaplikasikan ke pengguna yang akan melakukan pencarian istilah komputer dan informatika yang menggunakan metode algoritme brute force berbasis android.

Adapun metode pencarian kata yang digunakan adalah algoritme *Brute Force*, yang dimana cara kerja dari algoritme tersebut sebagai berikut:

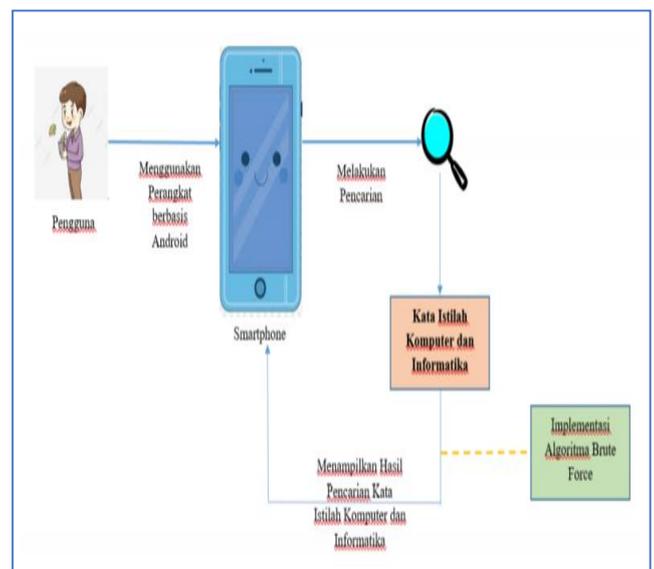
- a. *Algorithm BruteForceStringMatch* (  $T[0..n-1]$ ,  $P[0..m-1]$  )
- b. Implementasi *brute force string matching*
- c. Masukan : *array*  $T[0..n-1]$  , teks yang terdiri dari n karakter.
- d. *array*  $P[0..m-1]$  , *pattern/string* yang terdiri dari m karakter
- e. Keluaran : indeks dari karakter pertama pada *teks* ketika ada substring yang cocok, jika tidak ada yang cocok maka return = -1.
- f. for  $i \leftarrow 0$  to  $n - m$  do  $j \leftarrow 0$
- g. while  $j < m$  and  $P[j] = T[i + j]$  do  $j \leftarrow j + 1$
- h. if  $j = m$  return 1 else return = -1



Gambar 1. Cara Kerja Algoritme Brute Force [27]

### 3. HASIL DAN PEMBAHASAN

Adapun arsitektur sistem dari penggunaan algoritme brute force pada sistem yang dirancang dapat dilihat pada gambar 2.



Gambar 2. Arsitektur Sistem

Gambar 2 merupakan arsitektur sistem yang dimana pengguna menggunakan smartphone yang berbasis android melakukan pencarian kamus istilah komputer dan informatika yang prosesnya menggunakan algoritme *brute force* setelah itu akan menampilkan kembali hasil di *smarthphone* beserta dengan artinya.

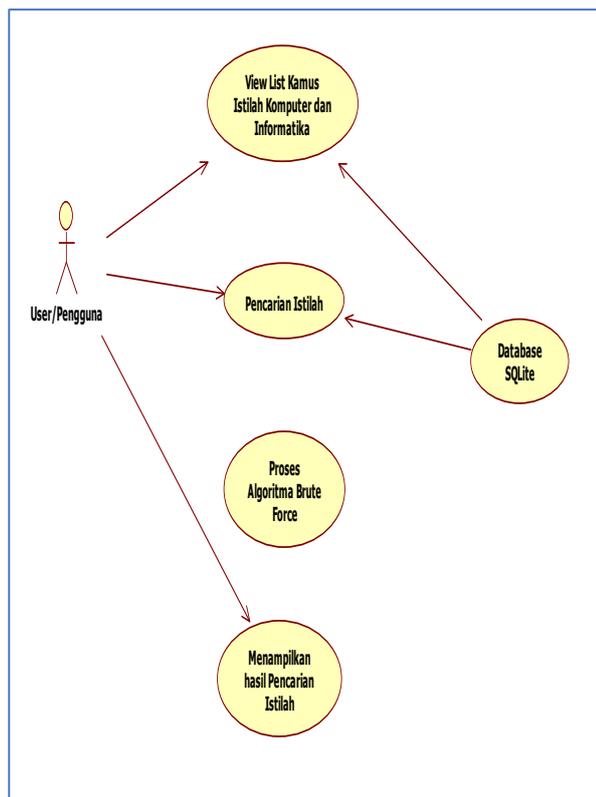
#### A. Analisis Sistem

Pada penelitian ini aplikasi yang dibuat akan digunakan untuk mencari istilah komputer dan informatika menggunakan metode pencarian

