

USER MANUAL
PROGRAM EDIT DISTANCE



SURYA
UNIVERSITY

Oleh:

Dedy Van Hauten	1400910039
Farouq Fasial Anam	1400910045
Richard Nathaniel Chandra	1400910015

PROGRAM STUDI HUMAN-COMPUTER INTERACTION
FAKULTAS ILMU HAYATI
UNIVERSITAS SURYA
SERPONG
2015

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas kasih- Nya yang besar sehingga kami dapat menyelesaikan program *Edit Distance* ini dengan baik dan tepat waktu. Pertama kami mengucapkan terima kasih kepada Ibu Jelita Asian, Ph.D, selaku dosen mata kuliah pemrograman lanjut, atas bimbingan yang diberikan selama kami mengerjakan program ini. Tidak lupa juga kami mengucapkan terima kasih kepada teman-teman dan keluarga yang mendukung kami selama kami menyelesaikan program ini.

Program ini kami buat untuk memenuhi tugas pertama dari mata kuliah Algoritma dan Struktur Data. Banyak hal positif yang kami dapatkan selama mengerjakan program ini. Melalui tugas ini, kami telah belajar lebih dalam lagi mengenai algoritma *Edit Distance*.

Akhir kata, kami berharap program ini dapat berguna bagi banyak orang sebagai karya orisinil anak bangsa, demi membangun Indonesia jaya.

Tangerang, 2 Desember 2015

Penyusun

DAFTAR ISI

Cover Depan.....	1
Kata Pengantar	2
Daftar Isi.....	3
Bab I Pendahuluan	4
A. <i>Edit Distance</i>	4
B. Program Secara Umum	5
C. Keistimewaan Program	5
Bab II Petunjuk Praktis Pemakaian.....	6
A. Menu Utama.....	6
B. Program Berjalan	7
C. <i>Error Checking</i>	8
C.1. Input Angka Negatif	9
C.2. Input Karakter.....	9
C.3. Input Karakter dan Angka	9
C.4. Input Spesial Karakter	10
D. Request Program	10
D.1. Jika Input Program Salah	11
D.2. Jika Input Program Benar.....	11
Daftar Pustaka	12

BAB I

PENDAHULUAN

A. *Edit Distance*

Edit Distance adalah sebuah algoritma untuk mengukur jarak perbedaan antara dua tipe data string dengan cara menghitung angka operasi minimum yang dibutuhkan untuk mengubah string yang menjadi string lainnya yang dibandingkan. Edit Distance biasa lebih dikenal dengan nama Levenshtein Distance yang diambil dari nama penemunya, yaitu Vladimir Levenshtein, yang menemukan Edit Distance pada tahun 1965.

Kegiatan Edit Distance antara lain mengganti karakter, menambah karakter, atau menghapus karakter.

1. Mengganti karakter:

String 'commuter' dan 'computer', dengan mengubah huruf m pada string 1 menjadi huruf p.

2. Menambah karakter

String 'sort' dan 'sport', dengan menambahkan huruf p pada string 1.

3. Menghapus karakter

String 'sport' dan 'sort', dengan menghilangkan huruf p pada string 1.

B. Program Secara Umum

Program Edit Distance merupakan sebuah program yang bertujuan untuk mencari dan mengurutkan semua kata dalam file input yang memiliki edit distance tertentu dengan kata yang kita inginkan. Program akan meng-upload file input txt terlebih dahulu, kemudian string dalam kata tersebut diubah menjadi huruf kecil semua dan menghilangkan semua tanda baca, sehingga hanya meninggalkan angka 0-9, huruf a-z, dan tanda '. Selanjutnya, program akan menghitung frekuensi kata-kata dan disimpan di array. Setelah itu, program akan meminta user memasukkan kata yang diinginkan dan nilai edit distance. Program akan menampilkan dan mengurutkan kata pada file input yang memiliki edit distance tertentu dari kata yang dimasukkan oleh user.

