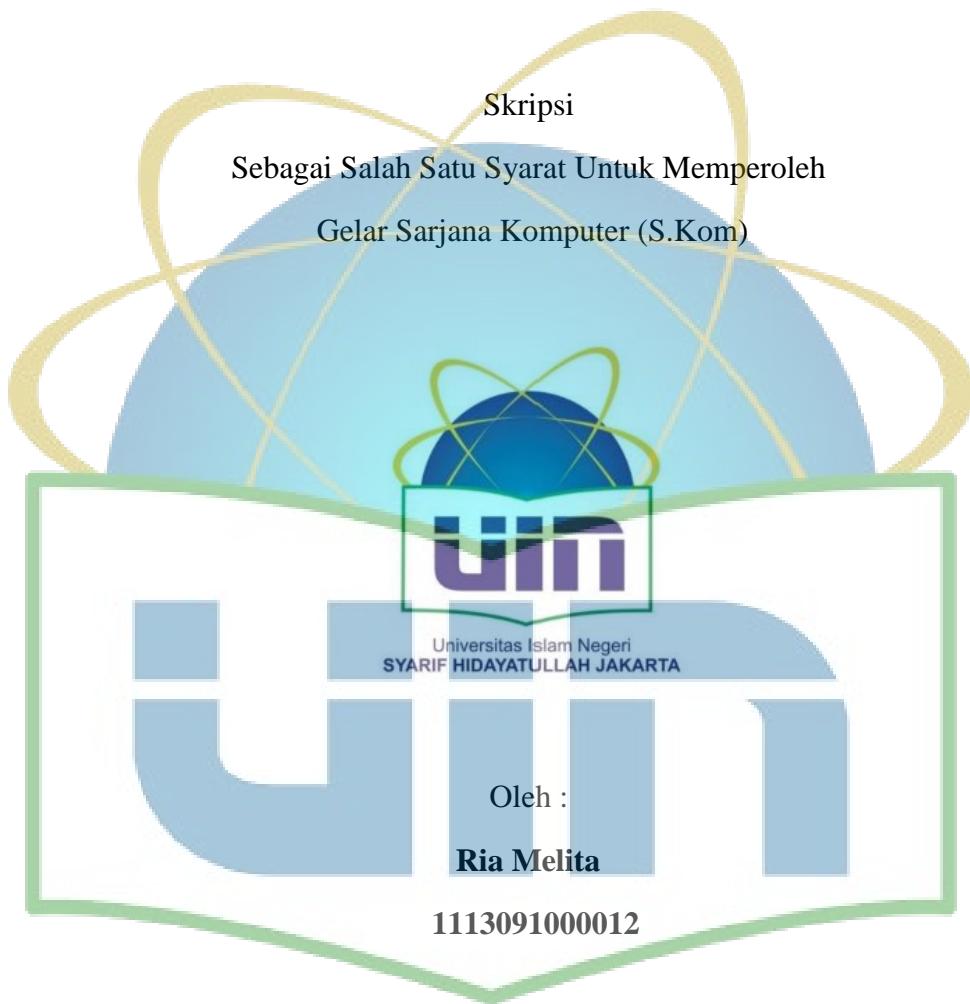


**PENERAPAN METODE *TERM FREQUENCY INVERSE DOCUMENT FREQUENCY* (TF-IDF) DAN *COSINE SIMILARITY* PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI *SYARAH HADITS* BERBASIS WEB  
(STUDI KASUS: *HADITS SHAHIH BUKHARI-MUSLIM*)**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
1439 H/2018 M**

**PENERAPAN METODE *TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF)* DAN *COSINE SIMILARITY* PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI *SYARAH HADITS* BERBASIS WEB  
(STUDI KASUS: *HADITS SHAHIH BUKHARI-MUSLIM*)**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2018 M/ 1439 H**

**LEMBAR PERSETUJUAN SKRIPSI**  
**PENERAPAN METODE TERM FREQUENCY INVERSE DOCUMENT**  
**FREQUENCY (TF-IDF) DAN COSINE SIMILARITY PADA SISTEM**  
**TEMU KEMBALI INFORMASI UNTUK MENGETAHUI**  
**SYARAH HADITS BERBASIS WEB**  
**(STUDI KASUS: HADITS SHAHIH BUKHARI-MUSLIM)**



Mengetahui,  
Ketua Program Studi Teknik Informatika

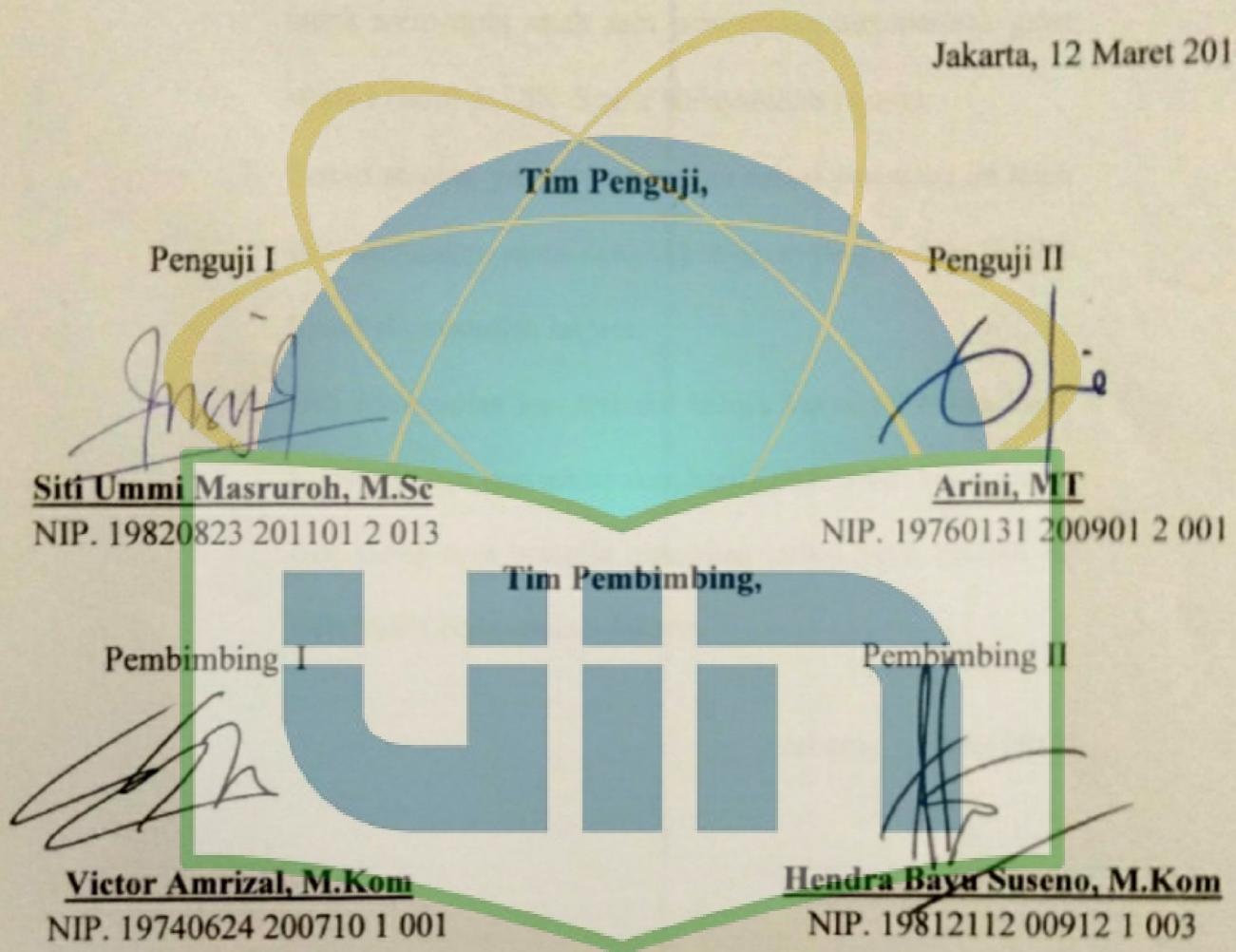
  
Arini, MT.

NIP. 19760131 200901 2 001

## LEMBAR PENGESAHAN

Skripsi yang berjudul “*Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)*” telah diujikan dan dinyatakan lulus dalam Sidang *Munaqasyah* Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta pada 12 Maret 2018, skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika.

Jakarta, 12 Maret 2018



Mengetahui,

Dekan Fakultas Sains dan Teknologi,  
  
**Dr. Agus Salim, S.Ag, M.Si**  
NIP. 19720816 199903 1 003

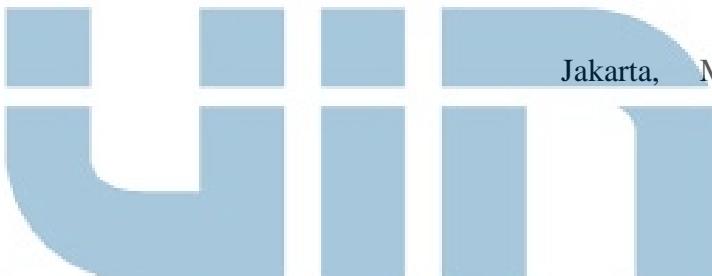
Ketua Program Studi Teknik  
Informatika,

**Arini, MT**  
NIP. 19760131 200901 2 001

## **PERNYATAAN ORISINALITAS**

Dengan Ini Saya Menyatakan Bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar strata 1 (satu) di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Jika dikemudian hari terbukti bahwa karya ini bukan hasil karya asli saya atau merupakan hasil jiplakan dari karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.



Jakarta, Maret 2018

Ria Melita

## **PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI**

### **HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademik UIN Syarif Hidayatullah Jakarta, saya yang bertanda tangan dibawah ini:

Nama : Ria Melita  
NIM : 1113091000012  
Program Studi : Teknik Informatika  
Fakultas : Sains dan Teknologi  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Islam Negeri Syarif Hidayatullah Jakarta **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah yang berjudul :

**Penerapan Metode *Term Frequency Inverse Document Frequency* (Tf-Idf) dan *Cosine Similarity* pada Sistem Temu Kembali Informasi untuk Mengetahui *Syarah Hadits* Berbasis Web (Studi Kasus: *Hadits Shahih Bukhari-Muslim*)**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Islam Negeri Syarif Hidayatullah Jakarta berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 17 Februari 2018

Ria Melita

(.....)

## KATA PENGANTAR

*Assalamu'alaikum Wr.Wb*

Puji syukur kehadirat Allah SWT yang telah memberikan nikmat iman dan islam serta kesehatan dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Sholawat serta salam tak lupa penulis curahkan kepada junjungan kita Rasulullah Sallahu'alaihi Wasallam beserta keluarga, para sahabat, dan seluruh umatnya.

Skripsi yang berjudul "**Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)**" ini disusun untuk memenuhi salah satu syarat untuk mendapatkan gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika Fakultas Sains dan Teknologi, Universitas Islam Negeri UIN Syarif Hidayatullah Jakarta.

Dalam penyusunan skripsi ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan banyak terima kasih kepada:

1. Bapak Dr. Agus Salim, M. Si. selaku Dekan Fakultas Sains dan Teknologi.
2. Ibu Arini, MT, selaku Ketua Prodi Teknik Informatika Fakultas Sains dan Teknologi, dan Bapak Feri Fahrianto,M.Sc. selaku sekretaris Program Studi Teknik Informatika.
3. Bapak Victor Amrizal, M.Kom, selaku dosen pembimbing I dan Bapak Hendra Bayu Suseno, M.Kom, selaku dosen pembimbing II yang telah banyak memberikan arahan dan bimbingan kepada penulis sehingga sangat membantu dalam menyelesaikan Skripsi ini.
4. Bapak dan Ibu dosen Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta yang telah memberikan ilmunya kepada penulis semasa pekuliahannya.

- 
5. Orang tua penulis yang telah memberikan motivasi dan do'a yang tiada henti-hentinya serta kasih sayang yang tak pernah habis untuk penulis.
  6. Kakak tercinta Yani Soraya dan Mas Adnan Dendy Mardika yang selalu memberikan dukungan dan arahan kepada penulis.
  7. Husnil Khuluqi yang telah memberikan motivasi, semangat, sekaligus tekanan kepada penulis untuk segera menyelesaikan skripsi ini.
  8. Teman seper-kostan penulis yang tergabung dalam "Nero Squad" yang selalu meluangkan waktu untuk berbagi canda dan tawa serta semangat selama penggeraan skripsi ini seperti Zhella Apriesta, Silvia Puji Lestari, Nurapzafidah, dan Ika Riyanti.
  9. Teman seperjuangan dari awal hingga akhir perkuliahan yang tergabung dalam "Muslimah Penuh Berkah Group" seperti Azizah Nasution, Ridhati Mardhiyah, Zhella Apriesta, dan Aslamia Nashiha yang selalu memberikan semangat serta dukungan dan kebahagiaan yang tak pernah habis.
  10. Rekan-rekan mahasiswa Program Studi Teknik Informatika angkatan 2013 khususnya TI-A, yang telah memberikan masukan, serta motivasi kepada penulis.
  11. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam penyelesaian penulisan Skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan. Oleh karena itu, penulis mengharapkan saran dan kritik untuk dapat memperbaiki dan menyempurnakan skripsi ini sehingga dapat bermanfaat bagi kita semua yang dapat disampaikan melalui *email* : ria.melita17@gmail.com.

Jakarta, Maret 2017

Penulis

## ABSTRAK

**Ria Melita**, Penerapan Metode *Term Frequency Inverse Document Frequency* dan *Cosine Similarity* Pada Sistem Temu Kembali Informasi Untuk Mengetahui *Syarah Hadits* Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim), dibimbing oleh **Victor Amrizal, M.Kom** dan **Hendra Bayu Suseno, M.Kom.**

*Hadits* merupakan sumber ajaran Islam disamping *Al-Qur'an*. Tanpa menggunakan *hadits*, syari'at Islam tidak dapat dimengerti secara utuh dan tidak dapat dilaksanakan. Namun, tidak dapat dipungkiri bahwa saat ini, tidak sedikit orang yang keliru dalam memahaminya, hal tersebut disebabkan oleh banyaknya orang-orang yang memahami *hadits* hanya dengan mengandalkan teks lahiriyah *hadits* saja, dimana hal tersebut dirasa sangat tidak cukup untuk mempelajari suatu *hadits* dengan sebenarnya. Salah satu hal yang dapat kita tempuh untuk mengetahui makna yang terkandung dalam sebuah hadits adalah dengan mempelajari *syarah hadits*, dimana mempelajari *syarah hadits* ini sangat penting dilakukan untuk meminimalisir kesalahan penafsiran terhadap suatu *hadits* agar kemudahan menjalankan ajaran Islam ikut terbantu sehingga memudahkan hidup berdasarkan petunjuk Rasulullah SAW guna kesuksesan kehidupan di dunia dan akhirat. Sejauh ini aplikasi *syarah hadits* yang dapat digunakan oleh orang awam masih terbatas. Karena sejauh ini aplikasi yang ada masih berbahasa full arab yang tidak semua orang dapat memahaminya. Oleh karena itu, diperlukan suatu sistem yang dapat digunakan untuk solusi permasalahan tersebut, yaitu Sistem Temu Kembali Informasi yang dapat dimanfaatkan sebagai solusi karena memberikan alternatif berupa metode *similarity* yang dapat digunakan untuk melakukan pencarian dokumen yang relevan dengan yang kita inginkan. Metode *similarity* yang penulis gunakan adalah *cosine similarity*, dimana metode ini merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* atau *query* dari sebuah dokumen sebagai ukuran. Untuk menghitung nilai similaritas, diperlukan pembobotan kata dengan metode TFIDF dan menerapkan *teks preprocessing* terlebih dahulu untuk memperkecil *term* sehingga bisa mempercepat proses perhitungan *term*. *Teks preprocessing* tersebut meliputi *tokenizing*, *stopword removal* atau *filtering*, dan *stemming*. Uji coba yang dilakukan adalah dengan pengujian *matrix confusion* dengan hasil: *recall* 88,7%, *precision* 100%, *accuracy* 88,73 %, dan *error rate* 11,27 %.

**Kata Kunci** : TF-IDF, *Cosine Similarity*, Sistem Temu Kembali Informasi, *Syarah*, *Hadits*, *Recall*, *Precision*, *Accuracy*, *Error Rate*.

**Jumlah Halaman** : xx + 130 halaman + Lampiran.

## **ABSTRACT**

**Ria Melita, Implementation Of Term Frequency Inverse Document Frequency and Cosine Similarity Methods In The Information Retrieval System to Know The Syarah Hadits with Web-Based (Case Study: Hadith Shahih Bukhari-Muslim), guided by Victor Amrizal, M.Kom and Hendra Bayu Suseno, M.Kom.**

*Hadith is a source of Islamic teachings besides the Qur'an. Without using the hadith, the syari'at of Islam can not be fully understood and can not be implemented. However, it is undeniable that today, not a few people are mistaken in understanding it, it is caused by the many people who understand the hadith only by relying on the lahiriyyah text of hadith, where it is considered very inadequate to learn a hadith in the fullest. One of the things that we can take to know the meaning contained in a hadith is to study the science of syarh hadith, where studying the science of this syarh hadith is very important to minimize the mistake of interpretation of a hadith so that ease of running the teachings of Islam helped to facilitate life based on the guidance of Rasulullah SAW for the success of life in the world and the hereafter. So far the application of syarh hadith that can be used by the common people is still limited. Because so far the existing applications are still full Arab language that not everyone can understand it. Therefore, we need a system that can be used for the solution of the problem, that's the Information Retrieval System which can be utilized as a solution because it provides an alternative in the form of similarity method that can be used to search documents relevant to what we want. Similarity method that writer use is cosine similarity, where this method is a method to calculate the similarity between two objects stated in two vector by using keywords or query from a document as a measure. To calculate the value of similarity, word weighting is required with the TFIDF method and applying preprocessing text first to minimize the term so that it can speed up the term calculation process. The preprocessing text includes tokenizing, stopword removal or filtering, and stemming. The test is done by testing the confusion matrix. The result of system test is 88.7% recall, precision 100%, accuracy 88,73%, and error rate 11,27%.*

**Keyword** : TF-IDF, Cosine Similarity, Information Retrieval System, Syarah, Hadith, Recall, Precision, Accuracy, Error Rate.

**Number of Pages** : xx + 130 pages + Attachment

## DAFTAR ISI

HALAMAN PERSETUJUAN .....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN ORISINALITAS .....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI.....	v
KATA PENGANTAR.....	vi
ABSTRAK.....	viii
<i>ABSTRACT</i> .....	ix
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL .....	xviii
DAFTAR LAMPIRAN .....	xx

### BAB I PENDAHULUAN

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian .....	6
1.5 Manfaat Penelitian .....	7
1.5.1 Bagi Pengguna .....	7
1.5.2 Bagi Mahasiswa .....	7
1.5.3 Bagi Universitas .....	7
1.6 Metode Penelitian .....	8
1.6.1 Metode Pengumpulan Data .....	8
1.6.1 Metode Pengembangan Sistem .....	8
1.7 Sistematika Penulisan .....	9

### BAB II LANDASAN TEORI

2.1 Hadits .....	11
2.1.1 Pengertian Hadits .....	11

2.1.2 Struktur Hadits.....	11
2.1.3 Fungsi Hadits Disamping Al-Qur'an.....	13
2.1.4 Hadits Bukhari-Muslim.....	14
2.2 <i>Syarah Hadits</i> .....	14
2.2.1 Pengertian <i>Syarah Hadits</i> .....	14
2.2.2 Unsur-Unsur Pen-syarah-an.....	15
2.2.3 Macam-macam <i>Syarah</i> .....	16
2.2.4 <i>Taysirul 'Allam Syarah 'Umdatil Ahkam</i> .....	16
2.3 <i>Text Mining</i> .....	17
2.4 Sistem Temu Kembali Informasi ( <i>Information Retrieval</i> ).....	18
2.4.1 Sistem Temu Kembali Informasi ( <i>Information Retrieval</i> ).....	18
2.4.2 Proses Sistem Temu Kembali Informasi .....	20
2.5 <i>Preprocessing Text</i> .....	21
2.5.1 Pengertian <i>Preprocessing Text</i> .....	21
2.5.2 Tahap-tahap <i>Preprocessing Text</i> .....	21
2.6 <i>Stemming</i> Nazief Adriani.....	22
2.6.1 Algoritma <i>Stemming</i> Nazief Adriani.....	22
2.6.2 Alasan Menggunakan <i>Stemming</i> Nazief Adriani.....	23
2.6.3 Aturan <i>Stemming</i> Nazief Adriani.....	23
2.7 <i>Term Frequency Inverse Document Frequency</i> (TF-IDF).....	26
2.8 <i>Cosine Similarity</i> .....	28
2.9 <i>Confusion Matrix</i> .....	30
2.10 Metode Pengumpulan Data.....	31
2.10.1 Pengertian Pengumpulan Data.....	31
2.10.2 Pengertian Studi Pustaka.....	31
2.10.3 Pengertian Wawancara.....	32
2.11 Metode Pengembangan Sistem.....	32
2.11.1 <i>Rapid Application Development</i> (RAD).....	32
2.11.1.1 <i>Requirement Planning</i> .....	33
2.11.1.2 <i>RAD Design Workshop</i> .....	33
2.11.1.3 <i>Implementation</i> .....	34

2.11.2 Perbandingan Metode Pengembangan Sistem.....	34
2.12 <i>Unified Modelling Language (UML)</i> .....	37
2.12.1 Pengertian UML.....	37
2.12.2 Jenis-Jenis Diagram UML.....	37
2.13 PHP .....	42
2.14 <i>Flowchart Diagram</i> .....	42
2.15 MySql.....	43
2.16 Xampp.....	44
2.17 <i>Database</i> .....	44
2.17.1 Pengertian Database.....	44
2.17.2 Tujuan Database.....	45
2.18 Alasan Membuat Sistem Berbasis Website.....	45
2.19 Studi Literatur Sejenis .....	46

### **BAB III METODE PENELITIAN**

3.1 Metode Pengumpulan Data.....	50
3.1.1 Metode Studi Pustaka.....	50
3.1.2 Metode Wawancara.....	50
3.2 Metode Pengembangan Sistem.....	51
3.2.1 Model <i>Rapid Application Development</i> .....	51
3.2.2 <i>Requirement Planning</i> .....	51
3.2.3 <i>RAD Design Workshop</i> .....	51
3.2.4 <i>Implementation</i> .....	52
3.3 Kerangka Berfikir.....	52

### **BAB IV ANALISA DAN PERANCANGAN SISTEM**

4.1 Fase Perencanaan Syarat.....	54
4.1.1 Mengidentifikasi Masalah.....	54
4.1.2 Menganalisis Solusi Permasalahan.....	55
4.1.3 Menganalisis Kebutuhan Sistem.....	56
4.2 Fase <i>Workshop Design</i> .....	56
4.2.1 Merancang Sistem.....	57
4.2.2 Merancang UML.....	72

4.2.3 Merancang <i>Database</i> .....	96
4.2.4 Merancang <i>User Interface</i> .....	99
4.2.5 Pengkodean ( <i>Coding</i> ).....	105
<b>BAB V HASIL DAN PEMBAHASAN</b>	
5.1 Fase Implementasi.....	113
5.1.1 Implementasi Perangkat.....	113
5.1.2 Hasil Penelitian.....	114
5.2 Pembahasan Penelitian.....	120
5.3 Pengujian ( <i>Testing</i> ).....	121
5.3.1 Pengujian Stemming Nazief Adriani.....	121
5.3.2 Pengujian Sistem.....	123
<b>BAB VI KESIMPULAN DAN SARAN</b>	
6.1 Kesimpulan.....	125
6.2 Saran.....	125
<b>DAFTAR PUSTAKA</b> .....	127
<b>LAMPIRAN</b>	

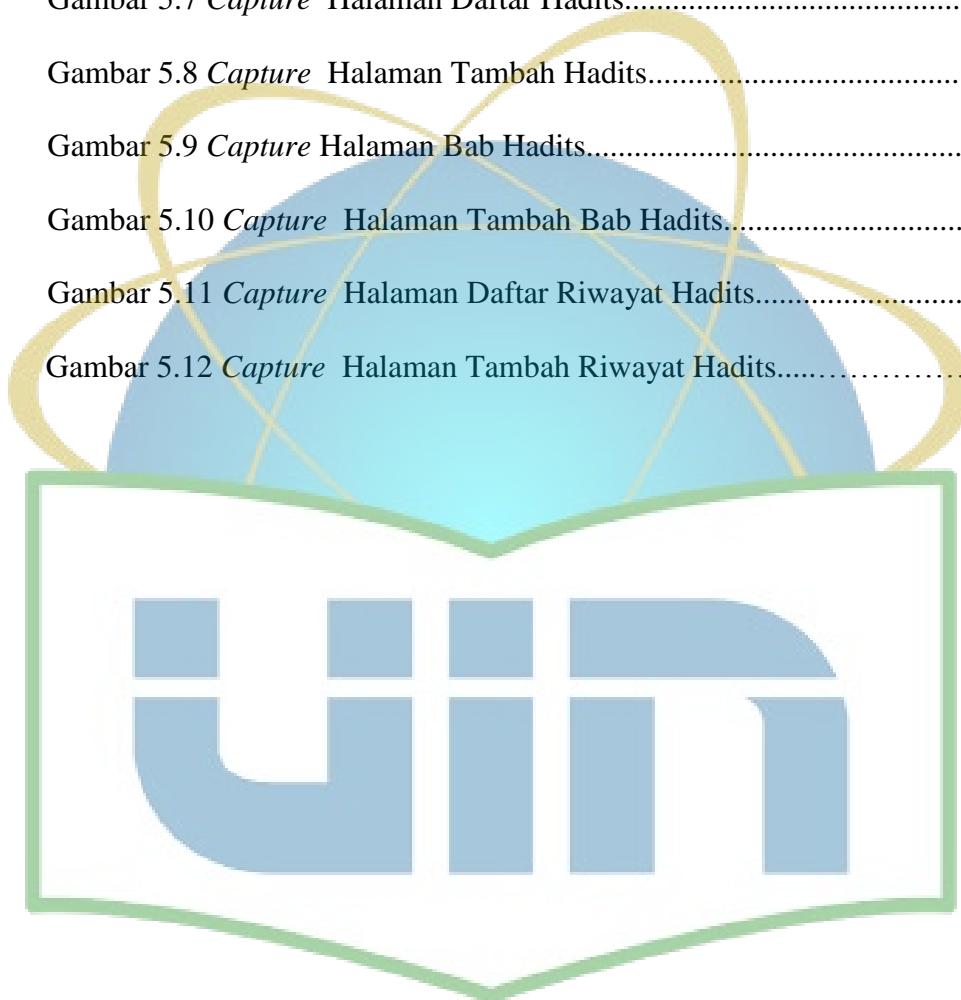
## DAFTAR GAMBAR

Gambar 2.1 Proses Sistem Temu Kembali Informasi.....	20
Gambar 2.2 Kerangka IRS.....	21
Gambar 2.3 Fase RAD.....	33
Gambar 3.1 Kerangka Berfikir.....	53
Gambar 4.1 <i>Flowchart Tokenizing</i> .....	58
Gambar 4.2 <i>Flowchart Stopword Removal</i> atau <i>Filtering</i> .....	60
Gambar 4.3 <i>Flowchart Stemming</i> .....	62
Gambar 4.4 <i>Flowchart Tf-Idf</i> .....	64
Gambar 4.5 <i>Flowchart Cosine Similarity</i> .....	68
Gambar 4.6 <i>Usecase Diagram</i> .....	73
Gambar 4.7 <i>Activity Diagram Login</i> .....	74
Gambar 4.8 <i>Activity Diagram</i> Daftar Hadits.....	74
Gambar 4.9 <i>Activity Diagram</i> Tambah Hadits.....	75
Gambar 4.10 <i>Activity Diagram</i> Hapus Hadits.....	75
Gambar 4.11 <i>Activity Diagram</i> Edit Hadits.....	76
Gambar 4.12 <i>Activity Diagram</i> Cari Hadits.....	76
Gambar 4.13 <i>Activity Diagram</i> Daftar Bab Hadits.....	77
Gambar 4.14 <i>Activity Diagram</i> Tambah Bab Hadits.....	77
Gambar 4.15 <i>Activity Diagram</i> Hapus Bab Hadits.....	78
Gambar 4.16 <i>Activity Diagram</i> Edit Bab Hadits.....	78
Gambar 4.17 <i>Activity Diagram</i> Cari Bab Hadits.....	79
Gambar 4.18 <i>Activity Diagram</i> Daftar Riwayat Hadits.....	79

Gambar 4.19 <i>Activity Diagram</i> Tambah Riwayat Hadits.....	80
Gambar 4.20 <i>Activity Diagram</i> Hapus Riwayat Hadits.....	80
Gambar 4.21 <i>Activity Diagram</i> Edit Riwayat Hadits.....	81
Gambar 4.22 <i>Activity Diagram</i> Cari Riwayat Hadits.....	81
Gambar 4.23 <i>Activity Diagram</i> Logout.....	82
Gambar 4.24 <i>Activity Diagram</i> Baca Hadits.....	82
Gambar 4.25 <i>Activity Diagram</i> Analisa Hadits.....	83
Gambar 4.26 <i>Sequence Diagram</i> Login.....	83
Gambar 4.27 <i>Sequence Diagram</i> Daftar Hadits.....	84
Gambar 4.28 <i>Sequence Diagram</i> Tambah Hadits.....	85
Gambar 4.29 <i>Sequence Diagram</i> Hapus Hadits.....	85
Gambar 4.30 <i>Sequence Diagram</i> Edit Hadits.....	86
Gambar 4.31 <i>Activity Diagram</i> Cari Hadits.....	87
Gambar 4.32 <i>Sequence Diagram</i> Daftar Bab Hadits.....	87
Gambar 4.33 <i>Sequence Diagram</i> Tambah Bab Hadits.....	88
Gambar 4.34 <i>Sequence Diagram</i> Hapus Bab Hadits.....	89
Gambar 4.35 <i>Sequence Diagram</i> Edit Bab Hadits/.....	89
Gambar 4.36 <i>Activity Diagram</i> Cari Bab Hadits.....	90
Gambar 4.37 <i>Sequence Diagram</i> Daftar Riwayat Hadits.....	91
Gambar 4.38 <i>Sequence Diagram</i> Tambah Riwayat Hadits.....	91
Gambar 4.39 <i>Sequence Diagram</i> Hapus Riwayat Hadits.....	92
Gambar 4.40 <i>Sequence Diagram</i> Edit Riwayat Hadits.....	93
Gambar 4.41 <i>Activity Diagram</i> Cari Riwayat Hadits.....	93

Gambar 4.42 <i>Sequence Diagram Logout</i> .....	94
Gambar 4.43 <i>Sequence Diagram Baca Hadits</i> .....	94
Gambar 4.44 <i>Sequence Diagram Analisa Hadits</i> .....	95
Gambar 4.45 <i>Class Diagram</i> .....	96
Gambar 4.46 <i>User Interface Halaman Utama</i> .....	99
Gambar 4.47 <i>User Interface Baca Hadits (user)</i> .....	99
Gambar 4.48 <i>User Interface Halaman Baca Hadits (user)</i> .....	100
Gambar 4.49 <i>User Interface Halaman Analisis Hadits (User)</i> .....	100
Gambar 4.50 <i>User Interface Halaman Login (Admin)</i> .....	101
Gambar 4.51 <i>User Interface Halaman Utama Admin</i> .....	101
Gambar 4.52 <i>User Interface Halaman Daftar Hadits</i> .....	102
Gambar 4.53 <i>User Interface Halaman Tambah Hadits</i> .....	102
Gambar 4.54 <i>User Interface Halaman Bab Hadits</i> .....	103
Gambar 4.55 <i>User Interface Halaman Tambah Bab Hadits</i> .....	103
Gambar 4.56 <i>User Interface Halaman Daftar Riwayat Hadits</i> .....	104
Gambar 4.57 <i>User Interface Halaman Tambah Riwayat Hadits</i> .....	104
Gambar 4.58 <i>Algoritma Tokenizing</i> .....	105
Gambar 4.59 <i>Algoritma Stopword Removal</i> .....	106
Gambar 4.60 Algoritma Stemming Nazief Adriani.....	109
Gambar 4.61 Algoritma TF-IDF.....	110
Gambar 4.62 Algoritma Cosine Similarity.....	112
Gambar 5.1 <i>Capture Halaman Utama</i> .....	114
Gambar 5.2 <i>Capture Baca Hadits (user)</i> .....	114

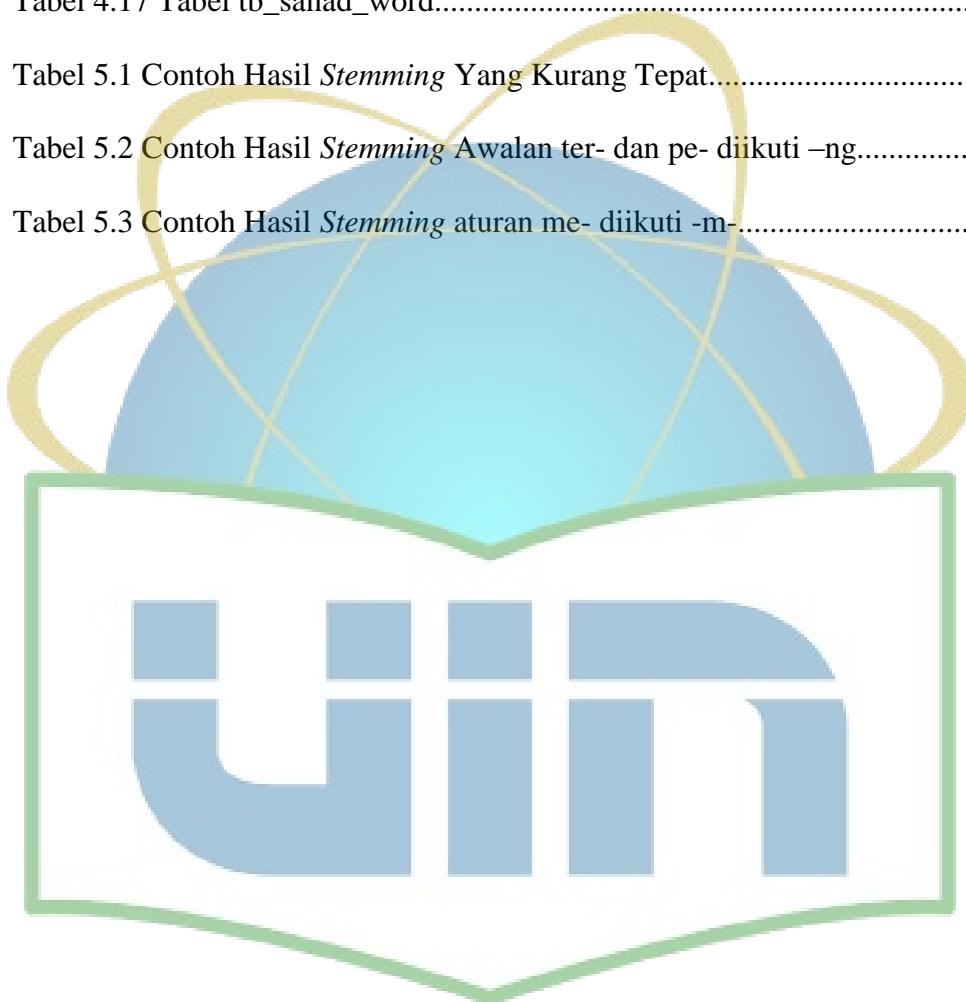
Gambar 5.3 <i>Capture</i> Halaman Baca Hadits ( <i>user</i> ).....	115
Gambar 5.4 <i>Capture</i> Halaman Analisa Hadits ( <i>User</i> ).....	116
Gambar 5.5 <i>Capture</i> Halaman Login (Admin).....	116
Gambar 5.6 <i>Capture</i> Halaman Utama Admin.....	117
Gambar 5.7 <i>Capture</i> Halaman Daftar Hadits.....	117
Gambar 5.8 <i>Capture</i> Halaman Tambah Hadits.....	118
Gambar 5.9 <i>Capture</i> Halaman Bab Hadits.....	118
Gambar 5.10 <i>Capture</i> Halaman Tambah Bab Hadits.....	119
Gambar 5.11 <i>Capture</i> Halaman Daftar Riwayat Hadits.....	119
Gambar 5.12 <i>Capture</i> Halaman Tambah Riwayat Hadits.....	119



## DAFTAR TABEL

Tabel 2.1 Kombinasi Awalan Akhiran yang Tidak Diizinkan.....	24
Tabel 2.2 Jenis Awalan Berdasarkan Tipe Awalan.....	25
Tabel 2.3 <i>Confusion Matrix</i> .....	30
Tabel 2.4 Perbandingan Metode Pengembangan Sistem.....	35
Tabel 2.5 Komponen Pembentuk <i>Use Case Diagram</i> .....	38
Tabel 2.6 Simbol <i>Class Diagram</i> .....	41
Tabel 2.7 Simbol-Simbol <i>Flowchart Diagram</i> .....	42
Tabel 2.8 Perintah Dasar MySQL .....	43
Tabel 2.9 Folder Penting Xampp.....	44
Tabel 2.10 Studi Literatur Sejenis.....	47
Tabel 4.1 Proses <i>Tokenizing</i> .....	58
Tabel 4.2 Hasil <i>Stopword Removal</i> atau <i>Filtering</i> .....	60
Tabel 4.3 Hasil <i>Stemming</i> .....	63
Tabel 4.4 Dokumen <i>Output</i> .....	65
Tabel 4.5 Hasil <i>Preprocessing Text</i> .....	65
Tabel 4.6 Hasil Pembobotan Kata ( <i>term</i> ).....	66
Tabel 4.7 Hasil Panjang Vektor.....	69
Tabel 4.8 Daftar Tabel dalam <i>Database</i> .....	97
Tabel 4.9 Tabel master_user.....	97
Tabel 4.10 Tabel isi_hadits.....	97
Tabel 4.11 Tabel bab_hadits.....	97
Tabel 4.12 Tabel riwayat_hadits.....	98

Tabel 4.13 Tabel tbcache.....	98
Tabel 4.14 Tabel tbindex.....	98
Tabel 4.15 Tabel tbvektor.....	98
Tabel 4.16 Tabel tb_katadasar.....	98
Tabel 4.17 Tabel tb_sanad_word.....	98
Tabel 5.1 Contoh Hasil <i>Stemming</i> Yang Kurang Tepat.....	121
Tabel 5.2 Contoh Hasil <i>Stemming</i> Awalan ter- dan pe- diikuti –ng.....	122
Tabel 5.3 Contoh Hasil <i>Stemming</i> aturan me- diikuti -m-.....	122



## **DAFTAR LAMPIRAN**

Lampiran I. SK Dosen Pembimbing.....	-1-
Lampiran II. Transkip Wawancara.....	-2-
Lampiran III. Transkip Wawancara Pengujian.....	-5-



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

*Hadits* merupakan sumber ajaran Islam disamping *Al-Qur'an*. Tanpa menggunakan *hadits*, syari'at Islam tidak dapat dimengerti secara utuh dan tidak dapat dilaksanakan. (Muh.Zuhri, 2011). *Hadits* sebagai sumber hukum kedua setelah *Al-Qur'an*, memiliki otoritas dan posisi penting yang nampaknya sudah tidak perlu dipertanyakan lagi, karena selain sebagai penjelas (*bayan*) bagi otoritas wahyu Allah (*Al-Qur'an*) juga sebagai sumber hukum Islam kedua yang menjadi rujukan para *fuqaha* yaitu ahli fiqh atau hukum islam. (Khoirul Rosyid, 2016)

Pentingnya posisi *Hadits* tehadap *Al-Qur'an* turut menempatkan posisi *Syarah Hadits* sebagai sesuatu yang tidak bisa diabaikan. Teks-teks *hadits* yang ada menjadi penjelas atas teks-teks *Al-Qur'an* yang dianggap masih bersifat global. Dalam hal ini, *hadits* merupakan entitas penting dalam bidang penafsiran *Al-Qur'an*. Begitu pula dalam tradisi *hadits*, *syarah* memegang peranan penting untuk menjelaskan hal-hal yang dianggap masih umum, sulit dipahami, terlihat bertentangan, maupun hal-hal yang menyimpan keganjilan dalam teks-teks *hadits*. Tanpa keberadaan *syarah hadits*, susah kiranya untuk memahami makna sesungguhnya dari suatu *hadits*. (Kholila Mukaromah, 2015)

Hasil wawancara yang telah penulis lakukan dengan Bapak Fiqri Muhammad Fatkhi, MA. selaku Ketua Program Studi Hadits Fakultas Ushuluddin Universitas Islam Negeri Syarif Hidayatullah Jakarta, didapatkan bahwa *hadits* mendapatkan perhatian lebih dari pada *Al-Qur'an*. Hal ini dikarenakan *hadits* berisi hal-hal yang praktis yang tidak terdapat dalam al-qur'an, seperti tata cara shalat, tentang bagaimana berhubungan dengan sesama dan lain sebagainya. Adapun berdasarkan hasil wawancara, beliau mengatakan bahwa "saat ini tidak sedikit orang yang keliru dalam memahami *hadits*, hal itu dikarenakan oleh 3 hal, yaitu : 1. Mereka menganut pemahaman

literalis (tekstualis), 2.Semangat keberagaman yang ingin kembali mengamalkan ajaran agama yang otentik sesuai dengan *Al-Qur'an* dan *hadits* tanpa didukung oleh literasi terhadap pendapat-pendapat ulama, dan 3. Keterbatasan kemampuan yang mereka miliki sehingga mereka lebih memilih atau hanya mengakses pada buku-buku terjemah sehingga tidak mewakili konten *hadits* dari apa yang disampaikan Nabi.”. Padahal menurut beliau, untuk mengetahui makna yang sebenarnya dari suatu *hadits* sangat tidak cukup jika hanya dengan mengandalkan teks lahiriyah *hadits* saja kecuali *hadits-hadits* yang berkaitan dengan akidah.

Menurut Bapak Fiqri Muhammad Fatkhi, hal yang dapat kita tempuh untuk mengetahui makna sebenarnya dari *hadits* adalah salah satunya dengan mempelajari *syarah hadits*, dimana mempelajari *syarah hadits* untuk mengetahui makna yang terkandung didalamnya adalah sangat penting untuk dilakukan sepanjang diperoleh dari orang yang memiliki otoritas dibidang penjelasan *hadits*, bukan hanya dari terjemahnya saja tanpa penjelasan maknanya.

Merujuk pada penjelasan beliau, sejauh ini aplikasi *syarah hadits* yang ada masih terbatas, yaitu dalam bahasa *full arab* yang tidak semua orang dapat memahaminya. Sedangkan untuk bahasa indonesia sendiri baru ada lidwa dan arbain, namun masih sangat luas jangkaunya yaitu dari berbagai perawi.

Berdasarkan pemaparan diatas, untuk mengetahui makna yang terkandung dari suatu *hadits* guna meminimalisir kesalahan penafsiran terhadap suatu *hadits* agar kemudahan menjalankan ajaran Islam ikut terbantu sehingga memudahkan hidup berdasarkan petunjuk Rasulullah SAW guna kesuksesan kehidupan di dunia dan akhirat, maka penulis ingin melakukan penelitian dengan merancang suatu sistem temu kembali informasi untuk mengetahui *syarah hadits* dengan bahasa indonesia yang berpedoman pada kitab *Taysirul 'Allam Syarah 'Umdatil Ahkam* yang akan dibuat dengan menerapkan metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity*.

*Taysirul 'Allam Syarah 'Umdatil Ahkam* adalah kitab *syarah* yang sangat populer di kalangan penuntut ilmu di seluruh dunia, termasuk pesantren-pesantren di Indonesia yang merupakan kumpulan *hadits-hadits* riwayat imam Bukhari-Muslim—keduanya sudah sangat *masyhur* di telinga kaum Muslimin seantero dunia sebagai jaminan keshahihan suatu *hadits*. (Ibnu Katsir, 2013).

Sistem temu kembali informasi ini merupakan suatu sistem yang menemukan (*retrieve*) informasi yang sesuai dengan kebutuhan *user* dari kumpulan informasi secara otomatis yang cocok digunakan untuk menemukan pernyataan pencarian dengan data yang tersimpan dalam *database* yang ada sesuai dengan *query* yang dimasukkan. Dengan begitu, penelitian ini diharapkan dapat membuat sistem yang dapat memberikan *output* berupa *syarah hadits* yang sesuai berdasarkan *input* yang dimasukkan dengan menggunakan Metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity*. Metode *Term Frequency Inverse Document Frequency* (TF-IDF) ini merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Sedangkan, Metode *cosine similarity* merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. (Rizki Tri Wahyuni, ddk. 2017).

