



KERJA PRAKTIK - IF184801

Aplikasi Cross Lingual Information Retrieval Berbasis Website untuk Bahasa Indonesia- Bahasa Arab

**Departemen Teknik Informatika Institut Teknologi
Sepuluh Nopember
Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus
ITS Sukolilo, Surabaya, Jawa Timur 60111**

Periode: 8 Juni 2020 - 9 Agustus 2020

Oleh:

Muhammad Harits Fadlilah

0511174000102

Pembimbing Jurusan
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Pembimbing Lapangan
Dini Adni Navastara, S.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



KERJA PRAKTIK - IF184801

Aplikasi Cross Lingual Information Retrieval Berbasis Website untuk Bahasa Indonesia-Bahasa Arab

Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember

Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya, Jawa Timur 60111

Periode: 8 Juni 2020 - 9 Agustus 2020

Oleh:

Muhammad Harits Fadlilah

05111740000102

Pembimbing Jurusan

Dr. Agus Zainal Arifin, S.Kom., M.Kom.

Pembimbing Lapangan

Dini Adni Navastara S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**


**Aplikasi Cross Lingual Information Retrieval Berbasis
Website untuk Bahasa Indonesia-Bahasa Arab**

Oleh:


Muhammad Harits Fadlilah 05111740000102

Disetujui oleh Pembimbing Kerja Praktik:

1. Prof. Dr. Agus Zainal Arifin,
S.Kom., M.Kom.
NIP 197208091995121001


.....
(Pembimbing Departemen)

2. Dini Adni Navastara S.Kom.,
M.Sc.
NIP 198510172015042001


.....
(Pembimbing Lapangan)

**SURABAYA
Januari, 2021**

[Halaman ini sengaja dikosongkan]

***Aplikasi Cross Lingual Information Retrieval
untuk Bahasa Indonesia-Bahasa Arab***

Nama Mahasiswa : Muhammad Harits Fadlilah
NRP : 05111740000102
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Jurusan : Dr. Agus Zainal Arifin, S.Kom., M.Kom.

ABSTRAK

Dalam berbagai bidang kehidupan, seperti bidang pendidikan dan pemerintahan dokumen-dokumen berbahasa Arab menjadi salah satu yang dijadikan sebagai referensi. Hal tersebut menunjukkan adanya kebutuhan yang tinggi dari masyarakat Indonesia terhadap dokumen-dokumen berbahasa Arab, yang mana bahasa Arab sendiri merupakan bahasa peribadatan umat Islam dan merupakan bahasa yang umum digunakan dalam dokumen-dokumen Islam. Dalam melakukan pencarian dokumen-dokumen referensi dalam hal keagamaan (Islam) dibutuhkan mesin pencari yang dapat mengembalikan dokumen berbahasa Arab. Hal ini dikarenakan koleksi dokumen terkait permasalahan dalam Islam lebih lengkap disajikan dalam bahasa *native*-nya yaitu bahasa Arab.

Oleh karena itu, pada kerja praktik ini akan dikembangkan sebuah sistem *Cross Lingual Information Retrieval* yang dapat melakukan pengembalian dokumen lintas bahasa Arab dengan input kueri bahasa Indonesia.

Kata kunci:

Cross Lingual Information Retrieval, Bahasa Indonesia, Bahasa Arab

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat limpahan rahmat dan lindungan-Nya penulis dapat melaksanakan salah satu kewajiban sebagai mahasiswa Teknik Informatika ITS yaitu Kerja Praktik (KP).

Penulis menyadari masih terdapat banyak kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini, namun kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan penulisan buku laporan ini.

Melalui laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu dalam pelaksanaan kerja praktik hingga penyusunan laporan Kerja praktik baik secara langsung maupun tidak langsung. Orang-orang tersebut antara lain adalah:

1. Bapak Dr. Agus Zainal Arifin, S.Kom., M.Kom., selaku dosen pembimbing kerja praktik yang telah membimbing penulis selama kerja praktik berlangsung.
2. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D, selaku koordinator Kerja Praktik.
3. Ibu Dini Adni Navastara, S.Kom. selaku pembimbing lapangan selama kerja praktik yang telah memberikan bimbingan serta ilmunya kepada penulis.
4. Mahasiswa pascasarjana bimbingan Bapak Dr. Agus Zainal Arifin, S.Kom., M.Kom., yang telah membantu penulis selama kerja praktik berlangsung.

Surabaya, Agustus 2020

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN..... V

ABSTRAK..... VII

KATA PENGANTAR VIII

DAFTAR ISI X

DAFTAR GAMBAR XV

DAFTAR TABELXVI

BAB I PENDAHULUAN 1

1.1. LATAR BELAKANG..... 1

1.2. TUJUAN 3

1.3. MANFAAT 3

1.4. RUMUSAN PERMASALAHAN 3

1.5. LOKASI DAN WAKTU KERJA PRAKTIK 4

1.6. METODOLOGI KERJA PRAKTIK 4

1.7. SISTEMATIKA LAPORAN 6

BAB II PROFIL INSTANSI..... 9

2.1. PROFIL INSTANSI..... 9

2.2. LOKASI..... 10

BAB III TINJAUAN PUSTAKA..... 12

3.1. CLIR 12

3.2. DATASET MAKTABAH SYAMILAH 12

3.3. APLIKASI WEBSITE 13

3.4. DATA PREPROCESSING 14

3.5. TF-IDF VECTORIZER 14

3.6. WORD EMBEDDING..... 15

3.7. EKSPANSI KUERI.....	15
3.8. ANACONDA	16
3.9. PYTHON.....	16
3.10. DJANGO	17
3.11. HTML.....	17
3.12. CSS	17
3.13. MYSQL.....	18
3.14. WEB HOSTING	19
3.15. VISUAL STUDIO CODE	19
BAB IV METODOLOGI SISTEM.....	21
4.1. Metodologi Implementasi.....	21
4.1.1. Persiapan Dataset.....	22
4.1.2. Preprocessing Data.....	23
4.1.2.1. Remove Diacritics.....	23
4.1.2.2. Normalization.....	23
4.1.2.3. Remove Punctuations.....	23
4.1.2.4. Remove Punctuations Chars.....	23
4.1.2.5. Remove Alphabet and Numeric.....	24
4.1.2.6. Remove Stopword.....	24
4.1.2.7. Remove Stemming.....	24
4.1.2.8. Tokenizing.....	24
4.1.3. Word Embedding.....	25
4.1.4. Proses Kueri.....	25
4.1.4.1. Input Kueri.....	25
4.1.4.2. Preprocessing Kueri.....	25
4.1.4.3. Translasi Kueri.....	26
4.1.4.4. Ekspansi Kueri.....	26
4.1.4.5. Pencarian Dokumen.....	26
4.1.5. Pembuatan Aplikasi.....	27
4.2. Metodologi Evaluasi.....	28
4.2.1. Uji Coba Sistem terhadap Ground Truth.....	28

4.2.2. Uji Coba Fungsionalitas Aplikasi.....	29
4.3. Metodologi Maintenance.....	29
BAB V IMPLEMENTASI SISTEM	32
5.1. IMPLEMENTASI ANACONDA	32
5.1.1. Import Library.....	32
5.1.2. Preprocessing Data.....	33
5.1.3. Word Embedding dan TF-IDF Vectorizer.....	42
5.1.4. Search Engine.....	43
5.1.5. Pembuatan Kueri Uji Ground Truth.....	46
5.2. Implementasi Website.....	48
5.2.1. App Api.....	49
5.2.1.1. Api urls.py.....	49
5.2.1.2. Api views.py.....	49
5.2.1.3. Api templates base.html.....	49
5.2.1.4. Api templates index.html.....	50
5.2.1.5. Api templates snippets form_styles.html.....	52
5.2.1.6. Api templates snippets scripts.html.....	52
5.2.1.7. Api templates snippets navbar.html.....	53
5.2.2. App Predictor.....	54
5.2.2.1. Predictor urls.py.....	54
5.2.2.2. Predictor apps.py.....	55
5.2.2.3. Predictor views.py.....	55
5.2.2.4. Predictor forms.py.....	60
5.2.2.5. Predictor admin.py.....	61
5.2.2.6. Predictor models.py.....	61
5.2.2.7. Predictor templates index.html.....	62
5.2.2.8. Predictor templates review.html.....	64
5.2.2.9. Predictor static css styles.css.....	66
5.2.3. App Account.....	67
5.2.3.1. Account urls.py.....	67
5.2.3.2. Account views.py.....	67

5.2.3.3. Account forms.py.....	70
5.2.3.4. Account templates login.html.....	72
5.2.3.5. Account templates register.html.....	73
5.2.3.6. Account static css styles.css.....	75
5.2.4. App Kitab.....	76
5.2.4.1. Kitab urls.py.....	76
5.2.4.2. Kitab views.py.....	76
5.2.4.3. Kitab forms.py.....	77
5.2.4.4. Kitab models.py.....	78
5.2.4.5. Kitab admin.py.....	79
5.2.4.6. Kitab templates index.html.....	79
5.2.4.7. Kitab static css styles.css.....	81
5.3. Implementasi Antarmuka Pengguna.....	82
5.3.1. Halaman Utama.....	82
5.3.2. Halaman Registrasi User.....	84
5.3.3. Halaman Login User.....	84
5.3.4. Halaman Hasil Pencarian Dokumen.....	85
5.3.5. Halaman Upload Kitab.....	87
5.3.6. Halaman Review Hasil Pencarian Dokumen.....	89
BAB VI PENGUJIAN DAN EVALUASI.....	90
6.1. SKENARIO PENGUJIAN.....	90
6.1.1. Skenario Uji Coba Relevansi Aplikasi.....	90
6.1.2. Skenario Uji Coba Fungsionalitas Aplikasi.....	91
6.1.2.1. Membuka Halaman Utama Kitab Ulama..	91
6.1.2.2. Melakukan Registrasi Kitab Ulama.....	91
6.1.2.3. Melakukan Login Kitab Ulama.....	92
6.1.2.4. Menggunakan Search Engine pada Kitab Ulama.....	92
6.1.2.5. Melakukan Review Hasil Pencarian Dokumen Kitab Ulama.....	92
6.1.2.6. Melakukan Upload Kitab Ulama.....	93

6.2. EVALUASI PENGUJIAN	93
6.2.1. Evaluasi Uji Coba Relevansi Sistem.....	94
6.2.2. Evaluasi Uji Coba Fungsionalitas Sistem.....	101
BAB VII KESIMPULAN DAN SARAN	104
7.1. Kesimpulan.....	104
7.2. Saran.....	104
DAFTAR PUSTAKA	105

DAFTAR GAMBAR

Gambar 4.1. Diagram Metodologi Sistem.....	21
Gambar 4.2. Diagram Metodologi Implementasi.....	22
Gambar 4.3. Dataset Arabic.....	22
Gambar 4.4. Diacritics Arabic.....	23
Gambar 4.5. Formula <i>Recall</i> dan <i>Precision</i>	29
Gambar 5.1. Tampilan Halaman Utama Sebelum Login.....	83
Gambar 5.2. Tampilan Halaman Utama Setelah Login.....	83
Gambar 5.3. Tampilan Halaman Registrasi User.....	84
Gambar 5.4. Tampilan Halaman Login Akun.....	85
Gambar 5.5. Tampilan Halaman Hasil Pencarian Dokumen Bagian 1	86
Gambar 5.6. Tampilan Halaman Hasil Pencarian Dokumen Bagian 2	86
Gambar 5.7. Tampilan Halaman Upload Kitab.....	87
Gambar 5.8. Tampilan Halaman Review Hasil Pencarian Dokumen.....	88

DAFTAR TABEL

Tabel 6.1. Hasil Evaluasi Uji Coba Relevansi Aplikasi.....	94
Tabel 6.2. Hasil Evaluasi Uji Coba Fungsionalitas Aplikasi.....	101

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Di Indonesia mayoritas penduduknya menganut agama Islam. Dari tatanan pemerintahan sendiri terdapat beberapa forum pembahasan masalah seperti *Bahtsul Masail*, *Majelis Tarjih*, MUI (Majelis Ulama Indonesia) yang bertujuan untuk menjawab permasalahan terkini dalam masyarakat dengan mengacu pada dokumen-dokumen Islam. Hal tersebut menunjukkan adanya kebutuhan yang tinggi dari masyarakat Indonesia terhadap dokumen-dokumen bahasa Arab, yang mana bahasa Arab sendiri merupakan bahasa peribadatan umat Islam dan merupakan bahasa yang umum digunakan dalam dokumen-dokumen Islam. Akan tetapi tidak semua masyarakat Indonesia yang menganut agama Islam juga fasih dan mengerti bahasa Arab sebagaimana mereka mengerti bahasa Indonesia. Selain itu perbedaan penulisan antara bahasa Indonesia dan bahasa Arab yang sangat mencolok, dimana bahasa Arab menggunakan huruf hijaiyah, menimbulkan ketidakcocokan dengan alat masukan seperti *keyboard* yang umumnya menggunakan alphabet biasa.

Dalam proses pencarian dokumen, masukan dari pengguna merupakan hal yang esensial dalam menentukan relevansi dari dokumen hasil. Pada umumnya dalam melakukan pencarian dokumen, antara *query* masukan dari pengguna dan dokumen yang dihasilkan memiliki bahasa yang sama. Sehingga tidak

ada *gap* antara bahasa *query* dengan bahasa dari dokumen yang hendak ditampilkan. Dalam melakukan pencarian dokumen referensi dalam hal keagamaan (Islam) dibutuhkan mesin pencari yang dapat mengembalikan dokumen berbahasa Arab. Hal ini dikarenakan koleksi dokumen terkait permasalahan dalam Islam lebih lengkap disajikan dalam bahasa *native*-nya yaitu bahasa Arab. Akan tetapi bagi masyarakat Indonesia yang merupakan *non-native speaker* untuk bahasa Arab, melakukan penyusunan *query* pencarian dalam bahasa Arab lebih susah dibandingkan dengan bahasa Indonesia, yang merupakan bahasa Ibu. Sehingga disini muncul *gap* bahasa antara bahasa *native* dari pengguna dengan bahasa *non-native* dari dokumen koleksi. Oleh karena itu, pada kerja praktik ini dikembangkan sistem *cross lingual information retrieval* bahasa Arab dengan input bahasa Indonesia. Sistem pencarian ini mampu mengakomodasi masukan dari pengguna dalam bahasa Indonesia dan mengembalikan hasil berupa dokumen relevan dalam bahasa Arab. Dengan demikian pengguna akan lebih mudah melakukan penyusunan *query* pada sistem pencarian dikarenakan *query* dapat ditulis dengan bahasa *native*-nya, yaitu bahasa Indonesia.

Hasil akhir dari kerja praktik ini merupakan sebuah aplikasi CLIR yang akan dipergunakan sebagai salah satu aplikasi penunjang pembelajaran pada institusi mitra. Dengan adanya aplikasi ini diharapkan mampu meningkatkan kualitas, efektifitas, dan efisiensi

dari institutsi mitra dalam melakukan proses pembelajaran.

