Optimasi Dashboard Information System STIKOM Bali dengan Algoritma Levenshtein Distance

Dasboard Information System Optimization of STIKOM Bali uses Levenshtein Distance Algorithm

Ni Kadek Sumiari*[1], Ni Ketut Dewi Ari Jayanti lis[2]
[1,2]Program Studi Sistem Informasi, STMIK STIKOM Bali
E-mail: ***[1]sumiari@stikom-bali.ac.id mail, [2]daj@stikom-bali.ac.id**

Abstrak

Kebutuhan data dan informasi bagi perguruan tinggi sangatlah penting salah satunya adalah bagi para eksekutif. Pada perguruan tinggi data dan informasi diperlukan untuk membuat dokument keperluan akreditasi atau sebagai pelaporan tahunan pada Rapat Tinjauan Manajemen (RTM). Untuk mempermudah eksekutif dalam memperoleh informasi data yang akurat maka keberadaan Dashboard Information System dalam perguruan tinggi cukup penting. STMIK STIKOM Bali merupakan salah satu perguruan tinggi yang sudah menerapkan Dashboard Infromation System bagi para eksekutif untuk memperoleh data dari semua system informasi yang sudah terintegrasi. Melalui Dashboard Information System Eksekutif STMIK STIKOM Bali dapat melihat informasi data yang realtime. Dashboard Infromation System STMIK STIKOM Bali dirancang menggunakan metode analisa FAST (Framework for the Application of Sistem Thinking) serta dibangun berbasis web dengan bahasa pemrograman PHP dan framework bootstrap. Pada Dashboard Infromation System pencarian data merupakan salah satu fungsi penting. Agar pencarian data lebih optimal maka perlu diterapkan sebuah algoritma untuk pencarian data. Pada Dashboard Infromation System STMIK STIKOM Bali diimplementasikan algoritma levenshtein distance untuk optimasi dalam pencarian data. Hasil setelah diimplemnatsikan algoritma levensthein distance pada Dashboard Infromation System adalah terdapat rekomendasi terdekat kata apabila user salah dalam melakukan input data sehingga pencarian lebih optimal.

***Kata Kunci** — Dashboard Information System, Optimasi, Levenshtein Distance*

Abstract

Data requirements and information on universities are very large, especially for management executives. In higher education the data and information needed to make documents are in accordance with the annual management report. To facilitate communication in using the right information data and Information System Dashboard Information in higher education is quite important. STMIK STIKOM Bali is one of the universities that has implemented an Information System Dashboard for executives to obtain data from all integrated information systems. Through the Executive Dashboard Information System STMIK STIKOM Bali can see realtime data information. Dashboard Information System STMIK STIKOM Bali will use the FAST analysis method (Thinking System Application Framework) and build web-based with PHP and bootstrap framework. On the Infromation System Dashboard look for data for one important function. In order to search data more optimally it is necessary to apply an algorithm to find data. In the STMIK STIKOM Bali Dashboard Information System levenshtein distance algorithm is implemented for optimization in data search. The results after implementing the Levensthein distance algorithm on the Information System Dashboard are one of the most complete of user errors in performing optimal data input.

***Keywords** — Dashboard Information System, Optimization, Levenshtein Distance*

Dikirim: 13 Desember 2018; Direvisi: 05 Januari 2019; Diterima: 8 Januari 2019

1. PENDAHULUAN

Kemajuan Teknologi Informasi sangat berpengaruh dan memberikan manfaat yang luar biasa bagi bidang pendidikan. Seperti yang telah kita ketahui sebelumnya bahwa teknologi informasi memiliki peranan yang penting dan sangat mempengaruhi sistem pendidikan yang berjalan. Memenuhi kebutuhan masyarakat akan informasi yang cepat, tepat, dan akurat merupakan tantangan tersendiri bagi suatu instansi pendidikan baik formal maupun non formal [1].

STMIK STIKOM Bali merupakan salah satu perguruan tinggi IT di Bali yang sudah menerapkan teknologi informasi dalam sebagian besar proses bisnisnya. STMIK STIKOM Bali memiliki basis data yang cukup besar karena hampir seluruh proses akademik yang ada memiliki sistem informasi yang terintegrasi dan menyimpan cukup banyak data dan informasi. Beberapa sistem informasi yang mendukung dalam proses akademik di STMIK STIKOM Bali adalah (1) sistem informasi akademik (SINAK) yaitu sistem informasi yang digunakan oleh bagian akademik (BAAK) dan program studi dalam mengelola proses perkuliahan, (2) sistem infromasi online (SION) yaitu sistem yang digunakan oleh mahasiswa untuk mendapat informasi tentang kegiatan akademik maupun non akdemik, (3) Sistem Infromasi Dosen (SID) yaitu sistem infromasi yang digunakan oleh dosen di STMIK untuk memperoleh infomasi dan mengelola data dosen. (4) Sistem Infromasi E-Research yaitu sistem informasi untuk pengelolaan data penelitian dan pengabdian pada masyarakat. (5) Sistem Informasi Keuangan, (6) Sistem informasi kepegawaian untuk mengelola data karyawan dan dosen. Serta (7) Sistem Penerimaan Mahasiswa Baru (PMB). Dengan semakin banyaknya jumlah mahasiswa di STMIK STIKOM Bali dan semakin berkembangnya proses bisnis di STMIK STIKOM Bali menyebabkan pengkategorian data semakin beragam sesuai dengan kasus yang diperlukan oleh unit di STMIK STIKOM Bali. Permintaan data yang paling sering adalah dari bagian Pembantu Ketua I (PK I), bagian Program Studi (Prodi) dan bagian akademik (BAAK). Data rekap biasanya digunakan ketika ada pelaporan ke DIKTI atau untuk akreditasi program studi maupun institusi dan juga digunakan sebagai rekap setiap semester serta pelaporan setiap akhir tahun yang disampaikan dalam rapat koordinasi (Rakor).

Terkait dengan kebutuhan data dan infromasi yang cukup besar STMIK STIKOM Bali telah membangun sebuah dashboard information system yang mengintegrasikan keseluruhan data dari semua system yang ada di STMIK STIKOM Bali [2]. Dasboard Information System di STMIK STIKOM Bali memberikan informasi bagi eksekutif manajemen dalam hal ini adalah Pembantu Ketua 1 (PK 1) tentang informasi mahasiswa, dosen dan proses akademik. Infromasi yang diperoleh seperti informasi mahasiswa baru, kelulusan mahasiswa, mahasiswa aktif, perkuliahan mahasiswa serta informasi yang berhubungan dengan proses akademik di STMIK STIKOM Bali. Pada dashboard Infromation System juga terdapat grafik yang dapat memperlihatakan perkembangan proses akademik. Data dan infromasi yang diperoleh digunakan oleh eksekutif manajemen dalam membuat dokument untuk akreditasi atau untuk dokumen pelaporan pada rapat tinjauan manajemen. Pada penelitian sebelumnya yang telah dilakukan Dashboard Infromation System di STMIK STIKOM Bali dirancangan menggunakan metode analisa FAST (Framework for the Application of Sistem Thinking) serta dibangun berbasis web dengan bahasa pemrograman PHP dan framework bootstrap serta database menggunakan SQL Server [2].