*algoritme bruteforce*. User dapat melihat seluruh daftar istilah pada aplikasi tanpa melakukan pencarian terlebih dahulu dan untuk menemukan kata atau istilah maka *user* harus memasukkan kata kunci yang ingin dicari. Kata kunci tersebut akan diproses menggunakan *algoritme brute force* dengan membandingkan kesamaan data dengan kata kunci istilah yang dicari.

### B. Use Gase Diagram

Adapun *use case* diagram terlihat pada gambar 3.



Gambar 3. Use Case Diagram

Gambar 3 merupakan *use case diagram* pada aplikasi terdiri dari satu aktor yaitu aktor *user/pengguna* dimana kegiatan yang dilakukan oleh actor ialah melihat *view list* kamus istilah komputer dan informatika kemudian memasukkannya pada pencarian istilah dan melihat hasil pencarian istilah. Dimana hasil pencarian tersebut melewati proses algoritme *brute force*. Dan semua kegiatan tersebut terhubung dengan database pada aplikasi.

### C. Tampilan Aplikasi

Adapun tampilan dari aplikasi yang dibuat dapat dilihat pada gambar 4, 5, 6, dan 7.

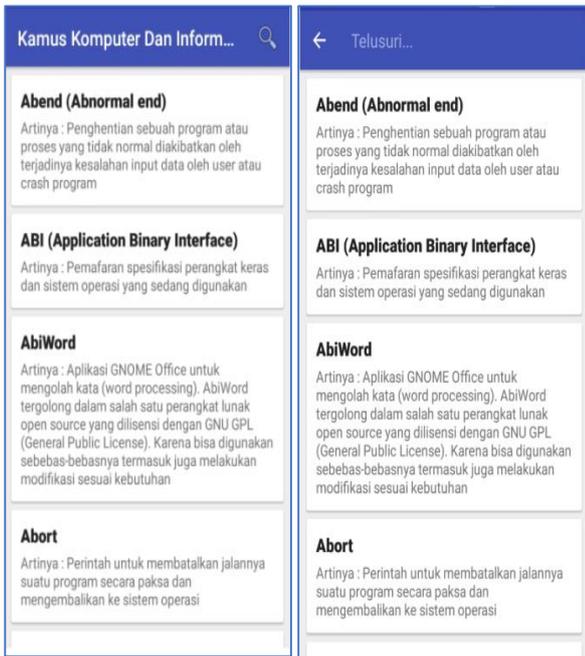


Gambar 4. Tampilan Icon Aplikasi Kamus



Gambar 5. Tampilan Utama Aplikasi Kamus

Gambar 5 memperlihatkan tampilan utama dari kamus istilah Komputer dan Informatika, bagian utama akan tampil setelah melakukan klik *icon* kamus tersebut pada perangkat *smartphone* seperti pada gambar 4.



Gambar 6. Tampilan Pencarian Kata



Gambar 7. Tampilan Hasil Pencarian Kata

Pada gambar 7 merupakan tampilan hasil dari pencarian kata dari istilah komputer dan informatika, yang dimana kata yang akan di cari diinputkan pada ikon *search* seperti yang ada pada gambar 6.

#### D. Implementasi Algoritme *Brute Force*

Adapun implementasi algoritme *Brute Force* pada kamus Istilah Komputer dan Informatika sebagai berikut :

1. Misalkan kata yang akan dicari adalah kata istilah *Cell* sebagai *pattern* dan *teks*-nya adalah *Binary Cell* maka langkah-langkahnya dapat dilihat pada gambar 8.

usaha ke

1

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

2

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

3

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

4

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

5

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

6

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

7

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Usaha ke

8

Teks

B	I	N	A	R	Y		C	E	L	L
---	---	---	---	---	---	--	---	---	---	---

Pattern

C	E	L	L
1	2	3	4

Gambar 8. Pencarian Kata *Cell*

2. Misalkan kata yang akan dicari adalah kata istilah *Oriented Language* sebagai *Pattern* dan *teks*-nya adalah *Application Oriented Language* maka langkah-langkahnya dapat dilihat pada

gambar 9.