1.2.Tujuan

Tujuan kerja praktik kali ini adalah menyelesaikan kewajiban nilai mata kuliah kerja praktik sebesar 2 SKS serta membangun aplikasi *cross lingual information retrieval* berbasis website yang mampu mengakomodasi kebutuhan pengguna dengan membangun metode yang dapat secara otomatis melakukan ekstraksi relasi semantic pada pencarian dokumen.

1.3. Manfaat

Berikut manfaat yang diperoleh melalui kerja praktik dalam pembangunan aplikasi CLIR berbasis website sesuai kebutuhan pengguna:

- Dapat digunakan sebagai salah satu aplikasi penunjang pembelajaran dalam hal keagamaan (Islam) pada institusi mitra.
- Dapat meningkatkan kualitas, efektifitas, dan efisiensi dari institutsi mitra dalam melakukan proses pembelajaran.

1.4.Rumusan Permasalahan

Berikut rumusan masalah pelaksanaan kerja praktik dalam pembangunan aplikasi CLIR berbasis website sesuai kebutuhan pengguna:

- Bagaimana membangun aplikasi *dual lingual information retrieval* berbasis website yang dapat mengakomodasi kebutuhan pengguna.

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik kali ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Online

Waktu : 8 Juni 2020 – 9 Agustus 2020

1.6. Metodologi Kerja Praktik

1. Perumusan Masalah

Untuk mengetahui domain dan fungsionalitas, dijelaskan secara rinci bagaimana sistem yang harus dibuat. Penjelasan oleh pembimbing lapangan kerja praktik kali ini menghasilkan beberapa catatan mengenai gambaran secara garis besar tentang kebutuhan atau fitur apa saja yang harus ada di dalam aplikasi. Setelah mendapatkan gambaran aplikasi, diskusi lebih lanjut dilakukan guna menentukan rancangan serta *tools* pendukung pembuatan sistem.

2. Studi Literatur

Pada tahap ini, setelah ditentukannya rancangan *database*, bahasa pemrograman sampai dengan teknologi beserta *tools* tambahan yang digunakan, dilakukan studi literatur lanjut mengenai bagaimana penggunaannya dalam membangun sistem sesuai yang diharapkan.

Aplikasi yang akan dibuat merupakan system yang akan dibangun, ada beberapa tools yang digunakan. Untuk frontend digunakan CSS. Sedangkan untuk backend digunakan Python. Ada beberapa tools tambahan yang mendukung pembuatan website tersebut yaitu XAMPP, Visual Studio Code, dan Anaconda. Framework yang digunakan yaitu Django berbasis Python.

3. Analisis dan Perancangan Sistem

Langkah ini meliputi tahapan-tahapan pekerjaan yang dilakukan dalam pembangunan sistem. Bagaimana cara kerja sistem dengan skenario tertentu. Dari penjelasan awal telah didapatkan beberapa kebutuhan fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas dan menspesifikkan kebutuhan-kebutuhan tersebut. Dilanjutkan berdiskusi dengan pembimbing lapangan untuk mengetahui apakah kebutuhan-kebutuhan tersebut sudah tepat.

4. Implementasi Sistem

Implementasi sistem didasarkan oleh perancangan dan analisis sebelumnya. Penentuan atribut atau fitur yang akan digunakan pada model juga didasari pada analisis sebelumnya. Penentuan tipe data dan format keluaran juga disesuaikan dengan kebutuhan.

Pengerjaan dilakukan dengan progres setiap minggu, dengan setiap minggunya menargetkan

perkembangan dari hari sebelumnya. Progres penyelesaian aplikasi terus dipantau oleh tim penelitian dengan adanya *zoom meeting* secara daring setiap hari Rabu pukul 15.00 WIB. Selain itu pembimbing lapangan juga memberikan masukan-masukan apabila terdapat kendala.

5. Pengujian dan Evaluasi

Pengujian dilakukan oleh penulis dan para user pada setiap fitur yang sudah selesai untuk memberikan evaluasi ketika ada yang tidak sesuai dan persetujuan apabila sudah sesuai.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari 7 bab dengan rincian sebagai berikut :

1. Bab I: Pendahuluan

Bab ini berisi tentang latar belakang masalah, tujuan, manfaat, rumusan masalah, lokasi dan waktu, metodologi kerja praktik, dan sistematika laporan.

2. Bab II: Profil Instansi

Bab ini berisi sekilas tentang profil Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember.

3. Bab III: Tinjauan Pustaka

Dalam bab ini dibahas mengenai konsep-konsep pembuatan aplikasi, dasar teori, teknologi yang dipakai dalam pembuatan model.

4. Bab IV: Metodologi Sistem

Dalam bab ini dibahas tentang tahapan-tahapan pembuatan suatu aplikasi sesuai dengan kebutuhan dari pengguna.

5. Bab V: Implementasi Sistem

Dalam bab ini dibahas tentang implementasi secara keseluruhan.

6. Bab VI: Pengujian dan Evaluasi

Dalam bab ini dibahas tentang skenario pengujian dan evaluasi pengujian setelah aplikasi selesai dibangun.

7. Bab VII: Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan dan saran yang didapatkan dari tugas selama kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL INSTANSI

2.1. Profil Instansi

Sebagai perkiraan untuk saat ini dan masa yang akan datang, teknologi informasi menjadi tulang punggung pertumbuhan ekonomi bangsa. Saat ini pun inovasi teknologi informasi sudah terasa di berbagai bidang kehidupan manusia. Hal ini sepenuhnya disadari oleh pemerintah, sehingga sejak Repelita V yang lalu, pemerintah telah mencanangkan bahwa pengembangan pendidikan tinggi dalam bidang komputer dan informatika merupakan salah satu program prioritas, bersama-sama dengan disiplin ilmu lainnya seperti rekayasa, perilaku, manajemen, akuntansi, dan kesenian.

Pendidikan tinggi diarahkan untuk mempersiapkan bangsa Indonesia dalam menghadapi era pembangunan industri dan informasi. Untuk itu pemerintah melalui Direktorat Jendral Pendidikan Tinggi pada tahun 1985 menginstruksikan untuk membuka Program Studi S1 baru untuk bidang ilmu teknologi komputer di empat universitas atau institut di mana ITS termasuk di dalamnya. Di ITS, program ini awalnya diberi nama Program Studi Teknik Komputer. Namun sejak tahun 1993, nama Program Studi Teknik Komputer diubah menjadi Jurusan Teknik Komputer. Akhirnya, pada tahun 1996 secara resmi jurusan ini berganti nama menjadi Jurusan Teknik Informatika berdasarkan Surat Keputusan

Direktur Jendral Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996, tanggal 11 Juli 1996. Pada saat ini, Jurusan Teknik Informatika memperoleh nilai akreditasi A berdasarkan Surat Keputusan Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) Nomor 003/BAN-PT/Ak-X/S1/V/2006, tanggal 18 Mei 2006.

Selain program Sarjana (S1), Jurusan Teknik Informatika juga menyelenggarakan program Pasca Sarjana (S2) yang dirintis sejak tahun 1994, dengan surat keputusan Direktur Jendral Pendidikan Tinggi No. 2851/D/T/2001, perihal ijin penyelenggaraan Program-Program Studi Jenjang Program Strata-2 (S2) pada Institut Teknologi Sepuluh Nopember Surabaya. Dan pada tahun 2011, Jurusan Teknik Informatika mulai menyelenggarakan program Doktor (S3).

2.2.Lokasi

Jl. Teknik Kimia – Gedung Departemen Teknik Informatika Kampus Institut Teknologi Sepuluh Nopember Surabaya Jalan Raya ITS, Sukolilo, Surabaya 60111, Indonesia.

[Halaman ini sengaja dikosongkan]

BAB III TINJAUAN PUSTAKA

3.1. CLIR

CLIR (*Cross Lingual Information Retrieval*) memiliki tujuan untuk melakukan pencarian dokumen yang memiliki bahasa yang berbeda dari bahasa yang digunakan dalam *query* [7]. CLIR sendiri terdiri dari 2 macam yaitu DLIR (*Dual Lingual Information Retrieval*) dan MLIR (*Multi Lingual Information Retrieval*). CLIR dengan berbasis pada penerjemahan terdiri dari 2 jenis, yaitu: penerjemahan *query* ke dalam bahasa dari dokumen koleksi dan penerjemahan dokumen koleksi ke dalam bahasa dari *query*.

3.2. Dataset Maktabah Syamilah

Maktabah Syamilah (مكتبة شاملا; *Maktaba Shamela*) merupakan sebuah peranti lunak komputer gratis berbasis Microsoft Windows yang berbentuk Pembaca (*reader*) buku elektronik dan perpustakaan digital. Peranti ini dapat dimasukkan dengan berkas (*file*) buku elektronik hingga puluhan ribu kitab. Peranti ini memiliki berbagai fitur seperti pencarian cepat, pencatatan, referensi silang, *portabel*-itas dan berbagai kemudahan lainnya yang tidak didapat pada kitab berwujud fisik biasa. [1]

Maktabah Syamilah sudah sangat dikenal di kalangan pesantren, akademisi dan cendekiawan muslim karena dalam software ini tergabung beragam kitab ulama islam klasik dan kontemporer disusun perkategori misalkan Tafsir, Hadits, Aqidah, Fiqih, dan lain-lain.

Salah satu situs yang menyediakan kitab-kitab klasik yang dapat dipasang di Maktabah Syamilah adalah www.shamela.ws.

3.3. Aplikasi Website

Dalam rekayasa perangkat lunak, suatu aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu *thin client* (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. [2]

3.4. Data Preprocessing

Data Preprocessing adalah teknik data mining yang melibatkan transformasi data mentah ke dalam format yang dapat mudah dimengerti. Data pada dunia nyata seringkali tidak lengkap, tidak konsisten, dan/atau kurang dalam perilaku tertentu, dan kemungkinan besar mengandung banyak kesalahan. Data Preprocessing adalah metode yang terbukti untuk menyelesaikan masalah tersebut. [3]

3.5. TF-IDF Vectorizer

Data teks memerlukan persiapan khusus sebelum dapat digunakan untuk pemodelan prediktif. Teks harus diurai untuk menghapus kata yang disebut *tokenization* yang kemudian kata-kata tersebut perlu diekode sebagai bilangan bulat atau nilai *floating point* yang digunakan sebagai input pada algoritma *machine learning* yang disebut *feature extraction*. Salah satu metode yang efektif dalam menghitung frekuensi kata-kata tersebut adalah TF-IDF (*Term Frequency – Inverse Document*).

TF-IDF dapat digunakan menggunakan model `TfidfVectorizer` dari *library Scikit-learn*. `TfidfVectorizer` akan memberikan token pada dokumen, mempelajari kosakata dan membalikkan pembobotan frekuensi dokumen, serta digunakan untuk encode dokumen baru. [4]

3.6. **Word Embedding**

Word Embedding adalah teknik mengubah sebuah kata menjadi sebuah vektor atau array yang terdiri dari kumpulan angka dengan memperhatikan relasi antara satu kata dengan kata yang lain. Salah satu metode Word Embedding adalah FastText.

FastText adalah pengembangan dari library Word2Vec yang telah lama dikenal sebagai library untuk word embedding. Salah satu kemampuan FastText dibanding library word embedding sebelumnya adalah dapat menangani kata yang tidak pernah dijumpai sebelumnya (*Out of Vocabulary Word* atau dikenal OOV). [5]

3.7. **Ekspansi Kueri**

Ekspansi kueri (QE) adalah proses memformulasi ulang kueri yang diberikan untuk meningkatkan kinerja pengambilan dalam operasi pengambilan informasi khususnya dalam konteks pemahaman kueri. Dalam konteks *search engine*, perluasan kueri melibatkan evaluasi masukan pengguna dan memperluas kueri penelusuran agar cocok dengan dokumen tambahan. Perluasan kueri melibatkan teknik seperti menemukan sinonim kata, kata-kata yang berhubungan secara semantik, dan lain-lain. [6]

3.8. Anaconda

Anaconda adalah distribusi gratis dan *open source* dari bahasa pemrograman Python dan R yang digunakan untuk komputasi ilmiah (*data science, machine learning*, pemrograman data skala besar, dan lain-lain) yang bertujuan untuk menyederhanakan manajemen paket dan penyebaran. Distribusi mencakup paket *data science* yang sesuai untuk Windows, Linux, dan macOS. [7]

3.9. Python

Python adalah bahasa pemrograman tingkat tinggi yang ditafsirkan, berorientasi objek, serta memiliki semantik dinamis. Struktur data bawaan tingkat tinggi yang dikombinasikan dengan pengetikan dan pengikatan dinamis membuat Python sangat menarik pada *Rapid Application Development* serta digunakan sebagai bahasa skrip atau perekat untuk menghubungkan komponen yang ada secara bersamaan. Sintaks Python yang sederhana dan mudah dipelajari menekankan keterbacaan yang dapat mengurangi biaya pemeliharaan program. Python mendukung model dan paket yang mendorong modularitas program dan penggunaan kode kembali. [8]

3.10. Django

Django adalah kerangka kerja web Python tingkat tinggi yang mendorong pengembangan aplikasi yang cepat dan bersih serta desain yang pragmatis. Django dibangun oleh pengembang berpengalaman yang dapat menangani banyak kerumitan pengembangan web dapat fokus terhadap penggunaan aplikasi. Penggunaan Django dapat digunakan secara gratis dan *open source*. [9]

3.11. HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.[10]

3.12. CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. [11]

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

3.13. MySQL

MySQL merupakan salah satu sistem manajemen database relasional yang bersifat *open source*. Database relasional mengatur data dalam satu atau lebih tabel data di mana tipe data mungkin terkait satu sama lain yang membantu dalam penyusunan data. SQL adalah bahasa pemrograman yang digunakan untuk membuat, memodifikasi, dan mengekstrak data dari database relasional, serta mengontrol akses pengguna ke database. [12]

Selain database relasional dan SQL, RDBMS seperti MySQL bekerja dengan sistem operasi untuk mengimplementasikan database relasional dalam sistem penyimpanan komputer, mengelola pengguna, memungkinkan akses jaringan, serta memfasilitasi pengujian integritas database dan pembuatan cadangan.

3.14. Web Hosting

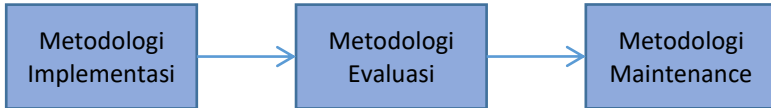
Hosting atau web hosting adalah sebuah tempat di mana file dan data yang diperlukan website disimpan serta dapat diakses dan dikelola melalui internet. File website yang disimpan pada hosting berupa video, gambar, email, script, aplikasi dan database. [13]

3.15. Visual Studio Code

Visual Studio Code adalah editor kode yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS. Visual Studio Code memberikan dukungan untuk debugging, kontrol Git tertanam, penyorotan sintaksis, penyelesaian kode cerdas, snippet, dan refactoring kode. Visual Studio Code juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, pintasan keyboard, dan preferensi.

