



**SISTEM PENDETEKSI PENYESUAIAN KESALAHAN
INPUT DATA SISWA MENGGUNAKAN METODE
*LEVENSHTEIN DISTANCE***

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir
Memperoleh Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH :

**NAMA : SUMIRATNO
NPM : 1514371005
PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020**

LEMBAR PENGESAHAN

**SISTEM PENDETEKSI PENYESUAIAN KESALAHAN INPUT
DATA SISWA MENGGUNAKAN METODE *LEVENSHTEIN*
*DISTANCE***

Disusun Oleh :

Nama : SUMIRATNO
NPM : 1514371005
Program Studi : Sistem Komputer

Skripsi telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 02 Mei 2020 :

Dosen Pembimbing I



Muhammad Iqbal, S.Kom., M.Kom

Dosen Pembimbing II



Ranti Eka Putri, S.Kom., M.Kom

Mengetahui,

Dekan Fakultas Sains dan Teknologi



Hamdani, ST., MT

Ketua Program Studi Sistem Komputer


Eko Hariyanto, S.Kom., M.Kom

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : SUMIRATNO
NPM : 1514371005
Program Studi : Sistem Komputer
Judul Skripsi : Sistem Pendeteksi Penyesuaian Kesalahan Input Data
Siswa Menggunakan Metode *Levenshtein Distance*

dengan ini menyatakan bahwa :

1. Tugas Akhir atau Skripsi saya bukan hasil plagiat.
2. Saya tidak akan menuntut perbaikan nilai Indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau.
3. Skripsi saya dapat di publikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut.

Demikian Pernyataan ini saya buat dengan sebenar-benarnya, terima kasih.

Medan, 2 Mei 2020

Yang membuat Pernyataan,



SUMIRATNO
1514371005

SURAT PERNYATAAN

Saya Yang Bertanda Tangan Dibawah Ini :

Nama : SUMIRATNO
L. P. M : 1514371005
Tempat/Tgl. Lahir : Pangkalanberandan / 28 Juli 1980
Alamat : Jl. Singapur Gg. Atmo Pelawi Darat Pangkalan Berandan
No. HP : 081289694572
Nama Orang Tua : SARTONO/SUMIRAH
Fakultas : SAINS & TEKNOLOGI
Program Studi : Sistem Komputer
Judul : Sistem Pendeteksi Penyesuaian Kesalahan Input Data Siswa menggunakan Metode Levenshtein Distance

Bersama dengan surat ini menyatakan dengan sebenar - benarnya bahwa data yang tertera diatas adalah sudah benar sesuai dengan ijazah pada pendidikan terakhir yang saya jalani. Maka dengan ini saya tidak akan melakukan tuntutan kepada UNPAB. Apabila ada kesalahan data pada ijazah saya.

Inikianlah surat pernyataan ini saya buat dengan sebenar - benarnya, tanpa ada paksaan dari pihak manapun dan kuat dalam keadaan sadar. Jika terjadi kesalahan, Maka saya bersedia bertanggung jawab atas kelalaian saya.

Medan, 21 Januari 2020
Sumiratno
Sumiratno
Sumiratno



SUMIRATNO
1514371005

Hal : Permohonan Meja Hijau

**Telah Diperiksa oleh LPMU
dengan Plagiarisme.....%²⁵**
Medan, 21 JANUARI 2020
AN Ka. LPMU
THARMIDI HARIM
Cahyo Pramono, SE, MM

Medan, 21 Januari 2020
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan

Di -
Tempat **Telah di terima
berkas persyaratan
dapat di proses**
Medan, 22 / 01 / 2020
Ka. BPAA
TEGUH WIRIYONO, SE, MM.

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : SUMIRATNO
Tempat/Tgl. Lahir : Pangkalanberandan / 28 Juli 1980
Nama Orang Tua : SARTONO
N. P. M : 1514371005
Fakultas : SAINS & TEKNOLOGI
Program Studi : Sistem Komputer
No. HP : 081289694572
Alamat : Jl. Singapur Gg. Atmo Pelawi Darat Pangkalan Berandan

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul Sistem Pendeteksi Penyesuaian Kesalahan Input Data Siswa menggunakan Metode Levenstein Distance, Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 exemplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 exemplar untuk penguji (bentuk dan warna penjilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya yang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	600.000
2. [170] Administrasi Wisuda	: Rp.	1.500.000
3. [202] Bebas Pustaka	: Rp.	100.000
4. [221] Bebas LAB	: Rp.	5.000
Total Biaya	: Rp.	2.205.000 1.605.000

5. UK 2019 Ganjil

Tax: Rp. 3.750.000 Periode Wisuda Ke : **64**

Ukuran Toga : **L**

28/1/20
Diketahui/Disetujui oleh
Hamdani, S., MT
Dekan Fakultas SAINS & TEKNOLOGI

Hormat saya
SUMIRATNO
1514371005

Catatan :

- 1. Surat permohonan ini sah dan berlaku bila ;
 - o a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - o b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asti) - Mhs.ybs.

Telah Diperiksa oleh UKM-C
Medan 21, 01, 2020
* Ka. UKM-C
Roro Rian Agustin, S SOS MSP

TANDA BEBAS PUSTAKA
No. 1492/BERP/BP/2020
Dinyatakan tidak ada sangkut
Pungutan UPT. Perpustakaan
21 JAN 2020
UNPAB
INDONESIA
UPT. PERPUSTAKAAN
SALNIA S.IP



UNIVERSITAS PEMBANGUNAN PANCA BUDI FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR*

Yang bertanda tangan di bawah ini :

Nama Lengkap : SUMIRATNO
 Tempat/Tgl. Lahir : PANGKALAN BRANDAN / 28 Juli 1980
 Nomor Pokok Mahasiswa : 1514371005
 Program Studi : Sistem Komputer
 Konsentrasi : Sistem Kendali Komputer
 Jumlah Kredit yang telah dicapai : 143 SKS, IPK 3.69
 Nomor Hp : 081289694572
 Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut :

Judul : Sistem Pendeteksi Penyesuaian Kesalahan Input DATA Siswa menggunakan Metode Levenstein Distance

21/1/20

Diisi Oleh Dosen Jika Ada Perubahan Judul

Yang Tidak Perlu

(Ir. Bhakti Alamshyah, M.T., Ph.D.)

30 Juli 2019
 Medan, ~~15 Januari 2020~~
 Pemohon,

 (Sumiratno)

Tanggal : *21/1/20*
 Disetujui oleh :
 Dekan

 (Hamdani, ST., MT)

Tanggal : 21 Januari 2020
 Disetujui oleh :
 Dosen Pembimbing I :

 (Muhammad Iqbal, S.Kom., M.Kom.)

Tanggal :
 Disetujui oleh :
 Ka. Prodi Sistem Komputer

 (Eko Hariyanto, S.Kom., M.Kom.)

Tanggal :
 Disetujui oleh :
 Dosen Pembimbing II :

 (Ranti Eka Putri, S.Kom., M.Kom.)

No. Dokumen: FM-UPBM-18-02 Revisi: 0 Tgl. Eff: 22 Oktober 2018

Plagiarism Detector v. 1460 - Originality Report

Analyzed document: 01/21/20 15:03:20

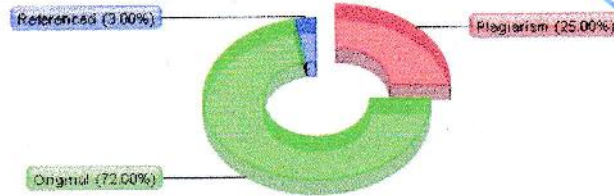
"Sumiratno_1514371005_Sistem Komputer.docx"

Check Type: Internet - via Google and Bing

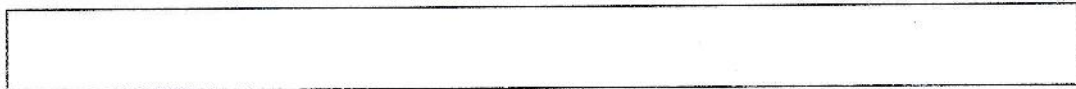
Licensed to: Universitas Pembangunan Panca Budi_License03



Relation chart:



Distribution graph:



Comparison Preset: Rewrite. Detected language: Indonesian

Top sources of plagiarism:

Percentage	Words	Source URL
% 13	wrds: 3923	https://id.123dok.com/document/7q012rvz-implemientasi-algoritma-levenshtein-dista...
% 9	wrds: 1022	http://repository.uinjkt.ac.id/dspace/bitstream/123456789/41996/1/FIRMAN%20NUGRA...
% 8	wrds: 971	https://repository.bst.ac.id/index.php/funduh/item/1725/9.-BAB-III-LANDASAN-TEORI...

[Show other Sources:]

Processed resources details:

103 - Ok / 8 - Failed

[Show other Sources:]

Important notes:

Wikipedia:	Google Books:	Ghostwriting services:	Anti-cheating:
Wiki Detected!	[not detected]	[not detected]	[not detected]

Active References (Urls Extracted from the Document):

No URLs detected

Excluded Urls:

No URLs detected

Included Urls:

No URLs detected

Detailed document analysis:

SISTEM PENDETEKSI PENYESUAIAN KESALAHAN INPUT DATA SISWA MENGGUNAKAN METODE LEVENSHTEIN DISTANCEDisusun dan Diajukan Untuk M emenuhi Persyaratan Ujian Akhir Memperoleh Gelar Sarjana Komputer

Plagiarism detected: 0.11% <http://repository.uinjkt.ac.id/dspa...> id: 1

Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan
SKRIPSI
OLEH :
NAMA



YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambang Telp. 061-8455571
Medan - 20122

KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama : SUMIRATNO
N.P.M. : 1514371005
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 09 Januari 2020
Ka. Laboratorium





UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
 Fakultas : SAINS & TEKNOLOGI
 Dosen Pembimbing I : MUHAMMAD IQBAL
 Dosen Pembimbing II : RANTI EKA.P.
 Nama Mahasiswa : SUMIRATNO
 Jurusan/Program Studi : Sistem Komputer
 Nomor Pokok Mahasiswa : 1514371005
 Bidang Pendidikan : SI.
 Tugas Akhir/Skripsi : SISTEM PENDETEKSI KESALAHAN INPUT DATA SISWA
 MENGGUNAKAN METODE LEVENSHTEIN DISTANCE.

ANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
22/09	Ace Gulat	[Signature]	
	Ace Guro	[Signature]	
30/09	Peng Bab I	[Signature]	
10/10	Ace Bab I	[Signature]	
11/10	Ace Bab II	[Signature]	
	Peng Bab III.	[Signature]	
12/10	Ace Bab III.	[Signature]	
12/10	Ace Bab IV & V	[Signature]	
12/10	Ace Semu	[Signature]	
10/11	Ace Gulat	[Signature]	
17/07	Ace Gulat	[Signature]	

13/8/2020
 Medan, 41 September 2019
 Diketahui/Disetujui oleh :
 Dekan.



Sri Shindi Indira, S.T., M.Sc.



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
 Fakultas : SAINS & TEKNOLOGI
 Pembimbing I : Dr. MUHAMMAD IQBAL, S.Kom, M.Kom :
 Pembimbing II : RANTI EKA PUTRI, S.Kom, M.Kom :
 Mahasiswa : SUMIRATNO
 Jurusan/Program Studi : Sistem Komputer
 Nomor Pokok Mahasiswa : 1514371005
 Tingkat Pendidikan : S1.
 Tugas Akhir/Skripsi : SISTEM PENDETEKSI KESALAHAN INPUT DATA SISWA
 MENGGUNAKAN METODE LEVENSHTEIN DISTANCE.

WAKTU	PEMBAHASAN MATERI	PARAF	KETERANGAN
September 19	Perbaiki bab I, tantumkan jurnal penelitian sebelumnya yg telah dipublish yang sejenis dg Penelitian		ACC Sempuro
September 19	ACC Bab I, Bab II perbaiki format tabel dan spasi. Saran sumber menggunakan Mendeley		
September 19	Perbaiki bab II, persiapkan draft bab III		
September 19	Perbaiki diagram gerantangan pd bab III, bawa aplikasi bimbingan selanjutnya		
September 19	ACC Bab II, bab III sesuaikan dg program		
September 19	ACC bab III, bab IV tambahkan pengujian black box / white box, persiapkan draft lengkap. Next pertemuan		
September 19	ACC bab IV, V.		ACC semhas.
Desember 2019	ACC sidang		
Agus 2020	ACC jilid		

12/8/2020
 Medan, 02 Desember 2019
 Diketahui/Disetujui oleh :
 Dekan



Sri Shindi Indira, S.T., M.Sc.

ABSTRAK

SUMIRATNO

SISTEM PENDETEKSI PENYESUAIAN KESALAHAN INPUT DATA SISWA MENGGUNAKAN METODE *LEVENSHTEIN DISTANCE* TAHUN 2020

Kesalahan input data sering terjadi terutama pada penulisan nama siswa dan siswi SMP Negeri 2 Babalan Pangkalan Berandan. Hal ini menyebabkan data tidak akurat dan tidak sesuai dengan data yang kita inginkan. Laporan tugas akhir ini merupakan skripsi yang berisi suatu pokok bahasan untuk mencari dan memeriksa kesamaan penulisan pada nama siswa dan siswi yang akan mengikuti Ujian Nasional. Pada penelitian ini, deteksi kemiripan penyesuaian nama dilakukan dengan menggunakan algoritma *Levenshtein Distance*. Pendeteksi dilakukan menggunakan satu data awal dan satu data target yang akan diuji tingkat kesamaan pada kesalahan pengetikan nama siswa tersebut. Proses pendeteksian diawali dengan melakukan tahap penginputan data nama siswa, setelah itu masuk ke tahap pencocokan string menggunakan *edit distance*, yaitu isian nilai pada matriks tersebut adalah jumlah operasi penghapusan, penyisipan dan penukaran yang dibutuhkan dalam mengubah string sumber ke string target. Hasil dari matrik tersebut adalah nilai dari *edit distance* dengan melihat jarak yang terkecil atau paling sedikit modifikasinya, maka dianggap sebagai kata yang cocok atau paling mendekati. Dan pengujian ini dilakukan dengan menguji kode-kode program yang dibuat pada aplikasi atau dengan kata lain pengujian *white box* dan melakukan pengujian apakah sistem yang dikembangkan sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem dengan kata lain pengujian dengan menggunakan *black box*.

Kata Kunci : Data, Pendeteksi, *Levenshtein Distance*, *White Box* dan *Black Box*.

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Diagram <i>Unified Modelling Language</i>	13
Gambar 3.1 <i>Use Case Diagram</i>	50
Gambar 3.2 <i>Activity Diagram</i>	51
Gambar 3.3 <i>Sequence Diagram</i>	52
Gambar 3.4 <i>Sequence Diagram</i> pada menu <i>Levenshtein Distance</i> <i>Input Data</i>	54
Gambar 3.5 <i>Sequence Diagram</i> pada menu <i>Levenshtein Distance</i> nama siswa	54
Gambar 3.6 <i>Sequence Diagram</i> pada menu Tentang	55
Gambar 3.7 <i>Flowchart</i> dari Sistem Pendeteksi Penyesuaian Data Menggunakan <i>Levenshtein Distance</i>	56
Gambar 3.8 Tampilan Menu Utama	57
Gambar 3.9 Tampilan Menu Info	58
Gambar 3.10 Tampilan Menu <i>Levenshtein Distance</i>	59
Gambar 3.11 Tampilan Menu Tentang	59
Gambar 3.12 Tampilan Menu Keluar	60
Gambar 4.1 Tampilan Halaman Menu Utama	64
Gambar 4.2 Tampilan Halaman Menu Info	64
Gambar 4.3 Tampilan Halaman Menu <i>Levenshtein Distance</i> pada Tombol <i>Test LD</i>	65
Gambar 4.4 Tampilan Halaman Menu <i>Levenshtein Distance</i> pada Tombol <i>Levenshtein Distance</i>	65
Gambar 4.5 Tampilan Halaman Menu Tentang	66
Gambar 4.6 Tampilan Halaman Menu Keluar	67

DAFTAR TABEL

	Halaman
Tabel 2.1	Simbol-simbol <i>Use Case</i> Diagram 14
Tabel 2.2	Simbol-simbol <i>Activity</i> Diagram 16
Tabel 2.3	Simbol-simbol <i>Sequence</i> Diagram 17
Tabel 2.4	Simbol-simbol <i>Flowchart</i> 19
Tabel 3.1	Tampilan perhitungan <i>string</i> awal dan <i>string</i> target 27
Tabel 3.2	Hasil perhitungan <i>string</i> awal dan <i>string</i> target untuk baris pertama 28
Tabel 3.3	Hasil perhitungan <i>string</i> awal dan <i>string</i> target untuk baris kedua 32
Tabel 3.4	Hasil perhitungan <i>string</i> awal dan <i>string</i> target untuk baris ketiga 37
Tabel 3.5	Hasil perhitungan <i>string</i> awal dan <i>string</i> target untuk baris keempat 41
Tabel 3.6	Hasil perhitungan <i>string</i> awal dan <i>string</i> target untuk hasil akhir 45
Tabel 4.1	Pengujian Menu Utama 69

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. <i>Script</i> Program	L-1
Lampiran 2. Permohonan Meja Hijau	L-2
Lampiran 3. Permohonan Judul Tesis/Skripsi/Tugas Akhir	L-3
Lampiran 4. Hasil <i>Plagiat Checker</i>	L-4
Lampiran 5. Kartu Bebas Praktikum	L-5
Lampiran 6. Berita Acara Bimbingan Skripsi	L-6
Lampiran 7. Izin Riset dari Universitas Pembangunan Panca Budi . . .	L-7
Lampiran 8. Izin Melaksanakan Riset dari SMP Negeri 2 Babalan . . .	L-8