Usaha ke 1	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 2	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 3	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 4	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 5	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 6	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 7	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 8	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 9	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 10	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 11	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E
Usaha ke 12	
Text	A P P L I C A T O N O R I E N T E D L A N G U A G E
Pattern	O R I E N T E D L A N G U A G E

Gambar 9. Pencarian Kata *Oriented Language*

Algoritme *brute force* mulai mencocokkan *pattern* pada awal *teks*, dari kiri ke kanan, algoritme ini akan mencocokkan karakter per karakter *pattern* dengan karakter di *teks* yang

bersesuaian, sampai salah satu kondisi berikut dipenuhi:

- Karakter di *pattern* dan di *teks* yang dibandingkan tidak cocok (*mismatch*).
- Semua karakter di *pattern* cocok kemudian algoritme akan memberitahukan penemuan di posisi ini.
- Algoritme kemudian terus menggeser *pattern* sebesar satu ke kanan, dan mengulangi langkah ke-2 sampai *pattern* berada di ujung *teks*

### Coding Brute Force

```
public static int
findSubString(String input, String
pattern) {
    input = input.toLowerCase();
    pattern =
pattern.toLowerCase();
    int output = 0;
    for (int i = 0; i <= input.length() -
pattern.length(); i++) {
        boolean ok = true;
        for (int k = 0; k < pattern.length();
k++) {
            if (input.charAt(i + k) !=
pattern.charAt(k)) {
                ok = false;
            }
        }
        if (ok) output++;
    }
    return output;
}
```

### E. Pengujian

Adapun pengujian terhadap fungsional aplikasi kamusi istilah Komputer dan Informatika menggunakan pengujian *Blackbox*, dengan hasil dapat dilihat pada tabel 1.

Tabel 1. Pengujian *Blacbox*

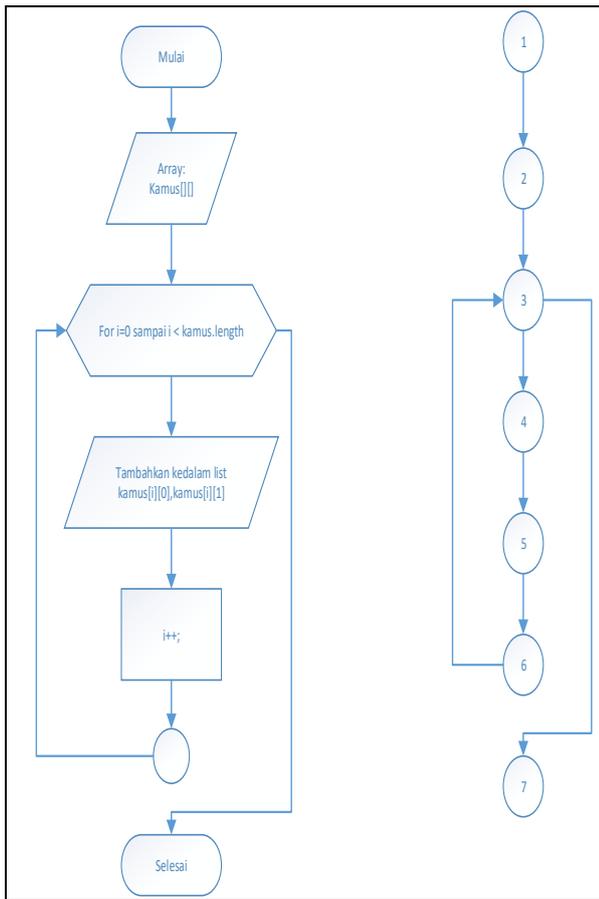
No	Modular	Hasil
1	Pengujian <i>View List</i> Kamus Istilah Komputer dan informatika	<i>Valid</i>
2	Pengujian Pencarian Istilah Komputer dan Informatika	<i>Valid</i>
3	Pengujian Hasil Pencarian	<i>Valid</i>

Selanjutnya pengujian yang dilakukan adalah pengujian terhadap logika algoritme *Brute Force* yang digunakan, pengujian yang digunakan adalah *Whitebox* sebagai berikut:

- View* Istilah

Untuk *Flowchart* dan *Flowgraph View* Istilah dapat dilihat

pada gambar 10.