Digital Dashboard merupakan Dashboard yang menampilkan, pada satu layar, semua hasil pengukuran yang penting untuk mengarahkan perusahaan. Dashboard tersebut menampilkan indikator-indikator kinerja kunci sebagai grafik dan diagram dalam format browser Web, memberikan gambaran satu halaman dari semua pengukuran penting yang diperlukan untuk mengambil keputusan di tingkat eksekutif [3]. Dengan adanya Dashboard information system pemantauan kinerja perusahaan dapat lebih efektif. Dalam perguruan tinggi penerapan Dashboard information system juga bisa berperan penting dalam pemenatauan data dan informasi di perguruan tinggi tersebut. Suatu perguruan tinggi mempunyai banyak data dan informasi yang harus sering dipantau oleh eksekutif perguruan tinggi tersebut seperti data mahasiswa baru, data

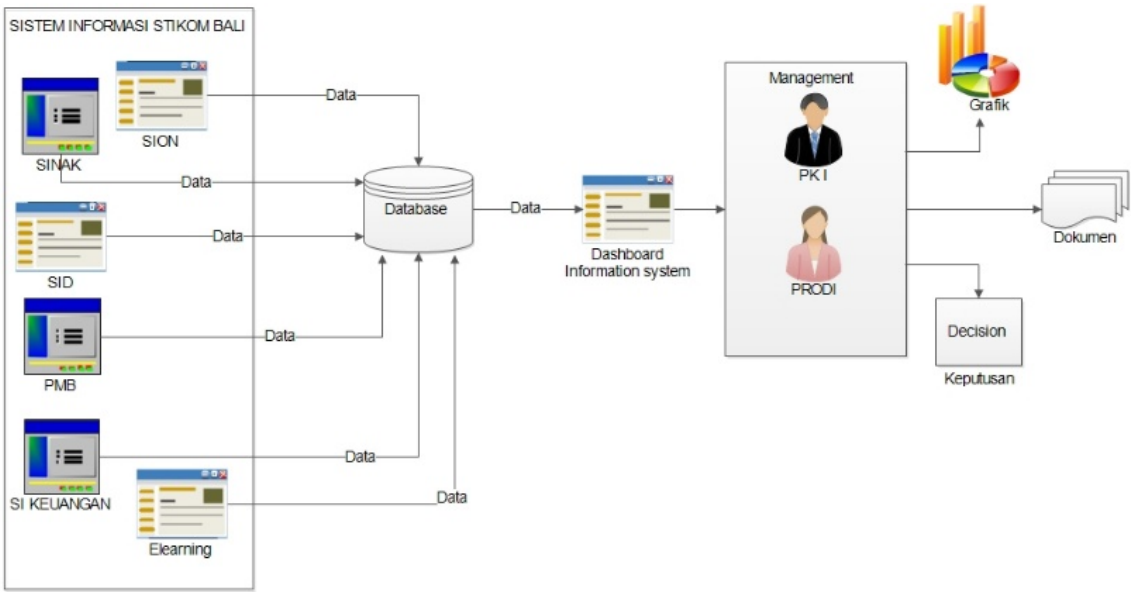
status mahasiswa, data dosen, kelas perkuliahan dan masih banyak lagi data akademik maupun yang non akademik. Dengan adanya Dashboard information system, top level managemen dalam perguruan tinggi tersebut dapat memantau proses data dan informasi yang ada secara real time [1].

Salah satu fitur yang sangat penting yang terdapat pada Dashboard Information System di STMIK STIKOM Bali adalah fitur pencarian data. Pada fungsi pencarian data dalam Dashboard Infromation System di STMIK STIKOM Bali belum menerapkan sebuah algoritma pencarian sehingga sering muncul permsalahan seperti kesalahan pengetikan ketika input kata menyebabkan data tidak muncul. Terkait dengan permasalahn tersebut maka perlu diimplemetasikan sebuah algoritma untuk pencarian data agar pencarian lebih optimal. Salah satu algoritma yang digunakan dalam pencarian data adalah algoritma levenshtein distance. Algoritma levenshtein distance merupakan algoritma yang cukup sederhana dan cukup mudah diterapkan pada sistem infromasi berbasis web karena sudah terdapat fungsi untuk memanggil algoritma tersebut dalam bahasa pemrograman berbasis web [4]. Algoritma ini merupakan perkembangan dari dynamic programming dimana algoritma ini melakukan perhitungan dengan menggunakan matriks pembanding dan memberi output jumlah perbedaan di antara dua string yang disebut dengan distance. Algoritma ini menentukan distance berdasarkan jumlah minimum perubahan yang terjadi ketika terjadi transformasi dari bentuk string awal ke bentuk string lain [4].

Penelitian ini berfokus pada penerapan algoritma levenshtein distance pada Dashboard Infromation System di STMIK STIKOM Bali serta melihat akurasi ketepatan kata menggunakan algoritma levenshtein distance.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini yaitu pengumpulan data dan perancangan data, sedangkan pada teknik pengumpulan data menggunakan studi pustaka dan observasi. Metode observasi adalah teknik atau metode pengumpulan data dengan melakukan pengamatan secara langsung di lokasi yang menjadi tempat penelitian [1]. Tempat dalam melakukan observasi penelitian ini adalah STMIIK STIKOM Bali.



Gambar 1. Konsep Dashboard Information System

Model Konsep dashboard information system di STIKOM Bali seperti yang ditunjukkan pada Gambar 1 dengan proses detailnya adalah:

1. Data masukan yang diterima oleh Dashboard information system adalah dari system infromasi sebelumnya yang sudah ada di STIKOM Bali yaitu Sistem Infromasi Akademik (SINAK), Sistem Informasi Online (SION), Elearning, Sistem Inofrmasi Dosen (SID), Sistem Keuangan dan Sistem Penerimaan Mahasiswa Baru (PMB).
2. Data dari system informasi yang sudah ada sebelumnya disimpan dalam satu database dan akan diolah melalui Dashboard information system.
3. Dashboard information system akan digunakan oleh management STIKOM Bali yaitu Pembantu Ketua I (PK I) dan Program Studi (Prodi) STIKOM Bali dimana melalui Dashboard information system management bisa membuat grafik, dokumen pelaporan dan pembuatan keputusan untuk permasalahan tertentu.

Dalam penelitian ini algoritma yang diterapkan adalah algoritma pencarian *Levenshtein Distance*. Algoritma *Levenshtein Distance* atau Edit Distance adalah algoritma pencarian jumlah perbedaan string yang ditemukan oleh Vladimir Levenshtein, seorang ilmuwan Rusia, pada tahun 1965. Algoritma ini banyak digunakan dalam berbagai bidang, misalnya pencarian string, pendeteksi plagiarisme dan *speech recognition*. Algoritma ini merupakan perkembangan dari dynamic programming dimana algoritma ini melakukan perhitungan dengan menggunakan matriks pembanding dan memberi output jumlah perbedaan di antara dua string yang disebut dengan distance. Algoritma ini menentukan *distance* berdasarkan jumlah minimum perubahan yang terjadi ketika terjadi transformasi dari bentuk string awal ke bentuk string lain. Dalam algoritma *Levenshtein Distance* terdapat 3 macam operasi yang digunakan, yaitu [4]:

1. *Insertion* adalah operasi melakukan penyisipan sebuah karakter ke dalam sebuah string tertentu.
2. *Deletion* adalah operasi melakukan penghapusan sebuah karakter di dalam sebuah string.
3. *Substitution* adalah operasi penggantian pada sebuah karakter pada posisi tertentu dengan karakter lain.