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
DAFTAR LAMPIRAN	vi
BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
BAB II LANDASAN TEORI	
2.1. Pengertian Kata	5
2.2. Pengertian Data	6
2.3. <i>Levenshtein Distance</i>	7
2.3.1 Pengertian <i>Levenshtein Distance</i>	7
2.3.2 Langkah-Langkah <i>Levenshtein Distance</i>	9
2.4. <i>Unified Modelling Language (UML)</i>	12
2.5. Macam-macam Diagram <i>Unified Modelling Language (UML)</i> ...	12
2.5.1 <i>Use Case Diagram</i>	14
2.5.2 <i>ActivityDiagram</i>	15
2.5.3 <i>SequenceDiagram</i>	16
2.6. <i>Flowchart</i> ..	18
2.7. <i>Visual Basic</i>	20
BAB III METODE PENELITIAN	
3.1 Tahapan Penelitian	23
3.2 Tahapan Pengumpulan Data	24
3.3 Analisa Sistem yang Sedang Berjalan	25
3.4 Analisa Sistem yang Diusulkan	26
3.5 Pencarian Jarak Metode <i>Levenshtein Distance</i>	26
3.6 Perancangan Sistem	49
3.6.1 <i>Use Case Diagram</i>	49
3.6.2 <i>Activity Diagram</i>	50
3.6.3 <i>Sequence Diagram</i>	51

3.7	<i>Flowchart</i> dari Sistem Pendeteksi Penyesuaian Data Menggunakan <i>Levenshtein Distance</i>	56
3.8	Perancangan Antarmuka	57
3.8.1	Menu Utama	57
3.8.2	Menu Info	57
3.8.3	Menu <i>Levenshtein Distance</i>	58
3.8.4	Menu Tentang	59
3.8.5	Menu Keluar	60
BAB IV HASIL DAN PEMBAHASAN		
4.1	Deskripsi Hasil Penelitian	61
4.2	Komponen Utama dalam Implementasi Sistem	61
4.3	Implementasi Antar Muka	63
4.3.1	Halaman Menu Utama	63
4.3.2	Halaman Menu Info	64
4.3.3	Halaman Menu <i>Levenshtein Distance</i>	65
4.3.4	Halaman Menu Tentang	66
4.3.5	Halaman Menu Keluar	66
4.4	Pengujian Sistem	67
BAB V PENUTUP		
5.1	Kesimpulan	70
5.2	Saran	71
DAFTAR PUSTAKA		72
BIOGRAFI PENULIS		
LAMPIRAN		

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT atas berkah dan kemurahan-Nya kepada kita semua, khususnya kepada penulis sehingga Alhamdulillah penyusunan skripsi dapat diselesaikan. Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat dalam menempuh Ujian Akhir untuk memperoleh Gelar Sarjana Komputer pada Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan, serta sebagai bahan pertanggung jawaban penulis terhadap penyusunan skripsi dengan judul :

“SISTEM PENDETEKSI PENYESUAIAN KESALAHAN INPUT DATA SISWA MENGGUNAKAN METODE *LEVENSHTTEIN DISTANCE*.”

Dalam penyusunan skripsi ini penulis menyadari banyak mengalami kesulitan namun berkat bantuan dan dorongan dari berbagai pihak, akhirnya penyusunan skripsi ini dapat juga diselesaikan. Penulis dengan segala kerendahan hati menyampaikan terima kasih kepada :

1. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M selaku Rektor Universitas Pembangunan Panca Budi Medan.
2. Bapak Hamdani, ST., MT, selaku Dekan Fakultas Ilmu Komputer Universitas Pembangunan Panca Budi Medan.
3. Bapak Eko Haryanto, S.Kom, M.Kom selaku Ketua Program Studi Sistem Komputer pada Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Dosen Pembimbing I Bapak Muhammad Iqbal, S.Kom, M.Kom.
5. Dosen Pembimbing II Ibu Ranti Eka Putri, S.Kom, M.Kom.
6. Orang tua khususnya Almarhum Bapak Sartono dan Ibu Sumirah, serta kakak dan adik-adik dan juga keponakan-keponakan yang selalu mendoakan, memberi semangat moral dan spiritual, serta selalu berharap yang terbaik bagi Penulis.
7. Bapak Sahtiar, S.Pd, M.Pd selaku kepala sekolah SMP Negeri 2 Babalan Pangkalan Berandan.
8. Guru dan Tata Usaha di SMP Negeri 2 Babalan Pangkalan Berandan.
9. Teman-teman yang ikut memberi masukan dan dukungan dalam laporan ini terkhusus buat Harip, Rianto, Mentari dan Iin Puspita Dewi.
10. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dan memberikan dorongan serta masukan sebagai acuan dalam pembuatan skripsi ini. Semoga Allah SWT membalas segala kebaikan Anda semua, Aamiin.

Semoga Skripsi ini dapat memberikan semangat untuk terus menggali ilmu terutama dalam bidang Sains dan Teknologi.

Medan, Februari 2020
Penulis,

SUMIRATNO
NPM. 1514371005

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dunia pendidikan dan dunia kerja merupakan salah satu yang paling banyak memanfaatkan teknologi dalam hal ini teknologi informasi, karena teknologi informasi menyediakan proses yang cepat, tepat, dan menghasilkan tingkat keakuratan yang lebih. Dalam lingkungan perusahaan dan sekolah, komputer adalah alat bantu yang sangat mutlak diperlukan sebagai pendukung pemrosesan data dan informasi. Tetapi sampai saat ini masih banyak yang melakukan proses pengolahan data secara manual, sehingga menyebabkan beberapa permasalahan dan kendala seperti kesalahan-kesalahan yang tidak disengaja yang dapat menyebabkan kebenaran dari data kurang terjamin, serta pencarian data yang membutuhkan waktu yang lama dan banyak lagi permasalahan lainnya yang timbul dari pengolahan data secara manual.

Salah satu cara untuk menghasilkan informasi berbasis teknologi adalah sistem informasi yang terkomputerisasi. Dengan menggunakan sistem informasi yang terkomputerisasi maka pengguna sistem bisa mendapatkan informasi yang akurat secara efektif dan efisien. Sekolah Menengah Pertama Negeri 2 Babalan Pangkalan Berandan, merupakan sekolah negeri dimana dalam sistem administrasi akademiknya masih menggunakan sistem manual. Pencatatan dan penyimpanan data selama ini ditulis secara manual salah satunya seperti penulisan nama-nama siswa dan siswi yang masih ditulis tangan di buku induk, buku klapper siswa serta

pengetikan nama siswa tersebut masih menggunakan *Ms. Excel*. Akurasi data yang dihasilkan dari nama siswa pun terkadang mengalami kesalahan yang telah ditulis sebelumnya. Pada penelitian ini, penulis merancang sebuah aplikasi untuk mendeteksi kesalahan input data siswa khususnya pada nama siswa dengan uji coba menggunakan salah satu metode yang tepat untuk melakukan deteksi kemiripan dokumen teks adalah dengan melakukan perhitungan dengan metode *Levenshtein Distance*. *Levenshtein Distance* memperhatikan tiga operasi dalam menentukan jarak diff, yaitu (1) operasi penyisipan (*insertion*), (2) operasi penghapusan (*deletion*), (3) operasi penggantian (*subtitution*), sebuah huruf yang berdekatan. Hasil akhir yang diberikan dalam sistem ini adalah memberikan presentase nilai *similarity* antara kedua dokumen (Ariyani, Sutardi, & Ramadhan, 2016).

Dengan menggunakan sistem komputer, akan dibangun sebuah program aplikasi untuk menerapkan metode *Levenshtein Distance* dalam pengecekan kesalahan penginputan data nama siswa berdasarkan data nama yang sudah adasebelumnya sehingga dapat meminimalisir kesalahan tersebut. Hasil dari pengecekan nama siswa tersebut digunakan untuk pendataan Peserta Ujian Nasional dengan menggunakan metode *Levenshtein Distance* dalam bentuk tugas akhir dengan judul **“Sistem Pendeteksi Penyesuaian Kesalahan Input Data Siswa Menggunakan Metode *Levenshtein Distance*.”**

1.2. Perumusan Masalah

Mengingat pentingnya pengolahan data dan informasi yang berjalan di SMP Negeri 2 Babalan Pangkalan Berandan dirasakan kurang efektif dan efisien dalam pengetikan nama siswa tersebut, sehingga perlu adanya perancangan sistem secara komputerisasi.

Berdasarkan latar belakang diatas, maka dapat dirumuskan beberapa permasalahan antara lain :

1. Bagaimana caranya untuk melakukan pengecekan ketelitian *entry* data nama siswa yang salah input pada saat pengetikan?
2. Bagaimana hasil nilai perhitungan menggunakan metode *Levenshtein Distance*?

1.3. Batasan Masalah

Dalam menganalisa dan menyelesaikan suatu masalah, maka perludiberikan pembatasan atau ruang lingkup pembahasan. Adapun batasan-batasan masalah adalah sebagai berikut :

1. Data yang diuji adalah nama siswa kelas IX yang akan mengikuti Calon Peserta Ujian Nasional tahun 2019/2020, dengan menggunakan huruf kapital karena jika tidak menggunakan huruf kapital hasil dari perhitungan *levenshtein Distance* akan berbeda.
2. Sistem yang digunakan tidak menggunakan *stemming* dan *stopword*.
3. Penggunaan *Visual Basic 10* untuk sistem informasi pengecekan kesalahan nama siswa yang selama ini masih dilakukan secara manual.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam tugas akhir ini adalah :

1. Untuk merancang suatu sistem dalam melakukan pengecekan ketelitian data untuk nama siswa.
2. Sistem ini untuk menganalisis hasil perhitungan menggunakan metode *Levenshtein Distance*.

1.5. Manfaat Penelitian

Adapun manfaat yang akan diperoleh dalam pembuatan sistem ini adalah sebagai berikut:

1. Memberikan informasi nama siswa yang akurat dan benar dalam pengolahan datanya sehingga kesalahan-kesalahan yang sering dialami bisa diperbaiki datanya.
2. Mempermudah pihak sekolah melakukan tes ketelitian dan kecepatan *entry* data pada nama siswa sehingga dapat mempercepat dalam pengecekan nama siswa yang sesuai kebutuhan.

BAB II

LANDASAN TEORI

2.1 Pengertian Kata

Sebagai alat komunikasi atau interaksi antar manusia, bahasa memiliki satuan-satuan yang digunakan untuk mengungkapkan, menyampaikan ataupun menuliskan sesuatu. Dalam mendeskripsikan banyak bahasa di dunia, diperlakukan sebuah unit yang disebut dengan “kata”, namun bagi sebagian pengertian kata dibatasi secara fonologis, sedangkan bagi bahasa yang lain dibatasi secara morfologis.

Berikut defenisi kata menurut beberapa ahli yaitu :

1. Menurut (Abdul Chaer, 2011:16) dalam bahasa, terdapat urutan dari satuan-satuan yang dimiliki bahasa, mulai dari yang terkecil hingga terbesar yaitu: kata, frase, klausa, kalimat, paragraf dan wacana.
2. Menurut (Kridalaksana, 2008:110) kata (*word*) memiliki pengertian 1) morfem atau kombinasi morfem yang oleh bahasawan dianggap sebagai satuan terkecil yang dapat diujarkan sebagai bentuk bebas, 2) satuan bahasa yang dapat berdiri sendiri terdiri atas morfem tunggal atau gabungan morfem, 3) satuan terkecil dalam sintaksis yang berasal dari leksem yang telah mengalami proses morfologis.
3. Menurut (Keraf, 1991:44) kata adalah satuan-satuan terkecil yang diperoleh sesudah sebuah kalimat dibagi atas bagian-bagiannya, dan mengandung sebuah ide.

4. Menurut (Kushartanti, 2005:151) kata adalah satuan bebas yang paling kecil, atau dengan kata lain setiap satuan bebas merupakan kata.

Sepertiyang telah kita lihat pada paparan diatas, bahwa "kata" merupakan bentuk terkecil dari satuan bahasa. Meskipun bentuk terkecil, tapi ia adalah bentuk terdasar dalam landasan bahasa dan tentu saja dalam komunikasi dan interaksi manusia dalam kehidupan sehari-hari. Istilah kata sering kita dengar dan kita gunakan. Jadi dapat disimpulkan bahwa kata adalah satuan bebas, atau bentuk yang paling kecil, mampu berdiri sendiri, dan sudah mempunyai arti. Kata merupakan dua macam satuan, ialah satuan fonologik dan satuan gramatik. Sebagai satuan fonologi, kata terdiri satu atau beberapa suku, dan suku itu terdiri dari satu atau beberapa fonem. Sebagai satuan gramatik, kata terdiri dari satu atau beberapa morfem.

2.2 Pengertian Data

Dari segi Bahasa kata "data" ini diambil dari kata "*datum*" yang dalam Bahasa Romawi memiliki arti sebagai sesuatu yang diberikan. Oleh sebab itu itu definisi sesungguhnya dari data ini ialah diberikan bukan memberikan, sebab apabila memberikan maka data itu sudah menjadi informasi yang baku serta juga diakui kebenarannya. Data yang diperoleh tersebut dapat menjadi sebuah anggapan atau fakta disebabkan karena memang belum diolah dengan lebih lanjut. Setelah diolah dengan melalui penelitian atau percobaan maka suatu data tersebut dapat menjadi bentuk yang lebih kompleks seperti suatu database, informasi atau juga bahkan solusi untuk menyelesaikan masalah tertentu.

Data adalah sesuatu yang diberikan untuk kemudian diolah (Taufiq, 2013:13).

Data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan tetapi belum terorganisir untuk menyampaikan arti tertentu (Turban, 2010:41).

Data adalah kumpulan dari fakta, konsep, atau instruksi pada penyimpanan yang digunakan untuk komunikasi, perbaikan dan diproses secara otomatis yang mempresentasikan informasi yang dapat di mengerti oleh manusia (Inmon, 2005:493).

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar tetapi belum diolah.

2.3 *Levenshtein Distance*

Edit distance dalam komputer *science* dikenal dengan nama *Levenshtein Distance*. Nama tersebut diambil dari nama penemunya, yaitu Vladimir Levenshtein. *Levenshtein Distance* ditemukan pada tahun 1965.

2.3.1 *Pengertian Levenshtein Distance*

Levenshtein Distance lebih dikenal dengan *edit distance*. *Levenshtein Distance* merupakan algoritma yang digunakan untuk mencari jumlah operasi *string* yang paling sedikit untuk mentransformasikan suatu *string* menjadi *string* yang lain (Mulyanto, 2010).

Algoritma ini digunakan dalam pencarian *string* dengan pendekatan perkiraan (*Approximate String Matching*) (Adiwidya, 2009).

Levenshtein Distance adalah sebuah matriks untuk mengukur angka perbedaan antara 2 *string*, jarak antara *string* diukur berdasarkan angka penambahan karakter, penghapusan karakter ataupun penggantian karakter yang diperlukan untuk mengubah *string* sumber menjadi *string* target (Ishak, dkk., 2012).

Pada *Levenshtein Distance* ada tiga macam operasi utama, yaitu :

1. Operasi Pengubahan Karakter

Pada operasi ini dapat kita lakukan dengan menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan *string* “bagos” menjadi “bagus”. Dalam kasus ini karakter “o” diganti dengan huruf “u”.

2. Operasi Penambahan Karakter

Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu *string*. Contohnya *string* “bersi” menjadi *string* “bersih”, dilakukan penambahan karakter “h” di akhir *string*. Penambahan karakter tidak hanya dilakukan diakhir kata, namun bisa ditambahkan diawal maupun disisipkan di tengah *string*.

3. Operasi Penghapusan Karakter

Operasi penghapusan karakter dapat dilakukan untuk menghilangkan karakter dari suatu *strings* saja. Contohnya *string* “kursis” karakter terakhir dihilangkan sehingga menjadi *string* “kursi”.

Penggunaan metode ini cukup luas, tidak hanya untuk mendeteksi plagiarisme, tetapi juga untuk hal-hal lainnya yang berkaitan dengan pencocokan *string*.

2.3.2 Langkah-Langkah *Levenshtein Distance*

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* (Adriyani, Santiyasa, & Muliantara, 2010). Nilai *cost* dapat ditambah dengan nol ataupun satu, dengan ketentuan jika membandingkan huruf yang sama maka ditambah dengan nol dan jika membandingkan huruf yang berbeda maka ditambah dengan satu. Jarak (*distance*) pada metode ini adalah nilai minimum dari operasi pengubahan, penambahan dan penghapusan karakter yang dibutuhkan untuk merubah *string* awal (s) menjadi *string* target (t), contoh :

1. Jika s adalah “aku” dan t adalah “aku”, maka jarak antara kedua *string* adalah nol, dikarenakan tidak ada transformasi yang dibutuhkan. Maka dapat disimpulkan kedua *string* tersebut identik.
2. Jika s adalah “aku” dan t adalah “abu”, maka jarak antara kedua *string* tersebut adalah satu, dikarenakan adanya pengubahan karakter dimana huruf “k” pada kata “aku” diubah menjadi huruf “b” pada kata “abu”.

