



# **PENERAPAN METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK MENGKLASIFIKASI TEKS TERHADAP HADITS PALSU**

## **TUGAS AKHIR**

Diajukan untuk Memenuhi Salah Satu Syarat  
dalam Menempuh Ujian Sidang Sarjana  
di Program Studi Informatika

Oleh :

**Asep Maulana Ismail**

**0617124001**



**JURUSAN INFORMATIKA  
FAKULTAS TEKNIK UNIVERSITAS WIDYATAMA  
BANDUNG  
2019**

**LEMBAR PENGESAHAN**  
**PENERAPAN METODE *SUPPORT VECTOR MACHINE* (SVM)**  
**UNTUK MENGLASIFIKASI TEKS TERHADAP HADITS**  
**PALSU**

**TUGAS AKHIR**

Program Studi Informatika  
Fakultas Teknik  
Universitas Widyatama

Oleh :

**Asep Maulana Ismail**

**0617124001**

Telah disetujui dan disahkan di Bandung, Tanggal \_\_ / \_\_ / 2019

Pembimbing,

---

NIP: .....

# LEMBAR PENGESAHAN

Nama Mahasiswa : .....  
NIM/NIRM : .....  
Judul Skripsi : .....  
.....  
.....

Telah sidang pada tanggal :

Yang bersangkutan dinyatakan lulus dengan nilai

PENILAI	TANDA TANGAN
Pembimbing: 1. 2.	
Penguji: 1. 2.	

Mengetahui,  
Ketua Program Studi Informatika,

Bandung, .....

Ketua Sidang

\_\_\_\_\_

\_\_\_\_\_,

# SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : .....

Tempat dan Tanggal  
Lahir : .....

Alamat Orang Tua : .....

Menyatakan bahwa skripsi ini adalah benar dan hasil karya saya sendiri. Bila terbukti tidak demikian, saya bersedia menerima segala akibatnya, termasuk pencabutan kembali gelar Sarjana Teknik yang telah saya peroleh.

Bandung, .....

Asep Maulana Ismail

# ABSTRAK

Pada bagian ini dituliskan secara ringkas apa yang telah dikerjakan pada KP/TA.

Ringkasan biasanya memuat :

- apa yang telah dikerjakan
- hasil yang dicapai
- kesimpulan secara ringkas
- jika menyangkut implementasi program, tuliskan platform dan development tools yang dipakai pada pengembangan
- kata kunci, untuk memudahkan orang mencari di perpustakaan.
- Dibat dalam dua bahasa (Indonesia dan Inggris)

**Kata kunci :**

# KATA PENGANTAR

Dengan menyebut nama Allah SWT yang maha pengasih lagi maha penyayang, segala puji dan syukur penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat-Nya sehingga dapat menuntaskan laporan tugas akhir yang berjudul **“PENERAPAN METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK MENGLASIFIKASI TEKS TERHADAP HADITS PALSU”**.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun dari semua pihak sangat penulis harapkan guna perbaikan penulis pada masa yang akan datang.

Untuk itu dengan segala ketulusan, penulis menghaturkan terima kasih kepada:

1. **Kedua Orangtua**, yang selalu memberikan dorongan, kasih sayang, dan doa untuk kelancaran dalam penulisan laporan ini.
2. **Iwa Ovyawan Herlistiono, S.T., M.T.**, selaku Dosen pembimbing kampus yang telah memberikan bimbingan, terima kasih atas kritik dan saran Bapak dalam penyusunan laporan ini.
3. **Sriyani Violina, S.T., M.T.**, selaku Ketua Program Studi Informatika Fakultas Teknik Universitas Widyatama.
4. Pihak-pihak lain yang tidak bisa saya sebutkan satu persatu, terima kasih atas semua bantuan dan motivasinya.

Akhir kata semoga Allah SWT membalas segala kebaikan dari semua pihak yang telah membantu dalam penyusunan Laporan Kerja Praktek ini.

Bandung, ..... 2019

Asep Maulana Ismail

## DAFTAR ISI

## DAFTAR TABEL



## **DAFTAR GAMBAR**

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Hadits merupakan sumber dasar hukum kedua setelah Al-Qur'an. Oleh karena itu mengetahui suatu hadits tersebut adalah hadits *sahih* dan *non-sahih* yang termasuk di dalamnya yaitu *dhaif* (lemah) dan *maudhu* (palsu) sangatlah penting. Hadits itu adalah seluruh perkataan, perbuatan, ketetapan, serta persetujuan Nabi Muhammad S.A.W yang menjadi salah satu dasar pengambilan hukum dalam Islam [1].

Mempelajari dan mengamalkan sebuah hadits sangatlah penting bagi seorang Muslim, karena hadits merupakan salah satu sumber hukum dalam kehidupan sehari-hari. Beberapa peneliti telah melakukan penelitian tentang penerapan hadits pada kehidupan manusia. Bahkan ada beberapa peneliti yang membandingkan buku hadits yang diterbitkan oleh ulama. Tujuannya adalah untuk mencari kebenaran hadits-hadits tersebut dan menerapkannya dalam kehidupan sehari-hari [2].

Berkembangnya teknologi saat ini berdampak kepada derasnya arus informasi yang didapatkan, terutama melalui sosial media. Salah satu informasi yang mudah menyebar yaitu informasi mengenai sebuah hadits. Dengan berharap mendapatkan nilai ibadah, banyak yang menyebarkan kembali informasi hadits yang ia terima. Namun bila tidak diimbangi dengan ketelitian dalam memilih dan memilah hadits mana saja yang bersifat *sahih* dan tidak, dapat beresiko kepada menyebarnya hadits palsu di masyarakat.

Studi tentang hadits ini telah dilakukan oleh para peneliti secara konvensional dengan membandingkan satu hadits dengan yang lain [3]. Penelitian menggunakan literatur tentang hadits, dengan menerapkan metode sosiologis untuk memeriksa penyebaran dari sebuah hadits dan untuk mendokumentasikan apa yang membuat suatu hadits lebih menonjol di masanya [4].

Pada penelitian sebelumnya yang dilakukan oleh Agung B. Prasetyo pada tahun 2017 yang berjudul “*Hoax Detection System on Indonesian News Sites Based on Text Classification using SVM dan SGD*” menyebutkan bahwa *Support Vector Machine* atau SVM merupakan salah satu algoritma *supervised learning* yang baik digunakan untuk klasifikasi sebuah teks yang memiliki dimensi *input* yang banyak. Dari 200 data uji yang digunakan

dalam penelitian tersebut yang terdiri dari 100 berita hoax dan 100 berita asli, SGD mendeteksi 48 berita hoax sebagai berita asli, sedangkan SVM mendeteksi 38 berita hoax sebagai berita asli [5]. Ini berarti metode SVM memiliki *error-rate* yang lebih kecil dibandingkan dengan SGD.

Adapun penelitian lain yang dilakukan oleh Ina Najiyah pada tahun 2017 yang berjudul “*Hadith Degree Classification for Shahih Hadith Identification Web Based*”, memperkuat permasalahan yang sudah diutarakan diatas bahwa permasalahan tersebut yang harus dan mampu dipecahkan oleh teknologi.

Berdasarkan hal tersebut maka dilakukan penelitian tugas akhir yang berjudul “**PENERAPAN METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK MENGLASIFIKASI TEKS TERHADAP HADITS PALSU**”.

## **1.2. Rumusan Masalah**

Berdasarkan pada permasalahan yang telah disebutkan pada bagian latar belakang masalah, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana membuat sistem klasifikasi teks yang dapat menentukan apakah suatu hadits termasuk hadits palsu atau bukan berdasarkan karakteristik dari teks hadits dengan menggunakan algoritma SVM?
2. Bagaimana tingkat akurasi algoritma SVM dalam mengklasifikasi hadits palsu ?

## **1.3. Tujuan**

Penelitian pada permasalahan yang telah dijelaskan pada bagian latar belakang masalah, bertujuan sebagai berikut:

1. Untuk membuat sistem klasifikasi teks yang dapat menentukan apakah suatu hadits termasuk hadits palsu atau bukan berdasarkan karakteristik dari teks hadits dengan menggunakan algoritma SVM.
2. Untuk mengetahui tingkat akurasi algoritma SVM dalam mengklasifikasi hadits palsu.

## **1.4. Batasan Masalah**

Adanya batasan masalah dalam pembuatan sistem klasifikasi hadits palsu , yaitu:

1. Klasifikasi hadits terbagi menjadi tiga, yaitu hadits *shahih*, *dhaif*, dan *maudhu* (palsu).
2. Data hadits sahih, dhaif, dan maudhu menggunakan hadits-hadits yang diambil dari kitab hadits baik berupa fisik maupun digital.
3. Hadits yang digunakan adalah hadits berbahasa Indonesia.

## **1.5. Metodologi Penelitian**

Metode penelitian yang dilakukan dalam penyusunan laporan tugas akhir ini adalah eksperimen. Metodologi penelitian yang diterapkan dalam penulisan tugas akhir ini meliputi:

### **1.5.1. Studi Literatur**

Pada tahap studi literatur, mempelajari literatur dan mengumpulkan data yang berkaitan dengan penelitian tugas akhir. Pengumpulan data tersebut dilakukan dengan beberapa cara diantaranya mencari referensi di perpustakaan maupun di *internet* yang berupa *paper*, jurnal, dan buku-buku.

### **1.5.2. Pengembangan Sistem Klasifikasi**

Pada tahap ini akan dilakukan pengembangan sistem klasifikasi teks terhadap hadits palsu. Pengembangan sistem akan menggunakan metode waterfall. Sebelum dilakukan pengembangan, akan dilakukan tahap pengumpulan data dan *data preprocessing* terlebih dahulu.