C. Keistimewaan Program

Program ini tidak membutuhkan memory yang banyak. Selain itu, program ini juga memiliki code yang mudah dipahami sehingga mudah di-maintenance. Dengan demikian, programmer manapun dapat melakukan maintenance secara berkala pada program ini. Kompleksitas program ini pun hanya $O(n^2)$ jauh lebih cepat dibanding program sejenisnya. Program ini juga memiliki user interface yang rapi dan menarik sehingga nyaman dan mudah digunakan.

BAB II

PETUNJUK PRAKTIS PEMAKAIAN

A. Menu Utama

```
=====PROGRAM EDIT DISTANCE=====
                        Dibuat oleh:
Dedy Van Hauten        <1400910039>
Farouq Faisal Anam     <1400910045>
Richard Nathaniel C    <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya
```

Ketika program dijalankan, program akan terlebih dahulu mengupload file input dalam folder yang sama dengan program. Setelah itu, program akan menampilkan menu utama. Menu Utama merupakan tampilan awal ketika program dijalankan. Dalam Menu Utama ini, program meminta kita untuk memasukkan kata yang ingin diketahui distance-nya.

```
=====PROGRAM EDIT DISTANCE=====
                        Dibuat oleh:
Dedy Van Hauten        <1400910039>
Farouq Faisal Anam     <1400910045>
Richard Nathaniel C    <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya hound
Masukkan Edit Distance <Minimum 0>:
```

Setelah itu, program akan meminta memasukkan jarak Edit Distance yang diinginkan.

B. Program Berjalan

Program akan berjalan dengan baik ketika input kata yang dimasukkan berupa string dan input jarak edit distance yang dimasukkan berupa angka. Di bawah ini merupakan tampilan yang muncul jika program berjalan.

```
=====PROGRAM EDIT DISTANCE=====
                        Dibuat oleh:
Dedy Van Hauten      <1400910039>
Farouq Faisal Anam   <1400910045>
Richard Nathaniel C  <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya hound
Masukkan Edit Distance (Minimum 0): 2
hound 0 73
bound 1 5
found 1 47
hounds 1 4
pound 1 1
round 1 46
sound 1 25
wound 1 4
around 2 10
bond 2 1
bounds 2 1
could 2 201
fond 2 4
ground 2 17
hand 2 39
hold 2 11
hoped 2 6
hour 2 21
hours 2 13
house 2 65
hung 2 9
hunt 2 9
loud 2 2
mould 2 1
pounds 2 6
should 2 142
sounds 2 1
would 2 193
young 2 21
Apakah Anda ingin melakukan pencarian lagi? [Jika YA, masukkan Y]
```

Format bentuk output adalah kolom pertama menunjukkan kata yang memiliki jarak edit distance tertentu dengan kata yang dimasukkan ke input, kolom kedua adalah jarak edit distance kata tersebut dengan kata yang

dimasukkan ke input, dan kolom ketiga adalah jumlah kata tersebut dalam file input.

Setelah menyelesaikan pencarian dan menampilkan output, program akan bertanya apakah ingin melakukan pencarian lagi. Jika ya, masukkan Y/y lalu enter. Jika tidak, tekan tombol lain lalu enter. Jika kita memutuskan Ya, maka program akan kembali lagi ke awal.

```
would 2 193
young 2 21
Apakah Anda ingin melakukan pencarian lagi? [Jika YA, masukkan Y] y
```

```
=====PROGRAM EDIT DISTANCE=====
```

```
          Dibuat oleh:
Dedy Uan Hauten      <1400910039>
Farouq Faisal Anam   <1400910045>
Richard Nathaniel C  <1400910015>
```

```
PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya _
```

Jika kita memutuskan tidak, program akan keluar.

```
would 2 193
young 2 21
Apakah Anda ingin melakukan pencarian lagi? [Jika YA, masukkan Y] n
Terima kasih telah menggunakan program Edit Distance!
=====
```

C. Error Checking

Input jarak Edit Distance yang diperbolehkan untuk di *input* kedalam program hanyalah angka dan angka tersebut tidak boleh negatif.