Model ruang vektor dan pembobotan tf-idf digunakan untuk merepresentasikan nilai numerik dokumen yang kemudian dapat dihitung kedekatan antar dokumen sehingga dengan menggunakan metode ini diharapkan sistem dapat memberikan *output* yang sesuai berdasarkan *input* yang diberikan dengan memperhitungkan kemiripan antar keduanya. Metode tf-idf ini penulis gunakan, karena merupakan metode pembobotan kata yang terkenal efisien, mudah dan memiliki hasil yang akurat. (Robertson dan Stephen dalam Abdul Aziz Maarif, 2015). Sedangkan Metode *Cosine Similarity* penulis gunakan dengan alasan bahwa, metode ini mempunyai nilai akurasi yang tinggi dimana menurut (Rozas dan Sarno, 2011) kelebihan

utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen. Sehingga, dengan melakukan perbandingan *keyword* yang dihasilkan, maka kedekatan antara *item*-pun dapat dipastikan.

Penelitian terkait dengan menggunakan metode tf-idf dan *cosine similarity* ini telah dilakukan sebelumnya oleh Rizki Tri Wahyuni, Dhidik Prastyianto, dan Eko Supraptono dalam jurnalnya yang berjudul “Penerapan Algoritma *Cosine Similarity* dan Pembobotan *Term Frequency Inverse Document Frequency* pada Sistem Klasifikasi Dokumen Skripsi” mengenai perancangan sistem yang dapat mengklasifikasikan dokumen secara otomatis kedalam folder berbeda pada *database* agar lebih mudah dalam mengelola dokumen yang ada.

Penelitian yang dilakukan oleh Dewa Ayu, Arie Lumenta, dan Agustinus Jacobus dalam jurnalnya yang berjudul “Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode *Cosine Similarity*” yaitu penelitian yang mengukur kemiripan dokumen untuk mengetahui adanya plagiarisme terhadap suatu karya tulis.

Penelitian dengan judul “Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode *Term Frequency Inverse Document frequency*” yang dilakukan oleh Dhony Syafe’i, Sukmawati, dan Nurdin Bahtiar, mengenai pembuatan suatu perangkat lunak yang dapat mencari dokumen-dokumen penulisan ilmiah yang relevan sesuai tingkat pembobotannya.

Dari ketiga penelitian terkait yang telah penulis sebutkan diatas, terdapat beberapa irisan yang penulis tekanan dalam penelitian tersebut yaitu dalam penggunaan *stemming* pada *text preprocessing*. Dimana penelitian yang dilakukan oleh Rizki Tri Wahyuni, dkk dan Dewa Ayu, dkk tidak melakukan *stemming* pada *text preprocessing* dan penelitian yang dilakukan oleh Dhony Syafe’i menggunakan *Stemming* dengan jenis *Porter Stemmer* model Fadhillah Z.Tala. Oleh karena itu, penulis menggunakan *Stemming* model Nazief dan Adriani pada *text preprocessing* untuk meningkatkan performa sistem yang akan dibuat, dimana *stemming* model ini

merupakan *stemming* yang memiliki tingkat akurasi (presisi) lebih tinggi dibandingkan dengan *stemming* model lain. Penelitian yang penulis lakukan menggunakan metode yang sama yaitu *term frequency inverse document frequency* dan *cosine similarity* pada sistem temu kembali informasi, namun terdapat perbedaan yang mendasar dari semua penelitian yang ada yaitu bahwa penulis melakukan penerapan metode tersebut pada sistem temu kembali informasi untuk mengetahui *syarah hadits* yang belum pernah dilakukan sebelumnya.

Sistem temu kembali informasi ini akan dibuat dengan menggunakan aplikasi berbasis web, dimana aplikasi dengan basis web ini memiliki beberapa kelebihan yaitu tidak memerlukan proses instalasi, bersifat terpusat, dapat dijalankan di os manapun asalkan memiliki *browser* dan akses internet, dan tidak perlu spesifikasi komputer *client* yang tinggi.

Berdasarkan permasalahan yang penulis paparkan diatas, maka penulis tertarik untuk melakukan penelitian dengan judul “**PENERAPAN METODE TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF) DAN COSINE SIMILARITY PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI SYARAH HADITS BERBASIS WEB (STUDI KASUS: HADITS SHAHIH BUKHARI-MUSLIM)**”.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dikemukakan, maka rumusan masalah yang ada pada penelitian ini adalah:

1. Bagaimana menerapkan metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* pada sistem temu kembali informasi untuk mengetahui *syarah hadits* berbasis web?
2. Bagaimana menghitung nilai akurasi pada *Stemming Nazief* dan *Adriani* yang digunakan dalam *text preprocessing*?

### 1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, untuk membuat penelitian lebih terarah dan lebih fokus maka batasan masalah dari penelitian ini adalah:

1. Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode studi pustaka dan wawancara.
2. Metode pengembangan system yang digunakan adalah model *Rapid Application Development* (RAD) dengan tahapan fase perencanaan syarat, fase *workshop design*, dan fase implementasi.
3. Penelitian ini menggunakan metode *term frequency inverse document frequency* (tf-idf) sebagai pembobotan kata dan *cosine similarity* untuk menemukan dokumen yang relevan.
4. Menggunakan *stemming* model Nazief Adriani dalam *text preprocessing*.
5. Aplikasi ini berfokus memberikan *output* berupa *syarah* dari suatu *hadits* yang berupa point-point penting sesuai dengan teks berbahasa indonesia yang telah di *input*-kan.
6. *Hadits* yang digunakan adalah *hadits shahih* Imam Bukhari-Muslim saja.
7. Penelitian ini menggunakan buku *Taysirul 'Allam Syarah 'Umdatil Ahkam* yang berisi kumpulan *hadits-hadits* riwayat Bukhari-Muslim yang dilengkapi dengan penjelasan serta makna dari masing-masing *hadits*.
8. Hadits yang digunakan untuk data hanya berjumlah 204 dari 64 sub-bab yang merupakan belum keseluruhan dari isi buku *Taysirul 'Allam Syarah 'Umdatil Ahkam*.
9. Sistem yang dibangun menggunakan *framework Codeigniter* dengan *database MySQL* berbasis *web*.
10. Untuk perancangan aplikasi menggunakan *tools Unified Modelling Language* (UML) berupa *use case diagram*, *sequence diagram*, *activity diagram* dan *class diagram*.
11. Aplikasi ini menggunakan bahasa pemrograman *PHP*.

#### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* pada sistem temu kembali informasi untuk

mengetahui *syarah hadits* berbasis web (studi kasus: hadits shahih bukhari-muslim).

2. Menghitung nilai akurasi pada *Stemming* Nazief dan Adriani yang digunakan dalam *text preprocessing*.

## 1.5 Manfaat Penelitian

Berdasarkan dengan permasalahan dan tujuan penelitian, maka penulis mengharapkan penelitian ini dapat memberikan beberapa manfaat dibawah ini:

### 1.5.1 Bagi Pengguna

Dengan penelitian ini diharapkan dapat memudahkan pengguna untuk mengetahui *syarah hadits* agar tidak terjadi kesalahpahaman atau kekeliruan dalam mempelajari dan memahami suatu *hadits* tertentu dengan sebenarnya.

### 1.5.2 Bagi Mahasiswa

1. Memberikan pengetahuan lebih kepada penulis dalam membangun suatu sistem dengan menggunakan metode penelitian yang ada yaitu metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity*.
2. Lebih mengenal dan memahami pemaknaan *hadits* dengan benar.
3. Menerapkan ilmu-ilmu yang diperoleh selama perkuliahan di Universitas Islam Negeri Syarif Hidayatullah Jakarta.
4. Sebagai salah satu syarat dalam penyelesaian gelar Strata Satu (SI) program studi Teknik Informatika fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.

### 1.5.3 Bagi Universitas

1. Mengetahui kemampuan mahasiswa dalam menyerap ilmu yang telah diterima selama menjalankan masa pendidikan.
2. Mengetahui kemampuan mahasiswa dalam menerapkan ilmu yang telah diterima selama menjalankan masa pendidikan yang dapat digunakan sebagai bahan evaluasi mahasiswa.

## 1.6 Metode Penelitian

Metode penelitian yang digunakan dalam penyusunan penelitian ini adalah metode pengumpulan data dan metode pengembangan sistem sebagai berikut:

### 1.6.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan adalah sebagai berikut:

#### 1. Metode Studi Pustaka.

Dalam metode ini penulis melakukan studi pustaka dengan pencarian data secara manual maupun secara *online* melalui *browsing* internet dan melakukan pembelajaran terhadap jurnal dan skripsi yang relevan sesuai dengan topik penelitian serta melakukan pencarian dan pembelajaran mengenai buku yang membahas mengenai *hadits* terutama *hadits Bukhari-Muslim* dan buku mengenai metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* beserta buku yang menunjang dalam setiap proses penyelesaian penelitian ini.

#### 2. Metode Wawancara

Dalam metode wawancara ini penulis melakukan wawancara dengan narasumber yang menguasai hal yang berhubungan dengan permasalahan agar dapat diperoleh data-data dan informasi yang dibutuhkan sesuai dengan masalah yang akan diteliti.

### 1.6.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah model *Rapid Application Development* (RAD). Alasan mengapa penulis menggunakan metode pengembangan sistem model RAD dibanding model pengembangan sistem yang lainnya dikarenakan model ini cocok untuk mengerjakan suatu sistem dengan jangka waktu pengerjaan yang singkat dan tidak memerlukan sumber daya yang cukup besar. Dimana model ini dibagi menjadi 3 fase yaitu (**Kendall, 2010**) :

1. Fase Perencanaan Syarat
2. Fase *Design Workshop*
3. Fase Implementasi

*Tools* yang digunakan untuk perancangan aplikasi yaitu dengan UML (*Unified Modelling Language*), yang terdiri dari: *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram* (Evi Triandini & I Gede Suardika, 2012).

### 1.7 Sistematika Penulisan

Dalam menyusun penelitian ini penulis menggunakan sistematika penulisan sebagai berikut:

#### BAB I PENDAHULUAN

Pada Bab I yaitu pendahuluan ini dikemukakan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

#### BAB II LANDASAN TEORI

Pada bab landasan teori ini membahas mengenai dasar teori dari penelitian dan permasalahan yang berhubungan dengan topik serta penelitian yang terkait yang pernah dilakukan sebelumnya dan rencana penelitian yang akan dilakukan dalam penelitian selanjutnya.

#### BAB III METODOLOGI PENELITIAN

Pada bab ini membahas mengenai metode yang dipakai dalam melakukan penelitian yang meliputi metode pengumpulan data dan metode pengembangan sistem.

#### BAB IV ANALISA DAN PERANCANGAN

Pada bab ini menjelaskan mengenai analisa permasalahan, solusi permasalahan, perancangan sistem, alur kerja sistem, dan melakukan pengkodean.

## BAB V HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang hasil pembahasan yang diperoleh dari sistem yang telah dibuat untuk mewujudkan tujuan dan manfaat dari penelitian yang ingin dicapai.

## BAB VI PENUTUP

Pada bab ini di kemukakan kesimpulan yang diperoleh dari hasil penelitian berdasarkan rumusan masalah yang telah dikemukakan, serta saran untuk melakukan pengembangan pada penelitian berikutnya.



## BAB II

### LANDASAN TEORI

#### 2.1 Hadits

##### 2.1.1 Pengertian Hadits

Menurut (M. Abduh Almanar, 2012) *Hadits* secara etimologi berarti “baru dari segala sesuatu”. Kata *hadits* mengandung pengertian sedikit dan banyak. Firman Allah SWT:

فَأَعْلَكْ بُخْ نَفْسَكْ عَلَىٰ إِثْرِهِمْ إِن لَّمْ يُؤْمِنُوا بِهُدَا الْحَدِيثِ أَسْفَا

Artinya:

“Maka (apakah) barangkali kamu akan membunuh dirimu karena bersedih hati setelah mereka berpaling, sekiranya mereka tidak beriman kepada keterangan ini (*Al-Hadits*)”. (Q.S.Al-Kahfi:6).

Maksud kata *Hadits* dalam ayat tersebut adalah *Al-Qur'an*. Demikian pula firman Allah SWT:

وَأَمَّا بِذِيْمَةِ رَبِّكَ فَهَذِهِ

Artinya:

“Dan terhadap nikmat Tuhanmu maka hendaklah kamu menyebutnya (dengan bersyukur)”. (Q.S.Ad-Dhuha: 11).

Maksudnya, sampaikanlah risalah. Dengan demikian, secara etimologi kata *Hadits* sinonim dengan kata *khabar*.

Menurut M. Abduh Almanar (2012: 2) *Hadits* secara terminologis sinonim dengan *Sunnah*. Keduanya diartikan sebagai segala sesuatu yang diambil dari Rasul SAW sebelum dan sesudah diangkat menjadi Rasul. Akan tetapi, bila disebut kata *hadits*, umumnya dipakai sebagai segala sesuatu yang diriwayatkan dari Rasul SAW setelah kenabian, baik berupa sabda, perbuatan, maupun *taqrir*.

##### 2.1.2 Struktur Hadits

Menurut (M. Abduh Almanar, 2012) Struktur *hadits* meliputi *Sanad*, *Matan*, dan *Mukharij (rawi)*. Sebelum mendefinisikan istilah-istilah tersebut, kita perhatikan sebuah *hadits* yang mengandung ketiga istilah tersebut. Imam Bukhari meriwayatkan:

Berkata Imam Bukhari: telah menceritakan kepada kami Adam (ia berkata): Telah menceritakan kepada kami Ibnu Abi Dzi'b (ia berkata): Telah menceritakan kepada kami Sa'id Al-Maqburiy dari Abu Hurairah dari Nabi Shallallahu 'alaihi wa Sallam beliau bersabda; "Sesungguhnya akan datang kepada manusia satu zaman dimana seseorang tidak akan memperdulikan lagi tentang harta yang ia peroleh, apakah dari (hasil) yang halal atau dari (hasil) yang haram. (H.R. Bukhari).

Rangkaian kata "Berkata Imam Bukhari....." disebut *Sanad*, redaksi *hadits* "Sesungguhnya akan datang ....." disebut *Matan*, dan sebagai penutup *hadits* "Imam Bukhari" disebut *Mukhaarij (Rawi)*.

### **1. Pengertian Sanad**

Menurut (M. Abduh Almanar, 2012) *Sanad* secara terminologis adalah jalur matan, yakni rangkaian para perawi yang memindahkan matan dari sumber utamanya. Jalur itu disebut *Sanad* adakalanya karena periyawat bersandar kepadanya dalam menisbatkan matan kepada sumbernya, dan adakalanya karena para *Haafizh* (penghapal *hadits*) bertumpu kepada "yang menyebutkan *Sanad*" dalam mengetahui *Shahih* atau *Dha'if* suatu *hadits*.

### **2. Pengertian Matan**

Menurut (M. Abduh Almanar, 2012) *Matan* secara etimologis berarti segala sesuatu yang keras bagian atasnya. Bentuk jamaknya adalah *Mutun* (الْمُتَّنْ) dan *Mitan* (الْمِتَانْ). *Matan* dari segala sesuatu adalah bagian permukaan yang tampak darinya, juga bagian yang tampak menonjol dan keras.

*Matan* secara terminologis adalah redaksi hadits yang menjadi unsur pendukung pengertiannya. Penamaan seperti itu barangkali didasarkan pada alasan bahwa bagian itulah yang tampak dan yang menjadi sasaran utama *hadits*. Jadi penamaan itu diambil dari pengertian etimologisnya.

### 3. Pengertian *Mukhaarij*

Menurut (M. Abduh Almanar, 2012) *Mukharij* sama dengan *Rawi* (الرَاوِيْ , secara etimologis berarti : orang yang mengeluarkan atau yang meriwayatkan. Jadi, pengertian terminologisnya adalah orang yang mengeluarkan atau yang meriwayatkan sebuah hadits atau orang yang menuliskan sebuah *hadits* Nabi SAW. dalam bahasa Arabnya مخارج :

#### 2.1.3 Fungsi Hadits Disamping Al-Qur'an

Menurut (Idri, 2010) dilihat dari segi posisinya, *Al-Qur'an* dan *hadits* merupakan pedoman hidup dan sumber ajaran Islam, antara keduanya tidak dapat dipisahkan. *Al-Qur'an* sebagai sumber yang memuat ajaran-ajaran yang bersifat umum dan global, yang perlu dijelaskan lebih lanjut dan terperinci. Oleh karena itu fungsi *hadits* sebagai penjelas *al-qur'an* yang mempunyai enam fungsi, yaitu:

##### 1. *Bayan Al-Taqrir*

*Bayan Al-Taqrir* disebut pula *Bayan al-taqyid* dan *Bayan al-itsbat*, yaitu menetapkan dan memperkuat apa yang telah diterangkan dalam *Al-qur'an*.

##### 2. *Bayan Tafshil*

*Bayan Tafshil* berarti penjelasan dengan memerinci kandungan ayat-ayat yang *mujmal*, yakni ayat-ayat yang bersifat ringkas atau singkat, sehingga maknanya kurang atau bahkan tidak jelas kecuali ada penjelasan atau perincian. Dengan kata lain, ungkapan ayat itu masih bersifat global yang memerlukan *mubayyin*.

##### 3. *Bayan Taqyid*

*Bayan Taqyid* adalah penjelasan hadis dengan cara membatasi ayat-ayat yang bersifat mutlak dengan sifat, keadaan, atau syarat tertentu.

##### 4. *Bayan Takhshish*

*Bayan Takhshish* adalah penjelasan Nabi dengan cara membatasi atau mengkhususkan ayat-ayat al-Qur'an yang bersifat umum, sehingga tidak berlaku pada bagian-bagian tertentu yang mendapat perkecualian.

## 5. *Bayan Tasyri*

*Bayan Tasyri* adalah penjelasan hadits yang berupa penetapan suatu hukum atau aturan syar'i yang tidak didapati nash-nya dalam *al-Qur'an*.

## 6. *Bayan Nasakh*

*Bayan Nasakh* adalah penjelasan hadis yang menghapus ketentuan hukum yang terdapat dalam *al-Qur'an*.

### 2.1.4 Hadits Bukhari Muslim

Menurut Al Imam Ibnu Katsir –*rahimahullah*- (muslim.or.id, 2012) Orang pertama yang memiliki perhatian untuk mengumpulkan *hadits-hadits shahih* secara khusus adalah Abu Abdillah Muhammad bin Isma'il Al Bukhari (Imam Al Bukhari) dan diikuti oleh sahabat sekaligus muridnya, Muslim bin Al-Hajjaj bin Muslim Al-Qusyairi An-Naisaburi (Imam Muslim). Dan dari total tujuh kitab *hadits* yang paling populer, dua kitab *hadits* dari Imam Bukhari-Muslim merupakan kitab *hadits* yang dinilai oleh para ulama ahli *hadits*, sebagai kitab *hadits* yang paling *shahih*. *Hadits-hadits shahih* yang dikeluarkan oleh kedua pemuncak ilmu *hadits* itu sering disebut “Rowahu Bukhari-Muslim” atau “Mutafaq ‘alaihi”.

Oleh Karena itu, penulis memilih *hadits shahih* Imam Bukhari-Muslim sebagai studi kasus dalam penelitian ini dikarenakan dua Imam tersebut merupakan Imam dengan Kitab yang tidak dapat diragukan lagi ke *shahihannya*.

## 2.2 *Syarah Hadits*

### 2.1 Pengertian *Syarah Hadits*

Menurut Muhammad bin Ya'qub (dalam Kholila Mukaromah, 2015), jika berbicara mengenai terma “*syarah hadits*”, maka diketahui bahwa terma ini merupakan gabungan dari kata “*syarah*” dan “*hadits*”. Kata ”*syarah*” merupakan adaptasi dari kata berbahasa Arab *syaraha-yasyrahu-syarhan* yang menunjukkan beberapa makna berikut : membuka (*fataha*); menjelaskan (*bayana*); menyingkap (*kasyafa*); memutuskan (*qata'a*); memahami (*fahima*); dan meluaskan (*wasa'a*). Terma ini kemudian digunakan untuk menunjuk sebuah penjelasan terhadap obyek studi di segala bidang ilmu

pengetahuan khususnya dalam studi agama yang menggunakan bahasa Arab. Ketika kata “*syarah*” ini disandingkan dengan kata “*hadits*”, maka bisa dipahami bahwa maksud dari *syarah hadits* adalah suatu aktivitas penyingkapan atau penjelasan atas makna-makna dan pemahaman atas sesuatu yang disandarkan pada Rasulullah SAW baik yang berupa perkataan, perbuatan, ketetapan, maupun sifat-sifatnya. (Muhammad ‘Ajjaj al-Khattib dalam Kholila Mukaromah, 2015).

Kata *syarah hadits* memiliki padanan dengan terma lain dalam bahasa Arab, seperti: *fiqh al-hadits*, *ma'an al-hadits*, dan juga *tafsir al-hadits*. Muhammad Thahir al-Jawwabi menyatakan bahwa istilah *syarah hadits* yang dikenal luas saat ini merupakan transformasi dari istilah *fiqh al-hadits*. Objek dari aktivitas pen-syarahannya ini tidak lain adalah *hadits* Nabi SAW dari segi keseluruhan kaidah-kaidah serta persoalan-persoalan yang terkait dengan penjelasan makna-makna *hadits*. Diantara yang dimaksud antara lain: pengungkapan maksud, tujuan, dan makna suatu *hadits* secara umum, penjelasan atas adanya pertentangan dan *nasakh*, serta penjelasan kata-kata (menurut konteks disabdkannya). (Kholila Mukaromah, 2015).

## 2.2 Unsur-Unsur Pen-syarah-an

Menurut Hasan Asy-ari (dalam Kholila Mukaromah, 2015), Pen-syarah-an *hadits* memiliki unsur-unsur aspek sebagai berikut:

1. Penjelasan umum yang meliputi : penjelasan bunyi lafad (*huruf wa syakl*), kaedah bahasa (*nahw wa sarf*), arti kamus (*ma'na lugawi*), arti istilah atau maksud (*ma'na istilahi*).
2. Aspek *sanad* yang meliputi : penjelasan nama seluruh *rijal* atau sebagian *rijal*, penilaian terhadap *rijal* dan sebab penilaian tersebut (*sabab al-jarh wa at-ta'dih*), nilai status *hadits*, serta argumentasi nilai status tersebut.
3. Aspek *matan* yang meliputi : penjelasan kata perkata, penjelasan perkalamat, penjelasan keseluruhan matan, penjelasan kata-kata yang sulit (*garib*), penjelasan *matan* atau redaksi lain sebagai *syahid*.

4. Aspek pemahaman yang meliputi : penjelasan hukum yang terkandung didalamnya, pendapat multi madzhab, pendapat aliran tertentu, pendapat satu mazhab saja, pemaparan pendapat sendiri, penjelasan dalil yang digunakan mazhab, penjelasan terkait faedah dan hikmah, dan penjelasan pendapat *syarah* terdahulu.

### **2.3 Macam-Macam *Syarah***

Menurut Hasan Asy-ari (dalam Kholila Mukaromah, 2015), berdasarkan pendekatan yang digunakan, *syarah* terbagi kedalam 3 macam:

1. *Syarah* hukum (Fikih) yang menonjol dalam menjelaskan *nash* yang ditandai dengan penekanan atas penyertaan penjelasan hukum, serta pendapat mazhab dan juga dalil yang digunakan.
2. *Syarah* kebahasaan yang lebih menonjolkan unsur-unsur penjelasan buniyi lafad, kaidah bahasa, arti kamus, dan juga penjelasan maksud (*ma'na istilahi*).
3. *Syarah* komprehensif yang mengandung hampir keseluruhan unsur yang ada.

### **2.4 *Taysirul 'Allam Syarah 'Umdatil Ahkam***

Kitab *Taysirul 'Allam Syarah 'Umdatil Ahkam* karya Syaikh Abdullah bin Abdurrahman bin Shalih Ali Bassam ini adalah kitab yang memuat *hadits-hadits shahih* bukhari-muslim dengan pendekatan disiplin ilmu fikih. Dimana buku ini berisi penjelasan makna *hadits* dengan disertai pembahasan kosakata yang dirasa memerlukan penjelasan sehingga hadits tertentu dapat dipahami secara utuh, mudah dan lebih sempurna. Selain itu, dalam buku ini juga dicantumkan perbedaan pendapat terkait persoalan-persoalan makna yang terkandung didalam *hadits* dengan disertai penjelasan mana pendapat yang lebih kuat, atau *rajih* dalam perbedaan pendapat tersebut sehingga memudahkan pembaca untuk menarik kesimpulan.

Kitab *Al-Umdah* ini merupakan pilihan dari *atsar-atsar* Nabi SAW paling *shahih* bersumber dari dua kitab agung; *Shahih al-Bukhari* dan *Shahih Muslim* yang dapat dijadikan rujukan dalam pembelajaran *hadits* bagi siapapun yang sudah mencapai tingkatan mahir, sekaligus tangga bagi pemula

menuju kitab-kitab Islam yang diriwayatkan dari manusia terbaik, Rasulullah SAW.

Dibandingkan dengan kitab *syarah* lain seperti *syarah Al-Allamah ahli ijtihad*, dan Ibnu Daqiq Id yang beredar luas dimana metode pembahasan yang digunakan penulis rumit dan kuat, jauh diatas pemahaman sebagian besar para penuntut ilmu dan mereka yang mencari informasi, keutamaan dari kitab ini adalah menggunakan bahasa yang mudah, mudah dipahami, keterangannya tidak bertele-tele, ringkas, dan jelas bagian-bagiannya, agar permasalahan-permasalahan yang ada beserta penjelasannya tidak tumpang tindih dan bercampur, yang tentu saja akan membingungkan. Kitab ini menjelaskan makna *hadits* secara global dengan menyesuaikan redaksi *hadits*, menjelaskan hikmah dan hukum yang tertera dibalik kosa kata *hadits*, atau memberikan pendahuluan dan penjelasan lain yang diisyaratkan oleh rangkaian kalimat kosakata *hadits*. Selain itu kitab ini juga menyimpulkan hukum dan adab yang ditunjukkan *hadits* berupa *point-point* penting yang peneliti gunakan sebagai *output* penelitian ini. Dimana dalam kesimpulan itu terdapat penjelasan hikmah syariat, keindahan dan tujuan-tujuan mulia Islam di balik *nash-nash* yang tertera, agar dapat direnungkan apa saja keindahan agama Islam dan tujuan-tujuannya nan mulia.

Berdasarkan pemaparan tersebut, dimana kitab ini lebih mudah dipahami dalam pembahasan *syarah* yang berupa *point-point* penting yang langsung terhadap kesimpulan makna dari suatu *hadits* untuk memudahkan pembaca, maka penulis menggunakannya sebagai data penelitian.

### 2.3 *Text Mining*

Menurut (Ronen Feldman dan James Sanger, 2007) *text mining* dapat didefinisikan secara luas sebagai suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi. *Text mining* bisa dianggap subjek riset yang tergolong baru. *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian atau pengelompokan dan menganalisa

*unstructured text* dalam jumlah besar. Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *Data mining*, *Information Retrieval*, Statistik dan Matematik, *Machine Learning*, *Linguistic*, *Natural Language Processing*, dan *Visualization*. Kegiatan riset untuk *text mining* antara lain ekstraksi dan penyimpanan text, *preprocessing* akan konten text, pengumpulan data statistik dan *indexing* dan analisa konten.

Tujuan dari *Text Mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen tetapi Tujuan utama *text mining* adalah mendukung proses *knowledge discovery* pada koleksi dokumen yang besar. Jadi, sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokkan teks (*text clustering*). (Feldman, dan James. S, 2007)

## 2.4 Sistem Temu Kembali Informasi (*Information Retrievel*)

### 2.4.1 Sistem Temu Kembali Informasi (*Information Retrievel*)

*Information Retrieval* (IR) merupakan bagian dari *computer science* yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. *Information Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (inputan *user*) yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada. Sedangkan, definisi *query* dalam IR menurut referensi merupakan sebuah formula yang digunakan untuk mencari informasi yang dibutuhkan oleh *user*, dalam bentuk yang paling sederhana, sebuah *query* merupakan suatu *keywords* (kata kunci) dan dokumen yang mengandung *keywords* merupakan dokumen yang dicari dalam sistem *information retrieval* (Manning, Raghavan, & Schütze, 2009).

Sistem temu kembali informasi merupakan suatu sistem yang menemukan (*retrieve*) informasi yang sesuai dengan kebutuhan *user* dari

kumpulan informasi secara otomatis. Prinsip kerja sistem temu kembali informasi jika ada sebuah kumpulan dokumen dan seorang *user* yang memformulasikan sebuah pertanyaan (*request* atau *query*). Jawaban dari pertanyaan tersebut adalah sekumpulan dokumen yang relevan dan membuang dokumen yang tidak relevan. (Salton, 1989, dalam Fatkhul Amin dan Purwatiningsyah, 2015).

*Information Retrieval System* dibagi dalam dua komponen utama yaitu sistem pengindeksan (*indexing*) menghasilkan basisdata sistem dan temu kembali merupakan gabungan dari *user interface* dan *look-up-table*. *Information Retrieval System* didesain untuk menemukan dokumen atau informasi yang diperlukan oleh *user*. *Information Retrieval System* bertujuan untuk menjawab kebutuhan informasi *user* dengan sumber informasi yang tersedia dalam kondisi seperti sebagai berikut (Salton, 1989, dalam Fatkhul Amin dan Purwatiningsyah, 2015);

1. Mempresentasikan sekumpulan ide dalam sebuah dokumen menggunakan sekumpulan konsep.
2. Terdapat beberapa pengguna yang memerlukan ide, tapi tidak dapat mengidentifikasi dan menemukannya dengan baik.
3. Sistem temu kembali informasi bertujuan untuk mempertemukan ide yang dikemukakan oleh penulis dalam dokumen dengan kebutuhan informasi pengguna yang dinyatakan dalam bentuk *keyword query* atau istilah penelusuran.

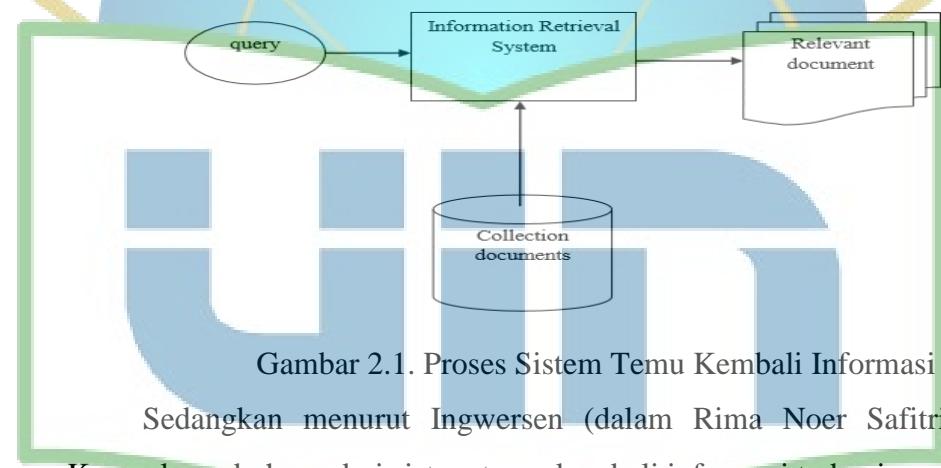
Fungsi utama sistem temu kembali informasi (Salton, 1989, dalam Fatkhul Amin dan Purwatiningsyah, 2015);

1. Mengidentifikasi sumber informasi yang relevan dengan minat masyarakat pengguna yang ditargetkan.
2. Menganalisis isi sumber informasi (dokumen).
3. Merepresentasikan isi sumber informasi dengan cara tertentu yang memungkinkan untuk dipertemukan dengan pertanyaan pengguna.

4. Merepresentasikan pertanyaan (*query*) *user* dengan cara tertentu yang memungkinkan untuk dipertemukan sumber informasi yang terdapat dalam basis data.
5. Mempertemukan pernyataan pencarian dengan data yang tersimpan dalam basis data.
6. Menemukan-kembalikan informasi yang relevan.
7. Menyempurnakan unjuk kerja sistem berdasarkan umpan balik yang diberikan oleh *user*.

#### 2.4.2 Proses Sistem Temu Kembali Informasi

Menurut (Rima Noer Safitri, 2013) Proses dalam Sistem Temu Kembali Informasi dapat digambarkan sebagai sebuah proses untuk mendapatkan *relevant documents* dari *collection documents* yang melalui pencarian *query* yang diinputkan *user* seperti gambar dibawah ini;

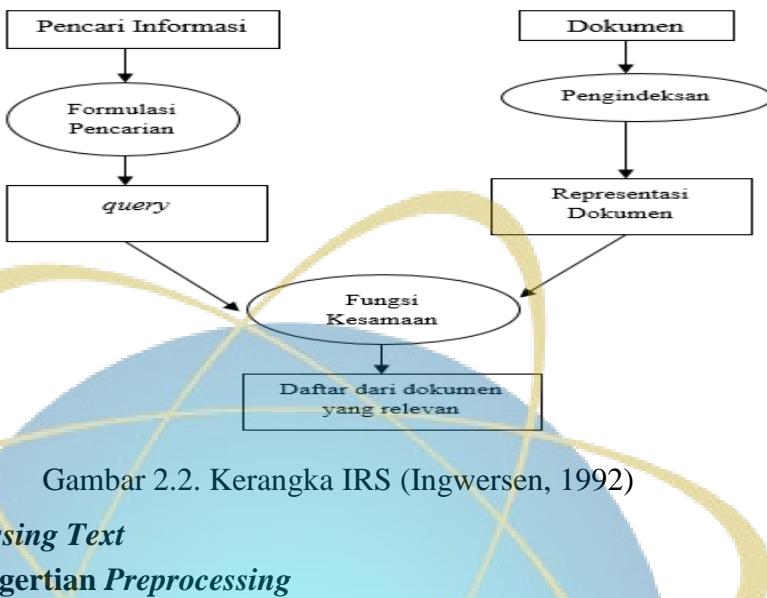


Gambar 2.1. Proses Sistem Temu Kembali Informasi

Sedangkan menurut Ingwersen (dalam Rima Noer Safitri, 2013)

Kerangka sederhana dari sistem temu-kembali informasi terbagi menjadi dua bagian. Bagian yang pertama adalah bagian pengguna dari sistem. Pengguna dari sistem temu-kembali informasi harus menerjemahkan informasi yang dicarinya agar dapat diproses oleh sistem dengan cara memasukan *query*. *Query* tersebut nanti di proses menjadi sebuah *query* yang dapat dimengerti oleh komputer. Bagian yang kedua adalah bagian dari dokumen. Pada bagian ini dokumen-dokumen direpresentasikan dalam bentuk indeks. *Query* dari pengguna akan diproses melalui fungsi kesamaan untuk membandingkan *query* dengan indeks dari dokumen untuk mendapatkan dokumen yang

relevan. Untuk lebih jelasnya mengenai kerangka sistem temu kembali informasi dapat dilihat pada Gambar dibawah ini:



Gambar 2.2. Kerangka IRS (Ingwersen, 1992)

## 2.5 Preprocessing Text

### 2.5.1 Pengertian Preprocessing

*Text Preprocessing* merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Suatu teks tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan *preprocessing text* untuk mengubah teks menjadi data *numeric*. Sebuah teks yang ada harus dipisahkan, hal ini dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen atau *tweet* dapat dipecah menjadi bab, sub-bab, paragraf, kalimat dan pada akhirnya menjadi potongan kata/token. Selain itu pada tahapan ini keberadaan digit angka, huruf kapital, atau karakter-karakter yang lainnya dihilangkan dan dirubah (Ronen Feldman, 2007).

Menurut (Nugroho, 2011) Proses *Preprocessing* berfungsi untuk proses awal sebelum dokumen teks diolah pada tahap selanjutnya dimana akan dilakukan proses seleksi data yang akan di proses pada setiap dokumen. Proses ini terdiri dari beberapa proses pembersihan dokumen, yaitu *case folding*, *tokenizing*, *filtering* atau *stopword removal*, dan *stemming*.

### 2.5.2 Tahap Preprocessing Text

Tahap-tahap *preprocessing text* yang digunakan adalah sebagai berikut (Rizki, Dhidik, dan Eko Supraptono. 2017) ;

## 1. Tokenizing

*Tokenizing* adalah proses memecah dokumen menjadi kumpulan kata. *Tokenization* dapat dilakukan dengan menghilangkan tanda baca dan memisahkannya per-spasi. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).

## 2. Stopword/Filtering

*Stopwords removal* merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil *parsing* deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak. Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan di-*remove* dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi dianggap sebagai kata-kata penting atau *keywords*.

## 3. Stemming

Menurut (Devi Basuma, 2013) *stemming* adalah salah satu cara yang digunakan untuk meningkatkan performa sistem *information retrieval* dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya untuk mengurangi ukuran *index* atau *record*, contohnya kata-kata menyuksesan, tersuksesan dan disukseskan akan ditransformasi ke stem yang sama yaitu sukses.

### 2.6 Stemming Nazief dan Adriani

#### 2.6.1 Algoritma Stemming Nazief Adriani

Menurut (Ivan Arwanda, 2013) *Stemming* merupakan bagian yang tidak terpisahkan dalam *Information Retrieval* (IR). Tidak banyak algoritma yang dikhususkan untuk stemming bahasa Indonesia dengan berbagai keterbatasan didalamnya. Algoritma *Porter* misalnya, algoritma ini membutuhkan waktu yang relatif lebih singkat dibandingkan dengan stemming menggunakan algoritma Nazief dan Adriani, namun proses *stemming* menggunakan algoritma *Porter* memiliki persentase keakuratan lebih kecil dibandingkan dengan *stemming* menggunakan algoritma Nazief dan Adriani. Algoritma Nazief dan Adriani sebagai algoritma *stemming* untuk teks berbahasa

Indonesia yang memiliki kemampuan persentase keakuratan lebih baik dari algoritma lainnya. Algoritma ini sangat dibutuhkan dan menentukan dalam proses IR dalam dokumen berbahasa Indonesia. Algoritma *stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya. Sebagai contoh bahasa Inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma *stemming* untuk kedua bahasa tersebut juga berbeda.

### 2.6.2 Alasan Menggunakan *Stemming* Nazief Adriani

Pada tahap *stemming* dibutuhkan suatu algoritma, oleh karena itu penulis menggunakan Algoritma *Stemming* Nazief dan Adriani yang menurut (Manase Sahat, 2017) dalam penelitiannya yang berjudul “Studi Perbandingan Algoritma -Algoritma *stemming* Untuk Dokumen Teks Bahasa Indonesia” menunjukkan bahwa diantara metode *Stemming* yang berbasis kamus (seperti *stemming* Nazief Adriani, dan *stemming* Arifin Setiono) maupun yang berbasis non-kamus (seperti algoritma *stemming* Vega dan algoritma *stemming* Tala) memberikan hasil bahwa algoritma *stemming* yang paling baik dan memiliki tingkat akurasi paling tinggi adalah algoritma Nazief Adriani dibanding algoritma *stemming* lain. Meskipun dari segi waktu, Nazief Adriani termasuk lebih lama dibandingkan dengan algoritma vega dan algoritma tala, namun dalam penelitian ini penulis memilih Nazief Adriani dikarenakan untuk suatu sistem yang terkait *hadits* lebih diutamakan keakuratan ketimbang waktu untuk meminimalisir kesalahan yang akan dilakukan sistem.

### 2.6.3 Aturan *Stemming* Nazief Adriani

Menurut Nazief, B.A.A. dan Andriani (dalam Manase Sahat, 2017) Algoritma Nazief & Adriani mengacu pada aturan morfologi bahasa Indonesia yang mengelompokkan imbuhan, yaitu imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan. Pengelompokan ini termasuk imbuhan di depan (awalan), imbuhan kata di belakang (akhiran), imbuhan kata di tengah (sisipan) dan kombinasi imbuhan pada awal dan akhir kata (konfiks). Algoritma ini menggunakan kamus kata keterangan yang

digunakan untuk mengetahui bahwa proses stemming telah mendapatkan kata dasar.

DP + DP + DP + root word + DS + PP + P

Langkah-langkah pada Algoritma Nazief & Adriani adalah:

1. Kata yang belum di-stemming dicari pada kamus. Jika kata itu langsung ditemukan, berarti kata tersebut adalah kata dasar. Kata tersebut dikembalikan dan algoritma dihentikan.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa partikel (“-lah”, “-kah”, “-tah”, atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possessive Pronouns (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an”, atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
  - a. Jika “-an”, telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus kata dasar maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
  - b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix DP* (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te”) dengan iterasi maksimum adalah 3 kali.
  - a. Langkah 4 berhenti jika :
    1. Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada tabel 2.1 dibawah ini.

Tabel. 2.1 Kombinasi Awalan Akhiran yang Tidak Diizinkan

<b>Awalan</b>	<b>Akhiran yang tidak diizinkan</b>
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

2. Tiga awalan telah dihilangkan.
- b. Tipe awalan ditentukan melalui langkah-langkah berikut:
1. Jika awalannya adalah : “di-”, “ke-”, atau “se-”, maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
  2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-”, maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.

Tabel. 2.2 Jenis Awalan Berdasarkan Tipe Awalan

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V...   be-rV...
2	berCAP...	ber-CAP... dimana C != 'r' dan P != 'er'
3	berCAPerV...	ber-CaerV... dimana C != 'r'
4	belajar	bel-ajar
5	berC1erC2...	be-C1erC2... dimana C1 != {r l}
6	terV...	ter-V...   te-rV...
7	terCerV...	ter-CerV... dimana C != 'r'
8	terCP	ter-CP... dimana C != 'r' dan P != 'er'
9	teC1erC2	te-C1erC2... dimana C1 != 'r'
10	me{l r}w{y}V...	me- {l r w y} V...
11	mem{b f v}...	mem-{b f v}...
12	mempe{r l}...	mem-pe...
13	mem{rV V}...	me-m{rV V}...   me-p{rV V}...
14	men{c d j z}...	men-{c d j z}...
15	menV...	me-nV...   me-tV...
16	meng{g h q}...	meng-{g h q}...
17	mengV...	meng-V...   meng-kV...
18	meny...	meny-sV...
19	mempV...	mem-pV... dimana V != 'e'
20	pe{w y}V...	pe-{w y}V...
21	perV...	per-V...   pe-rV...
22	perCAP	per-CAP... dimana C != 'r' dan P != 'er'

Aturan	Format Kata	Pemenggalan
23	perCAerV...	per-CAerV... dimana C != 'r'
24	pem {b f v} ...	pem- {b f V} ...
25	pem {rV V} ...	pe-m {rV V} ...   pe-p {rV V} ...
26	pen {c d j z} ...	pen- {c d j z} ...
27	penV...	pe-nV...   pe-tV...
28	peng {g h q} ...	peng- {g h q} ...
29	pengV...	peng-V...   peng-kV...
30	penyV...	peny-sV...
31	pelV...	pe-IV... kecuali "pelajar" yang menghasilkan "ajar"
32	peCerV...	per-erV... dimana C != {r w y l m n}
33	perCP...	pe-CP... dimana C != {r w y l m n} dan P != 'er'

3. Cari kata yang telah dihilangkan awalannya ini di dalam kamus.

Apabila tidak ditemukan, maka langkah 4 diulangi kembali.

Apabila ditemukan, maka keseluruhan proses berhenti.

5. Setelah tidak ada lagi imbuhan yang tersisa, maka algoritma ini dihentikan kemudian kata dasar tersebut dicari pada kamus, jika kata dasar tersebut ketemu berarti algoritma ini berhasil tapi jika kata dasar tersebut tidak ketemu pada kamus, maka dilakukan *recoding*.
6. Jika semua langkah telah dilakukan tetapi kata dasar tersebut tidak ditemukan pada kamus juga maka algoritma ini mengembalikan kata yang asli sebelum dilakukan *stemming*.

## 2.7 Term Frequency Inverse Document Frequency (TF-IDF)

Menurut Robertson dan Stephen (dalam Abdul Aziz Maarif, 2015) Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval*. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat.

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk

menevaluasi seberapa penting sebuah kata di dalam sebuah dokumen atau dalam sekelompok kata. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. (Agung Auliaguntary, 2016).

