

**MESIN PENCARI KATA PADA TERJEMAHAN AL-QURAN
DENGAN MENGGUNAKAN METODE ALGORITMA
LEVENSHTEIN**

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk
mencapai gelar Sarjana Komputer



Disusun Oleh:

MUHAMMAD FAJAR DEWANTARA

0610960041

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU
KOMPUTER
MALANG
2013**

LEMBAR PERSETUJUAN

MESIN PENCARI KATA PADA TERJEMAHAN AL-QURAN DENGAN
MENGGUNAKAN METODE ALGORITMA LEVENSHTEIN

SKRIPSI

Untuk memenuhi sebagian persyaratan untuk

mencapai gelar Sarjana Komputer



Disusun Oleh:

MUHAMMAD FAJAR DEWANTARA

0610960041

Telah diperiksa dan disetujui oleh

Dosen Pembimbing

Pembimbing I

Drs. Marji, MT.
NIP. 196708011992031001

Pembimbing II

Lailil Muflikhah, S.Kom., M.Sc.
NIP. 197411132005011001

LEMBAR PENGESAHAN
MESIN PENCARI KATA PADA TERJEMAHAN AL-QURAN DENGAN
MENGGUNAKAN METODE ALGORITMA LEVENSHTEIN

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana

Komputer

Disusun oleh :

M FAJAR DEWANTARA

NIM. 0610960041

Skripsi ini telah diuji dan dinyatakan lulus tanggal 26 Juli 2013

Penguji I

Penguji II

Candra Dewi, S.Kom., M.Sc
NIP. 198111162005012004

Indriati, ST., M.Kom
NIK. 83101306120035

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom
NIK. 82062316110425

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, MT.
NIP. 196708011992031001

**PERNYATAAN
ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan dafta rupustaka.

Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Juli 2013
Mahasiswa,

Muhammad Fajar Dewantara
NIM. 0610960041

ABSTRAK

Pada masa sekarang ini sudah banyak aplikasi penunjang Al-Quran untuk memudahkan dalam mempelajarinya, yang biasa disebut Al-Quran Digital. Akan tetapi, ketika kita menggunakan aplikasi Al-Quran Digital yang sudah banyak beredar, Pengguna aplikasi mengalami kesulitan pada saat mencari sebuah kata yang pengguna inginkan. Hal ini terjadi ketika pengguna salah menuliskan sebuah kata yang ingin dicari dan aplikasi yang ada saat ini belum bisa mengenali atau membenarkan kata yang salah. Pada skripsi ini dibuat sistem pengecekan kata atau biasa disebut *string suggestion*, sehingga pengguna akan diarahkan pada kata-kata yang benar.

Algoritma yang digunakan untuk memberikan *string suggestion* adalah Algoritma *Levenshtein Distance* yang dapat menghitung *distance* atau jarak antar kata. *Levenshtein Distance* antara dua string ditentukan berdasarkan jumlah minimum perubahan yang diperlukan untuk melakukan transformasi dari satu bentuk string ke bentuk string yang lain. Melalui 3 tahapan yaitu, *Insertion*, *Deletion* dan *Substitution*.

Algoritma Levenshtein diterapkan untuk menganalisa arti dari surat Yasin. Hasil menunjukkan semakin sedikit huruf yang dirubah dalam suatu kata maka persentase yang diperoleh akan semakin besar. Kata dengan panjang 12 karakter, kemudian hanya satu huruf yang dirubah diperoleh persentase 92%, sedangkan kata yang memiliki panjang lima karakter diperoleh persentase 80%.

Kata Kunci: Al-Quran Digital, Algoritma Levenshtein, String Matching



ABSTRACT

These days there are many supporting applications that facilitate Al-Quran studies, which is called Digital Al-Quran. However, when we use Digital Quran that has been widely circulated, users experience difficulties when searching for a word. This happens when the user misspelled the searched word and the applications that exist today can not recognize or corrected the wrong word. For this problems, came the idea to make the word checking system or usually called string suggestion, therefore users will be directed to the correct words.

The algorithm used to provide string suggestion is Levenshtein Distance algorithm that can calculate the distance or spacing between words. Levenshtein Distance between two strings is determined based on the minimum number of changes necessary to transform from one form of string to another form. Through 3 stages, namely, Insertion, Deletion and Substitution.

Levensthein algorithm is applied to analyze the meaning of the Yasin. Results showed the fewer letters in a word is changed then the percentage earned will be even greater. Words with a length of 12 characters, then only one letter changed, obtained 92% percentage, while words with the length of five characters obtained the percentage of 80%.

Keywords: Digital Quran, Levenshtein Algorithm, String Matching



KATA PENGANTAR

Alhamdulillahi rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, Skripsi yang berjudul **“Mesin Pencari Kata pada Terjemahan Al-Quran dengan menggunakan Metode Algoritma Levenshtein”** ini dapat berjalan dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, Program Teknologi Informatika dan Ilmu Komputer (PTIIK), Universitas Brawijaya Malang.

Shalawat serta salam tetap tercurahkan kepada Baginda Rasulullah Muhammad *Shalallahu 'alaihi wasallam*, makhluk paling mulia yang senantiasa memberikan cahaya petunjuk, seorang uswatan hasanah yang telah membawa agama Allah yaitu agama Islam menjadi agama yang *Rahmatan Lil 'Alamin*.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Drs. Marji, M.T., selaku ketua program studi Teknik Informatika sekaligus pembimbing utama yang telah banyak memberikan bimbingan serta bantuan.
2. Lailil Muflikhah, S.Kom., M.Sc., selaku dosen pembimbing kedua yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
3. Issa Arwani, S.Kom., M.Sc. selaku sekretaris jurusan.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Teknik Informatika / Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.
5. Unit konseling Program Teknologi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah memberikan dukungan motivasi dan berbagai masukan terkait pelaksanaan penyusunan skripsi ini.



6. Segenap staf dan karyawan di Program Teknologi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
7. Kedua orang tua dan keluarga tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.
8. Rekan-rekan ilmu komputer 2006 di Program Studi Ilmu Komputer, Program Teknologi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan skripsi ini.
9. Dan semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu terima kasih atas semua bantuan yang telah diberikan.

Semoga skripsi ini bermanfaat bagi pembaca sekalian. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca.

Malang, 17 Juli 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
PERNYATAAN ORISINALITAS SKRIPSI	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR SOURCE CODE	xiii
BAB I : PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Metodologi	4
1.7 Sistematika Penulisan	4
BAB II : KAJIAN PUSTAKA DAN DASAR TEORI	6
2.1 Algoritma Levenshtein Distance	6



2.2 String Matching	10
2.3 Codelobster 4	12
2.4 PHP	12
2.5 MySQL	13
2.6 XAMPP	14
2.7 PhpMyAdmin	14
BAB III : METODE PENELITIAN DAN PERANCANGAN	16
3.1 Studi Literatur	17
3.2 Data Penelitian	17
3.3 Deskripsi Umum Sistem	17
3.4 Perancangan Proses	18
3.5 Perancangan Antarmuka Sistem	21
3.6 Perancangan Antarmuka Home	23
3.7 Perancangan Tabel	24
3.7.1 Tabel Data Quran	24
3.7.2 Tabel Data Quran_text	24
3.7.3 Tabel Nama Surat Al-Quran	25
3.7.4 Tabel Keywords	25
3.7.5 Tabel Hadits	25
3.7.6 Tabel Arab Latin	26
3.7.7 Tabel Users	26

3.7.8 Tabel Users_input	26
3.8 Entity Relationship Diagram	27
3.9 Perhitungan Manual	37
3.10 Skenario Ujicoba	37
BAB IV : IMPLEMENTASI	38
4.1 Lingkungan Implementasi	38
4.1.1 Lingkungan Perangkat Keras	38
4.1.2 Lingkungan Perangkat Lunak	38
4.2 Implementasi Program	38
4.2.1 Form Registrasi	39
4.2.2 Form Login	41
4.2.3 Proses Pencarian dan Menampilkan Hasil Kata	43
4.2.4 Form Daftar Surat Al-Quran	50
4.2.5 Form Daftar Hadits	54
4.2.6 Form User Input dan Update	56
4.3 Implementasi Antarmuka	59
4.3.1 Antarmuka Home	59
4.3.2 Antarmuka Register	60
4.3.3 Antarmuka Login	60
4.3.4 Antarmuka Daftar Surat Al-Quran	61
4.3.5 Antarmuka Daftar Hadits	61

4.3.6 Antarmuka Hasil Pencarian	62
4.3.7 Antarmuka Hasil Perhitungan	62
4.3.8 Antarmuka Input Catatan dan Arab Latin	63
BAB V : ANALISIS DAN PEMBAHASAN	64
5.1 Pengujian	64
5.2 Hasil Uji	64
5.3 Hasil Uji Program	65
5.3.1 Ujicoba pada satu kata dengan pengubahan pada setiap hurufnya	65
5.3.2 Ujicoba pada dua kata dengan panjang yang berbeda	76
5.4 Analisa Hasil Uji Program	77
BAB VI : KESIMPULAN	83
6.1 Kesimpulan	83
6.2 Saran	83
DAFTAR PUSTAKA	84



DAFTAR GAMBAR

Gambar 2.1 Tabel matriks perhitungan Edit Distance	9
Gambar 2.2 Pseudocode Algoritma Levenshtein	9
Gambar 2.3 Control Panel XAMPP	14
Gambar 2.4 Menu Utama PhpMyAdmin	15
Gambar 3.1 Tahapan-tahapan penelitian	16
Gambar 3.2 Flowchart proses sistem	18
Gambar 3.3 Flowchart Algoritma Levenshtein Distance	19
Gambar 3.4 Flowchart Proses Similarity	20
Gambar 3.5 Antarmuka sistem	22
Gambar 3.6 Antarmuka Home	23
Gambar 3.7 Entity Relationship Diagram (ERD)	27
Gambar 4.1 Tampilan Home	59
Gambar 4.2 Tampilan Register	60
Gambar 4.3 Tampilan Login	60
Gambar 4.4 Tampilan Daftar Surat Al-Quran	61
Gambar 4.5 Tampilan Daftar Hadits	61
Gambar 4.6 Tampilan Hasil Pencarian	62
Gambar 4.7 Tampilan Hasil Perhitungan	63
Gambar 4.8 Tampilan Input Catatan dan Arab Latin	63
Gambar 5.1 Grafik presentase kemiripan kata “surga”	66
Gambar 5.2 Grafik presentase kemiripan kata “sesungguhnya”	67
Gambar 5.3 Grafik presentase kemiripan kata “menciptakannya”	69
Gambar 5.4 Grafik presentase kemiripan kata “segera”	70
Gambar 5.5 Grafik presentase kemiripan kata “jahanam”	72

Gambar 5.6 Grafik presentase kemiripan kata “kemudharatan”	73
Gambar 5.7 Grafik presentase kemiripan kata “mata”	74
Gambar 5.8 Grafik presentase kemiripan kata “rahmat”	76
Gambar 5.9 Grafik panjang kata	77
Gambar 5.10 Grafik rata-rata persentase	78



DAFTAR TABEL

Tabel 3.1 Operasi matriks	21
Tabel 3.2 Atribut tabel data Quran	24
Tabel 3.3 Atribut tabel Quran_text	24
Tabel 3.4 Atribut tabel nama surat Al-Quran	25
Tabel 3.5 Atribut tabel keywords	25
Tabel 3.6 Atribut tabel hadits	25
Tabel 3.7 Atribut tabel arab latin	26
Tabel 3.8 Atribut tabel users	26
Tabel 3.9 Atribut tabel users_input	27
Tabel 3.10 Operasi Matriks GHAIB	28
Tabel 3.11 Hasil Perhitungan Matriks GHAIB	31
Table 3.12 Operasi Matriks GAIB	32
Tabel 3.13 Hasil Perhitungan Matriks GAIB	34
Tabel 3.14 Operasi Matriks GHOIB	35
Tabel 3.15 Hasil Perhitungan Matriks GHOIB	38
Tabel 3.16 Skenario Ujicoba	38
Tabel 5.1 Persentase tingkat kemiripan pada kata “surga”	65
Tabel 5.2 Persentase jumlah huruf yang berubah pada kata “surga”	65
Tabel 5.3 Persentase tingkat kemiripan pada kata “sesungguhnya”	66
Tabel 5.4 Persentase jumlah huruf yang berubah pada kata “sesungguhnya”	67
Tabel 5.5 Persentase tingkat kemiripan pada kata “menciptakannya”	68
Tabel 5.6 Persentase jumlah huruf yang berubah pada kata “menciptakannya” ..	68
Tabel 5.7 Persentase tingkat kemiripan pada kata “segera”	69
Tabel 5.8 Persentase jumlah huruf yang berubah pada kata “segera”	70



Tabel 5.9 Persentase tingkat kemiripan pada kata “jahanam”	71
Tabel 5.10 Persentase jumlah huruf yang berubah pada kata “jahanam”	71
Tabel 5.11 Persentase tingkat kemiripan pada kata “kemudharatan”	72
Tabel 5.12 Persentase jumlah huruf yang berubah pada kata “kemudharatan”	73
Tabel 5.13 Persentase tingkat kemiripan pada kata “mata”	73
Tabel 5.14 Persentase jumlah huruf yang berubah pada kata “mata”	74
Tabel 5.15 Persentase tingkat kemiripan pada kata “rahmat”	75
Tabel 5.16 Persentase jumlah huruf yang berubah pada kata “rahmat”	75
Tabel 5.17 Hasil uji coba menggunakan dua kata yang berbeda	76
Tabel 5.18 Rata-rata persentase tiap kata yang diujikan	78



DAFTAR SOURCE CODE

Source Code 4.1 Form Register.php	39
Source Code 4.2 Form Proses_registrasi.php	40
Source Code 4.3 Form Login.php	41
Source Code 4.4 Form Proses_login.php	42
Source Code 4.5 Pencarian dan Menampilkan hasil kata	43
Source Code 4.6 Levenshtein Distance	47
Source Code 4.7 LevenshteinDistance2	48
Source Code 4.8 Proses hasil_perhitungan.php	49
Source Code 4.9 Form daftar_surat_alquran.php	50
Source Code 4.10 Proses daftar_ayat.php	52
Source Code 4.11 Form daftar_hadits.php	54
Source Code 4.12 Proses detail_hadits.php	55
Source Code 4.13 Form user_input.php	56
Source Code 4.14 Proses proses_user_input.php	58



BAB I

PENDAHULUAN

1.1 Latar Belakang

Al-Quran merupakan kitab suci Agama Islam yang dipergunakan sebagai pedoman hidup umat Islam di dunia dan di akhirat. Semua norma-norma kehidupan tercantum dalam Al-Quran. Setiap kali umat Islam mempunyai masalah dalam kehidupan sehari-harinya, mereka selalu berpedoman pada Al-Quran. Al-Quran yang digunakan dalam keseharian adalah model Al-Quran konvensional, hal ini menimbulkan permasalahan baru ketika melakukan pencarian terhadap kasus tertentu. Pada masa sekarang ini sudah banyak aplikasi penunjang Al-Quran untuk memudahkan dalam mempelajarinya, yang biasa disebut Al-Quran Digital. Akan tetapi, ketika kita menggunakan aplikasi Al-Quran Digital yang sudah banyak beredar. Pengguna aplikasi mengalami kesulitan pada saat mencari sebuah kata yang pengguna inginkan. Hal ini terjadi ketika pengguna salah menuliskan kata tersebut dan aplikasi yang ada saat ini belum bisa mengenali atau membenarkan kata yang salah.

Latar belakang pengambilan judul ini adalah sulitnya mengimplementasikan Al-Quran dalam kehidupan sehari – hari terutama pemecahan suatu masalah oleh beberapa kalangan masyarakat, dikarenakan bentuk Al-Quran konvensional sulit untuk dipelajari. Ayat-ayat Al-Quran disusun tidak berdasarkan sebuah permasalahan, akan tetapi disusun berdasarkan sebuah peristiwa dalam masa lampau, sehingga memakan waktu yang cukup lama untuk mencari ayat-ayat yang dibutuhkan. Sebenarnya model pencarian digital seperti ini sudah banyak di internet (misalnya : <http://www.indoquran.web.id/>), kebanyakan diantaranya berfokus pada aqidah akhlaq. Nantinya pada penelitian ini tidak akan berfokus pada aqidah akhlaq saja, melainkan mencakup semua permasalahan yang tercantum pada Al-Quran.

Dengan adanya penelitian ini diharapkan masalah-masalah pencarian seperti itu dapat teratasi dengan baik, dan dapat memudahkan proses pencarian ayat-ayat



Al-Quran. Judul yang diambil dalam penelitian ini adalah **"Mesin pencari kata pada terjemahan Al-Quran dengan menggunakan metode Algoritma Levenshtein"**.

Pada penelitian sebelumnya membahas tentang Al-Quran Digital menggunakan text mining yang berbasis dekstop. Kata yang digunakan adalah kata berimbuhan yang ada pada terjemahan Al-Quran, sehingga dilakukan *preprosesing* yang mencakup beberapa tahapan yaitu, penghapusan format dan mark up, penghapusan tanda baca, pengubahan huruf besar ke huruf kecil, parsing dan stemming, kemudian dilakukan pembobotan. Pencarian kata sudah meliputi semua ayat pada Al-Quran [HER-10]. Adapun penelitian Muni Adriyani menggunakan algoritma *Levensthein Distance* dapat membantu mengatasi permasalahan pada kesalahan penulisan pada dokumen berbahasa indonesia [MUN-12]. Penelitian L.Cheng dan H.C.Ferreira untuk penyelesaian permasalahan *Pruned Convolutional codes dan Viterbi Decoding* menggunakan algoritma *Levensthein Distance* pada *Asynchronous Noisy Channels* menunjukan akurasi perbandingan yang lebih baik, Pa (subsitution error rate) = 0,001 dan Pd (deletion error rate) 0 [CHE-06].

Pada program Al-Quran Digital ini yang digunakan adalah kata dasar dan program yang digunakan berbasis web service, dengan menambahkan suatu fasilitas *string suggestion*, fasilitas *string suggestion* ini memungkinkan user melakukan kesalahan dalam pengetikan pada mesin pencari Al-Quran Digital. Hal ini akan mempermudah pencarian kata pada Al-Quran ketika user salah menuliskan kata yang dimaksud. Dalam hal ini digunakan *string matching* untuk menerapkan fasilitas *string suggestion*.

Beberapa algoritma dapat diimplementasikan dalam memberikan kata saran yang paling mendekati dari kata yang salah pengetikannya. Salah satunya algoritma *Levenshtein Distance* yang dapat menghitung perbedaan jarak antara dua string. Kemudian mencocokkan kata ke dalam *database* utama dan memberikan *string suggestion* untuk pemberian penulisan kata. Algoritma Levenshtein Distance merupakan matrik yang digunakan untuk mengukur

perbedaan jarak antara dua string, Levenshtein Distance antara dua string ditentukan berdasarkan jumlah minimum perubahan / pengeditan yang diperlukan untuk melakukan transformasi dari satu bentuk string ke bentuk string yang lain. Operasi yang dilakukan dan diperbolehkan digunakan dalam menentukan Levenshtein distance ini ada 3 macam yaitu ; Insertion, Deletion, dan Substitution [VLA-66].

1.2 Rumusan Masalah

Dari latar belakang yang sudah dijelaskan diatas, rumusan masalah yang dapat diambil adalah sebagai berikut :

1. Bagaimana mengimplementasi Algoritma Levenshtein Distance pada permasalahan mesin pencari kata pada terjemahan Al-Quran?
2. Mencari tingkat kemiripan antar kata, kemudian memberikan *string suggestion* kosa kata ketika user mencari suatu kata pada terjemahan Al-Quran.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Pencarian kata pada terjemahan Al-Quran dengan menggunakan masukan satu kosa kata.
2. Menampilkan ayat beserta terjemahannya dalam Bahasa Indonesia.
3. Diterapkannya metode pencarian arab latin pada Al-Quran, dikhkususkan pada surat Yasin.
4. Mencari tingkat kemiripan antar kata dan memberikan *string suggestion* kosa kata pada mesin pencari.

1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengimplementasi Algoritma Levenshtein Distance pada permasalahan mesin pencari kata pada terjemahan Al-Quran?



2. Tingkat kemiripan kata dengan menampilkan *string suggestion* pada mesin pencari Al-Quran.

1.5 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah dapat memudahkan dalam mempelajari dan mencari kata yang ada didalam terjemahan Al-Quran.