Proses algoritma *levenshtein distance* membandingkan kata “aku” sebagai *string* awal (s) dengan “abu” sebagai *string* target (t), yaitu :

1. Membuat Matriks

t/s		a	b	u
	0	1	2	3
a	1	A	B	C
k	2			
u	3			

2. Melakukan pencocokan dengan melakukan perbandingan dari setiap karakter s dengan karakter t, dengan ketentuan :

t/s		a	b	u
	0	1	2	3
a	1	0	1	2
k	2			
u	3			

COST = jika isi t dan s sama, contoh di atas t = a dan s = a, maka COST bernilai 0 (nol).

$$A1 = \text{Nilai di atas A ditambah } 1 = (1+1) = 2$$

$$A2 = \text{Nilai di kiri A ditambah } 1 = (1+1) = 2$$

$$A3 = \text{Nilai di kiri atas A ditambah COST} = (0+0) = 0$$

A = Nilai yang paling kecil antara A1 sampai A3 yaitu $A3 = 0$

COST = jika isi t dan s beda, contoh di atas $t = b$ dan $s = a$, maka COST bernilai 1 (satu).

$$B1 = \text{Nilai di atas B ditambah 1} = (2+1) = 3$$

$$B2 = \text{Nilai di kiri B (A) ditambah 1} = (0+1) = 1$$

$$B3 = \text{Nilai di kiri atas B ditambah COST} = (1+1) = 2$$

$$B = \text{Nilai yang paling kecil antara B1 sampai B3 yaitu } B2 = 1$$

COST = jika isi t dan s beda, contoh di atas $t = u$ dan $s = a$, maka COST bernilai 1 (satu).

$$C1 = \text{Nilai di atas C ditambah 1} = (3+1) = 4$$

$$C2 = \text{Nilai di kiri C (B) ditambah 1} = (1+1) = 2$$

$$C3 = \text{Nilai di kiri atas C ditambah COST} = (2+1) = 3$$

$$C = \text{Nilai yang paling kecil antara C1 sampai C3 yaitu } C2 = 2$$

3. Proses terus berjalan sampai akhir karakter s dan karakter t.

t/s		a	b	u
	0	1	2	3
a	1	0	1	2
k	2	1	1	2
u	3	2	2	1

Maka didapatkan jarak *string* awal (s) dengan *string* target (t) adalah satu. Nilai *edit distance*-nya yaitu pada ujung kanan bawah matriks.

2.4 *Unified Modelling Language (UML)*

Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa permodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu: *Unified Modeling Language (UML)*. *UML* muncul karena adanya kebutuhan permodelan *visual* untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa *visual* untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML (Unified Modeling Language) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem” (Mulyani, 2016:48).

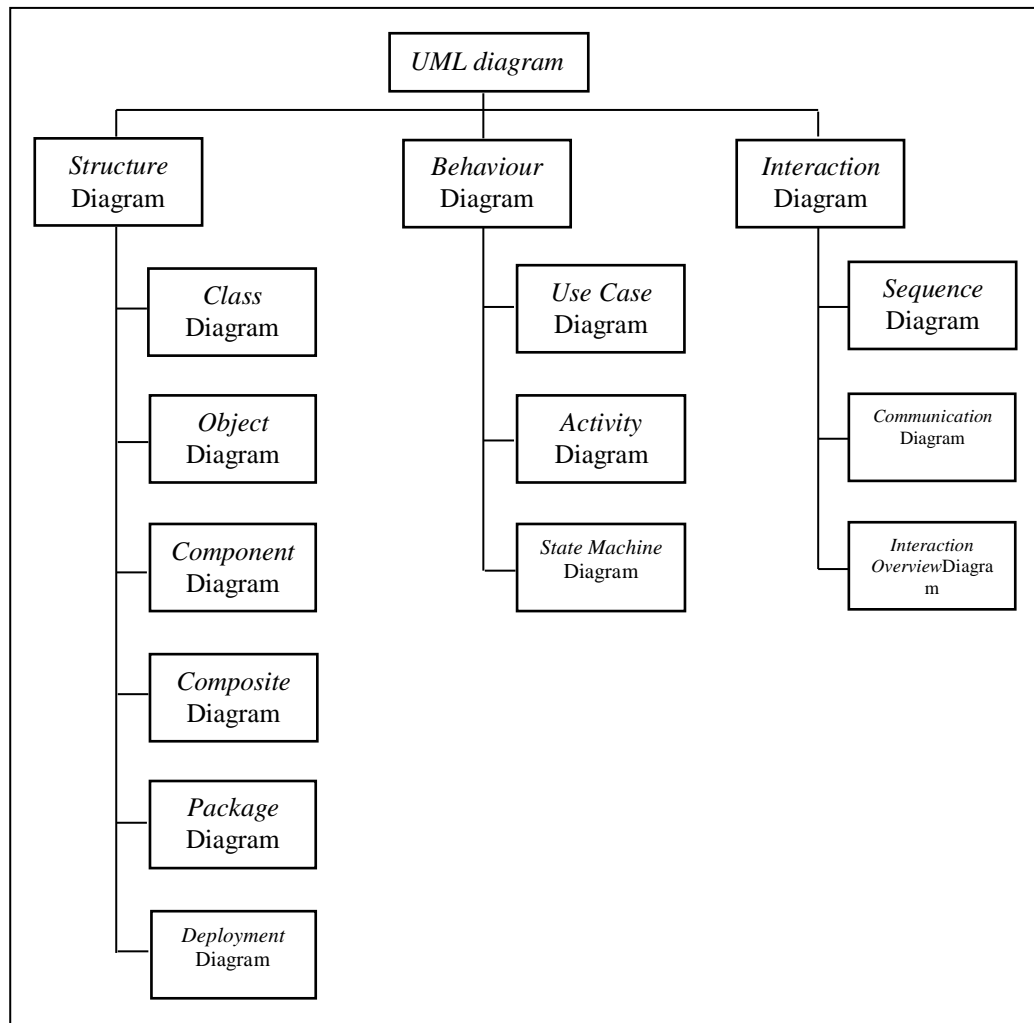
UML (Unified Modelling Language) adalah “Salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefenisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” (Sukamto dan Shalahuddin, 2014:133).

UML singkatan dari “*Unified Modelling Language* yang berarti bahasa permodelan standar” (Asropuddin, 2013:102).

2.5 **Macam-macam Diagram UML**

Unified Modelling Language (UML) terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori (Sukamto dan Shalahuddin, 2014:140).

Adapun pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini :



Gambar 2.1 Diagram *Unified Modelling Language*
 Sumber : Sukamto dan Shalahuddin (2014)

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. Behaviour Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

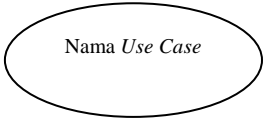
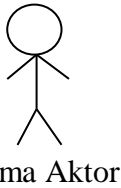
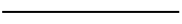
3. Interaction Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

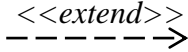

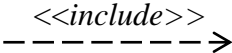
2.5.1 Use Case Diagram

“Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu” (Sukanto dan Shalahuddin, 2014:155). Adapun simbol-simbol yang digunakan dalam use case adalah sebagai berikut:

Tabel 2.1 Simbol-simbol Use Case Diagram

No	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
2.		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.		Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.

Tabel 2.1 Simbol-simbol *Use Case* Diagram (Lanjutan)


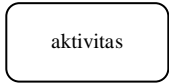
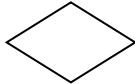


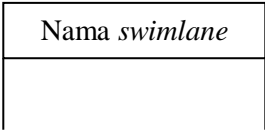
No	Simbol	Deskripsi
4.	Ekstensi / <i>Extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5.	Generalisasi / <i>Generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.	Menggunakan / <i>Include / Uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

Sumber: Buku Karangan Rosa A.S dan M. Shalahuddin (2014:156)

2.5.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggunakan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses atau menu yang ada pada perangkat lunak (Sukamto dan Shalahuddin, 2014:161). Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.2 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>Join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Buku Karangan Rosa A.S dan Shalahuddin (2014:162)

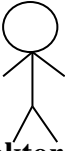


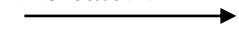
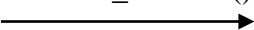

2.5.3 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sukanto dan Shalahuddin, 2014:165). Oleh karena itu, untuk menggambar *sequence diagram* harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi

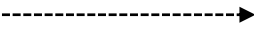
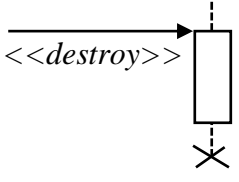
menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Simbol-simbol yang digunakan dalam *sequence diagram* adalah:

Tabel 2.3 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama aktor atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">nama aktor</div> <p>tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.	<p>Garis hidup / <i>Lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">nama objek : namakelas</div>	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	<p>Pesan tipe <i>create</i> <<create>></p> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe <i>call</i> 1 : nama_metode ()</p> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.	<p>Pesan tipe <i>send</i> 1 : masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

Tabel 2.3 Simbol-simbol *Sequence* Diagram (Lanjutan)

No	Simbol	Deskripsi
8.	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Buku Karangan Rosa A.S dan Shalahuddin (2014:165)

2.6 *Flowchart*

Flowchart merupakan suatu kumpulan atau aliran bagan yang menggambarkan terjadinya proses prosedur atau program. Jadi dalam suatu *flowchart* yang dibuat, bisa menggambarkan bagaimana suatu program itu berjalan. Sehingga jika penjelasan secara deskripsi masih kurang dipahami oleh orang lain, maka ada bagan atau *flowchart* yang bisa mudah dipahami karena tersusun lebih sederhana dan singkat. Diagram alur dapat menunjukkan secara jelas arus pengendalian suatu algoritma, yakni melaksanakan suatu rangkaian kegiatan secara logis dan sistematis (Sutabri, 2004:21).

Untuk memulai membuat *flowchart*, berikut ini merupakan hal-hal yang harus diperhatikan dalam proses pembuatannya :

1. *Flowchart* dimulai dari atas kebawah dan dari kiri ke kanan.



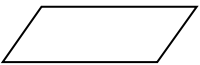
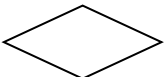

2. Proses yang digambarkan harus dijelaskan secara jelas dan dimengerti oleh pembaca.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan dengan jelas.
4. Harus dimulai dari urutan yang benar, jangan sampai terbalik-balik.
5. Gunakan simbol *flowchart* yang standar, dan sesuai dengan jalannya proses program.

Tujuan utama pembuatan *flowchart* ini adalah untuk menggambarkan suatu tahapan penyelesaian masalah sederhana, terurai, rapi dan jelas.

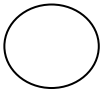
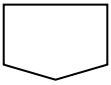
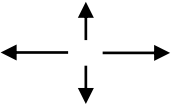
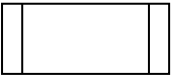

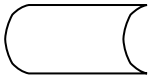
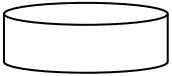
Flowchart atau diagram alir merupakan kumpulan simbol-simbol atau skema yang menunjukkan/menggambarkan rangkaian kegiatan-kegiatan program dari awal hingga akhir. *Flowchart* ini merupakan penggambaran dari urutan langkah-langkah pekerjaan dari suatu algoritma.

Adapun simbol-simbol *flowchart* lihat pada tabel sebagai berikut :

Tabel 2.4 Simbol – Simbol *Flowchart*

No	Simbol	Fungsi
1.		Terminal , untuk memulai atau mengakhiri suatu program
2.		Proses , suatu simbol yang menunjukkan setiap pengolahan yang dilakukan.
3.		Input-Output , untuk memasukkan menunjukkan hasil dari suatu proses
4.		Decision , suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5.		Preparation , suatu simbol yang menyediakan tempat pengolahan

Tabel 2.4 Simbol – Simbol *Flowchart* (Lanjutan)

No	Simbol	Fungsi
6.		Connector , suatu prosedur penghubung yang akan masuk atau keluar melalui simbol ini dalam lembar yang sama
7.		Off-Page Connector , merupakan simbol masuk atau keluarannya suatu prosedur pada lembaran kertas lainnya
8.		Arus/Flow , dari pada prosedur yang dapat dilakukan atas ke bawah dari bawah ke atas, ke atas dari kiri ke kanan ataupun dari kanan ke kiri
9.		Predefined Process , untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
10.		Simbol untuk <i>output</i> , yang ditunjukkan ke suatu device, seperti printer, dan sebagainya
11.		Penyimpanan file secara sementara
12.		Menunjukkan <i>input</i> / <i>Output Hardisk</i> (media penyimpanan)

Sumber : Rasim, Setiawan, dan Rahman (2008:8)

2.7 Visual Basic

Visual Basic.NET adalah *Visual Basic* yang direkayasa kembali untuk digunakan pada *platform.NET* sehingga aplikasi yang dibuat menggunakan *Visual Basic.NET* dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari *server* dengan tipe apa pun asalkan terinstall *NET framework* (Hidayatullah, 2014:5).

Visual Basic.NET 2010 tidak jauh berbeda dengan *Visual Basic.NET* sebelumnya atau *Visual Basic 6.0 IDE (Interface Development Environment)* (Darmayuda, 2014:14).

Visual Basic.NET ini memang dibuat supaya mudah dipahami dan dipelajari, namun bahasa pemrograman ini juga cukup *powerful* untuk memenuhi kebutuhan dari *programmer* yang berpengalaman. Bahasa pemrograman *Visual Basic.NET* mirip dengan bahasa pemrograman *Visual Basic*, namun keduanya tidak sama.

Bahasa pemrograman *Visual Basic.NET* memiliki struktur penulisan yang mirip dengan bahasa Inggris, di mana hal ini juga menyebabkan kemudahan dalam membaca dan mengerti dari sebuah kode *Visual Basic.NET*. Di mana dimungkinkan, kata ataupun frasa yang memiliki arti digunakan dan bukannya menggunakan singkatan, akronim ataupun *specialcharacters*. Pada intinya *Visual Basic.NET* ini adalah sebuah bahasa pemrograman yang berorientasi pada *object*, yang bisa dianggap sebagai evolusi selanjutnya dari bahasa pemrograman *Visual Basic* standar.

Adapun antarmuka *Visual Basic.NET* 2010 yaitu :

1. *Title Bar*, berfungsi untuk menampilkan nama *project* yang aktif atau sedang dikembangkan.
2. *Menu Bar*, berfungsi untuk pengelolaan fasilitas yang dimiliki oleh *Visual Basic.NET* 2010, sedangkan *Tool Bar*, berfungsi untuk melakukan perintah khusus secara cepat.

3. *Form*, adalah objek utama berfungsi untuk melakukan objek-objek yang terdapat pada *Toolbox* yang digunakan dalam melakukan perancangan sebuah tampilan program aplikasi.
4. *Toolbox*, berfungsi untuk menyediakan objek-objek atau komponen yang digunakan dalam merancang sebuah *form* pada program aplikasi.
5. *Solution Explorer*, berfungsi untuk menampilkan nama *project*, *file* konfigurasi beserta folder, *file-file* pendukung yang terdapat pada sebuah program aplikasi.
6. *Properties Window*, berfungsi untuk mengatur *properties-properties* pada objek (*settingobject*) yang diletakkan pada sebuah *form*.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian dibuat oleh penulis dengan menggunakan perhitungan *Levenshtein Distance* untuk mencari kesalahan kata pada pengetikan input data nama-nama siswa kelas IX yang akan mengikuti Ujian Nasional Tahun Pelajaran 2019/2020.

Adapun tahapan pada penelitian ini yaitu :

1. Studi Literatur

Dalam melakukan penelitian ilmiah harus dilakukan teknik penyusunan yang sistematis untuk memudahkan langkah-langkah yang akan diambil. Begitu pula yang dilakukan penulis dalam penelitian ini, langkah pertama yaitu dengan melakukan studi literatur. Dengan studi literatur tidak harus turun ke lapangan dan bertemu dengan responden. Data-data yang dibutuhkan dalam penelitian dapat diperoleh dari sumber pustaka atau dokumen. Penelitian dengan studi literatur adalah penelitian yang persiapannya sama dengan penelitian lainnya akan tetapi sumber dan metode pengumpulan data dengan mengambil data di pustaka, membaca, mencatat, dan mengolah bahan penelitian. Meskipun terlihat mudah, studi literatur membutuhkan ketekunan yang tinggi agar data dan analisis data serta kesimpulan yang dihasilkan sesuai dengan tujuan yang diharapkan. Untuk itu dibutuhkan persiapan

dan pelaksanaannya yang optimal. Penelitian studi literatur membutuhkan analisis yang matang dan mendalam agar mendapatkan hasil.

2. Analisa

Analisa terhadap permasalahan dan penentuan model penyelesaian terhadap suatu masalah, termaksud dalam proses ini adalah melakukan analisis terhadap permasalahan yang terjadi dan bagaimana cara menyelesaikannya.

3. Pengujian

Ditahapan pengujian hasil aplikasi atau program sistem pendeteksi penyesuaian kesalahan input data siswa yang akan dibuat dengan *Microsoft Visual Basic* serta dibandingkan dengan perhitungan manual yang dilakukan secara manual.

3.2 Tahapan Pengumpulan Data

Pengumpulan data merupakan tahapan yang sangat penting dalam sebuah penelitian. Pengumpulan data merupakan cara yang digunakan oleh peneliti untuk mengumpulkan data-data penelitian dari sumber data (subyek maupun sampel penelitian). Teknik pengumpulan data merupakan suatu kewajiban, karena teknik pengumpulan data ini nantinya digunakan sebagai dasar untuk menyusun instrumen penelitian.

Metode pengumpulan data penulis membagi menjadi dua bagian antara lain yaitu :

1. Studi Keputusan Data

Studi keputusan dilakukan dengan cara mengumpulkan data, membaca, mempelajari, dan mengambil referensi baik itu buku, jurnal, makalah, internet dan berbagai sumber sehingga memperoleh informasi.

2. Pengamatan

Pengamatan dilakukan sebagai pengumpulan data, bahan, studi lapangan dengan cara terjun langsung ke lapangan supaya dapat mengetahui kegiatan dan aspek yang ada di SMP Negeri 2 Babalan Pangkalan Berandan.