#### **1.5.2.1. Pengumpulan Data**

Dalam usaha untuk mendapatkan data yang benar sehingga tercapai maksud dari tujuan penelitian tugas akhir ini, penulis melakukan pengumpulan data hadits asli melalui internet dengan bersumber kepada kitab hadits baik berupa fisik maupun digital. Proses pengumpulan data ini harus dilakukan diawal karena akan melibatkan data yang banyak sehingga membutuhkan waktu yang cukup lama. Setelah semua hadits dikumpulkan, maka akan dibuat *dataset* hadits yang nantinya akan digunakan pada tahap selanjutnya.

### **1.5.2.2. Data Preprocessing**

Setelah mendapatkan data yang dibutuhkan dalam penelitian ini, tahap selanjutnya adalah *data preprocessing* yaitu membersihkan data dari noise yang dapat mempengaruhi hasil klasifikasi.

### **1.5.2.3. Analisis**

Analisis yang akan dilakukan meliputi analisis proses klasifikasi, analisis sistem yang akan dibangun, dan analisis kebutuhan sistem.

#### **1.5.2.3.1. Analisis Proses Klasifikasi**

Pada tahap ini, akan dijelaskan tentang mekanisme proses klasifikasi termasuk juga apa yang akan dilakukan dengan data yang telah terkumpul. Analisis proses klasifikasi ini meliputi proses training dan proses testing.

#### **1.5.2.3.2. Analisis Sistem yang Akan Dibangun**

Tahap ini dilakukan dengan tujuan untuk mengetahui ruang lingkup dari sistem yang akan dibangun, sehingga akan mempermudah dalam mendefinisikan kebutuhan sistem.

#### **1.5.2.3.3. Analisis Kebutuhan Sistem**

Tahap ini dilakukan untuk mendefinisikan kebutuhan-kebutuhan sistem yang akan dibangun. Pada tahap analisis ini akan dihasilkan dua jenis kebutuhan sistem, yaitu kebutuhan fungsional dan kebutuhan non-fungsional atau kebutuhan tambahan.

### **1.5.2.4. Perancangan**

Pada tahap ini, akan dilakukan perancangan sistem yang akan dibangun. Perancangan ini bertujuan untuk memodelkan sistem yang akan dibangun untuk mendukung penelitian yang akan dilakukan. Pemodelan perancangan yang akan digunakan adalah menggunakan pendekatan Object Oriented. Perancangan sistem dimulai dengan pembuatan Use Case Diagram untuk menjelaskan apa saja kemampuan yang akan ada pada sistem. Selanjutnya adalah Sequence Diagram dan Activity Diagram untuk menjelaskan perilaku sistem. Lalu akan dilanjutkan dengan Entity Relationship Diagram untuk menggambarkan bagaimana

masing-masing entity yang ada pada sistem saling terhubung. Selain itu juga akan dibuatkan perancangan User Interface dengan menggunakan Command Line Interface (CLI).

#### **1.5.2.5. Implementasi**

Pada tahap ini akan dilakukan implementasi sistem klasifikasi teks terhadap hadits palsu. Implementasi dilakukan dengan mengeksekusi perancangan sistem yang telah dibuat pada tahap sebelumnya. Sistem yang dibangun digambarkan dengan Deployment Diagram. Selain itu juga akan disertakan tentang spesifikasi komputer yang digunakan untuk dapat melakukan implementasi sistem dan requirements apa saja yang telah berhasil diimplementasi. Lalu juga tidak lupa pada tahap ini, akan dijelaskan tentang implementasi user interface dari sistem, dan implementasi metode SVM itu sendiri.

#### **1.5.3. Pengujian**

Pada tahap ini akan dilakukan percobaan untuk melihat bagaimana tingkat akurasi dari metode SVM ini dalam mengklasifikasi teks terhadap hadits palsu. Pengujian dilakukan diawali dengan pembuatan skenario pengujian. Setelah itu dilakukan pengukuran terhadap tingkat akurasi metode SVM ini. Lalu selanjutnya yaitu pembahasan mengenai hasil eksperimen yang telah dilakukan.

#### **1.5.4. Kesimpulan**

Pada tahap ini akan dibuat kesimpulan berdasarkan hasil dari penelitian yang telah dilakukan berdasarkan hasil keseluruhan dari setiap tahap dalam penelitian ini, serta menambahkan saran untuk penelitian atau pengembangan penelitian selanjutnya.

### **1.6. Sistematika Penulisan**

Sistematika penulisan laporan penelitian tugas akhir ini terdiri dari beberapa bab. Masing-masing bab memiliki isi sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, metode penelitian dan sistematika penulisan.

#### **BAB II LANDASAN TEORI**

Bab ini berisi tentang teori - teori singkat yang menunjang dalam laporan penelitian tugas akhir ini.

### **BAB III PENGEMBANGAN SISTEM KLASIFIKASI**

Bab ini berisi tentang bagaimana pembangunan sistem klasifikasi teks terhadap hadits palsu dibuat. Mulai dari pengumpulan data, data preprocessing, analisis, perancangan, hingga implementasi sistem.

## BAB II

### LANDASAN TEORI

#### 2.1. Hadits

Al-Qur'an dan Hadits merupakan inti dari ajaran Islam. Sunnah memainkan peran penting dalam pengembangan seluruh kehidupan manusia dan peradaban [6]. Hadits merupakan segala sesuatu yang berasal dari Nabi SAW baik itu berupa perbuatan, ucapan, persetujuan yang menjadi dasar pengambilan hukum dalam Islam [1].

Hadits dibagi menjadi dua jenis yaitu hadits *Mutawatir* dan hadits *Ahad*. Hadits *Mutawatir* adalah hadits yang keasliannya sudah tidak diragukan lagi, sedangkan hadits *Ahad* adalah hadits yang derajatnya belum mencapai level seperti hadits *Mutawatir* [2].

Derajat dari sebuah hadits dibagi menjadi tiga, yaitu Hadits *Shahih*, *Dhaif*, dan *Maudhu*. Hadits *Shahih* adalah hadits yang diriwayatkan oleh seorang rawi (periwayat) dengan *sanad* yang tidak terputus. Sedangkan hadits *Dhaif* yaitu hadits yang tidak memiliki syarat-syarat hadits *Hasan* (baik) dikarenakan hilangnya beberapa kondisi seperti *sanad*-nya terputus (ujung *sanad* tidak merujuk kepada Nabi SAW) atau terdapat faktor lain seperti *matan* (redaksi) hadits itu sendiri. Selain itu ada juga yang disebut dengan hadits *Maudhu* (palsu), yaitu hadits yang dibuat oleh seseorang selain Nabi SAW dengan mengatasnamakan Nabi SAW [2].

#### 2.2. Supervised Learning

*Artificial Intelligence* (AI) merupakan sebuah teknik mengadaptasi pola pikir manusia atau tingkah laku manusia kepada sistem komputer. AI memiliki empat teknik dasar dalam pemecahan suatu masalah, yaitu: *Searching*, *Reasoning*, *Planning*, dan *Learning*. Dari masing-masing teknik tersebut terdapat berbagai macam metode yang dapat digunakan yang telah diusulkan oleh para pakar AI. Masing-masing metode memiliki kelebihan dan kekurangan. Sebuah metode yang bekerja sangat baik untuk permasalahan A belum tentu dapat digunakan untuk permasalahan B. Dengan kata lain, penggunaan metode ini sangat bergantung kepada permasalahan apa yang sedang terjadi [7].

*Machine Learning* (ML) merupakan sebuah bidang pada *Computer Science* dalam memberikan kemampuan belajar kepada sebuah komputer tanpa diprogram secara eksplisit.



Dalam mempelajari sebuah permasalahan, ML memerlukan sebuah model yang sudah didefinisikan berdasarkan parameter-parameter tertentu. Proses learning pada ML adalah proses dimana *program* mengoptimasi parameter-parameter pada model tersebut dengan memanfaatkan *data training* atau *past experiences* [8].

ML melibatkan berbagai macam bidang ilmu seperti statistika, ilmu komputer, matematika, dan bahkan neurologi. ML menggunakan teori-teori statistika untuk membuat model matematika. Model dapat bersifat *predictive* yaitu untuk memprediksi masa depan, *descriptive* yaitu untuk memperoleh pengetahuan baru dari data, atau bahkan gabungan dari keduanya. Inti dari ML adalah bagaimana membuat sebuah komputer dapat menyelesaikan berbagai macam permasalahan dengan mempelajari data (*past experiences*) sebagaimana manusia mempelajari sesuatu [8]. Salah satu bagian dari ML ini adalah *Supervised Learning*.

*Supervised Learning* adalah sebuah cara belajar dengan mencari hubungan (*mapping function*) dari *input data* (x) yang sudah diberi *label* (y). Permasalahan yang berkaitan dengan *supervised learning* dapat dikategorikan menjadi dua jenis, yaitu:

1. *Classification*

*Classification* bertujuan untuk mencari *outcome* dari *input* yang sudah diberikan, dimana output berupa kategori-kategori dari input data. Contoh: pria/wanita, tinggi/rendah, baik/buruk, dan sebagainya.