[Halaman ini sengaja dikosongkan]

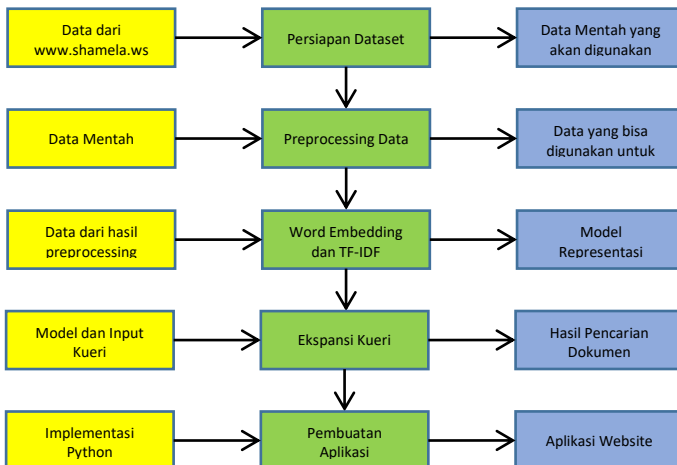
BAB IV METODOLOGI SISTEM



Gambar 4.1. Diagram Metodologi Sistem

Metodologi sistem pada proyek ini terbagi menjadi tiga tahapan, yaitu metodologi implementasi, metodologi evaluasi, dan metodologi maintenance.

4.1. Metodologi Implementasi



Gambar 4.2. Diagram Metodologi Implementasi

Metodologi ini membahas tahapan pembangunan aplikasi secara utuh dan dapat digunakan. Diagram metodologi sistem dapat dilihat pada gambar 4.2.

4.1.1. Persiapan Dataset

Dataset diperoleh dari website www.shamela.ws dengan bahasa Arab yang menyediakan berkas buku elektronik hingga puluhan ribu kita. Dataset yang digunakan terdiri dari 2 folder yang mendeskripsikan kategori pada dokumen. Kategori tersebut adalah Fiqih Syafi'I dan Fiqih Syafi'I Murqim Aliya. Setiap folder terdiri dari file-file .csv yang mendeskripsikan dokumen-dokumen referensi tersebut. Gambaran dataset dapat ditunjukkan pada gambar 4.3.

No Database	Kitab	Kategori	Nama Kitab	Pengarang	No Halaman	Teks
0	10025	الأجزاء الحديثية	التكثرة للمصنفي	محمد بن قنوح بن عبد الله بن فروح بن حميد الأن	(1/374)	... اخبار كتبها ابو نصر المصنفي ومعه شيوخه . اله
1	10025	الأجزاء الحديثية	التكثرة للمصنفي	محمد بن قنوح بن عبد الله بن فروح بن حميد الأن	(1/376)	... الرحيم ابن كريم الشيع الملقبه عماد بقرامني الع
2	10025	الأجزاء الحديثية	التكثرة للمصنفي	محمد بن قنوح بن عبد الله بن فروح بن حميد الأن	(1/377)	... رسول صلي وسلم : الدار نمره ، اجنوا طيبه . يت :
3	10025	الأجزاء الحديثية	التكثرة للمصنفي	محمد بن قنوح بن عبد الله بن فروح بن حميد الأن	(1/378)	... ابو : : السلي : ابن ومحمد سهلويه وابو الحاد -
4	10025	الأجزاء الحديثية	التكثرة للمصنفي	محمد بن قنوح بن عبد الله بن فروح بن حميد الأن	(1/379)	... حجاج ، ، افريش ، بنسبه ، يوسف مسلم جريج : ا ،

Gambar 4.3. Dataset Arabic

Pada dataset yang telah terhimpun terdapat 94.341 record dokumen. Atribut yang terdapat dalam dataset terdiri dari no. database kitab, kategori, nama kitab, pengarang, no. halaman, dan teks.

4.1.2. Preprocessing Data

Tahapan preprocessing data dilakukan dengan membersihkan data mentah ke dalam bentuk yang berkualitas dengan format yang tepat. Berikut adalah penjelasan tahapan preprocessing data secara lebih detail.

4.1.2.1. Remove Diacritics

Proses ini dilakukan dengan menghilangkan diacritics pada teks Arabic. Contoh diacritics Arabic dapat ditunjukkan pada gambar 4.4.

Shaddah	Sukun	Tanwin	Dammah	Fathah	Kasrah
◌ّ	◌◌	◌◌◌ ◌◌◌	◌◌◌	◌◌◌	◌◌◌
-	-	- - - ◌◌◌	-	-	-

Gambar 4.4. Diacritics Arabic

4.1.2.2. Normalization

Proses ini dilakukan dengan mentransformasikan huruf Arabic ke dalam bentuk tunggal.

4.1.2.3. Remove Punctuations

Proses ini dilakukan dengan menghilangkan simbol-simbol pada teks Arabic.

4.1.2.4. Remove Repeating Chars

Proses ini dilakukan dengan menghilangkan karakter berulang pada teks Arabic.

4.1.2.5. Remove Alphabet and Numeric

Proses ini dilakukan dengan menghilangkan karakter berupa angka dan huruf alfabet pada teks Arabic.

4.1.2.6. Remove Stopword

Proses ini dilakukan dengan menghilangkan stopword yang merupakan kata umum yang biasa muncul dan teks bermakna pada teks Arabic. Library yang digunakan adalah CLTK (*The Classical Language Toolkit*) dengan package yang digunakan adalah `stopwords_filter` khusus untuk stopword bahasa Arab.

4.1.2.7. Remove Stemming

Proses ini dilakukan dengan mentransformasikan kata pada teks Arabic ke dalam bentuk kata dasar. Library yang digunakan adalah Tashpyne dengan package yang digunakan adalah `ArabicLightStemmer` khusus untuk stemming bahasa Arab.

4.1.2.8. Tokenizing

Proses ini dilakukan dengan membagi teks yang berupa teks Arabic ke dalam bentuk token-token atau bagian-bagian tertentu. Teks dari hasil preprocessing yang

digunakan berisi minimal 3 kata dari hasil tokenizing agar hasil pencarian dokumen lebih akurat.

4.1.3. Word Embedding

Word Embedding digunakan untuk menangkap konteks kata dalam dokumen untuk dikonversi ke dalam bentuk *term vector* dengan memperhatikan makna semantic serta relasi satu kata dengan kata yang lain. Terdapat dua Word Embedding yang digunakan, yaitu FastText dan TF-IDF.

FastText digunakan untuk mencari daftar kata yang memiliki tingkat kemiripan tinggi pada input kueri. Sedangkan TF-IDF digunakan untuk mencocokkan kata pada kueri untuk menemukan dokumen yang relevan pada input kueri.

4.1.4. Proses Kueri

Tahapan ini dilakukan dengan memilih dan menambahkan istilah pada kueri pengguna untuk menampilkan output berupa hasil dokumen yang memiliki tingkat *similarity* paling tinggi dengan input kueri. Berikut adalah penjelasan tahapan proses kueri secara lebih detail.

4.1.4.1. Input Kueri

Proses ini dilakukan dengan memberikan input kueri dalam bentuk bahasa Indonesia.

4.1.4.2. Preprocessing Kueri

Proses ini dilakukan dengan melakukan preprocessing kueri ke dalam bentuk yang dapat digunakan untuk proses selanjutnya. Preprocessing yang digunakan adalah *remove stopword*, *casefolding*, *remove digit and alphabet*, dan *tokenizing*.

4.1.4.3. Translasi Kueri

Hasil kueri yang telah dipreprocessing kemudian ditranslasi ke dalam bahasa Arab. Library yang digunakan untuk translasi kueri adalah *googletrans* dengan package yang digunakan adalah *Translator*.

4.1.4.4. Ekspansi Kueri

Pada proses ini dilakukan penambahan maupun pemilihan kueri pengguna untuk meminimalkan ketidakcocokan dokumen dan meningkatkan performa *information retrieval*. Ekspansi kueri menggunakan hasil model *Word Embedding* *FastText* dengan output berupa 4 daftar kata yang memiliki kemiripan pada setiap kata dalam input kueri.

4.1.4.5. Pencarian Dokumen

Pada proses ini dilakukan untuk menampilkan hasil pencarian dokumen yang memiliki kemiripan tertinggi dengan hasil ekspansi kueri. Model yang

digunakan adalah hasil dari TF-IDF Vectorizer.

4.1.5. Pembuatan Aplikasi

Tahap pembuatan aplikasi merupakan tahap untuk mengimplementasikan hasil aplikasi sebelumnya ke dalam bentuk website. Terdapat 3 fitur utama pada aplikasi website, yaitu.

1. Kueri. Pada fitur ini ditampilkan suatu box input kueri untuk input dari pengguna. Tampilan output berupa list dokumen kitab referensi yang memiliki kemiripan paling tinggi dengan input kueri pengguna sebelumnya. Pada bagian ini juga terdapat review hasil pencarian dokumen dari pengguna apakah output dokumen yang ditampilkan relevan atau tidak.
2. Login/Sign up. Pada fitur ini berisi pendaftaran dan login dari pengguna untuk pencatatan data pengguna yang menggunakan aplikasi website.
3. Dataset Kitab. Pada fitur ini berisi upload dokumen kitab dari pengguna untuk menambahkan referensi yang kemudian dapat digunakan untuk output kueri kedepannya.

Aplikasi website dibuat dengan menggunakan *framework* Django berbasis Python. Aplikasi website dideploy menggunakan web hosting untuk penyimpanan database.

4.2. Metodologi Evaluasi

4.2.1. Uji Coba Sistem terhadap Ground Truth

Pada tahap ini dilakukan persiapan kueri uji ground truth dan list dokumen yang relevan terhadap kueri. Kueri uji yang diperoleh pada tahap ini adalah 50 kueri uji *ground truth* yang berhubungan dengan topik Fiqih Syafi’I yang diperoleh sesuai dengan topik pada website Piss-KTB. Setiap kueri uji *ground truth* akan disediakan dokumen-dokumen yang relevan dengan kueri uji *ground truth*.

Pengukuran efektivitas pencarian dokumen yang relevan terhadap kueri menggunakan *recall* dan *precision*. *Precision* dan *recall* diperoleh dari hasil uji coba ground truth. Relevansi dokumen yang ditampilkan dengan input kueri diperoleh berdasarkan kesesuaian topik dari input kueri dengan dokumen kitab referensi yang ditampilkan. Jika sesuai, maka dapat dikatakan bahwa dokumen yang ditampilkan tersebut relevan.

Berikut adalah formula *recall* dan *precision* pada *information retrieval* yang ditunjukkan pada gambar 4.5.

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{documents retrieved}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{documents retrieved}\}|}{|\{\text{documents retrieved}\}|}$$

Gambar 4.5. Formula *Recall* dan *Precision*

4.2.2. Uji Coba Fungsionalitas Aplikasi

Pada tahap ini akan dilakukan uji coba fungsionalitas untuk mengetes apakah aplikasi berjalan dengan baik atau tidak. Kemudian dilakukan evaluasi terhadap kesalahan atau kekurangan pada aplikasi. Terdapat 7 uji coba fungsionalitas yang perlu dievaluasi pada aplikasi yang dibuat, yaitu sebagai berikut.

1. Akses halaman utama aplikasi.
2. Registrasi akun.
3. Login akun.
4. Hasil search engine pada input kueri yang diberikan tanpa ekspansi kueri.
5. Hasil search engine pada input kueri yang diberikan dengan ekspansi kueri.
6. Review hasil pencarian dokumen.
7. Upload kitab dokumen.

4.3. Metodologi Maintenance

Metodologi maintenance merupakan tahapan pemeliharaan maupun perawatan fasilitas dan infrastruktur agar selalu dalam keadaan siap pakai pada aplikasi secara efektif dan efisien sesuai dengan standar fungsionalitas aplikasi.

Tahapan ini juga dilakukan penambahan dokumen pada proses implementasi agar referensi lebih lengkap dan meningkatkan hasil relevansi dokumen yang diretrieve.

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem yang penulis buat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian implementasi Anaconda, implementasi website, dan implementasi antarmuka pengguna.

5.1. Implementasi Anaconda

Implementasi Anaconda berisi proses pembuatan model yang akan digunakan untuk fitur *search engine*, yaitu model FastText dan TF-IDF Vectorizer. Proses implementasi Anaconda terbagi menjadi *import library*, preprocessing data, word embedding dan FastText, *search engine*, dan pembuatan kueri uji ground truth.

5.1.1. Import Library

Proses ini berisi *import library-library* yang akan digunakan pada proses implementasi.

```
# Import library-library
import os
import glob
from tqdm import tqdm

# Data Preparation and Preprocessing
import pandas as pd
import re
from cltk.stop.arabic.stopword_filter import
stopwords_filter as ar_stop_filter
from string import digits
from tashaphyne.stemming import ArabicLightStemmer

# Word Embedding
```

```

from sklearn.externals import joblib
from sklearn.feature_extraction.text import
TfidfVectorizer
import scipy.sparse

from sklearn.metrics.pairwise import
cosine_similarity

import matplotlib.pyplot as plt

# Input and Expansion Query
from textblob import TextBlob
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.tokenize import wordpunct_tokenize
from nltk.corpus import stopwords
from googletrans import Translator

```

5.1.2. Preprocessing Data

Proses ini berisi pembersihan data mentah ke dalam bentuk data yang berkualitas dengan format yang tepat. Pembersihan data yang dilakukan berupa *remove diacritics*, normalisasi kata Arabic, *remove punctuations*, *remove repeating chars*, *remove digits*, dan *remove stopwords*.

```

arabic_diacritics = re.compile("""
                                َ | # Tashdid
                                ُ | # Fatha
                                ِ | # Tanwin Fath
                                ِ | # Damma
                                ِ | # Tanwin Damm

```

```

        ◌ | # Kasra
        ◌ | # Tanwin Kasr
        ◌ | # Sukun
        adihsaK/liwtaT #
    """, re.VERBOSE)

# Remove Diacritics
def remove_diacritics(text):
    text = re.sub(arabic_diacritics, '', text)
    return text

# Normalisasi Text
def normalizeArabic(text):
    text = re.sub("[\u0305\u0306\u0307\u0308\u0309]", "", text)
    text = re.sub("ى", "ي", text)
    text = re.sub("ؤ", "ء", text)
    text = re.sub("غ", "ع", text)
    text = re.sub("ه", "ه", text)
    text = re.sub("ي", "ك", text)
    return text

# Remove Punctuations
def remove_punctuations(text):
    text = re.sub("`", "", text)
    text = re.sub("÷", "", text)
    text = re.sub("×", "", text)
    text = re.sub("_", "", text)
    text = re.sub("-", "", text)
    text = re.sub("\\", "", text)
    text = re.sub("...", "", text)
    text = re.sub("!", "", text)
    text = re.sub("|", "", text)
    text = re.sub("!", "", text)
    text = re.sub("~", "", text)
    text = re.sub("{", "", text)