3.3 Analisa Sistem yang Sedang Berjalan

Analisa sistem merupakan gambaran tentang sistem yang saat ini sedang berjalan di SMP Negeri 2 Babalan, tujuan dari analisa sistem ini yaitu untuk mencari kesalahan pengetikan nama siswa yang terdapat di data Dapodik dan kemudian akan disesuaikan dengan Ijazah yang sudah dikumpulkan sebelumnya. Sistem yang digunakan di SMP Negeri 2 Babalan masih sederhana dan manual yaitu dengan cara mengumpulkan Ijazah Sekolah Dasar (SD) dari siswa dan siswi yang akan mengikuti Ujian Nasional tahun ajaran 2019/2020, yang berfungsi sebagai pedoman untuk pengecekan data khususnya nama siswa dan siswisajadengan cara yaitu membandingkan penulisan nama siswa dan siswi tersebut yang ada di Ijazah dengan daftar nama yang sudah diunduh sebelumnya dari dapodik.

3.4 Analisa Sistem yang Diusulkan

Analisa sistem adalah metode untuk menemukan kelemahan-kelemahan sistem guna memperoleh gambaran terhadap sistem yang akan dikembangkan sehingga dapat diusulkan perbaikannya. Tahapan dalam menganalisa sistem diawali dengan mempelajari bagaimana mengidentifikasi masalah-masalah yang dihadapi. Analisa kebutuhan sistem dimaksudkan untuk mengetahui faktor-faktor apa saja yang mempengaruhi sistem. Faktor-faktor tersebut akan menjadi tolak ukur dalam proses pengembangan sistem selanjutnya.

Usulan sistem baru dapat dilakukan setelah melakukan analisa terhadap sistem yang sedang berjalan. Sistem baru yang dirancang ini merupakan perubahan dari sistem informasi yang selama ini diterapkan dengan dibantu oleh sebuah aplikasi bahasa pemrograman *Microsoft Visual Basic 10*.

Maka dalam hal ini untuk menyelesaikan permasalahan tersebut SMP Negeri 2 Babalan akan menggunakan metode *Levenshtein Distance* dengan menentukan jarak berdasarkan kepada nilai yang paling terkecil atau paling sedikit jumlah modifikasinya, sehingga *string* atau kata yang memiliki nilai modifikasi yang paling sedikit saat dibandingkan dengan *string* atau kata lain dianggap sebagai kata yang cocok atau paling mendekati.

3.5 Pencarian Jarak Metode *Levenshtein Distance*.

Adapun langkah-langkah proses pencarian jarak *Levenshtein Distance* :

1. Membandingkan *string*
2. Melakukan modifikasi dan perhitungan jarak *Levenshtein Distance*.

Tabel 3.2 Hasil perhitungan *string* awal dan *string* target untuk baris pertama

(Lanjutan)

t/s		T	I	A	R	A	_	M	A	U	L	I	D	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
L	10													
I	11													
D	12													
A	13													
_	14													
A	15													
L	16													
I	17													

COST = jika isi t dan s sama, contoh di atas t = 1 T dan s = 1 T, maka COST bernilai nol (0).

$$A1 = \text{Nilai di atas A ditambah } 1 = (1+1) = 2$$

$$A2 = \text{Nilai di kiri A ditambah } 1 = (1+1) = 2$$

$$A3 = \text{Nilai di kiri atas A ditambah COST} = (0+0) = 0$$

$$A = \text{Nilai yang paling kecil antara A1 sampai A3 yaitu } A3 = 0$$

COST = jika isi t dan s beda, contoh di atas t = 2 I dan s = 1 T, maka COST bernilai 1 (satu).

$$B1 = \text{Nilai di atas B ditambah } 1 = (2+1) = 3$$

$$B2 = \text{Nilai di kiri B (A) ditambah } 1 = (0+1) = 1$$

$$B3 = \text{Nilai di kiri atas B ditambah COST} = (1+1) = 2$$

$$B = \text{Nilai yang paling kecil antara B1 sampai B3 yaitu } B2 = 1$$

COST = jika isi t dan s beda, contoh di atas t = 3 A dan s = 1 T, maka COST bernilai 1 (satu).

$$C1 = \text{Nilai di atas C ditambah } 1 = (3+1) = 4$$

$$C2 = \text{Nilai di kiri C (B) ditambah } 1 = (1+1) = 2$$

$$C3 = \text{Nilai di kiri atas C ditambah COST} = (2+1) = 3$$

$$C = \text{Nilai yang paling kecil antara C1 sampai C3 yaitu } C2 = 2$$

COST = jika isi t dan s beda, contoh di atas t = 4 R dan s = 1 T, maka COST bernilai 1 (satu).

$$D1 = \text{Nilai di atas D ditambah } 1 = (4+1) = 5$$

$$D2 = \text{Nilai di kiri D (C) ditambah } 1 = (2+1) = 3$$

$$D3 = \text{Nilai di kiri atas D ditambah COST} = (3+1) = 4$$

$$D = \text{Nilai yang paling kecil antara D1 sampai D3 yaitu } D2 = 3$$

COST = jika isi t dan s beda, contoh di atas t = 5 A dan s = 1 T, maka COST bernilai 1 (satu).

$$E1 = \text{Nilai di atas E ditambah } 1 = (5+1) = 6$$

$$E2 = \text{Nilai di kiri E (D) ditambah } 1 = (3+1) = 4$$

$$E3 = \text{Nilai di kiri atas E ditambah COST} = (4+1) = 5$$

$$E = \text{Nilai yang paling kecil antara E1 sampai E3 yaitu } E2 = 4$$

COST = jika isi t dan s beda, contoh di atas t = 6 _ dan s = 1 T, maka COST bernilai 1 (satu).

$$F1 = \text{Nilai di atas F ditambah } 1 = (6+1) = 7$$

$$F2 = \text{Nilai di kiri F (E) ditambah } 1 = (4+1) = 5$$

$$F3 = \text{Nilai di kiri atas F ditambah COST} = (5+1) = 6$$

$$F = \text{Nilai yang paling kecil antara F1 sampai F3 yaitu } F2 = 5$$

COST = jika isi t dan s beda, contoh di atas $t = 7$ M dan $s = 1$ T, maka COST bernilai 1 (satu).

$$G1 = \text{Nilai di atas G ditambah } 1 = (7+1) = 8$$

$$G2 = \text{Nilai di kiri G (F) ditambah } 1 = (5+1) = 6$$

$$G3 = \text{Nilai di kiri atas G ditambah COST} = (6+1) = 7$$

$$G = \text{Nilai yang paling kecil antara G1 sampai G3 yaitu } G2 = 6$$

COST = jika isi t dan s beda, contoh di atas $t = 8$ A dan $s = 1$ T, maka COST bernilai 1 (satu).

$$H1 = \text{Nilai di atas H ditambah } 1 = (8+1) = 9$$

$$H2 = \text{Nilai di kiri H (G) ditambah } 1 = (6+1) = 7$$

$$H3 = \text{Nilai di kiri atas H ditambah COST} = (7+1) = 8$$

$$H = \text{Nilai yang paling kecil antara H1 sampai H3 yaitu } H2 = 7$$

COST = jika isi t dan s beda, contoh di atas $t = 9$ U dan $s = 1$ T, maka COST bernilai 1 (satu).

$$I1 = \text{Nilai di atas I ditambah } 1 = (9+1) = 10$$

$$I2 = \text{Nilai di kiri I (H) ditambah } 1 = (7+1) = 8$$

$$I3 = \text{Nilai di kiri atas I ditambah COST} = (8+1) = 9$$

$$I = \text{Nilai yang paling kecil antara I1 sampai I3 yaitu } I2 = 8$$

COST = jika isi t dan s beda, contoh di atas $t = 10$ L dan $s = 1$ T, maka COST bernilai 1 (satu).

$$J1 = \text{Nilai di atas J ditambah } 1 = (10+1) = 11$$

$$J2 = \text{Nilai di kiri J (I) ditambah } 1 = (8+1) = 9$$

$$J3 = \text{Nilai di kiri atas J ditambah COST} = (9+1) = 10$$

J = Nilai yang paling kecil antara J_1 sampai J_3 yaitu $J_2 = 9$

COST = jika isi t dan s beda, contoh di atas $t = 11$ I dan $s = 1$ T, maka COST bernilai 1 (satu).

K_1 = Nilai di atas K ditambah 1 = $(11+1) = 12$

K_2 = Nilai di kiri K (J) ditambah 1 = $(9+1) = 10$

K_3 = Nilai di kiri atas K ditambah COST = $(10+1) = 11$

K = Nilai yang paling kecil antara K_1 sampai K_3 yaitu $K_2 = 10$

COST = jika isi t dan s beda, contoh di atas $t = 12$ D dan $s = 1$ T, maka COST bernilai 1 (satu).

L_1 = Nilai di atas L ditambah 1 = $(12+1) = 13$

L_2 = Nilai di kiri L (K) ditambah 1 = $(10+1) = 11$

L_3 = Nilai di kiri atas L ditambah COST = $(11+1) = 12$

L = Nilai yang paling kecil antara L_1 sampai L_3 yaitu $L_2 = 11$

COST = jika isi t dan s beda, contoh di atas $t = 13$ A dan $s = 1$ T, maka COST bernilai 1 (satu).

M_1 = Nilai di atas M ditambah 1 = $(13+1) = 14$

M_2 = Nilai di kiri M (L) ditambah 1 = $(11+1) = 12$

M_3 = Nilai di kiri atas M ditambah COST = $(12+1) = 13$

M = Nilai yang paling kecil antara M_1 sampai M_3 yaitu $M_2 = 12$

Tabel 3.3 Hasil perhitungan *string* awal dan *string* target untuk baris kedua

t/s		T	I	A	R	A	_	M	A	U	L	I	D	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
T	1	0	1	2	3	4	5	6	7	8	9	10	11	12

Tabel 3.3 Hasil perhitungan *string* awal dan *string* target untuk baris kedua

(Lanjutan)

t/s		T	I	A	R	A	_	M	A	U	L	I	D	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
I	2	1	0	1	2	3	4	5	6	7	8	9	10	11
A	3													
R	4													
A	5													
_	6													
M	7													
A	8													
U	9													
L	10													
I	11													
D	12													
A	13													
_	14													
A	15													
L	16													
I	17													

COST = jika isi t dan s beda, contoh di atas t = 0 T dan s = 2 I, maka COST bernilai satu (1).

$$A1 = \text{Nilai di atas A ditambah } 1 = (0+1) = 1$$

$$A2 = \text{Nilai di kiri A ditambah } 1 = (2+1) = 3$$

$$A3 = \text{Nilai di kiri atas A ditambah COST} = (1+1) = 2$$

$$A = \text{Nilai yang paling kecil antara A1 sampai A3 yaitu } A1 = 1$$

COST = jika isi t dan s sama, contoh di atas t = 1 I dan s = 2 I, maka COST bernilai nol (0).

$$B1 = \text{Nilai di atas B ditambah } 1 = (1+1) = 2$$

$$B2 = \text{Nilai di kiri B (A) ditambah } 1 = (1+1) = 2$$

$$B3 = \text{Nilai di kiri atas B ditambah COST} = (0+0) = 0$$

$$B = \text{Nilai yang paling kecil antara B1 sampai B3 yaitu } B3 = 0$$

COST = jika isi t dan s beda, contoh di atas t = 2 A dan s = 2 I, maka COST bernilai satu (1)

$$C1 = \text{Nilai di atas C ditambah } 1 = (2+1) = 3$$

$$C2 = \text{Nilai di kiri C (B) ditambah } 1 = (0+1) = 1$$

$$C3 = \text{Nilai di kiri atas C ditambah COST} = (1+1) = 2$$

$$C = \text{Nilai yang paling kecil antara C1 sampai C3 yaitu } C2 = 1$$

COST = jika isi t dan s beda, contoh di atas t = 3 R dan s = 2 I, maka COST bernilai satu (1)

$$D1 = \text{Nilai di atas D ditambah } 1 = (3+1) = 4$$

$$D2 = \text{Nilai di kiri D (C) ditambah } 1 = (1+1) = 2$$

$$D3 = \text{Nilai di kiri atas D ditambah COST} = (2+1) = 3$$

$$D = \text{Nilai yang paling kecil antara D1 sampai D3 yaitu } D2 = 2$$

COST = jika isi t dan s beda, contoh di atas t = 4 A dan s = 2 I, maka COST bernilai satu (1)

$$E1 = \text{Nilai di atas E ditambah } 1 = (4+1) = 5$$

$$E2 = \text{Nilai di kiri E (D) ditambah } 1 = (2+1) = 3$$

$$E3 = \text{Nilai di kiri atas E ditambah COST} = (3+1) = 4$$

$$E = \text{Nilai yang paling kecil antara E1 sampai E3 yaitu } E2 = 3$$

COST = jika isi t dan s beda, contoh di atas t = 5 _ dan s = 2 I, maka COST bernilai satu (1)

$$F1 = \text{Nilai di atas F ditambah } 1 = (5+1) = 6$$

$$F2 = \text{Nilai di kiri F (E) ditambah } 1 = (3+1) = 4$$

$$F3 = \text{Nilai di kiri atas F ditambah COST} = (4+1) = 5$$

$$F = \text{Nilai yang paling kecil antara F1 sampai F3 yaitu } F2 = 4$$

COST = jika isi t dan s beda, contoh di atas t = 6 M dan s = 2 I, maka COST bernilai satu (1)

$$G1 = \text{Nilai di atas G ditambah } 1 = (6+1) = 7$$

$$G2 = \text{Nilai di kiri G (F) ditambah } 1 = (4+1) = 5$$

$$G3 = \text{Nilai di kiri atas G ditambah COST} = (5+1) = 6$$

$$G = \text{Nilai yang paling kecil antara G1 sampai G3 yaitu } G2 = 5$$

COST = jika isi t dan s beda, contoh di atas t = 7 A dan s = 2 I, maka COST bernilai satu (1)

$$H1 = \text{Nilai di atas H ditambah } 1 = (7+1) = 8$$

$$H2 = \text{Nilai di kiri H (G) ditambah } 1 = (5+1) = 6$$

$$H3 = \text{Nilai di kiri atas H ditambah COST} = (6+1) = 7$$

$$H = \text{Nilai yang paling kecil antara H1 sampai H3 yaitu } H2 = 6$$

COST = jika isi t dan s beda, contoh di atas t = 8 U dan s = 2 I, maka COST bernilai satu (1)

$$I1 = \text{Nilai di atas I ditambah } 1 = (8+1) = 9$$

$$I2 = \text{Nilai di kiri I (H) ditambah } 1 = (6+1) = 7$$

$$I3 = \text{Nilai di kiri atas I ditambah COST} = (7+1) = 8$$

$$I = \text{Nilai yang paling kecil antara I1 sampai I3 yaitu } I2 = 7$$

COST = jika isi t dan s beda, contoh di atas $t = 9$ L dan $s = 2$ I, maka COST bernilai satu (1)

$$J1 = \text{Nilai di atas J ditambah } 1 = (9+1) = 10$$

$$J2 = \text{Nilai di kiri J (I) ditambah } 1 = (7+1) = 8$$

$$J3 = \text{Nilai di kiri atas J ditambah COST} = (8+1) = 9$$

$$J = \text{Nilai yang paling kecil antara } J1 \text{ sampai } J3 \text{ yaitu } J2 = 8$$

COST = jika isi t dan s sama, contoh di atas $t = 10$ I dan $s = 2$ I, maka COST bernilai nol (0)

$$K1 = \text{Nilai di atas K ditambah } 1 = (10+1) = 11$$

$$K2 = \text{Nilai di kiri K (J) ditambah } 1 = (8+1) = 9$$

$$K3 = \text{Nilai di kiri atas K ditambah COST} = (9+0) = 9$$

$$K = \text{Nilai yang paling kecil antara } K1 \text{ sampai } K3 \text{ yaitu } K3 = 9$$

COST = jika isi t dan s beda, contoh di atas $t = 11$ D dan $s = 2$ I, maka COST bernilai satu (1)

$$L1 = \text{Nilai di atas L ditambah } 1 = (11+1) = 12$$

$$L2 = \text{Nilai di kiri L (K) ditambah } 1 = (9+1) = 10$$

$$L3 = \text{Nilai di kiri atas L ditambah COST} = (10+1) = 11$$

$$L = \text{Nilai yang paling kecil antara } L1 \text{ sampai } L3 \text{ yaitu } L2 = 10$$

COST = jika isi t dan s beda, contoh di atas $t = 12$ A dan $s = 2$ I, maka COST bernilai satu (1)

$$M1 = \text{Nilai di atas M ditambah } 1 = (12+1) = 13$$

$$M2 = \text{Nilai di kiri M (L) ditambah } 1 = (10+1) = 11$$

$$M3 = \text{Nilai di kiri atas M ditambah COST} = (11+1) = 12$$

M = Nilai yang paling kecil antara $M1$ sampai $M3$ yaitu $L2 = 11$

Tabel 3.4 Hasil perhitungan *string* awal dan *string* target untuk baris ketiga

t/s		T	I	A	R	A	_	M	A	U	L	I	D	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
T	1	0	1	2	3	4	5	6	7	8	9	10	11	12
I	2	1	0	1	2	3	4	5	6	7	8	9	10	11
A	3	2	1	0	1	2	3	4	5	6	7	8	9	10
R	4													
A	5													
_	6													
M	7													
A	8													
U	9													
L	10													
I	11													
D	12													
A	13													
_	14													
A	15													
L	16													
I	17													