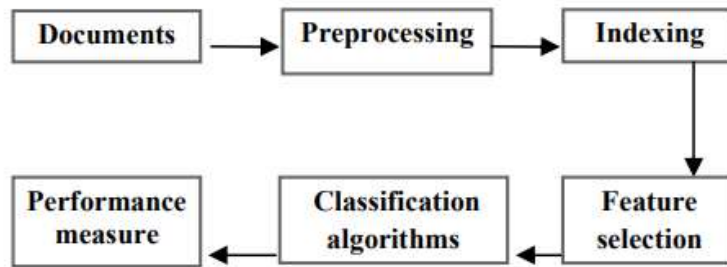
2. *Regression*

*Regression* bertujuan untuk mencari *outcome* dari *input* yang sudah diberikan, dimana output berupa prediksi yang berbentuk nilai aktual. Contoh: tinggi badan seseorang, curah hujan, dan sebagainya [8].

## 2.3. Text Classification

Text Classification (TC) adalah proses pemberian label terhadap sekumpulan teks berdasarkan isi dari konten teks tersebut. Contoh dari text classification adalah pemberian label ‘olahraga’, ‘politik’, ‘kesehatan’, dan lain-lain kepada sebuah berita. Text classification merupakan bagian yang sangat penting dalam dunia Text Mining [9].

Text classification memiliki proses sebagai berikut:



Gambar 1. Proses text classification [9]

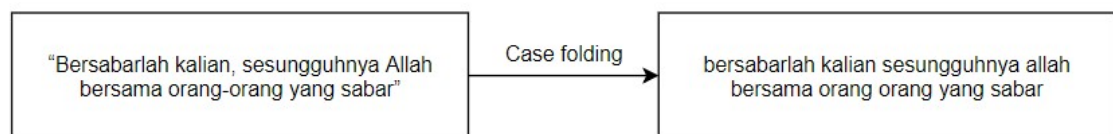
### 2.3.1. Documents Collecting

Documents Collecting adalah proses pengumpulan dokumen dari berbagai format seperti .html, .pdf, .doc, dan lain sebagainya [9]. Dokumen-dokumen tersebut nantinya akan digunakan sebagai data.

### 2.3.2. Preprocessing

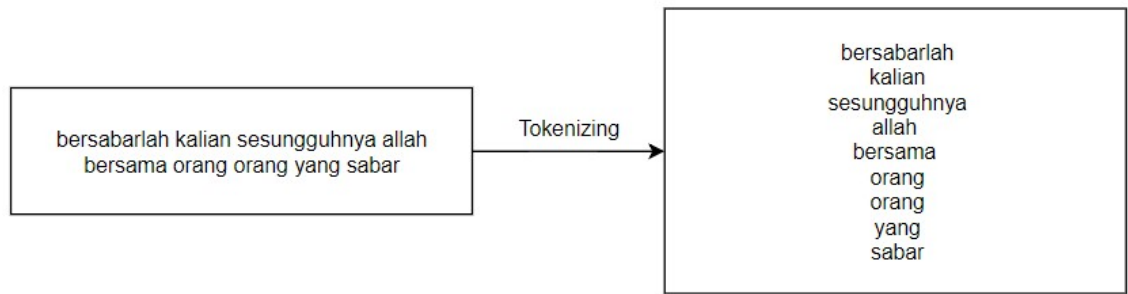
Salah satu bagian yang terpenting dari text classification ini adalah tahap preprocessing. Tahap preprocessing menurut Triawati, terdiri dari empat tahapan, yaitu: case folding, tokenizing/parsing, filtering dan stemming [10].

Case folding merupakan tahapan yang mengubah semua huruf dalam dokumen menjadi huruf kecil. Karakter yang diperbolehkan hanya karakter 'a' hingga 'z'. Karakter selain itu akan dihilangkan dan dianggap sebagai pembatas (delimiter) [10]. Berikut adalah contoh dari proses case folding:



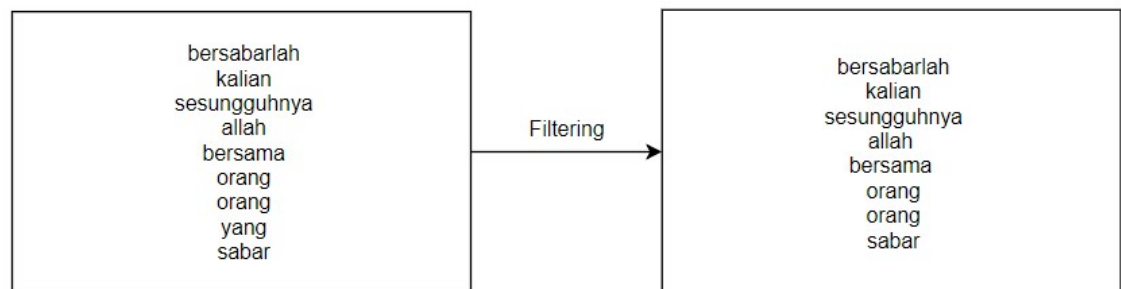
Gambar 2. Proses case folding

Setelah melewati proses case folding, maka proses selanjutnya adalah tokenizing. Tokenizing adalah tahap mengubah sebuah teks menjadi kumpulan kata yang menyusunnya [10]. Berikut adalah contoh dari proses tokenizing:



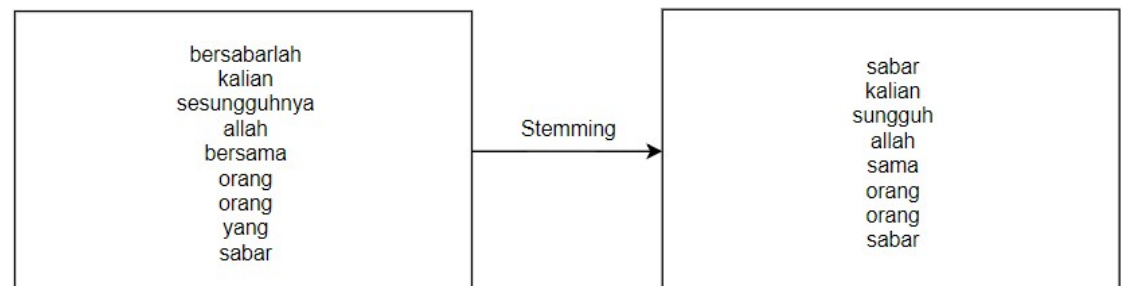
Gambar 3. Proses tokenizing

Setelah itu, tahap selanjutnya adalah proses filtering. Proses filtering ini bertujuan untuk menghilangkan kata-kata yang tidak memiliki makna yang ada pada sebuah kalimat, atau lebih dikenal dengan sebutan stop words. Stop words adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh dari stop words adalah ‘yang’, ‘dan’, ‘di’, ‘dari’, dan seterusnya [10]. Berikut contoh dari proses filtering:



Gambar 4. Proses filtering

Langkah terakhir dari tahap preprocessing adalah stemming. Stemming adalah proses mengubah kata berimbuhan menjadi kata dasar. Berikut contoh dari proses stemming:



Gambar 5. Proses stemming

### 2.3.3. Indexing

Indexing adalah proses konversi dokumen dari yang asalnya berbentuk full-text menjadi document vector atau yang lebih dikenal dengan vektor space model. Di dalam vector space model, dokumen direpresentasikan dalam bentuk vector of words. Baris dari matriks vektor tersebut adalah dokumen, sedangkan kolom dari matriks vektor tersebut adalah kata/term [9].

$$\begin{pmatrix} T_1 & T_2 & \dots & T_{at} & C_i \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} C_1 \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} C_2 \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \dots & w_{tn} C_n \end{pmatrix}$$

Gambar 6. Vector of words [10]

#### 2.3.4. Feature Selection

Feature selection adalah proses untuk membentuk vector space yang lebih unggul dari segi skalabilitas, efisiensi, dan akurasi. Ide utama dari feature selection ini adalah dengan mengambil sebagian fitur dari fitur-fitur yang ada pada sebuah dokumen. Hal tersebut dilakukan dengan tetap menyimpan fitur yang memiliki nilai tertinggi, dan menghilangkan fitur yang memiliki nilai yang rendah berdasarkan seberapa penting kata tersebut ada pada sebuah dokumen [9]. Dan untuk memberikan bobot pada masing-masing fitur, bisa menggunakan TF-IDF (Term Frequency – Inverse Document Frequency).

TF-IDF adalah salah satu metode pembobotan yang sangat disarankan dalam dunia text classification. Fitur pertama akan diberikan bobot lokal dengan menghitung *tfi*, formula dan pembobotan global akan dihitung menggunakan *idf*. *Tfi* adalah jumlah kemunculan kata tersebut dalam suatu dokumen, sedangkan *idf* adalah jumlah kemunculan kata tersebut pada seluruh dokumen [5].

$$idf_i = \log (N/df_i) \quad (1)$$

$$w_i = tf_{i,j} \times idf_i \quad (2)$$

Untuk mendapatkan nilai bobot yang selalu positif, maka persamaan rumus TF-IDF diubah menjadi:

$$w_i = tf_{i,j} \times (idf_i + 1) \quad (3)$$

Keterangan:

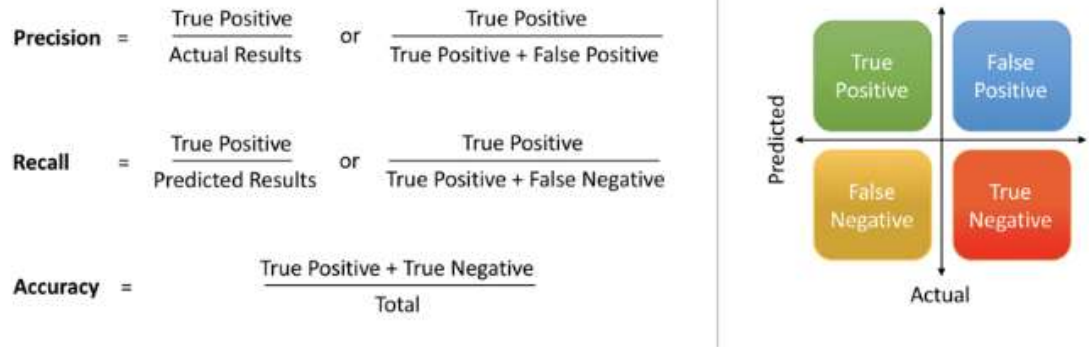
- $W_i$  : bobot kata *term* terhadap dokumen
- $Tf_{i,j}$  : jumlah kemunculan kata/*term* terhadap dokumen
- $N$  : jumlah semua dokumen yang ada dalam *database*
- $df_i$  : jumlah dokumen yang mengandung kata/*term*

### 2.3.5. Classification Algorithms

Sebuah dokumen dapat diklasifikasi melalui tiga cara, yaitu metode unsupervised, supervised, dan semi supervised. Salah satu penerapan text classification ini terdapat pada bidang Machine Learning (ML) dengan menggunakan beberapa algoritma, yaitu: Bayesian classifier, Decision Tree, K-nearest neighbor(KNN), Support Vector Machines (SVMs), Neural Networks, dan masih banyak lainnya [9].