```

```

text = re.sub(")", "", text)
text = re.sub("'", "", text)
text = re.sub("\\.", "", text)
text = re.sub(",", "", text)
text = re.sub("/", "", text)
text = re.sub(":", "", text)
text = re.sub(".", "", text)
text = re.sub("%", "", text)
text = re.sub("^", "", text)
text = re.sub("&", "", text)
text = re.sub("^", "", text)
text = re.sub("\\*", "", text)
text = re.sub(",", "", text)
text = re.sub("\\)", "", text)
text = re.sub("\\(", "", text)
text = re.sub(">", "", text)
text = re.sub("<", "", text)
text = re.sub(":", "", text)
text = re.sub("?", "", text)
text = re.sub(":", "", text)
text = re.sub("\\\"", "", text)
text = re.sub("\\[", "", text)
text = re.sub("\\]", "", text)
text = re.sub("\\+", "", text)
text = re.sub(">>", "", text)
text = re.sub("<<", "", text)

return text

def remove_repeating_char(text):
    return re.sub(r'(\.)\1+', r'\1', text)

def remove_digit(text):
    remove_digits = str.maketrans('', '', digits)
    text = text.translate(remove_digits)

```

```

return text

column = ['No Database Kitab', 'Kategori', 'Nama
Kitab', 'Pengarang', 'No Halaman', 'Teks',]

# Membuat kerangka dataframe
df_total = pd.DataFrame(columns=column)
document_text = []

# Merapikan Data dan Text Processing Menggunakan
Google Colab
filepath = '/content/drive/My Drive/Dataset DLIR'
dirpath = ['ايلى آ م ق ر م يع ف اش ه ق ف', 'يع ف اش ه ق ف']
print(dirpath)

for dir in dirpath:
    print(dir)
    path = os.path.join(filepath, dir, '*.csv')
    files = glob.glob(path)
    for file in tqdm(files):
        df_test = pd.read_csv(file, encoding="Windows-
1256")
        df_test.columns = column

        df = pd.DataFrame(df_test, columns=column)
        df = df.dropna(subset=['Teks'], axis=0)
        for x in range(0, df.shape[0]):
            text = remove_diacritics(df.iloc[x, -1])
            text = normalizeArabic(text)
            text = remove_punctuations(text)
            text = remove_repeating_char(text)
            text = remove_digit(text)
            stopword = ar_stop_filter(text)
            for sw in stopword:

```

```

        if(sw.isalpha() or sw.isdecimal() or
sw.isalnum()):
            stopword.remove(sw)

    hasil = []
    hasil.append(' '.join(stopword))

    if(hasil!=None):
        document_text.append(df.iloc[x, -1])
        df.iloc[x, -1] = hasil
        df_total =
df_total.append(pd.Series(df.iloc[x, :],
index=column), ignore_index=True)

# Print jumlah kategori pada data hasil text
processing
print(df_total.shape)
uniq = df_total.Kategori.unique()
print(uniq)
print("Jumlah Kategori : ", len(uniq))

# Simpan teks asli dokumen
joblib.dump(document_text,
"document_text_syafii.pkl")
!cp document_text_syafii.pkl '/content/drive/My
Drive/Models/Fix/document_text'

# Simpan dataframe hasil text processing sebelum
TF-IDF Vectorizer
df_total.to_csv('df_total_syafii.csv', index=False)
!cp df_total_syafii.csv '/content/drive/My
Drive/Models/Fix/df_total'

```

Data yang telah dilakukan preprocessing sebelumnya masih terdapat teks yang kosong. Oleh karena itu, data dengan teks yang kosong harus dihapus dari dataset agar dapat digunakan pada proses pembuatan model.

```
# Load Dataframe
df_total = pd.read_csv('/content/drive/My
Drive/Models/Fix/df_total/df_total_syafii.csv')
document_text = joblib.load('/content/drive/My
Drive/Models/Fix/document_text/document_text_syafii
.pkl')
df_total.info()

teks = []
for i in range(df_total.shape[0]):
    teks.append(df_total.iloc[i, -1])
print(len(teks))
print(teks[0])

text = []
for data in teks:
    data = wordpunct_tokenize(data)
    text.append(data)
print(text[4])

def isNaN(string):
    return string != string

# Remove NaN from data
index_nan = []
teks = []

for i in range(df_total.shape[0]):
    if(isNaN(df_total.iloc[i, -1])):
```

```

        index_nan.append(i)
    else:
        teks.append(df_total.iloc[i, -1])
df_total = df_total.drop(index_nan, axis=0)

minus_index = 0
for x in range(len(document_text)):
    if(x in index_nan):

document_text.remove(document_text[x+minus_index])
    minus_index -= 1

# Cek panjang dokumen
print("Panjang Hasil Processing Teks : ",
len(teks))
print("Panjang Dokumen Teks : ",
len(document_text))

# Simpan teks asli dokumen
joblib.dump(document_text,
"document_text_syafii.pkl")
!cp document_text_syafii.pkl '/content/drive/My
Drive/Models/Fix/document_text'

# Simpan dataframe hasil text processing sebelum
TF-IDF Vectorizer
df_total.to_csv('df_total_syafii.csv', index=False)
!cp df_total_syafii.csv '/content/drive/My
Drive/Models/Fix/df_total'

```

Setelah dilakukan pembersihan data pada tahap sebelumnya, akan dilakukan pembersihan data lagi menggunakan *stemming* dan mengambil data dengan

hasil preprocessing sebanyak minimal 3 kata pada bagian fitur teks dataset.

```
# Load Dataframe
df_total = pd.read_csv('/content/drive/My
Drive/Models/Fix/df_total/df_total_syafii.csv')
document_text = joblib.load('/content/drive/My
Drive/Models/Fix/document_text/document_text_syafii
.pkl')
df_total.info()

column = ['No Database Kitab', 'Kategori', 'Nama
Kitab', 'Pengarang', 'No Halaman', 'Teks',]

df_total_stem = pd.DataFrame(columns=column)
document_text_stem = []

df = pd.DataFrame(df_total, columns=column)
for x in range(df.shape[0]):
    if(x%10000==0):
        print("Preprocessing stemming : ", x)
        text = df.iloc[x, -1]

        ArListem = ArabicLightStemmer()
        stem = ArListem.light_stem(text)
        hasil = ArListem.get_root()
        token_stem = wordpunct_tokenize(hasil)

        if(len(token_stem) > 3):
            document_text_stem.append(df.iloc[x, -1])
            df.iloc[x, -1] = hasil
            df_total_stem =
df_total_stem.append(pd.Series(df.iloc[x, :],
index=column), ignore_index=True)
```

```

# Print jumlah kategori pada data hasil text
processing
print(df_total_stem.shape)
uniq = df_total_stem.Kategori.unique()
print(uniq)
print("Jumlah Kategori : ", len(uniq))

# Simpan teks asli dokumen
joblib.dump(document_text_stem,
"document_text_syafii_stem.pkl")
!cp document_text_syafii_stem.pkl
'/content/drive/My Drive/Models/Fix/document_text'

# Simpan dataframe hasil text processing sebelum
TF-IDF Vectorizer
df_total_stem.to_csv('df_total_syafii_stem.csv',
index=False)
!cp df_total_syafii_stem.csv '/content/drive/My
Drive/Models/Fix/df_total'

```

```

# Load Dataframe
df_total = pd.read_csv('/content/drive/My
Drive/Models/Fix/df_total/df_total_syafii_stem.csv'
)
document_text = joblib.load('/content/drive/My
Drive/Models/Fix/document_text/document_text_syafii
_stem.pkl')
df_total.info()

print(df_total.shape[0])
for i in range(df_total.shape[0]):
    if(i%10000==0):
        print("Preprocessing : ", i)

```

```

df_total.iloc[i, -1] = document_text[i]

# Simpan dataframe hasil text processing sebelum
TF-IDF Vectorizer
df_total.to_csv('hasil_total_syafii.csv',
index=False)
!cp hasil_total_syafii.csv '/content/drive/My
Drive/Models/Fix/df_total'

```

5.1.3. Word Embedding dan TF-IDF Vectorizer

Proses ini berisi pembentukan model TF-IDF Vectorizer dan FastText yang menggunakan dataset dari hasil preprocessing data sebelumnya.

```

from gensim.models import FastText
modelFT = FastText(text, size=100, window=5,
min_count=2, sg=1)

joblib.dump(modelFT, "modelFT_syafii.pkl")
!cp modelFT_syafii.pkl '/content/drive/My
Drive/Models'

modelFT = joblib.load('/content/drive/My
Drive/Models/Fix/FastText/modelFT_syafii.pkl')

# Fit transform menggunakan TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(teks)
print('tfidf_matrix')
print(tfidf_matrix.shape)

joblib.dump(tfidf_vectorizer,
"vectorizer_syafii.pkl")

```

```
!cp vectorizer_syafii.pkl '/content/drive/My
Drive/Models/Fix/tf-idf'

joblib.dump(tfidf_matrix, "tfidf_syafii.pkl")
!cp tfidf_syafii.pkl '/content/drive/My
Drive/Models/Fix/tf-idf'
```

5.1.4. Search Engine

Proses ini berisi fitur *search engine* yang berfungsi dalam pencarian hasil dokumen Arabic tanpa ataupun dengan menggunakan ekspansi kueri.

```
def preprocessing_query(input_query):
    NLTK_StopWords = stopwords.words('indonesian')
    NLTK_StopWords = set(NLTK_StopWords)
    hasilQuery=[]
    s = input_query
    s = s.lower()
    s = re.sub(r'[^a-zA-Z0-9\s]', ' ', s)
    tokens = [token for token in s.split(" ") if
token != ""]
    T = [t for t in tokens if t not in
NLTK_StopWords]

    # Use the zip function to help us generate n-
grams
    # Concatenate the tokens into ngrams and return
    for i in range (len(T)+1):
        ngrams = zip(*[T[i:] for i in range(i)])
        ngrams = [" ".join(ngram) for ngram in
ngrams]
        if (len(ngrams)!=0):
            if (ngrams not in hasilQuery):
                hasilQuery.append(ngrams)
```

```

    return hasilQuery

# Query Translation
def query_translation(hasilQuery):
    translator = Translator()
    hasilTranslate=[]
    for a in hasilQuery:
        for b in a:
            # print (b)
            translations = translator.translate(b,
dest='ar')
            # print (translations.text)
            hasilTranslate.append(translations.text)
    print ("Query Translation : ", hasilTranslate[-
1])
    return hasilTranslate[-1]

# Local Search Engine
def search_engine():
    input_query = input()
    qe = int(input())

    print("Input query: ", input_query)
    hasilQuery = preprocessing_query(input_query)
    print("Preprocessing query: ", hasilQuery[-1])
    query = query_translation(hasilQuery)
    print("Query translation: ", query)

    ArListem = ArabicLightStemmer()
    stem = ArListem.light_stem(query)
    hasil = ArListem.get_root()
    print("Stem: ", hasil)

# Query Expansion
if(qe==1):

```

```

token = wordpunct_tokenize(hasil)
query = []
for word in token:
    pq = modelFT.wv.most_similar(word)
    print(pq)
    words = []
    for i in range(4):
        words.append(pq[i][0])
    words.append(word)
    print(words)

    query.append(' '.join(words))
queries = []
queries.append(' '.join(query))
print("Query Expansion: ", queries)
hasil = queries[0]

# # Translation Expansion Query
# query_expansion_translation(words)

# Find Document
print("Hasil : ", [hasil])
query_vec = tfidf_vectorizer.transform([hasil])
print(query_vec)

results =
cosine_similarity(tfidf_matrix, query_vec).reshape((-1,))

print("\n===== \n")
print("Top 10 most similar documents in corpus:")

for i in results.argsort()[-10:][::-1]:
    print("No ID Dokumen : ", i)

```

```

print("Nomor Database : ", df_total.iloc[i,0])
print("Kategori      : ", df_total.iloc[i,1])
print("Nama Kitab    : ", df_total.iloc[i,2])
print("Pengarang     : ", df_total.iloc[i,3])
print("No Halaman    : ", df_total.iloc[i,4])
print("Teks          : ", df_total.iloc[i,5])
print("(Score: %.4f)" % results[i])

```

```
search_engine()
```

5.1.5. Pembuatan Kueri Uji Ground Truth

Proses ini berisi pembuatan kueri uji ground truth yang diambil dari website Piss-KTB agar kueri uji yang dibuat dapat dijadikan sebagai acuan dalam tahap evaluasi.

```

import csv
import os
import glob
from tqdm import tqdm

def cari(k, h):

    filepath = '/content/drive/My Drive/Dataset DLIR'
    kitabb=["ه قفلا لوصأ", "ة يه قفلا دع او قلا و ه قفلا لوصأ",
    "ء ازج ألأ", "آليا" والقواعد الفقهية مرقم
    "بدألأ", "ايلآ مرقم ة يثي دحلأ ء ازج ألأ", "الحديثية"
    "مرقم ة غ البلا و بدألأ", "والبد لاغة"
    "ن ادلبلا", "ايلآ مرقم باسن ألأ", "باسن ألأ", "آليا"
    "تالحرلا و ايفارغجلا و ن ادلبلا", "والجغرافيا والرحلات"
    "ديوجتلا", "ايلآ مرقم خيراتلا", "خيراتلا", "مرقم آليا"
    "تاقبطل او مج ارتلا", "والطبقات" مج ارتلا", "والقراءات"
    "عم او جلا", "ايلآ مرقم ريسافتلا", "ريسافتلا", "مرقم آليا"
    "ة و عدلا", "نيم لسملا ل اوح أو ة و عدلا", "والمجلات ونحوها"