Pada algoritma TF-IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus yaitu :

$$W_{dt} = tf_{dt} * Idf_t$$

Dimana:

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = Inversed Document Frequency

$$IDF = \log \left( \frac{N}{df} \right)$$

N = total dokumen

df = banyak dokumen yang mengandung kata yang dicari

*Term weighting* atau pembobotan term sangat dipengaruhi oleh hal-hal berikut ini (Mandala, 2004):

1. *Term Frequency (tf) factor*, yaitu faktor yang menentukan bobot term pada suatu dokumen berdasarkan jumlah kemunculannya dalam dokumen tersebut. Nilai jumlah kemunculan suatu kata (*term frequency*) diperhitungkan dalam pemberian bobot terhadap suatu kata. Semakin besar jumlah kemunculan suatu *term* (tf tinggi) dalam dokumen, semakin besar pula bobotnya dalam dokumen atau akan memberikan nilai kesesuaian yang semakin besar.
2. *Inverse Document Frequency (idf) factor*, yaitu pengurangan dominansi term yang sering muncul di berbagai dokumen. Hal ini diperlukan karena

term yang banyak muncul di berbagai dokumen, dapat dianggap sebagai term umum (*common term*) sehingga tidak penting nilainya. Sebaliknya faktor kejarangmunculan kata (*term scarcity*) dalam koleksi dokumen harus diperhatikan dalam pemberian bobot.

## 2.8 Cosine Similarity

Menurut (Dewa, Arie, dan Agustinus, 2016) *Cosine Similarity* adalah ukuran kesamaan antara dua buah vektor dalam sebuah ruang dimensi yang didapat dari nilai cosinus sudut dari perkalian dua buah vektor yang dibandingkan karena cosinus dari  $0^0$  adalah 1 dan kurang dari 1 untuk nilai sudut yang lain, maka nilai *similarity* dari dua buah vektor dikatakan mirip ketika nilai dari *cosine similarity* adalah 1.

Menurut (Dewa, Arie, dan Agustinus, 2016) *Cosine similarity* digunakan dalam ruang positif, dimana hasilnya dibatasi antara nilai 0 dan 1. Kalau nilainya 1 maka dokumen tersebut dikatakan mirip jika hasilnya 0 maka nilai tersebut dikatakan tidak mirip. Perhatikan bahwa batas ini berlaku untuk sejumlah dimensi, dan *Cosine similarity* ini paling sering digunakan dalam ruang positif dimensi tinggi. Misalnya, dalam *Information Retrieval*, masing-masing kata atau istilah (*term*) diasumsikan sebagai dimensi yang berbeda dan dokumen ditandai dengan *vector* dimana nilai masing-masing dimensi sesuai dengan berapa istilah muncul dalam dokumen. Berikut adalah rumus *cosine similarity*;

$$\text{Similarity} = \cos(\theta) = \frac{Q \cdot D}{|Q||D|} = \frac{\sum_{i=1}^n (wqi \times wdij)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wdij)^2}}$$

Keterangan :

$Q$  = Vektor  $Q$ , yang akan dibandingkan kemiripannya

$D$  = Vektor  $D$ , yang akan dibandingkan kemiripannya

$Q \cdot D$  = *dot product* antara vektor  $Q$  dan vektor  $D$

$|Q|$  = panjang vektor  $Q$

$|D|$  = panjang vektor  $D$

$|Q||D|$  = *cross product* antara  $|Q|$  dan  $|D|$

$wqi$  = bobot *term* pada *query* ke-  $i$ ,  $= tf \times idf$

$wd_{ij}$  = bobot *term* pada dokumen ke-i istilah ke-j =  $tf \times idf$

i = jumlah *term* dalam kalimat.

N = jumlah vektor.

Pengukuran kemiripan dapat dilakukan dengan membandingkan dokumen 1 dengan dokumen 2 kemudian sistem akan menghitung nilai kemiripan.  $wqi \times wd_{ij}$  adalah nilai yang diperoleh dari *term query* dan *term* dokumen kemudian kedua nilai tersebut dijumlahkan, sedangkan nilai  $wqi^2$  semua nilai *term query* nilainya dipangkatkan dua, begitu juga dengan *term*  $wd_{ij}^2$  semua nilai yang diperoleh dipangkatkan dua kemudian semua nilai yang diperoleh dijumlahkan lalu terakhir diakarkan sesuai rumus diatas. (Dewa, Arie, dan Agustinus , 2016)

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses *sorting* atau pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, demikian sebaliknya. (Agung Auliaguntary, 2016).

Pada penelitian yang dilakukan oleh Ogie Nurdiana, Jumadi, dan Dian Nursantika, 2016 dengan judul “Perbandingan Metode *Cosine Similarity* Dengan Metode *Jaccard Similarity* Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia” mendapatkan hasil bahwa nilai akurasi dari *cosine similarity* adalah nilai tertinggi dari metode *similarity* yang lain. Hal tersebut dikarenakan metode *cosine similarity* mempunyai konsep normalisasi panjang *vektor* data dengan membandingkan *N-gram* yang sejajar satu sama lain dari 2 pembanding. Sedangkan pada metode *jaccard* hanya membandingkan isi *N-gram* dengan *eksak* dan hanya melihat apakah ada suatu *N-gram* tertentu pada pembanding tanpa melihat posisi penulisan yang berbeda. Pada *euclidean distance* yang diterapkan di metode *k-nearest neighbor (K-NN)* tidak mempunyai konsep *normalisasi* panjang *vektor* data, sehingga nilai akurasi metode dipengaruhi oleh panjang 2 data pembanding dan harus menentukan nilai dari parameter K (jumlah dari tetangga terdekat). Maka metode *cosine similarity* menjadi usulan alternatif untuk mencari kemiripan dari teks *mining*.

Berdasarkan hal tersebut, penulis memilih menggunakan metode *cosine similarity* ini dikarenakan metode ini mempunyai nilai akurasi yang tinggi dimana menurut (Rozas dan Sarno, 2011) kelebihan utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen. Sehingga, dengan melakukan perbandingan *keyword* yang dihasilkan, maka kedekatan antara *item*-pun dapat dipastikan.

## 2.9 Confusion Matrix

*Confusion Matrix* adalah sebuah metode yang biasa digunakan untuk perhitungan akurasi dalam suatu Sistem Temu Kembali Informasi. Dalam pengujian keakuratan hasil pencarian akan dievaluasi nilai *recall*, *precision*, *accuracy*, dan *error rate*. Dimana *precision* mengevaluasi kemampuan sistem untuk menemukan peringkat yang paling relevan, dan didefinisikan sebagai persentase dokumen yang di *retrieve* dan benar-benar relevan terhadap *query*. *Recall* mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen dan didefinisikan sebagai persentase dokumen yang relevan terhadap *query*. *Accuracy* merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus dan *error rate* merupakan kasus yang diidentifikasi salah dengan jumlah seluruh kasus. (Subari Ferdinandus, 2015).

Rumus untuk *confusion matrix* dapat dilihat pada tabel dibawah ini:

Tabel. 2.3 *Confusion Matrix*

Document	Nilai Sebenarnya	
	Relevant	Non Relevant
Retrieved	True Positive (tp) <i>Correct result</i>	False Positive (fp) <i>Unexpected result</i>
	False Negative (fn) <i>Missing result</i>	True Negative (tn) <i>Correct absence of result</i>
Not Retrieved		

Keterangan:

TP (*True Positive*) = Jumlah prediksi yang benar dari data yang *relevant*.

FP (*False Positive*) = Jumlah prediksi yang salah dari data yang tidak *relevant*.

FN (*False Negative*) = Jumlah prediksi yang salah dari data yang tidak *relevant*.

TN (*True Negative*) = Jumlah prediksi yang benar dari data yang *relevant*.

Sehingga, rumusnya adalah sebagai berikut:

$$1. \text{ Precision} = \frac{tp}{(tp+fp)}$$

$$2. \text{ Recall} = \frac{tp}{(tp+fn)}$$

$$3. \text{ Accuracy} = \frac{(tp+tn)}{(tp+fp+tn+fn)}$$

$$4. \text{ Error rate} = \frac{(fn+tn)}{(tp+fp+tn+fn)}$$

*Recall* dinyatakan sebagai bagian dari dokumen relevan dalam dokumen yang ditemukan, *recall* merupakan jumlah dokumen yang seharusnya terambil oleh sistem berdasarkan perhitungan manual. *Precision* berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. *Accuracy* adalah dengan melakukan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus yang ada. *Error rate* adalah dengan melakukan perbandingan kasus yang diidentifikasi salah dengan jumlah seluruh kasus yang ada. Sistem yang dikatakan baik adalah sistem yang memiliki nilai *recall* dan *precision* tinggi.

## 2.10 Metode Pengumpulan Data

### 2.10.1 Pengertian Metode Pengumpulan Data

Menurut (Sugiyono, 2012) Pengumpulan data adalah Pengumpulan yang dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Tujuan yang diungkapkan dalam bentuk hipotesis merupakan jawaban sementara terhadap pertanyaan penelitian. Pengumpulan data dapat berupa suatu pernyataan (*statement*) tentang sifat, keadaan, kegiatan tertentu dan sejenisnya. Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian.

### 2.10.2 Pengertian Studi Pustaka

Menurut (Sugiyono, 2012), Studi Pustaka adalah kajian teoritis, referensi serta literatur ilmiah lainnya yang berkaitan dengan budaya, nilai dan norma yang berkembang pada situasi sosial yang teliti, selain itu studi

pustaka sangat penting dalam melakukan penelitian, hal ini dikarenakan penelitian tidak akan lepas dari literatur-literatur ilmiah.

### 2.10.3 Pengertian Wawancara

Menurut (Sugiono, 2012) Wawancara adalah merupakan pertemuan dua orang untuk bertukar informasi dan ide melalui tanya jawab, sehingga dapat dikonstruksikan makna dalam suatu topik tertentu. Esterberg, dalam (Sugiyono, 2012) mengemukakan beberapa macam wawancara yaitu *wawancara testruktur* (peneliti telah mengetahui dengan pasti informasi apa yang akan diperoleh sehingga peneliti menyiapkan instrumen penelitian berupa pertanyaan tertulis yang alternative jawabannya pun telah disiapkan), *wawancara semiterstruktur* (pelaksanaan wawancara lebih bebas, dan bertujuan untuk menemukan pemasalahan secara lebih terbuka dimana responden dimintai pendapat dan ide-idenya), dan *wawancara tidak terstruktur* (merupakan wawancara yang bebas dimana peneliti tidak menggunakan pedoman wawancara yang telah tersusun secara sistematis dan lengkap untuk pengumpulan datanya).

## 2.11 Metode Pengembangan Sistem

Metode pengembangan Sistem yang penulis gunakan adalah model *Rapid Application Development Life Cycle* dengan penjelasan berikut:

### 2.11.1 Rapid Application Development (RAD)

*Rapid Application Development (RAD)* adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat. (Kendall, 2010).



### 1. Fase Perencanaan Syarat (*Requierement Planning*)

Dalam fase ini pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.

### 2. Fase Desain (*RAD Design Workshop*)

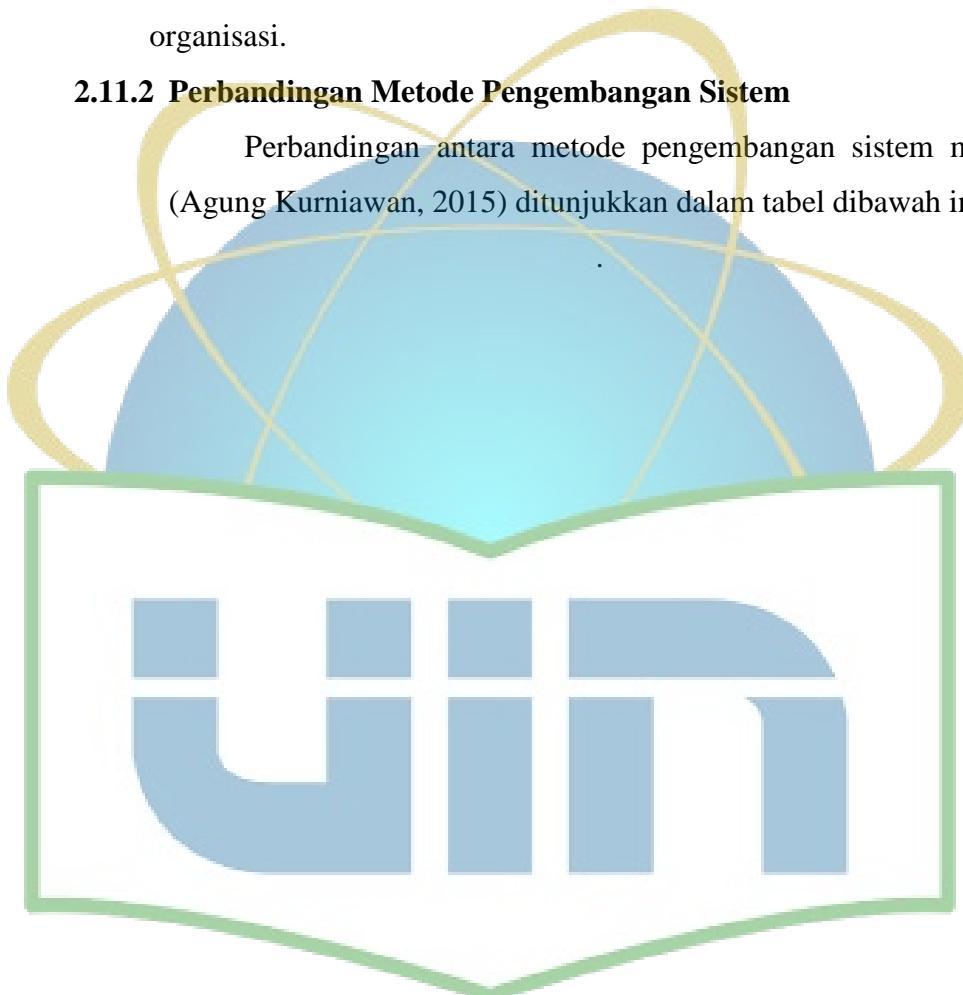
Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon *prototype* yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Apabila seorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi.

### 3. Fase Implementasi (*Implementation*)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi.

#### 2.11.2 Perbandingan Metode Pengembangan Sistem

Perbandingan antara metode pengembangan sistem menurut (Agung Kurniawan, 2015) ditunjukkan dalam tabel dibawah ini:



Tabel 2.4 Perbandingan Metode Pengembangan Sistem

No.	Metode	Sistem Kerja	Penekanan	Konsep	Pendekatan
1	Waterfall	Proyek dibagi menjadi beberapa fase yang berurutan, dengan beberapa tumpang tindih yang splashback yang diterima antar fase.	Penekanan pada perencanaan, jadwal waktu, tanggal target dan implementasi dari keseluruhan sistem pada satu waktu.	Control ketat dijaga selama umur proyek melalui dokumentasi tertulis yang luas, review formal dan persetujuan oleh pengguna dan managemen informasi teknologi yang terjadi pada akhir fase yang belum memulai tahap berikutnya.	Menerapkan pendekatan tradisional.
2	RAD	Metode RAD hampir sama dengan <i>waterfall</i> , perbedaannya siklus pengembangan yang ditempuh model ini sangat pendek dengan penerapan teknik yang cepat.	RAD menekankan pada siklus pembangunan pendek, singkat dan cepat.	RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem.	Proses pembangunan perangkat lunak yang tergolong dalam teknik <i>incremental</i> (bertingkat).
3	Prototyping	Upaya untuk mengurangi resiko proyek yang melekat dengan memecah proyek ke	Penekanan pada pengurangan waktu dan	Kecil <i>mockup</i> dari sistem yang dibentuk setelah melalui proses yang berulang-ulang sampai modifikasi	Pendekatan untuk menangani bagian-bagian yang dipilih

		dalam segmen yang lebih kecil dan menyediakan lebih banyak kemudahan perubahan selama proses pembangunan.	biaya, dan peningkatan keterlibatan pengguna.	<i>prototype</i> berkembang untuk memenuhi kebutuhan pengguna.	dari metodologi, pengembangan yang lebih besar yang lebih tradisional.
4	<i>Incremental</i>	Cara kerja sama seperti <i>waterfall</i> , dimana semua fase selesai untuk sebagian kecil dari suatu sistem, sebelum melanjutkan ke fase berikutnya.	Waktu kerja yang lebih cepat, dan pengurangan resiko kerja karena prosedur yang lebih sederhana.	Konsep perangkat lunak awal, analisis persyaratan, dan desain arsitektur di definisikan melalui <i>waterfall</i> .	Pendekatan metodologi pengembangan sistem.
5	<i>Spiral</i>	Menentukan tujuan alternatif dan kendala dari iterasi. Mengevaluasi dari alternatif, mengidentifikasi dan mengatasi resiko. Mengembangkan dan memverifikasi kiriman dari iterasi. Merencanakan iterasi berikutnya.	Penilaian resiko dan meminimal-kan resiko proyek.	Memulai setiap siklus dengan identifikasi pemangku kepentingan dan kondisi mereka, dan mengakhiri setiap siklus dengan <i>review</i> dan komitmen.	Pendekatan dari penggabungan keuntungan dari konsep <i>top-down</i> dan <i>bottom-up</i> .

Berdasarkan pemaparan diatas, penulis memilih menggunakan metode *Software Development Life Cycle* dengan model *Rapid Application Development* (RAD) karena model ini adalah model yang sesuai berdasarkan system kerja dan penekanan yang telah dijelaskan diatas, bahwa model pengembangan yang ditempuh model ini sangat pendek dengan penerapan teknik yang singkat dan cepat. Oleh karena itu, penggunaan RAD sesuai dengan perancangan sistem dalam penelitian ini yaitu penerapan metode *term frequency inverse document frequency* dan *cosine similarity* dalam sistem temu kembali informasi untuk mengetahui *syarah hadits* dengan basis web dikarenakan penelitian ini merupakan penelitian yang tidak memerlukan waktu yang lama karena sistem yang akan dibangun merupakan sistem yang tidak terlalu kompleks.

## 2.12 *Unified Modelling Language (UML)*

### 2.12.1 Pengertian *Unified Modelling Language (UML)*

Menurut Yuni Sugiarti (2013:34) UML (*Unified Modeling Language*) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

### 2.12.2 Jenis-Jenis Diagram UML

Jenis-jenis diagram *Unified Modelling Language* (UML) adalah sebagai berikut:

#### 1. *Use Case Diagram*

Menurut (Yuni Sugiarti, 2013) *Use Case Diagram* atau diagram *use case* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk

mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol *use case* dapat dilihat dapat dilihat pada tabel dibawah ini:

Tabel. 2.5 Komponen Pembentuk *Use Case Diagram*

Simbol	Deskripsi
	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>Use Case</i> .
	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda dari awal nama aktor.
	Komunikasi antara aktor dan Use Case yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan aktor.

<p><i>Extend</i></p>	<p>Relasi <i>Use Case</i> tambahan kesebuah <i>Use Case</i> Dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya <i>Use Case</i> tambahan memiliki nama depan yang sama dengan <i>Use Case</i> yang ditambahkan, arah panah menunjuk pada <i>use case</i> yang dituju, contoh:</p>
<p><i>Include</i></p>	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> di mana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>, <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, contoh :</p>

## 2. *Activity Diagram*

Menurut (Yuni Sugiarti, 2013) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

## 3. *Sequence Diagram*

Menurut (Yuni Sugiarti, 2013) diagram sekuen menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu.

## 4. *Class Diagram*

Menurut (Yuni Sugiarti, 2013) Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisan kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi, dimana:

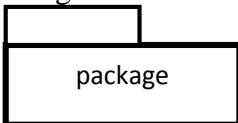
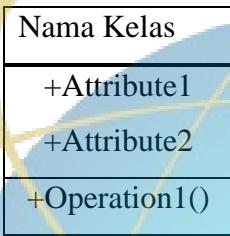
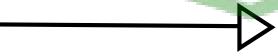
- Atribut merupakan variabel variabel yang dimiliki oleh suatu kelas.
- Atribut mendeskripsikan properti dengan sebaris teks didalam kotak kelas tersebut.
- Operasi atau metode adalah fungsi fungsi yang dimiliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan dengan statis yang terdapat di antara mereka. Diagram kelas juga menunjukan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Diagram kelas menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain lain seperti containment, pewarisan, asosiasi dan lain- lain.

Simbol-simbol *class diagram* menurut (Yuni Sugiarti, 2013) dapat dilihat pada tabel dibawah ini:

Tabel 2.6 Simbol *Class Diagram*

Simbol	Deskripsi
Package	 <i>Package</i> merupakan sebuah bungkus dari satu atau lebih kelas.
Operasi	 Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi	 Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi Berarah/ <i>Directed Assosiation</i>	 Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi	 Relasi antara kelas dengan makna generalisasi-generalisasi (umum khusus).
Kebergantungan/ <i>dependency</i>	 Relasi antar kelas dengan makna kebergantungan antar kelas

<b>Agregasi/Agregation</b> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )
---	--

## 2.13 PHP

Menurut (Nugroho, 2013) PHP kepanjangan dari *Hypertext Preprocessor* itu bahasa pemrograman berbasis web. Jadi, PHP itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis web (*website*, blog, atau aplikasi *web*). Sedangkan menurut Budi Rahardjo, Imam Heryanto dan E.Rosdiana K, PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di *parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai bahasa *sisi server* (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *user* memilih perintah “View Source” pada *web browser* yang mereka gunakan.

## 2.14 Flowchart Diagram

Menurut (Indrajani, 2015), *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. (Indrajani, 2015), menjelaskan simbol-simbol dalam Flow Chart adalah sebagai berikut:

Tabel 2.7 Simbol-Simbol *Flowchart Diagram*

<b>Simbol</b>	<b>Keterangan</b>
	<b>Penghubung</b> Simbol untuk keluar/masuk atau proses dalam lembar atau halaman lain.
	<b>Input Output</b> Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.

	<b>Dokumen</b> Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau output dicetak di kertas.
	<b>On Line Storage</b> Simbol yang menunjukkan bahwa data di dalam simbol ini akan disimpan.
	<b>Simbol Garis Alir</b> Digunakan untuk menunjukkan arah selanjutnya yang akan dituju dari simbol-simbol dalam <i>flowchart</i> .
	<b>Simbol Manual</b> Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<b>Terminal</b> Simbol yang menunjukkan untuk permulaan atau akhir suatu sistem.
	<b>Kondisi</b> Simbol keputusan yang menunjukkan kondisi.
	<b>Proses</b> Simbol yang menunjukkan pengolahan dilakukan oleh komputer.
	<b>Penghubung</b> Simbol untuk keluar/masuk prosedur atau proses dalam lembar atau halaman yang masih sama.

## 2.15 MySQL

Menurut (Nugroho, 2013) MySQL adalah *software* atau program *Database Server*. Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam *software database server* lain, seperti SQL Server, Oracle, PostgreSQL dan lainnya.

Adapun perintah yang terdapat pada MySQL menurut (Nugroho, 2013), adalah sebagai berikut :

Tabel. 2.8 Perintah Dasar MySQL

No.	Perintah	Keterangan
1	<i>show databases</i>	Perintah ini digunakan untuk menampilkan atau melihat daftar database yang sudah ada (sudah dibuat);

2	<i>Use</i>	Perintah ini digunakan untuk masuk atau keluar dari database yang sudah ada;
3	<i>Show tables</i>	Perintah ini digunakan untuk melihat atau menampilkan semua tabel yang ada di dalam database aktif (yang sudah dibuka, sudah di <i>use</i> );
4	<i>Desc/describe</i>	Perintah ini digunakan untuk melihat struktur tabel;
5	<i>Quit</i>	Perintah ini digunakan untuk keluar <i>MySQL Server</i>

## 2.16 Xampp

Menurut (Nugroho, 2013) XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web, khususnya PHP dan MySQL.

Dalam folder utama xampp, terdapat beberapa folder penting yang perlu diketahui. Untuk lebih memahami setiap fungsinya, penjelasannya terdapat dalam tabel dibawah ini (Nugroho, 2013):

Tabel 2.9 Folder Penting Xampp

No.	Folder	Keterangan
1	<i>Apache</i>	Folder utama dari Apache Webserver.
2	<i>Htdocs</i>	Folder utama untuk menyimpan data-data latihan web, baik PHP maupun HTML biasa.
3	<i>Manual</i>	Berisi subfolder yang di dalamnya terdapat manual program dan database, termasuk manual PHP dan MySQL.
4	<i>MySQL</i>	Folder utama untuk database MySQL Server.
5	<i>PHP</i>	Folder utama untuk program PHP.

## 2.17 Database

### 2.17.1 Pengertian Database

Menurut (Priyanto dan Jauhari, 2014) *database* dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang

diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Prinsip utamanya adalah pengaturan data. Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data.

### 2.17.2 Tujuan Database

Menurut (Priyanto dan Jauhari, 2014) secara lebih lengkap pemanfaatan basis data dilakukan untuk memenuhi tujuan berikut ini:

1. Kecepatan dan Kemudahan (*Speed*)
2. Efisiensi ruang penyimpanan (*Space*)
3. Keakuratan (*Accuracy*)
4. Ketersediaan (*Availability*)
5. Kelengkapan (*Completeness*)
6. Keamanan (*Security*)
7. Pemakaian bersama (*Sharability*)

### 2.18 Alasan Membuat Sistem Berbasis Website

Berikut ini merupakan alasan mengapa penulis lebih memilih untuk membuat sistem berbasis website dibanding dengan aplikasi lainnya, yaitu dapat dilihat dari beberapa aspek berikut ini (aplikasipemda.com, 2017):

#### 1. Instalasi

Aplikasi berbasis web tidak memerlukan proses instalasi *software* pada komputer *client*, instalasi cukup di *server* saja. Berbeda dengan aplikasi desktop maupun aplikasi android yang harus di-*install* pada masing-masing komputer client yang ingin menggunakannya.

#### 2. Maintenance

Aplikasi berbasis web bersifat terpusat, jadi setiap perubahan kode program dilakukan di komputer *server*. Sedangkan aplikasi desktop bersifat terdistribusi, jadi setiap perubahan kode program harus didistribusikan juga di semua komputer client satu persatu.

#### 3. Platform

Aplikasi berbasis web dapat dijalankan di sistem operasi mana pun, tidak perduli apakah menggunakan Linux, Windows atau OSX. Aplikasi

berbasis web dapat dijalankan asalkan memiliki browser dan akses Internet. Berbeda dengan aplikasi desktop dan aplikasi android yang mengharuskan aplikasi di-install dan dijalankan di sistem operasi tertentu.

#### 4. *Performance*

Tidak perlu spesifikasi komputer client yang tinggi untuk menggunakan aplikasi berbasis web ini, sebab di beberapa kasus, sebagian besar proses dilakukan di komputer server penyedia aplikasi berbasis web ini. Berbeda dengan aplikasi desktop yang prosesnya dikerjakan di komputer client sendiri dan membutuhkan spesifikasi komputer client tertentu.

#### 5. *Availability*

Aplikasi berbasis web dapat digunakan hanya dengan syarat ketersediaan *browser* dan akses internet sehingga membuatnya bersifat *multi-platform* dapat diakses dari sistem operasi mana pun dan dari perangkat mana pun. Berbeda dengan aplikasi desktop yang syarat ketersedianya adalah terinstallnya aplikasi tersebut di masing komputer client. Begitu juga dengan aplikasi android yang tidak dapat langsung digunakan dalam perangkat ataupun operasi tertentu sebelum di *install* terlebih dahulu.

Berdasarkan alasan yang telah dijelaskan diatas, maka penulis memilih untuk membangun sistem berbasis *website* dalam penelitian ini.

### 2.19 Studi Literatur Sejenis

Studi literatur yang digunakan dalam penyusunan penelitian ini adalah sebagai berikut:

Tabel 2.10 Studi Literatur Sejenis

No.	Judul Penelitian	Nama Peneliti	Deskripsi Penelitian	Hasil Penelitian
1	“Penerapan Algoritma <i>Cosine Similarity</i> dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi”. Universitas Negeri Semarang. (2017)	Rizki Tri Wahyuni, Dhidik Prastyianto, dan Eko Supraptono.	Penelitian mengenai sistem yang dapat mengklasifikasikan dokumen secara otomatis ke dalam folder berbeda pada database agar lebih mudah dalam mengelola dokumen yang ada dengan Metode TF-IDF dan <i>cosine similarity</i> .	<ol style="list-style-type: none"> <li>1. Sistem dapat mengklasifikasikan dokumen ke dalam kategori yang sesuai dengan tingkat akurasi yang tinggi.</li> <li>2. Sistem dapat berjalan dengan baik sesuai dengan tujuan penelitian.</li> <li>3. Tidak menggunakan <i>stemming</i> dalam teks <i>preprocessing</i>.</li> <li>4. Hanya melakukan proses <i>tokenizing</i> dan <i>stopword removal</i> dalam teks <i>preprocessing</i>.</li> </ol>
2	“Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode <i>Cosine Similarity”</i> Universitas Sam Ratulangi Manado (2016).	Dewa Ayu, Arie Lumenta, dan Agustinus Jacobus.	Penelitian ini mengukur kemiripan dokumen untuk mengetahui adanya plagiarisme terhadap karya tulis dengan menggunakan metode <i>cosine similarity</i> .	<ol style="list-style-type: none"> <li>1. Penggunaan data <i>dummy</i> pada sistem sudah berhasil diterapkan dan sudah bisa digunakan untuk mengukur tingkat kemiripan dokumen.</li> <li>2. Sistem berhasil mengukur tingkat kemiripan dokumen.</li> </ol>

				<ul style="list-style-type: none"> <li>3. Pembobotan kata dengan <i>term frequency</i> saja.</li> <li>4. Tidak menggunakan <i>stemming</i>.</li> <li>5. Hanya melakukan proses <i>tokenizing</i> dalam teks <i>preprocessing</i>.</li> </ul>
3	<p>“Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode <i>Term Frequency Inverse Document Frequency</i> (Tf-Idf)”. Universitas Diponegoro (2012)</p>	Dhony Syafe’i, Sukmawati, dan Nurdin Bahtiar.	<p>Penelitian ini membuat suatu perangkat lunak yang dapat mencari dokumen-dokumen penulisan ilmiah yang relevan sesuai tingkat pembobotannya.</p>	<ul style="list-style-type: none"> <li>1. sistem dapat me-<i>retrieve</i> dokumen sesuai dengan <i>query</i> pengguna dengan metode tf-idf.</li> <li>2. Melakukan proses <i>tokenizing</i>, <i>stopword removal</i>, dan <i>stemming</i>.</li> <li>3. Menggunakan <i>stemming</i> jenis <i>porter stemmer</i> Fadhillah. Z. Tala.</li> <li>4. Penelitian dengan metode ini tidak selalu memberikan hasil performansi yang baik pada koleksi pengujian, dikarenakan tidak semua dokumen ter-<i>retrieve</i> merupakan dokumen yang relevan.</li> </ul>

Berdasarkan analisis penulis, dengan membandingkan 3 studi literatur diatas, maka peneliti membuat sistem temu kembali informasi dengan fitur berikut:

1. Sistem ini menggunakan metode *term frequency inverse document frequency* sebagai pembobotan kata (*term*) di setiap dokumen untuk dilakukan proses selanjutnya.
2. Sistem ini menggunakan metode *cosine similarity* untuk melakukan *document matching*.
3. Sistem dibangun dengan menggunakan *stemming* jenis Nazief Adriani yang memiliki tingkat akurasi (presisi) lebih tinggi dari jenis *stemming* yang lain.
4. Sistem dapat menampilkan *syarah hadits* dengan point-point penting berdasarkan *query* yang telah diinputkan.
5. Sistem dibangun dengan berbasis web menggunakan bahasa pemrograman *PHP* serta media penyimpanan data yaitu dengan *MySql*.



## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Metode Pengumpulan Data**

Dalam melaksanakan penelitian ini diperlukan data dan informasi terkait yang nantinya akan digunakan sebagai bahan rujukan untuk pembuatan penelitian serta mendukung keabsahan pembahasan pada laporan penelitian. Metode pengumpulan data yang digunakan yaitu:

##### **3.1.1 Metode Studi Pustaka.**

Dalam metode ini penulis melakukan studi pustaka dengan pencarian data secara manual yaitu mempelajari buku-buku yang berhubungan dengan konsep, analisis, dan perancangan sistem. Selain itu penulis juga melakukan pencarian data secara online melalui *browsing* internet dengan mengunjungi *website* yang berhubungan dengan topik ataupun konten yang membantu dalam penelitian ini dan melakukan pembelajaran terhadap jurnal dan skripsi yang relevan sesuai dengan topik penelitian yang dijadikan sebagai bahan acuan dalam pembuatan penelitian. Selain itu, dilakukan juga pengumpulan bahan materi untuk melakukan penelitian seperti materi yang berhubungan dengan *tf-idf* dan *cosine similarity* maupun *hadits* yang terkait.

Data yang didapatkan digunakan dalam penyusunan landasan teori, metodologi penelitian dan pengembangan sistem yang referensinya dapat dilihat selengkapnya dalam daftar pustaka dengan rincian 17 buku, 8 skripsi, 11 jurnal, dan 3 website.

##### **3.1.2 Metode Wawancara**

Dalam metode wawancara ini penulis melakukan wawancara dengan Bapak Rifqi Muhammad Fatkhil, MA selaku Kepala Jurusan Program Studi Hadits Fakultas Ushuludin Universitas Islam Negeri Syarif Hidayatullah Jakarta, dimana beliau merupakan narasumber yang tidak diragukan lagi dalam menguasai hal yang berhubungan dengan permasalahan (*syarah hadits*) sehingga penulis memperoleh data-data dan informasi yang dibutuhkan sesuai dengan masalah yang diteliti. Wawancara dilakukan 2 kali

yaitu untuk mendukung permasalahan yang terkait, dan untuk pengujian sistem yang telah dibuat.

Sistem wawancara dilakukan melalui tatap muka. Hasil wawancara terlampir pada halaman lampiran.

### **3.2 Metode Pengembangan Sistem**

Dalam metode pengembangan system ini penulis menggunakan model *Rapid Application Development* (RAD) dengan alasan bahwa model ini sangat cocok digunakan untuk penggerjaan dengan waktu yang singkat dan tidak memerlukan sumber daya yang cukup besar, sehingga dianggap dapat dengan cepat menyelesaikan sistem ini. Metode ini dibagi menjadi 3 fase yaitu sebagai berikut:

#### **3.2.1 Fase Perencanaan Syarat (*Requirement Planning*)**

Dalam fase ini penulis melakukan syarat-syarat perencanaan dengan langkah sebagai berikut:

##### **3.2.1.1 Mengidentifikasi Masalah**

Pada tahap ini yaitu mengidentifikasi masalah yang menjadi alasan utama dibuatnya sistem.

##### **3.2.1.2 Menganalisis Solusi Permasalahan**

Setelah proses identifikasi masalah selesai, tahap selanjutnya yaitu menentukan solusi yang tepat dari masalah yang telah dianalisa.

##### **3.2.1.3 Menganalisis kebutuhan sistem**

Tahap berikutnya yaitu menganalisis terhadap kebutuhan sistem yaitu dengan menjelaskan data serta kelengkapan apa saja yang dibutuhkan dalam sistem yang akan dibangun agar sesuai dengan tujuan pembuatan sistem yang ingin dicapai.

#### **3.2.2 Fase Workshop Desain (*Design Workshop*)**

Dalam fase ini penulis melakukan perancangan system, desain sistem dan pengkodean sistem dengan penjelasan sebagai berikut:

##### **3.2.2.1 Merancang Sistem**

Pada tahap ini penulis akan menjelaskan proses apa saja yang berjalan dalam sistem.

### 3.2.2.2 Merancang UML

Pada perancangan pemodelan object ini penulis akan menjelaskan tentang proses-proses apa saja yang dibutuhkan oleh sistem yang akan dikembangkan dengan menggunakan diagram UML seperti *Use Case Diagram, Activity Diagram, Sequence Diagram* dan *Class Diagram*.

### 3.2.2.3 Merancang Database

Pada tahap ini penulis merancang *database* yang berisi desain tabel-tabel yang dibutuhkan dalam pengolahan data yang nantinya akan diimplementasikan kedalam bentuk *database*, dengan menggunakan *MySQL*.

### 3.2.2.4 Merancang user interface

Pada tahap ini penulis merancang *user interface* (tampilan sistem) yang akan diperlihatkan sebagai gambaran mengenai sistem yang akan dibuat dan diharapkan akan dapat memudahkan dalam melihat gambaran sistem ini.

### 3.2.2.5 Pengkodean (coding)

Pada tahap ini akan dilakukan *Coding* program terhadap hasil rancangan yang sudah dibuat.

## 3.2.3 Fase Implementasi (*Implementation*)

Fase ini merupakan fase terakhir setelah melalui fase-fase sebelumnya. Dimana dalam fase ini, hasil dari perancangan yang telah dilakukan akan diimplementasikan ke dalam sistem dengan penjelasan sebagai berikut:

### 3.2.3.1 Implementasi sistem

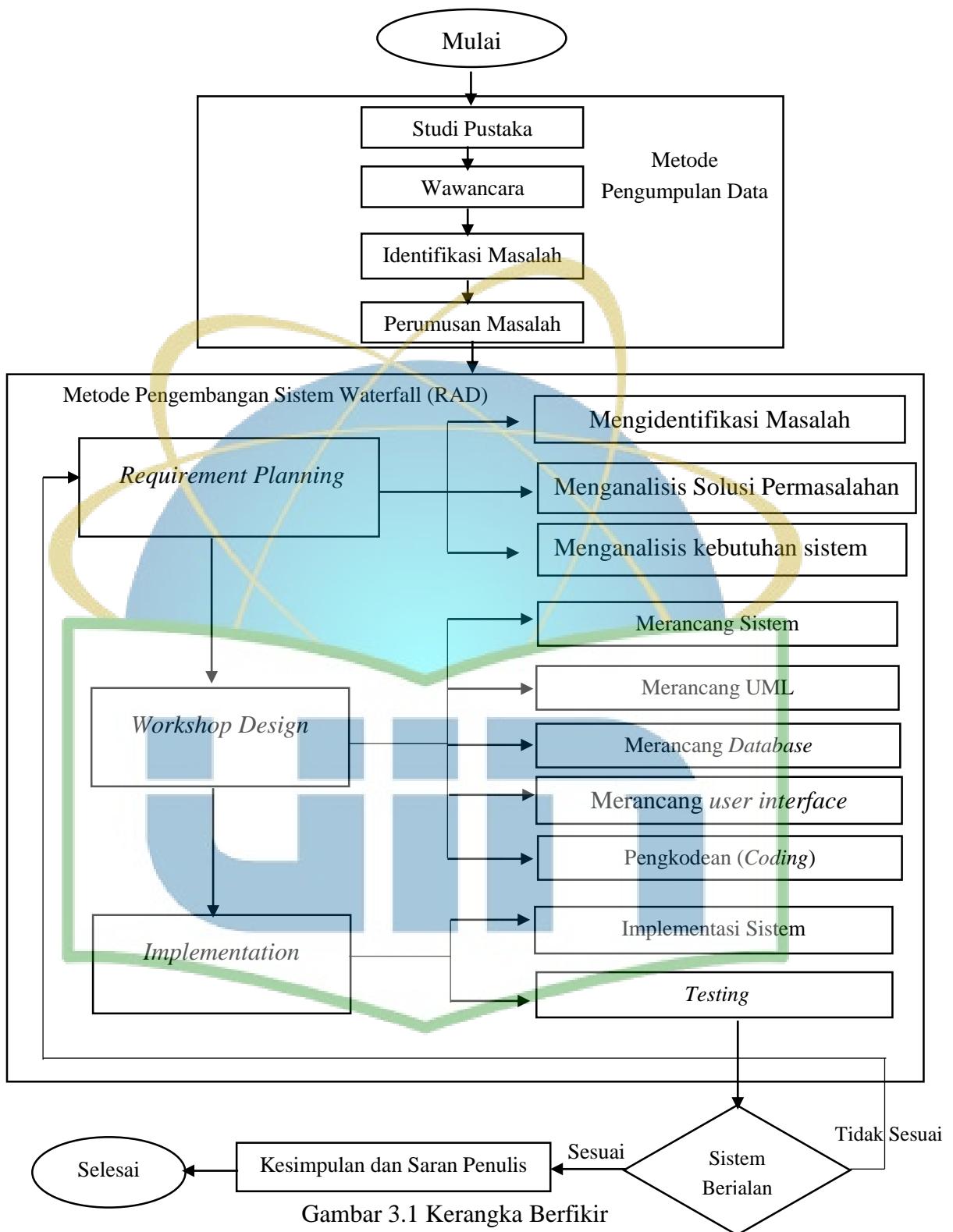
Pada tahap ini, sistem telah siap untuk dioperasikan dengan hasil apakah sistem sesuai dengan *user interface* yang telah dirancang sebelumnya atau tidak.

### 3.2.3.2 Testing

Setelah tahap implementasi sistem selesai, selanjutnya penulis melakukan pengujian sistem yang telah dibangun dengan menggunakan pengujian *confusion matrix* untuk menguji seberapa akurat sistem yang telah dibuat dengan metode yang digunakan.

## 3.3 Kerangka Berfikir

Kerangka berfikir dalam penelitian ini adalah sebagai berikut:



Gambar 3.1 Kerangka Berfikir

## **BAB IV**

### **ANALISA DAN PERANCANGAN SISTEM**

Pada analisa dan perancangan sistem ini, berisi perincian tahapan yang akan dibangun dengan menggunakan metode pengembangan sistem *Rapid Application Development* (RAD) yang meliputi Fase Perencanaan Syarat, dan Fase Workshop Desain.

#### **4.1 Fase Perencanaan Syarat**

Dalam fase ini penulis melakukan syarat-syarat perencanaan sebagai berikut:

##### **4.1.1 Mengidentifikasi Masalah**

Berdasarkan studi pustaka dan wawancara yang telah penulis lakukan, maka didapatkan permasalahan utama dalam penelitian ini yaitu sulitnya mengetahui makna *hadits* yang sebenar-benarnya jika dilihat hanya dari teks lahiriyah saja. *Hadits* yang merupakan sumber pedoman hidup umat muslim setelah *Al-Qur'an*, dewasa ini tidak sedikit orang yang keliru dalam memahaminya, hal itu dikarenakan oleh 3 hal, yaitu : 1. Mereka menganut pemahaman literalis (tekstualis), 2. Semangat keberagaman yang ingin kembali mengamalkan ajaran agama yang otentik sesuai dengan *Al-Qur'an* dan *hadits* tanpa didukung oleh literasi terhadap pendapat-pendapat ulama, dan 3. Keterbatasan kemampuan yang mereka miliki sehingga mereka lebih memilih atau hanya mengakses pada buku-buku terjemah sehingga tidak mewakili konten *hadits* dari apa yang disampaikan Nabi.”. Padahal untuk mengetahui makna yang sebenarnya dari suatu *hadits* sangat tidak cukup jika hanya dengan mengandalkan teks lahiriyah *hadits* saja kecuali *hadits-hadits* yang berkaitan dengan akidah.

Seperti penjelasan yang dikemukakan oleh Bapak Fiqri Muhammad Fatkhi, bahwa hal yang dapat kita tempuh untuk mengetahui makna sebenarnya dari *hadits* adalah salah satunya dengan mempelajari *syarah hadits*, dimana mempelajari ilmu *syarah hadits* untuk mengetahui makna yang terkandung didalamnya adalah sangat penting untuk dilakukan

sepanjang diperoleh dari orang yang memiliki otoritas dibidang penjelasan *hadits*, bukan hanya dari terjemahnya saja tanpa penjelasan maknanya.

Merujuk pada penjelasan beliau, sejauh ini aplikasi *syarah hadits* yang dapat digunakan oleh orang awam masih terbatas. Karena sejauh ini aplikasi yang ada masih berbahasa *full arab* yang tidak semua orang dapat memahaminya. Sedangkan untuk bahasa indonesia baru ada lidwa, dan arbain dimana isinya masih umum dari kumpulan *hadits* berbagai perawi.

#### 4.1.2 Menganalisis Solusi Permasalahan

Berdasarkan permasalahan yang peneliti kemukakan di atas, maka dibutuhkan solusi permasalahan untuk mengatasinya. Solusi tersebut yaitu dengan membangun suatu sistem yang mampu memberikan penjelasan mengenai makna *hadits* atau yang disebut dengan *syarah hadits* berbahasa indonesia berdasarkan *hadits* yang telah di *input*-kan oleh *user* sehingga *user* dapat mengetahui makna sebenarnya dari *hadits* yang ingin diketahui dengan akurat berdasarkan sumber yang valid guna meminimalisir kesalahan penafsiran terhadap suatu *hadits* agar kemudahan menjalankan ajaran Islam ikut terbantu sehingga memudahkan hidup berdasarkan petunjuk Rasulullah SAW guna kesuksesan kehidupan di dunia dan akhirat dengan berpedoman pada kitab *Taysirul 'Allam Syarah 'Umdatil Ahkam* yang akan dibuat dengan menerapkan metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* dengan harapan sistem yang akan dibuat dapat memberikan *output* berupa *syarah hadits* yang sesuai berdasarkan *input* yang dimasukkan dengan metode yang digunakan.