### 2.3.6. Performance Measure

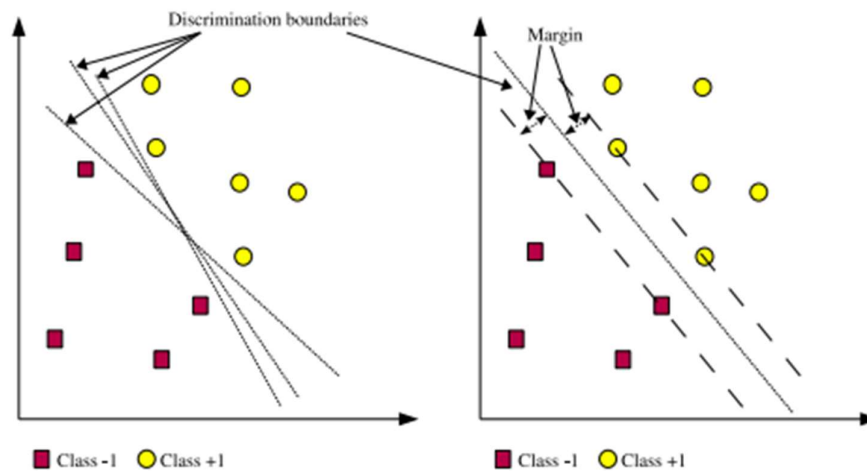
Ini merupakan tahap terakhir dari text classification, dimana evaluasi dari text classification ini lebih cenderung dilakukan secara eksperimental, bukan secara analitis. Tahap evaluasi ini biasanya berfokus kepada efektivitas, yaitu kemampuan dalam mengambil keputusan yang tepat. Langkah yang dapat dilakukan untuk mengukur performance dari text classification salah satunya adalah dengan metode Precision and Recall [9].



Gambar 7. Formula Precision and Recall [11]

## 2.4. Support Vector Machine (SVM)

*Support Vector Machine* (SVM) adalah salah satu algoritma *supervised learning* yang bekerja dengan baik untuk kasus klasifikasi teks yang memiliki banyak input berdasarkan pada teks sebagai fiturnya. Ide utama dari algoritma SVM adalah untuk membuat *hyper plane* yang memisahkan area menjadi beberapa bagian [5]. *Hyper plane* adalah sebuah garis yang memisahkan dua kategori klasifikasi.



Gambar 8. SVM berusaha menemukan *hyper plane* terbaik dalam memisahkan kedua *class -1* dan *+1* [12].

### 2.5.1. Linear Support Vector Machine

Dapat dilihat pada Gambar 1, bahwa terdapat dua kelas yang dipisahkan oleh sebuah garis *hyper plane*. Masalah yang dihadapi dalam klasifikasi ini yaitu bagaimana menemukan

garis (*hyper plane*) yang memisahkan antara kedua kelompok tersebut secara sempurna. Akan banyak alternatif garis yang memisahkan kedua kelas tersebut, namun *hyper plane* terbaik yang dapat memisahkan kedua kelas tersebut adalah *hyper plane* yang memiliki *margin* paling besar [12].

Margin adalah jarak antara hyper plane dengan vektor terdekat dari masing-masing kelas. Vektor yang paling dekat dengan hyper plane disebut dengan *support vector*. Pada Gambar 1 sebelah kanan dapat dilihat bahwa terdapat garis solid yang merupakan hyper plane terbaik, karena tepat berada di tengah antara kedua kelas. Usaha dalam mencari lokasi hyper plane ini merupakan inti dari proses pembelajaran pada SVM [12]

Setiap data dinotasikan sebagai  $x_i \in \mathbb{R}^D$ ,  $i = 1, 2, \dots, N$ .  $N$  adalah banyaknya data. Kelas positif dinotasikan sebagai  $+1$ , dan kelas negatif dinotasikan sebagai  $-1$ . Dengan demikian, tiap data dan label kelas nya dinotasikan sebagai :  $y_i \in \{-1, +1\}$  [12].

.Bila diasumsikan kedua kelas tersebut dipisahkan secara sempurna oleh *hyper plane* di  $D$ -dimensional feature space. *Hyper plane* tersebut dapat didefinisikan sebagai berikut:

$$w \cdot x_i + b = 0 \quad (1)$$

Data  $x_i$  yang tergolong kepada data kelas negatif adalah data yang memenuhi kriteria sebagai berikut:

$$w \cdot x_i + b \leq -1 \quad (2)$$

Data  $x_i$  yang tergolong kepada data kelas positif adalah data yang memenuhi kriteria sebagai berikut:

$$w \cdot x_i + b \geq 1 \quad (3)$$

Selanjutnya, masalah ini diformulasikan ke dalam *Quadratic Programming* (QP) problem, dengan meminimalkan persamaan 4 dibawah constraint persamaan 5.

Meminimalkan:

$$\|w\|^2 \quad (4)$$

Bergantung pada:

$$y_i (w \cdot x_i + b) - 1 \geq 0, \forall i \quad (5)$$

Optimasi ini dapat diselesaikan dengan *Lagrange Multipliers*:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b - 1) \quad (6)$$

$\alpha_i$  adalah *Lagrange multiplier* yang berkorespondensi dengan  $x_i$ . Nilai  $\alpha_i$  adalah nol atau positif. Untuk menyelesaikan masalah tersebut, pertama-tama meminimalkan  $L$  terhadap  $w$ , dan memaksimalkan  $L$  terhadap  $\alpha_i$ . Dengan memodifikasi persamaan 6, *maximization problem* di atas dapat direpresentasikan dalam  $\alpha_i$ :

Memaksimalkan:

$$\sum_{i=1}^N \alpha_i - \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (7)$$

Bergantung pada:

$$\alpha_i \geq 0, \sum_{i=1}^N \alpha_i y_i = 0 \quad (8)$$

Solusi dari *problem* ini akan menghasilkan banyak  $\alpha_i$  dengan nilai nol. Data berkorespondensi dengan  $\alpha_i$  yang tidak nol, merupakan *support vectors*, yaitu data yang memiliki jarak terdekat dengan *hyper plane* [12].

## 2.5.2. Soft Margin

Penjelasan yang sudah diutarakan diatas merupakan asumsi bahwa *hyper plane* dapat memisahkan kedua kelas dengan sempurna. Akan tetapi, seringkali ditemukan bahwa kedua buah kelas tersebut tidak terpisah secara sempurna. Hal ini menyebabkan optimasi tidak dapat diselesaikan, karena tidak ada  $w$  dan  $b$  yang memenuhi pertidaksamaan 5 [12].

Maka dari itu, pertidaksamaan 5 dimodifikasi dengan memasukkan slack variable  $\xi_i$  ( $\xi_i \geq 0$ ), menjadi:

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i, \forall i \quad (9)$$

Demikian juga dengan 4, sehingga diperoleh:

*Minimize*

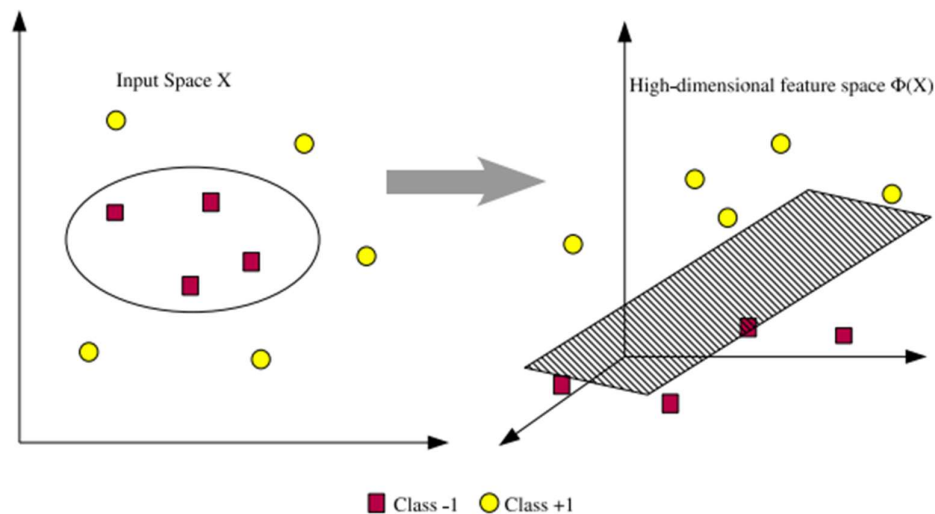


$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (10)$$

Parameter  $C$  bertugas mengontrol tradeoff antara margin dan *classification error*. Semakin besar nilai  $C$ , semakin besar penalty yang dikenakan untuk tiap *classification error* [12].