1.6 Metodologi

Metodologi yang digunakan dalam penulisan skripsi ini adalah:

a. Studi Literatur

Mempelajari teori yang berhubungan dengan *Algoritma Levenshtein Distance* dan *string matching* sebagai dasar penerapan string suggestion.

b. Pendefinisian dan Analisis Masalah

Mendefinisikan dan menganalisis masalah untuk memperoleh solusi yang tepat.

c. Perancangan dan Implementasi Sistem

Membuat perancangan perangkat lunak dan mengimplementasikan hasilnya untuk membuat perangkat lunak tersebut.

d. Uji Coba dan Analisis Hasil Implementasi

Menguji coba perangkat lunak yang dihasilkan dan menganalisa hasil dari implementasi perangkat lunak tersebut apakah telah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi menjadi lima bab dengan masing-masing bab diuraikan sebagai berikut:



BAB I : PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini menjelaskan dasar teori penunjang yang digunakan dalam penulisan skripsi, yaitu tentang Algoritma Levenshtein Distance.

BAB III : METODE DAN PERANCANGAN

Bab ini membahas mengenai penggunaan teori-teori dalam perancangan sistem. Serta membahas bagaimana sistem dan penelitian ini direncanakan. Perancangan sistem meliputi: desain arsitektur, proses, dan antarmuka perangkat lunak.

BAB IV : IMPLEMENTASI

Bab ini membahas tentang implementasi metode yang digunakan dalam Mesin pencari topik bahasan dalam Al-Quran dan Hadits dan uji coba terhadap program yang telah dibuat.

BAB V : HASIL DAN PEMBAHASAN

Bab ini membahas mengenai implementasi sistem disertai dengan potongan *source code* yang penting dalam aplikasi dan membahas tentang uji coba dari sistem yang dibuat, serta evaluasi untuk mengetahui kemampuan sistem.

BAB VI : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan penelitian lebih lanjut.



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Pada penelitian terdahulu pencarian ayat-ayat Al-Quran tentang permasalahan akidah berdasarkan konten menggunakan text mining telah berhasil membuat sistem pencarian ayat Al-Quran, hanya saja proses penyelesaian masalah berdasarkan akidah saja, jadi surat dan ayat yang ada pada sistem tidak menyeluruh di dalam Al-Quran Digital. Sistem yang digunakan adalah sistem *web service*. Kemudian dikembangkan lagi oleh peneliti yang lain agar lebih sempurna, tidak hanya pada permasalahan akidah saja, akan tetapi mampu mencakup semua permasalahan yang ada pada surat dan ayat Al-Quran. Sistem yang digunakanpun sudah berbeda yaitu menggunakan aplikasi berbasis desktop dengan metode yang sama yaitu text mining [HER-10].

Pada penelitian ini peneliti mencoba membuat sebuah aplikasi Al-Quran Digital, akan tetapi menggunakan metode yang berbeda yaitu menggunakan Algoritma Levenshtein. Penyelesaian masalah tidak hanya pada masalah akidah tetapi juga akan mencakup semua surat dan ayat pada Al-Quran, ditambahkan juga fitur *string suggestion* agar mempermudah dalam pencarian topik pada Al-Quran Digital.

2.1 Algoritma Levenshtein Distance

Algoritma Levenshtein Distance merupakan algoritma buatan Vladimir Levenshtein yang berhasil beliau temukan pada tahun 1966. Pada saat itu, beliau berhasil menemukan “distance” antara dua string masukan. Yang dimaksud dengan *distance* adalah jumlah modifikasi yang dibutuhkan untuk mengubah suatu bentuk string ke bentuk string yang lain. Algoritma Levenshtein Distance merupakan matrik yang digunakan untuk mengukur perbedaan jarak antara dua sekuens. Levenshtein Distance antara dua string ditentukan berdasarkan jumlah minimum perubahan / pengeditan yang diperlukan untuk melakukan transformasi dari satu bentuk string ke bentuk string yang lain.



Operasi yang dilakukan dan diperbolehkan digunakan dalam menentukan Levenshtein Distance ini ada 3 macam yaitu : [VLA-66]

1. *Insertion*

Insertion adalah operasi melakukan penyisipan sebuah karakter ke dalam sebuah string tertentu. Misalnya, user ingin menyisipkan sebuah karakter ‘m’ ke dalam sebuah string “julah” tepat setelah karakter ‘u’. Dengan dilakukannya *insertion*, string yang tadinya “julah” akan menjadi “jumlah”.

2. *Deletion*

Deletion adalah operasi melakukan penghilangan atau penghapusan sebuah karakter tertentu dari sebuah string. Misalnya, penulis ingin menghapus karakter ‘l’ pada string “julah”. Setelah dilakukan *deletion*, string akan menjadi “juah”.

3. *Substitution*

Substitution adalah operasi menukarkan sebuah karakter pada string tertentu dengan karakter lain. Misalnya, penulis ingin menukarkan karakter ‘b’ pada string “julah” dengan karakter baru ‘b’. Setelah dilakukannya *substitution*, string akan menjadi “jubah”.

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. Untuk penerapan algoritma ini dalam aplikasi, terdapat beberapa hal yang harus tersedia yaitu :

1. *Query* pengguna, yaitu kumpulan kata berupa string yang dimasukkan pengguna ke dalam sistem.
2. *Database* kata, yakni daftar seluruh kata yang valid dan dapat diterima sistem. *Database* digunakan sebagai pembanding antara *query* masukkan pengguna sehingga nantinya bias dicari kata-kata tertentu yang paling mendekati dengan masukkan pengguna.

Pengecekan dilakukan perkata masukkan pengguna, artinya setiap kata masukkan pengguna akan dicari dan dimiripkan dengan setiap kata yang ada di database lalu setiap kata yang paling mirip ditemukan di database akan disusun menjadi string suggestionnya.

Algoritma dasar penentuan jarak dua string ini dapat dibentuk melalui hubungan rekursif.

Basis:

$$\text{levDis} ("", "") = 0$$

$$\text{levDis} (s, "") = \text{levDis} ("", s) = |s|$$

Pada basis diatas terdapat dua baris. Baris pertama menyatakan dengan jelas bahwa dua string kosong tidak memiliki jarak, berarti untuk mengubah string yang satu ke yang lain tidak diperlukan operasi apapun. Baris kedua menyatakan bahwa jarak antara suatu string tidak kosong dengan string kosong adalah sebesar panjang (jumlah karakter) di dalam string yang tidak kosong [LLO-99].

		G	H	O	I	B
	0	1	2	3	4	5
G	1	0	1	2	3	4
H	2	1	0	1	2	3
A	3	2	1	1	2	3
I	4	3	2	2	1	2
B	5	4	3	3	2	1

Gambar 2.1 Tabel matriks perhitungan Edit Distance



Algoritma yang sesuai untuk matriks di atas adalah :

```

Function levDis (s1 : string, s2 : string) : integer kamus
    i, j, cost : integer
    m : array [0..s1.length, 0..s2.length] of integer
algoritma
for i = 0 to s1.length do
    for j = 0 to s2.length do
        if i = 0 then
            m[i,j] = j {perbandingan dengan kosong}
        else if j = 0 then
            m[i,j] = i {perbandingan dengan kosong}
        else {implementasi pemrograman dinamis}
            if s1[i] = s2[j] then
                cost = 0
            else
                cost = 1
            m[i,j] = minimum (
                m[i-1, j-1] + cost, {substitusi}
                m[i-1, j] + 1, {penghapusan}
                m[i, j-1] + 1, {penambahan}
            )
Return m[s1.length, s2.length]

```

Gambar 2.2 Pseudocode Algoritma Levenshtein

Pada pseudocode dijelaskan bahwa terdapat tiga alternatif untuk menentukan jarak kedua string. Pertama, Jika s1 dan s2 sama, maka s1 dan s2 tidak perlu dipertukarkan berarti jaraknya $\text{levDis}(s1, s2) + 0$, jika berbeda berarti hanya tinggal mengubah s1 menjadi s2 saja, berarti jaraknya $\text{levDis}(s1, s2) + 1$. Dalam hal ini operasi yang dilakukan adalah operasi substitusi. Kedua, dapat juga dilakukan operasi penghapusan s1 jaraknya menjadi $\text{levDis}(s1-1, s2) + 1$. Ketiga, mirip seperti yang kedua dapat dilakukan juga operasi penyisipan s2 sehingga jaraknya adalah $\text{levDis}(s1, s2-1) + 1$ [LLO-99].

Ketiga alternatif diatas adalah semua kemungkinan perubahan yang ada, dan dari antara ketiganya dicari yang mana yang paling sedikit jaraknya dengan fungsi min yang mencari nilai paling minimum diantara tiga nilai.

Karena fungsi levenshtein distance menghitung berapa banyak perubahan yang dibutuhkan untuk mengubah s1 menjadi s2, nilai kemiripan (*similarity*), dapat didefinisikan sebagai :

$$\text{similarity} = \left(1 - \frac{\text{levenshteinDistance}(s1, s2)}{\max(|s1|, |s2|)} \right) \times \text{limit}$$

Dimana,

s1 = kata yang dimasukkan

s2 = kata yang dimaksud

limit = 100%

Konstanta *limit* digunakan untuk mengatur batas atas skala distribusi nilai kemiripan.

2.2 String Matching

Pengertian *string* menurut *Dictionary of Algorithms and Data Structures*, *National Institute of Standards and Technology (NIST)* adalah susunan dari karakter-karakter (angka, alphabet atau karakter yang lain) dan biasanya direpresentasikan sebagai struktur data array. *String* dapat berupa kata, frase, atau kalimat [PAU-99].

Pencocokan *string* merupakan bagian penting dari sebuah proses pencarian *string* dalam sebuah dokumen. Hasil dari pencarian sebuah *string* dalam dokumen tergantung dari teknik atau cara pencocokan *string* yang digunakan.

Pencocokan *string* (*string matching*) secara garis besar dapat dibedakan menjadi dua yaitu :[BIN-95]

1. *Exact string matching*, merupakan pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokan memiliki jumlah maupun urutan karakter dalam *string* yang sama. Contoh : kata “halo” akan menunjukkan kecocokan hanya dengan kata “halo”.



2. *Inexact string matching* atau *Fuzzy string matching*, merupakan pencocokan *string* secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokan memiliki susunan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi *string – string* tersebut memiliki kemiripan baik kemiripan textual / penulisan (*approximate string matching*) atau kemiripan ucapan (*phonetic string matching*). *Inexact string matching* masih dapat dibagi lagi menjadi dua yaitu :
 - a. Pencocokan *string* berdasarkan kemiripan penulisan (*approximate string matching*) merupakan pencocokan *string* dengan dasar kemiripan dari segi penulisannya (jumlah karakter, susunan karakter dalam dokumen). Tingkat kemiripan ditentukan dengan jauh tidaknya beda penulisan dua buah *string* yang dibandingkan tersebut dan nilai tingkat kemiripan ini ditentukan oleh pemrogram (*programmer*). Contoh : *kharong* dengan *karung*, memiliki jumlah karakter yang berbeda. Jika perbedaan dua karakter ini dapat ditoleransi sebagai sebuah kesalahan penulisan maka dua *string* tersebut dikatakan cocok.
 - b. Pencocokan *string* berdasarkan kemiripan ucapan (*phonetic string matching*) merupakan pencocokan *string* dengan dasar kemiripan dari segi pengucapannya meskipun ada perbedaan penulisan dua *string* yang dibandingkan tersebut. Contoh *step* dengan *steb* dari tulisan berbeda tetapi dalam pengucapannya mirip sehingga dua *string* tersebut dianggap cocok. Contoh yang lain adalah *step*, dengan *steppe*, *step*, *stepp*, *stepe*.

Exact string matching bermanfaat jika user ingin mencari *string* dalam dokumen yang sama persis dengan *string* masukan. Tetapi jika user menginginkan pencarian *string* yang mendekati dengan *string* masukan atau terjadi kesalahan penulisan *string* masukan maupun dokumen objek pencarian, maka *inexact string matching* yang bermanfaat. Beberapa algoritma *exact string matching* antara lain : *Algoritma Knuth-Morris-Pratt*, *Boyer Moore*, dll. Beberapa algoritma *phonetic*

string matching antara lain : *Soundex*, *metaphone*, *caverphone*, *phonex*, *NYSIIS*, *Jaro-Winkler*, dll [CHR-97].

2.3 Codelobster 4

Codelobster adalah sebuah framework PHP editor atau IDE (Integrated Development Environment) untuk bahasa pemrograman PHP. Codelobster juga menyediakan beberapa plugin yaitu yii, joomla, jquery, CakePhp Symfony, Drupal, Smarty, Wordpress, Zen, Codeigniter dll.

Codelobster menggunakan semua fitur standar seperti PHP, HTML, CSS, Javascript dan juga kode validator beserta SQL manager.

2.4 PHP

PHP adalah salah satu bahasa pemrograman yang berjalan pada sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. Dengan menggunakan PHP sebuah *website* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website/computer client* akan diolah dan disimpan pada database *web server* dan dapat ditampilkan kembali apabila diakses [ABD-01].

Sebagian besar *web* yang ada di internet dibangun dengan menggunakan bahasa pemrograman PHP. Beberapa alas an penggunaan PHP adalah :

1. PHP merupakan bahasa pemrograman *Open Source* dan dikembangkan oleh komunitas sehingga bias didapatkan dengan mudah dan digunakan tanpa harus mengeluarkan biaya.
2. PHP dapat digunakan pada sistem operasi seperti *Linux*, *Microsoft Windows*, *Solaris*, *Open BSD* dan lain sebagainya.
3. PHP didukung oleh beberapa *web server* seperti *Apache*, *Personal Web Server* dan *Internet Information Server*.
4. Dalam penggunaannya PHP mendukung beberapa *database* seperti *Interbase*, *PostgreSQL*, *Sybase*, *MySQL*, *FrontBase*, *SQLite*, *Informix*, *Oracle* dan *ODBC*.

5. PHP juga memberikan kemudahan dalam menampilkan berbagai macam teks, gambar dan file PDF.

2.5 MySQL

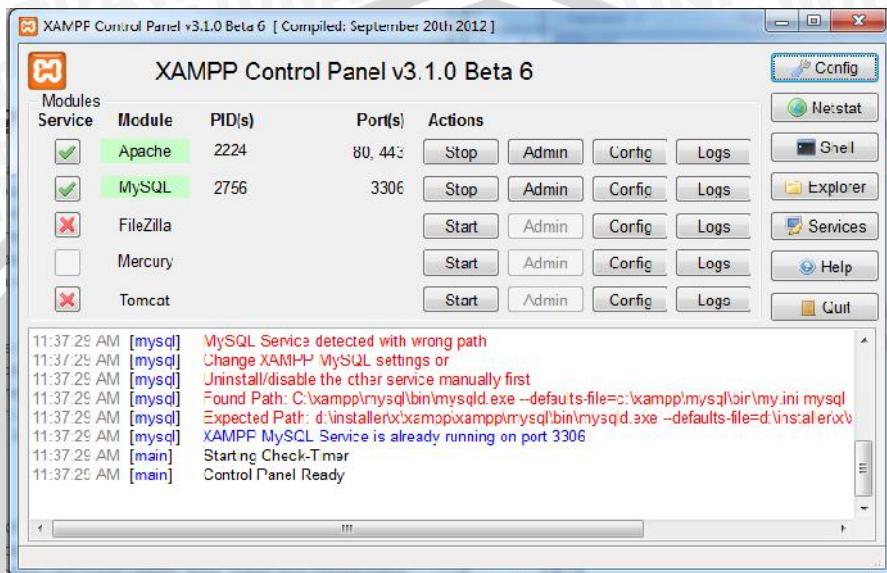
MySQL adalah suatu *database server* yang sangat terkenal di dunia dan merupakan *database SQL* yang *Open Source*. MySQL merupakan *database server* dimana pemrosesan data terjadi di *server* dan *client* hanya mengirim data dan memindah data. Pengaksesan dapat dilakukan dimana saja dan oleh siapa saja dengan catatan, computer telah terhubung ke *server*. Lain halnya dengan *database desktop* dimana segala pemrosesan data seperti penambahan dan penghapusan data harus dilakukan pada satu computer. MySQL termasuk dalam kategori *database management system*, yaitu *database* yang terstruktur dalam pengolahan dan penampilan data.

MySQL merupakan *Relational Database Management System* (RDBMS) yaitu hubungan antar tabel yang berisi data-data pada suatu *database*. Hal tersebut lebih baik daripada jika semua data terkumpul menjadi satu dalam satu tabel. Kelebihan hal diatas, yaitu dapat mempercepat, pencarian suatu data. Tabel-tabel tersebut di-link oleh suatu relasi yang memungkinkan untuk mengkombinasikan data dari beberapa tabel ketika seorang user menginginkan menampilkan informasi dari suatu *database* maka data akan dapat ditampilkan [ABD-01].

2.6 XAMPP

XAMPP atau X (*Cross Platform*) Apache MySQL PHP Perl adalah sebuah perangkat lunak (*software*) yang dibuat oleh tim dari Apache Friends (www.apachefriends.com) yang fungsinya adalah untuk menjalankan program PHP, MySQL dan Perl dalam satu waktu yang bersamaan. XAMPP memudahkan pada web developer untuk mengembangkan dan membuat sebuah website di local PC/Laptop, sehingga proses pembuatan sebuah website menjadi lebih aman dan cepat dibandingkan melakukan proses pembuatan website lewat online server.

Sebenarnya ada banyak sekali jenis software lain yang dapat menjalankan Apache, MySQL dan PHP. Namun karena proses instalasi dan pengembangan XAMPP lebih mudah, maka banyak yang menyarankan untuk menggunakan XAMPP untuk menjalankan AMP (Apache, MySQL, PHP).



Gambar 2.3 Control Panel XAMPP

2.7 PhpMyAdmin

PhpMyAdmin adalah sebuah aplikasi web yang ditulis menggunakan bahasa pemrograman PHP. Sebagaimana aplikasi-aplikasi lain untuk lingkungan web (aplikasi yang dibuka menggunakan peramban atau browser), PhpMyAdmin juga mengandung unsur HTML/XHTML, CSS dan juga kode JavaScript. Aplikasi web ini ditujukan untuk memudahkan pengelolaan basis data MySQL dengan penyajian tampilan web (user interface) yang lengkap dan menarik.

PhpMyAdmin merupakan aplikasi web yang bersifat Open Source sejak pertama kali dibuat dan dikembangkan. Dengan dukungan dari banyak developer dan translator, aplikasi web PhpMyAdmin mengalami perkembangan yang cukup pesat dengan ketersediaan banyak pilihan bahasa, sampai saat ini, ada kurang lebih 65 bahasa yang didukung oleh aplikasi web PhpMyAdmin.



Gambar 2.4 Menu Utama PhpMyAdmin

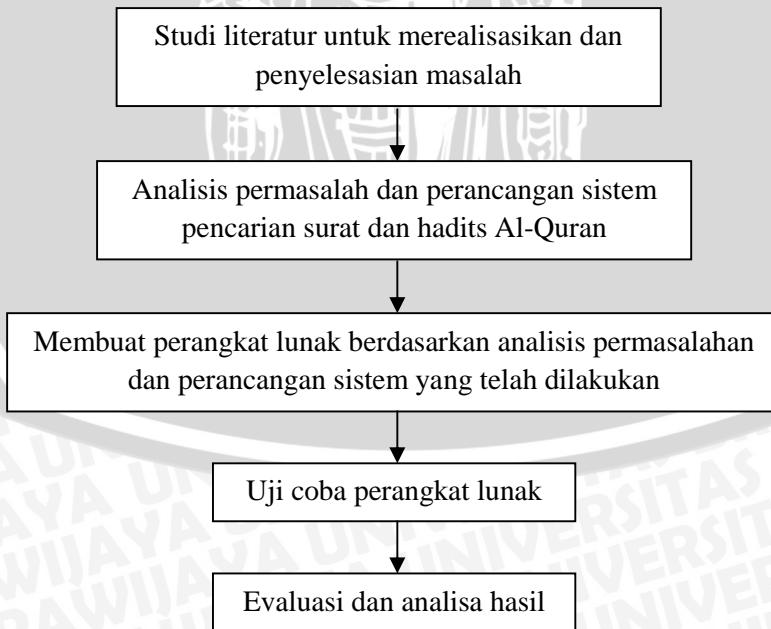
BAB III

METODE PENELITIAN DAN PERANCANGAN

Pada bab ini menjelaskan langkah-langkah yang akan ditempuh dalam menyusun penelitian. Tahapan penelitian ini dirancang untuk membangun sistem Al-Quran Digital yang dapat memberikan kemudahan dalam pencarian topik dan memahami isi Al-Quran dan Hadits. Tahap-tahap pembuatannya adalah sebagai berikut :

1. Studi literature untuk merealisasikan tujuan dan penyelesaian masalah.
2. Analisis permasalahan dan perancangan sistem dalam pencarian surat dan hadits Al-Quran.
3. Membuat sistem berdasarkan analisis permasalahan dan perancangan yang dilakukan.
4. Melakukan uji coba terhadap sistem.
5. Melakukan evaluasi (analisa hasil) yang diperoleh dari uji coba sistem.