```

ني و او دل "، و احوال المسلمين مرقم آليا
 مرقم ا ه و ح ن و ت ا ل ج م ل ا و ع م ا و ج ل ا "، الشعريه
 باد آ ل ا و ق ا ق ر ل ا "، " ر ا ك ذ ا ل ا و ب ا د آ ل ا و ق ا ق ر ل ا "، آليا
 ة ي ع ر ش ل ا ة س ا ي س ل ا "، الأذكار مرقم آليا و
 مرقم ء اض ق ل ا و ة ي ع ر ش ل ا ة س ا ي س ل ا "، والقضاء
 مرقم لئ ا م ش ل ا و ة ي س ل ا "، " لئ ا م ش ل ا و ة ي س ل ا "، آليا
 ل ل ع ل ا "، " ا ي ل آ م ق ر م ة د ي ق ع ل ا "، " ة د ي ق ع ل ا "، آليا
 ب ي ر غ ل ا "، " ا ي ل آ م ق ر م ت ا ل ا ف س ل ا و ل ل ع ل ا "، و السؤالات
 فقه ل ا ة غ ل و م ج ا ع م ل ا و ب ي ر غ ل ا "، و المعاجم و لغة الفقه
 ق ر ف ل ا "، " ا ي ل آ م ق ر م ي و ا ت ف ل ا "، " ي و ا ت ف ل ا "، مرقم آليا
 و ح ن ل ا "، " ا ي ل آ م ق ر م د و د ر ل ا و ق ر ف ل ا "، و الردود
 ث و ح ب "، " ا ي ل آ م ق ر م ف ر ص ل ا و و ح ن ل ا "، و الصرف
 ح و ر ش "، " ث ي د ح ل ا ح و ر ش "، " ا ي ل آ م ق ر م لئ ا س م و ث و ح ب "، و مسائل
 مرقم ي ر خ ا م و ل ع "، " ي ر خ ا م و ل ع "، الحديث مرقم آليا
 م و ل ع "، الحديث مرقم آليا " م و ل ع "، " ث ي د ح ل ا م و ل ع "، آليا
 ه ق ف "، " ي ل ب ن ح ه ق ف "، " ا ي ل آ م ق ر م ن ا ر ق ل ا م و ل ع "، القرآن
 مرقم - ي ف ن ح ه ق ف "، " ي ف ن ح ه ق ف "، حنبلي مرقم آليا
 ه ق ف "، " ا ي ل آ م ق ر م ي ع ف ا ش ه ق ف "، " ي ع ف ا ش ه ق ف "، آليا
 ب ت ك ل ا س ر ا ه ف "، " ي ك ل ا م ه ق ف "، " ا ي ل آ م ق ر م م ا ع ه ق ف "، عام
 ك ت ب ا ب ن "، " ا ي ل آ م ق ر م ة ل د ا ل ا و ب ت ك ل ا س ر ا ه ف "، و الأدلة
 ب ت ك "، " ة ي م ي ت ن ب ا ب ت ك "، " م ي ق ل ا ن ب ا ب ت ك "، أبي الدنيا
 ب ت ك "، مرقم آليا - ة م ا ع ة ي م ا ل س ا ب ت ك "، إسلامية عامة
 ج ي ر خ ت ل ا ب ت ك "، " ا ي ل آ م ق ر م ي ن ا ب ل ا ل ا ب ت ك "، الألباني
 ب ت ك "، " ا ي ل آ م ق ر م د ئ ا و ز ل ا و ج ي ر خ ت ل ا ب ت ك "، و الزوائد
 ن و ت م "، " ث ي د ح ل ا ن و ت م "، " ا ي ل آ م ق ر م ة غ ل ل ا ب ت ك "، اللغة
] " ة ي ث ي د ح ت ا ط و ط خ م "، " ة غ ر ف م ت ا ر ض ا ح م "، يث مرقم آليا الحد

```
hasil=[]
```

```
line_count = 0
```

```
x = 0
```

```
kitab = k
```

```
hal=h
```

```
print("Kitab : ", kitab)
```

```
for j in tqdm(kitabb):
```

```

path = os.path.join(filepath, j, '*.csv')
files = glob.glob(path)
x += 1
p = 0
nemu = False
for file in files:
    # p += 1
    with open(file, newline='',
encoding="Windows-1256") as csvfile:
        read = csv.reader(csvfile, delimiter=',')
        i = 0
        for row in read:
            namakitab = row[2]
            i += 1
            if(i==2):
                break
        line_count += 1
    if(kitab in namakitab):
        print("Kategori = ",j)
        print("ada nama kitab "+namakitab)
        for row in read:
            nohal = row[4]
            teks = row[5]
            if(nohal==hal):
                print("ada no hal"+nohal)
                for r in range(len(row)):
                    print(row[r])

```

5.2. Implementasi Website

Implementasi website berisi proses pembuatan website pada aplikasi dengan menggunakan framework Django berbasis Python yang didukung oleh komponen dan bahasa pemrograman lainnya seperti HTML, CSS, dan MySQL

Database. Implementasi website memiliki app yang terdiri dari api, predictor, account, dan kitab.

5.2.1. App Api

App api merupakan app utama yang dibangun pada aplikasi website ini.

5.2.1.1 Api urls.py

```
from django.contrib import admin
from django.urls import path, include
from . import views

urlpatterns = [
    path('', views.index),
    path('search/', include('predictor.urls')),
    path('account/', include('account.urls')),
    path('kitab/', include('kitab.urls')),
    path('admin/', admin.site.urls),
]
```

5.2.1.2 Api views.py

```
from django.shortcuts import render

def index(request):
    context = {
        'title': "Kitab Ulama",
    }
    return render(request, 'index.html', context)
```

5.2.1.3 Api templates base.html

```
{% load static %}
```

```

<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>

    {% block head %}
    {% endblock head %}

    {% block styles %}
    {% endblock styles %}

  </head>

  <body>
    {% include "snippets/navbar.html" %}

    {% block search %}
    {% endblock search %}

    {% block content %}
    {% endblock content %}

    {% include "snippets/scripts.html" %}

  </body>
</html>

```

5.2.1.4 Api templates index.html

```

{% extends 'base.html' %}
{% load static %}

{% block styles %}

```

```

<link rel="stylesheet" type="text/css" href="{%
static "css/styles.css" %}">
<link rel="stylesheet" type="text/css" href="{%
static "vendor/bootstrap/bootstrap.css" %}">
<link rel="stylesheet" type="text/css" href="{%
static "vendor/bootstrap/bootstrap.min.css" %}">
{% endblock styles %}

{% block search %}
<div class="d-flex justify-content-center">
<form method="POST" action="/search/">
    {% csrf_token %}
    <h1 style="text-align: center">Dual
Lingual Information Retrieval Indonesia-Arab</h1>
    <label for="exquery" style="text-align:
center">Ekspansi Kueri</label>
    <select id="exquery" name="exquery"
class="justify-content-center" style="text-align:
center">
        <option
value="Tidak">Tidak</option>
        <option value="Iya">Iya</option>
    </select> <br> <br>
    <input type="text" name="textfield"
class="keyboardInput" style="width: 450px;">
    <link rel="stylesheet" type="text/css"
href="http://www.arabic-
keyboard.org/keyboard/keyboard.css">
    <script type="text/javascript"
src="http://www.arabic-
keyboard.org/keyboard/keyboard.js" charset="UTF-8">
    </script>
    <br><br>
    <button type="submit">Search</button>
</form>

```

```

</div>
{% endblock search %}

```

5.2.1.5 Api templates snippets form_styles.html

```

{% load static %}

<link rel="stylesheet" type="text/css" href="{%
static "vendor/bootstrap/bootstrap.min.css" %}">
<link
href="https://cdnjs.cloudflare.com/ajax/libs/simple
-line-icons/2.4.1/css/simple-line-icons.min.css"
rel="stylesheet">

```

5.2.1.6 Api templates snippets scripts.html

```

{% load static %}
<script src="{% static
"vendor/bootstrap/bootstrap.js" %}"
type="text/javascript"></script>
<script src="{% static "vendor/jquery/jquery.js"
%}" type="text/javascript"></script>
<script src="{% static
"vendor/jquery/jquery.mask.min.js" %}"
type="text/javascript"></script>
<script src="{% static
"vendor/jquery/jquery.min.js" %}"
type="text/javascript"></script>
<script src="{% static "vendor/popper/popper.js"
%}" type="text/javascript"></script>

```

5.2.1.7 Api templates snippets navbar.html

```

<nav class="navbar navbar-expand-sm navbar-dark"
id="myNav" style='background-color:#3CB043;'>
  <button class="navbar-toggler" type="button"
data-toggle="collapse" data-
target="#navbarSupportedContent"
  aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
  <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse"
id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item">
        <a class="nav-link" href="/"
title="">Home</a>
      </li>

      {% if user.is_authenticated %}
      <li class="nav-item">
        <a class="nav-link" href="/kitab"
title="">Upload Kitab</a>
      </li>

      <li class="nav-item">
        <a class="nav-link"
href="/search/review" title="">Review Hasil
Pencarian Dokumen</a>
      </li>

      {% endif %}
    </ul>
    <ul class="navbar-nav ml-auto">

```

```

        {% if user.is_authenticated %}
        <li class="nav-item">
            <a class="nav-link"
href="/account/logout" title="">Logout</a>
        </li>
        {% else %}
        <li class="nav-item">
            <a class="nav-link"
href="/account/login" title="">Login</a>
        </li>
        <li class="nav-item">
            <a class="nav-link"
href="/account/register" title="">Register</a>
        </li>
        {% endif %}

    </ul>
</div>
</nav>

```

5.2.2. App Predictor

App predictor merupakan app yang berfungsi untuk proses pencarian hasil dokumen dengan memberikan masukan kueri pada fitur *search bar*.

5.2.2.1 Predictor urls.py

```

from django.urls import path

from . import views

urlpatterns = [
    path('', views.index),
    path('review', views.review)

```

```
1
```

5.2.2.2 Predictor apps.py

```
from django.apps import AppConfig
from django.conf import settings

import os
import joblib
from gensim.models import FastText

class PredictorConfig(AppConfig):
    name = 'predictor'
    path_modelFT = os.path.join(settings.MODELS,
    'modelFT_syafii.pkl')
    path_tfidf = os.path.join(settings.MODELS,
    'tfidf_syafii.pkl')
    path_vectorizer = os.path.join(settings.MODELS,
    'vectorizer_syafii.pkl')

    modelFT = joblib.load(path_modelFT)
    tfidf_matrix = joblib.load(path_tfidf)
    tfidf_vectorizer = joblib.load(path_vectorizer)
```

5.2.2.3 Predictor views.py

```
from django.shortcuts import render
from django.http import HttpResponseRedirect

from .forms import ReviewForm
from .models import Review

from .apps import PredictorConfig
```

```

from .models import Kitabs

import os
import joblib
from tashaphyne.stemming import ArabicLightStemmer
from sklearn.metrics.pairwise import
cosine_similarity

# input and expansion query
import pandas as pd
import re
from textblob import TextBlob
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from googletrans import Translator
from nltk.tokenize import wordpunct_tokenize

def index(request):
    # pass
    def preprocessing_query(input_query):
        NLTK_StopWords =
stopwords.words('indonesian')
        NLTK_StopWords = set(NLTK_StopWords)
        hasilQuery=[]
        s = input_query
        s = s.lower()
        s = re.sub(r'^a-zA-Z0-9\s|', ' ', s)
        tokens = [token for token in s.split(" ")
if token != ""]
        T = [t for t in tokens if t not in
NLTK_StopWords]

```

```

# Use the zip function to help us generate
n-grams
# Concatentate the tokens into ngrams and
return
for i in range (len(T)+1):
    ngrams = zip(*[T[i:] for i in
range(i)])
    ngrams = [" ".join(ngram) for ngram in
ngrams]
    if (len(ngrams)!=0):
        if (ngrams not in hasilQuery):
            hasilQuery.append(ngrams)
return hasilQuery

# Query Translation
def query_translation(hasilQuery):
    translator = Translator()
    hasilTranslate=[]
    for a in hasilQuery:
        for b in a:
            # print (b)
            translations =
translator.translate(b, dest='ar')
            # print (translations.text)
hasilTranslate.append(translations.text)
print ("Query Translation : ",
hasilTranslate[-1])
return hasilTranslate[-1]

# Local Search Engine
def search_engine(search_id):
    print("Input query: ", search_id)
    hasilQuery = preprocessing_query(search_id)

```

```

1))
    print("Preprocessing query: ", hasilQuery[-
query = query_translation(hasilQuery)
    print("Query translation: ", query)

    ArListem = ArabicLightStemmer()
    stem = ArListem.light_stem(query)
    hasil = ArListem.get_root()
    print("Stem: ", hasil)

    exquery = request.POST.get('exquery', None)
    print(exquery)

    # Query Expansion
    if(exquery=='Iya'):
        print("Pakai Ekspansi Query")
        # pass
        token = wordpunct_tokenize(hasil)
        query = []
        for word in token:
            pq =
PredictorConfig.modelFT.wv.most_similar(word)
            print(pq)
            words = []
            for i in range(4):
                words.append(pq[i][0])
            words.append(word)
            print(words)

            query.append(' '.join(words))
            queries = []
            queries.append(' '.join(query))
            print("Query Expansion: ",
queries)

            hasil = queries[0]

```

```

        query_vec =
PredictorConfig.tfidf_vectorizer.transform([hasil])

        print(query_vec)

        results =
cosine_similarity(PredictorConfig.tfidf_matrix, query_vec).reshape((-1,))

        list_object = []
        list_id = results.argsort()[-10:][::-1]
        list_id = [x+1 for x in list_id]
        for x in list_id:

list_object.append(Kitabs.objects.get(id=x))

        return list_object

    if request.method == 'POST':
search_id = request.POST.get('textfield',
None)
        print(search_id)
        list_object = search_engine(search_id)
        context = {
            'title': search_id + " | Hasil
Pencarian Kitab Ulama",
            'Kitabs': list_object,
        }
        return render(request, 'search/index.html',
context)
    else:
        context = {
            'title': 'Kitab Ulama',
        }

```

```

        return render(request, 'index.html')

def review(request):
    form_field = ReviewForm()

    context = {
        'title' : 'Review Hasil Pencarian Dokumen |
Kitab Ulama',
        'review_form' : form_field,
    }

    if(request.method=="POST"):
        print(request.POST)

        Review.objects.create(
            username = request.user,
            query = request.POST.get("Query"),
            review = request.POST.get("review"),
            dokumen_relevan =
request.POST.get('relevan')
        )
        return HttpResponseRedirect("/")

    return render(request, 'search/review.html',
context)

```

5.2.2.4 Predictor forms.py

```

from django import forms

class ReviewForm(forms.Form):
    Query = forms.CharField(max_length=100,
widget=forms.TextInput(attrs={

```

```

'class': 'form-control item',
)))
    review = forms.CharField(max_length=500,
widget=forms.Textarea(attrs={

'class': 'form-control item',

)))
    # CHOICE = [('Iya', 'Iya'), ('Tidak', 'Tidak')]
    # ANGKA = [(x, str(x)) for x in range(0, 11)]
    ANGKA = ((0, '0'), (1, '1'), (2, '2'), (3,
'3'), (4, '4'), (5, '5'), (6, '6'), (7, '7'), (8,
'8'), (9, '9'), (10, '10'))
    # ANGKA = range(0, 11, 1)
    relevan = forms.ChoiceField(choices=ANGKA,
label='Berapa jumlah dokumen yang relevan dari
hasil pencarian dokumen?')