### 2.5.3. Non Linear Support Vector Machine

Seperti yang sudah dijelaskan diatas, SVM adalah salah satu bentuk dari linear machine, sehingga hanya dapat digunakan untuk menyelesaikan permasalahan-permasalahan linear. Namun tak jarang permasalahan yang dihadapi bukanlah permasalahan linear (non-linear). Untuk mengatasi permasalahan non-linear, seluruh data yang ada pada ruang vektor awal harus dipindahkan ke ruang vektor baru yang memiliki dimensi lebih tinggi [12], seperti yang terlihat pada Gambar 2:



Gambar 9. Memetakan data pada ruang vektor awal ke ruang vektor baru berdimensi lebih tinggi [12]

Selanjutnya adalah melakukan proses training sama seperti yang dilakukan pada Linear SVM. Proses optimasi pada fase ini memerlukan perhitungan dot product dua buah vektor  $x_i$  dan  $x_j$  pada ruang vektor baru yang dinotasikan sebagai  $(x_i) \cdot (x_j)$ . Teknik komputasi ini dikenal dengan sebutan Kernel Trick, yaitu menghitung dot product dua buah vektor di ruang vektor baru menggunakan komponen kedua buah vektor tersebut di ruang vektor asal [12].

Klasifikasi pada non linear SVM dapat dirumuskan sebagai berikut [12]:

$$f(\varphi(x)) = w \cdot \varphi(x) + b \quad (11)$$

Terdapat beberapa jenis fungsi yang dapat digunakan sebagai fungsi Kernel K, diantaranya adalah sebagai berikut:

Tabel 1. Fungsi kernel yang sering digunakan pada SVM [12]

Nama Kernel	Definisi
Polynomial	$K(x_i, x_j) = (x_i \cdot x_j + 1)^p$
Gaussian	$K(x_i, x_j) = \exp(-(\ x_i - x_j\ ^2 / 2\sigma^2))$
Sigmoid	$K(x_i, x_j) = \tanh(x_i \cdot x_j + \beta)$

$$= \sum_{i=1, x_i \in SV}^N \alpha_i y_i \varphi(x) \cdot \varphi(x_i) + b \quad (12)$$

$$= \sum_{i=1, x_i \in SV}^N \alpha_i y_i K(x, x_i) + b \quad (13)$$

*Support vector* SV adalah subset dari *data training*, dengan nilai tidak negatif  $\{x_i \mid \alpha_i\}$  ( $i = 1, 2, \dots, l$ ) [12].

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1. Pengumpulan Data

Langkah pertama dalam pengembangan sistem klasifikasi adalah pengumpulan data. Data yang akan digunakan dalam pengembangan sistem ini adalah data hadits sahih, dhaif dan maudhu yang ditulis dalam bahasa Indonesia. Data yang terkumpul akan dibagi menjadi dua bagian, yaitu data training dan data testing. Data training adalah kumpulan hadits yang akan digunakan untuk pembelajaran sistem classifier, sedangkan data testing adalah kumpulan hadits yang akan digunakan untuk menguji akurasi dari sistem classifier.

Tabel 2. Contoh data hadits

Teks Hadits	Status Hadits
<i>Ibnu Umar berkata, "Seorang hamba tidak akan mencapai hakikat takwa yang sebenarnya kecuali ia dapat meninggalkan apa saja yang dirasa tidak enak dalam hati."</i>	Sahih
<i>"Diamnya orang yang puasa adalah tasbih, tidurnya adalah ibadah, do'a nya mustajab dan amalnya dilipatgandakan."</i>	Dhaif
<i>"Seandainya sekalian hamba mengetahui keutamaan bulan Ramadhan, niscaya mereka berangan-angan agar setiap tahun dijadikan bulan Ramadhan seluruhnya ...."</i>	Maudhu

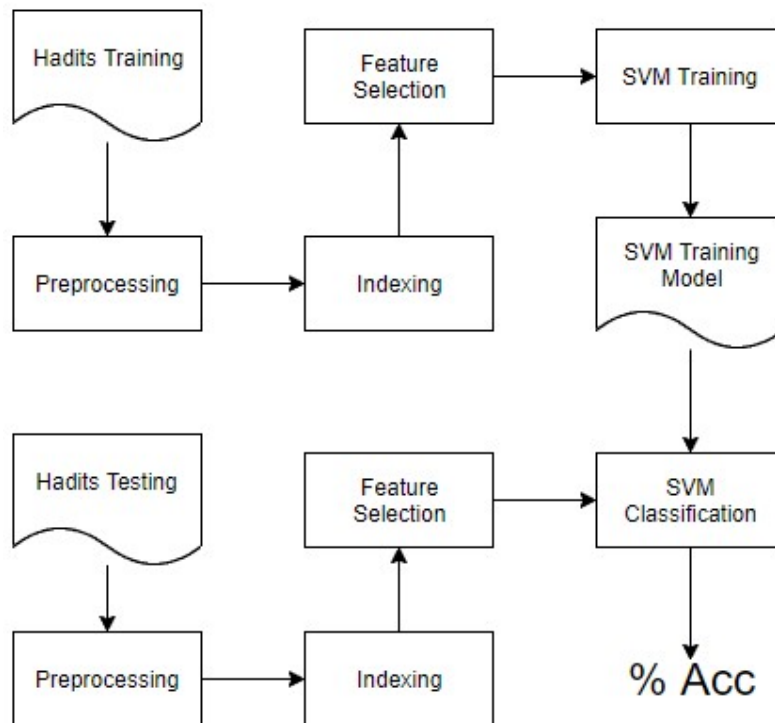
#### 3.2. Normalisasi Data

Pada tahap ini, data yang sudah dikumpulkan akan diproses terlebih dahulu, proses ini disebut dengan normalisasi data. Normalisasi data adalah suatu proses dimana data yang ada akan dibersihkan dari noise. Noise yang dimaksud dapat berupa duplikasi data, baik pada label hadits yang sama maupun pada label hadits yang berbeda. Data tersebut harus dihilangkan salah satu, sehingga setiap hadits hanya akan memiliki satu label.

### 3.3. Analisis

### 3.4. Analisis Proses Klasifikasi

Proses klasifikasi yang akan dilakukan secara umum dapat digambarkan seperti gambar berikut:



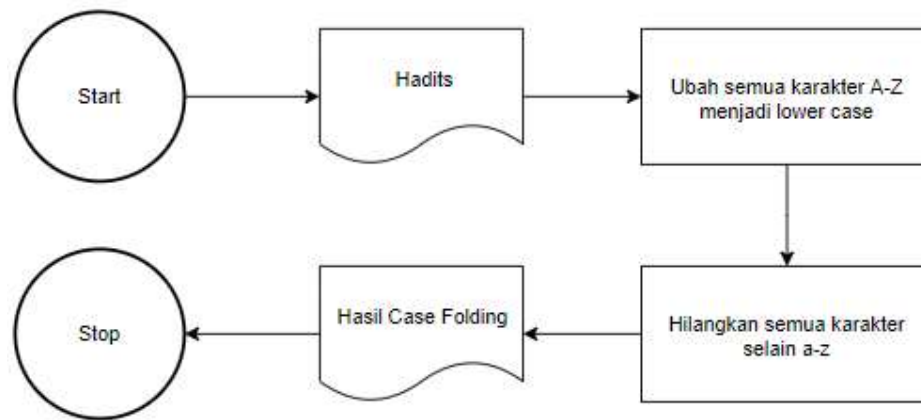
Gambar 10. Proses Umum Klasifikasi

Gambar 10. memperlihatkan bahwa terdapat dua tahapan dalam proses klasifikasi, yaitu tahap training dan tahap testing. Begitu juga dengan data hadits yang telah berhasil dikumpulkan akan dibagi menjadi dua kelompok data, yaitu data training dan data testing.

#### 3.4. 1. Preprocessing

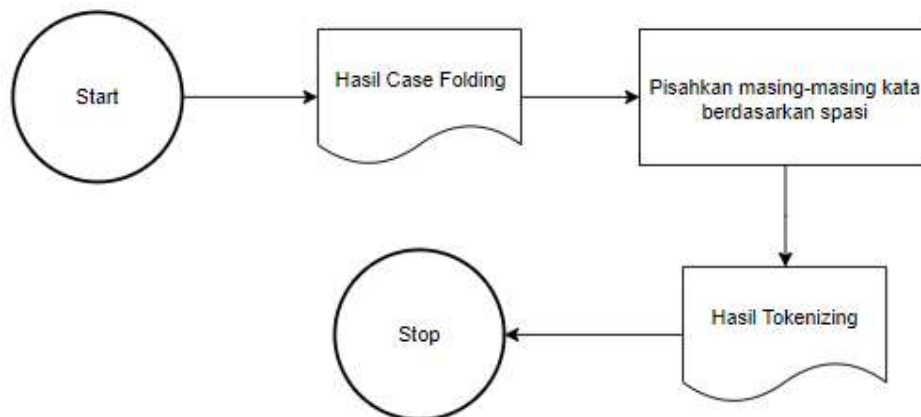
Setiap data hadits yang didapatkan, akan diproses terlebih dahulu pada tahap preprocessing sebelum data tersebut dapat digunakan untuk klasifikasi. Pada tahap preprocessing ini, terdapat beberapa proses yang akan dilakukan yaitu: case folding, tokenizing/parsing, filtering dan stemming.