Gambar 10. Flowchart dan Flowgraph View Istilah

Untuk menghitung *cyclomatic complexity* dari *edge* dan *node* dengan persamaan 1.

$$V(G) = E - N + 2 \quad (1)$$

Dimana :

E : jumlah *edge* pada *flowgraph*  
 N: jumlah *node* pada *flowgraph*

Menghitung *cyclomatic complexity* dari *p*, *p* adalah jumlah titik yang menyatakan logika dalam diagram alir dengan persamaan 2.

$$V(G) = p + 1 \quad (2)$$

Dimana  $p = 1$

Dari Gambar 10, berdasarkan persamaan 1, maka didapat :

$$V(G) = 7 - 7 + 2 = 2$$

Jadi jumlah *path* dari *flowgraph* di atas sebanyak 2 *path*.

Untuk *cyclomatic complexity* dari *p*, *p* adalah jumlah titik yang menyatakan logika dalam diagram alir berdasarkan persamaan 2, maka didapat :

$$V(G) = 1 + 1$$

$$V(G) = 2$$

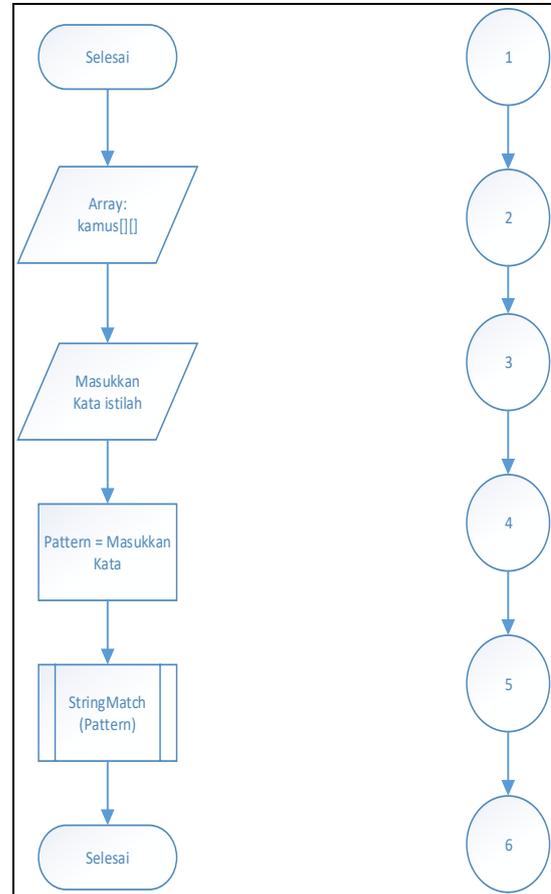
*Independent path* pada *flowgraph* adalah :

Path 1 = 1-2-3-4-5-6-3-4-5-6

Path 2 = 1-2-3-4-5-6-3-7

### b. Pencarian Istilah

Untuk *Flowchart* dan *Flowgraph View* Istilah dapat dilihat pada gambar 11.



Gambar 11. Flowchart dan Flowgraph Pencarian Istilah

Dari gambar 11, berdasarkan persamaan 1, maka didapat :

$$V(G) = 5 - 6 + 2$$

$$V(G) = 1$$

Jadi jumlah *path* dari *flowgraph* di atas sebanyak 1 *path*.

Untuk *Cyclomatic Complexity* dari *p*, *p* adalah jumlah titik yang menyatakan logika dalam diagram alir berdasarkan persamaan 2, maka didapat :