COST = jika isi t dan s beda, contoh di atas $t = 1$ T dan $s = 3$ A, maka COST bernilai satu (1)

$$A1 = \text{Nilai di atas A ditambah } 1 = (1+1) = 2$$

$$A2 = \text{Nilai di kiri A ditambah } 1 = (3+1) = 4$$

$$A3 = \text{Nilai di kiri atas A ditambah COST} = (2+1) = 3$$

$$A = \text{Nilai yang paling kecil antara } A1 \text{ sampai } A3 \text{ yaitu } A1 = 2$$

COST = jika isi t dan s beda, contoh di atas $t = 0$ I dan $s = 3$ A, maka COST bernilai satu (1)

$$B1 = \text{Nilai di atas B ditambah } 1 = (0+1) = 1$$

$$B2 = \text{Nilai di kiri B (A) ditambah } 1 = (2+1) = 3$$

$$B3 = \text{Nilai di kiri atas B ditambah COST} = (1+1) = 2$$

$$B = \text{Nilai yang paling kecil antara B1 sampai B3 yaitu } B1 = 1$$

COST = jika isi t dan s sama, contoh di atas $t = 1$ A dan $s = 3$ A, maka COST bernilai nol (0)

$$C1 = \text{Nilai di atas C ditambah } 1 = (1+1) = 2$$

$$C2 = \text{Nilai di kiri C (B) ditambah } 1 = (1+1) = 2$$

$$C3 = \text{Nilai di kiri atas C ditambah COST} = (0+0) = 0$$

$$C = \text{Nilai yang paling kecil antara C1 sampai C3 yaitu } C3 = 0$$

COST = jika isi t dan s beda, contoh di atas $t = 2$ R dan $s = 3$ A, maka COST bernilai satu (1)

$$D1 = \text{Nilai di atas D ditambah } 1 = (2+1) = 3$$

$$D2 = \text{Nilai di kiri D (C) ditambah } 1 = (0+1) = 1$$

$$D3 = \text{Nilai di kiri atas D ditambah COST} = (1+1) = 2$$

$$D = \text{Nilai yang paling kecil antara D1 sampai D3 yaitu } D2 = 1$$

COST = jika isi t dan s sama, contoh di atas $t = 3$ A dan $s = 3$ A, maka COST bernilai nol (0)

$$E1 = \text{Nilai di atas E ditambah } 1 = (3+1) = 4$$

$$E2 = \text{Nilai di kiri E (D) ditambah } 1 = (1+1) = 2$$

$$E3 = \text{Nilai di kiri atas E ditambah COST} = (2+0) = 2$$

$$E = \text{Nilai yang paling kecil antara E1 sampai E3 yaitu } E3 = 2$$

COST = jika isi t dan s beda, contoh di atas $t = 4$ dan $s = 3$ A, maka COST bernilai satu (1)

$$F1 = \text{Nilai di atas F ditambah } 1 = (4+1) = 5$$

$$F2 = \text{Nilai di kiri F (E) ditambah } 1 = (2+1) = 3$$

$$F3 = \text{Nilai di kiri atas F ditambah COST} = (3+1) = 4$$

$$F = \text{Nilai yang paling kecil antara F1 sampai F3 yaitu } F2 = 3$$

COST = jika isi t dan s beda, contoh di atas $t = 5$ M dan $s = 3$ A, maka COST bernilai satu (1)

$$G1 = \text{Nilai di atas G ditambah } 1 = (5+1) = 6$$

$$G2 = \text{Nilai di kiri G (F) ditambah } 1 = (3+1) = 4$$

$$G3 = \text{Nilai di kiri atas G ditambah COST} = (4+1) = 5$$

$$G = \text{Nilai yang paling kecil antara G1 sampai G3 yaitu } G2 = 4$$

COST = jika isi t dan s sama, contoh di atas $t = 6$ A dan $s = 3$ A, maka COST bernilai nol (0)

$$H1 = \text{Nilai di atas H ditambah } 1 = (6+1) = 7$$

$$H2 = \text{Nilai di kiri H (G) ditambah } 1 = (4+1) = 5$$

$$H3 = \text{Nilai di kiri atas H ditambah COST} = (5+1) = 6$$

$$H = \text{Nilai yang paling kecil antara H1 sampai H3 yaitu } H2 = 5$$

COST = jika isi t dan s beda, contoh di atas $t = 7$ U dan $s = 3$ A, maka COST bernilai satu (1)

$$I1 = \text{Nilai di atas I ditambah } 1 = (7+1) = 8$$

$$I2 = \text{Nilai di kiri I (H) ditambah } 1 = (5+1) = 6$$

$$I3 = \text{Nilai di kiri atas I ditambah COST} = (6+1) = 7$$

I = Nilai yang paling kecil antara I1 sampai I3 yaitu $I2 = 6$

COST = jika isi t dan s beda, contoh di atas $t = 8$ L dan $s = 3$ A, maka COST bernilai satu (1)

J1 = Nilai di atas J ditambah 1 = $(8+1) = 9$

J2 = Nilai di kiri J (I) ditambah 1 = $(6+1) = 7$

J3 = Nilai di kiri atas J ditambah COST = $(7+1) = 8$

J = Nilai yang paling kecil antara J1 sampai J3 yaitu $J2 = 7$

COST = jika isi t dan s beda, contoh di atas $t = 9$ I dan $s = 3$ A, maka COST bernilai satu (1)

K1 = Nilai di atas K ditambah 1 = $(9+1) = 10$

K2 = Nilai di kiri K (J) ditambah 1 = $(7+1) = 8$

K3 = Nilai di kiri atas K ditambah COST = $(8+1) = 9$

K = Nilai yang paling kecil antara K1 sampai K3 yaitu $K2 = 8$

COST = jika isi t dan s beda, contoh di atas $t = 10$ D dan $s = 3$ A, maka COST bernilai satu (1)

L1 = Nilai di atas L ditambah 1 = $(10+1) = 11$

L2 = Nilai di kiri L (K) ditambah 1 = $(8+1) = 9$

L3 = Nilai di kiri atas L ditambah COST = $(9+1) = 10$

L = Nilai yang paling kecil antara L1 sampai L3 yaitu $L2 = 9$

COST = jika isi t dan s sama, contoh di atas $t = 11$ A dan $s = 3$ A, maka COST bernilai nol (0)

M1 = Nilai di atas M ditambah 1 = $(11+1) = 12$

M2 = Nilai di kiri M (L) ditambah 1 = $(9+1) = 10$

$M3 = \text{Nilai di kiri atas M ditambah COST} = (10+0) = 10$

$M = \text{Nilai yang paling kecil antara M1 sampai M3 yaitu } M3 = 10$

Tabel 3.5 Hasil perhitungan *string* awal dan *string* target untuk baris keempat

t/s		T	I	A	R	A	_	M	A	U	L	I	D	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
T	1	0	1	2	3	4	5	6	7	8	9	10	11	12
I	2	1	0	1	2	3	4	5	6	7	8	9	10	11
A	3	2	1	0	1	2	3	4	5	6	7	8	9	10
R	4	3	2	1	0	1	2	3	4	5	6	7	8	9
A	5													
_	6													
M	7													
A	8													
U	9													
L	10													
I	11													
D	12													
A	13													
_	14													
A	15													
L	16													
I	17													

$\text{COST} = \text{jika isi t dan s beda, contoh di atas } t = 2 \text{ T dan } s = 4 \text{ R, maka COST}$

bernilai satu (1)

$A1 = \text{Nilai di atas A ditambah } 1 = (2+1) = 3$

$A2 = \text{Nilai di kiri A ditambah } 1 = (4+1) = 5$

$A3 = \text{Nilai di kiri atas A ditambah COST} = (3+1) = 4$

$A = \text{Nilai yang paling kecil antara A1 sampai A3 yaitu } A1 = 3$

COST = jika isi t dan s beda, contoh di atas $t = 1$ I dan $s = 4$ R, maka COST bernilai satu (1)

$$B1 = \text{Nilai di atas B ditambah } 1 = (1+1) = 2$$

$$B2 = \text{Nilai di kiri B (A) ditambah } 1 = (3+1) = 4$$

$$B3 = \text{Nilai di kiri atas B ditambah COST} = (2+1) = 3$$

$$B = \text{Nilai yang paling kecil antara B1 sampai B3 yaitu } B1 = 2$$

COST = jika isi t dan s beda, contoh di atas $t = 0$ A dan $s = 4$ R, maka COST bernilai satu (1)

$$C1 = \text{Nilai di atas C ditambah } 1 = (0+1) = 1$$

$$C2 = \text{Nilai di kiri C (B) ditambah } 1 = (2+1) = 3$$

$$C3 = \text{Nilai di kiri atas C ditambah COST} = (1+1) = 2$$

$$C = \text{Nilai yang paling kecil antara C1 sampai C3 yaitu } C1 = 1$$

COST = jika isi t dan s sama, contoh di atas $t = 1$ R dan $s = 4$ R, maka COST bernilai 0 (nol)

$$D1 = \text{Nilai di atas D ditambah } 1 = (1+1) = 2$$

$$D2 = \text{Nilai di kiri D (C) ditambah } 1 = (0+1) = 1$$

$$D3 = \text{Nilai di kiri atas D ditambah COST} = (0+0) = 0$$

$$D = \text{Nilai yang paling kecil antara D1 sampai D3 yaitu } D3 = 0$$

COST = jika isi t dan s beda, contoh di atas $t = 2$ A dan $s = 4$ R, maka COST bernilai satu (1)

$$E1 = \text{Nilai di atas E ditambah } 1 = (2+1) = 3$$

$$E2 = \text{Nilai di kiri E (D) ditambah } 1 = (0+1) = 1$$

$$E3 = \text{Nilai di kiri atas E ditambah COST} = (1+1) = 2$$

E = Nilai yang paling kecil antara $E1$ sampai $E3$ yaitu $E2 = 1$

$COST$ = jika isi t dan s beda, contoh di atas $t = 3$ dan $s = 4$ R, maka $COST$ bernilai satu (1)

$F1$ = Nilai di atas F ditambah 1 = $(3+1) = 4$

$F2$ = Nilai di kiri F (E) ditambah 1 = $(1+1) = 2$

$F3$ = Nilai di kiri atas F ditambah $COST = (2+1) = 3$

F = Nilai yang paling kecil antara $F1$ sampai $F3$ yaitu $F2 = 2$

$COST$ = jika isi t dan s beda, contoh di atas $t = 4$ M dan $s = 4$ R, maka $COST$ bernilai satu (1)

$G1$ = Nilai di atas G ditambah 1 = $(4+1) = 5$

$G2$ = Nilai di kiri G (F) ditambah 1 = $(2+1) = 3$

$G3$ = Nilai di kiri atas G ditambah $COST = (3+1) = 4$

G = Nilai yang paling kecil antara $G1$ sampai $G3$ yaitu $G2 = 3$

$COST$ = jika isi t dan s beda, contoh di atas $t = 5$ A dan $s = 4$ R, maka $COST$ bernilai satu (1)

$H1$ = Nilai di atas H ditambah 1 = $(5+1) = 6$

$H2$ = Nilai di kiri H (G) ditambah 1 = $(3+1) = 4$

$H3$ = Nilai di kiri atas H ditambah $COST = (4+1) = 5$

H = Nilai yang paling kecil antara $H1$ sampai $H3$ yaitu $H2 = 4$

$COST$ = jika isi t dan s beda, contoh di atas $t = 6$ U dan $s = 4$ R, maka $COST$ bernilai satu (1)

$I1$ = Nilai di atas I ditambah 1 = $(6+1) = 7$

$I2$ = Nilai di kiri I (H) ditambah 1 = $(4+1) = 5$

$$I3 = \text{Nilai di kiri atas I ditambah COST} = (5+1) = 6$$

$$I = \text{Nilai yang paling kecil antara I1 sampai I3 yaitu } I2 = 5$$

COST = jika isi t dan s beda, contoh di atas t = 7 L dan s = 4 R, maka COST bernilai satu (1)

$$J1 = \text{Nilai di atas J ditambah 1} = (7+1) = 8$$

$$J2 = \text{Nilai di kiri J (I) ditambah 1} = (5+1) = 6$$

$$J3 = \text{Nilai di kiri atas J ditambah COST} = (6+1) = 7$$

$$J = \text{Nilai yang paling kecil antara J1 sampai J3 yaitu } J2 = 6$$

COST = jika isi t dan s beda, contoh di atas t = 8 I dan s = 4 R, maka COST bernilai satu (1)

$$K1 = \text{Nilai di atas K ditambah 1} = (8+1) = 9$$

$$K2 = \text{Nilai di kiri K (J) ditambah 1} = (6+1) = 7$$

$$K3 = \text{Nilai di kiri atas K ditambah COST} = (7+1) = 8$$

$$K = \text{Nilai yang paling kecil antara K1 sampai K3 yaitu } K2 = 7$$

COST = jika isi t dan s beda, contoh di atas t = 9 D dan s = 4 R, maka COST bernilai satu (1)

$$L1 = \text{Nilai di atas L ditambah 1} = (9+1) = 10$$

$$L2 = \text{Nilai di kiri L (K) ditambah 1} = (7+1) = 8$$

$$L3 = \text{Nilai di kiri atas L ditambah COST} = (8+1) = 9$$

$$L = \text{Nilai yang paling kecil antara L1 sampai L3 yaitu } L2 = 8$$

COST = jika isi t dan s beda, contoh di atas t = 10 A dan s = 4 R, maka COST bernilai satu (1)

$$M1 = \text{Nilai di atas M ditambah 1} = (10+1) = 11$$

$M = \text{Nilai di kiri M (L) ditambah } 1 = (8+1) = 9$

$M = \text{Nilai di kiri atas M ditambah COST} = (9+1) = 10$

$M = \text{Nilai yang paling kecil antara M1 sampai M3 yaitu } M2 = 9$

3) Proses terus berjalan sampai akhir karakter s dan karakter t.

Tabel 3.6 Hasil perhitungan *string* awal dan *string* target untuk baris terakhir

t/s		T	I	A	R	A	_	M	A	U	L	I	D	A
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
T	1	0	1	2	3	4	5	6	7	8	9	10	11	12
I	2	1	0	1	2	3	4	5	6	7	8	9	10	11
A	3	2	1	0	1	2	3	4	5	6	7	8	9	10
R	4	3	2	1	0	1	2	3	4	5	6	7	8	9
A	5	4	3	2	1	0	1	2	3	4	5	6	7	8
_	6	5	4	3	2	1	0	1	2	3	4	5	6	7
M	7	6	5	4	3	2	1	0	1	2	3	4	5	6
A	8	7	6	5	4	3	2	1	0	1	2	3	4	5
U	9	8	7	6	5	4	3	2	1	0	1	2	3	4
L	10	9	8	7	6	5	4	3	2	1	0	1	2	3
I	11	10	9	8	7	6	5	4	3	2	1	0	1	2
D	12	11	10	9	8	7	6	5	4	3	2	1	0	1
A	13	12	11	10	9	8	7	6	5	4	3	2	1	0
_	14	13	12	11	10	9	8	7	6	5	4	3	2	1
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2
L	16	15	14	13	12	11	10	9	8	7	6	5	4	3
I	17	16	15	14	13	12	11	10	9	8	7	6	5	4

COST = jika isi t dan s beda, contoh di atas t = 15 T dan s = 17 I, maka COST

bernilai satu (1)

$A1 = \text{Nilai di atas A ditambah } 1 = (15+1) = 16$

$A2 = \text{Nilai di kiri A ditambah } 1 = (17+1) = 18$

$A3 = \text{Nilai di kiri atas A ditambah COST} = (16+1) = 17$

A = Nilai yang paling kecil antara A1 sampai A3 yaitu $A2 = 16$

COST = jika isi t dan s sama, contoh di atas $t = 14$ I dan $s = 17$ I, maka COST bernilai nol (0)

B1 = Nilai di atas B ditambah 1 = $(14+1) = 15$

B2 = Nilai di kiri B (A) ditambah 1 = $(16+1) = 17$

B3 = Nilai di kiri atas B ditambah COST = $(15+0) = 15$

B = Nilai yang paling kecil antara B1 sampai B3 yaitu $B1 = 15$

COST = jika isi t dan s beda, contoh di atas $t = 13$ A dan $s = 17$ I, maka COST bernilai satu (1)

C1 = Nilai di atas C ditambah 1 = $(13+1) = 14$

C2 = Nilai di kiri C (B) ditambah 1 = $(15+1) = 16$

C3 = Nilai di kiri atas C ditambah COST = $(14+1) = 15$

C = Nilai yang paling kecil antara C1 sampai C3 yaitu $C1 = 14$

COST = jika isi t dan s beda, contoh di atas $t = 12$ R dan $s = 17$ I, maka COST bernilai satu (1)

D1 = Nilai di atas D ditambah 1 = $(12+1) = 13$

D2 = Nilai di kiri D (C) ditambah 1 = $(14+1) = 15$

D3 = Nilai di kiri atas D ditambah COST = $(13+1) = 14$

D = Nilai yang paling kecil antara D1 sampai D3 yaitu $D3 = 13$

COST = jika isi t dan s beda, contoh di atas $t = 11$ A dan $s = 17$ I, maka COST bernilai satu (1)