Algoritma *Levenshtein Distance* selain untuk menghitung *distance*, juga digunakan untuk menghitung jumlah operasi minimum yang diperlukan untuk mengubah string pertama dengan string kedua.

Berikut ini adalah algoritma *Levenshtein Distance* yang digunakan untuk mencari nilai distance antar kedua string. Diketahui bahwa nilai m adalah panjang dari string pertama dan n adalah panjang dari string kedua [5].

```
1.  Int computeDistance (String str1, String str2)
2.  {
3.  declare m=length of str1
4.  declare n=length of str2
5.  declare costs[n+1]
6.  FOR i from 0 to m
7.    DECLARE lastValue=i
8.    FOR j from 0 to n
9.          IF i=0 THEN
10.               Cost[j]=j
11.         ELSE
12.               IF j IS GREATER THAN 0 THEN
13.                  DECLARE newValue=costs[j-1]
14.                  IF str1.charAt(i-1) IS NOT EQUAL TO
15.                     str2.charAt(j-1) THEN
16.                     newValue=minimum (newValue, lastValue, costs[j])+1
17.               END IF
18.         END IF
19.   END FOR
20.   IF is GREATER THAN 0 THEN
21.         costs[n]=lastValue
22.   END IF
```

```
23. END FOR
24.
25. REURN costs[n]
26. }
```

Dari *pseudocode* yang dijabarkan algoritma *Levenshtein Distance* menyimpan data dalam bentuk array yang berukuran n + 1 yang diberi nama costs. Dimana variabel m adalah panjang karakter string pertama dan n adalah panjang karakter string kedua. Selanjutnya, iterasi dilakukan sebanyak m dengan terdapat iterasi sebanyak n di setiap iterasi m. Pada setiap iterasi m, nilai lastValue dideklarasikan sama dengan nilai i. Kemudian, pada iterasi n, terjadi pengecekan apakah i bernilai 0. Jika tidak, maka akan terjadi pengecekan apakah nilai j pada n lebih besar dari angka 0. Apabila bernilai TRUE, maka nilai variabel *newValue* akan diberikan nilai costs pada j kurang dari 1. Pada tahap berikutnya, terjadi pengecekan kesamaan huruf antar karakter. Jika kondisi terpenuhi, maka nilai variabel newValue akan diberikan nilai minimum dari newValue, lastValue, atau costs pada index j. Nilai minimum yang sudah didapat akan ditambahkan 1. Hasil akhir dari iterasi ini berupa sebuah array yang telah memiliki nilai pada setiap index-nya. Algoritma ini akan mengembalikan nilai pada array costs yang berada pada posisi index ke-n. Nilai ini adalah nilai distance antara string pertama dengan string kedua. [6]

Contoh kasusnya, yaitu diantara dua string, yaitu "*kitten*" dan "*sitting*". Kedua kata ini mempunyai nilai distance sebesar 3. Berikut ini adalah operasi-operasi yang dilakukan untuk mengubah kata "*kitten*" menjadi kata "*sitting*.

1. Kitten > Sitten, terjadi subsitusi atau penggantian karakter dari "K" menjadi "S".
2. Sitten > Sittin, terjadi substitusi pada karakter "e" menjadi "i".
3. Sittin > Sitting, terjadi penyisipan karakter "g" di belakang kata "sittin".

Algoritma *Levenshtein Distance* dapat dihitung dengan menggunakan matriks perbandingan. Tabel 1 adalah matriks perhitungan yang dihasilkan untuk mencari nilai distance dari kata "*kitten*" dan "*sitting*".

Tabel 1. Matriks Perhitungan Levenshtein Disntance

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

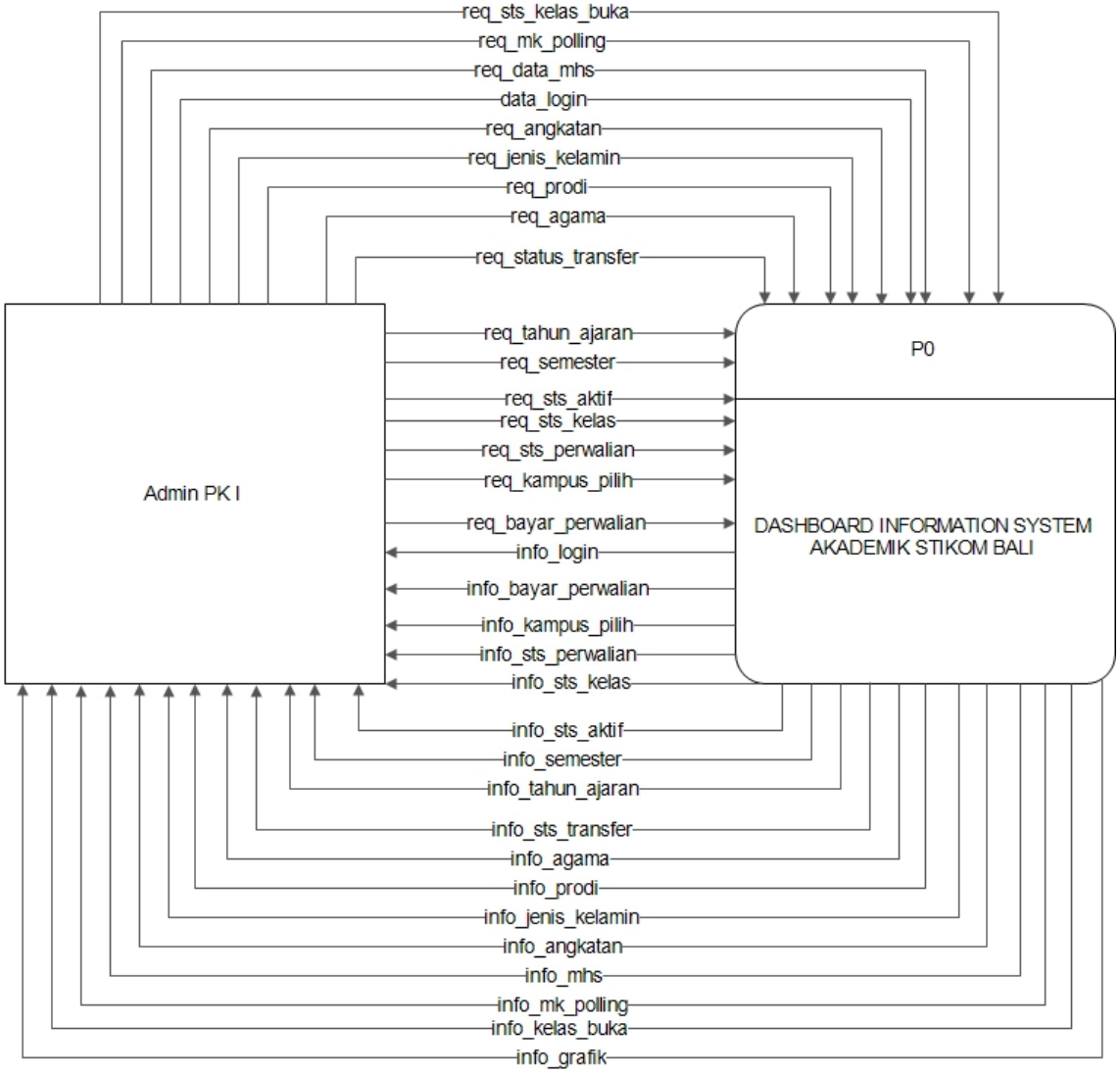
Tabel 1 menunjukkan bahwa kata "*kitten*" dan "*sitting*" mempunyai nilai distance sebesar 3. Dimana untuk setiap iterasi dicari operasi yang paling minimum yang bisa digunakan diantara operasi *insertion*, *deletion*, dan *substitution*. Pseudocode dari contoh operasi *insertion*, *deletion*, dan *substitution*, dapat didefinisikan seperti pada sintaks berikut [7].

```
1.  FUNCTION LevenshteinDistance (String str1, String str2)
2.  //initialize M as an empty Array [][]
3.  M [0] []=0
```

```
4.  FOR i = 1 to size of String S1
5.       M [i][0]=i
6.  FOR j = 1 to size of String S2
7.       M [j][0]=j
8.  FOR i = 1 to size of String S1
9.      FOR j = 1 to size of String S2
10.        IF s1[i]=S2 [j]
11.           cost=0
12.        ELSE
13.           Cost[0]
14.        M [i] [j] =min (M[i-1][j-1]+sost, M[i-1][j]+1, M[i][j-1]+1)
15. return M[i][j]
```