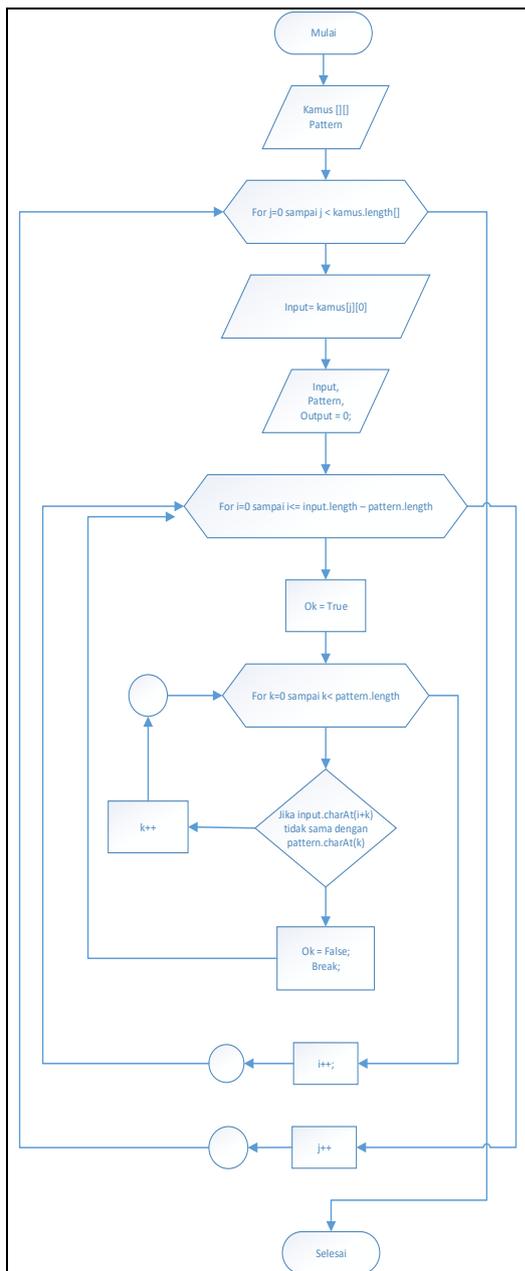
$$V(G) = 0 + 1$$

$$V(G) = 1$$

*Independent Path* pada *flowgraph* di atas adalah :

Path 1 = 1-2-3-4-5-6

### c. Hasil Pencarian Istilah



Gambar 12. Flowchart Hasil Pencarian Istilah

Berdasarkan persamaan 1, maka didapat CyclomaticComplexity dari Edge dan Node

$$V(G) = 20 - 17 + 2$$

$$V(G) = 5$$

Jadi jumlah path dari flowgraph di atas sebanyak 5 path.

Untuk *Cyclomatic Complexity* dari  $p$ ,  $p$  adalah jumlah titik yang menyatakan logika dalam diagram alir berdasarkan persamaan 2, maka didapat :

$$V(G) = 4 + 1$$

$$V(G) = 5$$

Independent Path pada flowgraph di atas adalah :