Sistem temu kembali informasi ini akan dibuat dengan menggunakan aplikasi berbasis web, dimana aplikasi dengan basis web ini memiliki beberapa kelebihan yaitu tidak memerlukan proses instalasi, bersifat terpusat, dapat dijalankan di os manapun asalkan memiliki *browser* dan akses internet, dan tidak perlu spesifikasi komputer *client* yang tinggi sehingga dapat diakses oleh siapapun dan dimanapun yang akan memudahkan user dalam menggunakan sistem ini.

### 4.1.3 Menganalisis Kebutuhan Sistem

Setelah mengetahui permasalahan dan solusinya, selanjutnya adalah mengidentifikasi kebutuhan sistem yang akan dibangun. Kebutuhan sistem ini dibagi menjadi 2, yaitu data *hadits shahih* imam Bukhari-Muslim dan alat bantu dalam mengembangkan sistem.

Data *hadits* yang peneliti gunakan terdiri dari 204 hadits shahih imam Bukhari-Muslim dari total 64 sub-bab yang akan digunakan untuk pelatihan dan uji coba. Data *hadits* tersebut peneliti dapatkan dari kitab *Taysirul 'Allam Syarah 'Umdatil Ahkam* yang berisi kumpulan-kumpulan *hadits* imam Bukhari-Muslim yang tidak diragukan lagi ke-shahihannya.

Peneliti menggunakan alat bantu berupa *software* dan *hardware* dalam mengembangkan sistem ini dengan rincian kelengkapan sebagai berikut:

#### 4.1.3.1 Kebutuhan Software

*Software* yang peneliti gunakan dalam membangun sistem ini adalah sebagai berikut:

- |               |                  |
|---------------|------------------|
| 1. Xampp      | : Xampp v.3.2.2  |
| 2. PHP        | : v.5.6.33       |
| 3. PhpMyAdmin | : v.4.7.4        |
| 4. Sublime    | : Sublime Text 3 |
| 5. Photoshop  | : cc 2017        |

#### 4.1.3.2 Kebutuhan Hardware

Spesifikasi *hardware* yang peneliti gunakan dalam membangun sistem ini adalah sebagai berikut:

1. Sistem Operasi : Windows 8.1 Pro 32-bit
2. Prosesor : 2.53GHz Intel Core I3
3. RAM : 3 Gigabyte
4. Harddisk : 300GB

## 4.2 Fase Workshop Desain

Dalam fase ini penulis melakukan perancangan sistem, desain sistem dan pengkodean sistem dengan penjelasan sebagai berikut:

### 4.2.1 Merancang Sistem

Dalam perancangan sistem ini, proses yang ada adalah teks *preprocessing* terhadap dokumen dan *query* dan menerapkan metode yang digunakan yaitu dengan metode *tf-idf* untuk pembobotan kata dan *cosine similarity* untuk menemukan dokumen dengan menghitung kedekatan antar dokumen. Penjelasannya sebagai berikut:

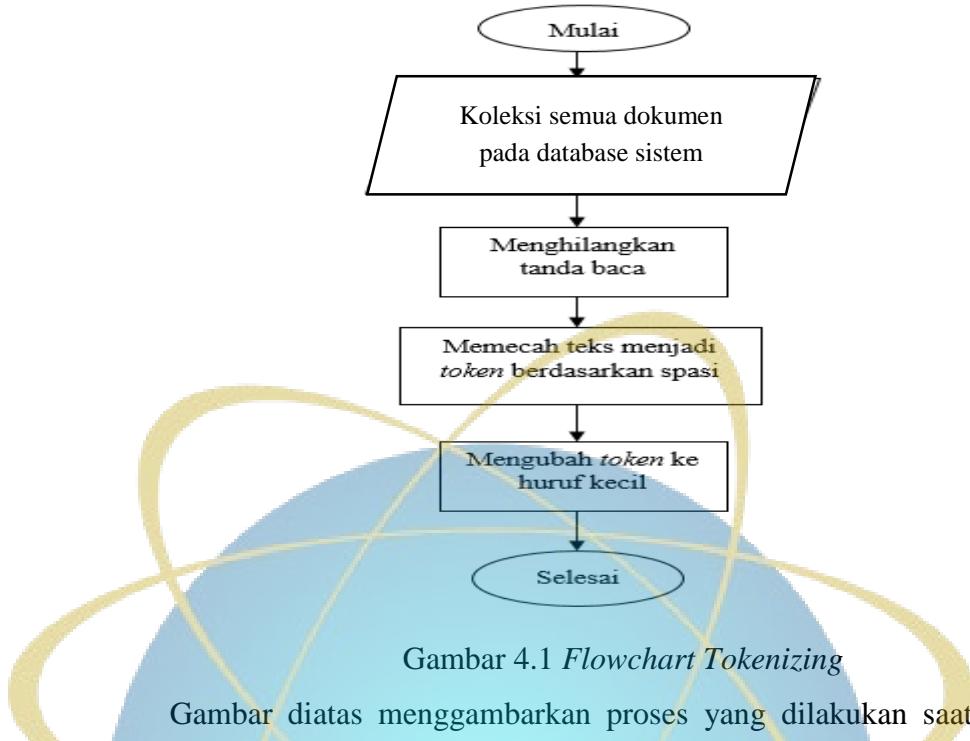
#### 4.2.1.1 Tahapan Teks *Preprocessing*

Perlu diketahui bahwa sebelum melakukan tahap teks *preprocessing*, yang harus dilakukan adalah menyimpan semua dokumen yang akan dicari dalam sebuah koleksi dokumen. Dimana dokumen ini merupakan koleksi *hadits* bahasa indonesia yang akan digunakan sebagai *query* beserta *syarahnya* (sebagai dokumen relevan) yang akan diproses dan disimpan dalam *database MySQL*. Selanjutnya tahap teks *preprocessing* dokumen, akan dijelaskan prosesnya sebagai berikut:

##### 1. *Tokenizing*

*Tokenizing* adalah proses memecah dokumen menjadi kumpulan kata. *Tokenization* dapat dilakukan dengan menghilangkan tanda baca dan memisahkannya per-spasi. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).

Proses yang berjalan saat tahap *tokenizing*, akan digambarkan dalam *flowchart* berikut:



Gambar 4.1 Flowchart Tokenizing

Gambar diatas menggambarkan proses yang dilakukan saat tahap *tokenizing*, dimana koleksi dokumen (*query* dan dokumen relevan) yang ada dalam *database* dilakukan penghilangan tanda baca, pemecahan teks menjadi *token* dengan delimeter spasi, dan pengubahan *token* yang terbentuk menjadi huruf kecil. Proses ini nantinya akan dilanjutkan dengan tahap berikutnya untuk dilakukan pegindeksan dokumen yang akan disimpan dalam *database* sistem.

Penerapan *tokenizing* penulis berikan pada salah satu contoh *hadits* dalam bab thaharah, dimana *hadits* yang diproses sebagai *query* harus memenuhi syarat untuk dapat diproses sistem, yaitu menggunakan *hadits* dengan struktur yang lengkap (terdapat *sanad* dan *matan*) karena tanpa salah satu nya maka *hadits* tidak dapat diproses (untuk membedakan antara aplikasi *hadits* dengan aplikasi pengolahan teks yang lain). Hasil dari penerapan proses *tokenizing* adalah sebagai berikut:

Tabel 4.1 Proses *Tokenizing*

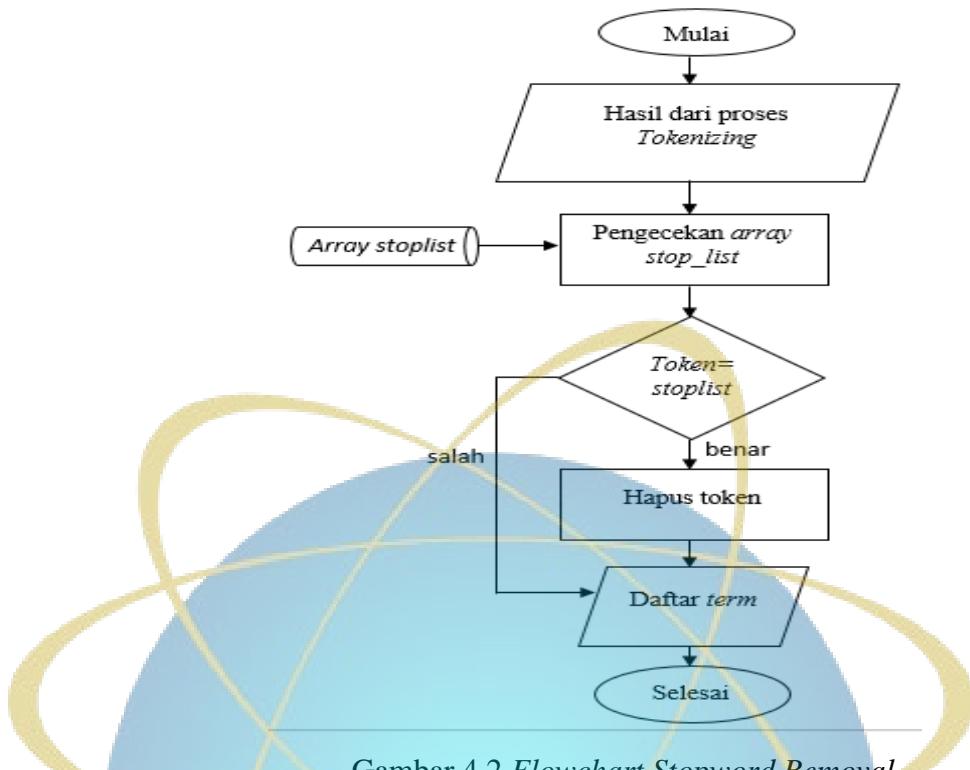
Sebelum <i>Tokenizing</i>	Sesudah <i>Tokenizing</i>
Dari Abu Hurairah ia berkata, “Rasulullah SAW bersabda:	dari abu



## 2. Stopword Removal/ Filtering

*Stopwords removal* merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil *parsing* deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak. Daftar kata tidak penting (*stoplist*) nya seperti : ("yang", "juga", "tidak", "diantara", "dari", "dia", "kami", "kamu", "ini", "itu", "jika", "apabila", "berkata", "atau", "hingga", "dan", "tersebut", "pada", "dengan", "adalah", "sebagai", "sebuah", "setiap", "untuk", "ketika", "yaitu", dan "ia").

Proses yang berjalan saat tahap *stopword removal* atau *filtering*, akan digambarkan dalam *flowchart* berikut:



Gambar 4.2 Flowchart Stopword Removal

Gambar diatas menggambarkan proses yang dilakukan saat tahap *stopword removal* atau *filtering*, dimana hasil dari proses *tokenizing* yang dilakukan sebelumnya, akan dicocokan dengan *array stoplist* yang ada, apabila *token* yang dicek merupakan *stoplist* maka token akan dihapus, apabila *token* bukan termasuk *stoplist* maka token akan dibiarkan tetap ada. Hasil dari proses *stopword removal* (*term*) ini yang berisi daftar *term-term* yang terbentuk, akan diproses pada tahap selanjutnya.

Contoh penerapan *stopword removal* atau *filtering* adalah sebagai berikut:

Tabel. 4.2 Hasil Stopword Removal atau Filtering

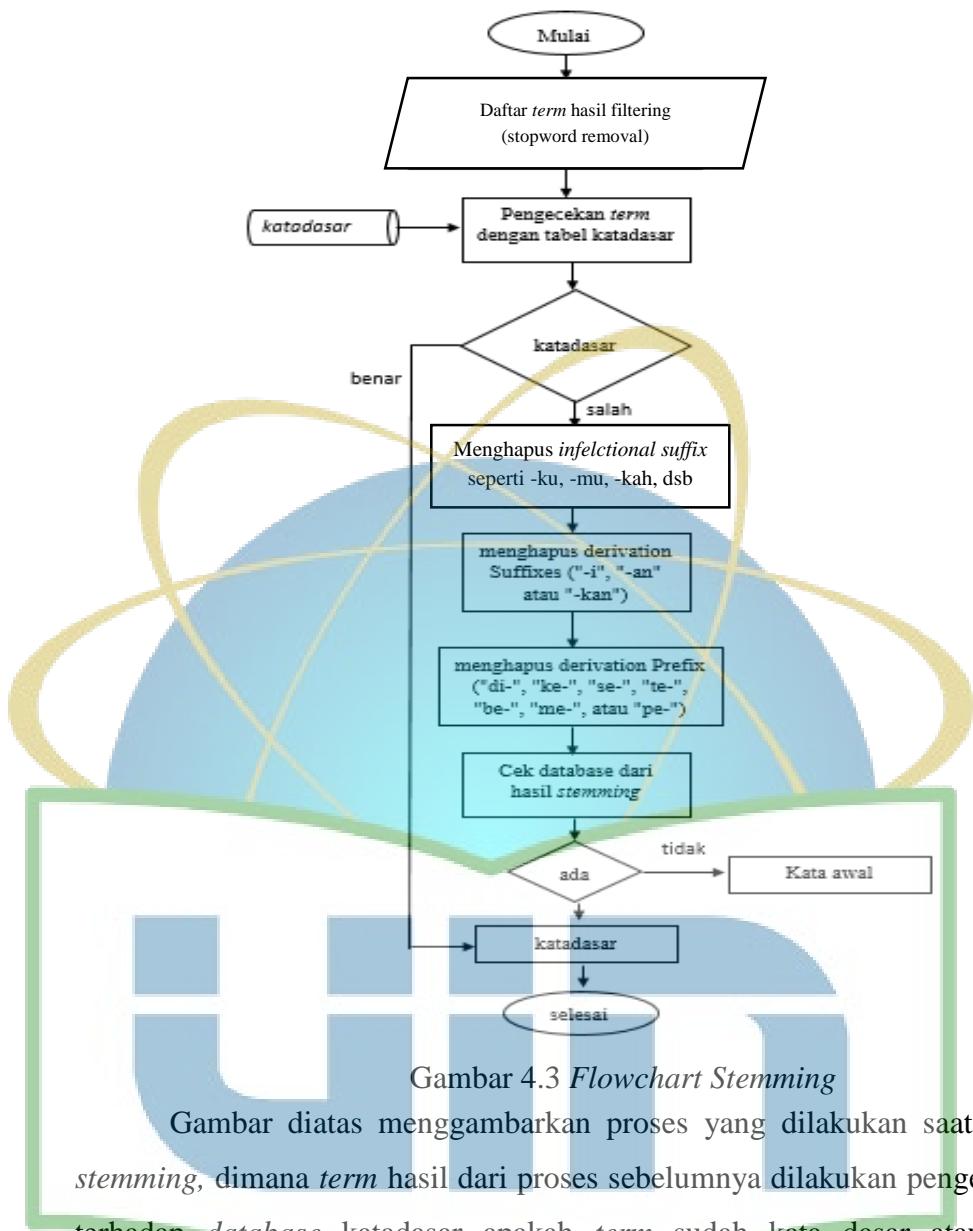
Sesudah Tokenizing	Sesudah Proses Stopword
dari	-
abu	abu
hurairah	hurairah
Ia	-
berkata	-
rasulullah	rasulullah
saw	saw
bersabda	bersabda

allah	allah
tidak	-
menerima	menerima
shalat	shalat
seseorang	seseorang
diantara	-
kalian	kalian
ketika	-
berhadats	berhadats
hingga	-
Ia	-
berwudhu	berwudhu

### 3. *Stemming*

*Stemming* adalah salah satu cara yang digunakan untuk mengembalikan kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya, contohnya kata-kata menyukseskan, tersukseskan dan disukseskan akan ditransformasi ke stem yang sama yaitu sukses. Adapun algoritma *stemming* yang akan digunakan yaitu algoritma *stemming* Nazief & Adriani, dimana *stemming* jenis ini merupakan *stemming* yang memiliki tingkat akurasi (presisi) lebih tinggi dari jenis *stemming* yang lain.

Proses yang berjalan saat tahap *stemming*, akan digambarkan dalam *flowchart* berikut:



Gambar 4.3 Flowchart Stemming

Gambar diatas menggambarkan proses yang dilakukan saat tahap *stemming*, dimana *term* hasil dari proses sebelumnya dilakukan pengecekan terhadap *database* katadasar apakah *term* sudah kata dasar atau kata berimbuhan. Apabila *term* merupakan kata berimbuhan maka akan dilakukan *stemming* dengan melalui 3 tahapan yaitu menghapus *inflection suffix* (seperti -ku, -mu, -kah, dsb), menghapus *derivation suffix* (seperti -i, -an, atau -kan), dan menghapus *derivation prefix* (seperti di-, ke-, se-, dsb). Hasil dari proses *stemming* ini akan dilanjutkan dengan tahap berikutnya untuk dilakukan pembobotan kata menggunakan algoritma tf-idf.

Hasil *stemming* dari algoritma nazief adriani adalah sebagai berikut:

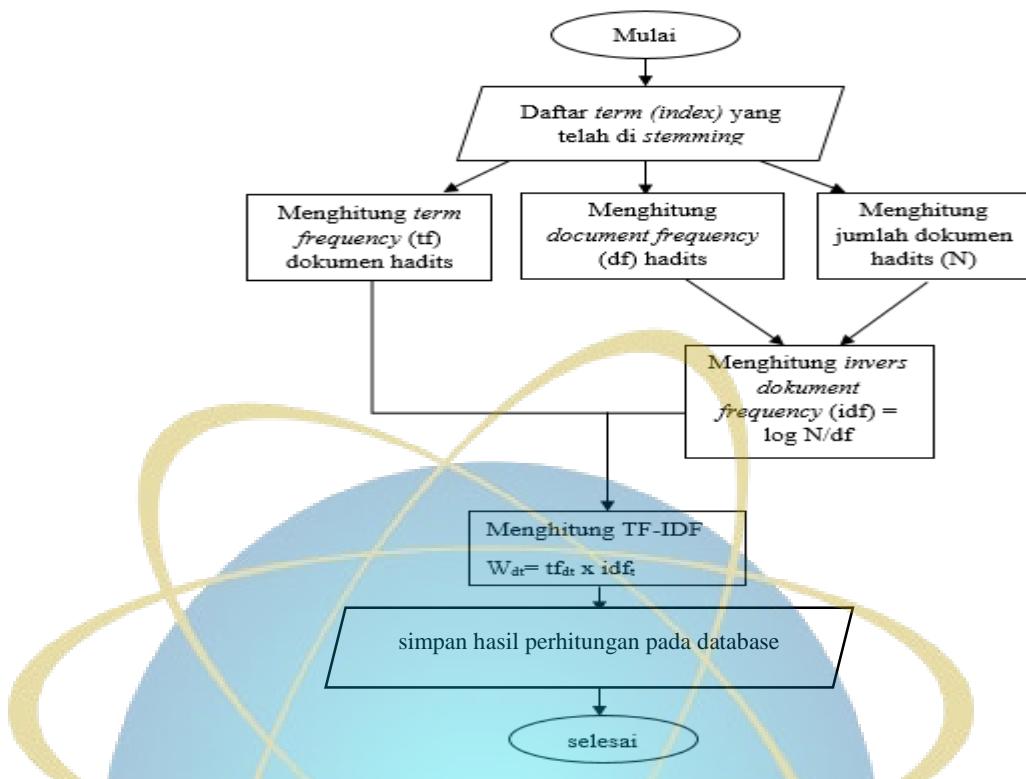
Tabel. 4.3 Hasil *Stemming*

Sesudah Proses <i>Stopword</i>	Hasil <i>Stemming</i>
abu	abu
hurairah	hurairah
rasulullah	rasulullah
saw	saw
bersabda	sabda
allah	allah
menerima	terima
shalat	shalat
seseorang	orang
kalian	kalian
berhadats	hadats
berwudhu	wudhu

#### 4.2.1.2 Pembobotan Kata (Tf-Idf)

Setelah tahap teks *preprocessing* selesai, tahap selanjutnya adalah pembobotan kata (*term*). Dalam pembobotan kata (*term*) ini, setiap kata yang telah melewati proses *preprocessing*, akan di *parsing* terlebih dahulu dan disimpan dalam *database*, kemudian dihitung jumlah kemunculan setiap katanya, dimana semakin besar kemunculan suatu kata dalam dokumen maka akan semakin besar pula kesesuaian kata tersebut.

Pada proses pembobotan kata (TF-IDF) akan digambarkan dalam *flowchart* berikut:



Gambar 4.4 Flowchart TF-IDF

Gambar diatas menggambarkan proses yang dilakukan saat tahap pembobotan kata dengan menggunakan metode *term frequency inverse document frequency* (TF-IDF), dimana daftar *term* yang telah di *stemming* dilakukan perhitungan untuk mengetahui bobot perkata dengan menghitung jumlah *term frequency* dokumen (TF) terlebih dahulu, kemudian menghitung nilai jumlah dokumen yang memiliki *term* (DF), dan selanjutnya menghitung nilai idf dengan rumus  $\log=N/df$ , dimana  $N$  merupakan jumlah seluruh dokumen yang ada. Setelah nilai TF dan IDF sudah didapat, maka langkah terakhir adalah menentukan bobot kata dengan mengalikan TF dan IDF dengan rumus  $W_{dt}=TF_{dt} \times IDF_t$ . Hasil dari proses perhitungan ini disimpan dalam database dan akan dilanjutkan dengan tahap berikutnya untuk dilakukan perhitungan *cosine similarity* yang merupakan tahap akhir proses.

Contoh perhitungan pembobotan kata dalam penelitian ini menggunakan dokumen yang telah dilakukan teks *preprocessing* diatas (sebagai *query*) terhadap dua dokumen *output* untuk mengetahui kemiripannya, adalah sebagai berikut:

Diketahui : Query => Dari Abu Hurairah ia berkata, “Rasulullah SAW bersabda: ‘Allah tidak menerima sholat seseorang di antara kalian ketika berhadats, hingga ia berwudhu’.”

Dimana terdapat 2 dokumen sebagai berikut:

Tabel. 4.4 Dokumen *Output*

d1	<ul style="list-style-type: none"> <li>-Shalat orang yang berhadats tidak diterima, sampai ia bersuci dari hadats besar dan kecil.</li> <li>-Hadats membatalkan wudhu, dan membatalkan shalat jika terjadi di sela-selanya.</li> <li>-Maksud tidak diterima dalam hadits ini adalah shalatnya tidak sah.</li> <li>-Hadits ini menunjukkan, thaharah adalah syarat sah shalat.</li> </ul>
d2	<ul style="list-style-type: none"> <li>-Wajib memerhatikan anggota-anggota wudhu dan tidak boleh mengabaikan sedikitpun diantaranya.</li> <li>-Ancaman keras bagi yang tidak baik dalam berwudhu.</li> <li>-Kedua kaki wajib dibasuh saat berwudhu, seperti disebutkan dalam banyak dalil shahih dan ijma' umat.</li> </ul>

Pembahasan: sebelum dilakukan pembobotan kata antara *query* dan dokumen yang ada adalah dengan melakukan teks *preprocessing* terlebih dahulu agar kata siap untuk dihitung. Hasil dari *preprocessing* dapat dilihat pada tabel dibawah ini:

Tabel. 4.5 Hasil *Preprocessing* Text

Dokumen	Dokumen yang sudah di <i>preprocessing</i>
Q	abu hurairah rasulullah saw sabda allah terima shalat orang kalian hadats wudhu
d1	shalat orang hadats terima suci hadats besar kecil hadats batal wudhu batal shalat sela maksud terima hadits shalat sah

	hadits tunjuk thaharah syarat sah shalat
d2	wajib hati anggota wudhu abai sedikit ancaman keras orang baik wudhu dua kaki wajib basuh wudhu banyak dalil shahih ijmak umat

Berdasarkan *term* yang terbentuk diatas, maka untuk melakukan pembobotan antara *query* dengan dokumen yang ada adalah dengan melakukan perhitungan tf-idf, dengan rincian sebagai berikut:

Tabel 4.6 Hasil Pembobotan Kata (*term*)

<i>term</i>	tf			df	Idf $\log(N/df)$	Wdt= $tf_{dt} \times idf_t$		
	Q	d1	d2			Q	d1	d2
abai			1	1	0,477	0	0	0,477
abu	1			1	0,477	0,477	0	0
allah	1			1	0,477	0,477	0	0
ancaman			1	1	0,477	0	0	0,477
anggota			1	1	0,477	0	0	0,477
baik			1	1	0,477	0	0	0,477
banyak			1	1	0,477	0	0	0,477
basuh			1	1	0,477	0	0	0,477
batal		2		2	0,176	0	0,352	0
besar		1		1	0,477	0	0,477	0
dalil			1	1	0,477	0	0	0,477
dua			1	1	0,477	0	0	0,477
hadats	1	3		4	0,125	0,125	0,375	0
hadits		2		2	0,176	0	0,352	0
hati			1	1	0,477	0	0	0,477
hurairah	1			1	0,477	0,477	0	0
ijmak			1	1	0,477	0	0	0,477
kaki			1	1	0,477	0	0	0,477
kalian	1			1	0,477	0,477	0	0

kecil		1		1	0,477	0	0,477	0
keras			1	1	0,477	0	0	0,477
maksud		1		1	0,477	0	0,477	0
orang	1		1	2	0,176	0,176	0	0,176
rasulullah	1			1	0,477	0,477	0	0
sabda	1			1	0,477	0,477	0	0
sah		2		2	0,176	0	0,352	0
saw	1			1	0,477	0,477	0	0
sedikit			1	1	0,477	0	0	0,477
selal		1		1	0,477	0	0,477	0
shahih			1	1	0,477	0	0	0,477
shalat	1	4		5	0,221	0,221	0,884	0
suci		1		1	0,477	0	0,477	0
syarat		1		1	0,477	0	0,477	0
terima	1	2		3	0	0	0	0
thaharah		1		1	0,477	0	0,477	0
tunjuk		1	1	1	0,477	0	0,477	0
umat			1	1	0,477	0	0	0,477
wajib			1	1	0,477	0	0	0,477
wudhu	1	1	3	5	0,221	0,221	0,221	0,663

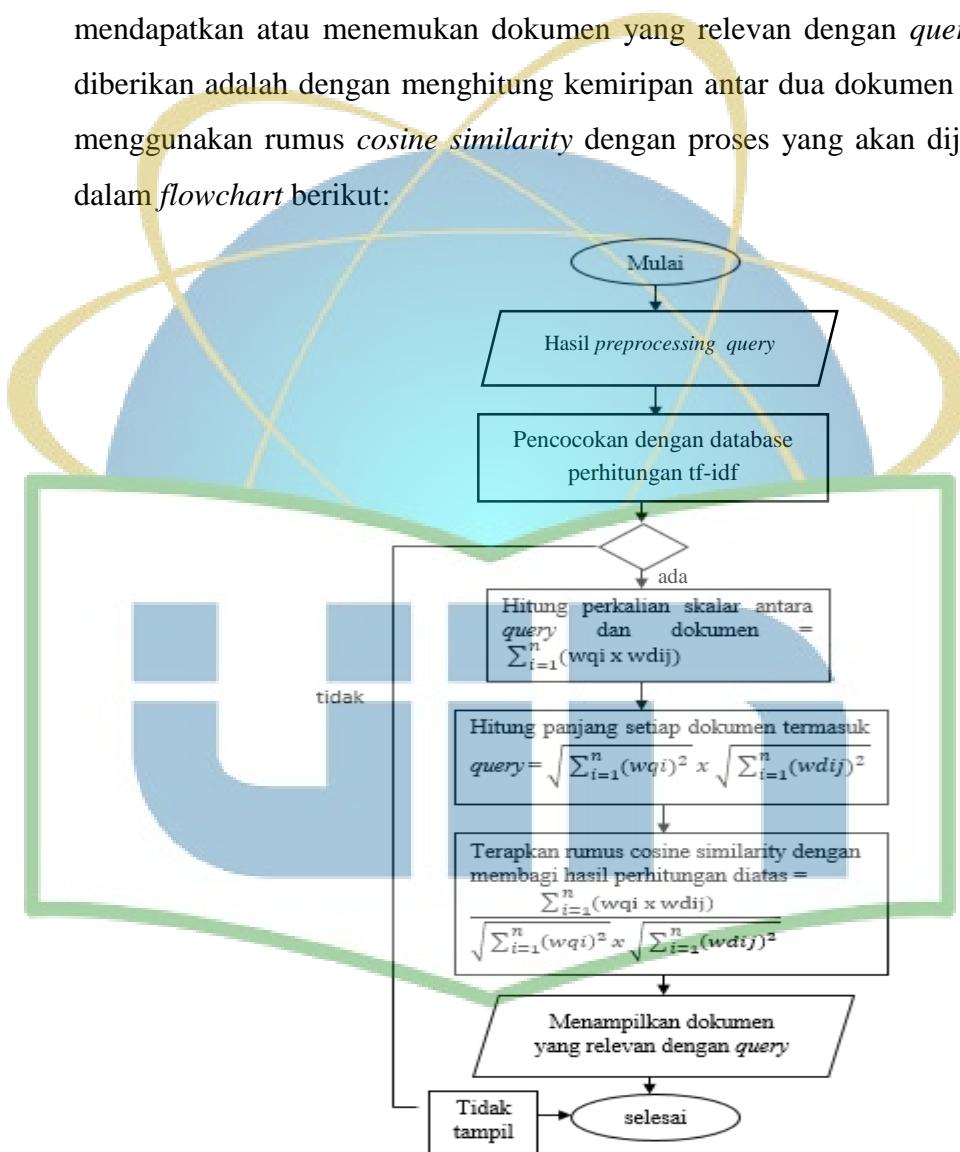
Penjelasan :

1. *Term frequency* (tf) merupakan frekuensi kemunculan *term* (t) pada dokumen (d).
2. *Document frequency* (df) adalah banyaknya dokumen dimana suatu *term* (t) muncul.
3. Menghitung *invers document frequency* (idf) dengan menggunakan rumus  $idf = \log N/df$ . Dimana N merupakan jumlah dokumen yaitu 3 (Q, d1, dan d2).

4. Untuk menghitung nilai *term frequency invers document frequency* (tf-idf) adalah  $\Rightarrow W_{dt} = tf_{dt} \times idf_t$  yaitu mengalikan nilai tf dokumen terhadap *term* dengan nilai idf *term*.

#### 4.2.1.3 Cosine Similarity

Setelah melakukan pembobotan dokumen terhadap *term* dengan menggunakan perhitungan tf-idf, langkah terakhir yang dilakukan untuk mendapatkan atau menemukan dokumen yang relevan dengan *query* yang diberikan adalah dengan menghitung kemiripan antar dua dokumen dengan menggunakan rumus *cosine similarity* dengan proses yang akan dijelaskan dalam *flowchart* berikut:



Gambar 4.5 Flowchart Cosine Similarity

Gambar diatas menggambarkan proses yang dilakukan saat tahap menemukan dokumen yang relevan dengan *query user* menggunakan metode *cosine similarity*, dimana *query* yang dimasukkan *user* dilakukan tahap

*preprocessing* yang hasilnya dicocokan dengan *database* bobot yaitu hasil perhitungan tf-idf, apabila *term* ditemukan maka akan dihitung perkalian skalar antara *term query* dengan dokumen dengan rumus  $w_{qi} \times w_{dij}$ , selanjutnya yaitu menghitung nilai panjang setiap dokumen termasuk *query* dengan mengkuadratkan bobot *query* dan bobot dokumen yang selanjutnya diakarkan. Setelah itu, terakhir dengan membagi hasil dari perkalian skalar dan hasil panjang vektor yang sudah dihitung untuk menemukan hasil kemiripan antara *query* dengan dokumen, lalu sistem akan menampilkan dokumen yang relevan dengan *query* berdasarkan hasil perhitungan kemiripan dengan *cosine similarity* tersebut.

Contoh perhitungan berdasarkan hasil dari pembobotan kata yang telah dilakukan sebelumnya, untuk mengetahui kemiripan *query* dengan  $d_1$  yang nanti akan dibandingkan dengan  $d_2$ , maka yang harus dilakukan pertama adalah dengan melakukan perhitungan skalar antara *query* dengan  $d_1$  dan  $d_2$  dan hasil perkalian dari dokumen dengan *query* dijumlahkan. Setelah itu menghitung panjang dokumen, termasuk *query*. Caranya, kuadratkan bobot setiap *term* dalam dokumen, jumlahkan nilai kuadrat dan akarkan, terakhir hitung nilai *cosine similarity* nya, seperti dalam tabel dibawah ini:

Tabel 4.7 Hasil Panjang Vektor

term	w <sub>qi</sub> x w <sub>dij</sub>		Panjang vektor		
	d <sub>2</sub>	d <sub>3</sub>	Q	d <sub>2</sub>	d <sub>3</sub>
abai	0	0	0	0	0,227
abu	0	0	0,227	0	0
allah	0	0	0,227	0	0
ancaman	0	0	0	0	0,227
anggota	0	0	0	0	0,227
baik	0	0	0	0	0,227
banyak	0	0	0	0	0,227
basuh	0	0	0	0	0,227
batal	0	0	0	0,124	0

besar	0	0	0	0,227	0
dalil	0	0	0	0	0,227
dua	0	0	0	0	0,227
hadats	0,046	0	0,015	0,14	0
hadits	0	0	0	0,124	0
hati	0	0	0	0	0,227
hurairah	0	0	0,227	0	0
ijmak	0	0	0	0	0,227
kaki	0	0	0	0	0,227
kalian	0	0	0,227	0	0
kecil	0	0	0	0,227	0
keras	0	0	0	0	0,227
maksud	0	0	0	0,227	0
orang	0	0,03	0,03	0	0,03
rasulullah	0	0	0,227	0	0
sabda	0	0	0,227	0	0
sah	0	0	0	0,124	0
saw	0	0	0,227	0	0
sedikit	0	0	0	0	0,227
sela	0	0	0	0,227	0
shahih	0	0	0	0	0,227
shalat	0,195	0	0,048	0,781	0
suci	0	0	0	0,227	0
syarat	0	0	0	0,227	0
terima	0	0	0	0	0
thaharah	0	0	0	0,227	0
tunjuk	0	0	0	0,227	0
umat	0	0	0	0	0,227
wajib	0	0	0	0	0,227
wudhu	0,048	0,146	0,048	0,048	0,439

Jumlahkan			Jumlahkan hasil kuadrat diatas		
	0,289	0,176	1.73	3.157	3.874
			Akarkan hasil penjumlahan diatas		
			1,315	1,776	1,968

Setelah diketahui nilai dari masing-masing *query* dan dokumen, selanjutnya adalah dengan menerapkan rumus *cosine similarity* sebagai berikut:

$$1. \text{ Cos}(q, d_1) = \frac{\sum_{i=1}^n (wqi \times wd_1)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wd_1)^2}} = \frac{0,289}{\sqrt{1,315} \times \sqrt{1,776}} = \frac{0,289}{2,335} = 0,123$$

$$2. \text{ Cos}(q, d_2) = \frac{\sum_{i=1}^n (wqi \times wd_2)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wd_2)^2}} = \frac{0,176}{\sqrt{1,315} \times \sqrt{1,968}} = \frac{0,176}{2,588} = 0,068$$

Atau dapat dilakukan dengan menggunakan perhitungan lain seperti berikut ini :

$$\begin{aligned} 1. \text{Similarity} &= \cos(Q, D_1) = \frac{Q \cdot D}{|Q||D|} = \frac{\sum_{i=1}^n (wqi \times wdij)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wdij)^2}} \\ &= \frac{(wq1x wd1)+(wq2x wd1)+(wq3x wd1)+(wq4x wd1)+(wq5x wd1)+ \\ &\quad (wq6x wd1)+(wq7x wd1)+(wq8x wd1)+(wq9x wd1)+(wq10x wd1)+ \\ &\quad (wq11x wd1)+(wq12x wd1)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wdij)^2}} \\ &= \frac{(0,477 \times 0)+(0,477 \times 0)+(0,477 \times 0)+(0,477 \times 0)+(0,477 \times 0)+ \\ &\quad (0,477 \times 0)+(0 \times 0)+(0,221 \times 0)+(0,176 \times 0,176)+(0,477 \times 0)+ \\ &\quad (0,125 \times 0,375)+(0,221 \times 0,221)}{\sqrt{(0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+0^2+ \\ &\quad 0,221^2+0,176^2+0,477^2+0,125^2+0,211^2)} \times \\ &\quad \sqrt{(0,352^2+0,477^2+0,375^2+0,352^2+0,477^2+0,477^2+0,352^2 \\ &\quad 0,477^2+0,884^2+0,477^2+0,477^2+0,477^2+0,477^2+0,221^2)}} \\ &= \frac{0,289}{\sqrt{1,73} \times \sqrt{3,158}} = \frac{0,289}{2,335} = 0,123. \end{aligned}$$

$$2. \text{Similarity} = \cos(Q, D_2) = \frac{Q \cdot D}{|Q||D|} = \frac{\sum_{i=1}^n (wqi \times wdij)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wdij)^2}}$$

$$\begin{aligned}
 &= \frac{(wq1x wd2)+(wq2x wd2)(wq3x wd2)+(wq4x wd2)+(wq5x wd2)+ \\
 &\quad (wq6x wd2)+(wq7x wd2)+(wq8x wd2)+(wq9x wd2)+(wq10x wd2)+ \\
 &\quad (wq11x wd2)+(wq12x wd2)}{\sqrt{\sum_{i=1}^n (wqi)^2} \times \sqrt{\sum_{i=1}^n (wdi)^2}} \\
 &= \frac{(0,477 \times 0)+(0,477 \times 0)+(0,477 \times 0)+(0,477 \times 0)+ \\
 &\quad (0,477 \times 0)+(0 \times 0)+(0,176 \times 0,176)+(0,176 \times 0)+(0,477 \times 0)+ \\
 &\quad (0,125 \times 0)+(0,221 \times 0,663))}{\sqrt{(0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+0^2+ \\
 &\quad 0,221^2+0,176^2+0,477^2+0,125^2+0,211^2) \times} \\
 &\quad \sqrt{(0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+ \\
 &\quad 0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+0,477^2+ \\
 &\quad 0,477^2+0,477^2+0,176^2+0,663^2)} \\
 &= \frac{0,176}{\sqrt{1,73 \times 3,874}} = \frac{0,176}{2,588} = 0,068.
 \end{aligned}$$

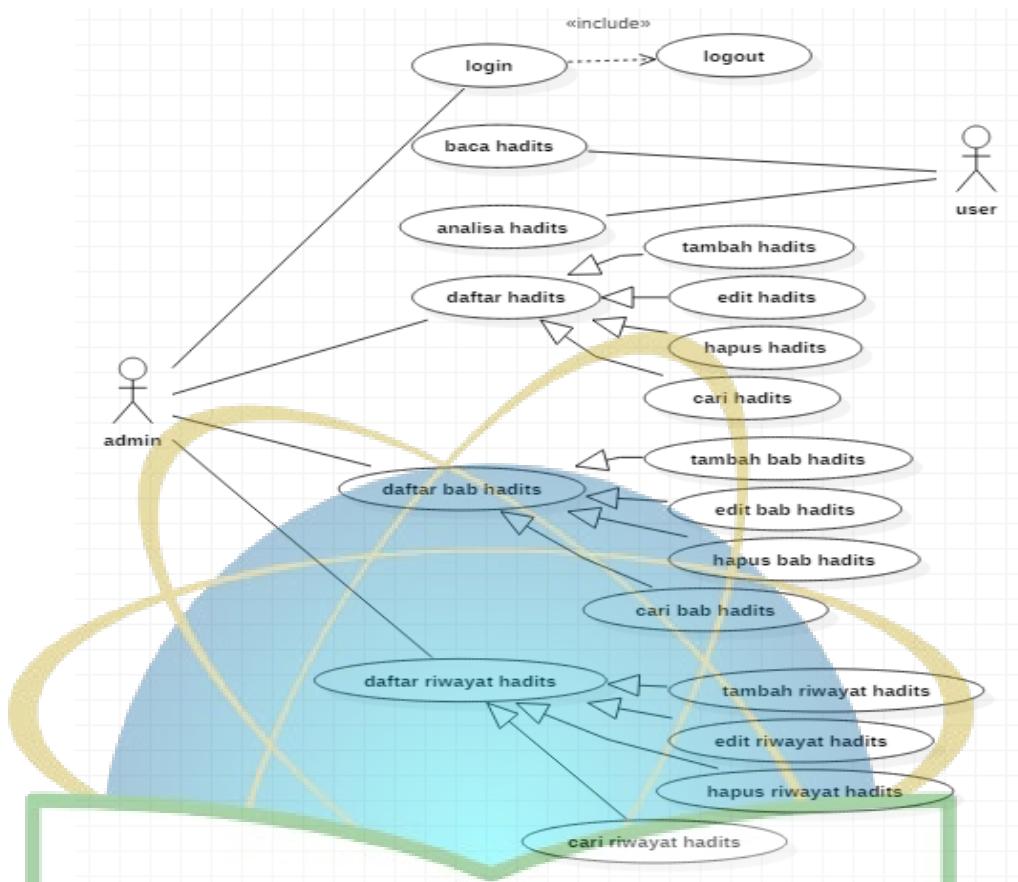
Berdasarkan hasil dari perhitungan *cosine similarity* diatas, hasil dari persamaan antara *query* dan d1 adalah 0,123 sedangkan hasil dari persamaan antara *query* dan d2 adalah 0,068. Maka dapat disimpulkan bahwa dokumen yang terdekat dengan *query* adalah dokumen 2 (d1). Hal tersebut dikarenakan hukum *cosine similarity* adalah semakin besar nilai *cosinus* (maksimal 1) maka semakin mirip dokumen dengan yang dibandingkan. Oleh karena itu, yang lebih mirip dengan *query* dan menjadi *output* dari *query* adalah d1 karena lebih mendekati angka 1 dibandingkan dengan dokumen 2 (d2).

#### 4.2.2 Merancang UML

Pada penelitian ini penulis merancang diagram UML (*Unified Modelling Language*) sebagai berikut:

##### 4.2.2.1 Usecase Diagram

Untuk menggambarkan interaksi yang terjadi dalam sistem maka dapat dilihat dari *Usecase Diagram* dibawah ini:



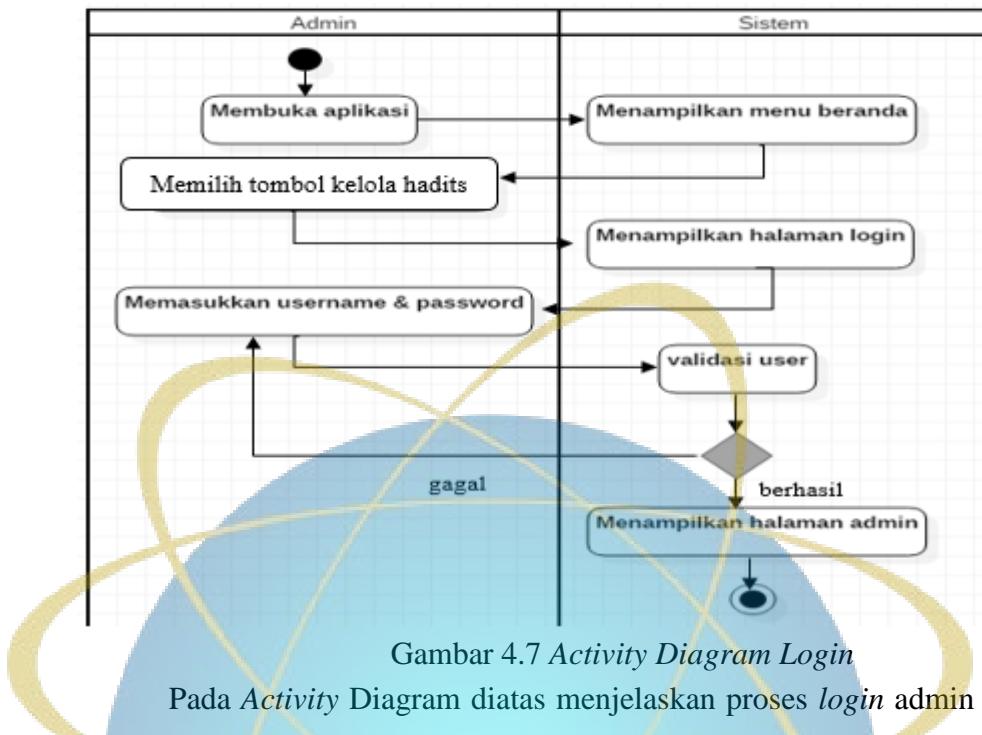
Gambar 4.6 Usecase Diagram

Pada gambar *usecase diagram* diatas dijelaskan bahwa terdapat dua aktor yaitu admin dan *user*, dimana aktivitas yang dapat dilakukan oleh admin meliputi *login <<include>> logout*, (menambah, menghapus, mengedit, dan mencari) *hadits*, (menambah, menghapus, mengedit, dan mencari) *bab hadits*, serta (menambah, menghapus, mengedit, dan mencari) *riwayat hadits*. Sedangkan aktivitas yang dapat dilakukan oleh aktor *user* adalah membaca *hadits* dan menganalisa *hadits* untuk mengetahui *syarah hadits* yang ingin diketahui.