Tahapan-tahapan penelitian yang dilakukan diilustrasikan pada gambar 3.1.



Gambar 3.1 Tahapan-tahapan penelitian

3.1 Studi Literatur

Dalam penelitian ini dibutuhkan studi literature untuk merealisasikan tujuan dan penyelesaian masalah. Teori-teori yang berkaitan dengan aplikasi Al-Quran Digital dan Algoritma Levenshtein digunakan sebagai dasar penelitian yang didapat dari buku, jurnal dan *browsing* dari internet. Dari data yang diperoleh akan diolah sehingga dapat digunakan untuk analisis. Setelah dilakukan analisis maka dapat diimplementasikan ke dalam aplikasi.

3.2 Data Penelitian

Untuk mengetahui bagaimana kinerja sistem aplikasi Al-Quran Digital, dibutuhkan database dari Al-Quran dan Hadits. Penelitian ini menggunakan database yang berasal dari Quran Database (www.qurandatabase.org), database yang disediakan dalam websites tersebut sudah sangat lengkap mulai dari database berupa MySQL, Ms Acces, Excel Format dll.

3.3 Deskripsi Umum Sistem

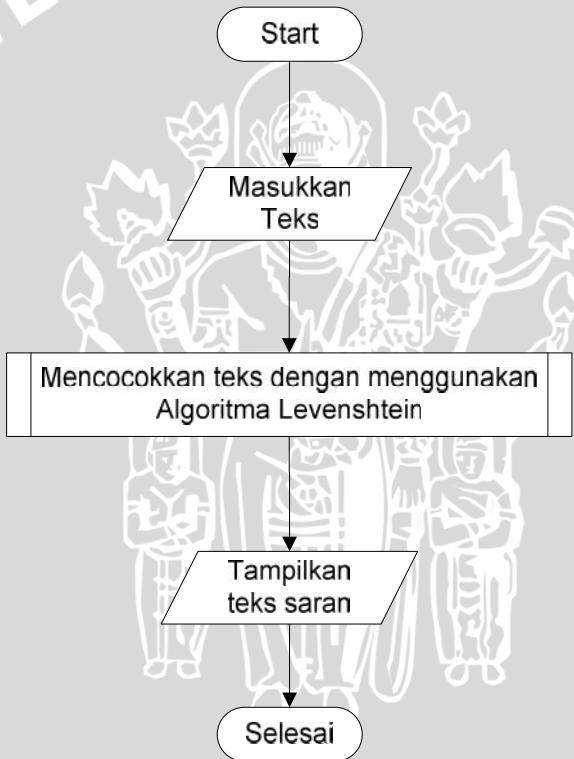
Secara umum sistem yang dibangun adalah perangkat lunak yang dapat memudahkan user untuk mempelajari isi dari Al-Quran dan Hadits. Metode yang digunakan dalam perangkat lunak ini adalah Algoritma Levenshtein Distance.

Dalam Algoritma Levenshtein Distance melakukan tiga bentuk operasi. Operasi pertama adalah operasi penyisipan sebuah karakter ke dalam sebuah string tertentu yang biasa disebut *Insertion*. Operasi kedua adalah operasi melakukan penghilangan atau penghapusan sebuah karakter tertentu dari sebuah string yaitu *Deletion*. Operasi ketiga adalah operasi menukar sebuah karakter pada string tertentu dengan karakter lain yang disebut *Substitution*.

3.4 Perancangan Proses

Tahapan proses dalam aplikasi Al-Quran Digital menggunakan Algoritma Levenshtein Distance yang terdiri dari tiga operasi yaitu *Insertion*, *Deletion* dan *Substitution*. Setelah tiga operasi selesai kemudian dilakukan perhitungan nilai kemiripan (*similarity*), dari perhitungan nilai kemiripan ini akan didapatkan persentase sebuah kata dengan pencapaian kemiripan mendekati 100%.

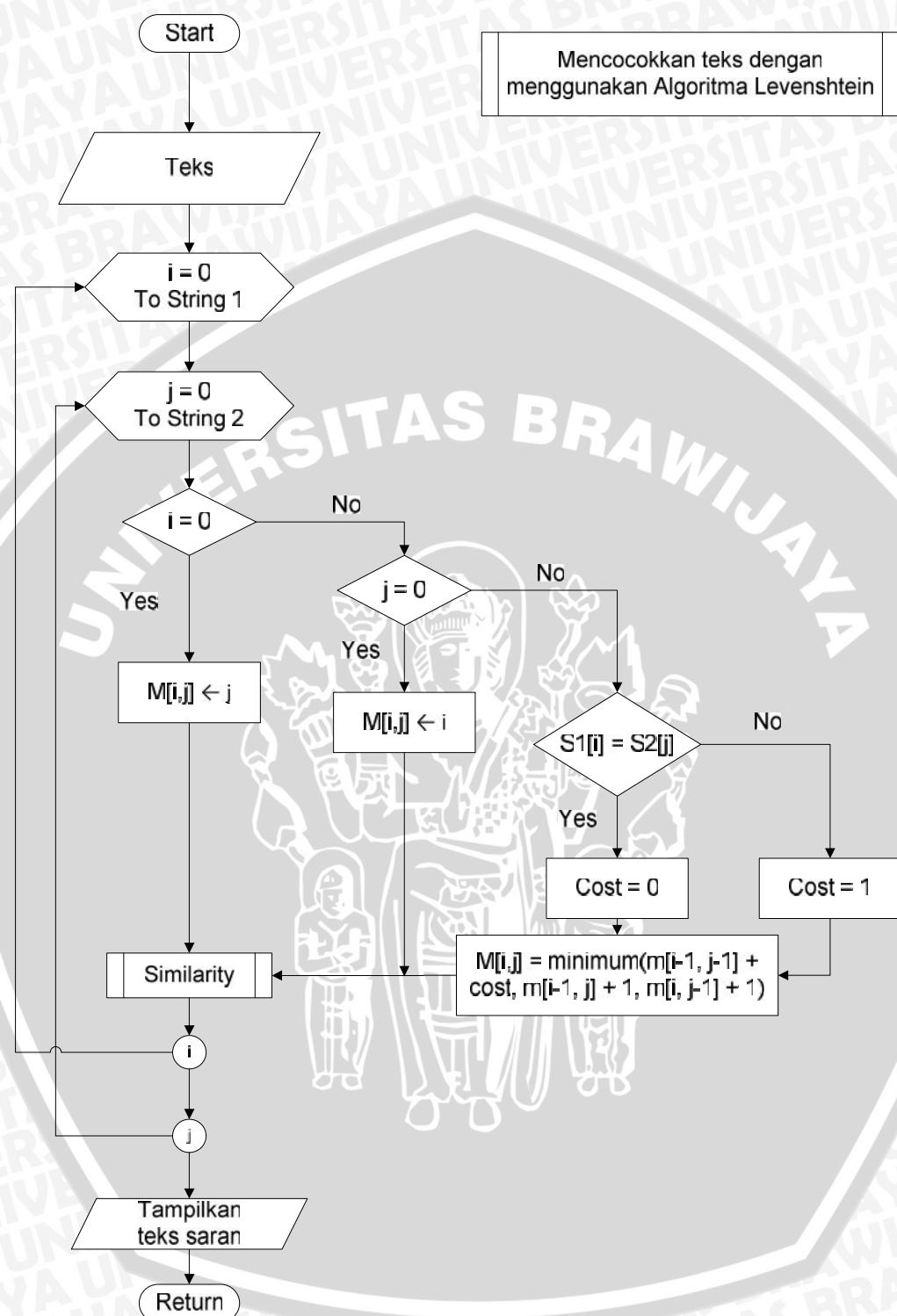
Diagram proses sistem dalam Al-Quran Digital dapat dilihat pada gambar 3.2 berikut ini.



Gambar 3.2 Flowchart proses sistem

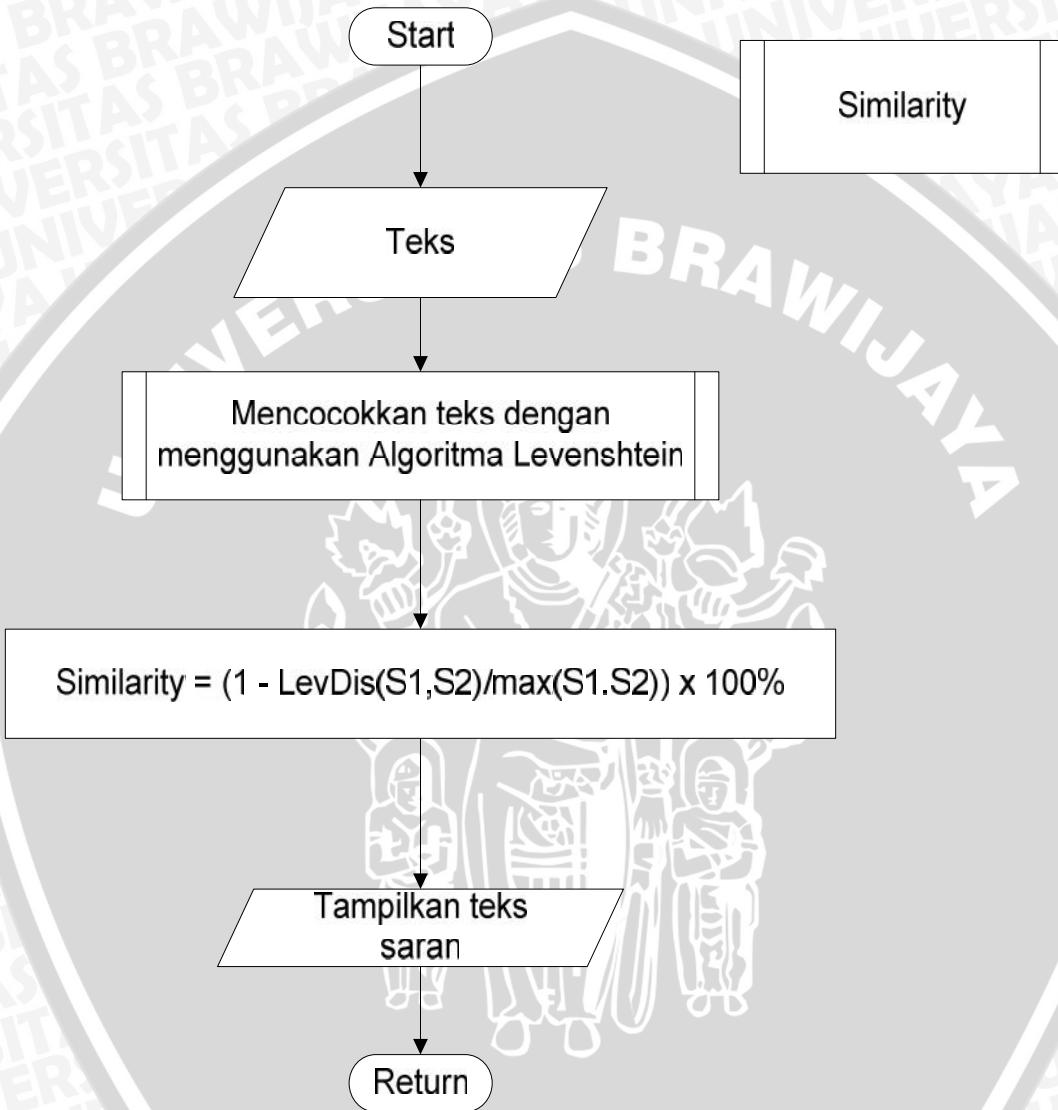
Pencocokkan teks menggunakan Algoritma Levenshtein, adapun flowchart untuk Algoritma Levenshtein dapat dilihat pada gambar 3.3.





Gambar 3.3 Flowchart Algoritma Levenshtein Distance

Setelah proses penentuan jarak antar kata terpenuhi kemudian akan dihitung similaritynya, similarity digunakan adalah untuk menghitung persentase setiap kata yang akan dijadikan untuk *string suggestion*. Untuk lebih jelasnya akan ditampilkan pada gambar 3.4.



Gambar 3.4 Flowchart Proses Similarity

Matriks yang akan digunakan dari operasi dapat dilihat pada Tabel 3.1 berikut ini.

Tabel 3.1 Operasi matriks

		G	H	O	I	B
	0	1	2	3	4	5
G	1					
H	2					
A	3					
I	4					
B	5					

Pada tabel diatas, elemen baris 1 kolom 1 ($M[1,1]$) adalah jumlah operasi yang diperlukan untuk mengubah substring dari kata GOIB yang diambil mulai dari karakter awal sebanyak 1 (G) ke substring dari kata GHOIB yang diambil mulai dari karakter awal sebanyak 1 (G).

Berarti elemen $M[p,q]$ adalah jumlah operasi antara substring kata pertama yang diambil mulai dari awal sebanyak p dengan substring kata kedua yang diambil dari awal sebanyak q .

Elemen terakhir (yang paling kanan bawah) adalah elemen yang nilainya menyatakan jarak kedua string yang dibandingkan.

3.5 Perancangan Antarmuka Sistem

Antarmuka sistem merupakan tampilan bagaimana proses Algoritma Levenshtein Distance berjalan pada Al-Quran Digital dan Hadits. Adapun antarmuka sistem ditunjukkan pada Gambar 3.2.



Gambar 3.5 Antarmuka sistem

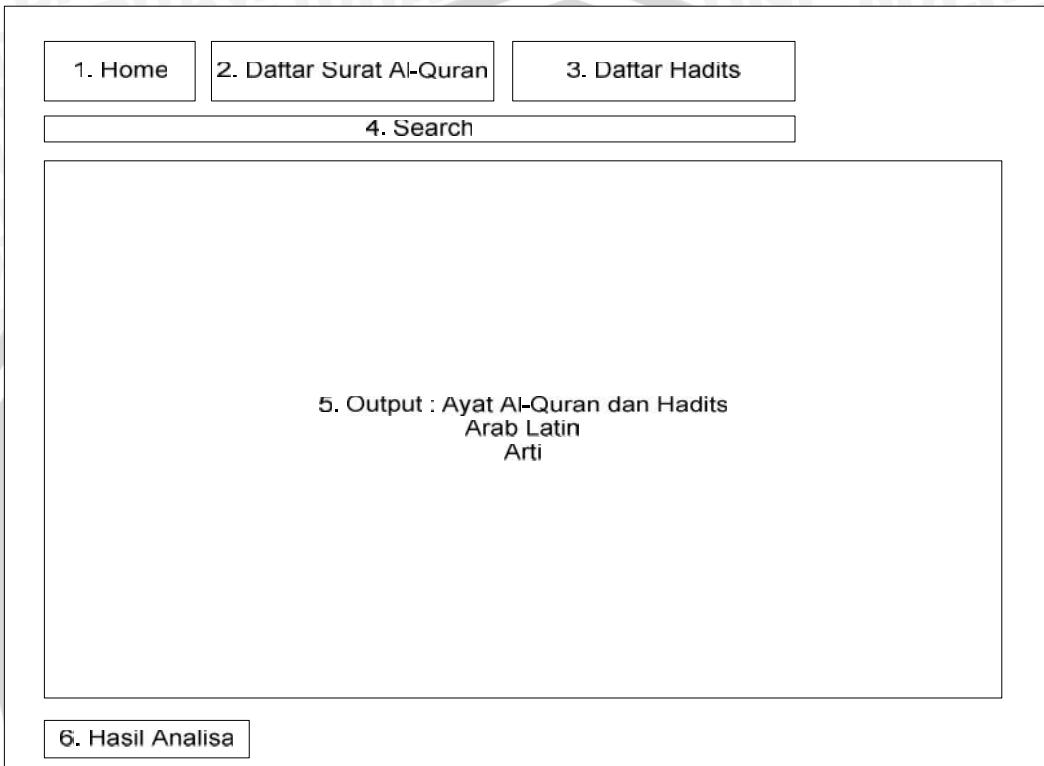
Penjelasan dari bagian-bagian antarmuka sistem adalah :

1. Home, digunakan ketika user ingin kembali ke menu utama website.
2. Daftar Surat Al-Quran, berfungsi untuk menampilkan daftar isi surat-surat pada Al-Quran.
3. Daftar Hadits, berfungsi untuk menampilkan daftar isi Hadits-hadits yang sudah terdaftar dalam database Al-Quran Digital.
4. Search, untuk pencarian *item*.
5. Analisis Persentase, untuk menampilkan hasil perhitungan kata berupa persentase.
6. Analisis Matriks, untuk menampilkan perhitungan manual kata berupa matriks berdasarkan kata yang dipilih.



3.6 Perancangan Antarmuka Home

Antarmuka *home* merupakan tampilan awal sistem. Pada antarmuka *home* terdapat sejumlah item yaitu ; Home, Daftar Surat Al-Quran dan Hadits. Adapun antarmuka home ditunjukkan pada Gambar 3.3 berikut.



Gambar 3.6 Antarmuka Home

Penjelasan dari bagian-bagian antarmuka home diatas adalah :

1. Home, digunakan ketika user ingin kembali ke menu utama website.
2. Daftar Surat Al-Quran, berfungsi untuk menampilkan daftar isi surat-surat pada Al-Quran.
3. Daftar Hadits, berfungsi untuk menampilkan daftar isi Hadits-hadits yang sudah terdaftar dalam database Al-Quran Digital.
4. Search, untuk pencarian *item*.
5. Output, merupakan tampilan hasil dari pencarian. Didalamnya berisi ayat-ayat Al-Quran, Arab Latin dan arti dari ayat-ayat Al-Quran.



- Hasil Analisa, pada hasil analisa ini akan ditampilkan tabel perhitungan berupa persentase dari sistem pencarian.

3.7 Database

3.7.1 Tabel Data Quran

Tabel data Quran berisi tentang surat-surat dan ayat-ayat yang tercantum dalam Al-Quran.

Tabel 3.2 Atribut tabel data Quran

Field	Type	Keterangan
Index	Int(4)	Kode nomor surat Al-Quran untuk memudahkan pemanggilan ke Database.
sura	Int(3)	Kode nomor surat pada Al-Quran.
aya	Int(3)	Kode nomor ayat pada suatu surat di dalam Al-Quran.
gambar	Varchar(16)	Gambar ayat-ayat Al-Quran berupa png beserta kode nomor.

3.7.2 Tabel Data Quran_text

Tabel data Quran_text berisi tentang terjemahan Al-Quran dalam Bahasa Indonesia.

Tabel 3.3 Atribut tabel Quran_text

Field	Type	Keterangan
ID	Int(3)	Kode nomor terjemahan Al-Quran dalam Bahasa Indonesia.
SuraID	Int(11)	Kode nomor terjemahan surat pada Al-Quran.
VerseID	Int(11)	Kode nomor terjemahan ayat Al-Quran.
AyahText	Text	Berisi terjemahan surat per-ayat.



3.7.3 Tabel Nama Surat Al-Quran

Nama surat dan jumlah ayat terletak pada tabel nama surat Al-Quran.

Tabel 3.4 Atribut tabel nama surat Al-Quran

Field	Type	Keterangan
ID	Int(5)	Kode nomor surat Al-Quran.
Nama	Text	Nama surat pada Al-Quran.
Ayat	Int(20)	Jumlah ayat pada satu surat Al-Quran.

3.7.4 Tabel Keywords

Tabel keywords digunakan untuk menyimpan sementara hasil dari pencarian pada database utama, kata yang sering dipakai akan masuk pada database ini.

Tabel 3.5 Atribut tabel keywords

Field	Type	Keterangan
keyword	Varchar(128)	Berisi kata yang sering dipakai dalam pencarian.
hit	Int(11)	Jumlah pemakaian kata, seberapa sering kata tersebut dipakai.
Id_ayat	Varchar(256)	Berisi ayat-ayat yang terpakai.