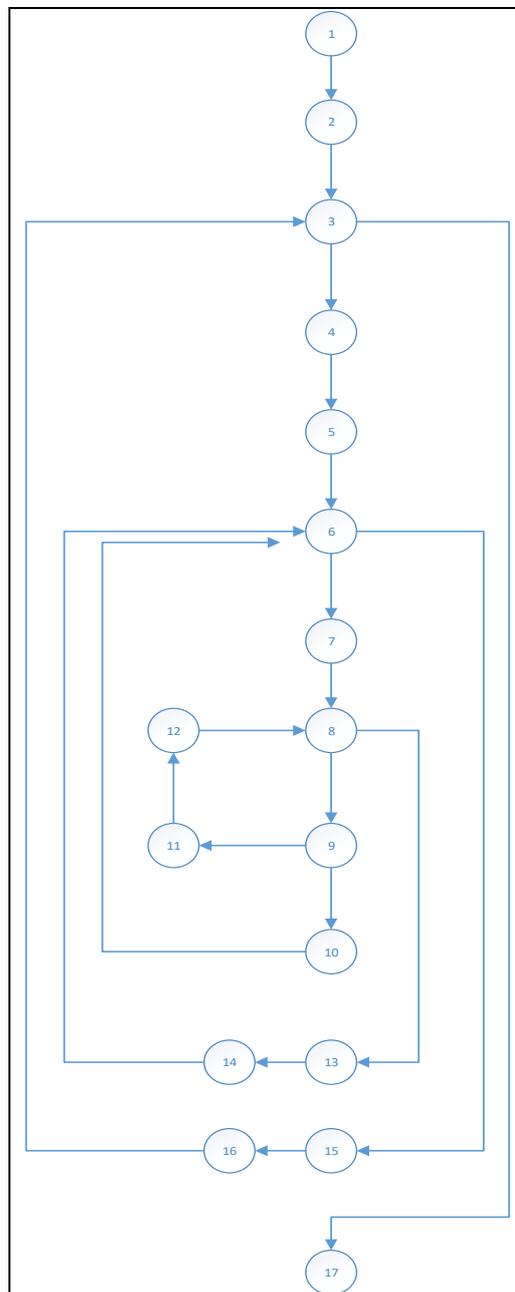
Path 1 = 1-2-3-17

Path 2 = 1-2-3-4-5-6-15-16-3-17

Path 3 = 1-2-3-4-5-6-7-8-9-10-6-15-16-3-17

Path 4 = 1-2-3-4-5-6-7-8-13-14-6-15-16-3-17

Path 5 = 1-2-3-4-5-6-7-8-9-11-12-8-13-14-6-15-16-3-17



Gambar 13. Flowgraph Hasil Pencarian Istilah

#### 4. KESIMPULAN

Kesimpulan dari penelitian adalah menghasilkan sebuah aplikasi pencarian kata pada kamus istilah komputer dan informatika yang dapat memudahkan user atau pengguna untuk mencari kata istilah dengan cepat dengan menerapkan algoritme brute force dapat menampilkan hasil dari pencarian kata istilah

komputer dan informatika dengan cepat. Berdasarkan pengujian *BlackBox* hasil *Test Factor* adalah bebas dari kesalahan dan dinyatakan valid dan berdasarkan pengujian *White Box* untuk penggunaan *algoritme brute force* pada kamus istilah komputer hasilnya adalah pengujian logika pada kamus istilah sudah sesuai dengan yang diharapkan dan bebas dari kesalahan logika.

Saran pengembangan sistem pada penelitian adalah aplikasi dapat di kembangkan menggunakan *Database server* agar data istilah dapat ditambahkan oleh administrator dan aplikasi dapat dikembangkan menggunakan *algoritme Boyer-Moore* ataupun *algoritme* lainnya yang relevan.

### UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih kepada RISTEK-BRIN yang telah memberikan dana, sehingga penelitian kami dapat terselesaikan dengan baik dan kami juga mengucapkan terima kasih kepada pihak STMIK Dipanegara Makassar yang telah banyak memberikan informasi yang dibutuhkan penulis.