Case folding adalah proses mengubah semua teks dalam sebuah hadits menjadi lower case. Berikut adalah alur proses case folding:



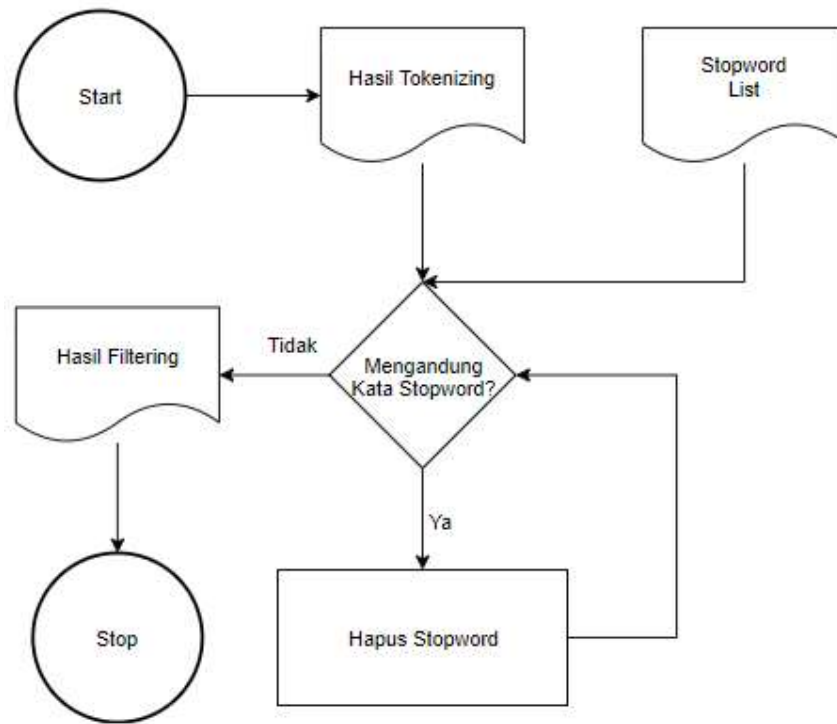
Gambar 11. Alur proses case folding

Tokenizing/parsing adalah proses memecah teks pada sebuah hadits menjadi kumpulan kata. Berikut adalah alur proses tokenizing:



Gambar 12. Alur proses tokenizing

Filtering adalah proses untuk menghilangkan kata-kata yang dianggap tidak memiliki makna. Berikut adalah alur proses filtering:



Alur proses filtering

Stemming adalah proses dimana setiap kata yang ada pada sebuah hadits diubah ke bentuk kata dasarnya. Ada banyak permasalahan yang mungkin terjadi pada proses stemming, diantaranya [13]:

- Imbuhan pada bahasa indonesia memiliki tingkat kekompleks-an yang cukup tinggi, yaitu terdiri dari:
  - Prefiks, yaitu imbuhan yang terdapat di awal kata, contoh: ber-tiga.
  - Suffiks, yaitu imbuhan yang terdapat di akhir kata, contoh: makan-an.
  - Konfiks, yaitu imbuhan yang terdapat di awal dan di akhir kata, contoh: per-ubah-an.
  - Infiks, yaitu imbuhan yang berada di tengah kata, contoh: k-emi-lau.
  - Imbuhan dari serapan bahasa asing, contoh: final-isasi, sosial-isasi.
  - Aturan perubahan prefiks, seperti imbuhan me- menjadi meng-, mem-, men-, dan meny-.
- Word-Sense Ambiguity, yaitu ketika satu kata memiliki lebih dari satu makna, dan berasal dari kata dasar yang berbeda, contoh:
  - Berikan -> Ber-ikan
  - Berikan -> Beri-kan.

- Overstemming, yaitu ketika hasil dari proses stemming suatu kata berimbuhan menghasilkan kata dasar yang salah. Misalkan pada kata “berikan”, dapat dipenggal menjadi “ber-i-kan”, sehingga pada proses akhir hanya akan didapatkan kata “i”, yang mana kata “i” tersebut tidak terdapat pada kata dasar bahasa Indonesia. Untuk mencegah hal ini terjadi, algoritma membutuhkan daftar kata dasar yang akan digunakan pada proses stemming. Sehingga ketika pada kata yang kita penggal terdapat pada daftar kata dasar, maka hentikan proses stemming.
- Understemming, yaitu ketika hasil pemenggalan kata berimbuhan menghasilkan kata dasar yang salah. Misalkan dalam memenggal kata “mengecek”, proses stemming akan menghasilkan penggalan “meng-ecek”, padahal seharusnya adalah “menge-cek”. Hal tersebut terjadi karena kata “ecek” merupakan kata dasar.
- Ketergantungan terhadap daftar kata dasar, sehingga setiap kekurangan atau kelebihan yang ada pada daftar kata dasar akan mempengaruhi hasil proses stemming, bisa berupa overstemming atau understemming.
- Pengguna bahasa Indonesia tidak konsisten dalam menentukan kata dasar secara manual.
- Kata bentuk jamak.
- Kata serapan dari bahasa asing seperti meng-akomodir.
- Kesalahan penulisan yang menyebabkan kata tidak dapat dicari kata dasarnya.
- Akronim atau singkatan, seperti kata “pemilu” akan menghasilkan kata “pe-milu”.
- Proper Noun (Nama Benda), seperti nama orang, dan nama kota. Nama-nama tersebut seharusnya tidak perlu dicari kata dasarnya.

Untuk meminimalisir permasalahan diatas, kita perlu menggunakan kombinasi dari beberapa algoritma stemming untuk bahasa Indonesia. Pertama proses stemming menggunakan algoritma Nazief dan Adriani, kemudian ditingkatkan oleh Algoritma CS (Confix Stripping), kemudian ditingkatkan lagi oleh algoritma ECS (Enhanced Confix Stripping), lalu ditingkatkan lagi oleh Modified ECS. Dengan penggabungan algoritma tersebut, beberapa permasalahan diatas dapat diatasi, yaitu overstemming, understemming, dan kata yang berbentuk jamak [14].

Dibawah ini adalah contoh input dan output yang akan dihasilkan dari masing-masing proses yang ada pada preprocessing:

Tabel 3. Contoh proses preprocessing

Proses	Output
Input awal	Barang siapa yang menyerupai suatu kaum, maka dia termasuk bagian dari kaum tersebut.
Case Folding	Barang siapa yang menyerupai suatu kaum, maka dia termasuk bagian dari kaum tersebut.
Tokenizing	[barang, siapa, yang, menyerupai, suatu, kaum, maka, dia, termasuk, bagian, dari, kaum, tersebut]
Filtering	[barang, menyerupai, kaum, kaum]
Stemming	[barang, rupa, kaum, kaum]

### 3.4. 2. Indexing

Teks hadits yang telah di preprocessing akan diubah ke dalam bentuk vector of words. Berikut adalah salah satu contoh bentuk vector of words:

Tabel 4. Contoh vector of words.

Term\Document	Doc 1	....	Doc-N
barang	1	...	...
rupa	1	...	...
kaum	2	...	...
....	...	...	...
Term - N	...	...	...



### 3.4. 3. Feature Selection

Setelah melakukan indexing, seluruh fitur yang didapat akan diseleksi dengan menggunakan metode TF-IDF. Untuk mendapatkan subset dari  $n$  fitur,  $n$  kata harus dipilih berdasarkan bobot TF-IDF yang tertinggi [15]. Misalkan terdapat tiga buah dokumen/hadits:

- Doc 1 = “Iman memiliki lebih dari enam puluh cabang, dan malu adalah bagian dari iman.”
- Doc 2 = “Tidaklah beriman seorang dari kalian hingga aku lebih dicintainya daripada orang tuanya, anaknya dan dari manusia seluruhnya.”
- Doc 3 = “Tinggalkanlah dia, karena sesungguhnya malu adalah bagian dari iman.”

Dari ke-tiga data tersebut, didapatkanlah matriks term sebagai berikut:

Tabel 5. Matriks term terhadap dokumen

Term\Doc	Doc 1	Doc 2	Doc 3	n
anak	0	1	0	1
cabang	1	0	0	1
cinta	0	1	0	1
daripada	0	1	0	1
hingga	0	1	0	1
iman	2	1	1	3
kalian	0	1	0	1
malu	1	0	1	2
manusia	0	1	0	1
tinggal	0	0	1	1

Kolom n diakhir hanya sebagai informasi tambahan agar lebih memudahkan dalam menghitung bobot TF-IDF. Sebagai contoh, untuk menghitung bobot  $W_{1,2}$ , didapatkan perhitungan sebagai berikut:

$$W_{1,2} = 1 \cdot (\log(3/1) + 1) = 1.47712125472$$

Sehingga didapatkanlah matriks hasil perhitungan bobot dengan TF-IDF sebagai berikut:

Tabel 6. Tabel hasil pembobotan TF-IDF

Term\Doc	Doc 1	Doc 2	Doc 3
anak	0	1.4771212547	0
cabang	1.4771212547	0	0
cinta	0	1.4771212547	0
daripada	0	1.4771212547	0
hingga	0	1.4771212547	0
iman	0	0	0
kalian	0	1.4771212547	0
malu	1.1760912591	0	1.1760912591
manusia	0	1.4771212547	0
tinggal	0	0	1.4771212547

Bisa dilihat dari hasil pembobotan, kata “iman” memiliki bobot 0, itu berarti kata “iman” tersebut terdapat di semua dokumen yang ada sehingga tidak memiliki dampak yang signifikan terhadap klasifikasi. Maka dari itu kata “iman” bisa dihapus agar proses klasifikasi menjadi lebih efisien.