3.7.5 Tabel Hadits

Tabel hadits berisi tentang hadits-hadits yang ada didalam Al-Quran Digital, total ada 38 hadits yang sudah terdaftar dalam Al-Quran Digital ini.

Tabel 3.6 Atribut tabel hadits

Field	Type	Keterangan
ID	Int(4)	Kode nomor hadits.
nama_hadits	Varchar(100)	Nama-nama hadits yang ada di dalam Al-Quran Digital.
sura	text	Surat Hadits.

arti	text	Terjemahan dari surat hadits dalam Bahasa Indonesia.
------	------	--

3.7.6 Tabel Arab Latin

Tabel arab latin digunakan untuk menyimpan tulisan arab latin dari bacaan Al-Quran.

Tabel 3.7 Atribut tabel arab latin

Field	Type	Keterangan
Index	Int(4)	Kode nomor surat Al-Quran.
sura	Int(4)	Kode nomor surat pada Al-Quran.
aya	Int(4)	Kode nomor ayat pada surat Al-Quran.
text	Text	Berisi bacaan surat Al-Quran dalam Bhs Indonesia.

3.7.7 Tabel Users

Tabel users digunakan untuk menyimpan Login user, berisi id, username dan password.

Tabel 3.8 Atribut tabel users

Field	Type	Keterangan
ID	Int(11)	Kode user.
Username	Varchar(15)	Nama user untuk <i>Login</i> .
Password	Varchar(32)	Password user untuk <i>Login</i> .
Nama	Varchar(45)	Nama lengkap user.

3.7.8 Tabel Users_input

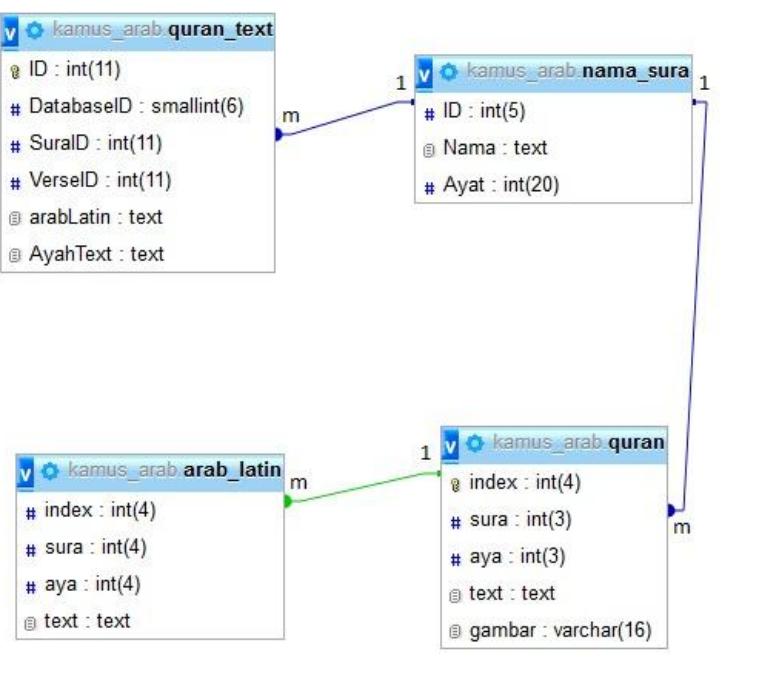
Tabel users_input digunakan untuk menyimpan data yang dimasukkan user kedalam program Al-Quran Digital.

Tabel 3.9 Atribut tabel users_input

Field	Type	Keterangan
ID	Int(11)	Kode database users_input.
User_id	Int(11)	Untuk pemanggilan ke tabel users.
Ayat_id	Int(11)	Untuk pemanggilan ke tabel quran atau quran_text.
Catatan	Text	Menyimpan catatan user.
Arab_latin	Text	Menyimpan arab latin yang dimasukkan user.

3.8 ERD (Entity Relationship Diagram)

ERD (Entity Relationship Diagram) berfungsi untuk menjelaskan hubungan antar tabel dalam sebuah basis data. Berikut ini adalah hubungan antar tabel data Al-Quran.



Gambar 3.6 Entity Realationship Diagram (ERD)

3.9 Perhitungan Manual

Pada sub bab ini akan dibahas tentang contoh perhitungan manual. Misalkan user akan mencari kata Ghoib di dalam kitab Al-Quran, akan tetapi user salah menuliskan kata yang dimaksud. Ketentuan similarity yang digunakan 65%. Rumus similarity yang akan dipakai adalah sebagai berikut :

$$\text{similarity} = \left(1 - \frac{\text{levenshteinDistance}(s1, s2)}{\max(|s1|, |s2|)} \right) \times \text{limit}$$

Dimana,

$s1$ = kata yang dimasukkan

$s2$ = kata yang dimaksud

limit = 100%

Konstanta limit digunakan untuk mengatur batas atas skala distribusi nilai kemiripan.

Membandingkan kata GHOIB dengan kata GHAIB. Membuat matriks untuk membandingkan dua kata tersebut. Matriks yang dimaksud adalah sebagai berikut:

Tabel 3.10 Operasi Matriks GHAIB

		G	H	O	I	B
	0	1	2	3	4	5
G	1					
H	2					
A	3					
I	4					
B	5					

$$m(i,j) = \min (m[i, j] + \text{cost}, m[i-1, j] + 1, m[i, j-1] + 1)$$

$$m(0,0) = 0 \quad m(1,0) = 1$$

$$m(0,1) = 1 \quad m(2,0) = 2$$

$$m(0,2) = 2 \quad m(3,0) = 3$$

$$m(0,3) = 3 \quad m(4,0) = 4$$

$$m(0,4) = 4 \quad m(5,0) = 5$$

$$m(0,5) = 5$$

$$m(1,1) = \min (m[0,0] + 0, m[0,1] + 1, m[1,0] + 1)$$

$$= \min (0, 2, 2)$$

$$= \min 0$$

$$\begin{aligned}
 m(1,2) &= \min(m[0,1] + 1, m[0,2] + 1, m[1,1] + 1) \\
 &= \min(2, 3, 1) \\
 &= \min 1
 \end{aligned}$$

$$\begin{aligned}
 m(1,3) &= \min(m[0,2] + 1, m[0,3] + 1, m[1,2] + 1) \\
 &= \min(3, 4, 2) \\
 &= \min 2
 \end{aligned}$$

$$\begin{aligned}
 m(1,4) &= \min(m[0,3] + 1, m[0,4] + 1, m[1,3] + 1) \\
 &= \min(4, 5, 3) \\
 &= \min 3
 \end{aligned}$$

$$\begin{aligned}
 m(1,5) &= \min(m[0,4] + 1, m[0,5] + 1, m[1,4] + 1) \\
 &= \min(5, 6, 4) \\
 &= \min 4
 \end{aligned}$$

$$\begin{aligned}
 m(2,1) &= \min(m[1,0] + 1, m[1,1] + 1, m[2,0] + 1) \\
 &= \min(2, 1, 3) \\
 &= \min 1
 \end{aligned}$$

$$\begin{aligned}
 m(2,2) &= \min(m[1,1] + 0, m[1,2] + 1, m[2,1] + 1) \\
 &= \min(0, 2, 2) \\
 &= \min 0
 \end{aligned}$$

$$\begin{aligned}
 m(2,3) &= \min(m[1,2] + 1, m[1,3] + 1, m[2,2] + 1) \\
 &= \min(2, 3, 1) \\
 &= \min 1
 \end{aligned}$$

$$\begin{aligned}
 m(2,4) &= \min(m[1,3] + 1, m[1,4] + 1, m[2,3] + 1) \\
 &= \min(3, 4, 2) \\
 &= \min 2
 \end{aligned}$$

$$\begin{aligned}
 m(2,5) &= \min(m[1,4] + 1, m[1,5] + 1, m[2,4] + 1) \\
 &= \min(4, 5, 3) \\
 &= \min 3
 \end{aligned}$$

$$\begin{aligned}
 m(3,1) &= \min(m[2,0] + 1, m[2,1] + 1, m[3,0] + 1) \\
 &= \min(3, 2, 4) \\
 &= \min 2
 \end{aligned}$$

$$\begin{aligned}
 m(3,2) &= \min(m[2,1] + 1, m[2,2] + 1, m[3,1] + 1) \\
 &= \min(2, 1, 3)
 \end{aligned}$$

$$\begin{aligned}
 &= \min 1 \\
 m(3,3) &= \min (m[2,2] + 1, m[2,3] + 1, m[3,2] + 1) \\
 &= \min (1, 2, 2) \\
 &= \min 1 \\
 m(3,4) &= \min (m[2,3] + 1, m[2,4] + 1, m[3,3] + 1) \\
 &= \min (2, 3, 2) \\
 &= \min 2 \\
 m(3,5) &= \min (m[2,4] + 1, m[2,5] + 1, m[3,4] + 1) \\
 &= \min (3, 4, 3) \\
 &= \min 3 \\
 m(4,1) &= \min (m[3,0] + 1, m[3,1] + 1, m[4,0] + 1) \\
 &= \min (4, 3, 5) \\
 &= \min 3 \\
 m(4,2) &= \min (m[3,1] + 1, m[3,2] + 1, m[4,1] + 1) \\
 &= \min (3, 2, 4) \\
 &= \min 2 \\
 m(4,3) &= \min (m[3,2] + 1, m[3,3] + 1, m[4,2] + 1) \\
 &= \min (2, 2, 3) \\
 &= \min 3 \\
 m(4,4) &= \min (m[3,3] + 0, m[3,4] + 1, m[4,3] + 1) \\
 &= \min (1, 3, 4) \\
 &= \min 1 \\
 m(4,5) &= \min (m[3,4] + 1, m[3,5] + 1, m[4,4] + 1) \\
 &= \min (3, 4, 2) \\
 &= \min 2 \\
 m(5,1) &= \min (m[4,0] + 1, m[4,1] + 1, m[5,0] + 1) \\
 &= \min (5, 4, 6) \\
 &= \min 4 \\
 m(5,2) &= \min (m[4,1] + 1, m[4,2] + 1, m[5,1] + 1) \\
 &= \min (4, 3, 5) \\
 &= \min 3 \\
 m(5,3) &= \min (m[4,2] + 1, m[4,3] + 1, m[5,2] + 1)
 \end{aligned}$$

$$\begin{aligned}
 &= \min (3, 4, 4) \\
 &= \min 3 \\
 m(5,4) &= \min (m[4,3] + 1, m[4,4] + 1, m[5,3] + 1) \\
 &= \min (4, 2, 4) \\
 &= \min 2 \\
 m(5,5) &= \min (m[4,4] + 0, m[4,5] + 1, m[5,4] + 1) \\
 &= \min (1, 3, 3) \\
 &= \min 1
 \end{aligned}$$

Tabel 3.11 Hasil Perhitungan Matriks GHAIB

		G	H	O	I	B
	0	1	2	3	4	5
G	1	0	1	2	3	4
H	2	1	0	1	2	3
A	3	2	1	1	2	3
I	4	3	2	2	1	2
B	5	4	3	3	2	1

Mengubah string GHAIB menjadi GHOIB dibutuhkan 1 Operasi yaitu:

1. Substitution A pada GHAIB

GHAIB menjadi GHOIB

Menghitung similarity kata GHAIB.

$$\text{similarity} = \left(1 - \frac{1}{5}\right) \times 100\% = \frac{4}{5} \times 100\% = 80\%$$

Membandingkan kata GHOIB dengan kata GAIB. Membuat matriks untuk membandingkan dua kata tersebut. Matriks yang dimaksud adalah sebagai berikut:



Table 3.12 Operasi Matriks GAIB

		G	H	O	I	B
	0	1	2	3	4	5
G	1					
A	2					
I	3					
B	4					

$$m(i,j) = \min (m[i, j] + \text{cost}, m[i-1, j] + 1, m[i, j-1] + 1)$$

$$m(0,0) = 0 \quad m(1,0) = 1$$

$$m(0,1) = 1 \quad m(2,0) = 2$$

$$m(0,2) = 2 \quad m(3,0) = 3$$

$$m(0,3) = 3 \quad m(4,0) = 4$$

$$m(0,4) = 4$$

$$m(0,5) = 5$$

$$m(1,1) = \min (m[0,0] + 0, m[0,1] + 1, m[1,0] + 1)$$

$$= \min (0, 2, 2)$$

$$= \min 0$$

$$m(1,2) = \min (m[0,1] + 1, m[0,2] + 1, m[1,1] + 1)$$

$$= \min (2, 3, 1)$$

$$= \min 1$$

$$m(1,3) = \min (m[0,2] + 1, m[0,3] + 1, m[1,2] + 1)$$

$$= \min (3, 4, 2)$$

$$= \min 2$$

$$m(1,4) = \min (m[0,3] + 1, m[0,4] + 1, m[1,3] + 1)$$

$$= \min (4, 5, 3)$$

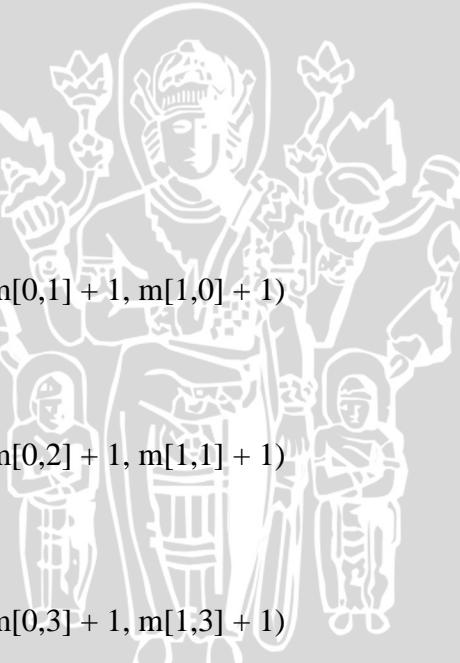
$$= \min 3$$

$$m(1,5) = \min (m[0,4] + 1, m[0,5] + 1, m[1,4] + 1)$$

$$= \min (5, 6, 4)$$

$$= \min 4$$

$$m(2,1) = \min (m[1,0] + 1, m[1,1] + 1, m[2,0] + 1)$$



$= \min (2, 1, 3)$
 $= \min 1$
 $m(2,2) = \min (m[1,1] + 1, m[1,2] + 1, m[2,1] + 1)$
 $= \min (1, 2, 2)$
 $= \min 1$
 $m(2,3) = \min (m[1,2] + 1, m[1,3] + 1, m[2,2] + 1)$
 $= \min (2, 3, 2)$
 $= \min 2$
 $m(2,4) = \min (m[1,3] + 1, m[1,4] + 1, m[2,3] + 1)$
 $= \min (3, 4, 3)$
 $= \min 3$
 $m(2,5) = \min (m[1,4] + 1, m[1,5] + 1, m[2,4] + 1)$
 $= \min (4, 5, 4)$
 $= \min 4$
 $m(3,1) = \min (m[2,0] + 1, m[2,1] + 1, m[3,0] + 1)$
 $= \min (3, 2, 4)$
 $= \min 2$
 $m(3,2) = \min (m[2,1] + 1, m[2,2] + 1, m[3,1] + 1)$
 $= \min (2, 2, 3)$
 $= \min 2$
 $m(3,3) = \min (m[2,2] + 1, m[2,3] + 1, m[3,2] + 1)$
 $= \min (2, 3, 3)$
 $= \min 2$
 $m(3,4) = \min (m[2,3] + 0, m[2,4] + 1, m[3,3] + 1)$
 $= \min (2, 4, 3)$
 $= \min 2$
 $m(3,5) = \min (m[2,4] + 1, m[2,5] + 1, m[3,4] + 1)$
 $= \min (4, 5, 3)$
 $= \min 3$
 $m(4,1) = \min (m[3,0] + 1, m[3,1] + 1, m[4,0] + 1)$
 $= \min (4, 3, 5)$
 $= \min 3$

$$m(4,2) = \min (m[3,1] + 1, m[3,2] + 1, m[4,1] + 1)$$

$$= \min (3, 3, 4)$$

$$= \min 3$$

$$m(4,3) = \min (m[3,2] + 1, m[3,3] + 1, m[4,2] + 1)$$

$$= \min (3, 3, 4)$$

$$= \min 3$$

$$m(4,4) = \min (m[3,3] + 1, m[3,4] + 1, m[4,3] + 1)$$

$$= \min (3, 3, 4)$$

$$= \min 3$$

$$m(4,5) = \min (m[3,4] + 0, m[3,5] + 1, m[4,4] + 1)$$

$$= \min (2, 4, 4)$$

$$= \min 3$$

Tabel 3.13 Hasil Perhitungan Matriks GAIB

		G	H	O	I	B
	0	1	2	3	4	5
G	1	0	1	2	3	4
A	2	1	1	2	3	4
I	3	2	2	2	2	3
B	4	3	3	3	3	2

Mengubah string GAIB menjadi GHOIB membutuhkan 2 operasi yaitu :

1. Substitution A menjadi H

GAIB menjadi GHIB

2. Insertion O

GHIB menjadi GHOIB

Menghitung similarity kata GAIB

$$\text{similarity} = \left(1 - \frac{2}{5}\right) \times 100\% = \frac{3}{5} \times 100\% = 60\%$$



Membandingkan kata GHOIB dengan kata GHOIB, Membuat matriks untuk membandingkan dua kata tersebut. Matriks yang dimaksud adalah sebagai berikut:

Tabel 3.14 Operasi Matriks GHOIB

		G	H	O	I	B
	0	1	2	3	4	5
G	1					
H	2					
O	3					
I	4					
B	5					

$$m(i,j) = \min (m[i, j] + \text{cost}, m[i-1, j] + 1, m[i, j-1] + 1)$$

$$m(0,0) = 0 \quad m(1,0) = 1$$

$$m(0,1) = 1 \quad m(2,0) = 2$$

$$m(0,2) = 2 \quad m(3,0) = 3$$

$$m(0,3) = 3 \quad m(4,0) = 4$$

$$m(0,4) = 4 \quad m(5,0) = 5$$

$$m(0,5) = 5$$

$$m(1,1) = \min (m[0,0] + 0, m[0,1] + 1, m[1,0] + 1)$$

$$= \min (0, 2, 2)$$

$$= \min 0$$

$$m(1,2) = \min (m[0,1] + 1, m[0,2] + 1, m[1,1] + 1)$$

$$= \min (2, 3, 1)$$

$$= \min 1$$

$$m(1,3) = \min (m[0,2] + 1, m[0,3] + 1, m[1,2] + 1)$$

$$= \min (3, 4, 2)$$

$$= \min 2$$

$$m(1,4) = \min (m[0,3] + 1, m[0,4] + 1, m[1,3] + 1)$$

$$= \min (4, 5, 3)$$

$$= \min 3$$



$$m(1,5) = \min (m[0,4] + 1, m[0,3] + 1, m[1,4] + 1)$$

$$= \min (5, 6, 4)$$

$$= \min 4$$

$$m(2,1) = \min (m[1,0] + 1, m[1,1] + 1, m[2,0] + 1)$$

$$= \min (2, 1, 3)$$

$$= \min 1$$

$$m(2,2) = \min (m[1,1] + 0, m[1,2] + 1, m[2,1] + 1)$$

$$= \min (0, 2, 2)$$

$$= \min 0$$

$$m(2,3) = \min (m[1,2] + 1, m[1,1] + 1, m[2,0] + 1)$$

$$= \min (2, 1, 3)$$

$$= \min 1$$

$$m(2,4) = \min (m[1,3] + 1, m[1,4] + 1, m[2,3] + 1)$$

$$= \min (3, 4, 2)$$

$$= \min 2$$

$$m(2,5) = \min (m[1,4] + 1, m[1,5] + 1, m[2,4] + 1)$$

$$= \min (4, 5, 3)$$

$$= \min 3$$

$$m(3,1) = \min (m[2,0] + 1, m[2,1] + 1, m[3,0] + 1)$$

$$= \min (3, 2, 4)$$

$$= \min 2$$

$$m(3,2) = \min (m[2,1] + 1, m[2,2] + 1, m[3,1] + 1)$$

$$= \min (2, 1, 3)$$

$$= \min 1$$

$$m(3,3) = \min (m[2,2] + 0, m[2,3] + 1, m[3,3] + 1)$$

$$= \min (0, 2, 2)$$

$$= \min 0$$

$$m(3,4) = \min (m[2,3] + 1, m[2,4] + 1, m[3,3] + 1)$$

$$= \min (2, 3, 1)$$

$$= \min 1$$

$$m(3,5) = \min (m[2,4] + 1, m[2,5] + 1, m[3,4] + 1)$$

$$= \min (3, 4, 2)$$

$$\begin{aligned}
 &= \min 2 \\
 m(4,1) &= \min (m[3,0] + 1, m[3,1] + 1, m[4,0] + 1) \\
 &= \min (4, 3, 5) \\
 &= \min 3 \\
 m(4,2) &= \min (m[3,1] + 1, m[3,2] + 1, m[4,1] + 1) \\
 &= \min (3, 2, 4) \\
 &= \min 3 \\
 m(4,3) &= \min (m[3,2] + 1, m[3,3] + 1, m[4,2] + 1) \\
 &= \min (2, 1, 3) \\
 &= \min 1 \\
 m(4,4) &= \min (m[3,3] + 0, m[3,4] + 1, m[4,3] + 1) \\
 &= \min (0, 2, 2) \\
 &= \min 0 \\
 m(4,5) &= \min (m[3,4] + 1, m[3,5] + 1, m[4,4] + 1) \\
 &= \min (2, 3, 1) \\
 &= \min 1 \\
 m(5,1) &= \min (m[4,0] + 1, m[4,1] + 1, m[5,0] + 1) \\
 &= \min (5, 4, 5) \\
 &= \min 4 \\
 m(5,2) &= \min (m[4,1] + 1, m[4,2] + 1, m[5,1] + 1) \\
 &= \min (4, 3, 5) \\
 &= \min 3 \\
 m(5,3) &= \min (m[4,2] + 1, m[4,3] + 1, m[5,2] + 1) \\
 &= \min (3, 2, 4) \\
 &= \min 2 \\
 m(5,4) &= \min (m[4,3] + 1, m[4,4] + 1, m[5,3] + 1) \\
 &= \min (2, 1, 3) \\
 &= \min 1 \\
 m(5,5) &= \min (m[4,4] + 0, m[4,5] + 1, m[5,4] + 1) \\
 &= \min (0, 2, 2) \\
 &= \min 0
 \end{aligned}$$

Tabel 3.15 Hasil Perhitungan Matriks GHOIB

		G	H	O	I	B
	0	1	2	3	4	5
G	1	0	1	2	3	4
H	2	1	0	1	2	3
O	3	2	1	0	1	2
I	4	3	2	1	0	1
B	5	4	3	2	1	0

Membandingkan kata GHOIB dengan kata GHOIB membutuhkan 0 operasi karena kedua kata tersebut sudah sama.

$$\text{similarity} = \left(1 - \frac{0}{5}\right) \times 100\% = 1 \times 100\% = 100\%$$

3.10 Skenario Ujicoba

Skenario pertama yang akan diujikan adalah menggunakan satu kata dengan berbagai macam perubahan pada setiap hurufnya, kemudian ditentukan persentase kemiripannya. Seperti yang dijelaskan pada tabel dibawah ini.