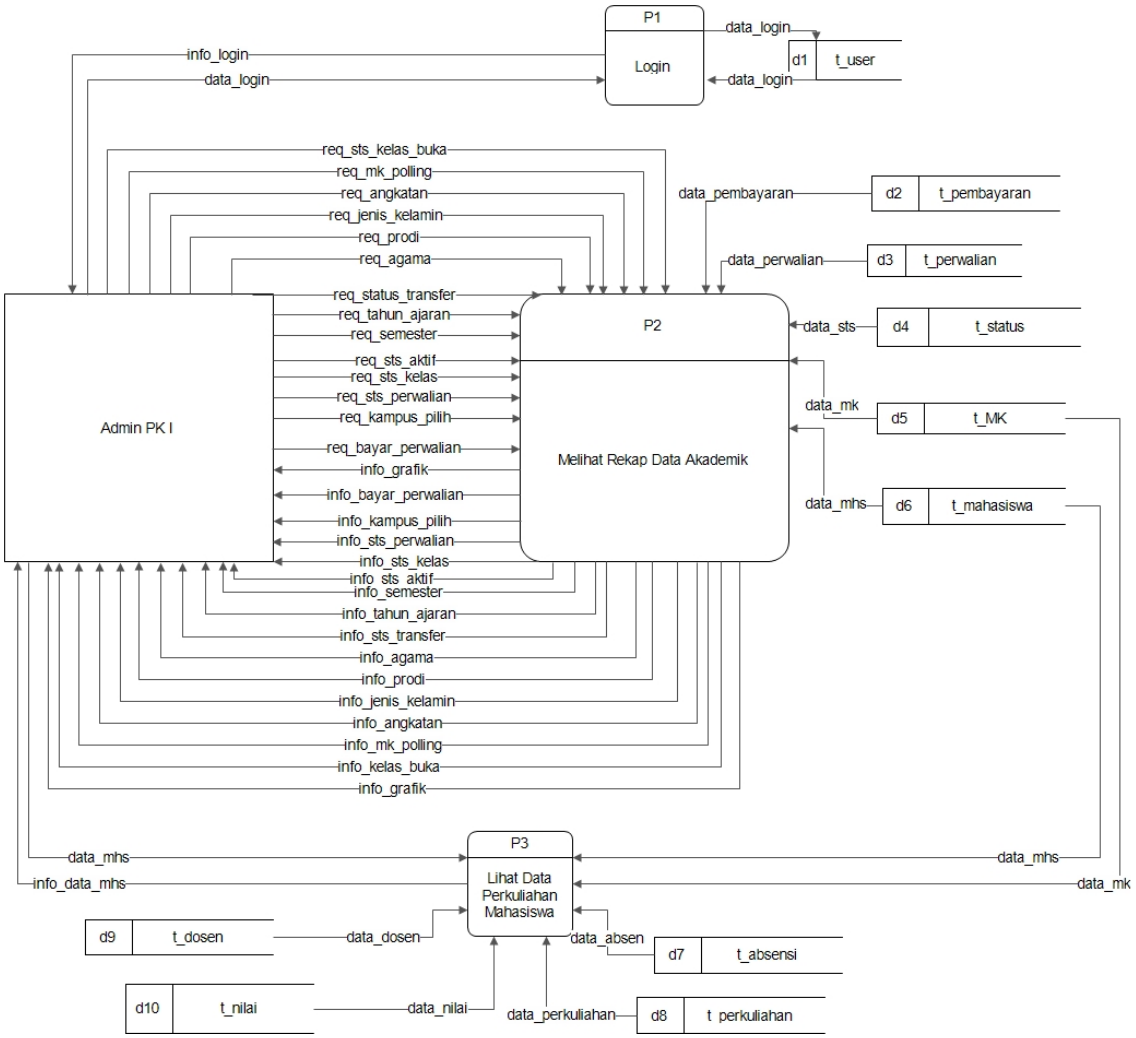
3. HASIL DAN PEMBAHASAN

Perancangan Dashboard Infromation System di STMIK STIKOM Bali menggunakan Teknik perancangan terstruktur yaitu menggunakan *Data Flow Diagram* dalam perancangan system Infromasinya serta *Entity Relationship Diagram* (ERD) dalam perancangan databaseny. Diagram konteks dan level 0 dari Dashboard Infromation System STMIK STIKOM Bali digambarkan pada Gambar 2 dan Gambar 3.



Gambar 2. Diagram Konteks

Pada diagram konteks *Dashboard* Infromation Sistem Akademik STIKOM Bali terdapat satu eksternal entity yaitu Admin PK I serta terdapat enam belas arus data input serta 17 arus data output.