E1 = Nilai di atas E ditambah 1 = $(11+1) = 12$

E2 = Nilai di kiri E (D) ditambah 1 = $(13+1) = 14$

$$E3 = \text{Nilai di kiri atas E ditambah COST} = (12+1) = 13$$

$$E = \text{Nilai yang paling kecil antara E1 sampai E3 yaitu } E1 = 12$$

COST = jika isi t dan s beda, contoh di atas t = 10 _ dan s = 17 I, maka COST bernilai satu (1)

$$F1 = \text{Nilai di atas F ditambah 1} = (10+1) = 11$$

$$F2 = \text{Nilai di kiri F (E) ditambah 1} = (12+1) = 13$$

$$F3 = \text{Nilai di kiri atas F ditambah COST} = (11+1) = 12$$

$$F = \text{Nilai yang paling kecil antara F1 sampai F3 yaitu } F1 = 11$$

COST = jika isi t dan s beda, contoh di atas t = 9 M dan s = 17 I, maka COST bernilai satu (1)

$$G1 = \text{Nilai di atas G ditambah 1} = (9+1) = 10$$

$$G2 = \text{Nilai di kiri G (F) ditambah 1} = (11+1) = 12$$

$$G3 = \text{Nilai di kiri atas G ditambah COST} = (10+1) = 11$$

$$G = \text{Nilai yang paling kecil antara G1 sampai G3 yaitu } G1 = 10$$

COST = jika isi t dan s beda, contoh di atas t = 8 A dan s = 17 I, maka COST bernilai satu (1)

$$H1 = \text{Nilai di atas H ditambah 1} = (8+1) = 9$$

$$H2 = \text{Nilai di kiri H (G) ditambah 1} = (10+1) = 11$$

$$H3 = \text{Nilai di kiri atas H ditambah COST} = (9+1) = 10$$

$$H = \text{Nilai yang paling kecil antara H1 sampai H3 yaitu } H1 = 9$$

COST = jika isi t dan s beda, contoh di atas t = 7 U dan s = 17 I, maka COST bernilai satu (1)

$$I1 = \text{Nilai di atas I ditambah 1} = (7+1) = 8$$

$$I2 = \text{Nilai di kiri I (H) ditambah 1} = (9+1) = 10$$

$$I3 = \text{Nilai di kiri atas I ditambah COST} = (8+1) = 9$$

$$I = \text{Nilai yang paling kecil antara I1 sampai I3 yaitu I1} = 8$$

COST = jika isi t dan s beda, contoh di atas t = 6 L dan s = 17 I, maka COST bernilai satu (1)

$$J1 = \text{Nilai di atas J ditambah 1} = (6+1) = 7$$

$$J2 = \text{Nilai di kiri J (I) ditambah 1} = (8+1) = 9$$

$$J3 = \text{Nilai di kiri atas J ditambah COST} = (7+1) = 8$$

$$J = \text{Nilai yang paling kecil antara J1 sampai J3 yaitu J3} = 7$$

COST = jika isi t dan s sama, contoh di atas t = 5 I dan s = 17 I, maka COST bernilai nol (0)

$$K1 = \text{Nilai di atas K ditambah 1} = (5+1) = 6$$

$$K2 = \text{Nilai di kiri K (J) ditambah 1} = (7+1) = 8$$

$$K3 = \text{Nilai di kiri atas K ditambah COST} = (6+0) = 6$$

$$K = \text{Nilai yang paling kecil antara K1 sampai K3 yaitu K3} = 6$$

COST = jika isi t dan s beda, contoh di atas t = 4 D dan s = 17 I, maka COST bernilai satu (1)

$$L1 = \text{Nilai di atas L ditambah 1} = (4+1) = 5$$

$$L2 = \text{Nilai di kiri L (K) ditambah 1} = (6+1) = 7$$

$$L3 = \text{Nilai di kiri atas L ditambah COST} = (5+1) = 6$$

$$L = \text{Nilai yang paling kecil antara L1 sampai L3 yaitu L1} = 5$$

COST = jika isi t dan s beda, contoh di atas t = 3 A dan s = 17 I, maka COST bernilai satu (1)

$M1 = \text{Nilai di atas } M \text{ ditambah } 1 = (3+1) = 4$

$M = \text{Nilai di kiri } M \text{ (L) ditambah } 1 = (5+1) = 6$

$M = \text{Nilai di kiri atas } M \text{ ditambah } \text{COST} = (4+1) = 5$

$M = \text{Nilai yang paling kecil antara } M1 \text{ sampai } M3 \text{ yaitu } M1 = 4$

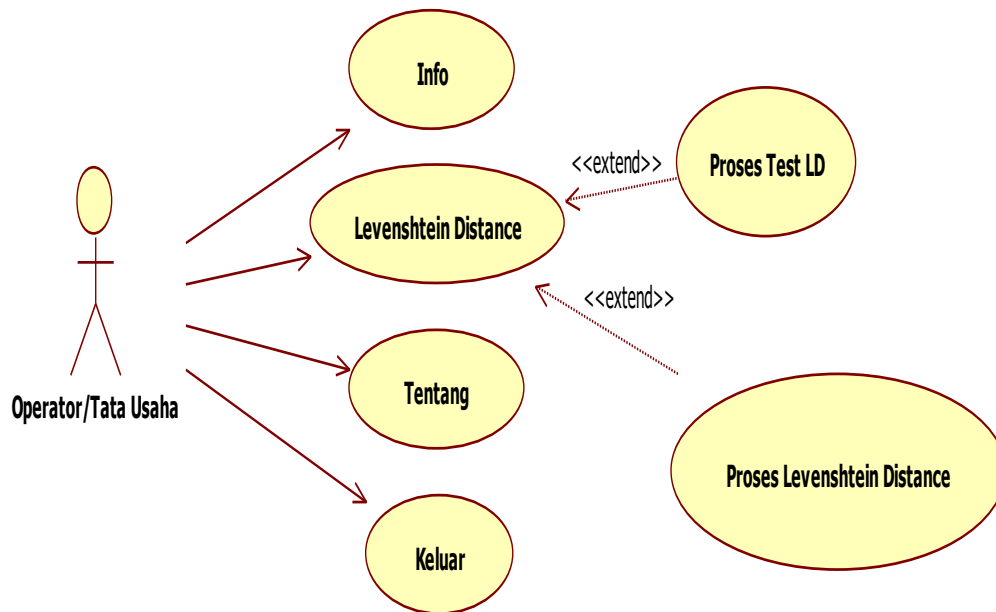
Maka didapatkan jarak *string* awal (s) dengan *string* target (t) adalah empat. Nilai *edit distance*-nya yaitu pada ujung kanan bawah matriks.

3.6 Perancangan Sistem

Perancangan aplikasi ini dirancang menggunakan *UML (Unified Modelling Language)* agar mempermudah memindahkan konsep yang dirancang ke dalam bentuk program. Perancangannya digambarkan dalam bentuk diagram-diagram berikut :

3.6.1 Use Case Diagram

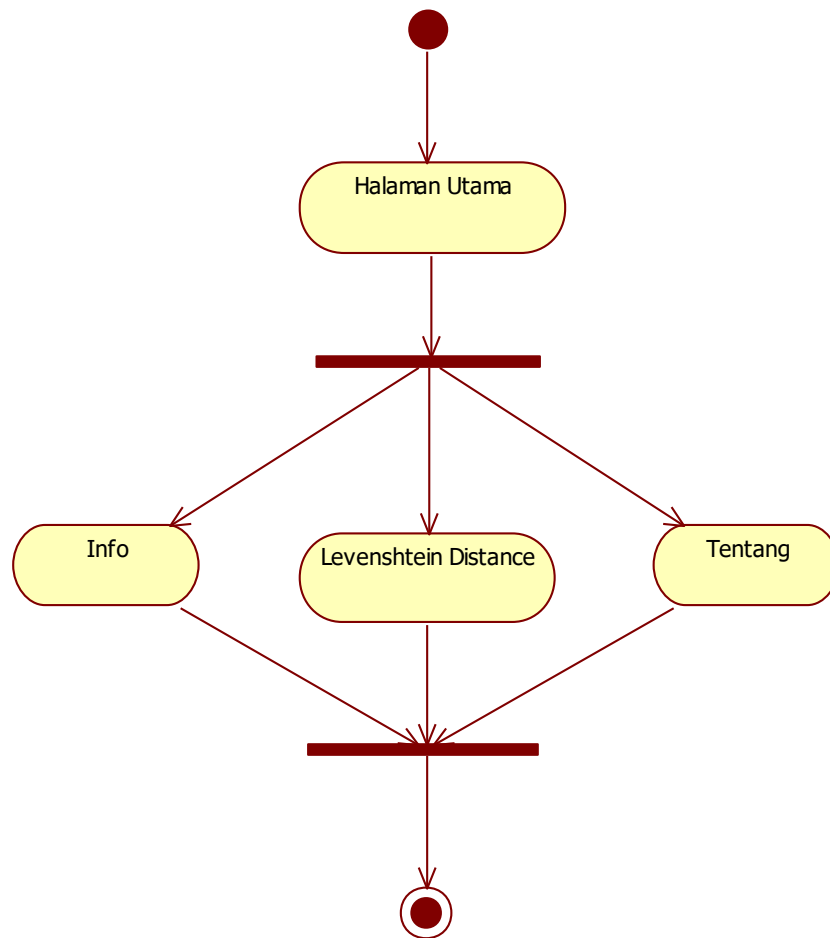
Use case adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Tujuan *Use Case* adalah dapat memetakan kebutuhan sistem, merepresentasikan interaksi pengguna terhadap sistem dan untuk mengetahui kebutuhan diluar sistem.



Gambar 3.1 Use Case Diagram

3.6.2 Activity Diagram

Activity Diagram atau Diagram aktivitas adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, atau pengulangan. Diagram aktivitas memiliki komponen dengan bentuk tertentu, dihubungkan dengan tanda panah. Panah tersebut mengarahkan urutan aktivitas yang terjadi, dari awal sampai akhir. Yang perlu diperhatikan yaitu diagram aktivitas bukan menggambarkan aktivitas sistem yang dilakukan aktor, tetapi menggambarkan aktivitas yang dapat dilakukan oleh sistem. Tujuan utama dari *activity* diagram adalah menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses agar lebih mudah dipahami. *Activity* diagram juga digunakan untuk menunjukkan aliran pesan dari satu aktivitas ke aktivitas lainnya.



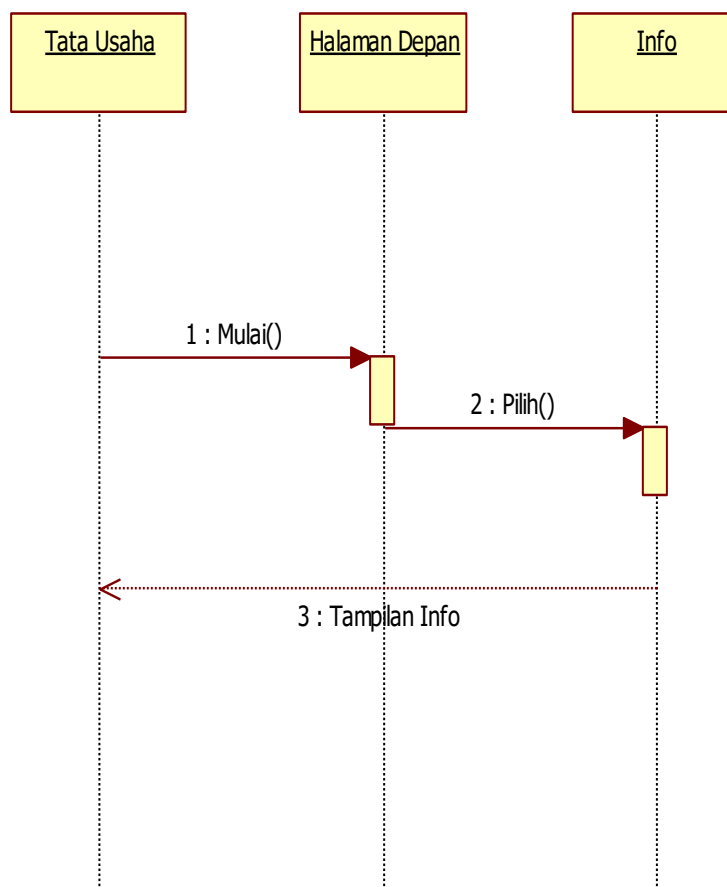
Gambar 3.2 Activity Diagram

3.6.3 Sequence Diagram

Sequence diagram/diagram sekuen menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use*

case yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.

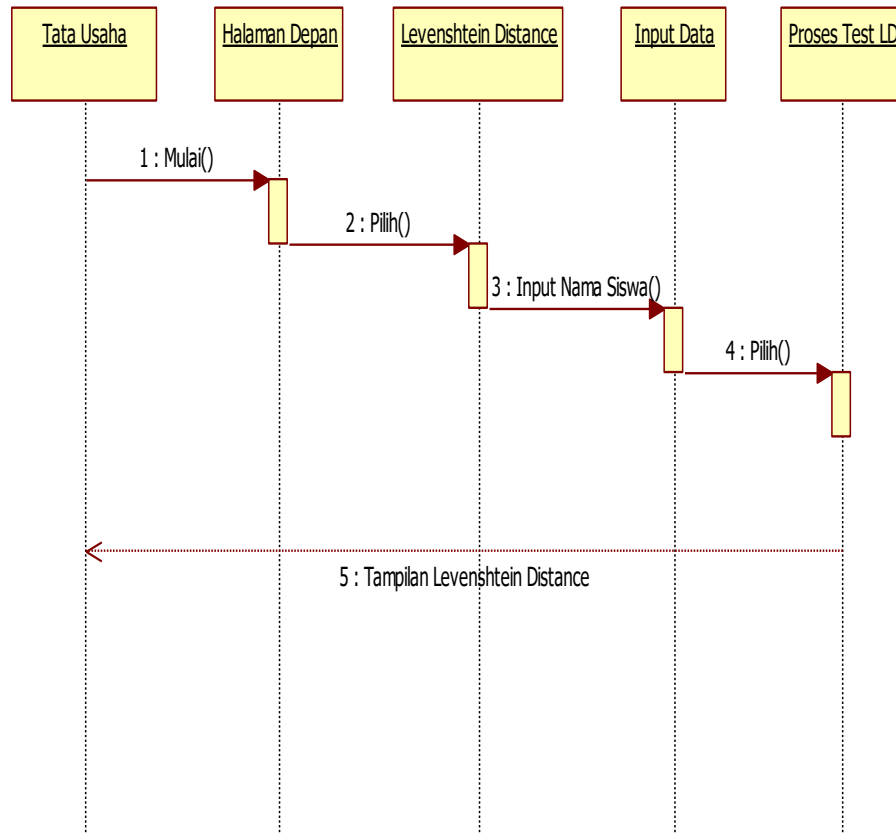
1. *Sequence* Diagram pada menu Info



Gambar 3.3 *Sequence* Diagram pada menu Info

Pada gambar diatas Tata Usaha memulai sistem *Levenshtein Distance* dengan membuka halaman depan yang tampilannya berupa judul skripsi, tombol Info, tombol *Levenshtein Distance*, tombol tentang data penulis, dan tombol keluar. Tombol Info berisi tentang deskripsi dari pengertian dan cara metode *Levenshtein Distance*.

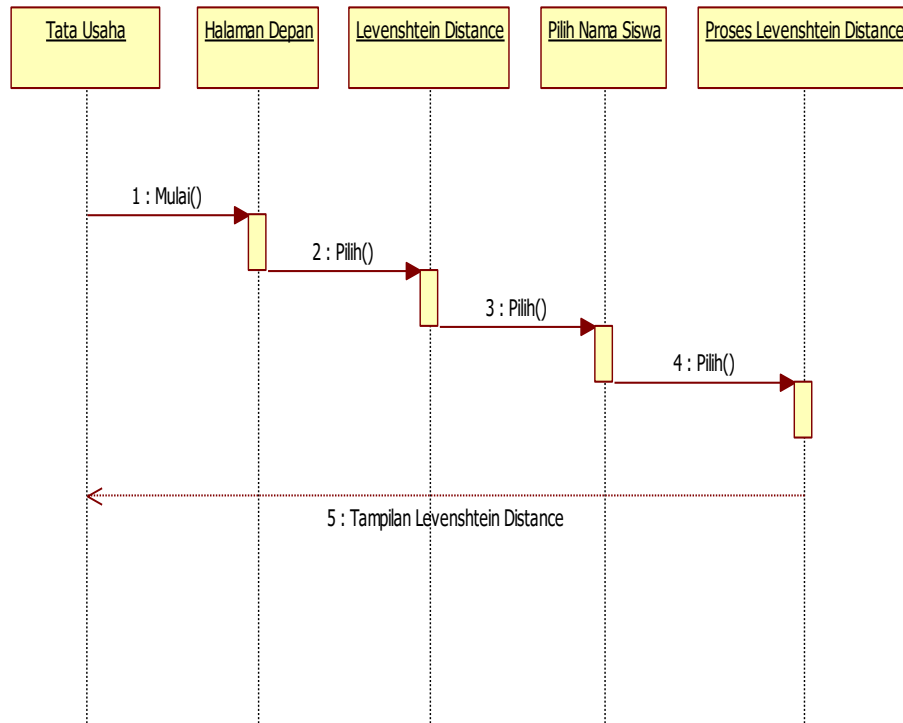
2. Sequence Diagram pada menu *Levenshtein Distance* Input Data



Gambar 3.4 Sequence Diagram pada menu *Levenshtein Distance* Input Data

Pada gambar diatas Tata Usaha memulai sistem *Levenshtein Distance* dengan membuka halaman depan yang tampilannya berupa judul skripsi, tombol Info, tombol *Levenshtein Distance*, tombol tentang data penulis, dan tombol keluar. Tombol *Levenshtein Distance* berisi tentang menginput data yang akan diuji pada kotak input dengan menuliskan nama siswa yang salah, kemudian pada kotak data dapat ditulis dengan menuliskan nama siswa yang benar dengan acuan Ijazah asli SD dari siswa yang akan diuji namanya. Kemudian kotak *LevenshteinDistance* akan menampilkan jarak terdekat nama yang telah diuji dengan menekan tombol *Test LD*.