#### 4.2.2.2 Activity Diagram

Berikut merupakan *Activity Diagram* dari beberapa proses berdasarkan *usecase* diatas:

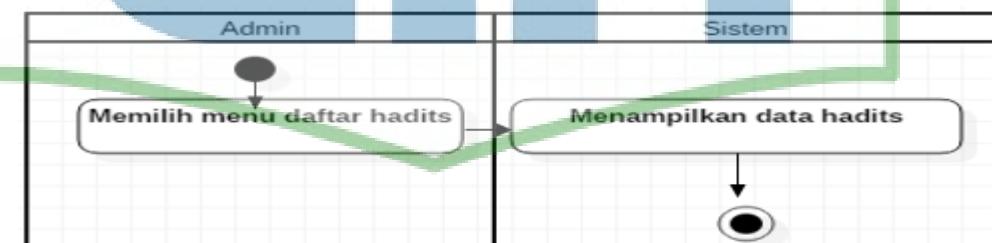
### 1. Activity Diagram Login



Gambar 4.7 Activity Diagram Login

Pada Activity Diagram diatas menjelaskan proses *login* admin untuk masuk ke dalam menu admin. Untuk dapat melakukan aktivitas sebagai admin, maka admin harus *login* terlebih dahulu dengan menekan tombol kelola admin lalu memasukkan *username* dan *password*, kemudian sistem akan memverifikasi. Apabila data valid maka admin dapat masuk kedalam sistem, apabila data tidak valid maka admin harus memasukkan ulang *username* dan *password* yang valid.

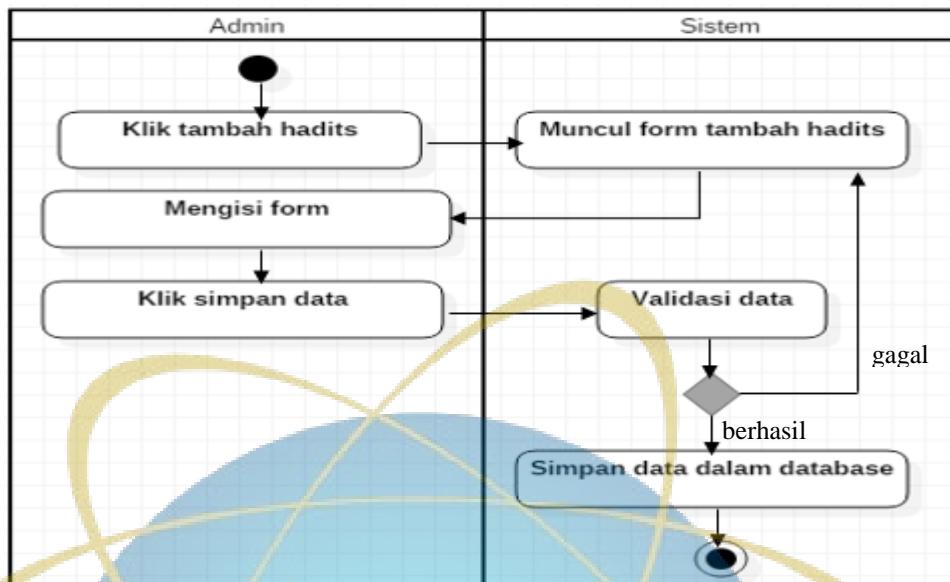
### 2. Activity Diagram Daftar Hadits



Gambar 4.8 Activity Diagram Daftar Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* untuk melihat daftar hadits, yaitu dengan memilih menu daftar hadits dan sistem akan menampilkan data hadits yang ada.

### 3. Activity Diagram Tambah Hadits



Gambar 4.9 Activity Diagram Tambah Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas tambah hadits. Admin mengklik tombol tambah hadits, lalu sistem akan menampilkan form tambah hadits. Selanjutnya admin mengisi form yang ada, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam database, dan apabila gagal maka sistem akan memunculkan kembali form tambah hadits.

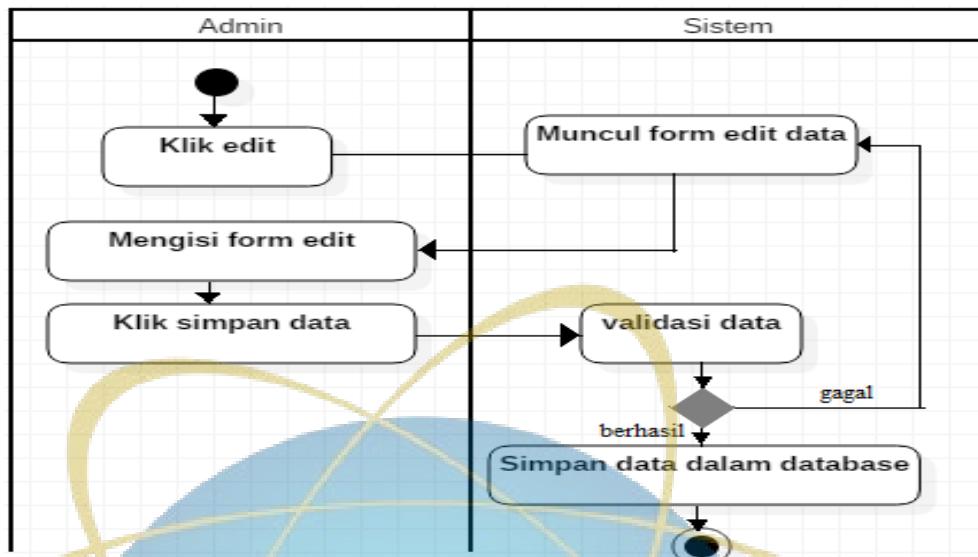
### 4. Activity Diagram Hapus Hadits



Gambar 4.10 Activity Diagram Hapus Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas hapus hadits, yaitu dengan klik tombol *delete* lalu sistem akan menghapus data dari *database*.

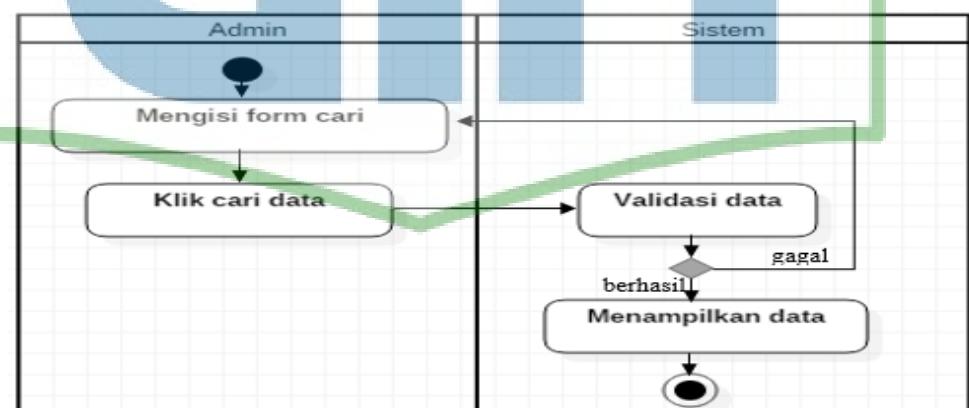
### 5. Activity Diagram Edit Hadits



Gambar 4.11 Activity Diagram Edit Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas edit hadits. Admin mengklik tombol edit yang ada, kemudian sistem akan menampilkan form edit data hadits, lalu admin mengedit data dengan mengisi form edit, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*, dan apabila gagal maka sistem akan memunculkan kembali form edit data hadits.

### 6. Activity Diagram Cari Hadits

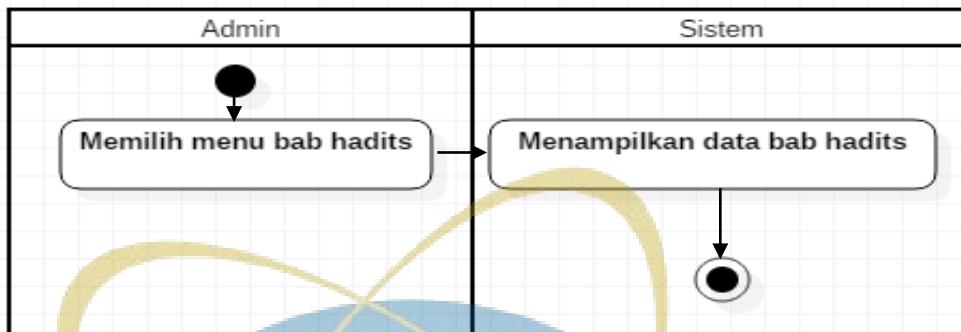


Gambar 4.12 Activity Diagram Cari Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas cari hadits, yaitu dengan mengisi form cari hadits lalu klik cari data dan sistem akan melakukan validasi. Apabila data yang dicari

ada, maka sistem akan menampilkan data hadits yang dicari, namun jika tidak ada maka admin akan mengisi form cari data kembali.

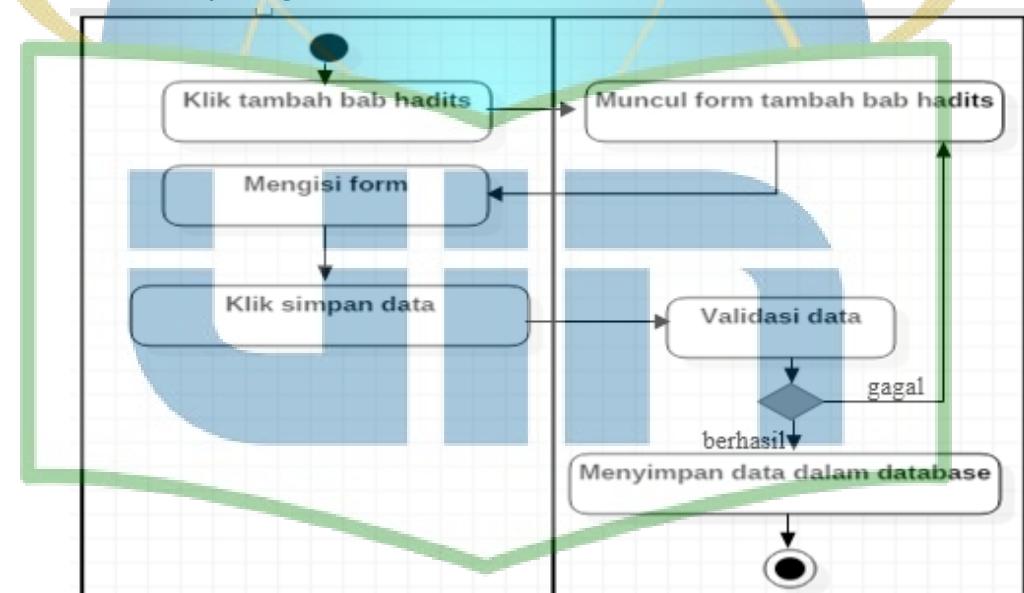
### 7. Activity Diagram Bab Hadits



Gambar 4.13 Activity Diagram Daftar Bab Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* untuk melihat daftar bab hadits, yaitu dengan memilih menu daftar bab hadits dan sistem akan menampilkan data dari bab hadits yang ada.

### 8. Activity Diagram Tambah Bab Hadits

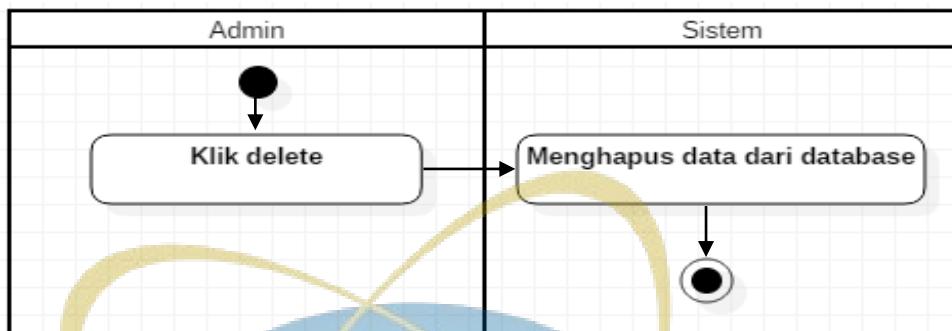


Gambar 4.14 Activity Diagram Tambah Bab Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas tambah bab hadits. Admin mengklik tombol tambah bab hadits, lalu sistem akan menampilkan form tambah bab hadits. Selanjutnya admin mengisi form yang ada, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan

tersimpan dalam *database*, dan apabila gagal maka sistem akan memunculkan kembali form tambah bab hadits.

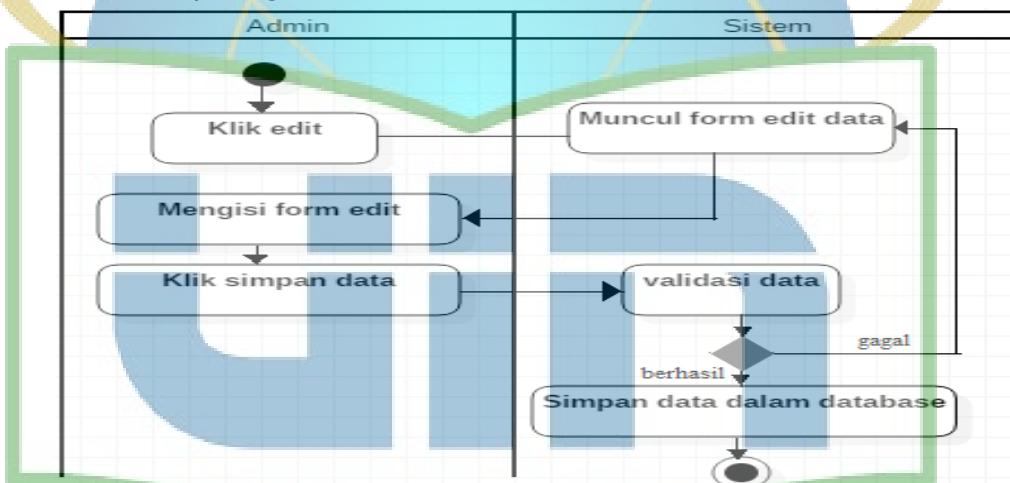
### 9. Activity Diagram Hapus Bab Hadits



Gambar 4.15 Activity Diagram Hapus Bab Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas hapus bab hadits, yaitu dengan klik tombol *delete* lalu sistem akan menghapus data dari *database*.

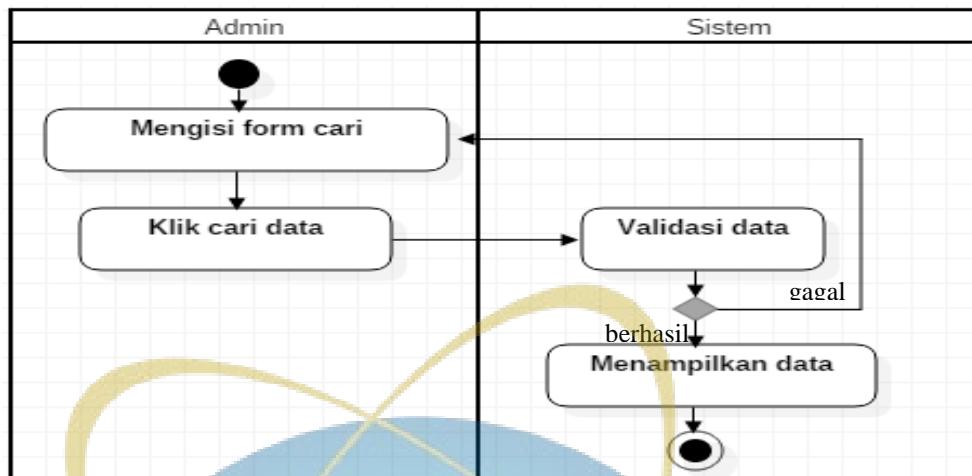
### 10. Activity Diagram Edit Bab Hadits



Gambar 4.16 Activity Diagram Edit Bab Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas edit bab hadits. Admin mengklik tombol edit yang ada, kemudian sistem akan menampilkan form edit data bab hadits, lalu admin mengedit data dengan mengisi form edit, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*, dan apabila gagal maka sistem akan memunculkan kembali form edit data bab hadits.

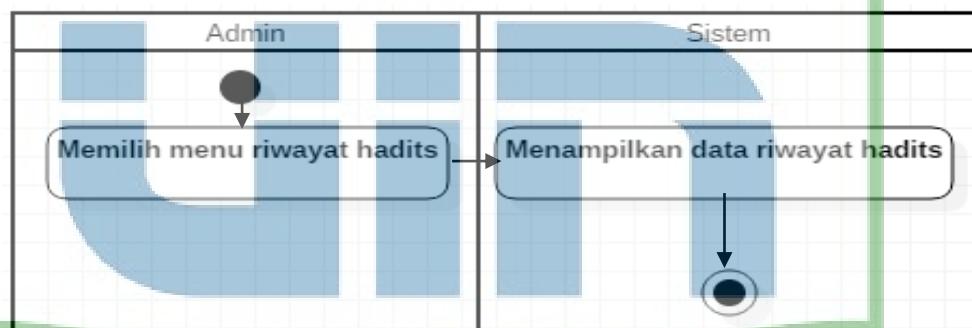
### 11. Activity Diagram Cari Bab Hadits



Gambar 4.17 Activity Diagram Cari Bab Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas cari bab hadits, yaitu dengan mengisi form cari bab hadits lalu klik cari data dan sistem akan melakukan validasi. Apabila data yang dicari ada, maka sistem akan menampilkan data bab hadits yang dicari, namun jika tidak ada maka admin akan mengisi form cari data kembali.

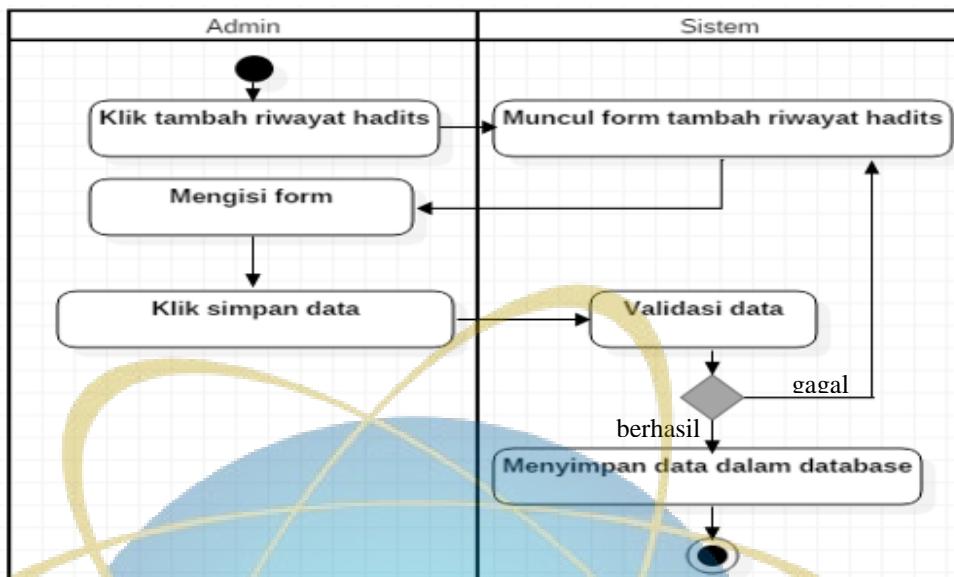
### 12. Activity Diagram Riwayat Hadits



Gambar 4.18 Activity Diagram Daftar Riwayat Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* untuk melihat daftar riwayat hadits, yaitu dengan memilih menu daftar riwayat hadits dan sistem akan menampilkan data dari riwayat hadits yang ada.

### 13. Activity Diagram Tambah Riwayat Hadits



Gambar 4.19 Activity Diagram Tambah Riwayat Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas tambah riwayat hadits. Admin mengklik tombol tambah riwayat hadits, lalu sistem akan menampilkan form tambah riwayat hadits. Selanjutnya admin mengisi form yang ada, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*, dan apabila gagal maka sistem akan memunculkan kembali form tambah riwayat hadits.

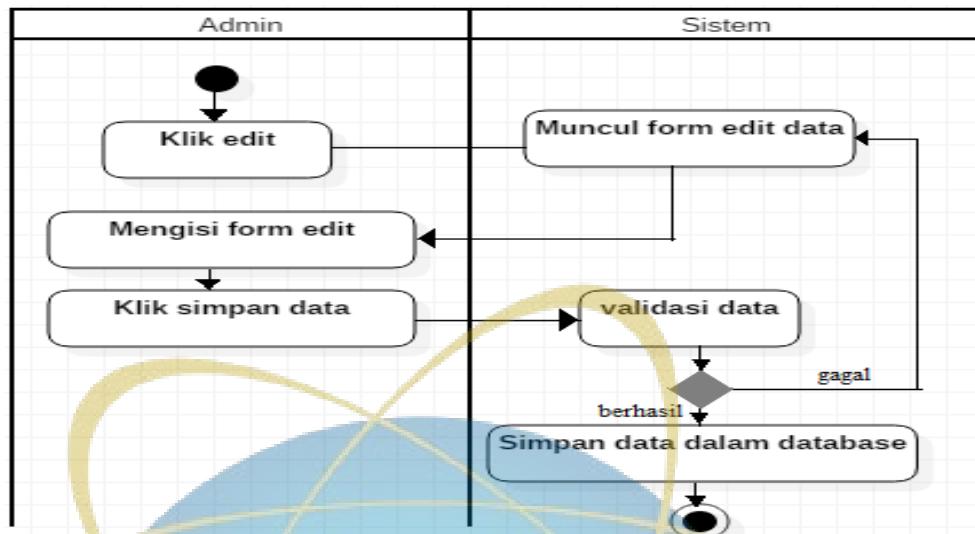
### 14. Activity Diagram Hapus Riwayat Hadits



Gambar 4.20 Activity Diagram Hapus Riwayat Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas hapus riwayat hadits, yaitu dengan klik tombol *delete* lalu sistem akan menghapus data dari *database*.

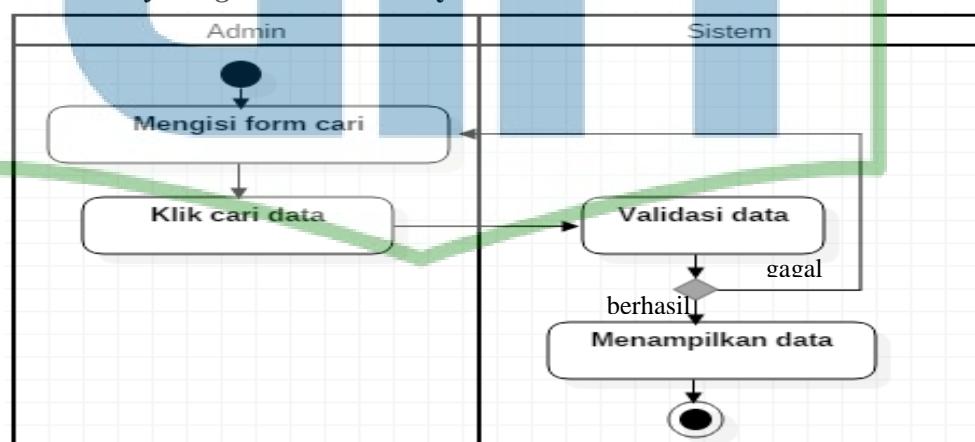
### 15. Activity Diagram Edit Riwayat Hadits



Gambar 4.21 Activity Diagram Edit Riwayat Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas edit riwayat hadits. Admin mengklik tombol edit yang ada, kemudian sistem akan menampilkan form edit data riwayat hadits, lalu admin mengedit data dengan mengisi form edit, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*, dan apabila gagal maka sistem akan memunculkan kembali form edit data bab hadits.

### 16. Activity Diagram Cari Riwayat Hadits

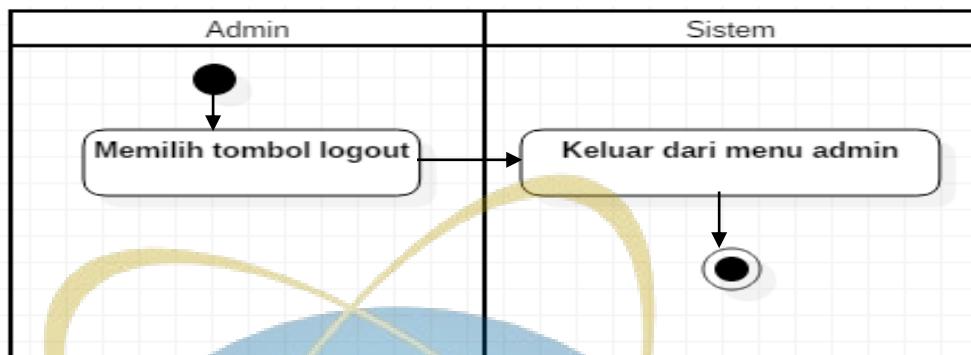


Gambar 4.22 Activity Diagram Cari Riwayat Hadits

Pada Activity Diagram diatas menjelaskan proses *admin* dalam melakukan aktivitas cari riwayat hadits, yaitu dengan mengisi form cari riwayat hadits lalu klik cari data dan sistem akan melakukan validasi. Apabila

data yang dicari ada, maka sistem akan menampilkan data riwayat hadits yang dicari, namun jika tidak ada maka admin akan mengisi form cari data kembali.

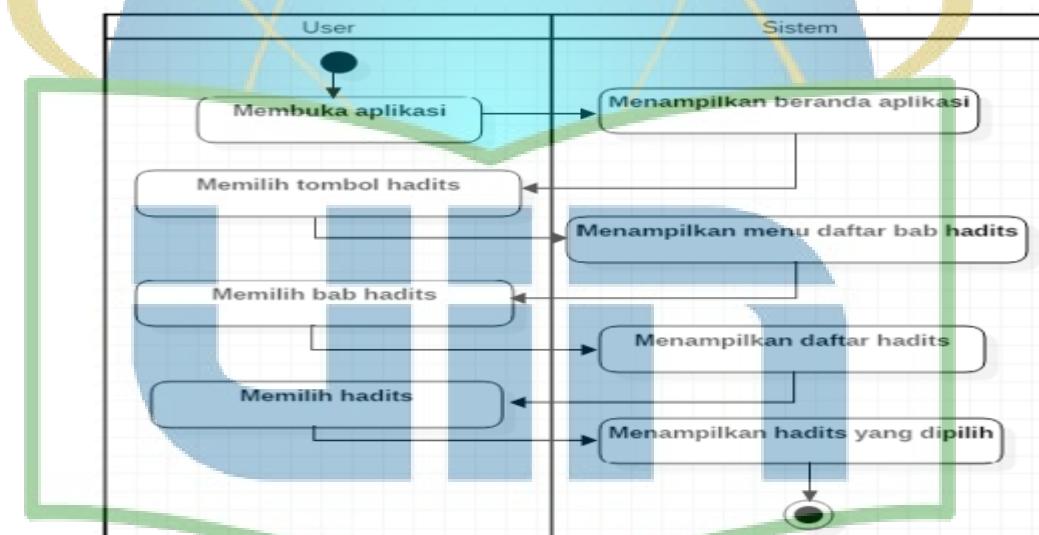
### 17. Activity Diagram Logout



Gambar 4.23 Activity Diagram Logout

Pada Activity Diagram diatas menjelaskan proses *logout* admin yaitu admin mengklik tombol *logout* lalu akan keluar dari menu admin.

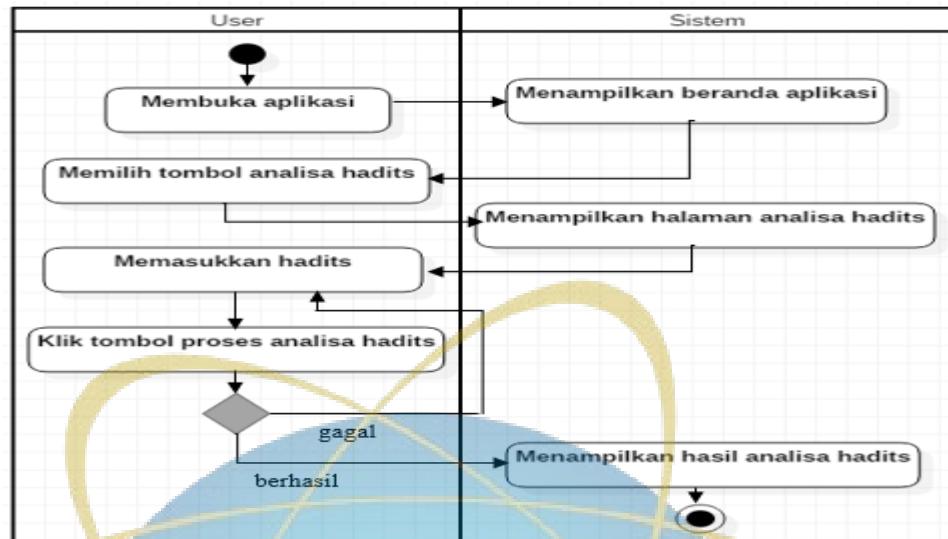
### 18. Activity Diagram Baca Hadits



Gambar 4.24 Activity Diagram Baca Hadits

Pada Activity Diagram diatas menjelaskan proses *user* membaca hadits. Untuk membaca hadits, *user* memilih menu hadits yang terdapat dalam beranda, lalu sistem akan menampilkan menu daftar bab hadits. Selanjutnya user memilih bab hadits, kemudian sistem akan menampilkan daftar hadits sesuai dengan bab yang telah user pilih. Selanjutnya, user memilih hadits yang akan dibaca, dan sistem menampilkan hadits yang telah dipilih *user* tersebut.

## 19. Activity Diagram Analisa Hadits



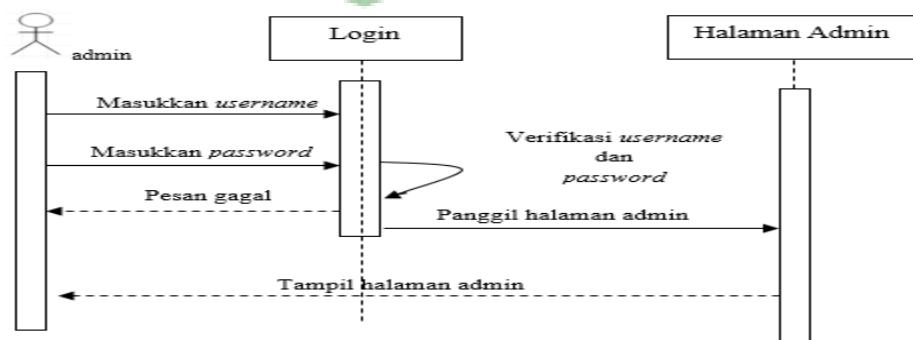
Gambar 4.25 Activity Diagram Analisa Hadits

Pada *Activity Diagram* diatas menjelaskan proses *user* dalam melakukan analisa. Untuk mengetahui *syarah hadits*, *user* memilih menu analisa hadits dalam beranda, maka sistem akan menampilkan halaman analisa hadits. Selanjutnya *user* memasukkan *hadits* lengkap (ada *sanad* dan *matan*) dalam bahasa indonesia pada kolom, lalu klik tombol proses analisa *hadits* dan sistem melakukan validasi apabila proses analisa berhasil maka sistem akan menampilkan *syarah hadits* sesuai *database* apabila proses gagal karena tidak memasukkan salah satu struktur *hadits* (*sanad* atau *matan*) maka *user* harus memasukkan *hadits* kembali secara lengkap.

### 4.2.2.3 Sequence Diagram

Proses yang digambarkan dengan *sequence diagram* adalah sebagai berikut:

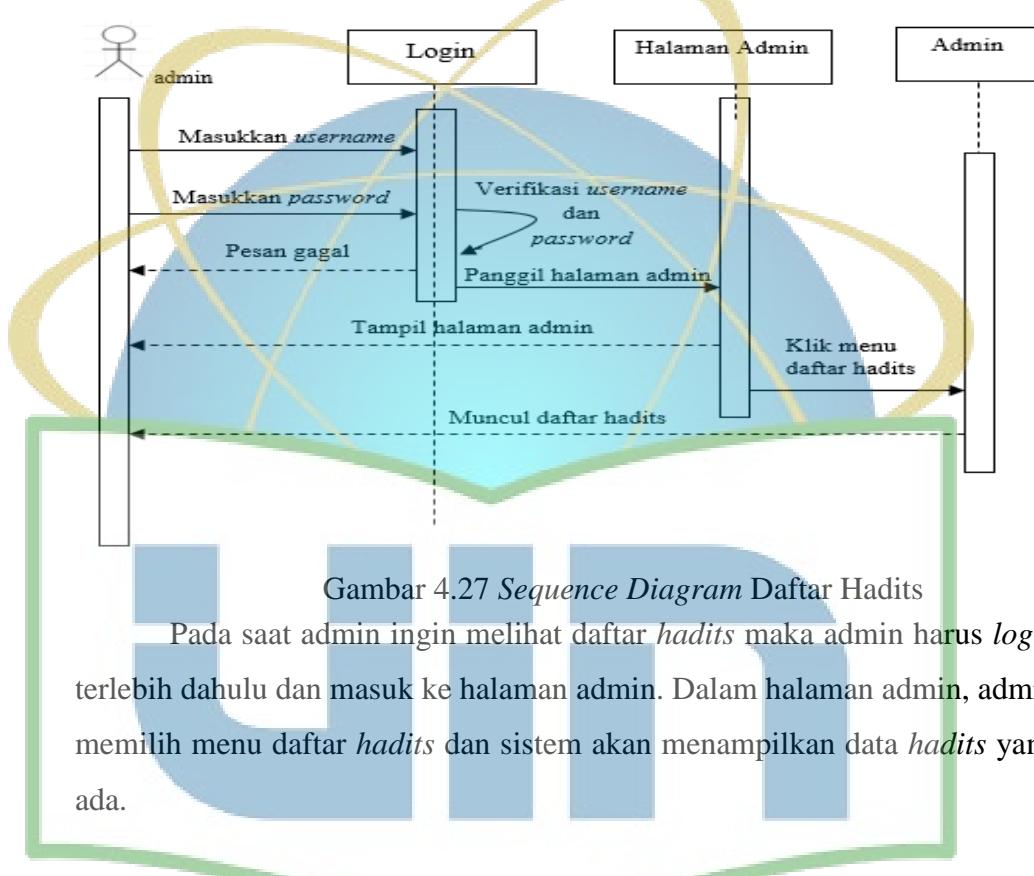
#### 1. Sequence Diagram Login



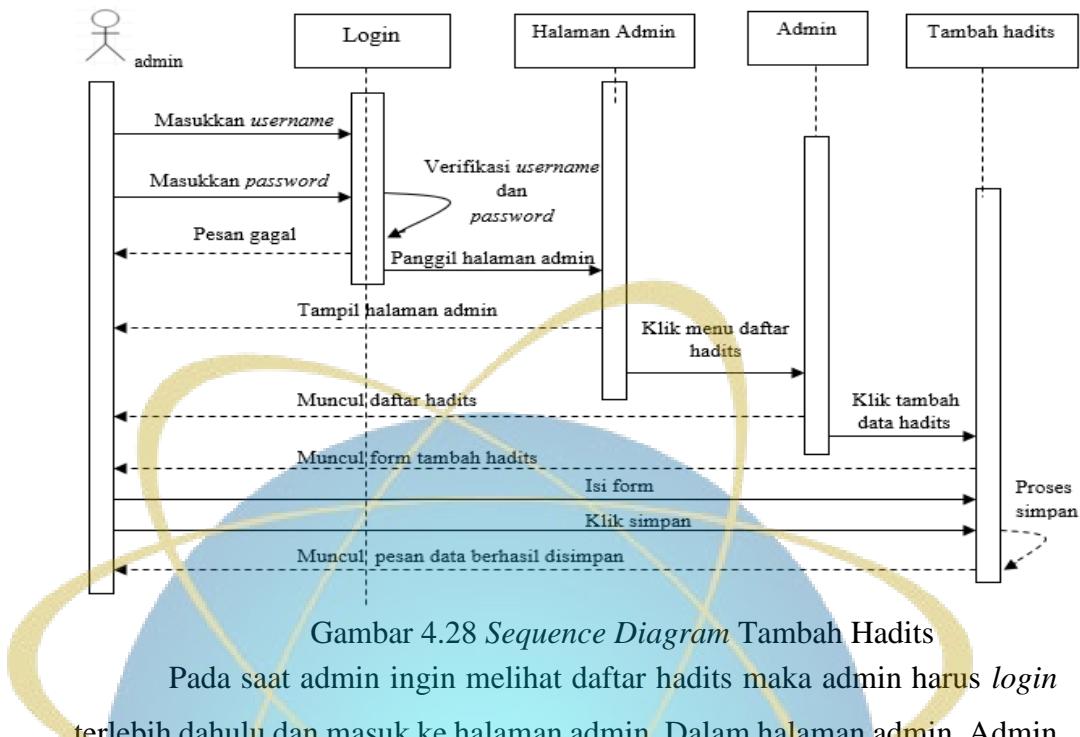
Gambar 4.26 Sequence Diagram Login

Pada saat admin melakukan *login*, sistem akan meminta *username* dan *password*, kemudian sistem akan memverifikasi *username* dan *password* tersebut. Apabila data valid maka admin dapat masuk kedalam sistem, apabila data tidak valid maka sistem akan menampilkan pesan kesalahan. Jika benar maka sistem akan memanggil halaman admin dan admin akan masuk ke halaman admin.

## 2. Sequence Diagram Daftar Hadits



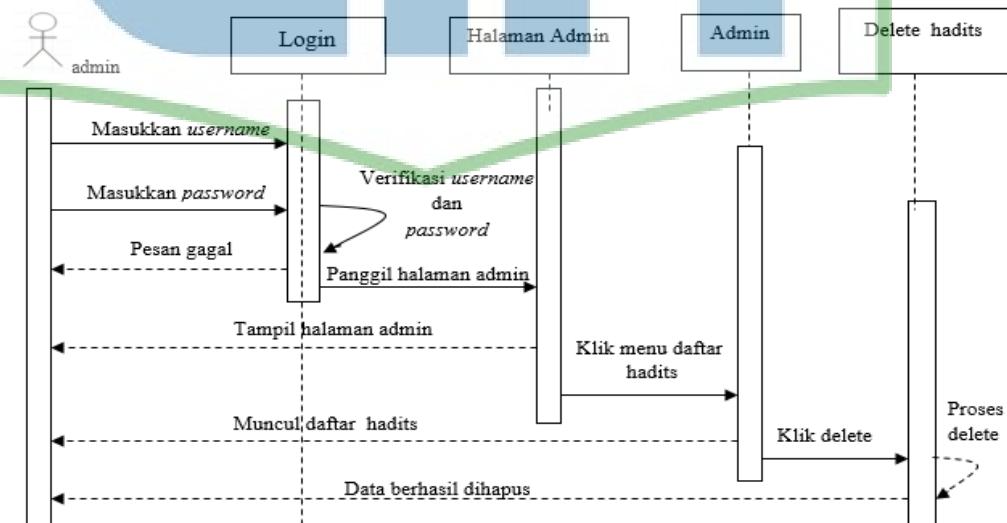
### 3. Sequence Diagram Tambah Hadits



Gambar 4.28 Sequence Diagram Tambah Hadits

Pada saat admin ingin melihat daftar hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar hadits, dan sistem akan menampilkan daftar hadits yang ada. Kemudian admin mengklik tombol tambah hadits, lalu sistem akan menampilkan form tambah hadits. Selanjutnya admin mengisi form yang ada, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*.

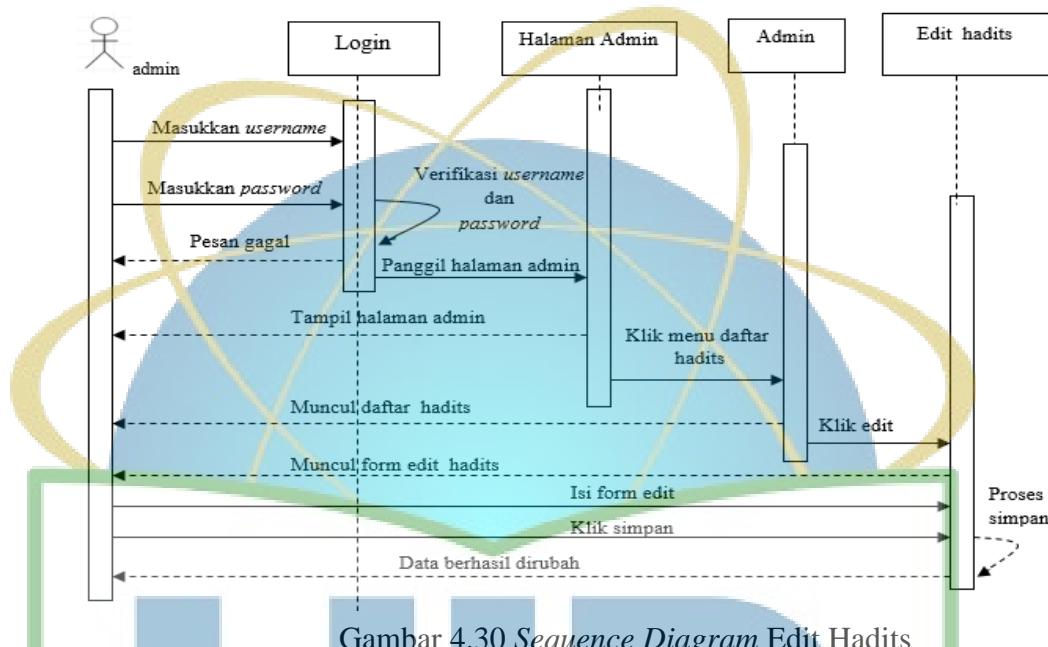
### 4. Sequence Diagram Hapus Hadits



Gambar 4.29 Sequence Diagram Hapus Hadits

Pada saat admin ingin menghapus hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar hadits, dan sistem akan menampilkan daftar hadits yang ada. Kemudian admin mengklik tombol delete, lalu sistem akan menghapus data yang dihapus.

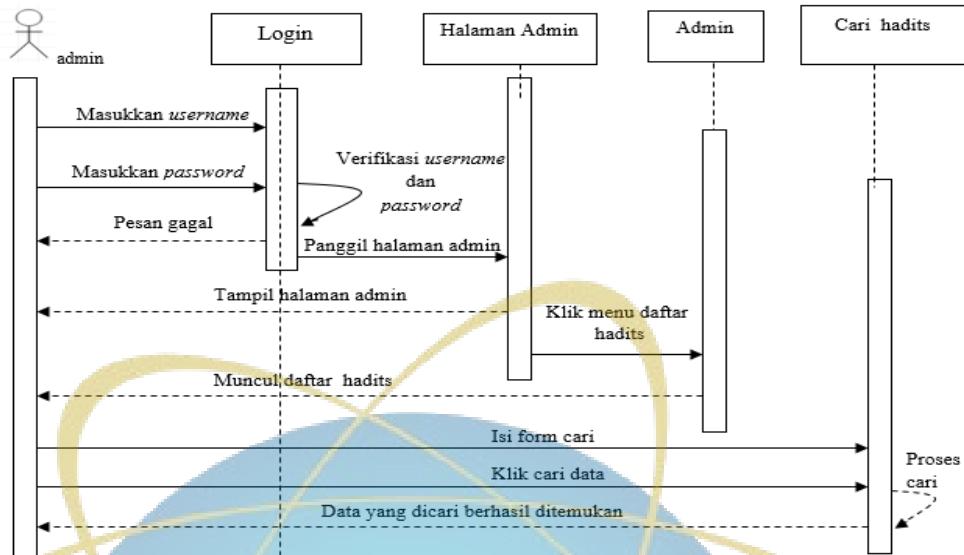
### 5. Sequence Diagram Edit Hadits



Gambar 4.30 Sequence Diagram Edit Hadits

Pada saat admin ingin mengedit hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar hadits dan sistem menampilkan daftar hadits. Lalu admin mengklik tombol edit yang ada, kemudian sistem akan menampilkan form edit data hadits, lalu admin mengedit data dengan mengisi form edit, kemudian klik simpan dan sistem akan menyimpan data dalam *database*.