Gambar 3. Diagram Level 0

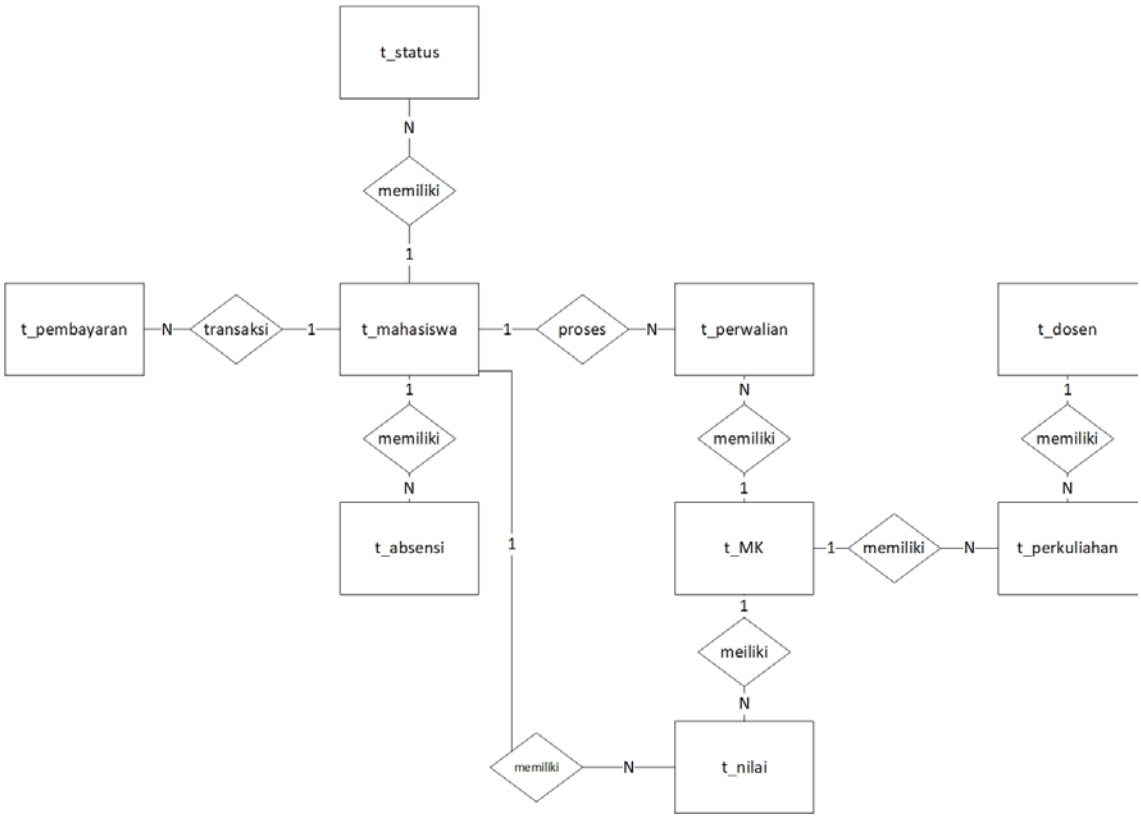
Pada Data Flow Diagram Level 0, menggambarkan arus data input dari eksternal entity ke proses dan arus data output dari proses ke eksternal entity Data Flow Diagram Level 0 ini, mendekomposisi proses pada level sebelumnya. Proses didekomposisi menjadi 3 proses yaitu:

a. Proses Login
   Pada proses ini entitas yang terlibat satu yaitu Admin PK I dan melihatkan satu data store yaitu t_user
b. Proses Melihat Rekap Data Akademik
   Pada proses ini melibatkan satu entitas yaitu Admin PK I dan lima data store yaitu t_pembayaran, t_MK, t_status, t_perwalian, t_mahasiswa
c. Proses Lihat Data Perkuliahan Mahasiswa
   Proses ini melibatkan satu entitas yaitu admin PK I dan empat data store yaitu t_dosen, t_nilai, t_perkuliahan dan t_absensi.

Sedangkan Entity Relationship Diagram (ERD) pada *Dashboard* Infromation System Akademik STIKOM Bali menggambarkan relasi data yang terjadi antar entitas. Terdapat sembilan entitas yang saling berelasi yaitu:

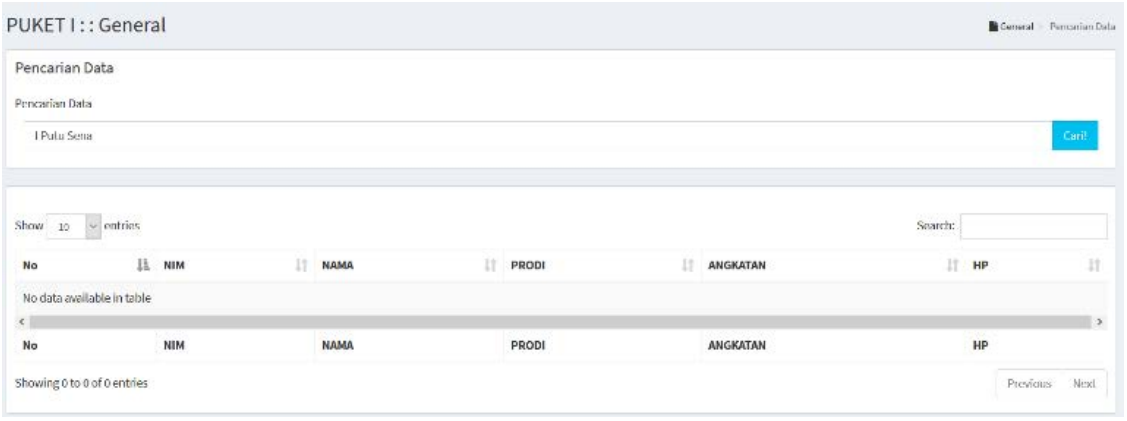
a. Entitas t_mahasiswa: menyimpan data mahasiswa
b. Entitas t_pembayaran: menyimpan data pembayaran
c. Entitas t_status: menyimpan status mahasiswa
d. Entitas t_absensi: menyimpan absensi mahasiswa
e. Entitas t_perwalian: menyimpan perwalian mahasiswa
f. Entitas t_dosen: menyimpan data dosen mahasiswa
g. Entitas t_MK: menyimpan data mata kuliah
h. Entitas t_perkuliana: menyimpan data perkuliahan dan KRS mahasiswa
i. Entitas t_nilai: menyimpan Nilai mahasiswa

Sedangkan untuk relasi entitas dijabarkan seperti pada Gambar 4.



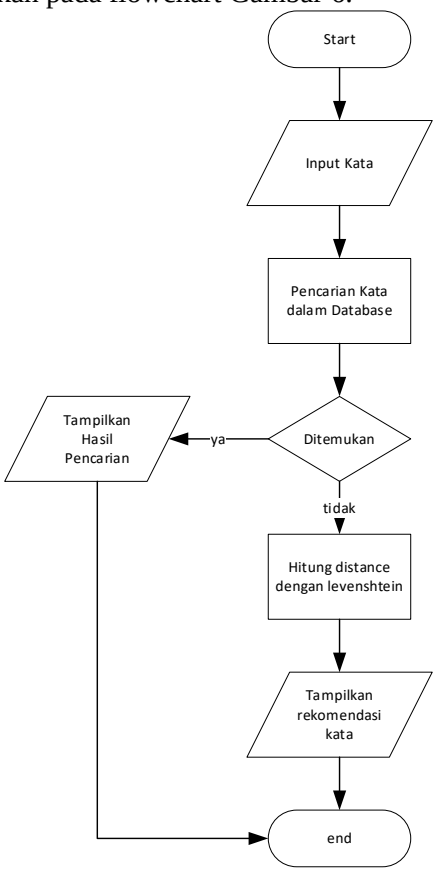
Gambar 4. Entity Relationsip Diagram

Pada Dashboard Information System di STMIK STIKOM Bali sebelumnya fungsi pencarian data belum menerapkan algoritma yaitu masih menggunakan fungsi query pencarian kovensional. Pencarian tanpa menggunakan algoritma pencarian apabila kata yang diinputkan dalam system tidak terdapat dalam database maka akan ditampilkan kosong dan tidak terdapat kata terdekat yang disarankan.