Tabel 3.16 Skenario Ujicoba

Kata Pertama	Kata Kedua	Panjang Kata	Persentase Kemiripan

Skenario kedua yang akan diujikan menggunakan dua buah kata dan panjang yang berbeda, kemudian ditentukan persentase kemiripannya.

Tabel 3.17 Skenario Ujicoba

Kata Pertama	Panjang Kata	Kata Kedua	Panjang Kata	Persentase Kemiripan

BAB IV

IMPLEMENTASI

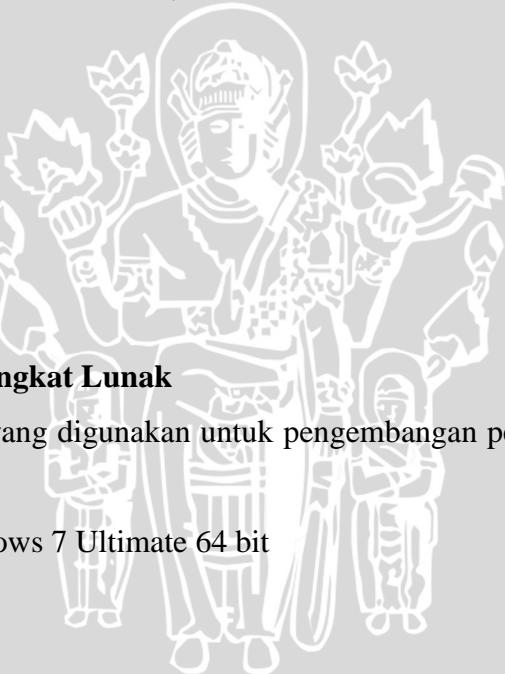
4.1 Lingkungan Implementasi

Lingkungan Implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan perangkat lunak ini adalah sebagai berikut :

1. Intel® Core™ i3-2230M CPU @ 2,20GHz
2. Memori 2 GB
3. Harddisk 500 GB
4. Monitor 14”
5. Keyboard
6. Mouse



4.1.2 Lingkungan Perangkat Lunak

Perangkat Lunak yang digunakan untuk pengembangan perangkat lunak ini adalah sebagai berikut :

1. Sistem Operasi Windows 7 Ultimate 64 bit
2. XAMPP 3.1.0 beta 6
3. Apache 2.4.3
4. MySQL 5.5.27
5. CodeLobster 4.5.3
6. PhpMyAdmin 3.5.2.2

4.2 Implementasi Program

Berdasarkan analisa dan perancangan proses pada bab 3, maka pada sub bab ini akan dijelaskan implementasi mengenai proses-proses tersebut.

4.2.1 Form Registrasi

Sebelum user menggunakan aplikasi Al-Quran Digital, user harus register dahulu. Dengan adanya fitur register ini, user mendapatkan akses untuk memberikan catatan atau menambahkan tulisan arab latin kedalam Al-Quran Digital. *Source Code* untuk form register akan ditunjukkan pada *Source Code 4.1*.

```
<?php $menu = 'register'; ?>
<?php include "header.php"; ?>

<div class="main" style="width: 960px;">
    <div class="row">
        <fieldset>
            <legend><i class="icon-user" style="margin-top: 7px;"></i> Register</legend>
            <?php if (isset($_GET['err'])) { ?>
                <div class="alert"><?php echo $_GET['err']; ?></div>
            <?php } ?>
            <?php $inputs = isset($_SESSION['inputs']) ? $_SESSION['inputs'] : array(); ?>
            <form action="proses_register.php" method="POST" class="form-horizontal">
                <div class="control-group">
                    <label class="control-label">Nama</label>
                    <div class="controls">
                        <input type="text" name="nama" maxlength="64" placeholder="Nama Lengkap" value="<?php echo isset($inputs['nama']) ? $inputs['nama'] : ''; ?>">
                    </div>
                </div>
                <div class="control-group">
                    <label class="control-label">Username</label>
                    <div class="controls">
                        <input type="text" name="username" maxlength="15" placeholder="Username" value="<?php echo isset($inputs['username']) ? $inputs['username'] : ''; ?>">
                    </div>
                </div>
                <div class="control-group">
                    <label class="control-label">Password</label>
                    <div class="controls">
                        <input type="password" name="password" maxlength="15" placeholder="Password/Kata Sandi">
                    </div>
                </div>
                <div class="control-group">
                    <label class="control-label">Ulangi Password</label>
                    <div class="controls">
                        <input type="password" name="ulangi_password">
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>
```



```
maxlength="15" placeholder="Ulangi Password/Kata Sandi">
    </div>
</div>
<div class="form-actions">
    <input type="submit" class="btn btn-primary"
name="register" value="Daftar">
</div>
</form>
<?php if (isset($_SESSION['inputs'])) {
unset($_SESSION['inputs']); } ?>
    </fieldset>
        </div>
    </div>
</body>
</html>
```

Source Code 4.1 Form Register.php

Form registrasi juga membutuhkan form proses_registrasi untuk menyimpan data register user kedalam database users. Form Proses_Registrasi akan ditunjukkan pada *Source Code 4.2* berikut ini.

```
<?php
    require "mysql.php";
    @session_start();

    if (isset($_POST['register'])) {

        $_SESSION['inputs'] = array();

        $nama = trim($_POST['nama']);
        $username =
trim(mysql_real_escape_string($_POST['username']));
        $password = trim($_POST['password']);
        $ulangi_password =
trim($_POST['ulangi_password']);

        $_SESSION['inputs']['nama'] = $nama;
        $_SESSION['inputs']['username'] = $username;

        if ($nama == '' || $username == '' || $password
== '' || $ulangi_password == '') {
            header('location: register.php?err=Terdapat
field yang masih kosong.');
            exit;
        }

        $query = "select * from users where username =
'$username'";
        $res = mysql_query($query);

        if (mysql_num_rows($res) == 0) {
```

```

        if ($password != $ulangi_password) {
            header('location:
register.php?err=Password tidak sama.');?>
        }
        else {
            $password = md5($password);

            $query = "insert into users values('',
'$username', '$password', '$nama')";
            $res = mysql_query($query);

            if ($res) {
                header('location:
login.php?err=Registrasi berhasil. Silakan Login melalui
form di bawah ini.');?>
            }
            else {
                header('location:
register.php?err=Registrasi Gagal.');?>
            }
        }
        else {
            header('location: register.php?err=Username
sudah terdaftar. Silakan pilih yg lain.');?>
        }
    }
}

```

Source Code 4.2 Form Proses_registrasi.php

4.2.2 Form Login

Setelah proses register selesai, user dapat melakukan login ke dalam aplikasi dengan menggunakan username dan password yang sebelumnya sudah didaftarkan. Dalam proses login ini, diperlukan 2 form yaitu login.php dan proses_login.php. *Source Code* untuk Login akan ditunjukkan pada *Source Code* 4.3.

```

<?php $menu = 'login'; ?>
<?php include "header.php"; ?>

        <div class="main" style="width: 90%; ">
            <div class="row text-center">
                <h3>LOGIN PENGGUNA</h3>
                <?php if (isset($_GET['err'])) { ?>
                    <h5 style="color: red;"><?php echo
$_GET['err']; ?></h5>
                <?php } ?>

```



```

<form action="proses_login.php"
method="POST" class="form-inline" style="margin-top: 50px;">
    <?php if (isset($_GET['next'])) {
        <input type="hidden" name="next"
value="<?php echo $_GET['next']; ?>">
        <?php } ?>
    <input type="text" name="user"
placeholder="Username">
    <input type="password"
name="password" placeholder="Password">
    <input type="submit" class="btn
btn-primary" name="login" value="Login">
</form>
</div>
</div>
</body>
</html>

```

Source Code 4.3 Form Login.php

Form proses_login digunakan untuk memverifikasi username dan password yang sudah terdaftar pada saat register user. Jika username dan password sesuai dengan yang sudah terdaftar maka user akan diredirect ke halaman home, jika tidak sama maka akan kembali lagi ke halaman login. Form proses_login akan ditunjukkan pada *Source Code 4.4*.

```

<?php
    require "mysql.php";
    @session_start();

    if (isset($_POST['login']) && !empty($_POST['user'])
&& !empty($_POST['password'])) {

        $user = mysql_real_escape_string($_POST['user']);
        $pass = md5($_POST['password']);

        $query = "select * from users where username =
'$user' and password = '$pass'";
        $res = mysql_query($query);

        $next = 'index.php';
        if (isset($_POST['next'])) {
            $next = $_POST['next'];
        }

        if (mysql_num_rows($res) == 1) {
            $user = mysql_fetch_assoc($res);

```



```

$_SESSION['user_id'] = $user['id'];
$_SESSION['username'] = $user['username'];
$_SESSION['nama'] = $user['nama'];

header('location: '.$next); exit;
}
else {
    header('location:
login.php?next='.$next.'&err=User atau password salah.');
exit;
}
}

```

Source Code 4.4 Form Proses_login.php

4.2.3 Proses Pencarian dan Menampilkan Hasil Kata

Pada proses ini akan dijelaskan mengenai jalannya proses pencarian kata dan menampilkan hasil dari proses pencarinya. Berikut ini adalah *Source Code* prosesnya.

```

<?php
require "mysql.php";

require "Levenshtein.php";

$similiarWords = array();
$exactWord = array();
$jml_page = 0;
$page = 0;

if (isset($_GET['keyword']) && isset($_GET['page']) &&
count($_SESSION['exact']) > 0) {

$field = $_GET['field'];
if ($field == 'arti') {
$tabel = 'quran_text';
$col = 'AyahText';
}
else if ($field == 'arab_latin') {
$tabel = 'arab_latin';
$col = 'text';
}
$keyword = $_GET['keyword'];
$page = $_GET['page'];
$ayat_per_page = 10;
$semua_record = count($_SESSION['exact'][$field][

```




```

$_SESSION['exact'][$f][$keyword] = explode(',', 
$data['id_ayat']);
header('location:
index.php?keyword='.$keyword.'&field='.$field.'&page=1');
exit;
}

else {

$query      = "select * from ".$tabel;
$res        = mysql_query($query);

while ($data = mysql_fetch_assoc($res)) {
$ayat = explode(' ', $data[$col]);
foreach($ayat as $word) {

$word = str_replace(array(',', '.', '(', ')', ':',
';', "'", '!', '?'), '', $word);
$word = strtolower($word);
$lvDistance = levenshteinDistance($keyword, $word);

$similiarity = (1 - $lvDistance / max(
strlen($keyword), strlen($word) ) ) * 100;

if ($similiarity == 100.0) {
$_SESSION['exact'][$field][$keyword][] = $data[
$index ];
}
else if ($similiarity >= 75.0) {

$similiarWords['IDs'][$word][] = $data[ $index ];

$similiarWords['similiarity'][$word] =
round($similiarity, 2);
}
}
}

if (count($_SESSION['exact']) > 0) {

header('location:
index.php?keyword='.$keyword.'&field='.$field.'&page=1');
exit;
}
if (isset($similiarWords['similiarity']) > 0)

arsort($similiarWords['similiarity']);
}
}
else {
echo '<strong style="color: red;">Keyword masih

```



```

kosong!</strong> ' ;
        }
    }
else
?>

<?php if (isset($similiarWords['similiarity'])) { ?>
<div class="row suggest">Mungkin maksud Anda :
<?php foreach($similiarWords['similiarity'] as $word =>
$percentage) { $ids =
array_splice($similiarWords['IDs'][$word], 0, 30); ?>
<strong><a href="index.php?keyword=<?php echo $word;
?>&field=<?php echo $field; ?>&page=1"><?php echo $word.''.
('. $percentage.'%); ?></a></strong>,
<?php } ?>
</div><!-- end hasil search -->
<?php } else if (isset($_GET['keyword']) &&
!isset($_GET['IDs']) && !isset($search_result) &&
count($similiarWords) == 0) { echo 'Tidak ada data.'; } ?>
<div class="row">
    <div class="well">
        <?php
        if (isset($search_result)) {
            while ($data = mysql_fetch_assoc($res)) {
?>

            <h4><?php echo 'Surat : '.$data['Nama']. ' /
Ayat : '.$data['VerseID']; ?></h4>
            <p class="text-right"></p>
            <h5>Arab Latin</h5>
            <p class="desc"><?php echo
str_ireplace($keyword, '<strong style="color:
orange;">'.$keyword.'</strong>', $data['text_arab_latin']);
?></p>
            <h5>Arti</h5>
            <p class="desc">
<?php
                echo str_ireplace($keyword, '<strong
style="color: orange;">'.$keyword.'</strong>',
$data['AyahText']);
            ?>
            </p>
            <hr>
            <?php
        } ?>
    </div>
</div>

<?php if ($jml_page > 1) { ?>
<!-- page -->
<div class="pagination">

```



```

<ul>
    <li class="disabled"><a href="#">Halaman &raquo;</a></li>
    <?php for($i = 1; $i <= $jml_page; $i++) {
?
    <li <?php echo ($page == $i) ?
'class="disabled"' : ''; ?><a href="index.php?keyword=<?php
echo $keyword; ?>&field=<?php echo $field; ?>&page=<?php
echo $i; ?>"><?php echo $i; ?></a></li>
    <?php } ?>
    </ul>
</div>
<?php } ?>
```

Source Code 4.5 Pencarian dan Menampilkan hasil kata

Di dalam *Class Levenshtein.php* menggunakan 2 Algoritma levenshtein dengan kegunaan yang berbeda. Pada Levenshtein pertama, digunakan untuk pencarian nilai akhir dari sebuah kata. Sedangkan untuk Levenshtein yang kedua digunakan untuk menampilkan hasil hitungan dalam bentuk tabel. *Source Code* levenshtein akan ditunjukkan pada *Source Code 4.6* dan *Source Code 4.7*.

```

<?php
function levenshteinDistance($str1, $str2) {

    $cost = 0;
    $m    = array();

    $str1 = strtolower($str1);
    $str2 = strtolower($str2);

    for($i = 0; $i <= strlen($str1); $i++) {

        for($j = 0; $j <= strlen($str2); $j++) {

            if ($i == 0)
                $m[$i][$j] = $j;

            else if ($j == 0)
                $m[$i][$j] = $i;
            else {

                if ($str1[$i - 1] == $str2[$j - 1])
                    $cost = 0;
                else
                    $cost = 1;

                $m[$i][$j] = min(
                    $m[$i - 1][$j - 1] + $cost,
```



```

        $m[$i - 1][$j] + 1,
        $m[$i][$j - 1] + 1
    );
}
}

return $m[ strlen($str1) ][ strlen($str2) ];
}

```

Source Code 4.6 Levenshtein Distance

Berikut ini adalah *Source Code* levenshteinDistance2.

```

function levenshteinDistance2($str1, $str2) {
    $cost = 0;
    $m = array();

    $str1 = strtolower($str1);
    $str2 = strtolower($str2);

    for($i = 0; $i <= strlen($str2); $i++) {
        for($j = 0; $j <= strlen($str1); $j++) {
            if ($i == 0)
                $m[$i][$j] = $j;

            else if ($j == 0)
                $m[$i][$j] = $i;
            else {
                if ($str2[$i - 1] == $str1[$j - 1])
                    $cost = 0;
                else
                    $cost = 1;

                $m[$i][$j] = min(
                    $m[$i - 1][$j-1] + $cost,
                    $m[$i - 1][$j] + 1,
                    $m[$i][$j - 1] + 1
                );
            }
        }
    }

    return $m;
}

```

Source Code 4.7 LevenshteinDistance2



LevenshteinDistance2 digunakan dalam proses hasil_perhitungan.php, yaitu untuk menampilkan hasil perhitungan dalam bentuk tabel. *Source Code* hasil_perhitungan dapat dilihat dibawah ini.

```
<?php
    require "mysql.php";
    require "Levenshtein.php";
    $similiarWords = array();
    $jml_page = 0;
    $page = 0;

    if (isset($_GET['keyword']) &&
    isset($_GET['similiars'])) {
        $keyword = $_GET['keyword'];
        $similiars = $_GET['similiars'];

        $similiars = explode(',', $similiars);

        foreach($similiars as $word) {
            $lvDistance = levenshteinDistance($keyword,
$word);
            $matriks = levenshteinDistance2($keyword,
$word);

            echo 'Perhitungan untuk keyword:
<strong>' . $keyword . '</strong> dan kata:
<strong>' . $word . '</strong><br>';
            echo '<table class="table table-bordered"
style="width: 500px; margin-bottom: 20px;">';

            for($i = -2; $i < strlen($word); $i++) {
                echo '<tr>';
                for($j = -2; $j < strlen($keyword); $j++) {

                    if ($i == -2 && $j == -2) {
                        echo '<td>&ampnbsp</td>';
                    }
                    else if ($i == -2 && $j == -1) {
                        echo '<td>&ampnbsp</td>';
                    }
                    else if ($i == -2) {
                        echo '<td style="font-weight:
bold;">' . $keyword[$j] . '</td>';
                    }

                    if ($i == -1 && $j == -2) {
                        echo '<td>&ampnbsp</td>';
                    }
                }
            }
        }
    }
}
```



```
        else if ($i > -2 && $j == -2) {
            echo '<td style="font-weight:
bold;">' . $word[$i] . '</td>';
        }
        else if ($i > -2 && $j > -2) {
            $bg = '';
            if ($i+1 == strlen($word) && $j+1 ==
strlen($keyword))
                $bg = 'style="background: green; color:
white;"';
            echo '<td ' . $bg . '>' . $matriks[$i+1][$j+1] . '</td>';
        }
        echo '</tr>';
    }
echo '</table>';

echo '
<pre>
Similarity = (1 - levenshteinDistance('. $keyword .',
'. $word .') / max(length('. $keyword .'), length('. $word .')) * 100
            = (1 - '. $lvDistance .') /
max('. strlen($keyword) .', '. strlen($word) .') * 100
            = (1 - '. $lvDistance .') / '. max(strlen($keyword),
strlen($word)) .') * 100
            = (1 - '. ($lvDistance / max(strlen($keyword),
strlen($word))) .') * 100
            = '. round((1 - $lvDistance /
max(strlen($keyword), strlen($word))) * 100, 2). ' %
</pre>';
    }
?
?>
```