#### 3.4. 4. Testing

Pada proses testing, data testing hadits yang sudah dipisahkan akan dijadikan input pada sistem classifier. Data testing akan diproses terlebih dahulu persis seperti proses yang dilakukan pada data training, sehingga menghasilkan matriks yang memiliki fitur yang sama dengan data training. Bila matriks dari data training sudah didapatkan, sistem classifier akan melakukan perhitungan dengan menggunakan metode SVM untuk menentukan label dari setiap hadits yang ada pada data training. Lalu label yang diberikan oleh sistem classifier akan dihitung dan dibandingkan dengan label aslinya untuk mendapatkan tingkat akurasi dari sistem classifier. Tingkat akurasi didapatkan dengan menggunakan rumus sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data testing benar}}{\text{Jumlah data testing}} \times 100\%$$

Gambar 13. Rumus menghitung tingkat akurasi

#### 3.5. Analisis Sistem yang Akan Dibangun

Berdasarkan analisis proses klasifikasi, dibutuhkan sebuah sistem yang dapat mempermudah proses klasifikasi. Sistem ini memiliki tiga fitur utama yaitu import data hadits dari file bertipe .csv, melakukan proses training terhadap data hadits, dan melakukan proses testing terhadap hadits yang diinput oleh pengguna.

Selain itu, terdapat fitur tambahan yang dapat menunjang untuk meningkatkan kinerja sistem, yaitu sistem mampu melakukan penghapusan otomatis terhadap data hadits yang memiliki teks dan label yang sama persis, mampu menampilkan data hadits yang memiliki teks yang sama persis sedangkan labelnya berbeda.

Untuk menunjang fitur-fitur yang telah terdefinisi, maka sistem akan menggunakan database untuk menyimpan data yang diperlukan, diantaranya yaitu hadits, filtered hadits, features, stop words, kata dasar, dan training model. Data training model akan disimpan dalam bentuk file binary, sedangkan sisanya akan disimpan ke dalam database SQL.

Tabel 7. Daftar database dan fungsinya

<b>Nama Database</b>	<b>Fungsi</b>
Hadits	Untuk menyimpan data hadits yang belum diproses apapun
Filtered Hadits	Untuk menyimpan data hadits yang sudah melalui proses Preprocessing
Features	Untuk menyimpan daftar fitur yang dihasilkan dari proses feature selection
Stop Words	Untuk menyimpan daftar kata dalam bahasa Indonesia yang tidak memiliki makna
Kata Dasar	Untuk menyimpan daftar kata dasar dalam bahasa Indonesia
Traning Model	Untuk menyimpan matriks TF-IDF yang merupakan Training Model

### 3.6. Analisis Kebutuhan Sistem

Berdasarkan analisis proses klasifikasi dan analisis sistem yang akan dibangun, maka didapatkanlah beberapa kebutuhan sistem yang harus dipenuhi, diantaranya:

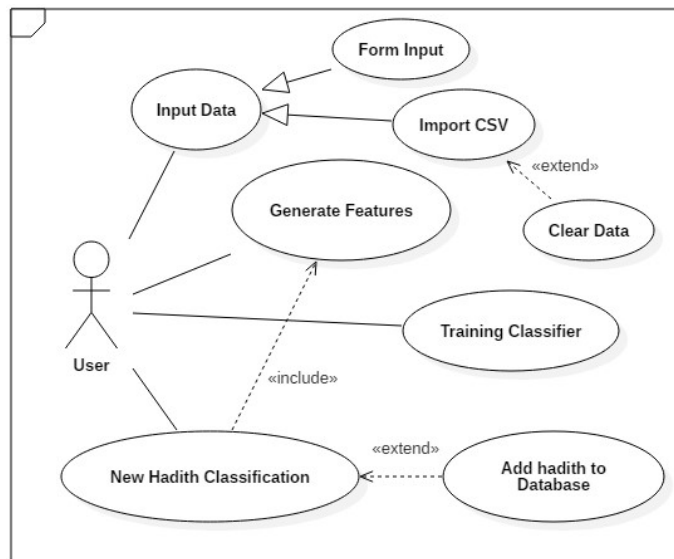
Tabel 8. Kebutuhan sistem

<b>No</b>	<b>ID</b>	<b>Requirement</b>	<b>Deskripsi</b>
1	REQ-01	Sistem dapat melakukan penambahan data hadits baik melalui form atau import dari file berformat .csv.	Sistem menyediakan fitur untuk penambahan data hadits, sehingga pengguna bisa menggunakan form penambahan data ataupun import melalui file berformat .csv yang sudah disediakan dalam sistem.
2	REQ-02	Sistem dapat memanipulasi data hadits.	Sistem menyediakan fitur untuk menampilkan, mengubah, dan

			menghapus data hadits yang pernah diinput, sehingga pengguna dapat menyesuaikan data yang ada bila terjadi kesalahan.
3	REQ-03	Sistem dapat melakukan generate features hadits dan melakukan training terhadap sistem klasifikasi.	Sistem dapat melakukan proses case folding, filtering, tokenizing, dan stemming terhadap data hadits. Setelah dilakukan stemming, maka sistem akan membuat matriks TFIDF dan menghilangkan feature yang memiliki nilai 0. Setelah itu dilakukan proses training untuk menghasilkan SVM Training Model.
4	REQ-04	Sistem dapat melakukan klasifikasi terhadap data hadits.	Sistem dapat melakukan klasifikasi terhadap data hadits, apakah hadits tersebut termasuk hadits sahih, dhaif, atau maudhu berdasarkan teks hadits.

### 3.7. Perancangan

#### 3.7.1. Use Case Diagram



Gambar 14. Use Case Diagram Sistem Klasifikasi

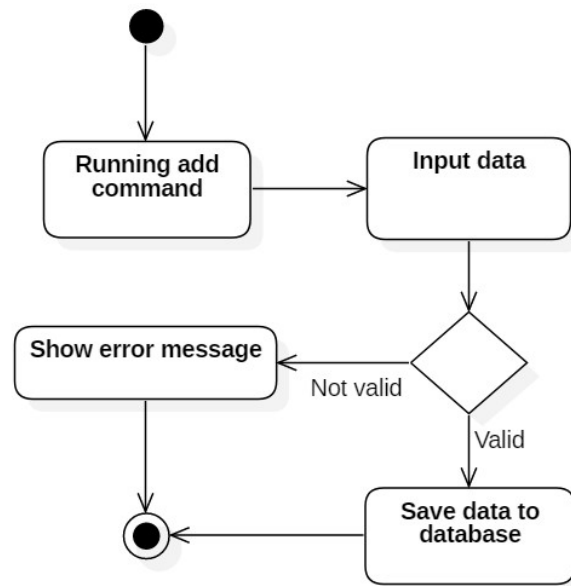
Interaksi pengguna dengan sistem digambarkan pada use case diagram diatas (Gambar 14). Pada gambar tersebut terlihat bahwa pengguna melakukan input data baik melalui form input atau import file berformat .csv, melakukan generate features, melakukan training classification, melakukan klasifikasi terhadap hadits baru, dan menambahkan hadits baru ke dalam database.

Sistem melakukan training untuk mengukur tingkat akurasi dari sistem classifier dan juga untuk menghasilkan SVM Training Model yang nantinya akan digunakan dalam melakukan klasifikasi terhadap hadits baru.

#### 3.7.2. Activity Diagram

Pada perancangan ini, terdapat lima activity diagram yang telah dibuat, yaitu activity diagram untuk input data melalui form input, input data melalui import csv, generate features, training classification, dan new hadith classification.

### 3.7.2.1. Input Data – Form Input

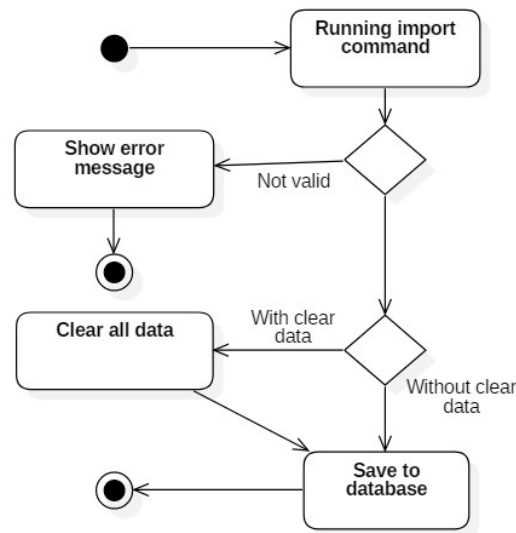


Gambar 15. Activity Diagram Input Data via Form

Input data dengan metode form input akan dilakukan melalui Command Line Interface (CLI). Ketika pengguna menjalankan perintah untuk menambahkan hadits baru, maka sistem akan meminta pengguna untuk mengisi teks hadits dan status hadits apakah hadits tersebut sah, dhaif, atau maudhu.

Setelah pengguna mengisi form yang diminta oleh sistem, maka sistem akan melakukan validasi terlebih dahulu terhadap data yang pengguna masukkan. Bila data valid, maka sistem akan menyimpan data tersebut ke dalam database, sedangkan bila tidak valid, maka sistem akan menampilkan pesan error.