Gambar 5. Tampilan Fungsi Pencarian Dashboard Infromation System

Untuk optimasi pencarian pada dashboard information system ini maka diimplementasikan algoritma untuk pencarian yaitu menggunakan algoritma *Levensthein Distance*. Alur dari algoritma levensthein distance untuk optimasi pencarian pada dashboard information system digambarkan pada flowchart Gambar 6.



Gambar 6. Flowchart Levenshtein Distance

Proses Implmenetasi algoritma levensthein adalah sebagai berikut:

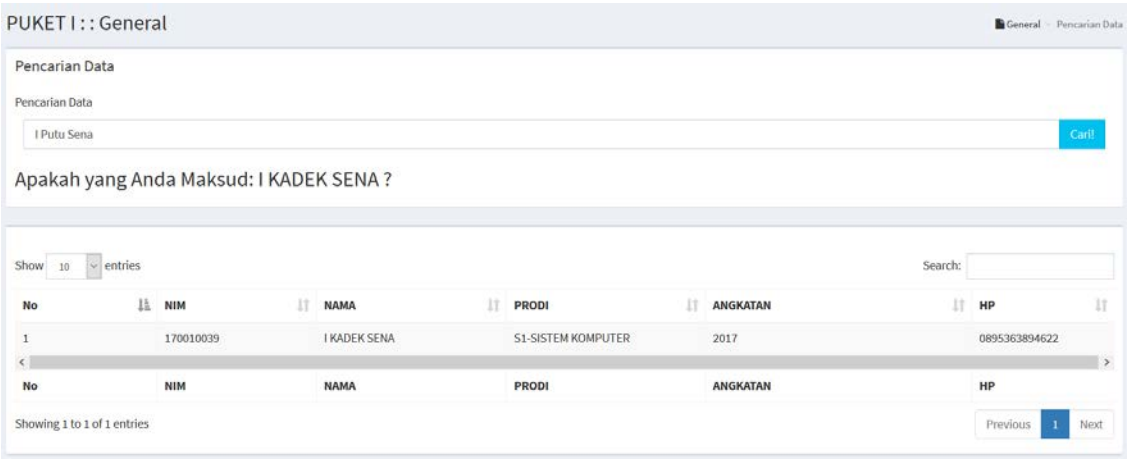
1. Kata diinputkan ke system apabila terdapat kata yang cocok 100% maka akan langsung ditampilkan ke user apabila tidak ada kata yang cocok maka dilanjut ke proses selanjutnya
2. Kata salah yang diinput dikonversi dalam bentuk array.
3. Tahap selanjutnya dilakukann proses seleksi terhadap semua kata dalam database dengan kata yang memiliki karakter dan Panjang yang sesuai.

4. Perhitungan jarak dengan algoritma *Levenshtein Distance* dengan mencarai kata dengan jarak terdekat.
5. Sistem akan menampilkan rekomendasi kata terdekat untuk ditampilkan ke user.

Implementasi levensthein distance dalam code PHP adalah sebagai berikut.

```
foreach ($dbword as $inputkata) {
    $lev = levenshtein ($input, $inputkata);
    if ($lev == 0) {
        $terdekat = $inputkata;
        $jarak = 0;
        break;
    }
    if ($lev <= $jarak || $jarak < 0) {
        $terdekat = $inputkata;
        $jarak = $lev;
    }
```

Setelah algoritma levhenstein akan mencari kata dengan jarak terdekat dari kata yang diinputkan. Setelah ditemukan kata terdekat maka akan menampilkan output saran dari kata yang dimaksud.



Gambar 7. Tampilan Optimasi Pencarian Menggunakan Levenshtein Distance

Pada Gambar 7 user menginpukan “I Putu Sena” yang tidak terdapat dalam daftar kata di dalam database. Dengan penerapan algoritma levenshtein maka system akan mencari kata terdekat dengan kata yang telah diinputkan sehingga keluar berupa output rekomendasi kata “I Kadek Sena”.

Dilakukan beberapa percobaan input kata ke system untuk mengukur optimasi pencarian kata dalam penerapan algoritma levenshtein distance menggunakan fungsi pada bahasa pemrograman PHP. Hasil pengukuran dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2. Tabel Optimasi Kata Levensthein Distance

Kata yang Ingin Dicari: DENPASAR		
Input ke	Huruf yang diinputkan	Hasil Sugesti Kata
1	DEN	ADENG
2	DENPA	DANGA
3	DENPS	ENDE
4	DENPAS	**DENPASAR**
5	DENPSR	**DENPASAR**
6	DNPASR	**DENPASAR**
7	DNPASA	**DENPASAR**
8	DNPSAR	**DENPASAR**

Pada kata Denpasar rekomendasi yang tepat baru ditampilkan oleh system ketika menginputkan minimal 6 huruf dari kata DENPASAR yang terdiri dari 8 huruf.

Tabel 3. Tabel Optimasi Kata Levensthein Distance

Kata yang Ingin Dicari: SURABAYA		
Input ke	Huruf yang diinputkan	Hasil Sugesti Kata
1	SUR	SUAI
2	SRBY	SUMBA
3	SURBY	SUMBA
4	SORBOY	SORONG
5	SURABY	**SURABAYA**
6	SRUBYA	**SURABAYA**
7	SRUBAY	**SURABAYA**
8	SURBAY	**SURABAYA**
9	SRABYI	**SURABAYA**
10	SUROBYO	**SURABAYA**
11	SRIBOYA	**SURABAYA**

Pada Tabel 3 percobaan kata Surabaya sugesti yang tepat muncul setelah minimal 6 huruf diinputkan dari kata SURABAYA yang terdiri dari 8 huruf. Kesalahan penulisan huruf akan berpengaruh apabila ada kata terdekat yang memiliki huruf vocal dan konsonan yang mirip.

Dari hasil pengujian kata diatas dapat diperoleh bahwa pada algoritma *Levenshtein Distance* kata yang ditampilkan sangat berpengaruh pada kedekatan jumlah karakater serta kemiripan dari karakter kata yang diinputkan dengan kata yang ada di database.

Pada Dashboard Information System STMIK STIKOM Bali implementasi *algoritma levensthein distance* digunakan untuk menampilkan saran kata yang ingin dicari apabila kata yang diinputkan salah. Setelah diimplementasikan dilakukan testing terhadap pencarian kata dengan menginputkan beberapa kata yang keliru. Pada tabel 4 dijabarkan hasil perbandingan sebelum dan setelah implementasi *algoritma levenshtein distance* pada sistem dengan melakukan uji coba input kata salah sebanyak 100 kata.