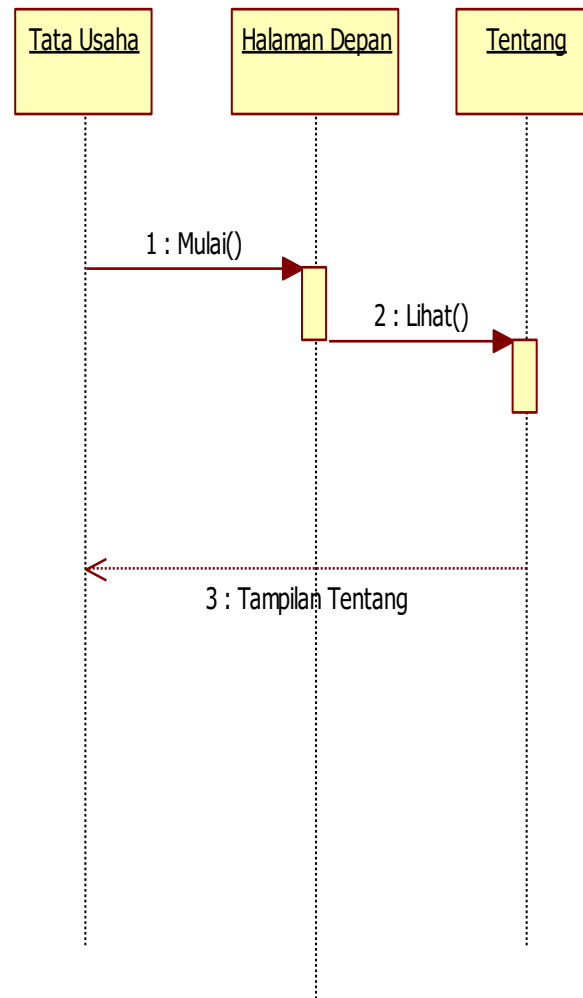
3. Sequence Diagram pada menu *Levenshtein Distance* Nama Siswa



Gambar 3.5 Sequence Diagram pada menu *Levenshtein Distance* nama siswa

Pada gambar diatas Tata Usaha memulai sistem *Levenshtein Distance* dengan membuka halaman depan yang tampilannya berupa judul skripsi, tombol Info, tombol *Levenshtein Distance*, tombol tentang data penulis, dan tombol keluar. Pada kolom Dapodik dan Ijazah sudah ada tampilan nama-nama siswa yang akan diuji pada masing-masing kolom, sebelumnya kita tuliskan dahulu nama siswa tersebut di pengkodean pada *formvb*, lalu untuk melihat hasilnya kita klik tombol *Levenshtein Distancenya*. Maka dapat dilihat hasil nama yang diuji dengan melihat jarak yang paling mendekati dengan nama yang ada di Ijazah tersebut. Dengan itu dapat ditarik kesimpulan bahwasanya jarak yang paling kecil itulah yang mendekati penulisan yang benar.

4. *Sequence* Diagram pada menu Tentang

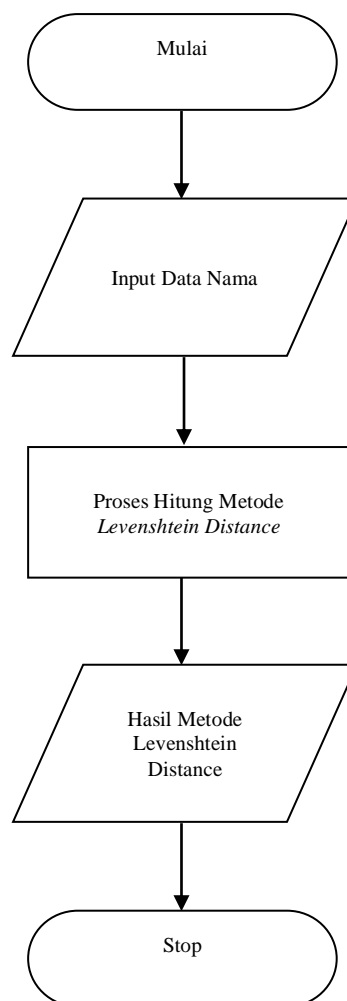


Gambar 3.6 *Sequence* Diagram pada menu Tentang

Pada gambar diatas Tata Usaha memulai sistem *Levenshtein Distance* dengan membuka halaman depan yang tampilannya berupa judul skripsi, tombol Info, tombol *Levenshtein Distance*, tombol tentang data penulis, dan tombol keluar. Tombol Tentang berisi tentang deskripsi dari penulis berupa nama, NPM dan Fakultas.

3.7 *Flowchart* dari Sistem Pendeteksi Penyesuaian Data Menggunakan *Levenshtein Distance*.

Ditahapan ini menggambarkan suatu tahapan penyelesaian terhadap suatu masalah secara sederhana, terurai rapi dan jelas menggunakan simbol-simbol standar yang sering digunakan untuk menyelesaikan *flowchart*.



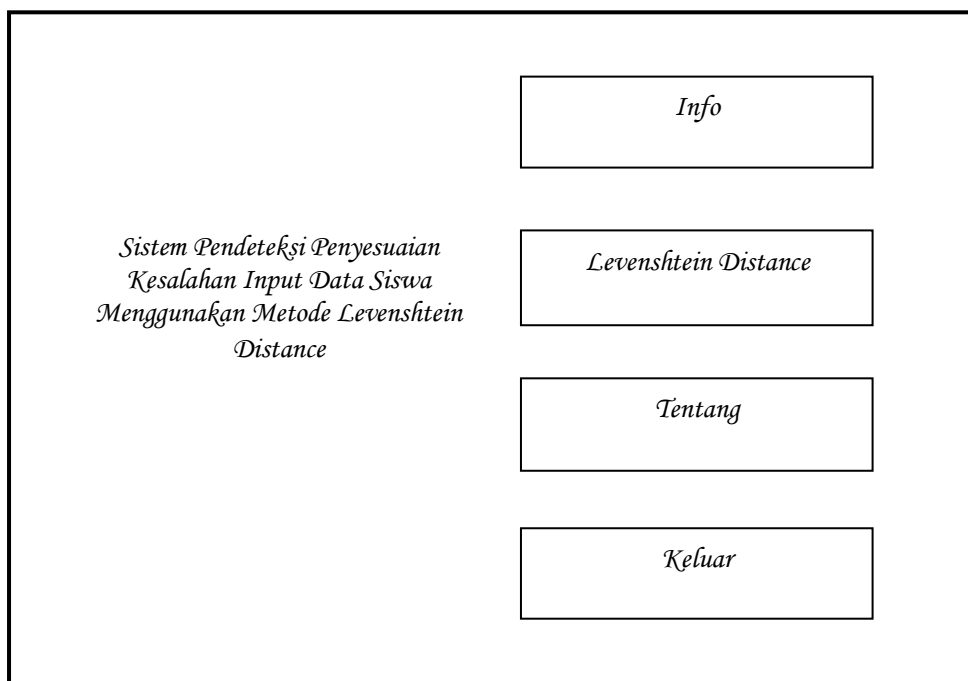
Gambar 3.7 *Flowchart* dari Sistem Pendeteksi Penyesuaian Data Menggunakan *Levenshtein Distance*

3.8 Perancangan Antarmuka

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap form yang akan digunakan pada tampilan sistem informasi sebenarnya.

3.8.1 Menu Utama

Ada empat menu yang ditampilkan pada halaman menu utama yaitu menu Info, menu *Levenshtein Distance*, menu Tentang, dan menu keluar.



Gambar 3.8 Tampilan Menu Utama

3.8.2 Menu Info

Tampilan pada menu info hanya menjelaskan keterangan pengertian dari *Levenshtein Distance*.

Algoritma Levenshtein Distance / Edit Distance adalah salah satu algoritma yang digunakan untuk pengambilan keputusan. Contoh yang dibahas kali ini adalah mengenai penentuan kata-kata usulan yang paling mendekati kata awal.

Fitur usulan dan juga auto koreksi biasanya sering terlihat pada mesin-mesin pencarian di internet. Sewaktu pengguna internet ingin mengetikkan sebuah kata misalnya "pegawai", pada saat pengetikkan masih pada saat "pegaw", mesin pencarian sudah memberikan usulan "pegawai" sebagai kata jawaban.

Juga pada saat ternyata pengguna internet salah menulis "pefawai", mesin pencarian juga melakukan auto koreksi, sehingga kata tersebut tetap dapat memberikan pencarian mengenai "pegawai". Kedua fitur tersebut dapat dihitung dengan algoritma ini.

Gambar 3.9 Tampilan Menu Info

3.8.3 Menu *Levenshtein Distance*

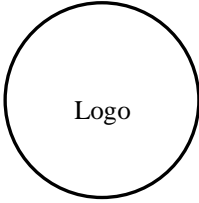
Tampilan menu *Levenshtein Distance* menjelaskan keterangan menu *Input* yang harus kita ketik data nama siswa yang salah, untuk keterangan menu *Data* dapat kita ketik nama siswa yang sudah sesuai dengan Ijazah, lalu kita dapat mengklik tombol *Test LD* maka hasil perhitungannya akan terlihat dikotak hasil, dengan perhitungan yang diuraikan satu persatu hingga mendapatkan jarak yang paling dekat nilainya, sedangkan untuk tombol *Levenshtein Distance* sendiri merupakan hasil dari tabel Dapodik dan tabel Ijazah yang berada di sebelah kanan dari Menu *Levenshtein Distance*, tetapi cara kerjanya tidak sama seperti kita mengklik tombol *Test LD*, hasil yang ditampilkan berupa nama yang diuji yang sudah ada ditabel Dapodik dan tabel Ijazah dan hasil dari pengujian tersebut sama seperti hasil dari pengujian dari Tombol *Test LD*.

Input	<table border="1"> <thead> <tr> <th></th> <th>Dapodik</th> <th>Ijazah</th> </tr> </thead> <tbody> <tr> <td>▶</td> <td>.....</td> <td>.....</td> </tr> <tr> <td></td> <td>.....</td> <td>.....</td> </tr> <tr> <td></td> <td>.....</td> <td>.....</td> </tr> <tr> <td></td> <td>.....</td> <td>.....</td> </tr> <tr> <td></td> <td>.....</td> <td>.....</td> </tr> <tr> <td></td> <td>.....</td> <td>.....</td> </tr> <tr> <td></td> <td>.....</td> <td>.....</td> </tr> </tbody> </table>		Dapodik	Ijazah	▶
	Dapodik		Ijazah																							
▶																							
																							
																							
																							
																							
																							
																								
Data																									
<i>Levenshtein</i>																									
Test LD																										
<i>Levenshtein Distance</i>																										
<p>Hasil dari <i>Levenshtein Distance</i></p>																										

Gambar 3.10 Tampilan Menu *Levenshtein Distance*

3.8.4 Menu Tentang

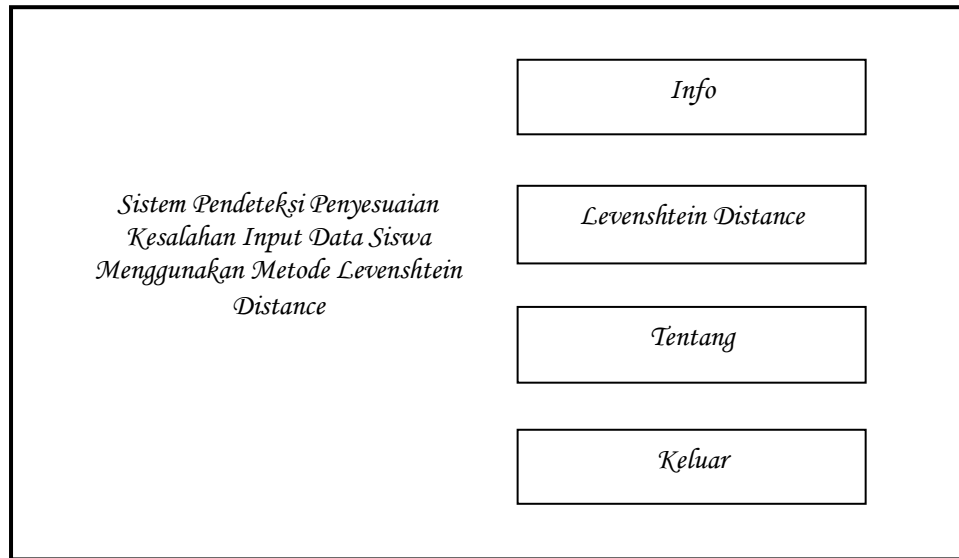
Tampilan menu tentang menjelaskan sebuah logo Universitas Pembangunan Panca Budi dan keterangan dari mahasiswa tersebut.

 <p>Logo</p>
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p>Keterangan</p> </div>

Gambar 3.11 Tampilan Menu Tentang

3.8.5 Menu Keluar

Tampilan keluar hampir sama tampilannya yang hanya menjelaskan tentang keluar dari program tersebut.



Gambar 3.12 Tampilan Menu Keluar

BAB IV

HASIL DAN PEMBAHASAN

4.1 Deskripsi Hasil Penelitian

Mendeskripsikan data hasil penelitian merupakan langkah yang tidak bisa dipisahkan dengan kegiatan analisis data sebagai prasyarat untuk memasuki tahap pembahasan dan pengambilan kesimpulan hasil penelitian. Sampel yang ditetapkan sebanyak 32 siswa SMP Negeri 2 Babalan yang telah dicek datanya berupa nama siswa tersebut dengan cara pengumpulan Ijazah SD. Sebelum melakukan penyesuaian data nama siswa, peneliti mengunduh nama-nama siswa tersebut dilaman Peserta Didik Ujian Nasional (PDUN) yang data tersebut diambil dari Dapodik, lalu peneliti menyesuaikan data tersebut berupa nama-nama siswa dengan Ijazah yang sudah dikumpulkan. Kemudian dari seluruh data yang diperoleh, masing-masing akan dicari *string* awal dan *string* targetnya dengan mencari hasil yang terkecil nilainya.

4.2 Komponen Utama dalam Implementasi Sistem

Agar sistem perancangan yang telah dikerjakan dapat berjalan dengan baik, maka perlu kiranya dilakukan pengujian terhadap sistem yang telah dikerjakan. Oleh karena itu, dibutuhkan beberapa komponen untuk mencakup perangkat keras (*Hardware*), perangkat lunak (*Software*), dan perangkat manusia (*Brainware*).

1. Perangkat Keras (*Hardware*)

Hardware merupakan komponen yang terlihat secara fisik yang saling bekerja sama dalam pengolahan data. Perangkat keras yang digunakan meliputi :

- a. Monitor atau Laptop
- b. CPU (*Central Processing Unit*)
- c. *Harddisk* sebagai tempat sistem beroperasi dalam media penyimpanan
- d. *Memory* minimal 2 GB

2. Perangkat Lunak (*Software*)

Software merupakan program-program komputer yang dapat digunakan oleh komputer dengan memberikan fungsi serta penampilan yang diinginkan. Perangkat lunak yang digunakan adalah :

- a. *Operating System Windows 7 Professional 32-bit*
- b. Bahasa Pemrograman yaitu *Microsoft Visual Basic 10.0*

3. Unsur Manusia (*Brainware*)

Istilah *brainware* komputer ditujukan untuk orang yang menggunakan atau mengoperasikan komputer. Secanggih-canggihnya sebuah komputer baik dari segi *hardware* atau *software* jika tidak ada *brainware* yang mengoperasikannya, maka fungsi *hardware* dan *software* komputer tersebut tidak akan bisa berjalan secara optimal. Berdasarkan tingkat pengoperasian komputer, *brainware* bisa digolongkan menjadi :

- a. Analisa adalah orang yang bertanggung jawab dalam merencanakan, menentukan serta memberi rekomendasi sistem atau software apa yang cocok untuk kebutuhan bisnis perusahaan.
- b. *Programmer* orang yang menguasai berbagai macam bahasa pemrograman sehingga ia mampu menciptakan sebuah program komputer.
- c. *Administrator* adalah orang yang bertanggung jawab mengatur serta mengendalikan program komputer dan juga jaringan komputer.
- d. Operator adalah orang yang mengoperasikan komputer atau biasa disebut juga sebagai *user*. Operator hanya bisa menjalankan aplikasi komputer yang diperbolehkan oleh administrator.
- e. Teknisi adalah orang yang bertanggung jawab memelihara komponen *hardware* serta memperbaiki jika ada kerusakan *hardware* dalam sistem komputer.

4.3 Implementasi Antar Muka

Subbab ini akan menunjukkan tampilan program dan desain program *Levenshtein Distance* dari hasil perancangan yang telah dibangun pada bab sebelumnya.