```

5.2.2.5 Predictor admin.py

```

from django.contrib import admin
from .models import Kitabs, Review

# Register your models here.
admin.site.register(Kitabs)
admin.site.register(Review)

```

5.2.2.6 Predictor models.py

```

from django.db import models

# Create your models here.

```

```

class Kitabs(models.Model):
    no_database = models.IntegerField()
    kategori = models.CharField(max_length=100)
    nama_kitab = models.CharField(max_length=1000)
    pengarang = models.CharField(max_length=1000)
    no_halaman = models.TextField()
    teks_processing = models.TextField()

    def __str__(self):
        return "{}. {} {}".format(self.id,
self.nama_kitab, self.no_halaman)

class Review(models.Model):
    username = models.CharField(max_length=100)
    query = models.CharField(max_length=100)
    review = models.CharField(max_length=500)

    CHOICE = [('Iya', 'Iya'), ('Tidak', 'Tidak')]

    dokumen_relevan =
models.IntegerField(default=0)
    konfirmasi = models.BooleanField(default=False)

    def __str__(self):
        return "{}. {}".format(self.id, self.query)

```

5.2.2.7 Predictor templates index.html

```

{% extends 'base.html' %}

{% load static %}

{% block styles %}
    {% include "snippets/form_styles.html" %}

```

```

{% endblock styles %}

{% block search %}
<div class="d-flex justify-content-center">
  <form method="POST" action="/search/">
    {% csrf_token %}
    <h1>Dual Lingual Information Retrieval
Indonesia-Arab</h1>
    <label for="exquery">Ekspansi Kueri</label>
    <select id="exquery" name="exquery">
      <option value="Tidak">Tidak</option>
      <option value="Iya">Iya</option>
    </select> <br> <br>
    <input type="text" name="textfield"
class="keyboardInput" style="width: 450px;">
    <link rel="stylesheet" type="text/css"
href="http://www.arabic-
keyboard.org/keyboard/keyboard.css">
    <script type="text/javascript"
src="http://www.arabic-
keyboard.org/keyboard/keyboard.js" charset="UTF-
8"></script>
    <br><br>
    <button type="submit">Search</button>
  </form>
</div>
{% endblock search %}

{% block content %}
<br>
<br>
<form>

    {% for kitab in Kitabs%}

```

```

        <div class="jumbotron jumbotron-fluid"
style="background-color: #90EE90;"
        <div class="container">
            <h4>Kategori :
{{kitab.kategori}}</h4>
            <h4>Nama Kitab :
{{kitab.nama_kitab}}</h4>
            <h4>Pengarang :
{{kitab.pengarang}}</h4>
            <h4>Nomor Halaman :
{{kitab.no_halaman}}</h4>
            <hr class="my-4">
            <h4>Teks</h4>
            <p>{{kitab.teks_processing}}</h4>
            <br>
        </div>
    </div>
    {% endfor %}
</form>
{% endblock content %}

```

5.2.2.8 Predictor templates review.html

```

{% extends 'base.html' %}

{% load static %}

{% block head %}
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content="width=device-
width, initial-scale=1">
{% endblock head %}

```

```

{% block styles %}
    {% include "snippets/form_styles.html" %}
    <link rel="stylesheet" type="text/css" href="{%
static "predictor/css/styles.css" %}">
{% endblock styles %}

{% block content %}
    <div class="registration-form">
        <form method="post">
            {% csrf_token %}
            <h1 style='text-align: center;'> Review
Hasil Pencarian Dokumen </h1>
            <br>
            <div class="form-icon"
style='background-color: #32CD32;'>
                <span><i class="icon icon-book-
open"></i></span>
            </div>
            {% for field in review_form %}
                <div class="form-group">
                    {{ field.label_tag}}
                    {{ field }}
                </div>
            {% endfor %}
            <div class="form-group">
                <button type="submit" class="btn
btn-block register" style="background-
color:#32CD32;">Kirim Review</button>
            </div>
        </form>
    </div>
{% endblock content %}

```

5.2.2.9 Predictor static css styles.css

```
html, body{ background-color: #98FB98;}

.registration-form{
    padding: 50px 0;
}

.registration-form form{
    background-color: #fff;
    max-width: 600px;
    margin: auto;
    padding: 50px 70px;
    border-top-left-radius: 30px;
    border-top-right-radius: 30px;
    box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.075);
}

.registration-form .form-icon{
    text-align: center;
    background-color: #5891ff;
    border-radius: 50%;
    font-size: 40px;
    color: white;
    width: 100px;
    height: 100px;
    margin: auto;
    margin-bottom: 50px;
    line-height: 100px;
}

.registration-form .item{
    border-radius: 20px;
    margin-bottom: 25px;
    padding: 10px 20px;
```

```

}

.registration-form .register{
    border-radius: 30px;
    padding: 10px 20px;
    font-size: 18px;
    font-weight: bold;
    background-color: #5791ff;
    border: none;
    color: white;
    margin-top: 20px;
}

```

5.2.3. App Account

App account merupakan app yang berfungsi untuk fitur pengelolaan akun seperti fitur login dan registrasi.

5.2.3.1 Account urls.py

```

from django.urls import path

from . import views

urlpatterns = [
    path('', views.index),
    path('login', views.logins),
    path('register', views.register),
    path('logout', views.logout_view)
]

```

5.2.3.2 Account views.py

```
from django.contrib.auth import login,
authenticate, logout
from django.contrib.auth.forms import
UserCreationForm
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import
login_required

from .forms import LoginForm
from .forms import RegisterForm

def index(request):
    pass

def logins(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            # form.save()
            username =
form.cleaned_data.get('username')
            password =
form.cleaned_data.get('password')
            user = authenticate(username=username,
password=password)
            login(request, user)
            context = {
                'title': "Kitab Ulama",
            }
            return redirect('/', context)
        else:
            form = LoginForm()
            context = {
                'title': "Login | Kitab Ulama",
                'form': form,
```

```

    }
    return render(request, 'account/login.html',
context)

# @login_required
def register(request):
    form = RegisterForm(request.POST or None)
    if request.method == 'POST':
        if form.is_valid():
            form.save()
            username =
form.cleaned_data.get('username')
            raw_password =
form.cleaned_data.get('password1')
            user = authenticate(username=username,
password=raw_password)
            login(request, user)
            context = {
                'title': "Kitab Ulama",
            }
            return redirect('/', context)
        else:
            form = RegisterForm()
            context = {
                'title': "Registrasi | Kitab Ulama",
                'form': form,
            }
            return render(request, 'account/register.html',
context)

def logout_view(request):
    logout(request)
    return redirect('/')

```

5.2.3.3 Account forms.py

```
from django import forms
from django.contrib.auth.forms import
UserCreationForm
from django.contrib.auth.models import User

class LoginForm(forms.Form):
    username = forms.CharField(max_length=100,
widget=forms.TextInput(attrs={'placeholder':
'Username',

'class': 'form-control item',

}))
    password =
forms.CharField(widget=forms.PasswordInput(attrs={'
placeholder': 'Password',

'class': 'form-control item',

}))

class RegisterForm(UserCreationForm):
    username = forms.CharField(max_length=100,
widget=forms.TextInput(attrs={'placeholder':
'Username',

'class': 'form-control item',

}))
    first_name = forms.CharField(max_length=30,
required=False, help_text='Optional.',
```

```
widget=forms.TextInput(attrs={'placeholder': 'First
Name',
'class': 'form-control item',
}))
    last_name = forms.CharField(max_length=30,
required=False, help_text='Optional.',
widget=forms.TextInput(attrs={'placeholder': 'Last
Name',
'class': 'form-control item',
}))
    email = forms.EmailField(max_length=254,
help_text='Required. Inform a valid email
address.',
widget=forms.EmailInput(attrs={'placeholder':
'Email',
'class': 'form-control item',
}))
    password1 =
forms.CharField(widget=forms.PasswordInput(attrs={'
placeholder': 'Password',
'class': 'form-control item',
}))
    password2 =
forms.CharField(widget=forms.PasswordInput(attrs={'
placeholder': 'Confirm Password',
'class': 'form-control item',
```

```

)))

class Meta:
    model = User
    fields = ('username', 'first_name',
'last_name', 'email', 'password1', 'password2',)

```

5.2.3.4 Account templates login.html

```

{% extends 'base.html' %}

{% load static %}

{% block head %}
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content="width=device-
width, initial-scale=1">

{% endblock head %}

{% block styles %}
    {% include "snippets/form_styles.html" %}
    <link rel="stylesheet" type="text/css" href="{%
static "account/css/styles.css" %}">

{% endblock styles %}

{% block content %}
    <div class="registration-form">
        <form method="post">

```

```

        {% csrf_token %}
        <h1 style='text-align: center;'> Login
</h1>
        <br>
        <div class="form-icon"
style='background-color: #32CD32;'>
            <span><i class="icon icon-
user"></i></span>
        </div>
        {% for field in form %}
            <div class="form-group">
                {{ field }}
            </div>
        {% endfor %}
        <div class="form-group">
            <button type="submit" class="btn
btn-block register" style="background-
color:#32CD32;">Login</button>
        </div>
    </form>
</div>
{% endblock content %}

```

5.2.3.5 Account templates register.html

```

{% extends 'base.html' %}

{% load static %}

{% block head %}
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">

```

```

    <meta name="viewport" content="width=device-
width, initial-scale=1">

{% endblock head %}

{% block styles %}
    {% include "snippets/form_styles.html" %}
    <link rel="stylesheet" type="text/css" href="{%
static "account/css/styles.css" %}">
{% endblock styles %}

{% block content %}
    <div class="registration-form">
        <form method="post">
            {% csrf_token %}
            <h1 style='text-align: center;'>
Registrasi </h1>
            <br>
            <div class="form-icon"
style='background-color: #32CD32;'>
                <span><i class="icon icon-
user"></i></span>
            </div>
            {% for field in form %}
                <div class="form-group">
                    {{ field }}
                </div>
            {% endfor %}
            <div class="form-group">
                <button type="submit" class="btn
btn-block register" style="background-
color:#32CD32;">Registrasi</button>
            </div>
        </form>

```

```

</div>
{% endblock content %}

```

5.2.3.6 Account static css styles.css

```

html, body{ background-color: #98FB98;}

.registration-form{
  padding: 50px 0;
}

.registration-form form{
  background-color: #fff;
  max-width: 600px;
  margin: auto;
  padding: 50px 70px;
  border-top-left-radius: 30px;
  border-top-right-radius: 30px;
  box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.075);
}

.registration-form .form-icon{
  text-align: center;
  background-color: #5891ff;
  border-radius: 50%;
  font-size: 40px;
  color: white;
  width: 100px;
  height: 100px;
  margin: auto;
  margin-bottom: 50px;
  line-height: 100px;
}

```

```

.registration-form .item{
  border-radius: 20px;
  margin-bottom: 25px;
  padding: 10px 20px;
}

.registration-form .register{
  border-radius: 30px;
  padding: 10px 20px;
  font-size: 18px;
  font-weight: bold;
  background-color: #5791ff;
  border: none;
  color: white;
  margin-top: 20px;
}

```

5.2.4. App Kitab

App Kitab merupakan app yang berfungsi sebagai fitur upload kitab referensi pada aplikasi website.

5.2.4.1 Kitab urls.py

```

from django.urls import path

from . import views

app_name = 'kitab'
urlpatterns = [
    path('', views.upload)
]

```

5.2.4.2 Kitab views.py

```

from django.shortcuts import render
from django.http import HttpResponseRedirect

from .forms import KitabForm
from .models import UploadKitab

def upload(request):
    form_field = KitabForm()

    context = {
        'title' : "Upload Kitab | Kitab Ulama",
        'kitab_form' : form_field,
    }

    if(request.method=="POST"):
        print(request.POST)
        UploadKitab.objects.create(
            username = request.user,
            kategori = request.POST['kategori'],
            nama_kitab =
request.POST['nama_kitab'],
            nama_pengarang =
request.POST['nama_pengarang'],
            file_kitab =
request.POST['file_kitab'],
        )
        return HttpResponseRedirect("/")

    return render(request, 'kitab/index.html',
context)

```

5.2.4.3 Kitab forms.py

```

from django import forms

```

```

class KitabForm(forms.Form):
    kategori = forms.CharField(max_length=100,
widget=forms.TextInput(attrs={

'class': 'form-control item',

}))
    nama_kitab = forms.CharField(max_length=100,
widget=forms.TextInput(attrs={

'class': 'form-control item',

}))
    nama_pengarang =
forms.CharField(max_length=100, required=False,
widget=forms.TextInput(attrs={

'class': 'form-control item',

}))
    file_kitab =
forms.FileField(widget=forms.ClearableFileInput())

```

5.2.4.4 Kitab models.py

```

from django.db import models

# Create your models here.
class UploadKitab(models.Model):
    username = models.CharField(max_length=150,
default='Admin')
    kategori = models.CharField(max_length=100)
    nama_kitab = models.CharField(max_length=100)

```

```

nama_pengarang =
models.CharField(max_length=1000)
file_kitab = models.FileField()

published =
models.DateTimeField(auto_now_add=True)
konfirmasi = models.BooleanField(default=False)

def __str__(self):
    return "{}. {}".format(self.id,
self.nama_kitab)

```

5.2.4.5 Kitab admin.py

```

from django.contrib import admin

# Register your models here.
from .models import UploadKitab

admin.site.register(UploadKitab)

```

5.2.4.6 Kitab templates index.html

```

{% extends 'base.html' %}

{% load static %}

{% block head %}
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content="width=device-
width, initial-scale=1">
{% endblock head %}

```

```

{% block styles %}
    {% include "snippets/form_styles.html" %}
    <link rel="stylesheet" type="text/css" href="{%
static "kitab/css/styles.css" %}">
{% endblock styles %}

{% block content %}
    <div class="registration-form">
        <form method="post">
            {% csrf_token %}
            <h1 style='text-align: center;'> Upload
File Kitab </h1>
            <br>
            <div class="form-icon"
style='background-color: #32CD32;'>
                <span><i class="icon icon-book-
open"></i></span>
            </div>
            {% for field in kitab_form %}
                <div class="form-group">
                    {{ field.label_tag}}
                    {{ field }}
                </div>
            {% endfor %}
            <div class="form-group">
                <button type="submit" class="btn
btn-block register" style="background-
color:#32CD32;">Upload Kitab</button>
            </div>
        </form>
    </div>
{% endblock content %}

```

5.2.4.7 Kitab static css styles.css

```
html, body{ background-color: #98FB98;}

.registration-form{
    padding: 50px 0;
}

.registration-form form{
    background-color: #fff;
    max-width: 600px;
    margin: auto;
    padding: 50px 70px;
    border-top-left-radius: 30px;
    border-top-right-radius: 30px;
    box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.075);
}

.registration-form .form-icon{
    text-align: center;
    background-color: #5891ff;
    border-radius: 50%;
    font-size: 40px;
    color: white;
    width: 100px;
    height: 100px;
    margin: auto;
    margin-bottom: 50px;
    line-height: 100px;
}

.registration-form .item{
    border-radius: 20px;
    margin-bottom: 25px;
    padding: 10px 20px;
```

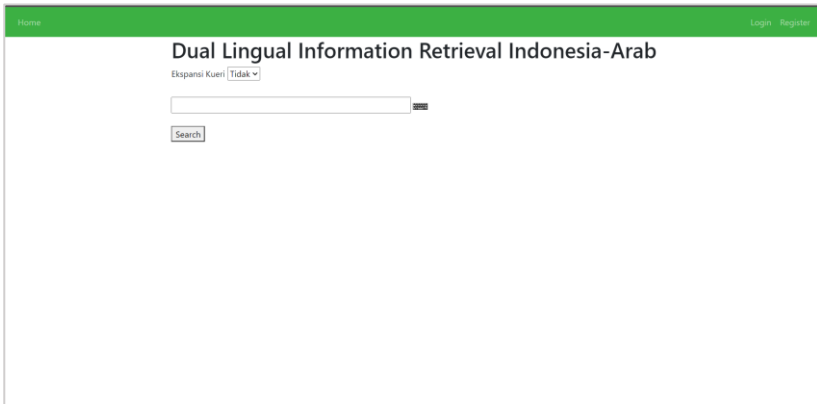
```
}  
  
.registration-form .register{  
  border-radius: 30px;  
  padding: 10px 20px;  
  font-size: 18px;  
  font-weight: bold;  
  background-color: #5791ff;  
  border: none;  
  color: white;  
  margin-top: 20px;  
}
```

5.3. Implementasi Antarmuka Pengguna

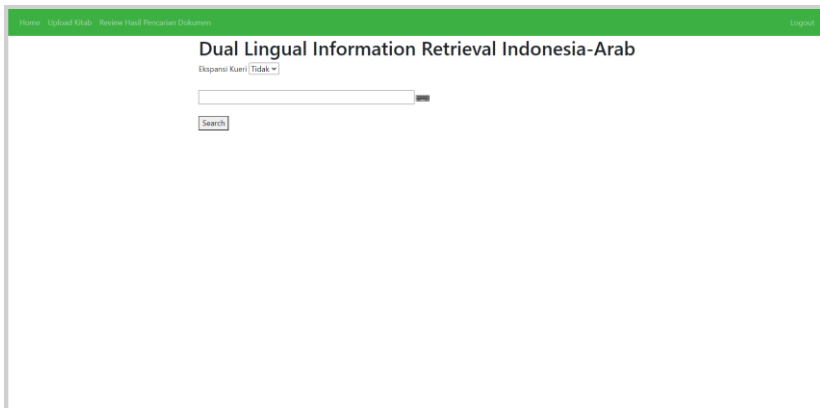
Pada bagian ini penulis akan menampilkan antarmuka halaman yang ada pada aplikasi CLIR berbasis website bahasa Arabic menggunakan input bahasa Indonesia. Berikut tampilan antarmuka aplikasi CLIR berbasis website bahasa Arabic menggunakan input kueri bahasa Indonesia:

5.3.1. Halaman Utama

Pada halaman utama memunculkan informasi umum aplikasi CLIR pada Gambar 5.1 sebelum login dan Gambar 5.2 setelah login. Informasi umum yang ditampilkan berupa fitur-fitur aplikasi seperti login, registrasi, input kueri, upload kitab, dan review halaman pencarian dokumen.