Tabel 4. Tabel Hasil Perbandingan Levensthein Distance

No	Kata yang ingin dicari	Kata yang diinput	Hasil Pencarian Sebelum Implementasi Algoritma	Hasil Rekomendasi Setelah Implementasi
1	Denpasar	Den	Tidak Tampil Kata	Adeng
2	Denpasar	Denp	Tidak Tampil Kata	Adeng
3	Denpasar	Denpa	Tidak Tampil Kata	Danga
4	Denpasar	Denps	Tidak Tampil Kata	Ende
5	Denpasar	Denpas	Tidak Tampil Kata	Denpasar
6	Denpasar	Denpsr	Tidak Tampil Kata	Denpasar
7	Denpasar	Dnpasr	Tidak Tampil Kata	Denpasar
8	Denpasar	Dnpasa	Tidak Tampil Kata	Denpasar
9	Denpasar	Dnpsar	Tidak Tampil Kata	Denpasar
10	Denpasar	Dps	Tidak Tampil Kata	Adeng
11	Denpasar	Denpasar	Denpasar	Denpasar
12	I Kadek Sena	Kadek Sen	Tidak Tampil Kata	I Kadek Sena
13	I Kadek Sena	I Putu Sena	Tidak Tampil Kata	I Kadek Sena
14	I Kadek Sena	I Kadek Sn	Tidak Tampil Kata	I Kadek Sena
15	I Kadek Sena	I Putu Sna	Tidak Tampil Kata	I Kadek Sena

No	Kata yang ingin dicari	Kata yang diinput	Hasil Pencarian Sebelum Implementasi Algoritma	Hasil Rekomendasi Setelah Implementasi
16	I Kadek Sena	Kdek Sena	Tidak Tampil Kata	I Kadek Sena
17	I Kadek Sena	I Wayan Sina	Tidak Tampil Kata	I Wayan Sinar
18	I Kadek Sena	Raden Sena	Tidak Tampil Kata	Raden Arya
19	I Kadek Sena	I Kadek Sena	Tidak Tampil Kata	Denpasar
20	Surabaya	Sur	Tidak Tampil Kata	Suai
21	Surabaya	Srby	Tidak Tampil Kata	Sumba
22	Surabaya	Surby	Tidak Tampil Kata	Sumba
23	Surabaya	Sorboy	Tidak Tampil Kata	Sorong
24	Surabaya	Suraby	Tidak Tampil Kata	Surabaya
25	Surabaya	Srubya	Tidak Tampil Kata	Surabaya
26	Surabaya	Srubay	Tidak Tampil Kata	Surabaya
27	Surabaya	Surbay	Tidak Tampil Kata	Surabaya
28	Surabaya	Surabay	Tidak Tampil Kata	Surabaya
29	Surabaya	Surabaya	Surabaya	Surabaya
30	Mochamad Soleh	Muhammad Soleh	Tidak Tampil Kata	Mochamad Soleh
31	Mochamad Soleh	Mochamad S	Tidak Tampil Kata	Mochamad Soleh
32	Mochamad Soleh	Soleh	Tidak Tampil Kata	Solihun Andi
33	Mochamad Soleh	Mad Soleh	Tidak Tampil Kata	Ahmad Soleh
34	Mochamad Soleh	Mohamad Soleh	Tidak Tampil Kata	Mochamad Soleh
35	Mochamad Soleh	Mohambad Solih	Tidak Tampil Kata	Mochamad Soleh
36	Mochamad Soleh	Mohamad Salih	Tidak Tampil Kata	Mochamad Soleh
37	Mochamad Soleh	moch. Soleh	Tidak Tampil Kata	Muh. Aldi
38	Mochamad Soleh	Mochamad Soleh	Mochamad Soleh	Mochamad Soleh
39	Buana Kubu	Buana	Tidak Tampil Kata	Boagan
40	Buana Kubu	Buaya Kubu	Tidak Tampil Kata	Buana Kubu
41	Buana Kubu	Kubu	Tidak Tampil Kata	Kuta
42	Buana Kubu	Buaya Kutuh	Tidak Tampil Kata	Buana Kubu
43	Buana Kubu	Boana Kuyu	Tidak Tampil Kata	Buana Kubu
44	Buana Kubu	Bona Kubu	Tidak Tampil Kata	Bona Kelod
45	Buana Kubu	Boana Kabu	Tidak Tampil Kata	Buana Kubu
46	Buana Kubu	Biana Kiba	Tidak Tampil Kata	Buana Kubu
47	Buana Kubu	Buana kubu	Buana Kubu	Buana Kubu
48	Waturenggong	Watu	Tidak Tampil Kata	Batur
49	Waturenggong	Watur	Tidak Tampil Kata	Batur
50	Waturenggong	Wature	Tidak Tampil Kata	Waturejo
51	Waturenggong	Watureng	Tidak Tampil Kata	Waturenggong
52	Waturenggong	Waturgong	Tidak Tampil Kata	Waturenggong
53	Waturenggong	Wturinggong	Tidak Tampil Kata	Waturenggong
54	Waturenggong	Watu Renggong	Tidak Tampil Kata	Waturenggong
55	Waturenggong	Waturanggeng	Tidak Tampil Kata	Waturenggong
56	Waturenggong	Baturenggong	Tidak Tampil Kata	Waturenggong
57	Waturenggong	Baturanggong	Tidak Tampil Kata	Waturenggong
58	Waturenggong	Waturenggong	Waturenggong	Waturenggong