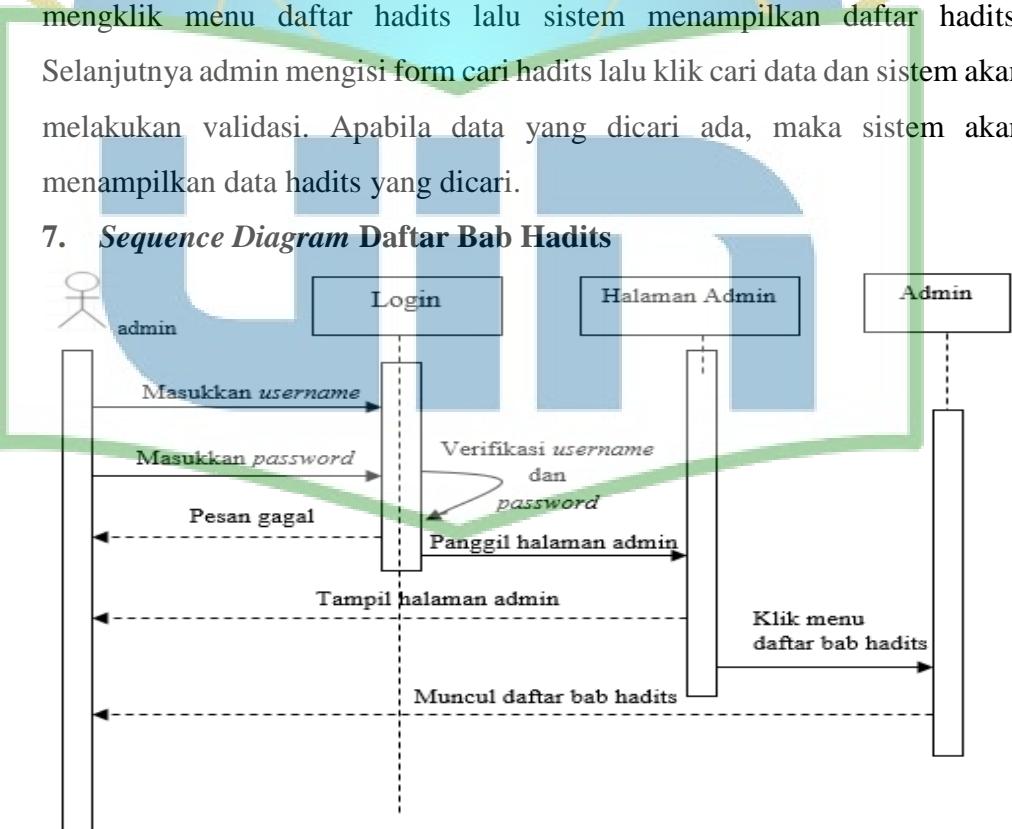
## 6. Sequence Diagram Cari Hadits



Gambar 4.31 Activity Diagram Cari Hadits

Pada saat admin ingin mencari hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, admin mengklik menu daftar hadits lalu sistem menampilkan daftar hadits. Selanjutnya admin mengisi form cari hadits lalu klik cari data dan sistem akan melakukan validasi. Apabila data yang dicari ada, maka sistem akan menampilkan data hadits yang dicari.

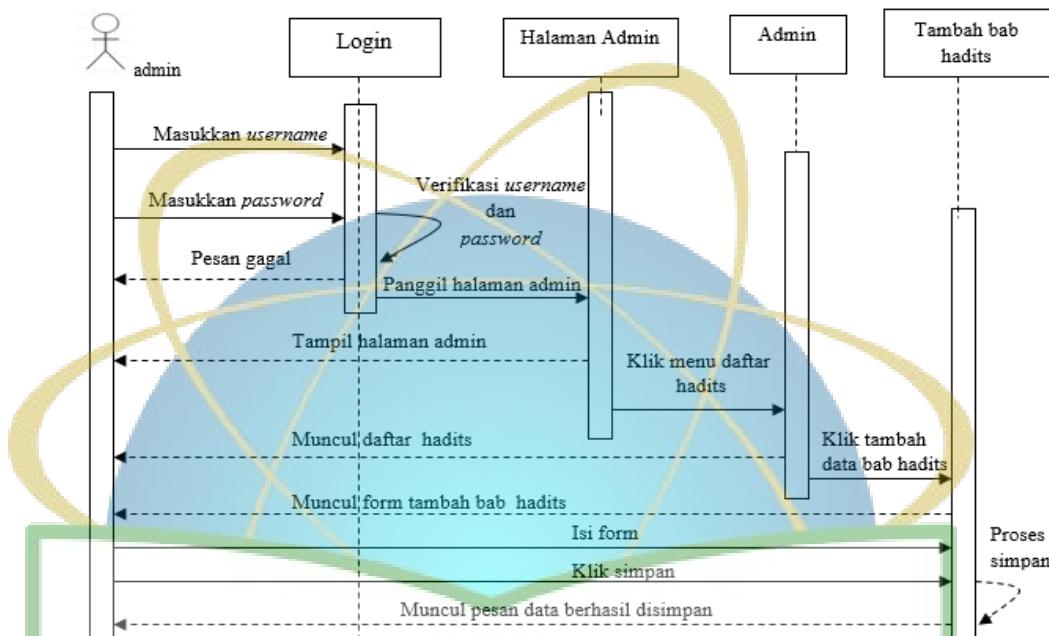
## 7. Sequence Diagram Daftar Bab Hadits



Gambar 4.32 Sequence Diagram Daftar Bab Hadits

Pada saat admin ingin melihat daftar bab hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, admin memilih menu daftar bab hadits dan sistem akan menampilkan data bab hadits yang ada.

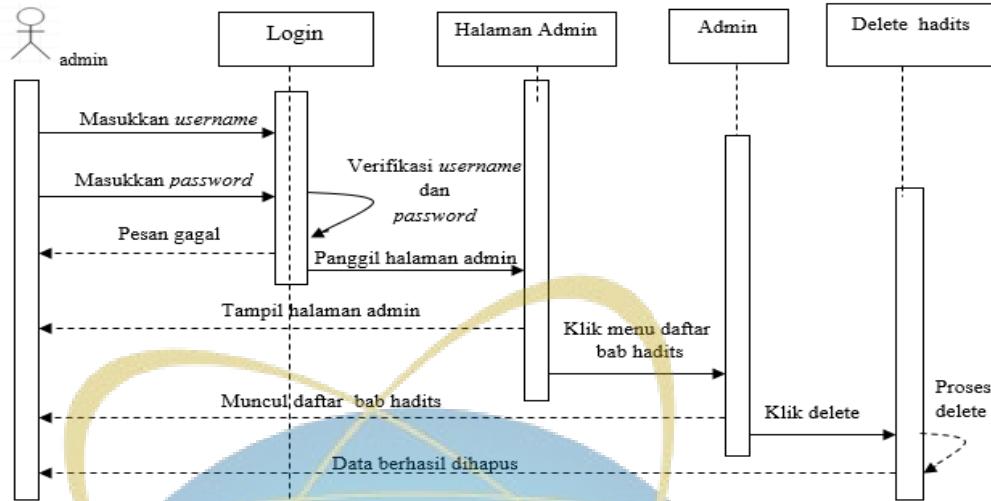
### 8. Sequence Diagram Tambah Bab Hadits



Gambar 4.33 Sequence Diagram Tambah Bab Hadits

Pada saat admin ingin melihat daftar bab hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar bab hadits, dan sistem akan menampilkan daftar bab hadits yang ada. Kemudian admin mengklik tombol tambah bab hadits, lalu sistem akan menampilkan form tambah bab hadits. Selanjutnya admin mengisi form yang ada, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*.

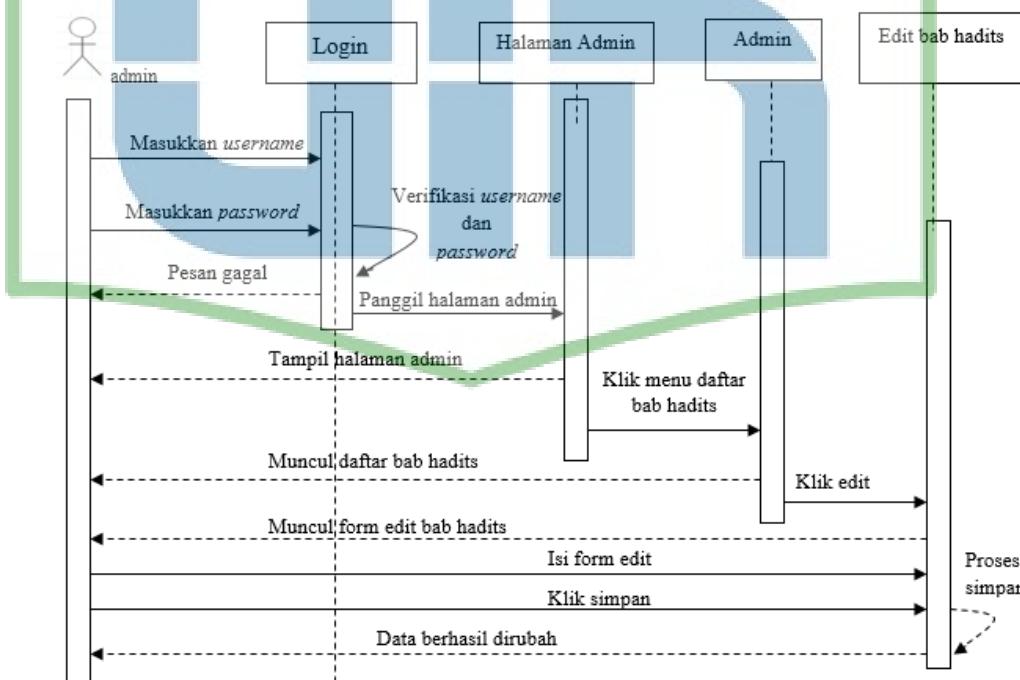
### 9. Sequence Diagram Hapus Bab Hadits



Gambar 4.34 Sequence Diagram Hapus Bab Hadits

Pada saat admin ingin menghapus bab hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar bab hadits, dan sistem akan menampilkan daftar bab hadits yang ada. Kemudian admin mengklik tombol delete, lalu sistem akan menghapus data yang dihapus.

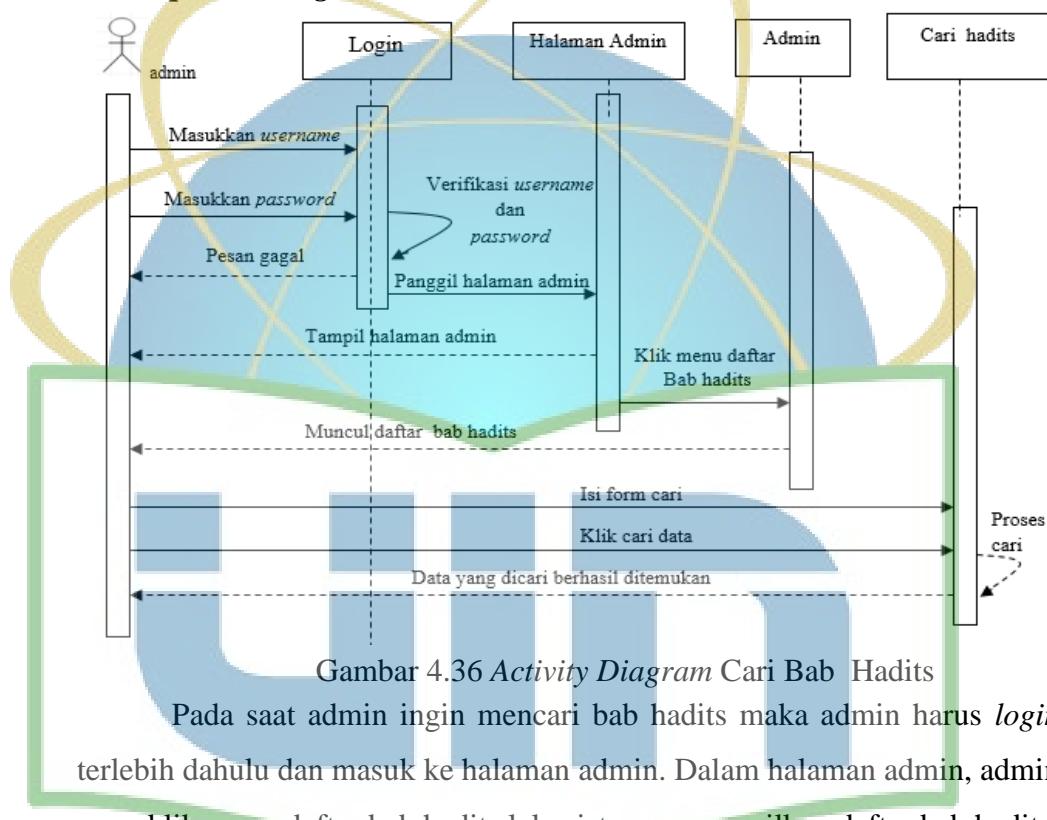
### 10. Sequence Diagram Edit Bab Hadits



Gambar 4.35 Sequence Diagram Edit Bab Hadits

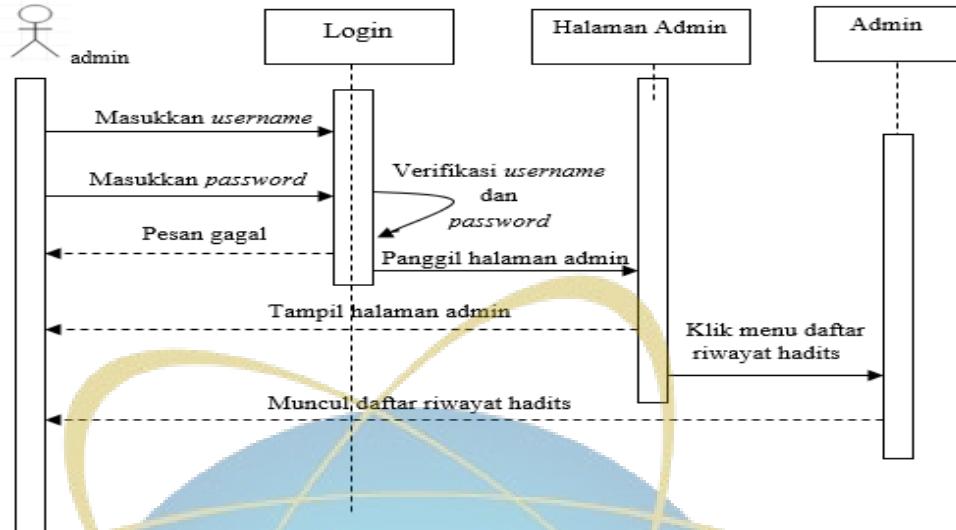
Pada saat admin ingin mengedit bab hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar bab hadits dan sistem menampilkan daftar bab hadits. Lalu admin mengklik tombol edit yang ada, kemudian sistem akan menampilkan form edit data bab hadits, lalu admin mengedit data dengan mengisi form edit, kemudian klik simpan dan sistem akan menyimpan data dalam *database*.

### 11. Sequence Diagram Cari Bab Hadits



Pada saat admin ingin mencari bab hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, admin mengklik menu daftar bab hadits lalu sistem menampilkan daftar bab hadits. Selanjutnya admin mengisi form cari bab hadits lalu klik cari data dan sistem akan melakukan validasi. Apabila data yang dicari ada, maka sistem akan menampilkan data bab hadits yang dicari.

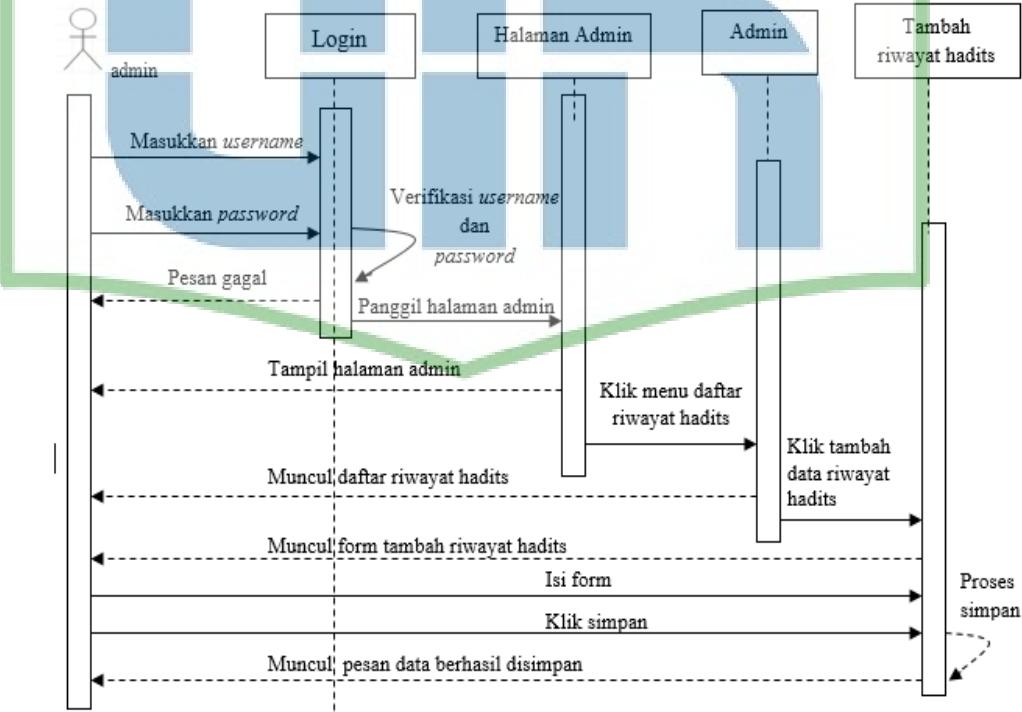
## 12. Sequence Diagram Daftar Riwayat Hadits



Gambar 4.37 Sequence Diagram Daftar Riwayat Hadits

Pada saat admin ingin melihat daftar riwayat hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, admin memilih menu daftar riwayat hadits dan sistem akan menampilkan data riwayat hadits yang ada.

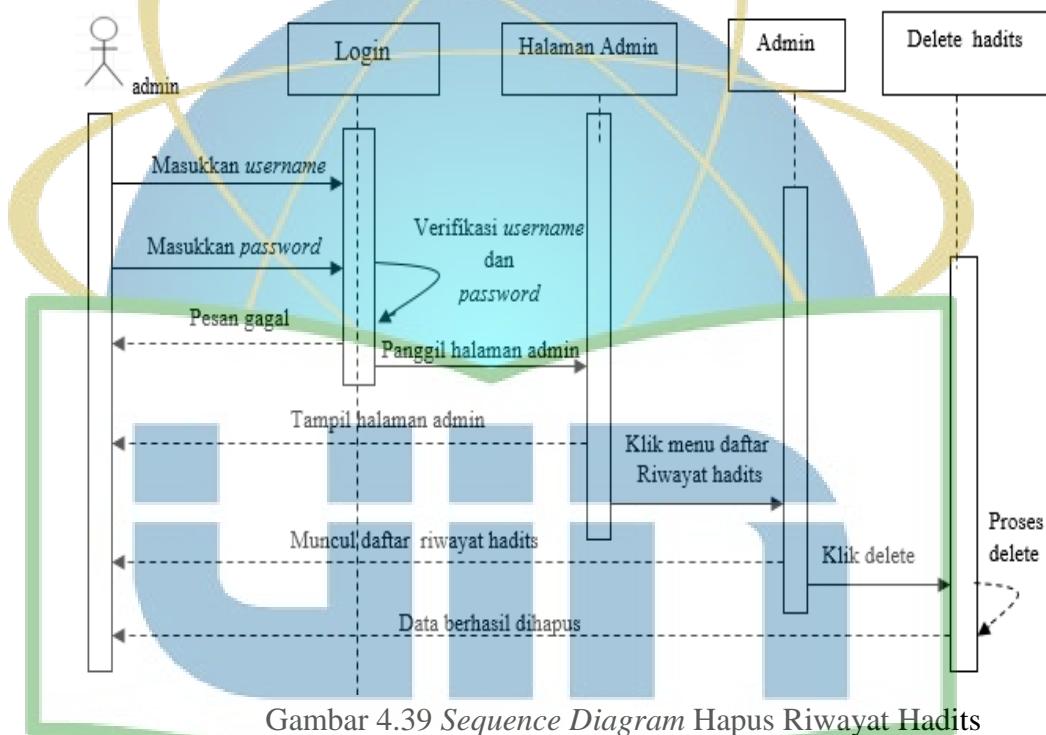
## 13. Sequence Diagram Tambah Riwayat Hadits



Gambar 4.38 Sequence Diagram Tambah Riwayat Hadits

Pada saat admin ingin melihat daftar riwayat hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar riwayat hadits, dan sistem akan menampilkan daftar riwayat hadits yang ada. Kemudian admin mengklik tombol tambah riwayat hadits, lalu sistem akan menampilkan form tambah riwayat hadits. Selanjutnya admin mengisi form yang ada, kemudian klik simpan data. Sistem akan melakukan validasi apabila data yang dimasukan berhasil maka akan tersimpan dalam *database*.

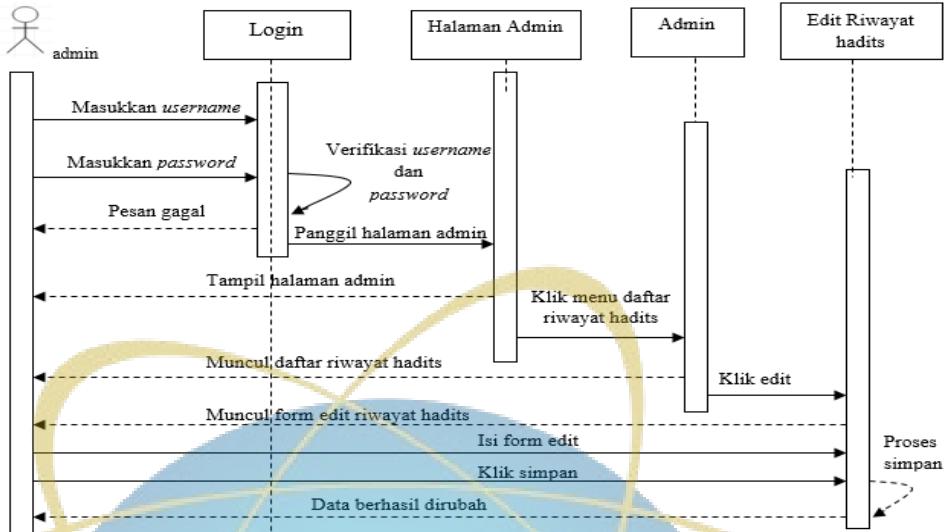
#### 14. Sequence Diagram Hapus Riwayat Hadits



Gambar 4.39 Sequence Diagram Hapus Riwayat Hadits

Pada saat admin ingin menghapus riwayat hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar riwayat hadits, dan sistem akan menampilkan daftar riwayat hadits yang ada. Kemudian admin mengklik tombol *delete*, lalu sistem akan menghapus data yang dihapus.

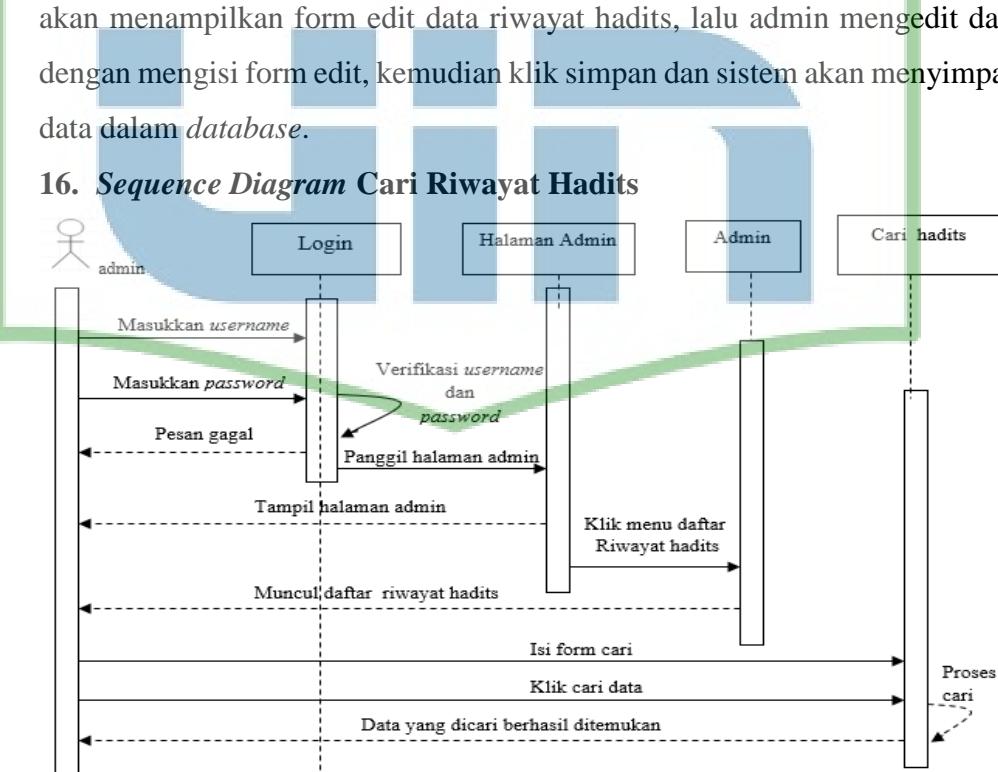
### 15. Sequence Diagram Edit Riwayat Hadits



Gambar 4.40 Sequence Diagram Edit Riwayat Hadits

Pada saat admin ingin mengedit riwayat hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, Admin mengklik menu daftar riwayat hadits dan sistem menampilkan daftar riwayat hadits. Lalu admin mengklik tombol edit yang ada, kemudian sistem akan menampilkan form edit data riwayat hadits, lalu admin mengedit data dengan mengisi form edit, kemudian klik simpan dan sistem akan menyimpan data dalam *database*.

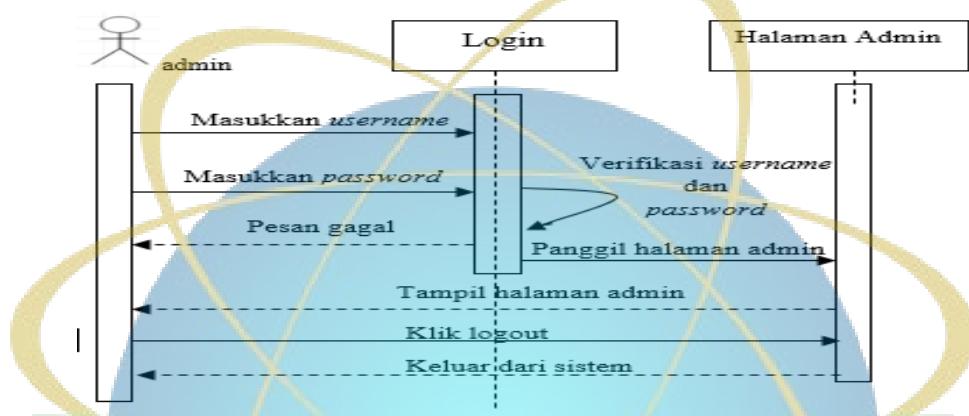
### 16. Sequence Diagram Cari Riwayat Hadits



Gambar 4.41 Activity Diagram Cari Riwayat Hadits

Pada saat admin ingin mencari riwayat hadits maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, admin mengklik menu daftar riwayat hadits lalu sistem menampilkan daftar riwayat hadits. Selanjutnya admin mengisi form cari riwayat hadits lalu klik cari data dan sistem akan melakukan validasi. Apabila data yang dicari ada, maka sistem akan menampilkan data riwayat hadits yang dicari.

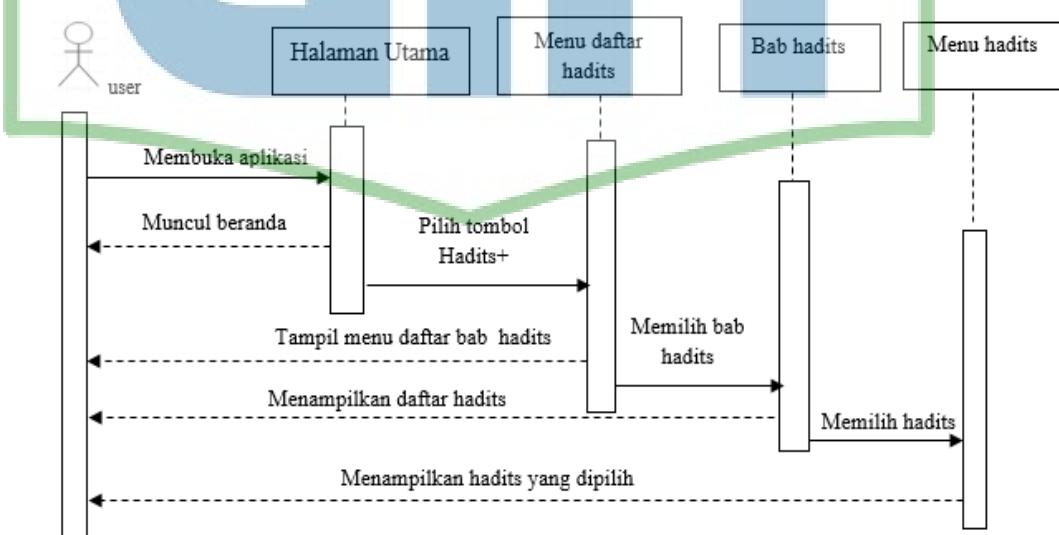
### 17. Sequence Diagram Logout



Gambar 4.42 Sequence Diagram Logout

Pada saat admin ingin melakukan *logout* maka admin harus *login* terlebih dahulu dan masuk ke halaman admin. Dalam halaman admin, admin memilih perintah *logout*, kemudian sistem akan melakukan proses *logout* dan kembali ke menu *login*.

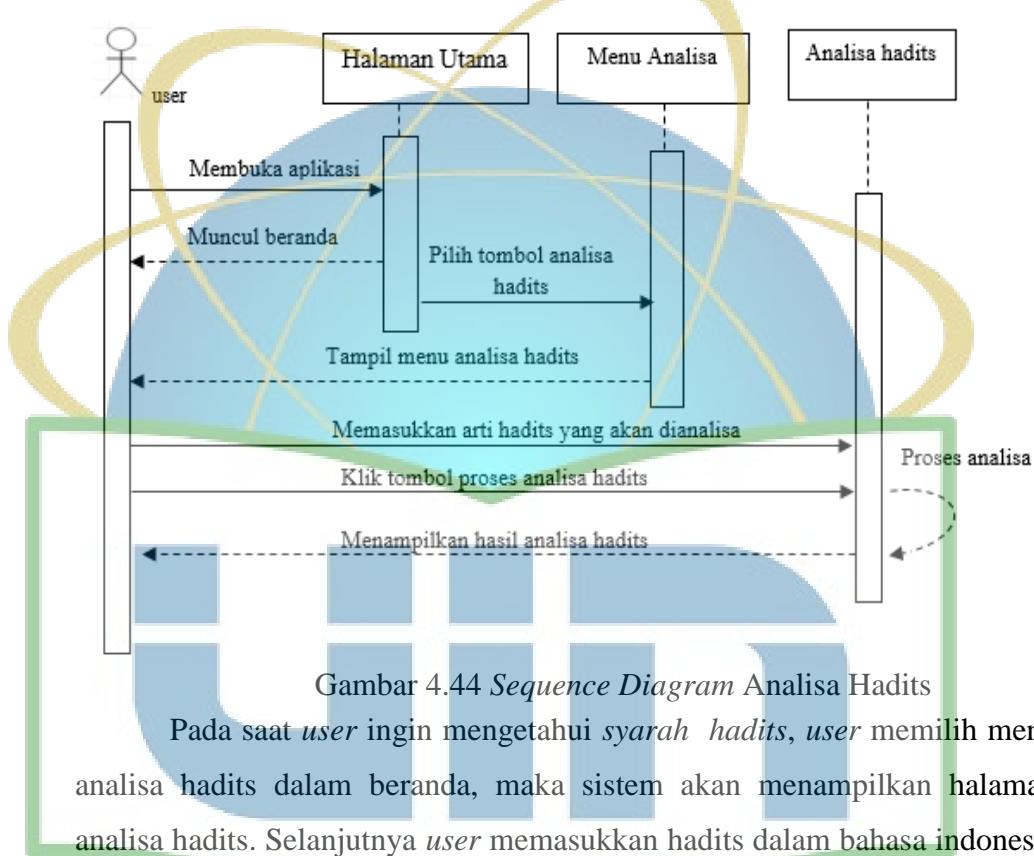
### 18. Sequence Diagram Baca Hadits



Gambar 4.43 Sequence Diagram Baca Hadits

Pada saat *user* ingin membaca hadits, *user* memilih menu hadits yang terdapat dalam beranda, lalu sistem akan menampilkan menu daftar bab hadits. Selanjutnya *user* memilih bab hadits, kemudian sistem akan menampilkan daftar hadits sesuai dengan bab yang telah *user* pilih. Selanjutnya, *user* memilih hadits yang akan dibaca, dan sistem menampilkan hadits yang telah dipilih *user* tersebut.

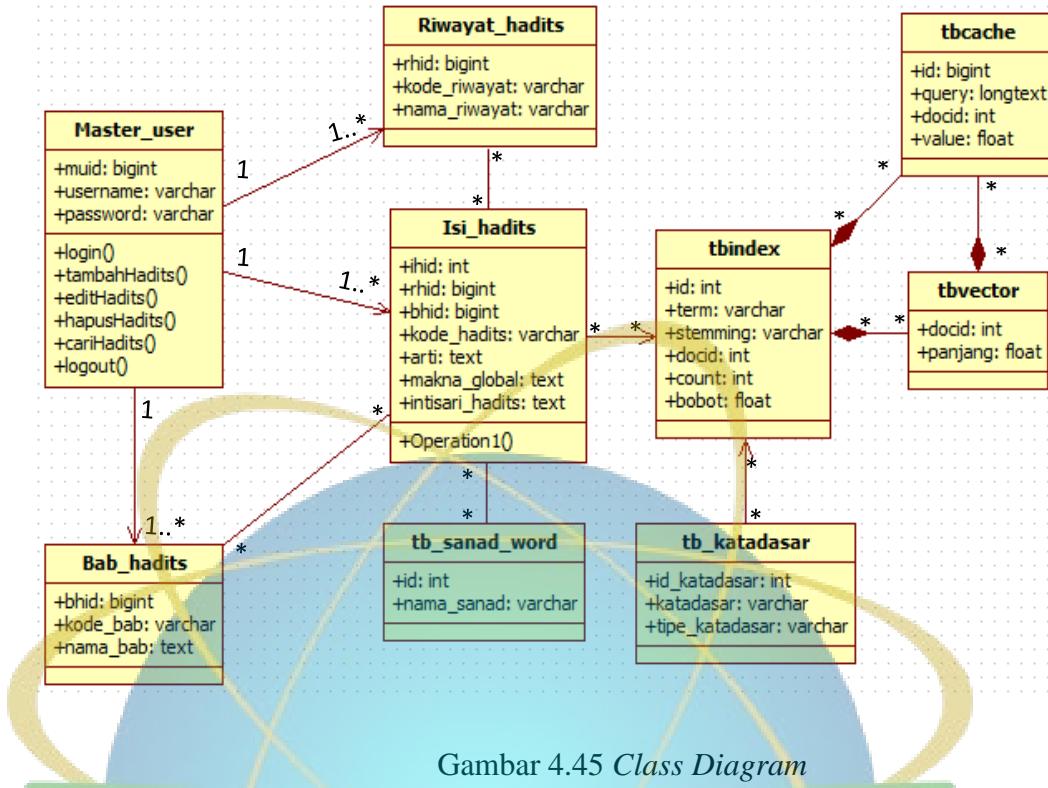
### 19. Sequence Diagram Analisa Hadits



Pada saat *user* ingin mengetahui *syarah hadits*, *user* memilih menu analisa hadits dalam beranda, maka sistem akan menampilkan halaman analisa hadits. Selanjutnya *user* memasukkan hadits dalam bahasa indonesia dengan lengkap (*sanad* beserta *matan*) dalam kolom, lalu klik tombol proses analisa hadits, kemudian sistem menampilkan *syarah* dari *hadits* yang telah di *input*-kan, apabila *user* memasukkan *hadits* tidak lengkap, dalam artian tidak mengandung dari salah satu struktur *hadits* (*sanad* atau *matan*) maka sistem tidak dapat memberikan *output* dan akan memberikan pesan untuk memasukkan *hadits* dengan benar.

#### 4.2.2.4 Class Diagram

*Class diagram* yang penulis buat dalam pembuatan sistem ini adalah sebagai berikut:



Gambar 4.45 Class Diagram

Pada gambar diatas menjelaskan sebuah *class diagram* yang menggambarkan hubungan atau relasi antara tiap kelas dalam sistem ini yang terbentuk.

#### 4.2.3 Merancang Database

Merancang beberapa tabel dalam satu *database* pusat berguna sebagai media penyimpanan data informasi sistem sehingga dapat membantu dalam proses analisa hadits dan proses lainnya yang menunjang. Berikut ini merupakan penjelasan lengkap mengenai tabel-tabel dan desain *database* yang telah penulis buat:

##### 4.2.3.1 Daftar Tabel

Daftar tabel yang tedapat dalam hadits.sql pada sistem ini adalah sebagai berikut:

Tabel 4.8 Daftar Tabel dalam Database

Nama Tabel	Primary Key	Data Store
master_user	moid	Hadits.sql
isi_hadits	ihid	Hadits.sql
bab_hadits	bhid	Hadits.sql
riwayat_hadits	rhid	Hadits.sql
tbcache	id	Hadits.sql
tbindex	id	Hadits.sql
tbvektor	docid	Hadits.sql
tb_katadasar	id_katadasar	Hadits.sql
tb_sanad_word	id	Hadits.sql

#### 4.2.3.2 Deskripsi Tabel

Deskripsi tabel dari daftar tabel diatas adalah sebagai berikut:

##### 1. Tabel master\_user

Tabel 4.9 Tabel master\_user

Nama Field	Tipe Data	Ukuran
moid	bigint	20
Username	varchar	50
Password	varchar	50

##### 2. Tabel isi\_hadits

Tabel 4.10 Tabel isi\_hadits

Nama Field	Tipe Data	Ukuran
ihid	int	11
rhid	bigint	20
bhid	bigint	20
kode_hadits	varchar	50
arti	text	-
makna_global	text	-
intisari_hadits	text	-

##### 3. Tabel bab\_hadits

Tabel 4.11 Tabel bab\_hadits

Nama Field	Tipe Data	Ukuran
bhid	bigint	20
kode_bab	varchar	50
nama_bab	text	-

#### 4. Tabel riwayat\_hadits

Tabel 4.12 Tabel riwayat\_hadits

Nama Field	Tipe Data	Ukuran
rhid	<i>bigint</i>	20
kode_riwayat	<i>varchar</i>	50
nama_riwayat	<i>varchar</i>	50

#### 5. Tabel tbcache

Tabel 4.13 Tabel tbcache

Nama Field	Tipe Data	Ukuran
id	<i>bigint</i>	20
query	<i>longtext</i>	-
docid	<i>int</i>	11
value	<i>float</i>	-

#### 6. Tabel tbindex

Tabel 4.14 Tabel tbindex

Nama Field	Tipe Data	Ukuran
id	<i>int</i>	11
term	<i>varchar</i>	50
stemming	<i>varchar</i>	50
docid	<i>int</i>	11
count	<i>int</i>	11
bobot	<i>float</i>	-

#### 7. Tabel tbvektor

Tabel 4.15 Tabel tbvektor

Nama Field	Tipe Data	Ukuran
docid	<i>int</i>	11
panjang	<i>float</i>	-

#### 8. Tabel tb\_katadasar

Tabel 4.16 Tabel tb\_katadasar

Nama Field	Tipe Data	Ukuran
Id_katadasar	<i>int</i>	10
katadasar	<i>varchar</i>	50
Tipe_katadasar	<i>varchar</i>	50

#### 9. Tabel tb\_sanad\_word

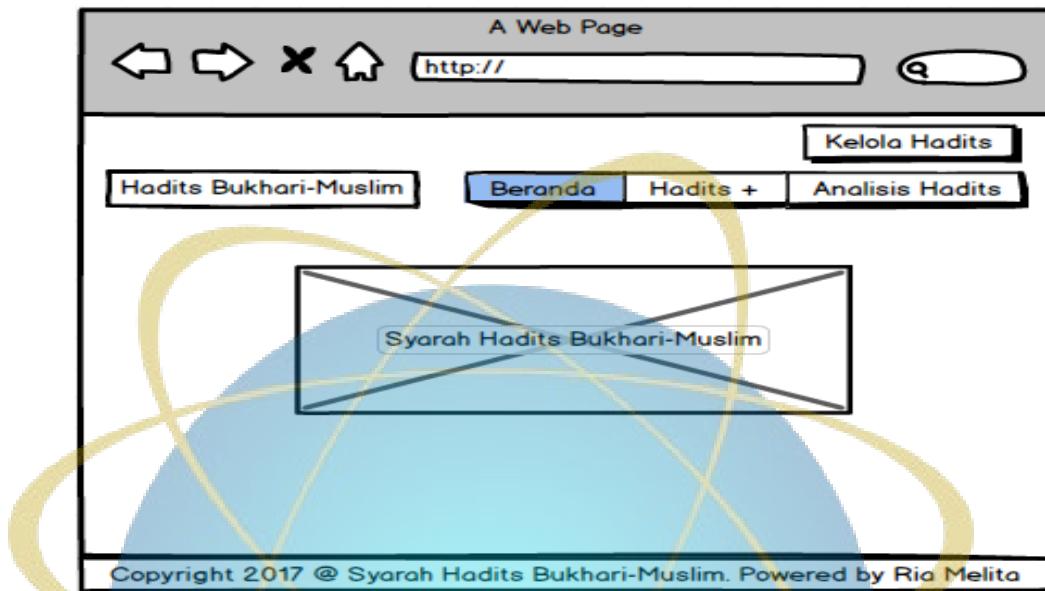
Tabel 4.17 Tabel tb\_sanad\_word

Nama Field	Tipe Data	Ukuran
Id	<i>int</i>	11
Nama_sanad	<i>varchar</i>	50

#### 4.2.4 Merancang *User Interface*

Pada tahap ini, penulis melakukan perancangan *user interface* untuk sistem ini sebagai berikut:

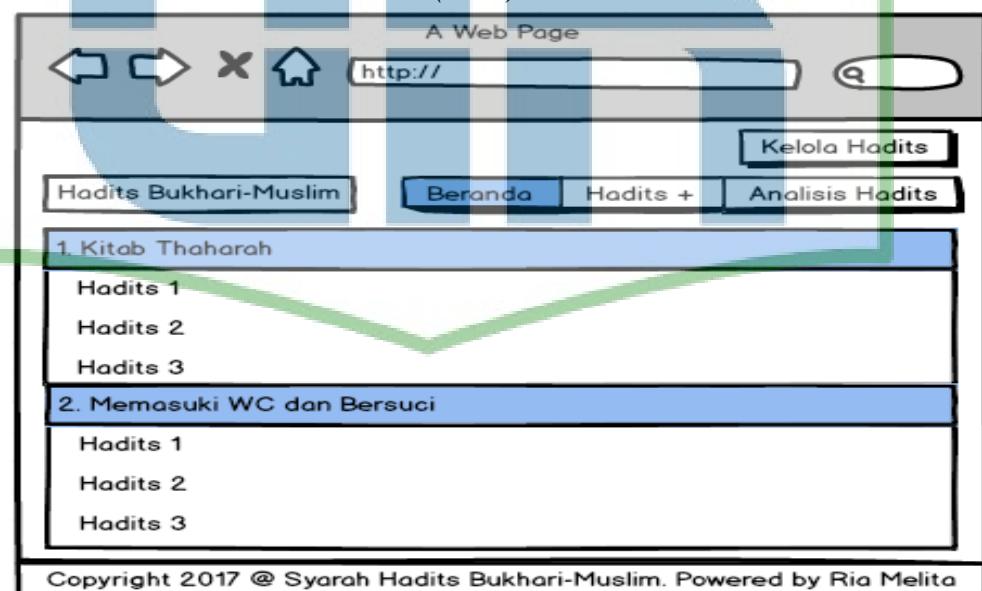
##### 4.2.4.1 Halaman Utama



Gambar 4.46 *User Interface* Halaman Utama

Gambar diatas merupakan desain dari halaman utama yang muncul ketika pertama kali membuka aplikasi.