Source Code 4.8 Proses hasil_perhitungan.php

4.2.4 Form Daftar Surat Al-Quran

Pada Form daftar surat akan ditampilkan nama surat-surat dan jumlah ayat dalam Al-Quran. Untuk pemanggilan surat dan ayat Al-Quran akan dibutuhkan juga Form daftar_ayat.php. *Source Code* untuk menampilkannya dapat dilihat pada *Source Code* 4.9.

```
<?php
require "mysql.php";
$jml_per_tabel = 30;

$query = "select nama_sura.ID, nama_sura.Nama,
```

```

count(quran.index) as jml_ayat from nama_sura join quran on
nama_sura.ID = quran.sura group by nama_sura.ID";
    $res = mysql_query($query);
    $data_surat = array();
    while ($data = mysql_fetch_assoc($res)) {
        $data_surat[ ] = array(
            'sura_ID' => $data['ID'],
            'nama_sura' => $data['Nama'],
            'jml_ayat' => $data['jml_ayat']
        );
    }

    $data_surat = array_chunk($data_surat,
    $jml_per_tabel);
?>

<?php
    foreach($data_surat as $index => $groups) {
?>
        <table class="table table-hover table-bordered"
style="display: inline-block; vertical-align: top; width:
auto; margin-left: 10px;">
        <thead>
        <tr>
            <th width="40" style="text-align: center">No</th>
            <th width="170">Nama Surat</th>
            <th style="text-align: center">Ayat</th>
        </tr>
        </thead>
        <tbody>
        <?php
            $no = $index * $jml_per_tabel + 1;
            foreach($groups as $sura) { ?>
        <tr>
            <td style="text-align: center;"><?php echo $no;
?></td>
            <td><a href=<?php echo
'daftar_ayat.php?id_surat='.$sura['sura_ID'];
?>&nama_surat=<?php echo $sura['nama_sura']; ?>"><?php echo
$sura['nama_sura']; ?></a></td>
            <td style="text-align: center"><?php echo
$sura['jml_ayat']; ?></td>
        </tr>
        <?php
            $no++;
        }
?>

```

Source Code 4.9 Form daftar_surat_alquran.php



Form daftar_ayat digunakan untuk menampilkan isi dari database utama ke halaman daftar_ayat. *Source Code* daftar_ayat.php dapat dilihat pada *Source Code* 4.10.

```
<?php
    require "mysql.php";
    $id_surat = (int)$_GET['id_surat'];
    $nama_surat = $_GET['nama_surat'];

    $jml_page = 1;
    $page = 1;
    if (isset($_GET['page'])) {
        $page = (int)$_GET['page'];
    }

    $per_page = 10;
    $start = floor($page - 1) * $per_page;

    $query = "select quran.index from quran join
    quran_text on quran.index = quran_text.ID where sura =
    $id_surat";
    $res = mysql_query($query);
    $semua_record = mysql_num_rows($res);
    $jml_page = floor($semua_record/$per_page) + 1;

    $query = "
        select quran.index, quran.aya, quran.gambar,
    quran_text.AyahText, arab_latin.text
        from quran
        join quran_text on quran.index =
    quran_text.ID
        left join arab_latin on quran.index =
    arab_latin.index
        where quran.sura = $id_surat
        limit $start, $per_page";
    $res = mysql_query($query);

    $no = $start + 1;
    $user_id = 0;
    if (isset($_SESSION['user_id'])) {
        $user_id = $_SESSION['user_id'];
    }
    while($data = mysql_fetch_assoc($res)) {

        $query = "select * from users_input where user_id =
    $user_id and ayat_id = ".$data['index'];
        $res2 = mysql_query($query);

        $data_user['catatan'] = '';
        $data_user['arab_latin'] = '';
    }
}
```



```

if (mysql_num_rows($res2) == 1) {
    $input = mysql_fetch_assoc($res2);

    $data_user['catatan'] = $input['catatan'];
    $data_user['arab_latin'] = $input['arab_latin'];
}
?>
<tr>
<td style="text-align: left;">
<strong>Ayat #<?php echo $data['aya']; ?></strong><br>
<strong>Arab Latin : </strong><br>
<p style="margin: 0 10px 10px;">
<?php

    if ($data['text'] != '') {
        echo '<strong>Asli:</strong><br>' . $data['text'];
    }
    else {
        echo '-';
    }

    if ($data_user['arab_latin'] != '') {
        echo      ' <br><br><em><small style="color: green;"><strong>User Input/Manual:</strong><br>' . $data_user['arab_latin'] . '</small></em>';
    }
?>
</p>

<strong>Arti : </strong><br>
<p style="margin: 0 10px 10px;">
<?php echo $data['AyahText']; ?>
<?php if ($data_user['catatan'] != '') { ?>
<br>
<br>
<em>

<small style="color: green;">
<strong>Catatan:</strong>
<br>
<?php echo $data_user['catatan']; ?>

</small>
</em>
<?php } ?>

```

```

</p>
<hr>
<?php if (isset($_SESSION['user_id'])) { ?>

    <a href="user_input.php?id_ayat=<?php echo $data['index']; ?>" target="_blank">Update Catatan/Arab Latin</a>
    <?php } else { ?>

        <a href="login.php?next=user_input.php?id_ayat=<?php echo $data['index']; ?>">Silakan login untuk menambahkan catatan/arab latin secara manual.</a>
        <?php } ?>
    </td>
    <td style="text-align: right; width: 50%">
        
    </td>
</tr>
<?php
    $no++;
}
?>

```

Source Code 4.10 Proses daftar_ayat.php

4.2.5 Form Daftar Hadits

Form daftar hadits berfungsi untuk menampilkan daftar hadits dalam bentuk tabel-tabel. *Source Code* dari daftar hadits dapat dilihat pada *Source Code* 4.11.

```

<?php
    require "mysql.php";
    $query = "select * from hadits order by ID limit 0,
19";
    $res = mysql_query($query);

    $no = 1;
    while ($data = mysql_fetch_assoc($res)) {
    ?>
    <tr>
        <td style="text-align: center;"><?php echo $no;
    ?></td>
        <td style="text-align: center;"><a href="detail_hadits.php?id=<?php echo $data['ID']; ?>"><?php echo $data['nama_hadits']; ?></a></td>
    </tr>
<?php
    $no++;
}

```



```
?>
</tbody>
</table>

    <table class="table table-bordered table-hover pull-right" style="width: 43%">
<thead>
    <tr>
        <th>No</th>
        <th style="text-align: center;">Nama Hadits</th>
    </tr>
</thead>
<tbody>
<?php
    require "mysql.php";
    $query = "select * from hadits order by ID limit 19, 19";
    $res = mysql_query($query);
    $no = 20;
    while ($data = mysql_fetch_assoc($res)) {
?
    <tr>
        <td style="text-align: center;"><?php echo $no;
?></td>
        <td style="text-align: center;"><a href="detail_hadits.php?id=<?php echo $data['ID']; ?>"><?php echo $data['nama_hadits']; ?></a></td>
    </tr>
<?php
    $no++;
}
?>
```

Source Code 4.11 Form daftar_hadits.php

Form daftar_hadits.php juga memerlukan Form detail_hadits.php untuk bisa menampilkan isi dari hadits-hadits tersebut. *Source Code* detail_hadits dapat dilihat pada *Source Code* 4.12 berikut ini.

```
<?php
    require "mysql.php";

    if (isset($_GET['id'])) {
        $id = $_GET['id'];
        $query = "select * from hadits where ID = ".$id;
        mysql_query('set names utf8');
        $res = mysql_query($query);

        $data = mysql_fetch_assoc($res);
   ?>
```



```

<h4>DETAIL HADITS</h4><br>
<h5>Nama Hadits : <?php echo $data['nama_hadits']; ?></h5>
<hr>
<div class="row">
    <div class="well">
        <p class="text-right" style="font-size: 18pt; line-height: 50px;"><?php echo $data['sura']; ?></p>
        <h5>Arti</h5>
        <p class="desc"><?php echo $data['arti']; ?></p>
    </div>
</div>
<?php
}
?>

```

Source Code 4.12 Proses detail_hadits.php

4.2.6 Form User Input dan Update

Pada aplikasi ini user dapat menambahkan catatan dan arab latin kedalam Al-Quran Digital, hal ini bertujuan agar user dapat lebih mudah memahami isi dari Al-Quran. Form yang digunakan adalah Form user_input.php sebagai tampilannya. Berikut ini adalah *Source Code* dari user_input.php dan proses_user_input.php.

```

<?php
    require "mysql.php";
    if (isset($_GET['id_ayat'])) {
        $id_ayat = (int)$_GET['id_ayat'];

        $query = "
            select quran.index, nama_sura.Nama, quran.aya, gambar,
            AyahText, arab_latin.text as text_arab_latin
            from quran
            join quran_text on quran.index = quran_text.ID
            join nama_sura on quran.sura = nama_sura.ID
            left join arab_latin on quran.index = arab_latin.index
            where quran.index = $id_ayat";

        mysql_query('set names utf8');
        $res = mysql_query($query);

        while ($data = mysql_fetch_assoc($res)) { ?>
<strong>Surat : <?php echo $data['Nama']; ?> / Ayat : <?php
echo $data['aya']; ?></strong><br>



```



```

<br>
<strong>Arab Latin:</strong><br>
<?php if ($data['text_arab_latin'] != '') { ?>
<p><?php echo $data['text_arab_latin']; ?></p>
    <?php } else { ?>
        <p style="color: red;">Arab Latin tidak tersedia.
Silakan isi secara manual melalui form di bawah ini.</p>
    <?php } ?>

    <?php if ($data['AyahText'] != '') { ?>
<strong>Arti :</strong>
<p><?php echo $data['AyahText']; ?></p>
<?php } ?>

<hr>
<?php
    $user = $_SESSION['user_id'];
    $query = "select * from users_input where user_id =
$user and ayat_id = $id_ayat";
    $res2 = mysql_query($query);
    $catatan = ''; $arab_latin = '';

    if (mysql_num_rows($res2) == 1) {
        $data2 = mysql_fetch_assoc($res2);

        $catatan = $data2['catatan'];
        $arab_latin = $data2['arab_latin'];
    }

?>
    <form action="proses_user_input.php" method="POST">
        <input type="hidden" name="id_ayat" value="<?php echo
$id_ayat; ?>">
        <h4>Input Manual</h4>

        <strong>Catatan:</strong><br>
            <textarea name="catatan" class="span6" rows="5"><?php
echo $catatan; ?></textarea><br>
        <strong>Arab Latin:</strong><br>
        <textarea name="arab_latin" class="span6" rows="5"><?php
echo $arab_latin; ?></textarea>

        <div class="form-action text-center">
            <input type="submit" name="simpan" class="btn btn-
primary" value="Simpan Catatan">
        </div>
    </form>
    <?php
    }
?>

```

Source Code 4.13 Form user_input.php



Form proses_user_input.php digunakan untuk memproses input dan update data. Proses input dan update terjadi dalam satu proses, ketika user input catatan form proses_user_input.php juga akan mengupdate datanya. *Source Code* proses dapat dilihat pada *Source Code* 4.14.

```
<?php
    require "mysql.php";
    @session_start();

    if (isset($_POST['simpan'])) {
        $user = $_SESSION['user_id'];
        $id_ayat = $_POST['id_ayat'];

        $catatan =
        mysql_real_escape_string($_POST['catatan']);
        $arab_latin =
        mysql_real_escape_string($_POST['arab_latin']);

        $query = "select * from users_input where user_id
        = $user and ayat_id = $id_ayat";
        $res = mysql_query($query);

        if (mysql_num_rows($res) == 1) {
            $query = "update users_input set catatan =
            '$catatan', arab_latin = '$arab_latin' where user_id = $user
            and ayat_id = $id_ayat";
        }
        else {
            $query = "insert into users_input
            values('', $user, $id_ayat, '$catatan', '$arab_latin')";
        }
        $res = mysql_query($query);

        if ($res) {

            $query = "select * from quran join
            nama_sura on quran.sura = nama_sura.ID where quran.index =
            $id_ayat";
            $res = mysql_query($query);

            $data = mysql_fetch_assoc($res);

            header('location: daftar_ayat.php?err=Data
            catatan/arab latin berhasil
            disimpan.&id_surat='.$data['sura'].'&nama_surat='.$data['Nam
            a']);
        }
    }
}
```

```
        }
        else {

            $query = "select * from quran join
nama_sura on quran.sura = nama_sura.ID where quran.index =
$id_ayat";
            $res = mysql_query($query);
            $data = mysql_fetch_assoc($res);
            header('location: daftar_ayat.php?err=Tidak
dapat menyimpan data catatan/arab
latin.&id_surat='.$data['ID'].'&nama_surat='.$data['Nama']);
            exit;

        }
    }
```

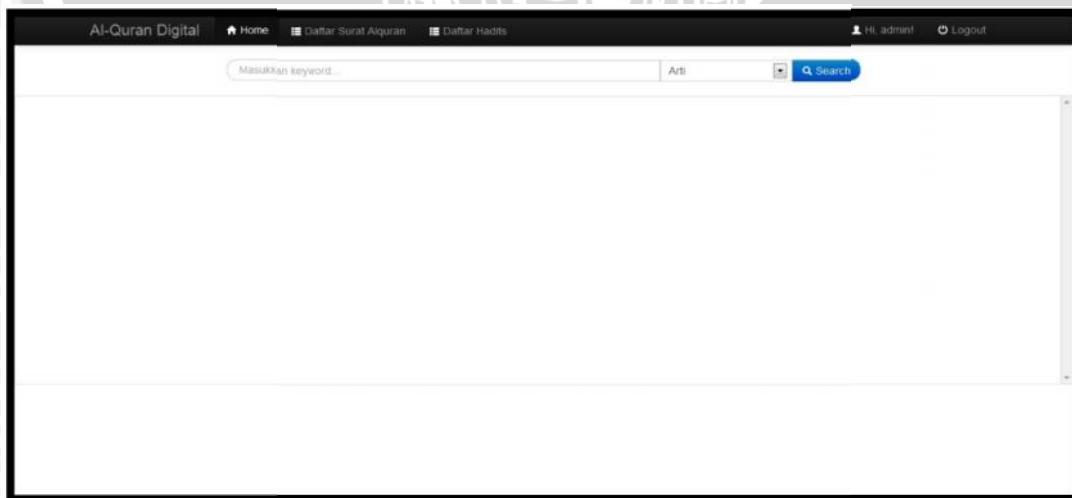
Source Code 4.14 Proses proses_user_input.php

4.3 Implementasi Antarmuka

Implementasi antarmuka pada aplikasi Al-Quran Digital ini terdapat 8 Form yaitu antarmuka Home, Register, Login, Daftar Surat Al-Quran, Daftar Hadits, Hasil Pencarian, Hasil Perhitungan, dan Input Catatan dan Arab Latin.

4.3.1 Antarmuka Home

Antarmuka *Home* merupakan tampilan awal sistem dari aplikasi Al-Quran Digital. Pada antarmuka ini, terdapat beberapa menu seperti Search, Daftar Surat Al-Quran dan Daftar Hadits. Tampilan *Home* ditunjukkan pada gambar 4.1.



Gambar 4.1 Tampilan Home

4.3.2 Antarmuka Register

Antarmuka *Register* adalah form pendaftaran user untuk melakukan login, diantaranya berisi Nama, Username, Password, dan Ulangi password. Gambar berikut ini adalah tampilan dari form register.

The screenshot shows the 'Register' page of the Al-Quran Digital application. At the top, there's a navigation bar with links for Home, Daftar Surat Alquran, Daftar Hadits, Register, and Login. The main content area is titled 'Register' and contains four input fields: 'Nama' (Full Name), 'Username', 'Password', and 'Ulangi Password' (Repeat Password). Below these fields is a blue 'Daftar' (Register) button.

Gambar 4.2 Tampilan Register

4.3.3 Antarmuka Login

Antarmuka login sarana user untuk masuk kedalam aplikasi, dengan cara memasukkan username dan password yang sudah di register sebelumnya. Ditunjukkan pada gambar 4.3 berikut ini.

The screenshot shows the 'LOGIN PENGGUNA' (User Login) page of the Al-Quran Digital application. At the top, there's a navigation bar with links for Home, Daftar Surat Alquran, Daftar Hadits, Register, and Login. The main content area is titled 'LOGIN PENGGUNA' and contains two input fields: 'Username' and 'Password'. Below these fields is a blue 'Login' button.

Gambar 4.3 Tampilan Login

4.3.4 Antarmuka Daftar Surat Al-Quran

Pada antarmuka daftar surat Al-Quran menampilkan daftar surat-surat Al-Quran yang tercantum di dalam Al-Quran beserta jumlah ayat dari masing-masing surat. Ditunjukkan pada gambar 4.4.

No	Nama Surat	Ayat	No	Nama Surat	Ayat	No	Nama Surat	Ayat	No	Nama Surat	Ayat
1	Al Fatihah	7	31	Luqman	34	61	Ash Shaff	14	91	Asy Syams	15
2	Al Baqarah	286	32	As Sajdah	30	62	Al Jumuah	11	92	Al Lail	21
3	Ali Imran	200	33	Al Ahzab	73	63	Al Munafiqun	11	93	Adz Dhuhra	11
4	An Nisa'	176	34	Saba'	54	64	Al Taghabun	18	94	Asy Syarh	8
5	Al Maidah	120	35	Faathir	45	65	Al Thalaqah	12	95	Al Tin	8
6	Al An'am	165	36	Yaa Sin	83	66	Al Tahrif	12	96	Al 'Alaq	19
7	Al A'rat	206	37	As Shaaftat	182	67	Al Muk	30	97	Al Gidr	5
8	Al Anfaal	75	38	Shaad	88	68	Al Qalam	52	98	Al Bayyinah	8
9	Al Taubah	129	39	Az Zumar	75	69	Al Haqqah	52	99	Az Zaizaran	8
10	Yunus	109	40	Al Ghaafr	85	70	Al Ma'aarij	44	100	Al 'Aadiyah	11
11	Hud	123	41	Al Fushiat	54	71	Nuh	28	101	Al Qaari'ah	11
12	Yusuf	111	42	Asy Syuura	53	72	Al Jin	28	102	Al Takaatsur	8
13	Ar Ra'du	43	43	Az Zukhruf	69	73	Al Muzammil	20	103	Al 'Ashr	3
14	Irtibat	42	44	Adz Dzharbiyyah	50	74	Al Huddath	56	104	Al 'Imran	9

Gambar 4.4 Tampilan Daftar Surat Al-Quran

4.3.5 Antarmuka Daftar Hadits

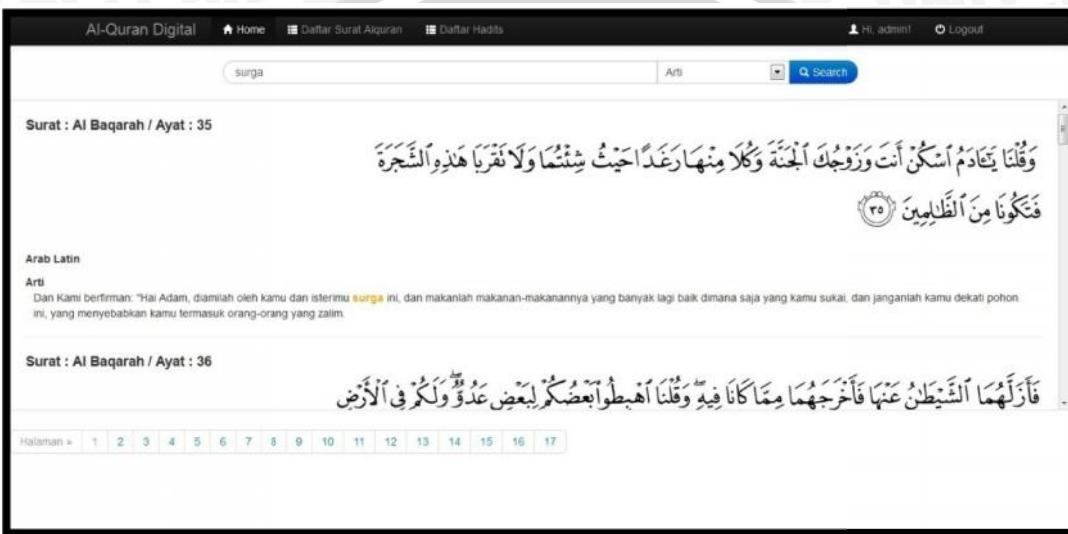
Antarmuka daftar hadits menampilkan daftar dari hadits-hadits yang telah dimasukkan di dalam database Al-Quran Digital. Gambar tampilan ditunjukkan pada gambar 4.5.