### DAFTAR PUSTAKA

- [1] T. EMS, *Kamus Komputer Lengkap*. Elex Media Komputindo, 2015.
- [2] Eko Endarmoko, *Tesaurus bahasa Indonesia*. PT. Gramedia Pustaka Umum, 2007.
- [3] T. P. M. P. dan M. Sains, *Teknologi Informatika Dengan Pendekatan Kontekstual*. Zifatama Jawara, 2020.
- [4] T. A. Sandy and W. N. Hidayat, *Game Mobile Learning*. Ahlimedia Book, 2019.
- [5] P. Kent, *Search Engine Optimization For Dummies*. Wiley, 2010.
- [6] M. C. Sinaga, *Kriptografi Python*. Matius Celcius Sinaga, 2017.
- [7] Dr Pulung Nurtantio Andono, S. T., *Konsep Grafika Komputer*. Penerbit Andi, 2016.
- [8] A. S. Saragih, "Implementasi Algoritme Knuth-Morris-Pratt Pada Pencarian Kumpulan Rumus Matematika", *J. Ilm. INFOTEK*, vol. 1, no. 2, 2020.
- [9] A. Mirza, "Pencarian Data Tiket Maintenance Menggunakan Metode Brute Force", *J. Inform. Univ. Pamulang*, vol. 2, no. 3, pp. 122-126, 2017.
- [10] I. Mahdalena, "Implementasi Algoritme String Matching Pada Kamus Istilah Akuntansi Berbasis Android", *JURIKOM J. Ris. Komput.*, vol. 4, no. 4, Art. no. 4, Aug. 2017.
- [11] M. Syarif, "Implementasi Algoritme String Matching Dalam Pencarian Surah Dan Ayat Dalam Al-Quran Berbasis Web," vol. 6, no. 2, p. 7, 2017.
- [12] Dicky Apdilah, "Analisa Suku Kata Yang Sama Menggunakan Metode Brute Force", *Semantika (Seminar Nasional Teknik Informatika)*, 2017.
- [13] A. S. Sumi, P. Purnawansyah, and L. Syafie, "Analisa Penerapan Algoritme Brute Force Dalam Pencocokan String," *Pros. SAKTI Semin. Ilmu Komput. Dan Teknol. Inf.*, vol. 3, no. 2, Art. no. 2, Sep. 2018.
- [14] I. F. Shiddiq and A. Witanti, "Aplikasi Kamus Bahasa Jawa Ngoko, Jawa Krama, dan Indonesia Berbasis Android Dengan Menggunakan Metode Sequential Search," *JMAI J. Multimed. Artif. Intell.*, vol. 2, no. 2, Art. no. 2, Aug. 2018.
- [15] T. Zebua and N. Silalahi, "Aplikasi Saran Buku Bacaan Bagi Pengunjung Perpustakaan AMIK STIEKOM Sumatera Utara Berdasarkan Algoritme Brute Force," *Jurasik J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 3, no. 0, Art. no. 0, Jul. 2018.
- [16] V. Frigustini, A. Erlanshari, and D. Andreswari, "Implementasi Algoritme Horspool Pada Aplikasi Kamus Bahasa Lintang - Indonesia Berbasis Android," *Rekursif J. Inform.*, vol. 6, no. 1, Art. no. 1, Mar. 2018, Accessed: Sep. 06, 2020.
- [17] L. C. Purba, K. D. R. Sianipar, S. W. Siahaan, I. Gunawan, and S. Sumarno, "Pengamanan File Gambar Menggunakan Fungsi Algoritme Steganografi Lsb Dari Serangan Brute Force," *TECHSI - J. Tek. Inform.*, vol. 10, no. 1, Art. no. 1, May 2018.
- [18] F. T. Waruwu, and P. Ramadhani, "Perancangan Aplikasi Translator Bahasa Indonesia Ke Bahasa Inggris Esp (English For Specific Purpose) Komputer Sains Untuk Peningkatan Informasi Mahasiswa Menggunakan Brute Force Method," *KOMIK Konf. Nas. Teknol. Inf. Dan Komput.*, vol. 2, no. 1, Art. no. 1, Oct. 2018.
- [19] D. Abdurahman and I. Kurniawan, "Rancang Bangun Aplikasi Kamus Fisika Dasar Menggunakan Algoritme String Matching Brute Force Berbasis Android", *Semin. Nas. Teknol. Inf.*, vol. 1, pp. 150-155, Jul. 2018.
- [20] B. Erdani, F. D. Aditia, S. Rodiah, C. Ciptyasih, and I. H. Santi, "Sistem Aplikasi Kamus Istilah Bahasa Pemrograman PHP Menggunakan Algoritme Brute Force," *JMAI J. Multimed. Artif. Intell.*, vol. 3, no. 1, Art. no. 1, Feb. 2019.
- [21] R. Rismayani, and A. Ardimansyah, "Aplikasi Berbasis Mobile untuk Pencarian Rute Angkutan Umum Kota Makassar Menggunakan Algoritme Depth First Search,"

*BPPSDM KOMINFO*, vol. 18, no. 3, pp. 171–180, Desember 2015.

- [22] H. Sy, and R. Rismayani, "Pemetaan Titik Perumahan Pada Kota Makassar Menggunakan Google Maps Api," *Semnasteknomedia Online*", vol. 3, no. 1, Art. no. 1, Feb. 2015.
- [23] R. Fitriyanto, A. Yudhana, and S. Sunardi, "Manajemen jpeg/exif file fingerprint dengan algoritme Brute Force string matching dan Hash Function SHA256," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 5, no. 2, Art. no. 2, Jul. 2019.
- [24] N. Marbun, "Implementasi Algoritme Zhu-Takaoka Pada Aplikasi Kamus Istilah Psikologi Berbasis Mobile", *Inf. Dan Teknol. Ilm. INTI*, vol. 6, no. 3, Art. no. 3, Sep. 2019.
- [25] N. Novianti, N. Marbun, M. Zarlis, and D. Hartama, "Analisis Pencarian Arti Istilah Statistika Menerapkan Algoritme Brute Force," *Pros. SiManTap Semin. Nas. Mat. Dan Terap.*, vol. 1, pp. 881–886, Dec. 2019.
- [26] R. K. Dinata, N. Hasdyna, and R. Mahendra, "Kombinasi Algoritme Brute Force Dan Stemming Pada Sistem Pencarian Mashdar," vol. 5, no. 2, p. 7, 2020.
- [27] M. Curtin, *Brute Force: Cracking the Data Encryption Standard*. Springer Science & Business Media, 2007.