4.3.1 Halaman Menu Utama

Form dibawah ini adalah *form* dari tampilan halaman depan menu utama yang berisikan judul dari skripsi, kemudian beberapa tombol menu untuk

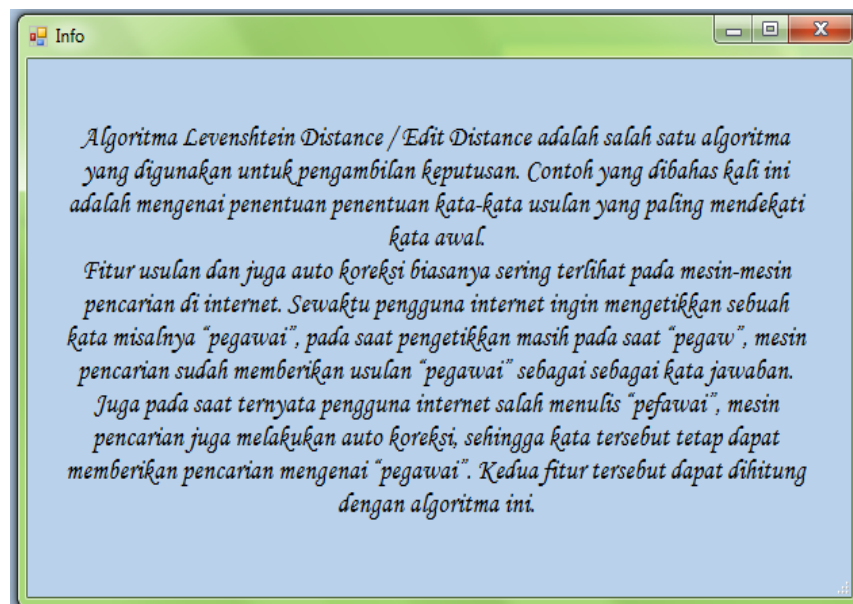
menjalankan program tersebut berupa tombol menu info, tombol menu *Levenshtein Distance*, tombol menu tentang, dan tombol menu keluar.



Gambar 4.1 Tampilan Halaman Menu Utama

4.3.2 Halaman Menu Info

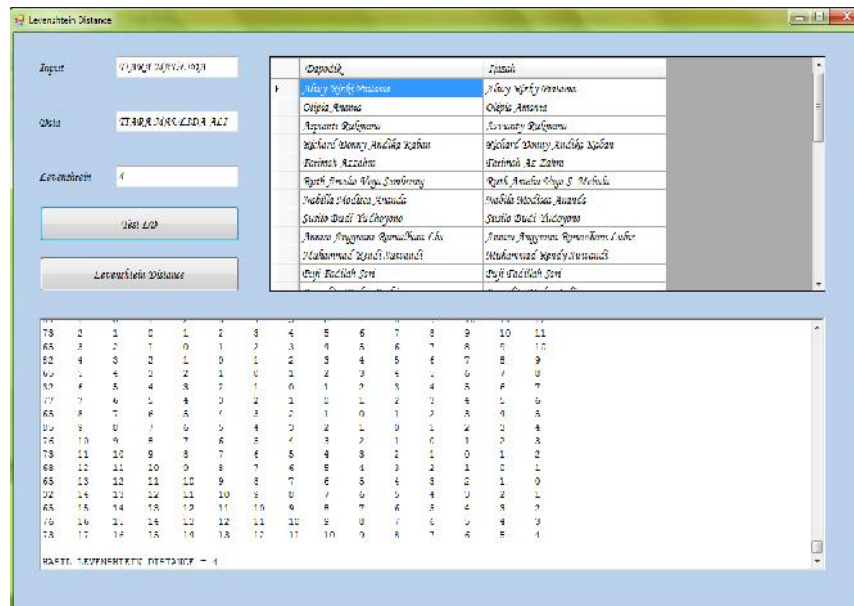
Form dibawah ini adalah *form* dari tampilan halaman menu info yang berisikan tentang pengertian dari Algoritma *Levenshtein Distance*.



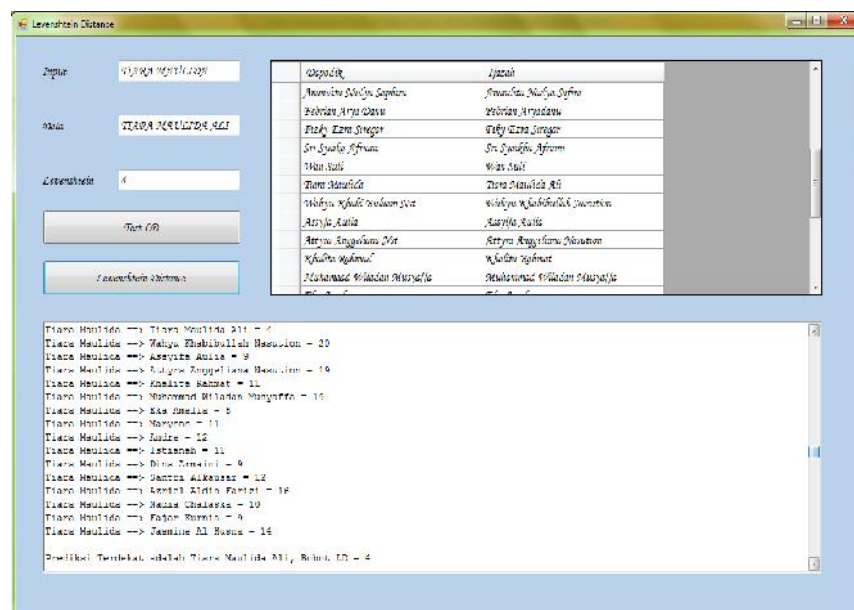
Gambar 4.2 Tampilan Halaman Menu Info

4.3.3 Halaman Menu *Levenshtein Distance*

Form dibawah ini adalah form dari tampilan halaman menu *Levenshtein Distance* berisikan tentang perhitungan dari sistem tersebut serta hasil dari pengujian nama siswa yang di *input* serta data awalnya.



Gambar 4.3 Tampilan Halaman Menu *Levenshtein Distance* pada Tombol *Test LD*



Gambar 4.4 Tampilan Halaman Menu *Levenshtein Distance* pada Tombol *Levenshtein Distance*

4.3.4 Halaman Menu Tentang

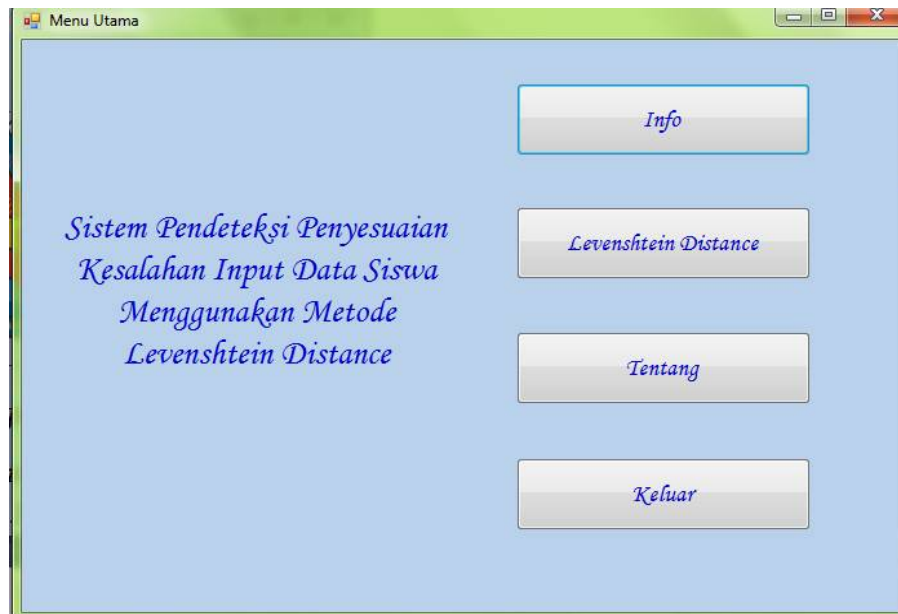
Form dibawah ini adalah *form* dari tampilan halaman menu tentang yang berisikan tentang gambar logo dan nama Universitas serta keterangan dari penulis.



Gambar 4.5 Tampilan Halaman Menu Tentang

4.3.5 Halaman Menu Keluar

Form dibawah ini adalah *form* dari tampilan halaman menu keluar yang tampilannya sama seperti tampilan pada menu utama. Tetapi bedanya pada halaman menu keluar tersebut cukup kita klik tombol menu keluar, maka program yang kita jalankan langsung keluar dari program yang kita jalankan.



Gambar 4.6 Tampilan Halaman Menu Keluar

4.4 Pengujian Sistem

Pengujian merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang akan diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Penjelasan pada bab ini terdiri dari implementasi pengujian *white box* dan *black box*.

1. Pengujian *white box* dilakukan dengan menguji kode-kode program yang dibuat pada aplikasi. Pengujian dilakukan dengan mengecek semua kode pada program telah dieksekusi paling tidak satu kali. Pengujian ini dilakukan pada proses pengembangan sistem yakni pengujian kode program (*coding*).

white box testing akan di uji dengan beberapa tahapan yaitu :

- a. Pengujian seluruh keputusan yang menggunakan logikal.
- b. Pengujian keseluruhan *loop* yang ada sesuai batasan-batasannya.
- c. Pengujian pada struktur data yang sifatnya internal dan yang terjamin validitasnya.

Persyaratan dalam Pengujian *White Box* Testing, terdiri atas:

1. Medefinisikan semua alur logika.
2. Membangun kasus untuk digunakan dalam pengujian.
3. Mengevaluasi semua hasil pengujian.
4. Melakukan pengujian secara menyeluruh.

Jenis-jenis *White Box* Testing, terdiri atas :

1. *Basis Path*
 2. *Cyclomatic Complexity*
 3. *Graph Matrix*
2. Pengujian *black box* dilakukan untuk menguji apakah sistem yang dikembangkan sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem. *Black box* juga digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

Tabel 4.1 Pengujian Menu Utama

Kasus dan Hasil Uji (Data Benar)			
Masukkan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
<i>Form Info</i>	Menampilkan keterangan dari Info Algoritma <i>Levenshtein Distance</i> .	Dapat menampilkan penjelasan tentang apa itu Algoritma <i>Levenshtein Distance</i> .	Berhasil
<i>Form Levenshtein Distance</i>	Menampilkan hasil proses dari nama siswa yang di <i>Input</i> dengan cara mengetikkan nama siswa yang akan diuji melalui data berupa Ijazah SD dan daftar nama dari Dapodik.	Dapat menampilkan hasil dari jarak terdekat nama siswa yang diuji.	Berhasil
<i>Form Tentang</i>	Menampilkan informasi tentang gambar logo, nama Universitas serta data dari penulis.	Dapat menampilkan informasi tentang gambar logo, nama Universitas serta data dari penulis.	Berhasil
<i>Form Keluar</i>	Keluar dari program yang dijalankan.	Keluar dari program yang dijalankan.	Berhasil

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian, analisis, perancangan sistem, pembuatan program sampai tahap penyelesaian program, maka penulis dapat mengambil kesimpulan sebagai berikut :

1. Sistem Pendeteksi Penyesuaian Kesalahan Input Data Siswa Menggunakan Metode *Levenshtein Distance* berhasil dibangun dan mampu menangani proses pengolahan data dengan cara melakukan pengetikan nama siswa di kotak *input* sebagai data target serta nama siswa di kotak data sebagai data awal.
2. Sistem yang dibangun ini dapat mempermudah pegawai di SMP Negeri 2 Babalan dalam melakukan tugasnya mengolah data nama siswa berdasarkan nama yang ada di Ijazah SD dan di Dapodik yang sebelumnya sudah di unduh.
3. Metode *Levenshtein Distance* dapat digunakan untuk mendeteksi kemiripan nama siswa yang akan mengikuti Ujian Nasional dengan nilai *edit distance* terkecil menandakan bahwa kedua nama siswa tersebut mirip.

5.2 Saran

Dari proses analisis, perancangan, implementasi hingga pengujian sistem pada penulisan skripsi, didapatkan beberapa saran untuk pengembangan lebih lanjut perangkat lunak Sistem Pendeteksi Penyesuaian Kesalahan Input Data Siswa Menggunakan Metode *Levenshtein Distance*, yaitu :

1. Metode yang digunakan dalam sistem lebih dikembangkan lagi untuk proses perhitungannya, atau dapat mencari nilai *edit distance* yang lebih sesuai agar hasil yang didapat lebih akurat dan tidak perlu campur tangan *user* lagi.
2. Sistem ini dapat dikembangkan lagi sehingga dapat menjadi aplikasi berbasis *web*, agar *user* yang menggunakan sistem ini tidak perlu menginstal sistem apabila suatu saat *user* diharuskan menginstal ulang komputer atau terjadi kerusakan pada komputer *user*, sehingga hanya diperlukan koneksi internet saja dan membuka situs untuk sistem ini, dan *user* dapat mengerjakan tugasnya dimana saja dan kapan saja.
3. Sistem yang sudah berjalan agar dapat dikembangkan lagi dengan cara menginputkan nama-nama siswa dan siswi tersebut kedalam *database*, agar proses penyesuaian kesalahan input data siswa dapat mempermudah *user* dalam pekerjaannya.

DAFTAR PUSTAKA

- Abdul Chaer. (2011). *Ragam Bahasa Ilmiah* (hal.16). Jakarta: Rineka Cipta.
- Adiwiidya, B. M. (2009). *Algoritma Levenshtein Dalam Pendekatan Approximate String Matching*. Bandung: Institut Teknologi Bandung.
- Adriyani, N. M. M., Santiyasa, I. W., & Muliantara, A. (2010). Implementasi Algoritma Levenshtein Distance Dan Metode Empiris Untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia. *Jurnal Elektronik Ilmu Komputer Universitas Udayana*
- Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." *Seminar Nasional Informatika (SNIF)*. Vol. 1. No. 1. 2017.
- Ariyani, N. H., Sutardi, & Ramadhan, R. (2016). Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance. *SemanTIK*, Vol 2(1), 279–286. Retrieved from <http://ojs.uho.ac.id/index.php/semantik/article/download/1030/661>
- Asropudin, Pipin. (2013). *Teknologi Informasi Komunikasi*. Bandung: CV Titian Ilmu.
- Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In *IOP Conference Series: Materials Science and Engineering* (Vol. 300, No. 1, p. 012067). IOP Publishing.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.
- Batubara, Supina, Sri Wahyuni, and Eko Hariyanto. "Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam." *Seminar Nasional Royal (SENAR)*. Vol. 1. No. 1. 2018.
- Darmayuda, Ketut. (2014). *Aplikasi Basis Data Dengan Visual Basic.Net*. Bandung: Informatika.
- Fachri, B. (2018). Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika)*, 3, 98-102.
- Fachri, B. (2018, September). APLIKASI PERBAIKAN CITRA EFEK NOISE SALT & PAPPER MENGGUNAKAN METODE CONTRAHARMONIC MEAN FILTER. In *Seminar Nasional Royal (SENAR)* (Vol. 1, No. 1, pp. 87-92).
- Fachri, B., Windarto, A. P., & Parinduri, I. (2019). Penerapan Backpropagation dan Analisis Sensitivitas pada Prediksi Indikator Terpenting Perusahaan Listrik. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 5(2), 202-208.

- Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. *Int. J. Recent Trends Eng. Res*, 3(8), 58-64.
- Hidayatullah, Priyanto. (2014). *Visual Basic .NET*. Edisi Revisi. Bandung: Informatika
- INDRA PERMANA, A. M. I. N. U. D. D. I. N. "SISTEM PAKAR MENDETEKSI HAMA DAN PENYAKIT TANAMAN KELAPA SAWIT PADA PT. MOEIS KEBUN SIPARE-PARE KABUPATEN BATUBARA." (2013).
- Inmon, William H. (2005). "Building The Data Warehouse (4th ed.)". Indianapolis: Wiley Publishing, Inc.
- Ishak, A., Chen, Y. Y., & Yong, S. P. (2012). Distance-based hoax detection system. 2012 International Conference on Computer and Information Science, ICCIS 2012 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2012 - Conference Proceedings, 1, 215–220.
- Keraf, Gorys. (1991). *Tata Bahasa Rujukan Bahasa Indonesia untuk Tingkat Pendidikan Menengah*. Jakarta: Gramedia Widiasrana Indonesia.
- Kridalaksana, Harimurti. (2008). *Kamus Linguistik*. Jakarta: Gramedia Pustaka Utama.
- Kushartanti, dkk. (2005). *Pesona Bahasa: Langkah awal memahami linguistik*. Jakarta: PT Gramedia Pustaka Utama.
- Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." *Int. J. Recent Trends Eng. Res* 2.12 (2016): 140-151.
- Mulyani, Sri. (2016). *Sistem Informasi Manajemen*. Bandung: Abdi Sistematika.
- Mulyanto, A., (2010). Analisis Edit Distance Menggunakan Algoritma Dynamic Programming. *Saintek*, V(2).
- Permana, Aminuddin Indra. "Kombinasi Algoritma Kriptografi One Time Pad dengan Generate Random Keys dan Vigenere Cipher dengan Kunci EM2B." (2019).
- Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." *Seminar Nasional Teknologi Informasi Dan Multimedia*, ISSN. 2015.
- Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. *Int. J. Secur. Its Appl*, 10(8), 173-180.
- Putra, Randi Rian. "IMPLEMENTASI METODE BACKPROPAGATION JARINGAN SARAF TIRUAN DALAM MEMPREDIKSI POLA PENGUNJUNG TERHADAP TRANSAKSI." *JurTI (Jurnal Teknologi Informasi)* 3.1 (2019): 16-20.

- Sukamto, R. A., dan Shalahudin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Sutabri, Tata. (2004). *Sistem Informasi Akuntansi*. Yogyakarta: Andi.
- Taufiq. (2013). *Sistem Informasi Manajemen Konsep Dasar, Analisis dan Metode Pengembangan*. Yogyakarta : Graha Ilmu.
- Turban, Efrain, Linda Volonino. (2010). *Information Technology for Management*. Asia : John Willey & Sons.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.
- Wawan Setiawan, Eka Fitrajaya Rahman, Rasim. (2008). *Metodologi Pembelajaran Berbasis Komputer Dalam Upaya Menciptakan Kultur Pembelajaran Berbasis Teknologi Informasi Dan Komunikasi*, *Jurnal Pendidikan Teknologi Informasi Dan Komunikasi*, Vol 1 No 2.