No	Nama Hadits	No	Nama Hadits
1	Hadits permulaan wahyu	20	Hadits Al Hawalah - Pengalihan Hartang
2	Hadits iman	21	Hadits Al Wakalah - Perwakilan
3	Hadits ilmu	22	Hadits Mencari Pinjaman dan Melunasi Hutang
4	Hadits Wudhu	23	Hadits Persengketaan
5	Hadits Mandi	24	Hadits Barang Temuan
6	Hadits Hadd	25	Hadits Perbuatan Zhalim dan Merampok
7	Hadits Tayammum	26	Hadits Gada
8	Hadits Sholat	27	Hadits Washiyat
9	Hadits Waktu Sholat	28	Hadits Talaq
10	Hadits Adzan	29	Hadits Nafkah
11	Hadits Junat	30	Hadits Aqiqah
12	Hadits Zakat	31	Hadits Kurban
13	Hadits Haj	32	Hadits Sakit
14	Hadits Shuruu	33	Hadits Respondees

Gambar 4.5 Tampilan Daftar Hadits

4.3.6 Antarmuka Hasil Pencarian

Pada antarmuka Hasil Pencarian akan ditampilkan hasil pencarian dari mesin pencari Al-Quran Digital, yang ditampilkan adalah ayat Al-Quran, Arab latin beserta Arti Ayat Al-Quran dalam Bahasa Indonesia. Dapat dilihat pada gambar-gambar dibawah ini.



Gambar 4.6 Tampilan Hasil Pencarian

4.3.7 Antarmuka Hasil Perhitungan

Antarmuka Hasil Perhitungan merupakan form hasil dari setiap perhitungan kata yang dicari. Berisi tabel matriks perbandingan 2 kata dan perhitungan similarity. Ditunjukkan pada gambar 4.7.

Hasil Perhitungan

Keyword : **syurga**
Kata-kata yg mirip: **surga**

Perhitungan untuk keyword: **syurga** dan kata: **surga**

	s	y	u	r	g	a
s	0	1	2	3	4	5
y	1	0	1	2	3	4
u	2	1	1	1	2	3
r	3	2	2	2	1	2
g	4	3	3	3	2	1
a	5	4	4	4	3	2

```

Similarity = (1 - levenshteinDistance(syurga, surga) / max(length(syurga), length(surga))) * 100
= (1 - 1 / max(6, 5)) * 100
= (1 - 1 / 6) * 100
= (1 - 0.16666666666667) * 100
= 83.33 %

```

Gambar 4.7 Tampilan Hasil Perhitungan

4.3.8 Antarmuka Input Catatan dan Arab Latin

Antarmuka input catatan dan arab latin merupakan form yang digunakan untuk menambahkan catatan bagi user dan juga untuk menambahkan arab latin untuk surat yang belum tersedia arab latin didalamnya. Ditunjukkan pada gambar 4.8.

Bismillah ar-Rahman ar-Rahim

Arab Latin:

Arab Latin tidak tersedia. Silakan isi secara manual melalui form di bawah ini.

Arti :

Dengan menyebut nama Allah Yang Maha Pemurah lagi Maha Penyayang.

Input Manual

Catatan:

Arab Latin:

Simpan Catatan

Gambar 4.8 Tampilan Input Catatan dan Arab Latin

BAB V

ANALISA DAN PEMBAHASAN

5.1 Pengujian

Pada subbab ini akan dilakukan pembahasan mengenai pengujian yang telah dilakukan pada sistem dan hasil evaluasi dari sistem tersebut. Pada pengujian sistem Al-Quran Digital ini menggunakan 1 database yang terdiri dari 8 tabel yaitu, tabel users, tabel keywords, tabel nama_sura, tabel quran, tabel quran_text, tabel users_input, tabel hadits dan tabel arab_latin.

Tabel users adalah tabel data user yang digunakan untuk proses login dan menyimpan proses register. Tabel keywords merupakan tabel untuk menyimpan kata yang sering dipakai di dalam pencarian. Tabel nama_sura berisi daftar nama dan jumlah ayat pada Al-Quran. Tabel quran adalah tabel data quran berupa gambar masing-masing perayat Al-Quran.

Tabel quran_text merupakan tabel arti dari ayat-ayat Al-Quran dalam bahasa Indonesia. Tabel users_input adalah tabel data user digunakan untuk menginput dan mengupdate data masukkan dari user berupa arab latin dan catatan. Tabel hadits berisi daftar nama-nama hadits yang digunakan dalam aplikasi Al-Quran Digital. Tabel arab_latin merupakan tabel arab latin dari ayat-ayat Al-Quran yang sudah berada di dalam database Al-Quran.

5.2 Hasil Uji

Seperti yang telah dibahas pada bab 3, dalam pengujian ini menggunakan Algoritma Levenshtein yang kemudian hasil dari Algoritma tersebut dimasukkan dalam rumus similarity agar persentase jarak kedekatannya dapat diprediksi dan bisa ditemukan kata-kata yang paling mirip dengan yang dicari.

5.3 Hasil Uji Program

Pengujian dilakukan dengan 2 skenario. Skenario pertama, satu kata dengan pengubahan satu huruf, dua huruf, tiga huruf, dan empat huruf dengan panjang kata yang berbeda. Skenario kedua, menggunakan dua buah kata dengan panjang yang berbeda. Kemudian akan ditentukan persentase tingkat kemiripannya.

5.3.1 Uji coba pada satu kata dengan pengubahan pada setiap hurufnya

Hasil uji coba menggunakan kata “surga” yang mempunyai panjang lima karakter.

Tabel 5.1 Persentase tingkat kemiripan pada kata “surga”

Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
surga	surga	5	100%
b urga	surga	5	80%
s arga	surga	5	80%
n irga	surga	5	60%
s ur h s	surga	5	60%
s ud g e	surga	5	60%
b ug g e	surga	5	40%
b und a	surga	5	40%
b agg a	surga	5	40%
b and a	surga	5	20%

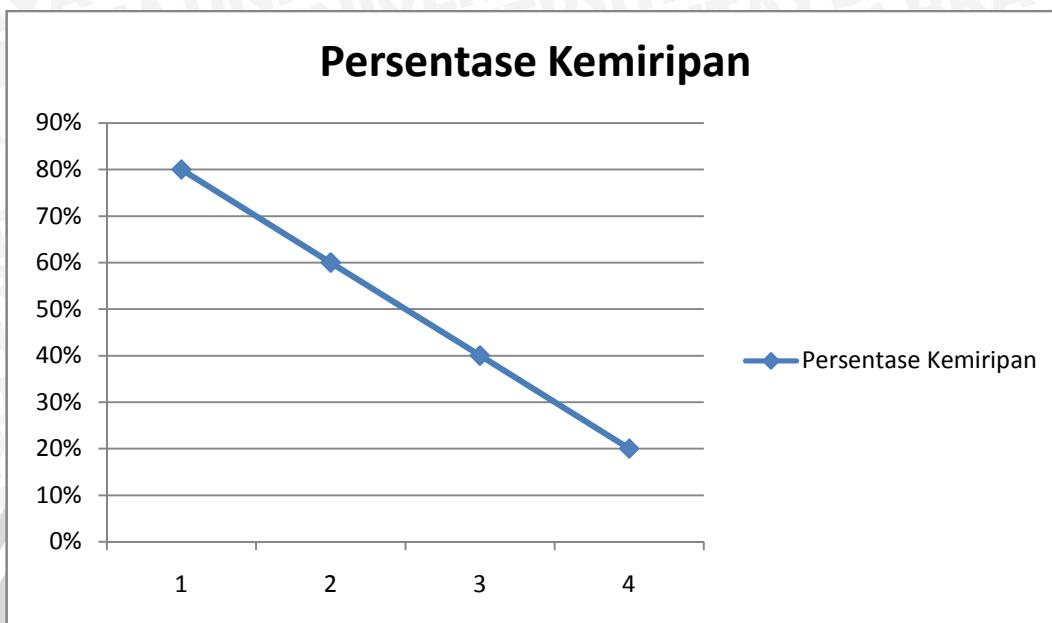
Persentase jumlah huruf yang berubah pada kata “surga” dapat dilihat pada tabel 5.2.

Tabel 5.2 Persentase jumlah huruf yang berubah pada kata “surga”

Jumlah huruf yang berubah	Persentase kemiripan
1	80%
2	60%
3	40%
4	20%



Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.1.



Gambar 5.1 Grafik persentase kemiripan kata "surga"

Hasil uji coba menggunakan kata "sesungguhnya" yang mempunyai panjang 12 karakter.

Tabel 5.3 Persentase tingkat kemiripan pada kata "sesungguhnya"

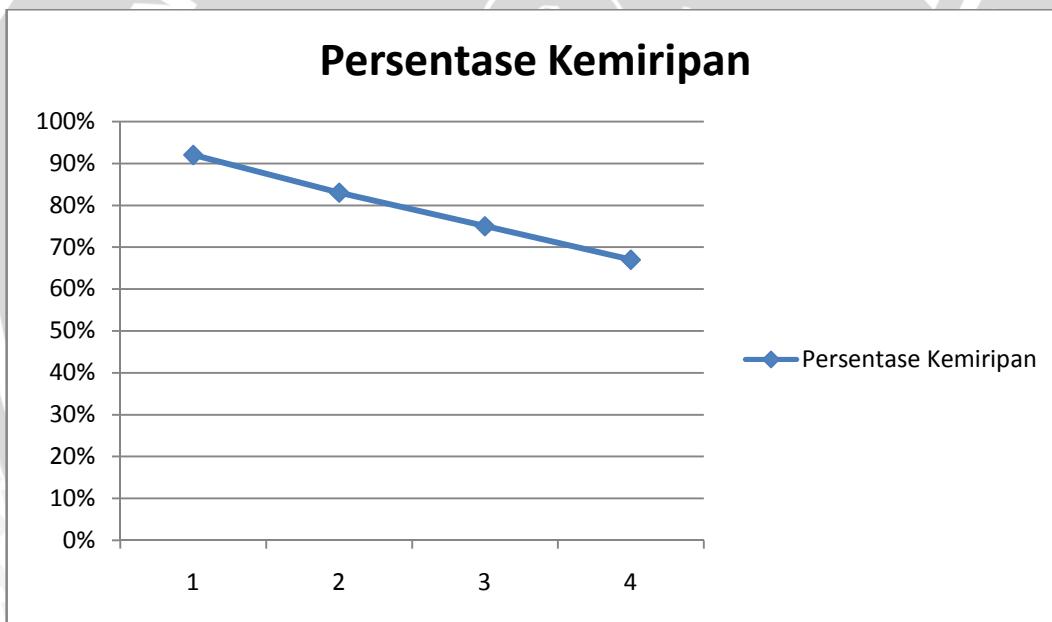
Kata pertama	Kata kedua	Panjang kata	Percentase kemiripan
sesungguhnya	sesungguhnya	12	100%
cesungguhnya	sesungguhnya	12	92%
sesun b guhnya	sesungguhnya	12	92%
a ssungguhnya	sesungguhnya	12	83%
sesung b ahnya	sesungguhnya	12	83%
se b unggugnya	sesungguhnya	12	83%
d esungguhny s	sesungguhnya	12	75%
d rsungguhny a	sesungguhnya	12	75%
s asungfuhng a	sesungguhnya	12	75%
c esanggahny e	sesungguhnya	12	67%

Persentase jumlah huruf yang berubah pada kata “sesungguhnya” yang memiliki panjang 12 karakter dapat dilihat pada tabel 5.4.

Tabel 5.4 Persentase jumlah huruf yang berubah pada kata “sesungguhnya”

Jumlah huruf yg berubah	Persentase kemiripan
1	92%
2	83%
3	75%
4	67%

Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.2.



Gambar 5.2 Grafik persentase kemiripan kata “sesungguhnya”

Hasil ujicoba menggunakan kata “menciptakannya” yang mempunyai panjang 14 karakter.

Tabel 5.5 Persentase tingkat kemiripan pada kata “menciptakannya”

Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
menciptakannya	menciptakannya	14	100%
n enciptakannya	menciptakannya	14	93%
m enciptakann e	menciptakannya	14	93%
n anciptakannya	menciptakannya	14	86%
n enciptajannya	menciptakannya	14	86%
n asciptakannya	menciptakannya	14	79%
m anciptakenmya	menciptakannya	14	79%
n anciptakanmya	menciptakannya	14	79%
n anciptakanmy e	menciptakannya	14	71%
n enviptakanmta	menciptakannya	14	71%

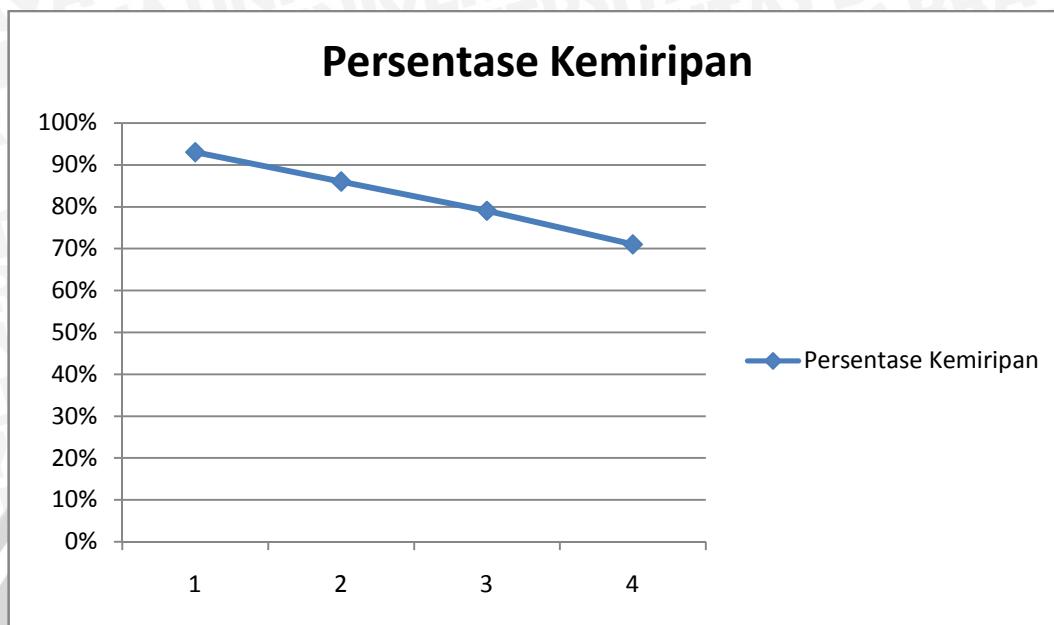
Persentase jumlah huruf yang berubah pada kata “menciptakannya” yang memiliki panjang 14 karakter.

Tabel 5.6 Persentase jumlah huruf yang berubah pada kata “menciptakannya”

Jumlah huruf yg berubah	Persentase kemiripan
1	93%
2	86%
3	79%
4	71%



Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.3.



Gambar 5.3 Grafik persentase kemiripan kata “menciptakannya”

Hasil ujicoba menggunakan kata “segera” yang mempunyai panjang enam karakter.

Tabel 5.7 Persentase tingkat kemiripan pada kata “segera”

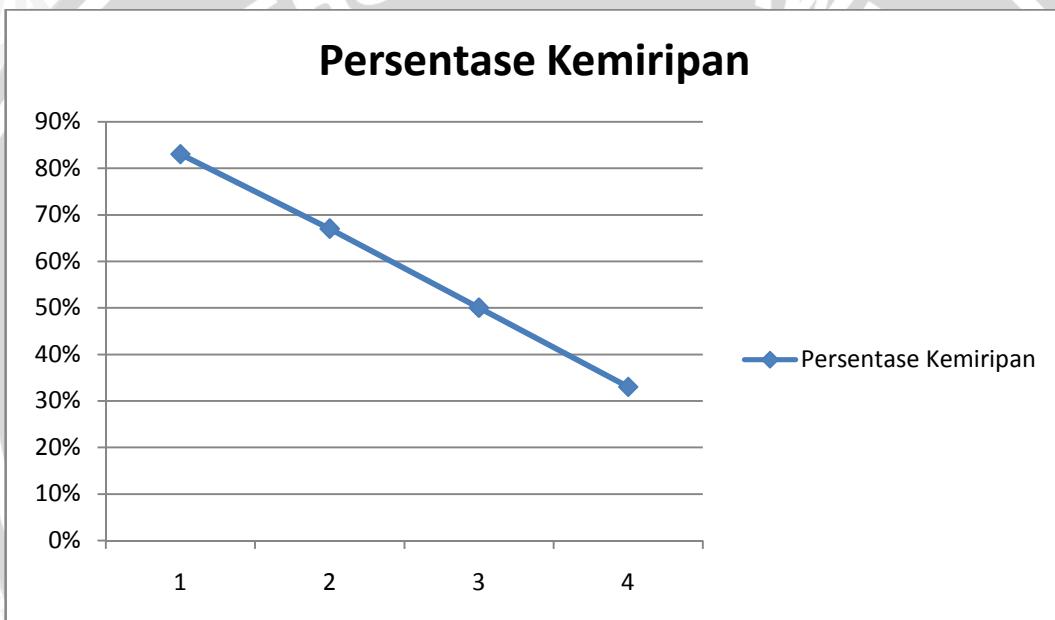
Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
segera	segera	6	100%
degera	segera	6	83%
sagera	segera	6	83%
sefeta	segera	6	67%
segets	segera	6	67%
dagenta	segera	6	50%
dagets	segera	6	33%

Persentase jumlah huruf yang berubah pada kata “segera” yang memiliki panjang enam karakter.

Tabel 5.8 Persentase jumlah huruf yang berubah pada kata “segera”

Jumlah huruf yg berubah	Persentase kemiripan
1	83%
2	67%
3	50%
4	33%

Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.4.



Gambar 5.4 Grafik persentase kemiripan kata “segera”

Hasil ujicoba menggunakan kata “jahanam” yang mempunyai panjang 7 karakter.

Tabel 5.9 Persentase tingkat kemiripan pada kata “jahanam”

Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
jahanam	jahanam	7	100%
kahanam	jahanam	7	86%
jaganam	jahanam	7	86%
kaganam	jahanam	7	71%
jahenan	jahanam	7	71%
kaganan	jahanam	7	57%
kesinam	jahanam	7	43%

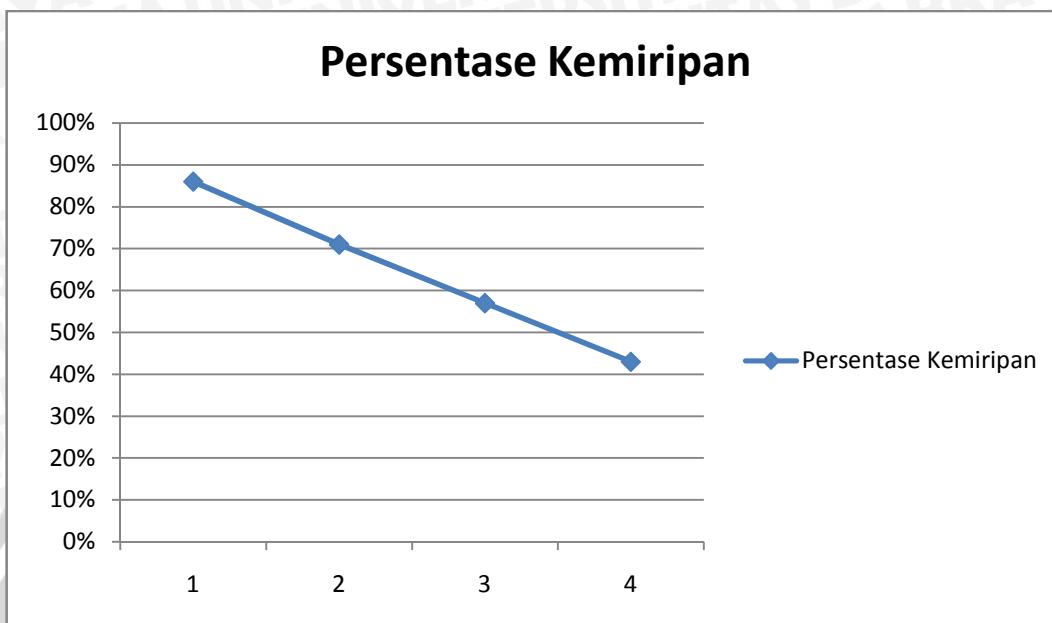
Persentase jumlah huruf yang berubah pada kata “jahanam” yang memiliki panjang tujuh karakter.