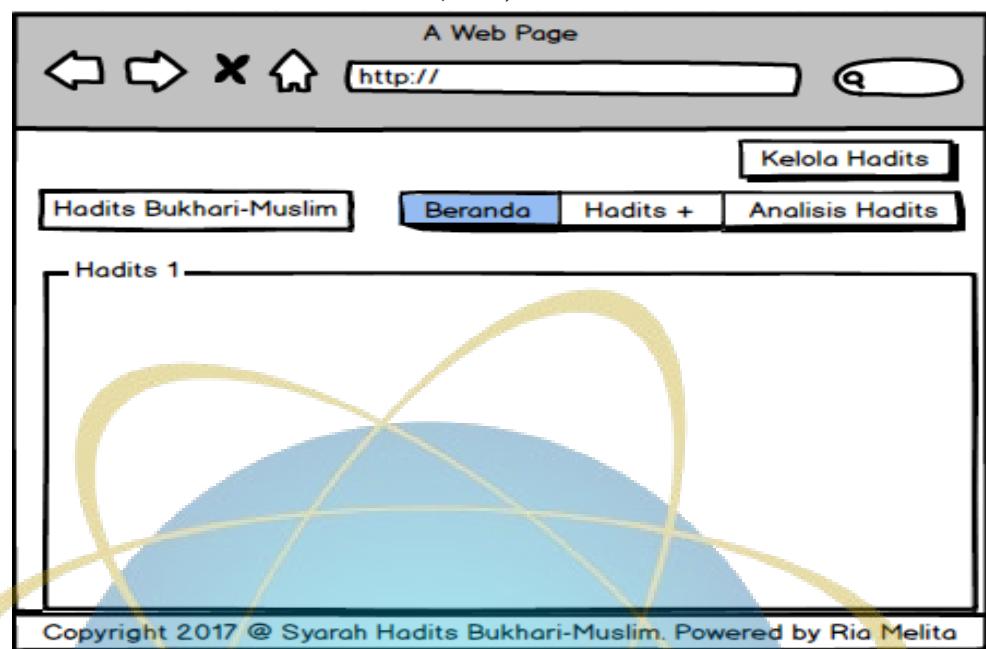
##### 4.2.4.2 Halaman Baca Hadits (*User*)



Gambar 4.47 *User Interface* Baca Hadits (*user*)

Gambar diatas merupakan desain dari halaman baca hadits yang muncul ketika *user* mengklik button ‘hadits’.

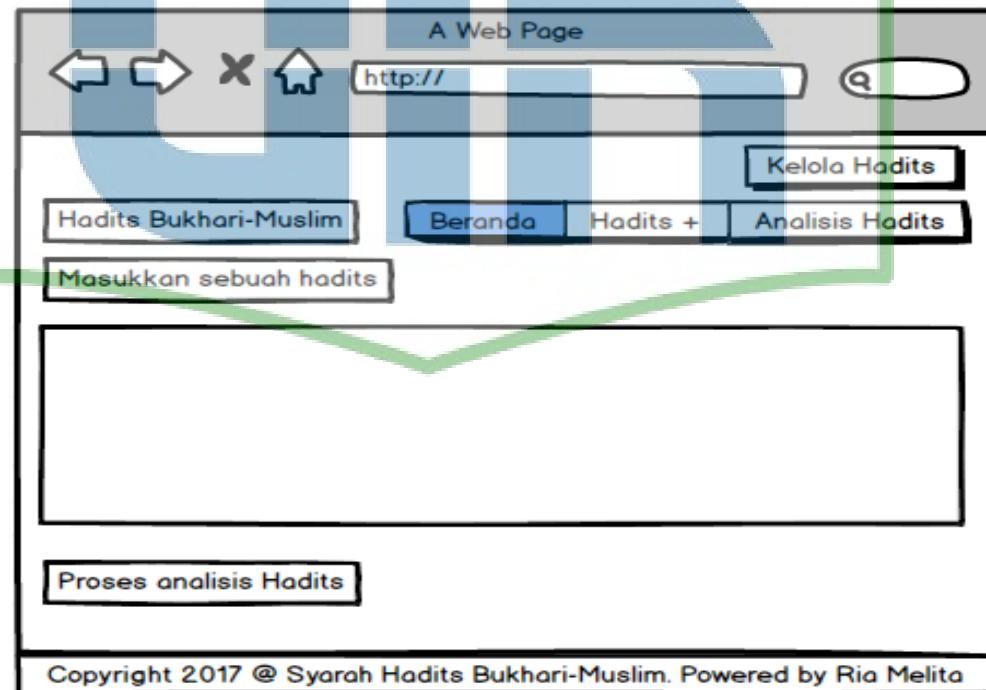
#### 4.2.4.3 Halaman Baca Hadits (*User*)



Gambar 4.48 *User Interface* Halaman Baca Hadits (*user*)

Gambar diatas merupakan desain dari halaman baca hadits yang muncul ketika *user* mengklik salah satu dari daftar hadits dari bab hadits yang tersedia.

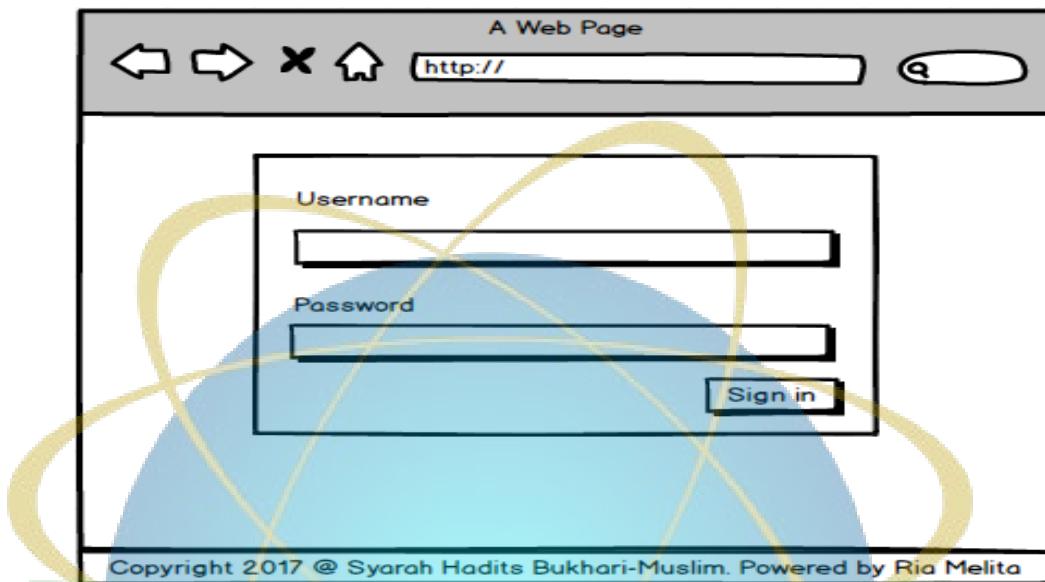
#### 4.2.4.4 Halaman Analisa Hadits (*User*)



Gambar 4.49 *User Interface* Halaman Analisis Hadits (*User*)

Gambar diatas merupakan desain dari halaman analisa hadits yang muncul ketika *user* mengklik tombol analisa hadits untuk melakukan proses analisa hadits.

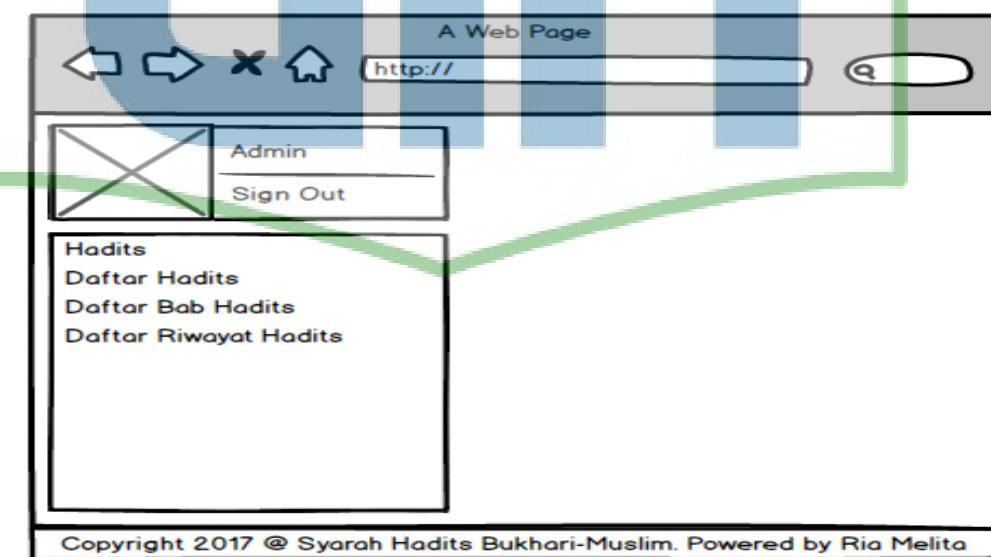
#### 4.2.4.5 Halaman *Login* (admin)



Gambar 4.50 *User Interface* Halaman *Login* (Admin)

Gambar diatas merupakan desain dari halaman *login* untuk admin yang muncul ketika admin mengklik tombol Kelola Hadits untuk melakukan tambah, edit, hapus, dan cari hadits.

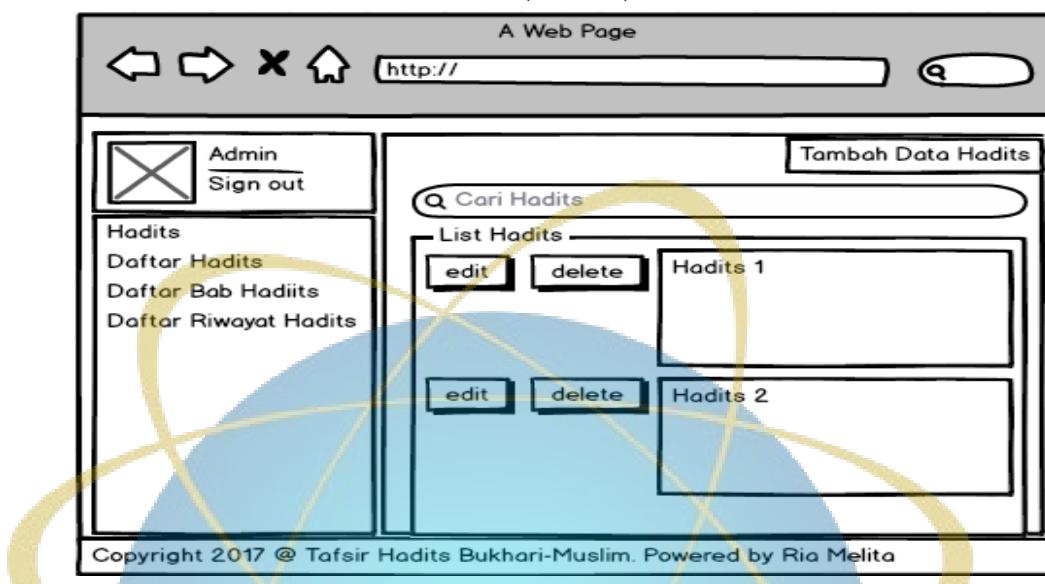
#### 4.2.4.6 Halaman Utama Admin



Gambar 4.51 *User Interface* Halaman Utama Admin

Gambar diatas merupakan desain dari halaman utama admin yang muncul ketika admin telah berhasil melakukan *login*.

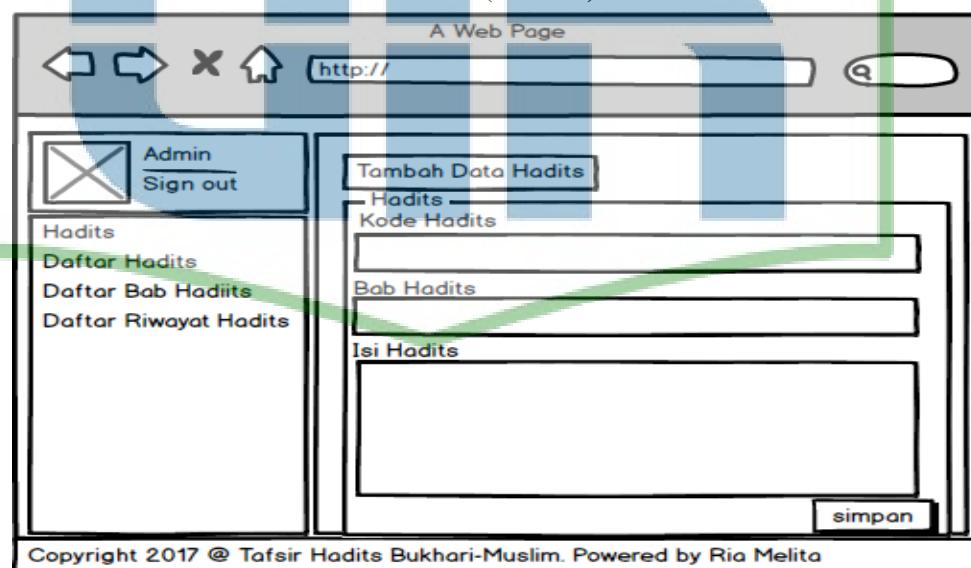
#### 4.2.4.7 Halaman Daftar Hadits (Admin)



Gambar 4.52 *User Interface* Halaman Daftar Hadits

Gambar diatas merupakan desain dari halaman daftar hadits untuk melakukan tambah data, edit dan hapus data hadits yang muncul ketika admin mengklik tombol Daftar Hadits pada menu admin.

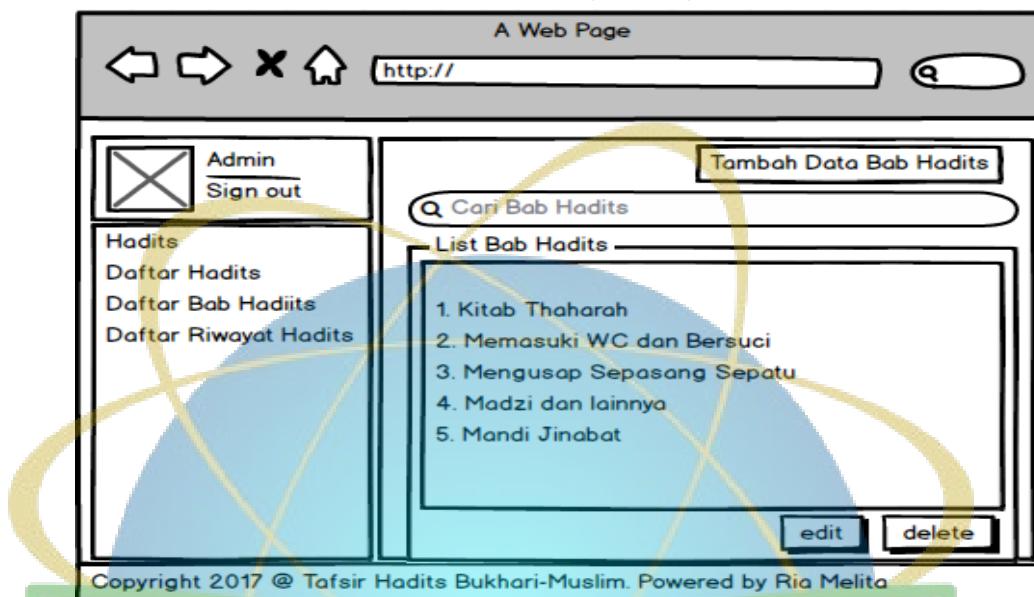
#### 4.2.4.8 Halaman Tambah Hadits (Admin)



Gambar 4.53 *User Interface* Halaman Tambah Hadits

Gambar diatas merupakan desain dari halaman tambah hadits yang muncul ketika admin mengklik tombol Tambah Data Hadits pada halaman daftar hadits.

#### 4.2.4.9 Halaman Daftar Bab Hadits (Admin)



Gambar 4.54 *User Interface* Halaman Bab Hadits

Gambar diatas merupakan desain dari halaman bab hadits untuk melakukan tambah data, edit dan hapus data bab hadits yang muncul ketika admin mengklik tombol Daftar Bab Hadits pada menu admin.

#### 4.2.4.10 Halaman Tambah Bab Hadits (Admin)

A screenshot of a web browser showing a form for adding a new 'Bab Hadits'. The form includes the following fields:

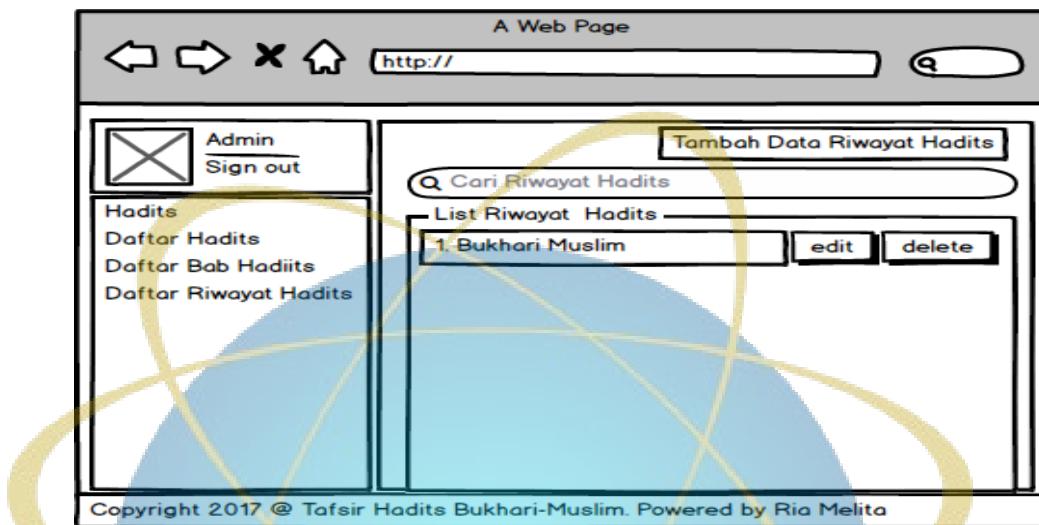
- Bab Hadits
- Kode Bab Hadits: An input field containing a placeholder value.
- Nama Bab Hadits: An input field containing a placeholder value.
- A 'simpan' button located at the bottom right of the form area.

The entire form area is highlighted with a green box, indicating it's the target of a user action.

Gambar 4.55 *User Interface* Halaman Tambah Bab Hadits

Gambar diatas merupakan desain dari halaman tambah bab hadits yang muncul ketika admin mengklik tombol Tambah Data Bab Hadits pada menu Daftar Bab Hadits.

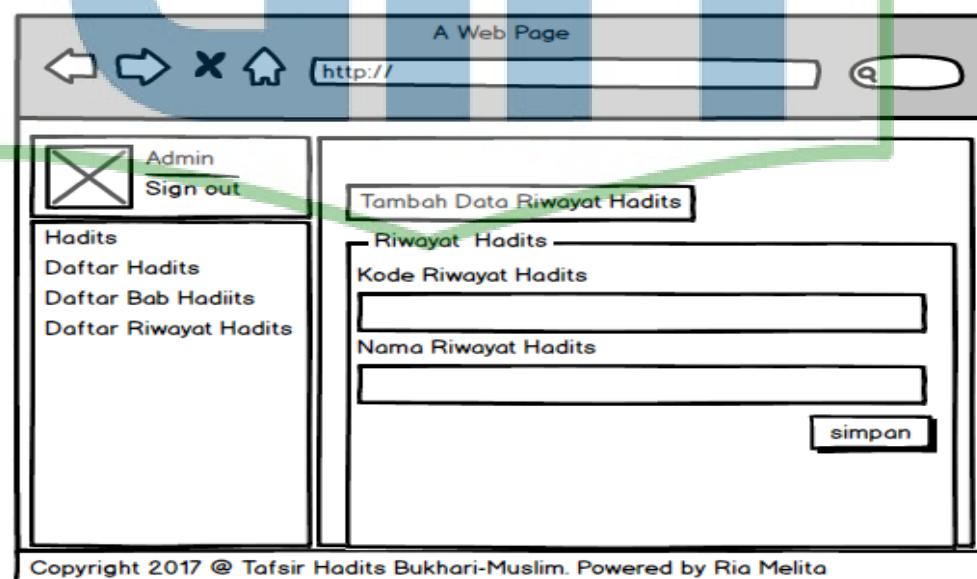
#### 4.2.4.11 Halaman Daftar Riwayat Hadits (Admin)



Gambar 4.56 *User Interface* Halaman Daftar Riwayat Hadits

Gambar diatas merupakan desain dari halaman daftar riwayat hadits untuk melakukan tambah data, edit dan hapus data riwayat hadits yang muncul ketika admin mengklik tombol Daftar Riwayat Hadits pada menu admin.

#### 4.2.4.12 Halaman Tambah Riwayat Hadits (Admin)



Gambar 4.57 *User Interface* Halaman Tambah Riwayat Hadits

Gambar diatas merupakan desain dari halaman tambah riwayat hadits yang muncul ketika admin mengklik tombol Tambah Data Riwayat Hadits pada menu Daftar Riwayat Hadits.

#### 4.2.5 Pengkodean (*Coding*)

Pada tahap pengkodean ini, penulis melakukan pengkodean terhadap hasil rancangan (*UserInterface*) yang sudah didefinisikan pada pembahasan sebelumnya untuk melakukan implementasi sistem sesuai dengan *user interface* yang telah dirancang. Pengkodean ini dilakukan dengan bahasa pemrograman PHP dengan *database* MySql dan menggunakan beberapa *tools* serta *software* seperti: Xampp v.3.2.2, Apache v. 2.4.29, MySql v.5.0.11, PHP v.5.6.33, PhpMyAdmin v.4.7.4, Sublime Text 3, Photoshop cc 2017, dan Google Chrome v.62.0.3. Pengkodean sistem yang telah penulis lakukan dapat dilihat pada *Pseudo Code* dibawah ini:

##### 1. Algoritma Tokenizing

*Pseudocode* algoritma *tokenizing* yang digunakan adalah dengan menggunakan *function stop\_kar(\$teks){}* sebagai berikut:

```
function stop_kar($teks) {
    $teks = preg_replace('/[^A-Za-z0-9\ ]/','', $teks); //menghilangkan karakter
    $teks = str_replace('ldquo', '', $teks);
    $teks = str_replace('rdquo', '', $teks);
    $teks = str_replace('lsquo', '', $teks);
    $teks = str_replace('rsquo', '', $teks);
    $teks = str_replace('nbsp', '', $teks);
    $teks = strtolower(trim($teks)); //mengubah ke huruf kecil
    return $teks;
}
```

Gambar 4.58 Algoritma *Tokenizing*

Gambar diatas merupakan potongan *pseudo code* algoritma *tokenizing* dalam analisa hadits yang berfungsi untuk menghilangkan tanda baca, memecah teks menjadi token, dan mengubah huruf besar menjadi huruf kecil.

##### 2. Algoritma Stopword Removal/Filtering

*Pseudocode* algoritma *stopword removal* yang digunakan adalah dengan menggunakan *function stop\_word(\$teks){}* sebagai berikut:

```

function stop_word($teks) {
    //hapus stoplist
    //daftar stop word diletakkan di array
    $astoplist = array ("yang", "juga", "dari", "dia", "kami", "kamu", "ini", "itu", "jika", "apabila",
    | "atau", "dan", "tersebut", "pada", "dengan", "adalah", "perkata", "vaitu",
    | "bingga", "ketika", "diantara", "sebuah", "sebagai", "tidak", "untuk",
    | "setiap");
    foreach ($astoplist as $i => $value) {
        $teks = str_replace($astoplist[$i], "", $teks);
    }
}

```

Gambar 4.59 Algoritma *Stopword Removal*

Gambar diatas merupakan potongan *pesudo code* algoritma *stopword removal* dalam analisa *hadits* yang berfungsi untuk menghilangkan kata yang dianggap tidak penting (*stoplist*).

### 3. Algoritma *Stemming*

*Pseudocode* algoritma *stemming* yang digunakan dalam *Stemming Nazief dan Adriani* adalah sebagai berikut:

```

//terapkan stemming
//fungsi untuk mengecek kata dalam tabel kata dasar
function cekKamus($kata){
    $result = $this->konten_mdl->model_list_kata_dasar($kata);
    return $result;
    if($result==1){
        return true; // True jika ada
    }else{
        return false; // jika tidak ada FALSE
    }
}
//fungsi untuk menghapus suffix seperti -ku, -mu, -kah, dsb
function Del_Inflection_Suffixes($kata){
    $kataAsal = $kata;
    if($this->cekKamus($kata)){
        if(preg_match('/([km]u|nya|[kl]ah|pun)\z/i', $kata)) { // Cek Inflection Suffixes
            $kata = preg_replace('/([km]u|nya|[kl]ah|pun)\z/i', '', $kata);
        }
    }
    return $kataAsal;
}

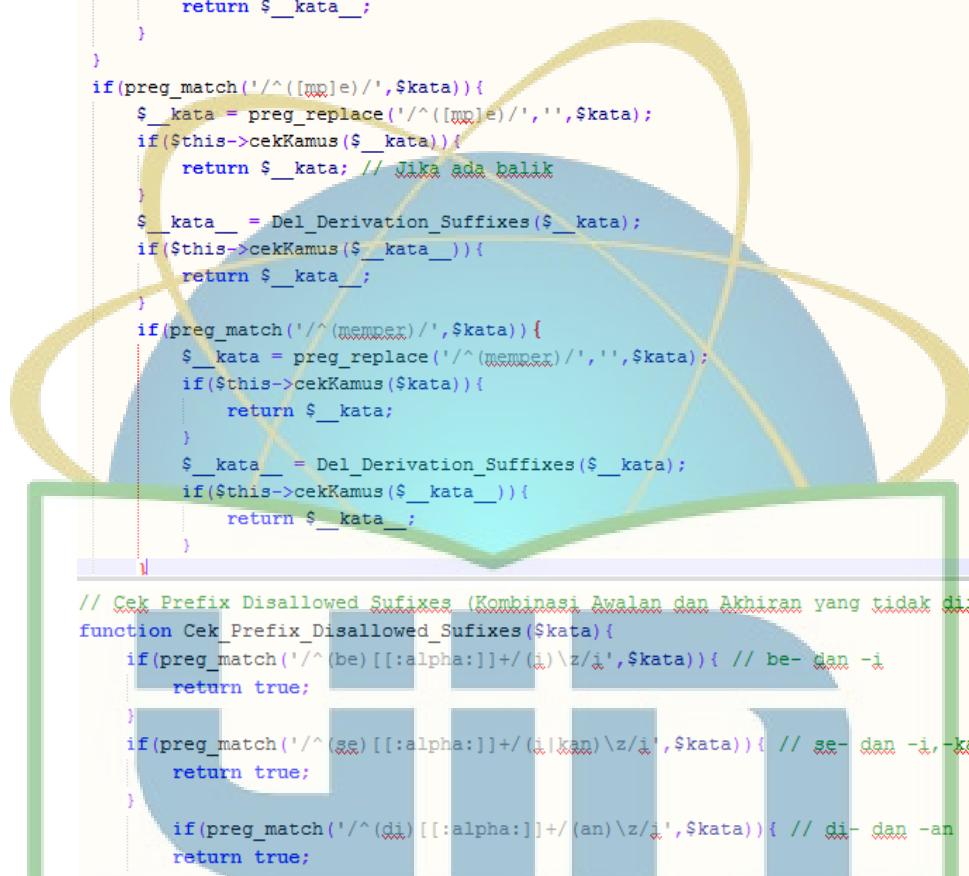
```

```

// Hapus Derivation Suffixes ("-i", "-an" atau "-kan")
function Del_Derivation_Suffixes($kata){
    $kataAsal = $kata;
    if(preg_match('/(i|an)\z/i', $kata)){ // Cek Suffixes
        $__kata = preg_replace('/(i|an)\z/i', '', $kata);
        if($this->cekKamus($__kata)){ // Cek Kamus
            return $__kata;
        }else if(preg_match('/(kan)\z/i', $kata)){
            $__kata = preg_replace('/(kan)\z/i', '', $kata);
            if($this->cekKamus($__kata)){
                return $__kata;
            }
        }
    }
    /*- Jika Tidak ditemukan di kamus -/
    return $kataAsal;
}

// Hapus Derivation Prefix ("di-", "ke-", "se-", "te-", "be-", "me-", atau "pe-")
function Del_Derivation_Prefix($kata){
    $kataAsal = $kata;
    /* Tentukan Tipe Awalan */
    if(preg_match('/^([di][ks]e)/', $kata)){ // Jika di-, ke-, se-
        $__kata = preg_replace('/^([di][ks]e)/', '', $kata);
        if($this->cekKamus($__kata)){
            return $__kata;
        }
        $__kata__ = Del_Derivation_Suffixes($__kata);
        if($this->cekKamus($__kata__)){
            return $__kata__;
        }
    }
    if(preg_match('/^([diper])/', $kata)){ //diper-
        $__kata = preg_replace('/^([diper])/', '', $kata);
        $__kata__ = Del_Derivation_Suffixes($__kata);
        if($this->cekKamus($__kata__)){
            return $__kata__;
        }
    }
    if(preg_match('/^([ke[bt]er])/', $kata)){ //keber- dan keter-
        $__kata = preg_replace('/^([ke[bt]er])/', '', $kata);
        $__kata__ = Del_Derivation_Suffixes($__kata);
        if($this->cekKamus($__kata__)){
            return $__kata__;
        }
    }
}

```



```

if(preg_match('/^([bt]e)/', $kata)){ //Jika awalannya adalah "te-", "be-", "bex-"
    $_kata = preg_replace('/^([bt]e)/', '', $kata);
    if($this->cekKamus($_kata)){
        return $_kata; // Jika ada balik
    }
    $_kata = preg_replace('/^([bx]e[lx])/', '', $kata);
    if($this->cekKamus($_kata)){
        return $_kata; // Jika ada balik
    }
    $_kata_ = Del_Derivation_Suffixes($_kata);
    if($this->cekKamus($_kata_)){
        return $_kata_;
    }
}
if(preg_match('/^([mp]e)/', $kata)){
    $_kata = preg_replace('/^([mp]e)/', '', $kata);
    if($this->cekKamus($_kata)){
        return $_kata; // Jika ada balik
    }
    $_kata_ = Del_Derivation_Suffixes($_kata);
    if($this->cekKamus($_kata_)){
        return $_kata_;
    }
}
if(preg_match('/^(memper)/', $kata)){
    $_kata = preg_replace('/^(memper)/', '', $kata);
    if($this->cekKamus($kata)){
        return $_kata;
    }
    $_kata_ = Del_Derivation_Suffixes($_kata);
    if($this->cekKamus($_kata_)){
        return $_kata_;
    }
}

// Cek Prefix Disallowed Sufixes (Kombinasi Awalan dan Akhiran yang tidak diizinkan)
function Cek_Prefix_Disallowed_Suffixes($kata){
    if(preg_match('/^(be)[[:alpha:]]+/(i|z|i)', $kata)){ // be- dan -i
        return true;
    }
    if(preg_match('/^(se)[[:alpha:]]+/(i|kan)\z/i', $kata)){ // se- dan -i,-kan
        return true;
    }
    if(preg_match('/^(di)[[:alpha:]]+/(an)\z/i', $kata)){ // di- dan -an
        return true;
    }
    if(preg_match('/^(me)[[:alpha:]]+/(an)\z/i', $kata)){ // me- dan -an
        return true;
    }
    if(preg_match('/^(ke)[[:alpha:]]+/(i|kan)\z/i', $kata)){ // ke- dan -i,-kan
        return true;
    }
    return false;
}

```

```

    }
    return $kataAsal;
}
//fungsi pencarian akar kata
function stemming($kata){
    $kataAsal = $kata;
    $cekKata = cekKamus($kata);
    if($cekKata == true){ // Cek Kamus
        return $kata; // Jika Ada maka kata tersebut adalah kata dasar
    }else{ //jika tidak ada dalam kamus maka dilakukan stemming
        $kata = Del_Inflection_Suffixes($kata);
        if(cekKamus($kata)){
            return $kata;
        }
        $kata = Del_Derivation_Suffixes($kata);
        if(cekKamus($kata)){
            return $kata;
        }
        $kata = Del_Derivation_Prefix($kata);
        if(cekKamus($kata)){
            return $kata;
        }
    }
}

```

Gambar 4.60 Algoritma Stemming Nazief Adriani

Gambar diatas merupakan potongan *pseudo code* algoritma *stemming* model Nazief dan Adriani dalam analisa *hadits* yang berfungsi untuk mengubah kata kedalam bentuk kata dasar.

#### 4. Algoritma TF-IDF

*Pseudo code* algoritma *tf-idf* yang digunakan adalah sebagai berikut:

```

class Tf_idf_mdl extends CI_Model {
    function __construct()
    {
        parent::__construct();
    }
    function model_buat_index($berita, $doc_id){/*{{{{*/
        #$this->db->debug =true;
        #$this->db->Execute("TRUNCATE TABLE thindex");
        $SQL = "SELECT COUNT(*) FROM thindex
                WHERE Term = '".$berita."' AND DocId = ".$doc_id."'";
        $num_rows = $this->db->getone($SQL);
        //jika sudah ada DocId dan Term tersebut

        //naikkan Count (+1)
        if ($num_rows > 0) {
            $count = $num_rows;
            $count++;
            $this->db->Execute("UPDATE thindex SET Count = ".$count." WHERE Term = '".$berita."' AND DocId = ".$doc_id."");
        }else{
            $this->db->Execute("INSERT INTO thindex (Term, DocId, Count) VALUES ('".$berita."', ".$doc_id.", 1)");
        }
    }/*}}}*/
}

```

```

function model_hitung_bobot() {
    //berapakah jumlah DocId total?, n
    $SQL = "SELECT COUNT(*)
            FROM (SELECT DISTINCT DocId FROM tbindex) AS tmp";
    $n   = $this->db->getone($SQL);
    $SQL_2 = "SELECT * FROM tbindex ORDER BY Id";
    $resBobot = $this->db->Execute($SQL_2);
    $num_rows = $this->db->getone("SELECT COUNT(*) FROM tbindex ORDER BY Id");
    print("Terdapat " . $num_rows .
        " Term yang diberikan bobot. <br />");
    while(!$resBobot->EOF)
    {
        //tf = tf * log (n/N)
        $term  = $resBobot->fields['Term'];
        $tf    = $resBobot->fields['Count'];
        $id    = $resBobot->fields['Id'];
        //berapa jumlah dokumen yang mengandung term itu?, N (df)
        $resNTerm = $this->db->getone("SELECT Count(*) as N FROM tbindex WHERE Term = '".$term."'");
        $NTerm   = $resNTerm;
        $w = $tf * log($n/$NTerm);
        $this->db->Execute("UPDATE tbindex SET Bobot = '".$w."' WHERE Id = ".$id.");
        $resBobot->movenext();
    }
}

```

Gambar 4.61 Algoritma TF-IDF

Gambar diatas merupakan potongan algoritma *term frequency inverse document frequency* dalam analisa hadits yang berfungsi untuk melakukan

pembobotan kata (*term*) yang akan digunakan dalam penghitungan selanjutnya yaitu *cosine similarity*.

## 5. Algoritma Cosine Similarity

*Pseudo code* algoritma *cosine similarity* yang digunakan adalah sebagai berikut:

```

function model_panjang_vektor() {/*
    $this->db->debug =true;
    $this->db->Execute("TRUNCATE TABLE tbvektor");
    //ambil setiap DocId dalam tbindex
    //hitung panjang vektor untuk setiap DocId tersebut
    //simpan ke dalam tabel tbvektor
    //perhitungan skalar
    $SQL = "SELECT COUNT(*)
            FROM (SELECT DISTINCT DocId FROM tbindex) AS tmp";
    $num_rows = $this->db->getone($SQL);

    print("Terdapat " . $num_rows .
        " dokumen yang dihitung panjang vektornya. <br />");
```

```

//perhitungan kuadrat
$SQL_2 = "SELECT DISTINCT DocId FROM tbindex";
$rowDocId = $this->db->Execute($SQL_2);
while(!$rowDocId->EOF)
{
    $docId = $rowDocId->fields['DocId'];
    $resVektor = $this->db->Execute("SELECT Bobot FROM tbindex WHERE DocId = ".$docId."");
    //jumlahkan semua bobot kuadrat
    $panjangVektor = 0;
    while(!$resVektor->EOF) {
        $panjangVektor = $panjangVektor + $resVektor->fields['Bobot'] * $resVektor->fields['Bobot'];
        $resVektor->moveNext();
    }
    //hitung akarnya
    $panjangVektor = sqrt($panjangVektor);
    //masukkan ke dalam tbvektor
    $resInsertVektor = $this->db->Execute("INSERT INTO tbvektor (DocId, Panjang) VALUES
    (".$docId.", ".$panjangVektor.")");
    $rowDocId->moveNext();
}

function model_hitung_sim($query){/*{{{{*/
    //$this->db->debug =true;
    $NTerm = $this->db->getOne("SELECT Count(*) as N FROM tbindex WHERE Term = '".$query."'");
    return $NTerm;
}

function model_hitung_semua_vektor(){/*{{{{*/
    //$this->db->debug =true;
    $NTerm = $this->db->getOne("SELECT Count(*) as n FROM tbvektor");
    return $NTerm;
}

function model_hitung_semua_vektor(){/*{{{{*/
    //$this->db->debug =true;
    $NTerm = $this->db->getOne("SELECT Count(*) as n FROM tbvektor");
    return $NTerm;
}

function model_hitung_kemiripan($aquery,$aBobotQuery,$panjangQuery,$query){/*{{{{*/
    $SQL_2 = "SELECT * FROM tbvektor ORDER BY DocId";
    $resDocId = $this->db->Execute($SQL_2);
    $jumlahmirip = 0;
    while(!$resDocId->EOF)
    {
        $dotproduct = 0;
        $docId = $resDocId->fields['DocId'];
        $SpanjangDocId = $resDocId->fields['Panjang'];
        $resTerm = $this->db->Execute("SELECT * FROM tbindex WHERE DocId = ".$docId."");
        while(!$resTerm->EOF)
        {
            for ($i=0; $i<count($aquery); $i++) {

                if ($resTerm->fields['Term'] == $aquery[$i]) {

                    $dotproduct = $dotproduct + $resTerm->fields['Bobot'] * $aBobotQuery[$i];
                } //end if
            }
        }
    }
}

```

```

        }
        $resTerm->movenext();
    }
    $sim = 0;
    if ($dotproduct > 0) {
        $sim = $dotproduct / ($panjangQuery * $panjangDocId);
        //simpan kemiripan > 0 ke dalam tbcache
        $resInsertCache = $this->db->Execute("INSERT INTO tbcache (Query, DocId, Value) VALUES ('".$query."', ".$docId.", ".$sim.")");
    }
    $resDocId->movenext();
}
if ($jumlahmirip == 0) {
    $resInsertCache = $this->db->Execute("INSERT INTO tbcache (Query, DocId, Value) VALUES ('".$query."', 0, 0)");
}

```

Gambar 4.62 Algoritma *Cosine Similarity*

Gambar diatas merupakan potongan *pseudo code* algoritma *cosine similarity* dalam analisa *hadits* yang berfungsi untuk penghitungan nilai relevansi antar dokumen dari hasil pembobotan kata dengan algoritma tf-idf untuk menemukan dokumen yang relevan (*syarah hadits* yang sesuai dengan *query*).



## **BAB V**

### **HASIL DAN PEMBAHASAN**

Fase ini merupakan fase terakhir setelah melalui fase-fase sebelumnya. Dimana dalam fase ini, hasil dari perancangan yang telah dilakukan akan diimplementasikan ke dalam sistem dengan penjelasan sebagai berikut:

#### **5.1 Fase Implementasi**

Pada fase implementasi ini akan dijelaskan implementasi perangkat yang digunakan, dan hasil dari penelitian yang dilakukan sebagai berikut:

##### **5.1.1 Implementasi Perangkat**

Perangkat yang digunakan dalam membangun sistem ini adalah sebagai berikut:

###### **5.1.1.1 Perangkat Lunak (*Software*)**

Perangkat lunak yang penulis gunakan dalam penelitian ini adalah sebagai berikut:

- |    |            |   |                |
|----|------------|---|----------------|
| 1. | Xampp      | : | Xampp v.3.2.2  |
| 2. | Apache     | : | v. 2.4.29      |
| 3. | MySql      | : | v.5.0.11       |
| 4. | PHP        | : | v.5.6.33       |
| 5. | PhpMyAdmin | : | v.4.7.4        |
| 6. | Sublime    | : | Sublime Text 3 |
| 7. | Photoshop  | : | cc 2017        |

###### **5.1.1.2 Perangkat Keras (*Hardware*)**

Perangkat keras yang penulis gunakan dalam penelitian ini adalah sebagai berikut:

- |    |                |   |                        |
|----|----------------|---|------------------------|
| 1. | Sistem Operasi | : | Windows 8.1 Pro 32-bit |
| 2. | Prosesor       | : | 2.53GHz Intel Core I3  |
| 3. | RAM            | : | 3 Gigabyte             |
| 4. | Harddisk       | : | 300GB                  |

## 5.1.2 Hasil Penelitian

Pada tahap ini, sistem telah siap untuk dioperasikan dengan hasil untuk mengetahui apakah sistem sesuai dengan *user interface* yang telah dirancang sebelumnya atau tidak. Hasil penelitian ini meliputi sebagai berikut:

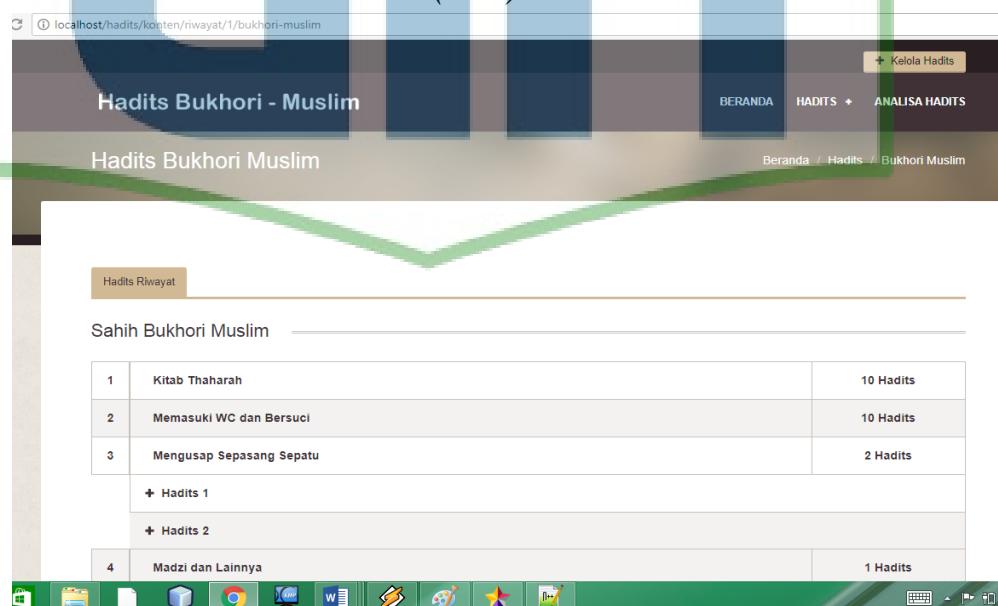
### 5.1.2.1 Halaman Utama



Gambar 5.1 Capture Halaman Utama

Gambar diatas merupakan tampilan dari halaman utama yang muncul ketika pertama kali membuka aplikasi.