No	Kata yang ingin dicari	Kata yang diinput	Hasil Pencarian Sebelum Implementasi Algoritma	Hasil Rekomendasi Setelah Implementasi
59	Tegal Permai	Tegal Permai	Tegal Permai	Tegal Permai
60	Tegal Permai	Tgl Permai	Tidak Tampil Kata	Tegal Permai
61	Tegal Permai	Tunggal Parmi	Tidak Tampil Kata	Tegal Permai
62	Tegal Permai	Tanggal Permi	Tidak Tampil Kata	Tegal Permai
63	Tegal Permai	Tegal Bari	Tidak Tampil Kata	Tegal Sari
64	Tegal Permai	Tegal Pari	Tidak Tampil Kata	Tegal Sari
65	Tegal Permai	Tgal Prmi	Tidak Tampil Kata	Tegal Sari
66	Tegal Permai	Tg Permai	Tidak Tampil Kata	Tegal Permai
67	Tegal Permai	Tingal Permai	Tidak Tampil Kata	Tegal Permai
68	Tegal Permai	Tanggul Permai	Tidak Tampil Kata	Tegal Permai
69	Tegal Permai	Tegal Parmi	Tidak Tampil Kata	Tegal Permai
70	Rudy Karwany	Rudy Karyawan	Tidak Tampil Kata	Rudy Karwany
71	Rudy Karwany	Rudi Karyawan	Tidak Tampil Kata	Rudy Karwany
72	Rudy Karwany	Rudy Aryawan	Tidak Tampil Kata	Rudy Karwany
73	Rudy Karwany	Rudykar	Tidak Tampil Kata	Rudiartha
74	Rudy Karwany	Rudykarwany	Tidak Tampil Kata	Rudy Karwany
75	Rudy Karwany	Rudy Kar	Tidak Tampil Kata	Rudiartha
76	Rudy Karwany	Rudy Karya	Tidak Tampil Kata	Rudiartha
77	Rudy Karwany	Rudy Karwa	Tidak Tampil Kata	Rudy Karwany
78	Rudy Karwany	Rudykarwan	Tidak Tampil Kata	Rudy Karwany
79	Rudy Karwany	Karyawan	Tidak Tampil Kata	Wayan Karyana
80	Rudy Karwany	Budi Karyawan	Tidak Tampil Kata	Rudy Karwany
81	Rudy Karwany	Rudy Karwany	Rudy Karwany	Rudy Karwany
82	Sukawati	Suk	Tidak Tampil Kata	Soka
83	Sukawati	Suka	Tidak Tampil Kata	Soka
84	Sukawati	Sukaw	Tidak Tampil Kata	Sukawati
85	Sukawati	Sukawa	Tidak Tampil Kata	Sukawati
86	Sukawati	Skwt	Tidak Tampil Kata	Sakti
87	Sukawati	Skawat	Tidak Tampil Kata	Sukawati
88	Sukawati	Sokasari	Tidak Tampil Kata	Sukawati
89	Sukawati	Sokasati	Tidak Tampil Kata	Sukawati
90	Sukawati	Kawati	Tidak Tampil Kata	Sukawati
91	Sukawati	Skwawati	Tidak Tampil Kata	Sukawati
92	Sukawati	Sukawati	Sukawati	Sukawati
93	I Wayan Winasta	Wayan	Tidak Tampil Kata	Wawan
94	I Wayan Winasta	Wayan Wi	Tidak Tampil Kata	Wayan Wira
95	I Wayan Winasta	Wayan Win	Tidak Tampil Kata	Wayan Wira
96	I Wayan Winasta	Wayan Wina	Tidak Tampil Kata	Wayan Wira
97	I Wayan Winasta	Wayan Winas	Tidak Tampil Kata	Wayan Winasta
98	I Wayan Winasta	Wyn Winast	Tidak Tampil Kata	Wayan Winasta
99	I Wayan Winasta	Wayin Waista	Tidak Tampil Kata	Wayan Winasta
100	I Wayan Winasta	Wayan Winasta	Wayan Winasta	Wayan Winasta

Dari hasil uji coba input kata pada Tabel 4 diperoleh hasil pada pencarian sebelum menggunakan levenshtein distance kata yang diinputkan apabila tidak terdapat pada database maka sistem tidak akan menampilkan data sedangkan pada pencarian dengan algoritma *levenshtein distance* akan ditampilkan rekomendasi kata terdekat dengan kata yang diinputkan. Untuk mendapatkan rekomendasi kata yang akurat pada algoritma *levenshtein distance* user harus menginputkan kata dengan jumlah karakter serta kemiripan huruf minimal 60% dari kata yang ingin dicari.

Sebelum *algoritma levenshtein* diterapkan dari hasil pengujian dari 100 kata diperoleh sebanyak 9 kata yang sesuai dengan kata yang diinginkan atau sebesar 9%. Setelah *algoritma levensthein* diimplementasikan dengan data yang sama diperoleh hasil pengujian sebanyak 66 rekomendasi kata yang sesuai dengan kata yang diinginkan atau sebanyak 66% kata yang tepat.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Telah diimplementasikan algoritma pencarian *Levenshtein Distance* pada dashboard Information System di STMIK STIKOM Bali
2. Implementasi algoritma *Levenshtein Distance* pada Dashboard Information System di STMIK STIKOM Bali berfungsi memberikan rekomendasi kata terdekat ketika user menginputkan kata yang salah atau tidak ada dalam database
3. Pada algoritma *Levenshtein Distance* kata yang ditampilkan sangat berpengaruh pada kedekatan jumlah karakater serta kemiripan dari karakter kata yang diinputkan dengan kata yang ada di database
4. Pada hasil uji coba pencarian kata tanpa menggunakan *algoritma* diperoleh sebanyak 9% kata yang tepat dari 100 kata yang diuji coba. Sedangkan setelah penerapan *algoritma levenshtein* diperoleh 66% kata yang tepat dari hasil uji coba data yang sama.
5. Pada uji coba pencarian kata apabila kata yang diinpukan sama dengan kata yang terdapat dalam database sistem maka algoritma *levenshtein distance* tidak akan menampilkan rekomendasi kata.
6. Akurasi algoritma *levensthein distance* dalam optimasi pencarian data cukup optimal akan tetapi masih terdapat kekuarangan seperti kesalahan terhadap memberikan saran kata apabila jumlah karakter kata yang diinputkan terlalu singkat atau kurang dari 60% jumlah karakter serta kemiripan huruf dari kata yang ingin dicari.

5. SARAN

Untuk pengembangan penelitian lebih lanjut diperlukan komparisasi menggunakan algoritma pencarian yang lain sehingga hasil yang diinginkan lebih akurat

DAFTAR PUSTAKA

[1] Untung, R., Handayani, I., Ningrum, A. A., 2017, Pemanfaatan Sistem iMe Berbasis WordPress sebagai Official Site RCEP pada Perguruan Tinggi, *Creative Information Technology Journal (CITEC)*, No. 3, Vol 4, Hal. 207-219.

[2] Sumiari, N. K., Jayanti, N. K. D. A., 2018, Implmentasi Metode Framework for The Application of System Thinking Pada Dashboard Information System STIKOM BALI, *Seminar Nasional Telekomunikasi dan Informatika (SELISIK) 2018*, Bandung, 25 Agustus.

[3] Handayani, I., Kusumahati, H., Badriah, A. N., 2017, Pemanfaatan Google Spreadsheet Sebagai Media Pembuatan Dashboard pada Official Site iFacility di Perguruan Tinggi. *Jurnal Sistem Informasi dan Teknik Informatika (SISFOTENIKA)*, No. 2, Vol. 7, Hal. 177-186.

[4] Lhoussain, A. S., Hicham, G., Abdellah, Y., 2015, Adaptating the Levenshtein Distance to Contextual Spelling Correction, *International Journal of Computer Science and Applications*, No. 1, Vol. 12, Hal. 127 – 133.

[5] Peggy., Hansun, S., 2015, Optimasi Pencarian Kata pada Aplikasi Penerjemah Bahasa Mandarin – Indonesia Berbasis Android dengan Algoritma Levenshtein Distance, *ULTIMA Computing*, No. 1, Vol. 7, Hal. 19-23.

[6] Pratama, B. P., Pamungkas, S. A., 2016, Analisis Kinerja Algoritma *Levenshtein Distance* Dalam Mendeteksi Kemiripan Dokumen Teks, *Jurnal Logika*, No. 2, Vol. 6, Hal. 131-143.

[7] Munif, A., Akbar, R. J., Tantra, R. I., Ilavi, R., 2017, Rancang Bangun Sistem E-Learning Pemrograman Pada Modul Deteksi Plagiarisme Kode Program Dan Student Feedback System, *JUTI: Jurnal Ilmiah Teknologi Informasi*, No. 1, Vol. 15, Hal. 104-118.