C.1. Input Angka Negatif

```
=====PROGRAM EDIT DISTANCE=====
                        Dibuat oleh:
Dedy Uan Hauten      <1400910039>
Farouq Faisal Anam   <1400910045>
Richard Nathaniel C  <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya hound
Masukkan Edit Distance <Minimum 0>: -1
```

Jika input adalah angka negatif maka muncul *notice* :

```
edit distance harus berupa bilangan bulat! dan lebih besar dari -1
Masukkan Edit Distance <Minimum 0>: _
```

C.2 Input Karakter

```
=====PROGRAM EDIT DISTANCE=====
                        Dibuat oleh:
Dedy Uan Hauten      <1400910039>
Farouq Faisal Anam   <1400910045>
Richard Nathaniel C  <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya hound
Masukkan Edit Distance <Minimum 0>: a
```

Jika input adalah karakter maka muncul *notice* :

```
edit distance harus berupa bilangan bulat! dan lebih besar dari -1
Masukkan Edit Distance <Minimum 0>: _
```

C.3. Input Karakter dan Angka

```
=====PROGRAM EDIT DISTANCE=====
                        Dibuat oleh:
Dedy Uan Hauten      <1400910039>
Farouq Faisal Anam   <1400910045>
Richard Nathaniel C  <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya hound
Masukkan Edit Distance <Minimum 0>: a4_
```

Jika input adalah karakter dan angka maka muncul *notice* :

```
edit distance harus berupa bilangan bulat! dan lebih besar dari -1
Masukkan Edit Distance <Minimum 0>: _
```

C.4. Input Spesial Karakter

```
=====PROGRAM EDIT DISTANCE=====
          Dibuat oleh:
Dedy Uan Hauten          <1400910039>
Farouq Faisal Anam       <1400910045>
Richard Nathaniel C      <1400910015>

PROGRAM EDIT DISTANCE
Masukkan kata yang ingin diketahui distancinya hound
Masukkan Edit Distance <Minimum 0>: e_
```

Jika input adalah karakter maka muncul notice :

```
edit distance harus berupa bilangan bulat! dan lebih besar dari -1
Masukkan Edit Distance <Minimum 0>: _
```

D. Request Program

Ketika suatu input kata dan edit distance dimasukkan, program akan berjalan. Jika input edit distance salah, program secara otomatis akan meminta user untuk memasukkan input edit distance yang benar. Setelah edit distance dimasukkan dengan benar, program akan berjalan dan di akhir program akan diberikan pilihan apakah user ingin melakukan pencarian lagi atau tidak. Jika iya, maka user akan langsung diarahkan ke menu utama. Jika tidak, program akan berakhir.

D.1 Jika Input Program Salah

Jika input edit distance selain angka positif maka akan muncul notice :

```
edit distance harus berupa bilangan bulat! dan lebih besar dari -1  
Masukkan Edit Distance <Minimum 0>: _
```

D.2 Jika Input Program Benar

```
wounded 3 1  
wrong 3 3  
you 3 892  
your 3 186  
yours 3 7  
youth 3 1  
Apakah Anda ingin melakukan pencarian lagi? [Jika YA, masukkan Y]
```

DAFTAR PUSTAKA

- CPP Reference. “std::remove, std::remove_if”. From:
<http://en.cppreference.com/w/cpp/algorithm/remove> >>>>punctuation accessed
19 November 2015 at 23:45 WIB.
- Mulyanto, Arip. 2010. “Analisis *Edit Distance* Menggunakan Algoritma *Dynamic Programming*”. Universitas Negeri Gorontalo. Volume 5, No. 7.
- Sanfoundry. “C++ Program to Implement Levenshtein Distance Computing Algorithm”. From: <http://www.sanfoundry.com/cpp-program-implement-levenshtein-distance-computing-algorithm/> accessed 11 November 2015 at 21:16 WIB.
- Stack Exchange. “Quick Sort – Implementation”. From:
<http://codereview.stackexchange.com/questions/77782/quick-sort-implementation> accessed 11 November 2015 at 21:16 WIB.
- Stack Overflow. “Tolower function for C++ strings”. From:
<http://stackoverflow.com/questions/3403844/tolower-function-for-c-strings>
accessed 13 November 2015 at 20:12 WIB.