Tabel 5.10 Persentase jumlah huruf yang berubah pada kata “jahanam”

Jumlah huruf yg berubah	Persentase kemiripan
1	86%
2	71%
3	57%
4	43%



Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.5.



Gambar 5.5 Grafik persentase kemiripan kata “jahanam”

Hasil ujicoba menggunakan kata “kemudharatan” yang mempunyai panjang 12 karakter.

Tabel 5.11 Persentase tingkat kemiripan pada kata “kemudharatan”

Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
kemudharatan	kemudharatan	12	100%
jemudharatan	kemudharatan	12	92%
kemudhatatan	kemudharatan	12	92%
jemudgaratan	kemudharatan	12	83%
kenidharatan	kemudharatan	12	83%
jenudgaratan	kemudharatan	12	75%
jenudgaratam	kemudharatan	12	67%

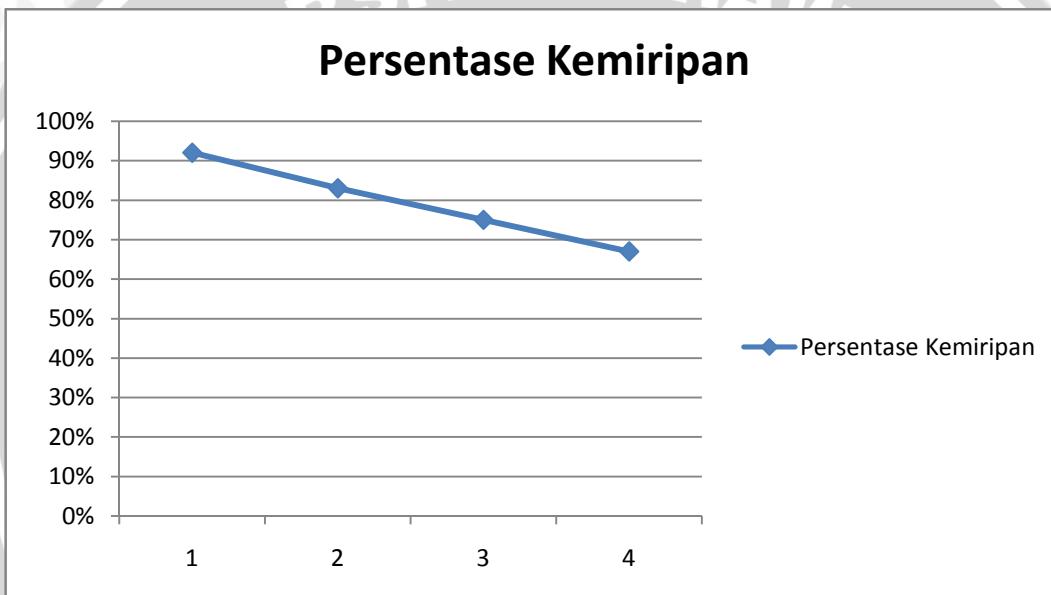
Persentase jumlah huruf pada kata “kemudharatan” yang memiliki panjang 12 karakter.



Tabel 5.12 Persentase jumlah huruf yang berubah pada kata “kemudharatan”

Jumlah huruf yg berubah	Persentase kemiripan
1	92%
2	83%
3	75%
4	67%

Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.6.



Gambar 5.6 Grafik Persentase kemiripan kata “kemudharatan”

Hasil ujicoba menggunakan kata “mata” yang mempunyai panjang empat karakter.

Tabel 5.13 Persentase tingkat kemiripan pada kata “mata”

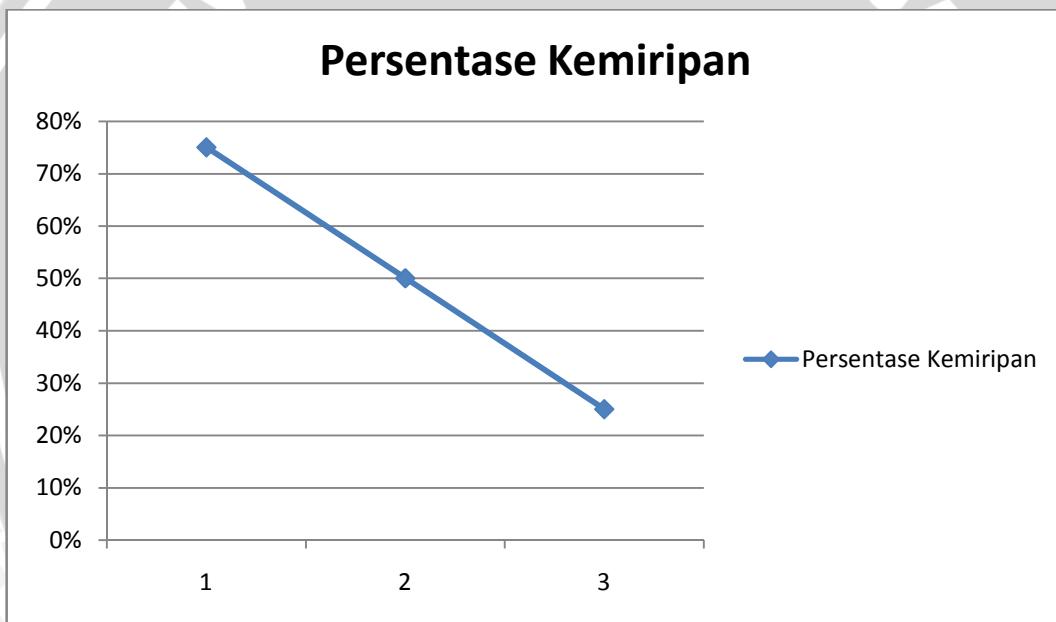
Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
mata	mata	4	100%
nata	mata	4	75%
mita	mata	4	75%
nita	mata	4	50%
niga	mata	4	25%

Persentase jumlah huruf yang berubah pada kata “mata” yang memiliki panjang empat karakter.

Tabel 5.14 Persentase jumlah huruf yang berubah pada kata “mata”

Jumlah huruf yg berubah	Persentase kemiripan
1	75%
2	50%
3	25%

Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.7.



Grafik 5.7 Grafik persentase kemiripan kata “mata”

Hasil ujicoba menggunakan kata “rahmat” yang mempunyai panjang enam karakter.

Tabel 5.15 Persentase tingkat kemiripan pada kata “rahmat”

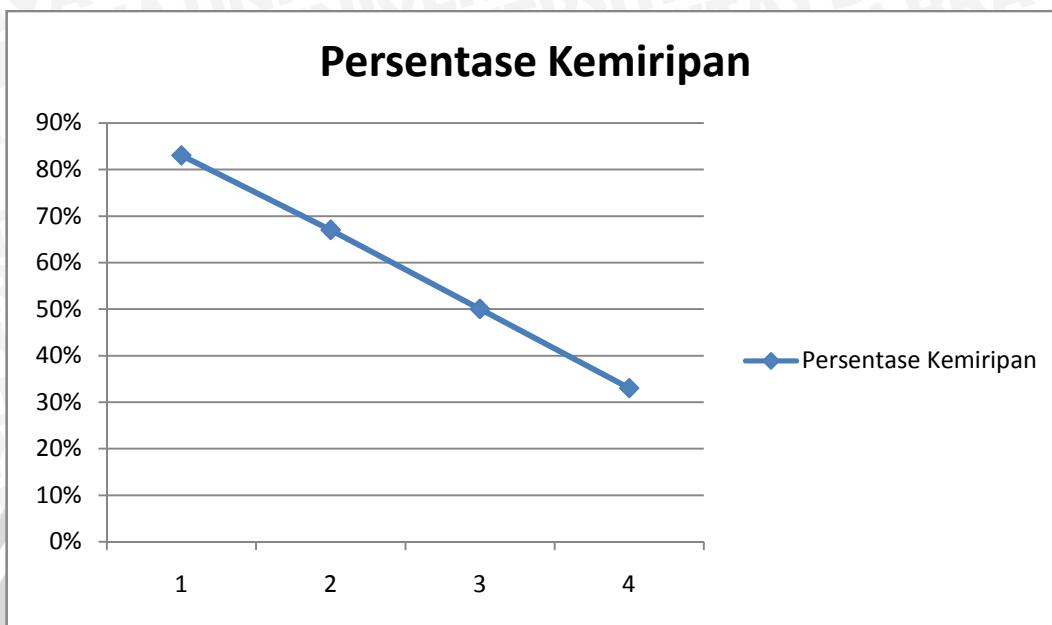
Kata pertama	Kata kedua	Panjang kata	Persentase kemiripan
rahmat	rahmat	6	100%
tahmat	rahmat	6	83%
tehmat	rahmat	6	67%
tehmår	rahmat	6	50%
tagnar	rahmat	6	33%

Persentase jumlah huruf yang berubah pada kata “rahmat” yang memiliki panjang enam karakter.

Tabel 5.16 Persentase jumlah huruf yang berubah pada kata “rahmat”

Jumlah huruf yg berubah	Persentase kemiripan
1	83%
2	67%
3	50%
4	33%

Grafik jumlah huruf yang berubah dapat dilihat pada Gambar 5.8.



Gambar 5.8 Grafik persentase kemiripan kata “rahmat”

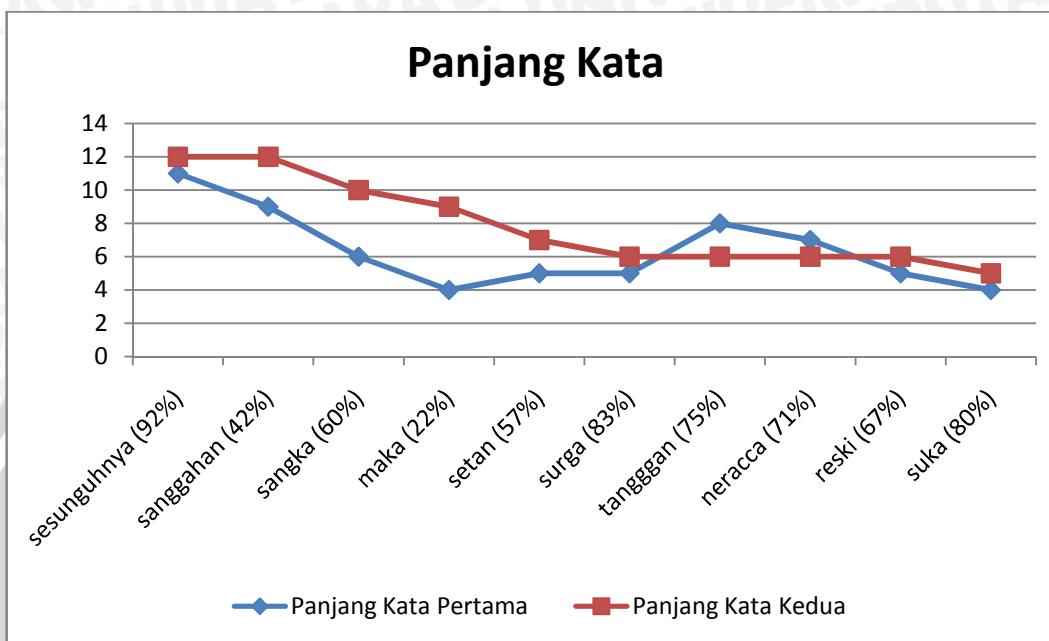
5.3.2 Uji coba pada dua kata dengan dengan panjang yang berbeda

Hasil ujicoba menggunakan kata yang mempunyai panjang karakter berbeda-beda.

Tabel 5.17 Hasil uji coba menggunakan dua kata yang berbeda

Kata pertama	Panjang kata	Kata kedua	Panjang kata	Percentase kemiripan
sesunguhnya	11	sesungguhnya	12	92%
surga	5	surgas	6	83%
suka	4	sukar	5	80%
tangggan	8	tangan	6	75%
neracca	7	neraka	6	71%
reski	5	rejeki	6	67%
sangka	6	sangkakala	10	60%
setan	5	syaitan	7	57%
sanggahan	9	sesungguhnya	12	42%
maka	4	menyembah	9	22%

Panjang dan pendeknya suatu kata tidak mempengaruhi persentase kata akan semakin besar atau kecil, yang menentukan besar atau kecilnya persentase kata adalah jumlah string yang berubah dalam suatu kata.



Gambar 5.9 Grafik panjang kata

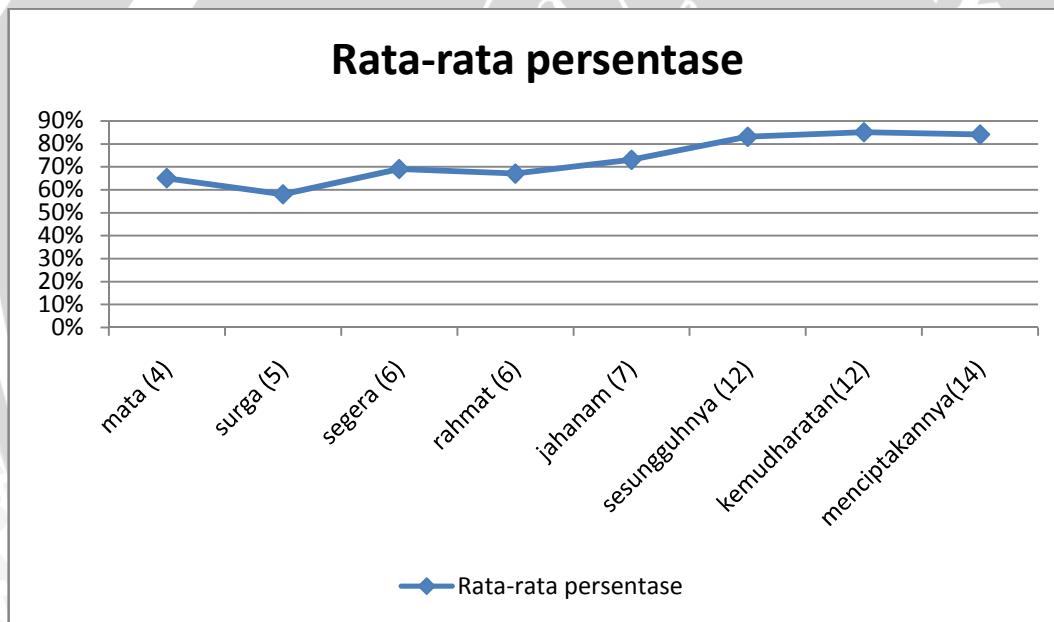
5.4 Analisa Hasil Uji Program

Pada uji coba program, kata “surga” dengan panjang lima karakter memiliki rata-rata persentasi 58%. Kata “sesunguhnya” dengan panjang 12 karakter memiliki rata-rata persentasi 83%. Kata “menciptakannya” dengan panjang 14 karakter memiliki rata-rata persentasi 84%. Kata “segera” dengan panjang enam karakter memiliki rata-rata persentasi 69%. Kata “jahanam” dengan panjang tujuh karakter memiliki rata-rata persentasi 73%. Kata “kemudharatan” dengan panjang 12 karakter memiliki rata-rata 85%. Kata “mata” dengan panjang empat karakter memiliki rata-rata persentasi 65%. Kata “rahmat” dengan panjang enam karakter memiliki rata-rata persentasi 67%.

Tabel 5.18 Rata-rata persentase tiap kata yang diujikan

Kata	Panjang Kata	Rata-rata Persentase
mata	4	65%
surga	5	58%
segera	6	69%
rahmat	6	67%
jahanam	7	73%
sesungguhnya	12	83%
kemudharatan	12	85%
menciptakannya	14	84%

Grafik dari rata-rata persentase tiap kata dapat dilihat pada gambar 5.10.



Gambar 5.10 Grafik rata-rata persentase

Panjang dan pendeknya suatu kata tidak mempengaruhi hasil persentase kata tersebut semakin tinggi. Tingkat persentase semakin tinggi bila jumlah huruf yang berubah pada suatu kata semakin rendah.



Pada program ditentukan persentase tingkat kemiripan kata yang ditampilkan pada suggestion kata adalah lebih dari sama dengan 65%. Hal ini dikarenakan hasil dari pengujian program menunjukkan bila kata dengan persentase kurang dari 65%, kata tersebut jauh berbeda dari yang diharapkan untuk bisa tampil pada *suggestion* kata.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil uji dan analisis yang telah dilakukan, dapat diambil kesimpulan yaitu :

1. Algoritma Levenshtein Distance merupakan matrik yang digunakan untuk mengukur perbedaan jarak antara dua string, Levenshtein Distance antara dua string ditentukan berdasarkan jumlah minimum perubahan / pengeditan yang diperlukan untuk melakukan transformasi dari satu bentuk string ke bentuk string yang lain. Setelah diketahui jarak minimumnya, kemudian ditentukan nilai persentasenya menggunakan rumus similarity.
2. Ditentukannya tingkat kemiripan sebesar 65% untuk menampilkan *string suggestion* kata yang cocok pada mesin pencari Al-Quran Digital khususnya pada surat Yasin.

6.2 Saran

Adapun saran untuk pengembangan lebih lanjut yang dapat diberikan oleh penulis adalah untuk pengembangan penelitian selanjutnya aplikasi ini mampu mendekripsi semua bentuk seperti yang berupa frase, kata yang dipisahkan oleh spasi atau lebih dari 1 kata. Menerapkan *Algoritma Levensthein* pada semua surat yang ada pada Al-Quran,



DAFTAR PUSTAKA

- [ABD-01] Abdul Kadir. (2001). *Dasar Pemrograman Web Dinamis Menggunakan PHP*. C.V Andi Offset. Yogyakarta.
- [ALE-96] Alexander Bogomolny. (1996). *Distance Between Strings*. University of Iowa. United State.
- [BIN-95] Binstock & Rex. (1995). "Partical Algorithm for Programmers". Addison Wesley.
- [BRU-07] Bruno Woltzenlogel Paleo. (2007). *An Approximate Gazetteer for GATE based on Levenshtein Distance*. Technische Universitaet Wien.
- [BOR-05] Borivoj Melichar, Jan Holub and Tomas Polcar. (2005). *Text Searching Algorithms*. Czech Technical University. Prague.
- [CHE-06] Cheng L, Ferreira H.C. (2006). *Pruned Convolutional Codes and Viterbi Decoding using The Levenshtein Distance Metric Applied to Asynchronous Noisy Channels*. Dept. of Electrical and Electronic Science, University of Johannesburg. South Africa.
- [CHR-97] Christian charras and Thierry Lecroq. (1997). *Exact String Matching Algorithms*. Universite de Rouen. France.
- [HER-10] Herdianto Aditya. (2010). *Pencarian Ayat-ayat Al-Quran Berdasarkan Konten menggunakan Text Mining Berbasis Aplikasi Dekstop*. Institut Teknologi Sepuluh Nopember. Surabaya.
- [LLO-99] Lloyd Allison. (1999). *Dynamic Programming Algorithm (DPA) for Edit-Distance*. School of Computer Science & SWE. Monash University. Australia.
- [MUN-12] Muni Adriyani. (2012). *Implementasi Algoritma Levensthein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa indonesia*. Universitas Udayana.

- [PAU-99] Paul Black E. (1999). *Dictionary of Algorithms and Data Structures : Levenshtein Distance*. U.S. National Institute of Standards and Technology. United State.
- [SYA-04] Syaroni, Mukhamad dkk. (2004). *Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Indonesia*. Departemen Teknik Informatika ITB. Bandung.
- [TRE-01] Trevisan, L. (2001). *Notes for Lecture 13 – Edit Distance*. United State.
- [VLA-66] Vladimir I. Levenshtein. (1966). *Binary codes capable of correcting deletions, insertions and reversals*. Technical Report 8. Russian.