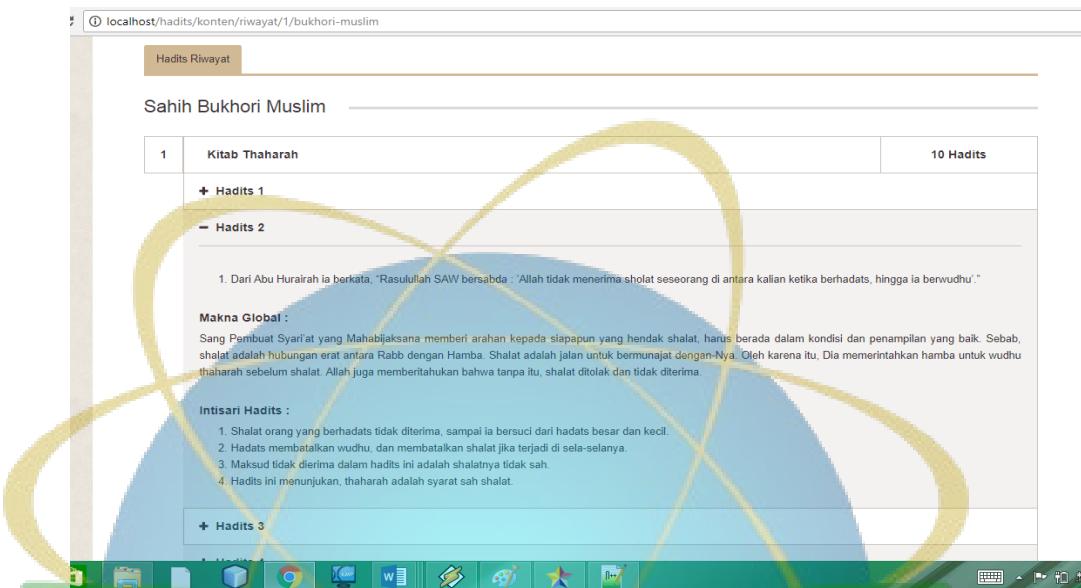
### 5.1.2.2 Halaman Baca Hadits (User)



Gambar 5.2 Capture Baca Hadits (user)

Gambar diatas merupakan tampilan dari halaman baca hadits yang muncul ketika *user* mengklik button “hadits”. Terdapat bab hadits dan isi hadits dari masing-masing bab hadits yang tersedia.

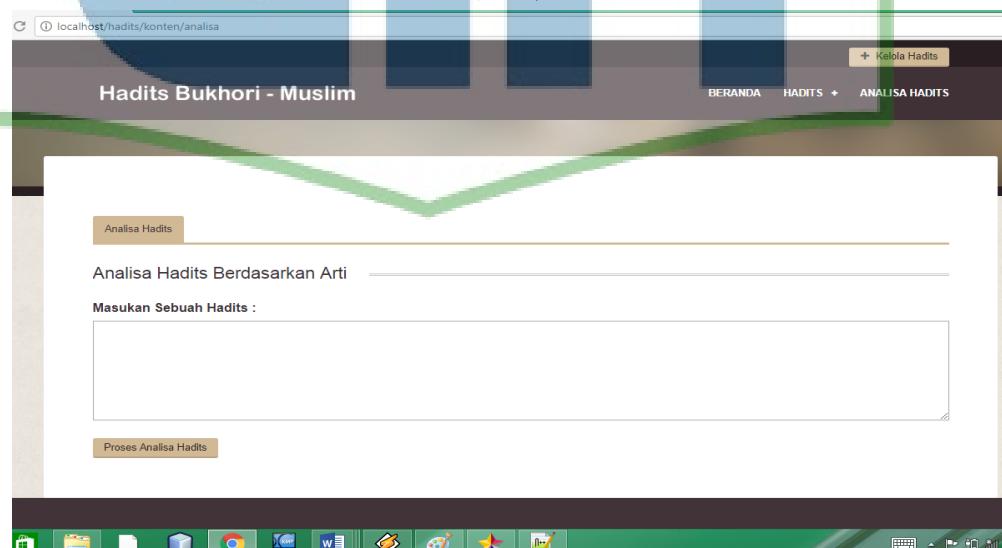
### 5.1.2.3 Halaman Baca Hadits (*User*)

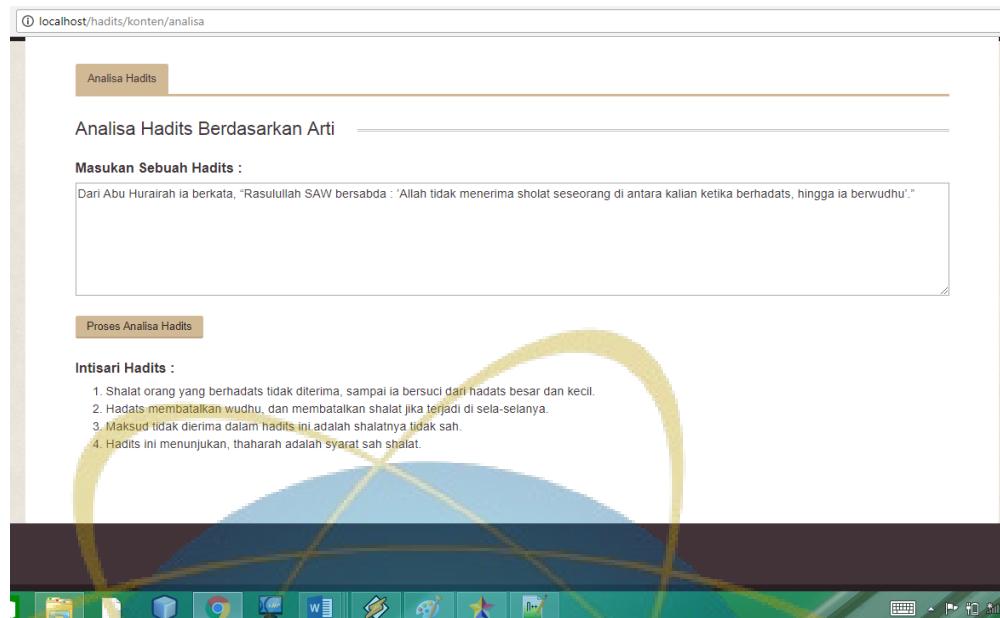


Gambar 5.3 Capture Halaman Baca Hadits (*user*)

Gambar diatas merupakan tampilan dari halaman baca hadits yang muncul ketika *user* mengklik salah satu dari daftar hadits dari bab hadits yang tersedia. Terdapat hadits (terjemahan) beserta makna hadits dan intisari hadits.

### 5.1.2.4 Halaman Analisa Hadits (*User*)

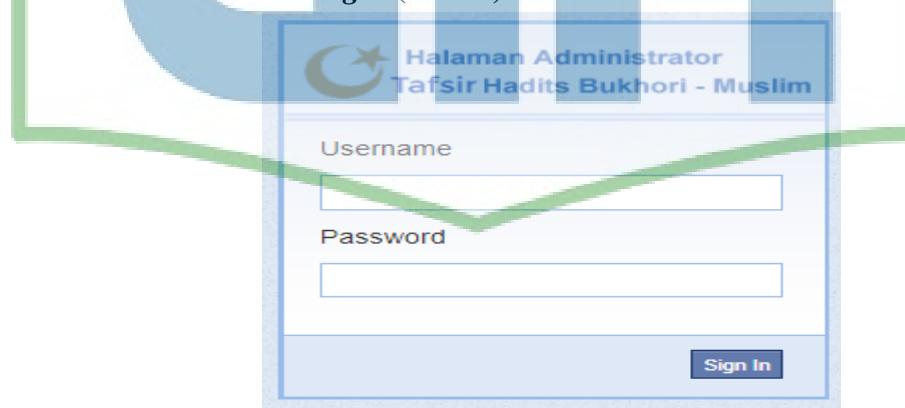




Gambar 5.4 Capture Halaman Analisa Hadits (*User*)

Gambar diatas merupakan tampilan dari halaman analisa hadits yang muncul ketika *user* mengklik tombol analisa hadits untuk melakukan proses analisa yaitu dengan mengisi *hadits* dalam bahasa indonesia dengan lengkap (*sanad* dan *matan*) pada kolom yang tersedia lalu setelah klik tombol proses analisa hadits maka akan muncul *syarah hadits* yang benar sesuai dengan hadits yang dimasukkan, sistem tidak akan memberikan *output* apabila *hadits* yang di *input*-kan tidak mengandung salah satu struktur *hadits* (*sanad/matan*).

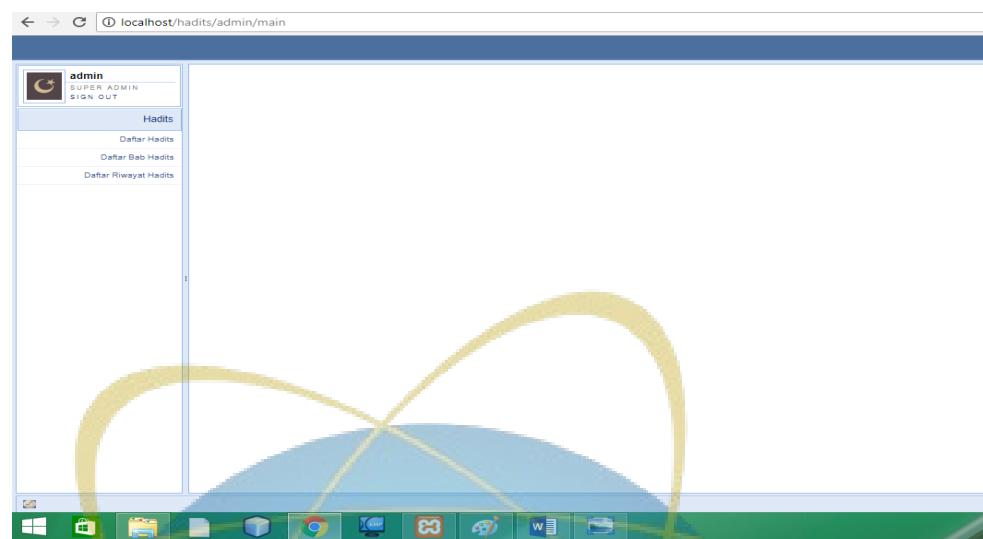
#### 5.1.2.5 Halaman *Login* (admin)



Gambar 5.5 Capture Halaman *Login* (Admin)

Gambar diatas merupakan tampilan dari halaman *login* untuk admin yang muncul ketika admin mengklik tombol Tambah Hadits untuk melakukan tambah, edit, hapus, dan cari hadits.

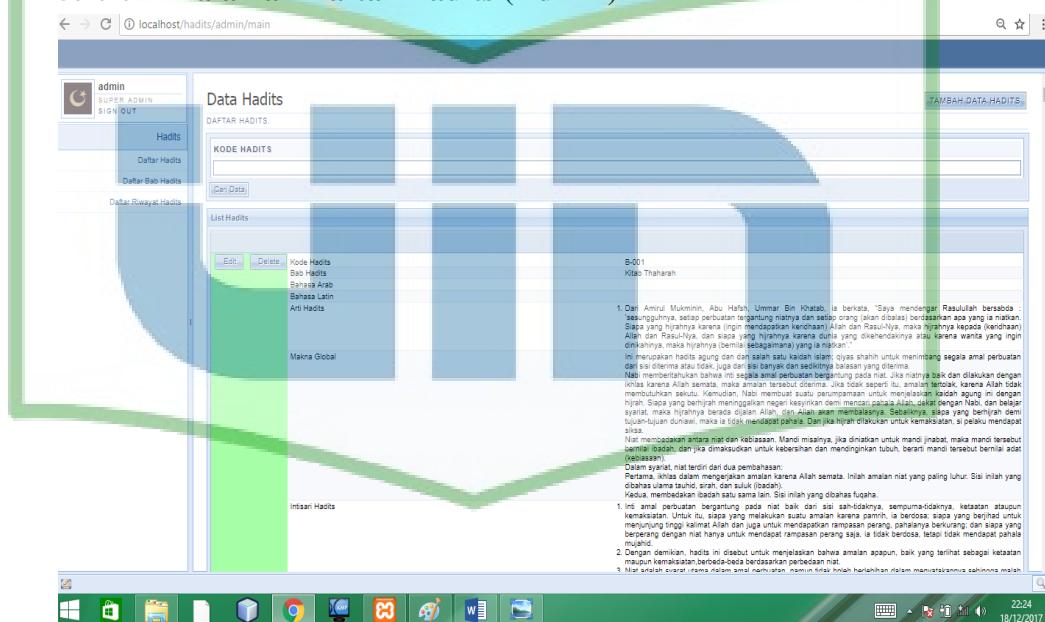
### 5.1.2.6 Halaman Utama Admin



Gambar 5.6 Capture Halaman Utama Admin

Gambar diatas merupakan tampilan dari halaman utama admin yang muncul ketika admin telah berhasil melakukan *login*.

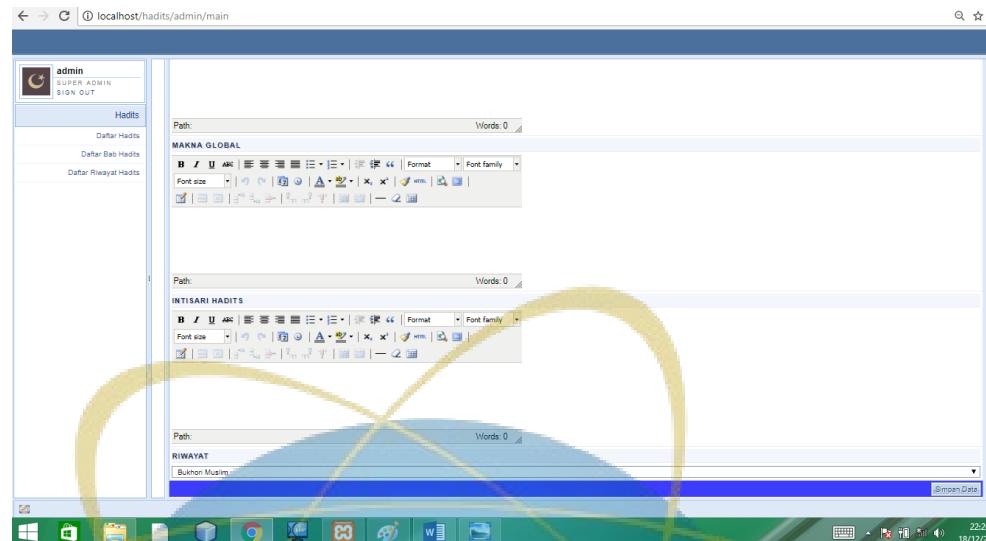
### 5.1.2.7 Halaman Daftar Hadits (Admin)



Gambar 5.7 Capture Halaman Daftar Hadits

Gambar diatas merupakan tampilan dari halaman daftar hadits untuk melakukan tambah data, edit dan hapus data hadits yang muncul ketika admin mengklik tombol Daftar Hadits pada menu admin.

### 5.1.2.8 Halaman Tambah Hadits (Admin)



Gambar 5.8 Capture Halaman Tambah Hadits

Gambar diatas merupakan tampilan dari halaman tambah hadits yang muncul ketika admin mengklik tombol Tambah Data Hadits pada halaman daftar hadits, admin mengisi kolom yang ada pada halaman tersebut lalu klik simpan untuk menyimpannya kedalam *database*.

### 5.1.2.9 Halaman Daftar Bab Hadits (Admin)

NO	KODE BAB	NAMA BAB	TAMBAH DATA BAB HADITS		
1	B-001	Kibab Thahrirah			
2	B-002	Manasik WC dan Berisuci			
3	B-003	Mengusap Sesapang Seputu			
4	B-004	Mandi dan Lainnya			
5	B-005	Hukum Ketika Mandi Hadits			
6	B-006	Cara Memperbaiki Tarah yang Terkena Kerong			
7	B-007	Mandi Jizbat			
8	B-008	Hukum Tidur dalam Keadaan Jumub			
9	B-009	Hukum Mimpi Basah Bay Wanita			
10	B-010	Jimak Menyebarkan Wajah Mandi, Balk Keluar Mani Atau Tidak			
11	B-011	Kerentakan Jumur Ah yang Cukup untuk Mandi Jizbat			
12	B-012	Tayammum			
13	B-013	Tata Cara Tayammum			
14	B-014	Ketarikjan Nabz Terkeni Lima Pakaian Istimewa			
15	B-015	Haid			
16	B-016	Hukum Mustahidah			
17	B-017	Hukum Mengganti Iban Saat Haid			
18	B-018	Wanta Haid sak Menghadia Shalat tapi Menghindari Shaum			

Gambar 5.9 Capture Halaman Bab Hadits

Gambar diatas merupakan tampilan dari halaman bab hadits untuk melakukan tambah data, edit dan hapus data bab hadits yang muncul ketika admin mengklik tombol Daftar Bab Hadits pada menu admin.

### 5.1.2.10 Halaman Tambah Bab Hadits (Admin)

Gambar 5.10 *Capture* Halaman Tambah Bab Hadits

Gambar diatas merupakan tampilan dari halaman tambah bab hadits yang muncul ketika admin mengklik tombol Tambah Data Bab Hadits pada menu Daftar Bab Hadits, admin mengisi kolom kode bab dan nama bab hadits yang tersedia lalu klik simpan untuk menyimpan data.

### 5.1.2.11 Halaman Daftar Riwayat Hadits (Admin)

Gambar 5.11 *Capture* Halaman Daftar Riwayat Hadits

Gambar diatas merupakan tampilan dari halaman daftar riwayat hadits untuk melakukan tambah data, edit dan hapus data riwayat hadits yang muncul ketika admin mengklik tombol Daftar Riwayat Hadits pada menu admin.

### 5.1.2.12 Halaman Tambah Riwayat Hadits (Admin)

Gambar 5.12 *Capture* Halaman Tambah Riwayat Hadits

Gambar diatas merupakan tampilan dari halaman tambah riwayat hadits yang muncul ketika admin mengklik tombol Tambah Data Riwayat Hadits pada menu Daftar Riwayat Hadits, admin mengisi kolom kode riwayat hadits dan nama riwayat hadits yang tersedia lalu klik simpan untuk menyimpan data.

## 5.2 Pembahasan Penelitian

Pada tahap ini, penulis melakukan pembahasan dengan meminta tanggapan kepada Bapak Rifqi Muhammad Fatkhil selaku Ketua Program Studi Hadits Fakultas Ushuluddin UIN Jakarta mengenai aplikasi yang telah penulis bangun. Setelah melalui proses percobaan, beliau mengatakan bahwa fungsi semua fitur dari aplikasi yang telah penulis bangun sudah berjalan dengan baik, dan sistem sudah berhasil memberikan *syarah hadits* sebagai *output* ketika *user* memasukkan *hadits* berbahasa indonesia. Menurut beliau pula, aplikasi sudah memenuhi kebutuhan dalam mempelajari *syarah hadits* sesuai tergantung *source*-nya yaitu sesuai dengan kitab yang digunakan sebagai acuan pen-syarah-an.

Dengan hasil tersebut, bapak Rifqi menilai bahwa aplikasi yang telah penulis bangun pasti bermanfaat, karena dengan penelitian ini orang-orang awam-pun dapat mengetahui makna yang terkandung dalam hadits dengan membaca *syarah hadits*-nya. Dengan begitu, kesalahan penafsiran terhadap suatu hadits dapat diminimalisir. Akan tetapi, bapak Rifqi menilai bahwa dalam penelitian ini pula terdapat beberapa kelemahan diantaranya yaitu tampilan aplikasi kurang menarik dan aplikasi kurang *user friendly* dikarenakan *user* yang ingin mengetahui *syarah hadits* harus mengetahui *hadits* terjemahannya secara lengkap untuk mendapatkan *output* yang sesuai, hal ini dapat dijadikan dasar acuan atau saran untuk pengembangan selanjutnya.

### 5.3 Pengujian (*Testing*)

#### 5.3.1 Pengujian *Stemming* Nazief Adriani

Adapun algoritma *stemming* yang digunakan dalam penelitian ini adalah algoritma Nazief & Adriani, dimana *stemming* jenis ini merupakan *stemming* yang memiliki tingkat akurasi (presisi) lebih tinggi dari jenis *stemming* yang lain. Namun, pada proses *stemming* algoritma ini, penulis menemukan beberapa kata yang tidak berhasil di *stemming* pada saat melakukan analisa secara manual, kata yang gagal di *stemming* tersebut terjadi karena:

1. Proses penghilangan imbuhan dalam *stemming* nazief adriani dilakukan dengan menghilangkan akhiran terlebih dahulu baru diikuti dengan penghilangan awalan, sehingga langkah ini dapat menyebabkan hasil dari kata yang di *stemming* menjadi kurang tepat pada beberapa kata seperti beberapa kata dibawah ini:

Tabel 5.1 Contoh Hasil *Stemming* Yang Kurang Tepat

Id	Token Awal	Hasil		Hasil seharusnya
		Stemming		
36	dinikahinya	nikahi		nikah
91	diampuni	mpuni		ampuni
236	memberkahai	ber		berkah
1897	sejumlah	sejum		jumlah
2226	setelah	sete		telah
2520	melakukan	laku		lakukan
2875	melangkah	lang		langkah

2. Pengecekan awalan kata pada algortima Nazief Adriani salah satunya hanya terbatas pada awalan te- dan pe- sehingga untuk kata-kata yang memiliki awalan ter- dan awalan pe- yang diikuti -ng tidak dapat dilakukan *stemming* seperti beberapa kata dibawah ini:

Tabel 5.2 Contoh Hasil *Stemming* Awalan ter- dan pe- diikuti -ng

<b>Id</b>	<b>Token Awal</b>	<b>Hasil <i>Stemming</i></b>	<b>Hasil seharusnya</b>
<b>4066</b>	terlimpah	terlimpah	limpah
<b>4162</b>	terpuji	terpuji	puji
<b>4372</b>	terakhir	terakhir	akhir
<b>4753</b>	termasuk	termasuk	masuk
<b>5226</b>	tersisa	tersisa	sisa
<b>5375</b>	terjatuh	terjatuh	jatuh
<b>3723</b>	pengenalan	pengenalan	kenal
<b>4062</b>	pengagungan	pengagungan	agung
<b>4252</b>	pengampunan	pengampunan	ampun

3. Algoritma Nazief Adriani masih belum bisa mengembalikan kata asli (kata dasar) yang benar dari kata dengan awalan me- dengan diikuti -m-, padahal algoritma ini melakukan pengecekan terhadap aturan tersebut, sehingga seperti beberapa kata dibawah ini masih belum bisa ter *stemming* dengan benar:

Tabel 5.3 Contoh Hasil *Stemming* aturan me- diikuti -m-

<b>Id</b>	<b>Token Awal</b>	<b>Hasil <i>Stemming</i></b>	<b>Hasil seharusnya</b>
<b>261</b>	meminta	minta	pinta
<b>873</b>	memercik	mercik	percik
<b>4691</b>	memisah	misah	pisah
<b>2128</b>	memerintah	merintah	perintah
<b>2554</b>	memalingkan	maling	paling
<b>3276</b>	memakai	makai	pakai

4. Pada kata yang merupakan bahasa asing atau nomor, tidak dilakukan *stemming*, hal ini dikarenakan cara kerja yang pertama dari algoritma nazief adriani adalah dengan melakukan pengecekan apakah kata yang akan di *stemming* terdapat pada kamus yang tersedia atau tidak, jika iya

maka akan dilakukan *stemming*, tapi jika tidak ada maka kata akan dianggap sebagai *root word* atau sebagai kata asal, sehingga kata asing dan nomor seperti : Musyik, qirath, jihad, mutalaffiat, khadaq, itikaf, shaum, qadha, haruriyah, istihadah, istinja, madzi, khitan, junub, syafaat, subhanallah, jinabat, saw, rasulullah, 1,2,3,4,5 dan sebagainya akan tetap menjadi seperti asalnya.

Berdasarkan analisa hasil *stemming* diatas, untuk mengetahui tingkat ke akuratan *Stemming* dengan Algoritma Nazief Adriani ini maka akan dihitung nilai persentase antara kata yang gagal di *stemming* dengan yang berhasil. Total token yang ada pada *database* hasil *preprocessing* adalah 6.983, namun dalam 6.983 itu terdapat banyak pengulangan beberapa token, sehingga untuk perhitungan hasil *stemming* kata yang terbentuk hanya berjumlah 2.051. Untuk itu, hasil dari akurasi *stemming* Nazief adriani adalah sebagai berikut:

$$1. \text{ Prosentase gagal} = \frac{186}{2051} \times 100\% = 9,07\%$$

$$2. \text{ Prosentase berhasil} = \frac{1865}{2051} \times 100\% = 90,93\%$$

### 5.3.2 Pengujian Sistem

Dalam fase *testing* ini untuk melakukan pengujian sistem, penulis menggunakan pengujian *confusion matrix* yang biasa digunakan dalam perhitungan akurasi pada suatu sistem temu kembali informasi untuk mengevaluasi seberapa baik kemampuan sistem dalam pencarian dokumen.

Pengujian sistem ini adalah dengan melakukan percobaan sebanyak 204 kali menggunakan *query* terhadap 204 dokumen yang ada dalam *database*. *Query* yang digunakan adalah *query* yang memiliki jumlah dokumen relevan masing-masing 1 untuk masing-masing *query*. Artinya, satu *query* hanya relevan dengan satu dokumen dimana dokumen tersebut merupakan *syarah hadits* yang relevan berdasarkan *query* yang dimasukkan.

Setelah melakukan percobaan terhadap 204 dokumen dengan *query* tersebut, didapatkan hasil benar yang ditemukan berjumlah 181 dokumen yang ditemukan relevan (sesuai dengan *query*), dan ditemukan 23 dokumen

yang ditemukan tidak relevan (tidak sesuai dengan *query*). Oleh karena itu, maka hasil pengujian *confusion matrix* sesuai dengan rumus yang telah dibahas pada point 2.9 tabel 2.3 adalah sebagai berikut:

1.  $Precision = \frac{181}{(181+0)} \times 100\% = 100\%$
2.  $Recall = \frac{181}{(181+23)} \times 100\% = 88,7\%$
3.  $Accuracy = \frac{(181+0)}{(181+0+0+23)} \times 100\% = 88,73\%$
4.  $Error rate = \frac{(23+0)}{(181+0+0+23)} \times 100\% = 11,27\%$

Hasil dokumen yang salah atau ditemukan tidak relevan dengan *query* yang di-input kan ini, disebabkan karena perhitungan *cosine similarity* menentukan mana dokumen yang terdekat berdasarkan frekuensi kemunculan kata (*term*) dalam pembobotan kata, sehingga tidak semua dokumen yang ditemukan (*ter-retrieve*) merupakan dokumen yang relevan. Hal ini dapat terjadi karena sebuah *term* yang muncul dibanyak dokumen tidak dianggap lebih penting daripada term lain yang hanya muncul pada sedikit dokumen, namun jika sebuah term memiliki frekuensi kemunculan lebih tinggi dalam sebuah dokumen daripada dokumen lain, maka dokumen tersebut dianggap lebih penting. Sehingga, jika *term* yang terdapat pada *query* lebih banyak muncul dalam dokumen lain meskipun itu bukan dokumen aslinya (dokumen yang relevan sesungguhnya) maka akan dianggap mirip karena hasil *cosine similarity* otomatis akan lebih tinggi daripada dokumen lainnya.

## BAB VI

### PENUTUP

Pada Bab penutup ini, akan dikemukakan kesimpulan dari penelitian yang telah dilakukan, berikut saran yang dapat digunakan untuk penelitian selanjutnya.

#### 6.1 Kesimpulan

Berdasarkan rumusan masalah yang telah dikemukakan, maka kesimpulan dalam penelitian ini adalah bahwa metode *term frequency inverse document frequency* (tf-idf) dan *cosine similarity* telah berhasil diterapkan dalam sistem dengan baik dimana sistem dapat memberikan *output* berupa dokumen yang relevan yaitu *syarah hadits* sesuai dengan *query* yang di-*input* kan, dengan melalui 3 tahapan teks *preprocessing* yaitu *tokenizing*, *stopword removal* atau *filtering*, dan *stemming*. Hasil pengujian *stemming* model Nazief Adriani yang telah dilakukan pada point 5.3.1 menunjukkan hasil akurasi sebesar 90,93% yang menyatakan bahwa *stemming* model ini memiliki tingkat akurasi yang tinggi. Kemudian, pengujian sistem yang dilakukan dengan menggunakan *confusion matrix* dalam penelitian ini pada point 5.3.2 didapatkan nilai *precision* 100%, *recall* 88,7%, *accuracy* 88,73 %, dan *error rate* 11,27 %. Sehingga, sistem dapat dikatakan baik, dikarenakan sistem yang baik adalah sistem yang memiliki nilai *recall* dan *precision* tinggi serta tingkat akurasi yang tinggi pula.

#### 6.2 Saran

Untuk pengembangan selanjutnya, maka saran-saran yang dapat penulis berikan adalah sebagai berikut:

1. Sistem dapat dilengkapi menjadi *full* satu kitab dan dapat dikembangkan dengan menggunakan kitab *syarah hadits* yang lebih spesifik lainnya seperti kitab syarah *Fatkhul Barri*, *syarah Imam An-Nawawi* atau kitab *syarah* dari hadits shahih periyat lain.
2. Sistem dapat dikembangkan dengan *user interface* yang lebih menarik dan *user friendly*.

3. Penerapan metode pembobotan kata dan metode *similarity* yang memiliki tingkat akurasi lebih tinggi.
4. Sistem dapat dikembangkan selain menggunakan manajemen *database* MySql, karena dengan menggunakan MySql sistem berjalan sedikit lebih lambat dikarenakan sistem harus menelusuri data yang ada sehingga waktu untuk menemukan dokumen yang relevan relatif lama.



## DAFTAR PUSTAKA

- Agusta, Lady. 2009. *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia*. Jurnal. Konferensi Nasional Sistem dan Informatika 2009, Bali, November 14, 2009.
- Almanar, M. Abduh. 2012. *Pengantar Studi Hadis*. Jakarta: Pustaka Firdaus.
- Amin, Fatkhul dan Purwatinetyas. 2015. *Rancang Bangun Information Retrieval System (IRS) Bahasa Jawa Ngoko pada Palintangan Penjebar Semangad dengan Metode Vector Space Model (VSM)*. Jurnal. Jurusan teknologi informasi. Fakultas Teknologi Informasi, Universitas Stikubank Semarang.
- Arifin, Zainul. 2010. *Studi Kitab Hadits*. Surabaya: Pustaka Al-Muna.
- Arwanda, Ivan. 2013. *Penerapan Metode Text Mining pada Aplikasi Chatbot*. Skripsi. Jurusan Teknik Informatika Universitas Ilmu Komputer. Bandung.
- Basuma, Devi. 2013. *Pencarian Alamat Fasilitas Umum Menggunakan Metode Vector Space Model ( Studi Kasus Kota Pekanbaru )*. Skripsi. Jurusan Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Sultan Syarif Kasim Riau Pekanbaru.
- Dewa, Arie, dan Agustinus. 2016. *Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity*. Jurnal. Teknik Informatika Universitas Sam Ratulangi Manado.
- Dhony, Sukmawati, dan Nurdin. 2012. *Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode Term Frequency Inverse Document Frequency (Tf-Idf)*. Jurnal. Jurusan Ilmu Komputer/Informatika. Fakultas Sains dan Matematika. Universitas Diponegoro.
- Feldman, Ronen, dan Sanger, James. 2007. *The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.
- Ferdinandus, Subari. 2015. *Sistem Information Retrieval Layanan Kesehatan Untuk Berobat dengan Metode Vector Space Model berbasis WebGis*. Jurnal. Teknik Informatika. STIKI Malang.

- Hidayatullah, Priyanto., dan Jauhari Khairul Kawistara. 2014. *Pemrograman Web*. Bandung: Informatika.
- Husni., 2010, "Sistem Temu-Balik Informasi", diktat kuliah, Teknik Informatika Universitas Trunojoyo.
- Indrajani. 2015. *Database Design (Case Study All in One)*. Jakarta: PT Elex Media Komputindo.
- Idri. 2010. *Studi Hadits*. Jakarta: Kencana.
- Katsir, Ibnu. 2013. *Tafsirul 'Allam Syarh 'Umdatul Ahkam*. Jakarta: Ummul-Qura.
- Kendall, Kennet E & Julie E. Kendall. 2010. *Analisis dan Perancangan Sistem*. Jakarta: PT.Index.
- Kurniawan, Agung. 2015. *Model Pengembangan Perangkat Lunak dengan Waterfall Rad Prototyping Incremental dan Spiral beserta Perbedaannya*. [http://www.academia.edu/17873145/Model Pengembangan Perangkat Lunak dengan Waterfall Rad Prototyping Incremental dan Spiral beserta Perbedaannya](http://www.academia.edu/17873145/Model_Pengembangan_Perangkat_Lunak_dengan_Waterfall_Rad_Prototyping_Incremental_and_Spiral_beserta_Perbedaannya). (Diakses pada 26 April 2017 pukul 23.58 WIB).
- Ma'arif, Abdul Aziz. 2015. *Penerapan Algoritma Tf-Idf Untuk Pencarian Karya Ilmiah*. Jurnal. Jurusan Teknik Informatika. Fakultas Ilmu Komputer. Universitas Dian Nuswantoro Semarang.
- Mandala. R. 2004. *Bahan Kuliah Sistem Temu Kembali Informasi*. Institut Teknologi Bandung. Departemen Teknik Informatika.
- Manning, C., Raghavan, P., 2009. *An Introduction to Information Retrieval*. Stanford. USA.
- Mukaromah, Kholila. 2015. *Kajian Syarah Hadits Subul Al-Salam*. Tesis. Jurusan Studi Agama dan Filsafat. Fakultas Humaniora. Universitas Islam Negeri Sunan Kalijaga. Yogyakarta.
- Nugroho. 2013. *Mengenal XAMPP Awal*. Yogyakarta: MediaKom.
- Nugroho Eko., 2011. *Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Robin-Karp*. Jurnal. Program Studi Ilmu Komputer. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Brawijaya Malang.

- Nurdiana,Ogie, dkk. 2016. Perbandingan Metode *Cosine Similarity* Dengan Metode *Jaccard Similarity* Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia. Jurnal. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Gunung Djati Bandung.
- Pemda. Alasan Memakai Aplikasi Berbasis Web. <http://aplikasipemda.com/5-alasan-memakai-aplikasi-berbasis-web/s>. (Diakses pada 2 September 2017, Pukul 15.23 WIB).
- Purnama, Yulian. 2012. *Mengenal Shahih Bukhari dan Shahih Muslim*. <https://muslim.or.id/10838-mengenal-shahih-bukhari-dan-shahih-muslim.html>. (Diakses pada 04 Mei 2017 Pukul 15.07 Wib).
- Putra, Agung Auliaguntary Arif. 2016. *Implementasi Text Summarization Menggunakan Metode Vector Space Model pada Artikel Berita Bahasa Indonesia*. Skripsi. Jurusan Teknik Informatika. Fakultas Teknik dan Ilmu Komputer. Universitas Komputer Indonesia.
- Rahardjo, Budi., Iman Haryanto., dan E. Rosiana K. 2014. *Modul Pemrograman WEB (HTML, PHP, & MySQL) Revisi Kedua*. Bandung: Modula.
- Rizki, Dhidik, dan Eko Supraptono. 2017. *Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen*. Skripsi. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang Kampus Sekaran, Gunungpati, Semarang.
- Rosyid, Khoirul. 2016. *Kepemimpinan Menurut Hadits Nabi Saw*. Skripsi. Jurusan Tafsir Hadits. Fakultas Ushuluddin. Institut Agama Islam Negeri (IAIN) Raden Intan Lampung.
- Rozas, I. R. dan Sarno, R. 2011. Sistem Pemilihan Kontrol Keamanan Informasi Berbasis ISO 27001. *Seminar Nasional Pascasarjana XI*. IT. Surabaya.
- Safitri, Rima Noer. 2013. *Temu Kembali Informasi Pada Pencarian Jurnal Skripsi Menggunakan Metode Single Pass Clustering*. Skripsi. Universitas Muhammadiyah Gresik.

- Salton, G., 1989. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of information by computer.* Addison – Wesly Publishing Company, Inc. USA.
- Satzinger, Jackson, Burd. 2012. *Introduction to System Analysis and Design : An Agile, Iterative Approach.* USA: Course Technology, Cengage Learning. (SDLC).
- Simorangkir, Manase Sahat. 2017. *Studi Perbandingan Algoritma-Algoritma Stemming Untuk Dokumen Teks Bahasa Indonesia.* Jurnal. Teknik Informatika. Universitas Presiden Jababeka Education.
- Sugiarti, Yuni. 2013. *Analisis dan Perancangan UML (Unified Modeling Language) Generated VB.6.* Yogyakarta: Graha Ilmu.
- Sugiyono. (2012). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D.* Cetakan ke-17. Bandung: Alfabeta.
- Triandini , Evi dan Suardika , Gede. 2012. *Step by Step Desain Proyek Menggunakan UML.* Yogyakarta: Andi Offset.
- Zuhri, Muhammad. 2011. *Hadis Nabi Telaah Historis dan Metodologis.* Yogyakarta: Tiara Wacana Yogyka.





KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI (UIN)  
SYARIF HIDAYATULLAH JAKARTA  
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia  
Telp.: (62-21) 7493606, 7493517, 7401925 Fax.: (62-21) 7493315

Email : fst@uinjkt.ac.id  
Website : fst.uinjkt.ac.id

Nomor : B-40/ F9/KM.01/02/2018  
Lampiran : -  
Perihal : Bimbingan Skripsi

Jakarta, 19 Februari 2018

Kepada Yth.

1. Victor Amrizal, MKom
  2. Hendra Bayu Suseno, M.Kom
- Dosen Pembimbing Skripsi

Assalamu 'alaikum Wr.Wb.

Dengan ini diharapkan kesediaan Saudara untuk menjadi pembimbing I/II/ (Materi/Teknis)\* penulisan skripsi mahasiswa:

Nama : Ria Melita  
NIM : 1113091000012  
Program Studi : Teknik Informatika  
Judul Skripsi : "Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web."

Judul tersebut telah disetujui oleh Program Studi bersangkutan pada tanggal 21 Januari 2017 dengan outline, abstraksi dan daftar pustaka terlainpir. Bimbingan skripsi ini diharapkan selesai dalam waktu 6 (enam) bulan setelah ditandatanganinya surat penunjukan pembimbing skripsi.

Apabila terjadi perubahan terkait dengan skripsi tersebut selama proses pembimbingan, harap segera melaporkan kepada Program Studi bersangkutan.

Demikian atas kesediaan Saudara, kami ucapkan terima kasih.

Wassalamu 'alaikum Wr.Wb.

a.n Dekan  
Wadek Bidang Akademik

Dr. Ir. Sipawati, MP

NIP 19641204 199203 2 001

Tembusan:  
Dekan (sebagai laporan)

## Lampiran 2. Transkip Wawancara

### TRANSKIP WAWANCARA

Judul Skripsi	: Penerapan Metode <i>Term Frequency Inverse Document Frequency</i> dan <i>Cosine Similarity</i> pada Sistem Temu Kembali Informasi untuk Mengetahui <i>Syiah Hadits</i> Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim).
Waktu	: 12.56 WIB.
Tanggal	: 6 Februari 2018
Tempat	: Ruang Kaprodi Hadits
Narasumber	: Bapak Rifqi Muhammad Fatkhil, MA.
Jabatan	: Ketua Program Studi Hadits.

1. Penulis : Menurut Zainul Arifin dalam bukunya yang berjudul “*Studi Kitab Hadits*” mengatakan bahwa “*Al-Qur'an* memperoleh perhatian penuh dari Rasulullah maupun dari sahabat. Beda halnya dengan *hadits*, walaupun *hadits* merupakan sumber yang penting, namun tidak mendapatkan perhatian yang sedemikian.”. Apakah benar? Mengapa demikian?

**Narasumber** : Keliru besar. Karena justru *hadits* mendapatkan perhatian lebih daripada *al-qur'an*. Kenapa? Karena hadits itu berisi hal-hal yang praktis tentang tata cara *shalat*, tentang bagaimana berhubungan dengan sesama dan sebagainya yang tidak terdapat didalam *al-qur'an*. Kalau sahabat tidak semuanya perhatian terhadap *al-qur'an* dari 12.000 kurang lebih jumlah sahabat, tetapi mereka lebih perhatian pada perilaku Nabi dan ucapan

Nabi bukan pada teks-teks *al-qur'an*, karena *al-qur'an* bahkan belum dibukukan pada saat itu.

- 2. Penulis** : Apakah benar bahwa saat ini tidak sedikit orang yang keliru dalam memahami hadits? Mengapa demikian?

**Narasumber** : Betul. Karena, 1). Mereka menganut pemahaman literalis (tekstual). 2). Semangat keberagaman yang ingin kembali mengamalkan ajaran agama yang otentik sesuai dengan *al-qur'an* dan hadits tanpa didukung oleh literasi terhadap pendapat-pendapat ulama. 3). Keterbatasan kemampuan (bahasa) yang mereka miliki sehingga mereka lebih memilih untuk mengakses buku-buku terjemah saja sehingga tidak mewakili konten hadits dari apa yang disampaikan Nabi.

- 3. Penulis** : Untuk mengetahui makna yang sebenarnya dari suatu hadits apakah cukup hanya dengan mengandalkan teks lahiriyah hadits saja?

**Narasumber** : Sangat tidak cukup. Kecuali *hadits-hadits* yang berkaitan dengan akidah. Contoh: Allah itu satu, itu cukup tanpa penjelasan. Tapi kalau yang namanya memahami *hadits*, banyak faktor yang harus diperhatikan.

- 4. Penulis** : Bagaimana atau cara apa yang dapat kita tempuh untuk mengetahui makna hadits dengan sebenar-benarnya?

**Narasumber** : Mempelajari ilmu *Ma'anil Hadits*, Metode Pemahaman *Hadits*, Mempelajari *Syarah Hadits* atau Ilmu *Fiqhul Hadits*.

- 5. Penulis** : Apakah mengetahui atau mempelajari *syarah hadits* untuk mengetahui makna sebenarnya yang terkandung, urgensi untuk dilakukan?

**Narasumber** : Sangat penting. Sepanjang diperoleh dari orang yang memiliki otoritas dibidang penjelasan hadits, bukan hanya dari terjemahannya saja tanpa penjelasan maknanya.

6. **Penulis** : Sejauh yang bapak tahu, apakah sudah ada aplikasi yang memuat *syarah hadits* dari kumpulan *hadits* imam bukhari-muslim?

**Narasumber** : Bahasa arab banyak. Tapi yang berbahasa indonesia belum ada. Baru ada lidwa dan arbain, namun masih sangat luas jangkauannya dari berbagai perawi. Sehingga untuk aplikasi *syarah hadits* dari shahih bukhari-muslim belum ada.

7. **Penulis** : Menurut bapak, setujukah apabila dibuat suatu sistem yang dapat digunakan untuk mengetahui *syarah hadits* dengan berbasis web agar dapat diakses oleh siapapun, kapanpun dan dimanapun?

**Narasumber** : Setuju.

8. **Penulis** : Apa yang diharapkan dari aplikasi yang akan dibuat?

**Narasumber** : Buatlah lebih spesifik, jangan terlalu luas atau besar jangkauannya. Agar aplikasi yang akan dibuat terarah. Gunakan perawi tertentu atau hadits-hadits pilihan dari kitab tertentu untuk membatasinya.

Narasumber,



(.....)

NIP. 19770120 200312 1 003

### Lampiran 3. Transkip Wawancara Pengujian

#### TRANSKIP WAWANCARA PENGUJIAN

Judul Skripsi	: Penerapan Metode <i>Term Frequency Inverse Document Frequency</i> dan <i>Cosine Similarity</i> pada Sistem Temu Kembali Informasi untuk Mengetahui <i>Syiah Hadits</i> Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim).
Waktu	: 14.40 WIB.
Tanggal	: 6 Februari 2018
Tempat	: Ruang Kaprodi Hadits
Narasumber	: Bapak Rifqi Muhammad Fatkhil, MA.
Jabatan	: Ketua Program Studi Hadits.

Wawancara pengujian ini dilakukan setelah sistem yang dibuat telah selesai. **Hasil wawancara penulis dengan narasumber adalah sebagai berikut:**

1. **Penulis** : Apakah aplikasi ini mudah digunakan?  
**Narasumber** : Hanya bisa digunakan oleh orang yang mempunyai atau mengetahui terjemahan hadits secara lengkap. Oleh karena itu, menurut saya aplikasi kurang *User Friendly*.
  
2. **Penulis** : Apakah tampilan dari aplikasi ini menarik?  
**Narasumber** : Kurang menarik.
  
3. **Penulis** : Apakah aplikasi ini sudah berjalan dengan baik sesuai fungsinya?

**Narasumber** : Fungsi semua fitur sudah berjalan dengan baik, dan sistem sudah berhasil memberikan *syarah hadits* sebagai *output* ketika *user* memasukkan *hadits* berbahasa indonesia.

4. **Penulis** : Apakah aplikasi ini bermanfaat?

**Narasumber** : Pasti bermanfaat, karena dengan aplikasi ini orang awam pun dapat mengetahui makna yang terkandung dalam hadits dengan membaca *syarah hadits*-nya.

5. **Penulis** : Apakah aplikasi ini sudah memenuhi kebutuhan dalam mempelajari *syarh hadits* sesuai dengan kitab yang digunakan?

**Narasumber** : Sudah sesuai tergantung *souce*-nya.