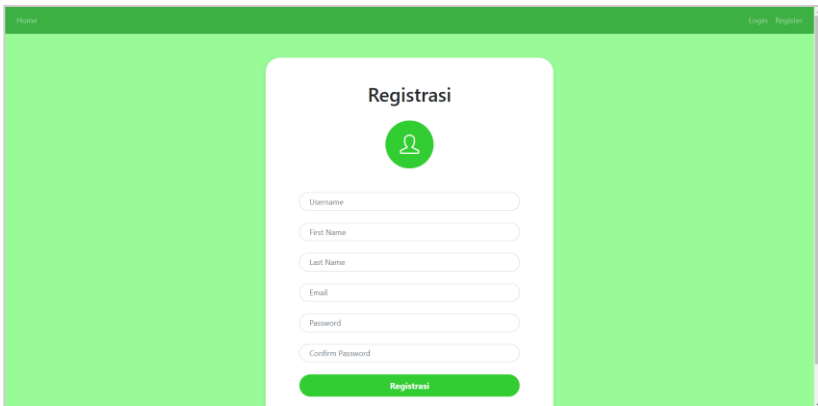
Gambar 5.1 : Tampilan Halaman Utama Sebelum Login



Gambar 5.2 : Tampilan Halaman Utama Setelah Login

5.2.2. Halaman Registrasi User

Pada halaman registrasi user gambar 5.3. menampilkan form pengisian registrasi akun pada aplikasi website berupa username, first name, last name, email, password, dan konfirmasi password.

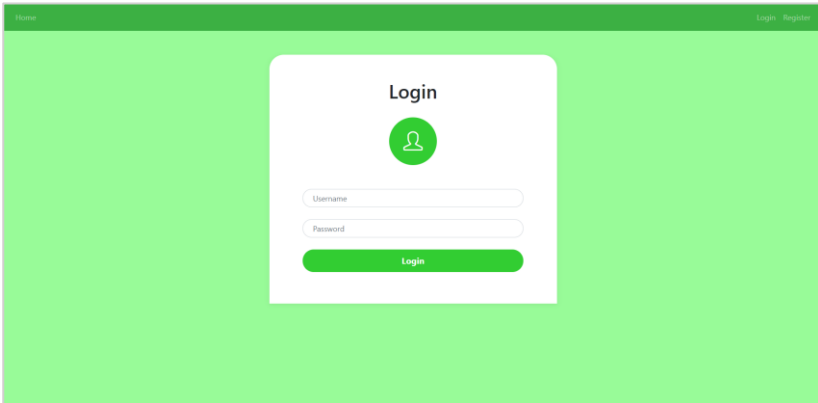


The image shows a web browser window with a green header bar. On the left side of the header is the word "Home" and on the right side are the links "Login" and "Register". The main content area has a light green background. In the center, there is a white rounded rectangle containing the registration form. At the top of this form is the title "Registrasi" and a green circular icon with a white person silhouette. Below the icon are six input fields: "Username", "First Name", "Last Name", "Email", "Password", and "Confirm Password". At the bottom of the form is a green button with the text "Registrasi" in white.

Gambar 5.3 : Tampilan Halaman Registrasi User

5.2.3. Halaman Login User

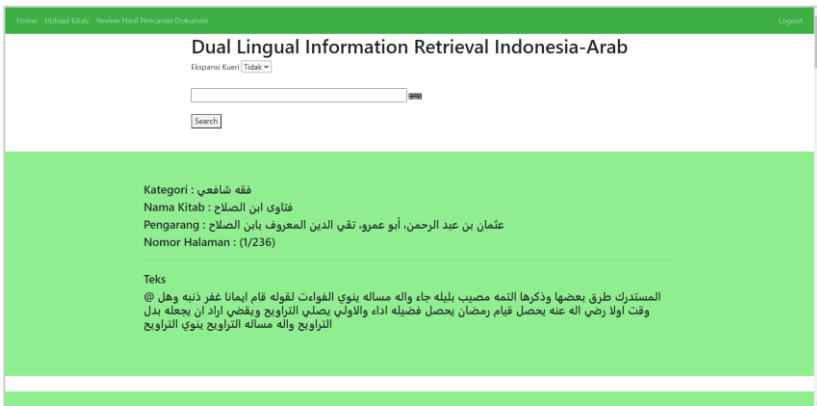
Pada halaman login user gambar 5.4. menampilkan form pengisian login akun pada aplikasi website berupa username dan password bagi user yang telah terdaftar dalam aplikasi website.



Gambar 5.4 : Tampilan Halaman Login Akun

5.2.4. Halaman Hasil Pencarian Dokumen

Pada halaman ini menampilkan hasil pencarian dokumen setelah memasukkan kueri pada fitur *search bar* tampilan halaman utama. Dokumen yang ditampilkan sebanyak 10 dokumen yang paling mendekati dengan masukan kueri oleh user. Tampilan halaman dapat ditunjukkan pada gambar 5.5. dan 5.6.



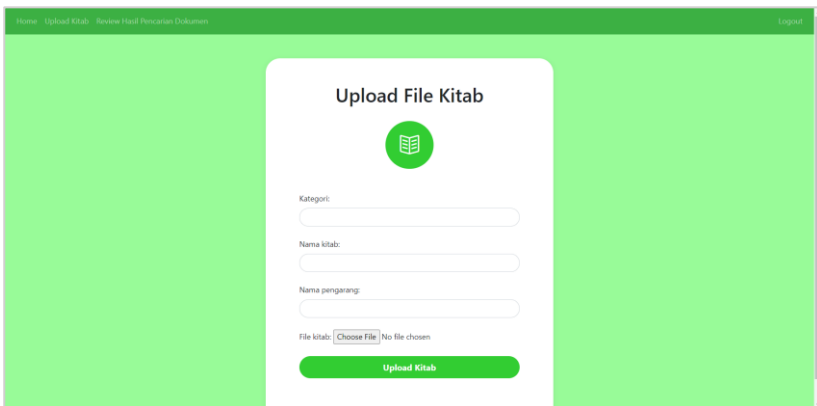
Gambar 5.5 : Tampilan Halaman Hasil Pencarian Dokumen Bagian 1



Gambar 5.6 : Tampilan Halaman Hasil Pencarian Dokumen Bagian 2

5.2.5. Halaman Upload Kitab

Pada halaman ini menampilkan form input upload kitab dokumen referensi oleh user dengan mengisi kategori, nama kitab, nama pengarang, serta file kitab dokumen referensi yang mau diupload ke dalam aplikasi website. Tampilan halaman dapat ditunjukkan pada gambar 5.7.



The screenshot shows a web application interface with a green header and a white central form titled "Upload File Kitab". The form contains a green circular icon with a book symbol, followed by input fields for "Kategori", "Nama kitab:", and "Nama pengarang:". Below these is a file upload section with a "Choose File" button and a "No file chosen" status. At the bottom of the form is a green "Upload Kitab" button. The navigation bar at the top includes "Home", "Upload Kitab", "Review Hasil Pencarian Dokumen", and "Logout".

Gambar 5.7 : Tampilan Halaman Upload Kitab

5.2.6. Halaman Review Hasil Pencarian Dokumen

Pada halaman ini menampilkan form input review terhadap hasil pencarian dokumen dengan memasukkan kueri yang diberikan sebelumnya, isi review terhadap hasil pencarian dokumen, serta jumlah dokumen yang relevan dengan masukan kueri. Tampilan halaman dapat ditunjukkan pada gambar 5.8.

Home | Daftar Ekst | Review Hasil Pencarian Dokumen | Logout

Review Hasil Pencarian Dokumen

Query:

Review:

Berapa jumlah dokumen yang relevan dari hasil pencarian dokumen?
0

[Kirim Review](#)

Gambar 5.8 : Tampilan Halaman Review Hasil Pencarian Dokumen

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap aplikasi *Cross Lingual Information Retrieval* berbasis website untuk Bahasa Indonesia dan Bahasa Arab. Pengujian dilakukan pada dua tahapan, yaitu uji coba relevansi aplikasi dan uji coba fungsionalitas aplikasi. Uji coba relevansi aplikasi dilakukan dengan mengukur tingkat *recall* dan *precision* terhadap hasil pencarian dokumen yang relevan pada aplikasi. Uji coba fungsionalitas aplikasi dilakukan dengan memastikan kualitas aplikasi yang dibangun dan kesesuaian hasil eksekusi aplikasi dengan analisis dan perancangan aplikasi pada website Kitab Ulama.

6.1. Skenario Pengujian

6.1.1. Skenario Uji Coba Relevansi Aplikasi

Skenario pengukuran tingkat *recall* dan *precision* aplikasi adalah sebagai berikut :

1. Membuka <https://kitab-ulama.com> pada browser.
2. Memberikan input kueri pada *search bar* yang tampil pada halaman awal website, di mana kueri sudah tersedia pada kueri *ground truth* yang telah diberikan.
3. Menampilkan hasil pencarian dokumen kitab referensi yang berelasi dengan input kueri yang diberikan.

4. Menghitung jumlah dokumen yang relevan dari 10 dokumen yang diretrieved pada setiap kueri *ground truth* dalam aplikasi.
5. Menghitung jumlah dokumen yang tidak relevan dari 10 dokumen yang diretrieved pada setiap kueri *ground truth* dalam aplikasi.
6. Menghitung jumlah dokumen relevan yang berasal dari dokumen-dokumen relevan pada kueri *ground truth*, yang tidak diretrieved dari 10 dokumen yang ditampilkan pada aplikasi.

6.1.2. Skenario Uji Coba Fungsionalitas Aplikasi

Skenario uji coba fungsionalitas aplikasi adalah sebagai berikut.

6.1.2.1. Membuka Halaman Utama Kitab Ulama

1. Membuka <https://kitab-ulama.com> pada browser untuk menampilkan halaman utama website.
2. Memastikan halaman utama sudah tampil pada browser.

6.1.2.2. Melakukan Registrasi Kitab Ulama

1. Membuka <https://kitab-ulama.com> pada browser untuk menampilkan halaman utama website.
2. Membuka halaman registrasi pada fitur register.
3. Mengisi form input registrasi pada halaman registrasi.

4. Memastikan data yang telah diisi pada form masuk ke dalam database halaman admin (admin).

6.1.2.3. Melakukan Login Kitab Ulama

1. Membuka <https://kitab-ulama.com> pada browser untuk menampilkan halaman utama website.
2. Membuka halaman login pada fitur login.
3. Mengisi form input login pada halaman login.

6.1.2.4. Menggunakan Search Engine pada Kitab Ulama

1. Membuka <https://kitab-ulama.com> pada browser untuk menampilkan halaman utama website.
2. Memberikan input kueri pada fitur *search bar* yang tampil pada halaman awal website, di mana kueri sudah tersedia pada kueri uji *ground truth* yang telah diberikan.
3. Menampilkan hasil pencarian dokumen kitab referensi yang berelasi dengan input kueri yang diberikan.

6.1.2.5. Melakukan Review Hasil Pencarian Dokumen Kitab Ulama

1. Membuka <https://kitab-ulama.com> pada browser untuk menampilkan halaman utama website.
2. Membuka halaman review hasil pencarian dokumen pada fitur review.
3. Mengisi form input review pada halaman review hasil pencarian dokumen.
4. Memastikan data yang telah diisi pada form review masuk ke dalam database halaman admin. (admin)

6.1.2.6. Melakukan Upload Kitab Ulama

1. Membuka <https://kitab-ulama.com> pada browser untuk menampilkan halaman utama website.
2. Membuka halaman upload kitab pada fitur upload kitab.
3. Mengisi form input serta melakukan upload dokumen kitab pada halaman upload kitab.
4. Memastikan data yang telah diisi pada form upload kitab masuk ke dalam database halaman admin. (admin)

6.2. Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi dilakukan pada uji coba relevansi aplikasi maupun uji coba fungsionalitas aplikasi.

6.2.1. Evaluasi Uji Coba Relevansi Sistem

Hasil evaluasi uji coba relevansi sistem dapat ditunjukkan pada tabel 6.1. di bawah ini.

Tabel 6.1. Hasil Evaluasi Uji Coba Relevansi Aplikasi

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
1	Hukum menginfakkan sebagian uang masjid kepada fakir miskin	6	2	4	75%	60%
2	Cara orang bisu membaca Al-Qur'an	0	2	10	0%	0%
3	Merenggangkan shaf shalat jamaah karena udzur	0	2	10	0%	0%
4	Hukum ulat dijadikan minyak goreng	2	2	8	50%	20%
5	Hukum pengajian tadarus Al-Qur'an dengan pengeras suara di bulan ramadhan	3	1	7	75%	30%

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
6	Hukum makan hingga kekenyangan	5	1	5	83%	50%
7	Etika mengucapkan salam dan mengetuk pintu	1	1	9	50%	10%
8	Hukum menjual pulsa pada orang hendak pacaran	2	2	8	50%	20%
9	Hukum menabung di bank	5	4	5	56%	50%
10	Hukum shalat tanpa memakai penutup kepala	0	1	10	0%	0%
11	Hukum terkait wanita hamil di bulan ramadhan	2	2	8	50%	20%
12	Daging qurban tidak dibagikan kepada fakir miskin	6	1	4	86%	60%
13	Hukum mengawetkan jenazah	1	1	9	50%	10%

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
14	Apakah makan sahur mencukupi sebagai niat puasa	4	1	6	80%	40%
15	Hukum jual beli uang yang rusak	1	1	9	50%	10%
16	Hukum menjual makanan yang terkena najis	2	1	8	67%	20%
17	Hukum menjual anggota tubuh	3	1	7	75%	30%
18	Hukum mencukur alis	2	1	8	67%	20%
19	Hukum sabun sebagai pengganti debu	2	1	8	67%	20%
20	Hukum menikahi wanita hamil	5	1	5	83%	50%
21	Hukum memberi karena malu	3	1	7	75%	30%

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
22	Hukum mendatangi gereja saat natal	0	2	10	0%	0%
23	Hukum berprofesi sebagai pelawak	0	1	10	0%	0%
24	Mengambil jamur di tanah orang lain	2	1	8	67%	20%
25	Hukum jual beli jin	0	2	10	0%	0%
26	Hukum mengumandangkan adzan dalam keadaan duduk	1	2	9	33%	10%
27	Aurat terbuka saat sholat secara tiba-tiba	1	2	9	33%	10%
28	Hukum mengedarkan kotak amal ketika khutbah jum'at	4	1	6	80%	40%
29	Kriteria halal haram hewan air	1	1	9	50%	10%

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
30	Makan dan minum ketika khutbah Jum'at	0	1	10	0%	0%
31	Hukum menahan kentut saat shalat	0	1	10	0%	0%
32	Shalat orang yang berada di penjara	0	1	10	0%	0%
33	Hukum menggabungkan shalat sunnah qobliyah dan ba'diyah	0	1	10	0%	0%
34	Hukum khitan dengan laser	0	2	10	0%	0%
35	Mengumandangkan adzan ketika bayi lahir	0	1	10	0%	0%
36	Mendapat undangan yang waktu acaranya bersamaan	3	2	7	60%	30%

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
37	Syarat perlombaan yang diperbolehkan	2	1	8	67%	20%
38	Hukum profesi tukang pijat	2	3	8	40%	20%
39	Hukum menerima kembali beras yang telah dizakatkan	3	1	7	75%	30%
40	Hukum menunaikan zakat fitrah sebelum bulan ramadhan	2	3	8	40%	20%
41	Hukum dana kotak amal masjid untuk dana sosial	3	3	7	50%	30%
42	Sholat Tahajud tanpa tidur dahulu	7	0	3	100%	70%
43	Adzan sebelum masuk waktu sholat	3	2	7	60%	30%
44	Makmum tidak ikut imam sujud sahwi	5	1	5	83%	50%

No	Query	Relevan - Retrieved	Relevan - Non Retrieved	Non Relevan - Retrieved	Recall	Precision
45	Menelan air mani	1	1	9	50%	10%
46	Tata cara sujud yang benar dalam sholat	6	1	4	86%	60%
47	Meninggalkan sholat fardhu	1	1	9	50%	10%
48	Membuka sangkar burung orang lain	0	1	10	0%	0%
49	Cara mensucikan mulut jika terkena najis	3	3	7	50%	30%
50	Hukum bermain catur	5	0	5	100%	50%

Berdasarkan hasil tabel evaluasi uji coba relevansi sistem, diperoleh rata-rata dari hasil *recall* secara keseluruhan adalah 47% dan *precision* secara keseluruhan adalah 22%.

6.2.2. Evaluasi Uji Coba Fungsionalitas Sistem

Hasil evaluasi uji coba fungsionalitas sistem dapat ditunjukkan pada tabel 6.2. di bawah ini.

Tabel 6.2. Hasil Evaluasi Pengujian Fungsionalitas Aplikasi

No	Kriteria Pengujian	Uji Coba	Hasil Pengujian
1	Dapat membuka halaman utama https://kitab-ulama.com	User membuka https://kitab-ulama.com pada browser untuk menampilkan halaman utama website.	Berhasil
2	Dapat melakukan registrasi akun	User mengisi form input registrasi pada halaman registrasi.	Berhasil
3	Dapat melakukan login akun	User mengisi form input login pada halaman login.	Berhasil
4	Dapat menampilkan hasil search engine pada input kueri yang diberikan tanpa menggunakan ekspansi kueri	User memberikan input kueri pada fitur <i>search bar</i> yang tampil pada halaman awal website, di mana kueri sudah tersedia pada kueri uji <i>ground truth</i> yang telah diberikan.	Berhasil

No	Kriteria Pengujian	Uji Coba	Hasil Pengujian
5	Dapat menampilkan hasil search engine pada input kueri yang diberikan tanpa menggunakan ekspansi kueri	User memberikan input kueri pada fitur <i>search bar</i> yang tampil pada halaman awal website, di mana kueri sudah tersedia pada kueri uji <i>ground truth</i> yang telah diberikan.	Berhasil
6	Dapat memberikan review hasil pencarian dokumen	User mengisi form input review pada halaman review hasil pencarian dokumen.	Berhasil
7	Dapat mengupload kitab referensi	User mengisi form input serta melakukan upload dokumen kitab pada halaman upload kitab.	Berhasil

[Halaman ini sengaja dikosongkan]

BAB VII KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pembuatan aplikasi *cross lingual information retrieval* bahasa Indonesia – Arab berbasis website adalah sebagai berikut:

- a. Diperoleh hasil metriks *recall* sebesar 47% dan *precision* sebesar 22%. Aplikasi dapat menampilkan dokumen kitab referensi yang relevan pada kueri-kueri tertentu. Beberapa kueri tidak dapat menampilkan hasil dokumen kitab referensi relevan karena representasi vector kata dari input kueri tidak dapat dikenal dari koleksi model representasi vektor kata yang dimiliki.
- b. Aplikasi yang dibangun dapat digunakan sesuai kegunaan fungsionalitas dari aplikasi.
- c. Adanya aplikasi website *cross lingual information retrieval* bahasa Indonesia – Arab dapat secara otomatis mengakomodasi kebutuhan pengguna dalam pencarian dokumen kitab referensi.

7.2. Saran

- a. Melakukan tuning parameter jumlah kata paling mirip dengan kueri yang digunakan pada ekspansi kueri agar hasil pencarian dokumen lebih optimal.
- b. Memperkaya jumlah dokumen kitab referensi agar hasil representasi vektor kata yang diperoleh lebih lengkap yang dapat lebih mengenali setiap kata dari input kueri.

DAFTAR PUSTAKA

- [1] Wikipedia (2020). Maktabah Syamilah – Wikipedia bahasa Indonesia, ensiklopedia bebas. [online] Available at: https://id.wikipedia.org/wiki/Maktabah_Syamilah [Accessed 9 September 2020].
- [2] Wikipedia (2020). Aplikasi web – Wikipedia bahasa Indonesia, ensiklopedia bebas. [online] Available at: https://id.wikipedia.org/wiki/Aplikasi_web [Accessed 9 September 2020].
- [3] Hackernoon (2020). Data Preprocessing: 6 Necessary Steps for Data Scientist | Hacker Noon. [online] Available at: <https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa> [Accessed 9 September 2020].
- [4] Machine Learning Mastery (2020). How to Encode Text Data for Machine Learning with scikit-learn. [online] Available at: <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/> [Accessed 9 September 2020].
- [5] Structilmy (2020). Word Embedding Bahasa Indonesia menggunakan FastText (dengan Gensim) – Structilmy. [online] Available at: <https://structilmy.com/2019/04/word-embedding-bahasa-indonesia-menggunakan-fasttext-part-1/> [Accessed 9 September 2020].
- [6] Wikipedia (2020). Query Expansion – Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Query_expansion [Accessed 9 September 2020].
- [7] Wikipedia (2020). Anaconda (Python distribution) – Wikipedia. [online] Available at: [https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)) [Accessed 9 September 2020].

- [8] Python.org (2020). What is Python? Executive Summary | Python.org. [online] Available at: <https://www.python.org/doc/essays/blurb/> [Accessed 9 September 2020].
- [9] Djangoproject.com (2020). The Web framework for perfectionists with deadlines | Django. [online] Available at: <https://www.djangoproject.com/> [Accessed 9 September 2020].
- [10] Wikipedia (2020). HTML - Wikipedia bahasa Indonesia, ensiklopedia bebas [online] Available at: <https://id.wikipedia.org/wiki/HTML> [Accessed 9 September 2020].
- [11] Wikipedia (2020). Cascading Style Sheets - Wikipedia bahasa Indonesia, ensiklopedia bebas [online] Available at: https://id.wikipedia.org/wiki/Cascading_Style_Sheets [Accessed 9 September 2020].
- [12] Wikipedia (2020). MySQL – Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/MySQL> [Accessed 9 September 2020].
- [13] Niagahoster (2020). Pengertian Hosting adalah Rumah Bagi Website (Terlengkap!). [online] Available at: <https://www.niagahoster.co.id/blog/hosting-adalah/> [Accessed 9 September 2020].

DOKUMENTASI

- a. 8 Juni 2020 – 17 Juni 2020 (Metodologi)
- Revisi draft ppt penyusunan metodologi karena penyusunan metodologi sistem tidak hanya terkait dengan pembangunan sistem, tetapi juga perlu ditambahkan tahap pembuatan aplikasi, evaluasi dan *maintenance* pada sistem.
 - Revisi penulisan sumber dokumen referensi bahasa Arab.
 - Penambahan kueri dengan menggunakan bahasa Arab pada tahap *Maintenance* sistem.
 - Penambahan *Mockup* halaman admin pada draft ppt penyusunan metodologi sistem.
 - Penambahan fitur login agar sistem yang dibangun lebih aman dan admin dapat mengetahui pengguna yang memberikan review terhadap input output kueri.
 - Penambahan kueri uji untuk persiapan tahap evaluasi.
 - Melakukan implementasi sistem.
 - Evaluasi penentuan parameter banyak kata yang diperoleh dari hasil ekspansi kueri.
 - Revisi keluaran teks dari dokumen referensi bahasa Arab yang ditampilkan karena keluaran teks berasal dari teks asli dokumen referensi bahasa Arab, bukan berasal dari hasil preprocessing data.

- Revisi metode *word embedding* karena dokumen referensi bahasa Arab yang ditampilkan masih belum relevan.
- b. 17 Juni 2020 – 24 Juni 2020 (Metodologi)
- Pemaparan draft PPT penyusunan metodologi tahapan implementasi, evaluasi, dan maintenance aplikasi DLIR.
 - Pemaparan progres implementasi aplikasi CLIR.
 - Melanjutkan progres implementasi aplikasi.
- c. 24 Juni 2020 – 1 Juli 2020 (Implementasi)
- Penyampaian progres implementasi dan diskusi web hosting aplikasi CLIR.
 - Melanjutkan progres implementasi aplikasi ke dalam bentuk web serta menyiapkan kueri uji ground truth untuk tahap evaluasi.
- d. 1 Juli 2020 – 8 Juli 2020 (Implementasi)
- Membersihkan file dokumen *corrupt* pada dataset Arabic Maktabah Syamilah.
 - Mempelajari penggunaan framework Django pada aplikasi website.
 - Menyiapkan kueri uji ground truth untuk tahap evaluasi.
 - Melanjutkan progres implementasi aplikasi ke dalam bentuk website.

- e. 8 Juli 2020 – 15 Juli 2020 (Evaluasi)
- Mempelajari penggunaan framework Django pada aplikasi website.
 - Memperlengkap kueri uji ground truth untuk tahap evaluasi.
 - Melanjutkan progres implementasi aplikasi ke dalam bentuk website.
- f. 15 Juli 2020 – 22 Juli 2020 (Evaluasi)
- Pemaparan hasil implementasi dataset Arabic.
 - Pemaparan hasil pencarian dokumen keseluruhan dataset Arabic.
 - Pemaparan hasil uji coba fungsional aplikasi website pada localhost. Aplikasi website yang dibuat masih belum diberikan front-end secara keseluruhan untuk memberikan tampilan yang lebih menarik. Terdapat perubahan rencana penggunaan framework Laravel berbasis bahasa pemrograman PHP menjadi menggunakan framework Django berbasis bahasa pemrograman Python karena kesulitan menjalankan NLP pada Laravel. Aplikasi website belum menampilkan hasil pencarian tanya jawab Bahasa Indonesia.
 - Pemaparan progres pembuatan kueri uji ground truth. Setiap kueri uji lebih baik menyediakan hasil dokumen lebih dari satu agar *precision recall* yang diperoleh lebih baik.
 - Penyampaian penggunaan web hosting pada aplikasi website. Pada web hosting yang digunakan

awal, tidak tersedia python pada web hosting tersebut sehingga kesulitan untuk melakukan set up Python setelah berdiskusi bersama dan menghubungi contact person web hosting tersebut. Oleh karena itu, aplikasi website pindah menggunakan paket web hosting yang baru.

- g. 22 Juli 2020 – 29 Juli 2020 (Maintenance)
- Menyelesaikan front-end pada aplikasi website.
 - Menyelesaikan hasil pencarian referensi tanya jawab dataset bahasa Indonesia.
 - Memasukkan aplikasi website ke dalam web hosting.
 - Memperlengkap list dokumen pada setiap kueri uji yang telah dibuat. Kebanyakan dokumen yang tersedia pada kueri uji hanya satu kitab referensi karena tidak semua kitab Maktabah Syamilah digunakan oleh piss-ktb dan kebanyakan berfokus pada mazhab tertentu misalkan Mazhab Syafi'i.
 - Memasukkan data pada database MySQL web hosting.
- h. 29 Juli 2020 – 8 Agustus 2020 (Maintenance)
- Mengeksekusi implementasi pencarian dokumen bahasa Arab dengan menggunakan Anaconda. Implementasi masih terdapat error pada alokasi array pada memori karena terdapat kemungkinan memory penuh pada harddisk laptop.

- Memasukkan data ke dalam MySQL Databases pada web hosting. Untuk dimasukkan ke dalam web hosting melalui server hosting secara langsung, format file harus menggunakan format SQL, sedangkan format file yang dimiliki masih berupa format CSV dan belum bisa dimasukkan dalam database.
- Melakukan konversi data format csv ke dalam bentuk format sql agar bisa dijalankan melalui server hosting secara langsung. Proses konversi masih terdapat kendala karena pada saat proses, laptop yang digunakan untuk konversi format data sedang lag dan data yang digunakan besar sekali.
- Pemaparan terkait penggunaan web hosting pada penelitian. Hasil implementasi belum bisa ditampilkan pada web hosting yang telah dibeli akibat resource yang sangat tinggi pada model yang dimiliki untuk proses hasil pencarian dokumen bahasa Arab.
- Mencari solusi alternatif penerapan implementasi aplikasi website agar dapat berjalan dengan baik pada server web hosting.
- Pengerjaan evaluasi dari kueri *ground truth* dan fungsionalitas aplikasi.

BIODATA PENULIS



Muhammad Harits Fadlilah, lahir pada tanggal 19 Juni 1999 di Marabahan, Kalimantan Selatan. Penulis merupakan mahasiswa yang sedang menempuh studi di Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS). Penulis aktif dalam berorganisasi pada Unit Kegiatan Mahasiswa (UKM) Catur Institut Teknologi Sepuluh Nopember 2018/2020 dalam Departemen Pengembangan Sumber Daya Anggota.