### 3.7.2.2. Input Data – Import CSV



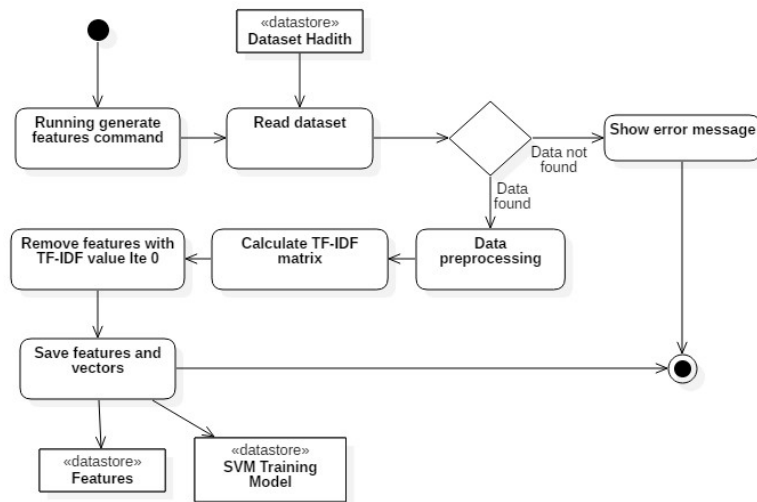
Gambar 16. Activity Diagram Input Data via Import CSV

Input data dengan metode import file .csv akan dilakukan melalui CLI. Pertama pengguna harus menjalankan perintah untuk melakukan import file disertai dengan parameter lokasi file .csv disimpan. Bila pengguna ingin menimpa data yang sudah tersimpan di database, maka pengguna bisa menyertakan parameter tambahan untuk melakukan penghapusan data sebelum file .csv disimpan ke dalam database.

Setelah perintah dijalankan, maka sistem akan melakukan validasi terhadap parameter dan data yang ada di dalam file .csv. Bila terdapat kesalahan, maka sistem akan menampilkan pesan error, dan bila tidak terdapat kesalahan, maka sistem akan melanjutkan ke pengecekan parameter clear data, bila terdapat parameter clear data, maka sistem akan menghapus semua data yang ada di database terlebih dahulu sebelum menyimpan data dari file .csv, sedangkan bila parameter clear data tidak ada, sistem akan langsung menambahkan data dari file .csv ke dalam database.



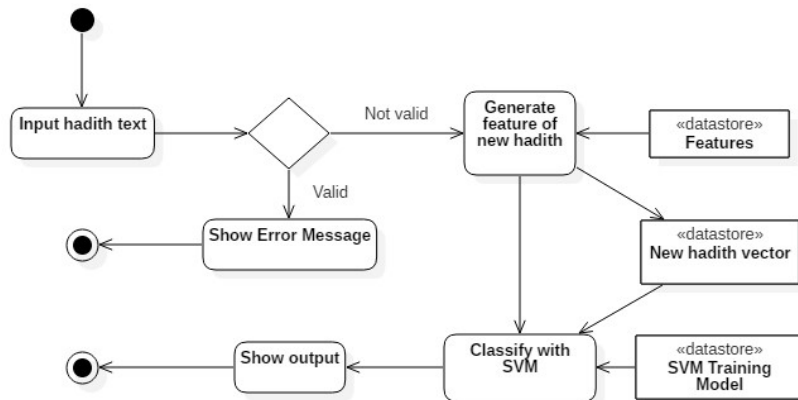
### 3.7.2.3. Generate Features dan Training Process



Gambar 17. Activity Diagram Generate Features

Generate features akan dilakukan melalui CLI. Sistem akan membaca dataset hadits ketika pengguna menjalankan perintah generate features, lalu bila tidak ada data yang ditemukan dari dataset, maka sistem akan memunculkan pesan error dan mengakhiri sistem. Namun bila terdapat data di dalam dataset maka sistem akan melanjutkannya kepada proses preprocessing, lalu diteruskan dengan menghitung matriks TF-IDF, menghilangkan kolom yang bernilai kurang dari sama dengan 0, dan yang terakhir menyimpan matriks tersebut menjadi dua jenis data, yaitu data features dan data vektor.

#### 3.7.2.4. New Hadith Classification



Gambar 18. Activity Diagram New Hadith Classification

Proses klasifikasi terhadap hadits baru dimulai dengan pengguna melakukan input text hadits ke dalam sistem. Lalu sistem akan memeriksa apakah teks yang dimasukkan pengguna valid, jika tidak maka sistem akan memunculkan pesan error, sedangkan jika valid, maka akan dilanjutkan ke proses generate features dari hadits yang dimasukkan tersebut. Feature yang digenerate akan disesuaikan dengan data features yang telah didapat pada proses training. Proses generate feature ini akan menghasilkan vektor baru dari hadits yang dimasukkan. Vector tersebut akan dibandingkan dengan vektor pada SVM Training Model pada proses klasifikasi untuk menentukan label yang paling mendekati dengan teks hadits yang dimasukkan.

#### 3.7.3. User Interface

Untuk dapat digunakan oleh pengguna, sistem harus memiliki User Interface (UI). Sistem ini akan memiliki dua jenis UI, yaitu UI yang bertipe Command Line Interface (CLI) dan UI yang bertipe Graphical User Interface (GUI).

##### 3.7.3.1. Command Line Interface

Sistem ini memiliki beberapa Command Line Interface (CLI) untuk mempermudah beberapa perintah operasi yang diperlukan agar sistem dapat berjalan dengan baik. Beberapa CLI yang akan dibuat diantaranya:

Tabel 9. Perancangan perintah untuk melakukan import data

Nomor UI	CLI-001
Nama UI	Perintah Import Data
Deskripsi UI	Perintah untuk melakukan import data .csv
Command	
Keterangan	Pertama pengguna memasukkan hadits yang hendak dicari statusnya ke dalam sebuah text area, lalu pengguna menekan tombol Classify. Setelah itu sistem akan menampilkan status dari hadits tersebut beserta dengan confidence level nya.

### 3.7.3.2. Graphical User Interface

Sistem ini hanya memiliki satu Graphical User Interface (GUI) yaitu untuk melakukan klasifikasi hadits. Berikut adalah perancangan tampilan GUI yang akan dibuat:

Tabel 10. Perancangan tampilan klasifikasi hadits

Nomor UI	GUI-001
Nama UI	Halaman Klasifikasi Hadits Baru
Deskripsi UI	Halaman bagi pengguna untuk melakukan klasifikasi terhadap suatu hadits.
Gambar UI	

<div><div>Hadith Classifier</div><div><div><div><div></div><div></div><div></div><div></div></div><div>http://hc.com</div><div></div></div></div><div><h2>Hadith Classifier</h2><div><div>Ibnu Umar berkata, "Seorang hamba tidak akan mencapai hakikat takwa yang sebenarnya kecuali ia dapat meninggalkan apa saja yang dirasa tidak enak dalam hati."</div><div><div><div></div><div>Classify</div></div></div><div><div>Hadith Status</div><div>Confidence Level</div><div>Sahih</div><div>95.6%</div></div></div></div></div>	
Keterangan	Pertama pengguna memasukkan hadith yang hendak dicari statusnya ke dalam sebuah text area, lalu pengguna menekan tombol Classify. Setelah itu sistem akan menampilkan status dari hadits tersebut beserta dengan confidence level nya.

## **BAB IV**

# **IMPLEMENTASI**

# **BAB V**

## **PENGUJIAN**

# **BAB VI**

## **PENUTUP**

## DAFTAR PUSTAKA

- [1]. Terjemah Kitab Bulughul Maram: Kumpulan Hadits Hukum Panduan Hidup Muslim Sehari-hari. Al-Hafizh Ibnu Hajar Al-Asqalani, Jakarta 2016
- [2]. Najiyah, Ina, Sari Susanti, Dwiza Riana, Mochammad Wahyudi, “Hadith Degree Classification for Shahih Hadith Identification Web Based”, STMIK Nusa Mandiri Jakarta, 2017.
- [3]. E., Wakil, “The Prophet’s Treaty with the Christians of Najran: An Analytical Study to Determine the Authenticity of the Covenants”, 2016 Journal of Islamic Studies, 2016
- [4]. C. Melchert, “Narrative Social Structure: Anatomy of the Ḥadīth Transmission Network”, Journal of Islamic Studies, 2008
- [5]. B. Prasetijo, Agung, R. Rizal Isnanto, Dania Eridani, “Hoax Detection System on Indonesian News Sites Based on Text Classification using SVM and SGD”, Proc. of 2017 4th Int. Conf. on Information Tech., Computer, and Electrical Engineering (ICITACEE), 2017
- [6]. Khilwani Ibrahim, Nuzulha, Mohamad Fauzan Noordin, Suhaila Samsuri, “Isnad Al-hadith Computational Authentication: An Analysis Hierarchically”, 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M), 2016
- [7]. Suyanto, “Artificial Intelligence: Searching, Reasoning, Planning, Learning - Revisi Kedua”, Informatika, 2014
- [8]. Primartha, Rifkie, “Belajar Machine Learning Teori dan Praktik”, Informatika, 2018
- [9]. Korde, Vandana, C Namrata Mahender, “Text Classification And Classifiers: A Survey”, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012
- [10]. C., Triawati, “Metode Pembobotan Statistical Concept Based untuk Klastering dan Kategorisasi Dokumen Berbahasa Indonesia”, Institute Teknologi Telkom Bandung, 2009



- [11]. Saxena, Shruti, “Precision vs Recall”, Towards Data Science, March 2018, <<https://towardsdatascience.com/precision-vs-recall-386cf9f89488>> [diakses 14 April 2019]
- [12]. Satriyo Nugroho, Anto, “Pengantar Support Vector Machine”, Anto Satriyo Nugroho, 2007
- [13]. J., Asian, “Effective Techniques for Indonesian Text Retrieval”, PhD thesis School of Computer Science and Information Technology RMIT University Australia, 2007
- [14]. Librian, Andy, “Stemming Bahasa Indonesia”, Sastrawi Wiki, 2016, <https://github.com/sastrawi/sastrawi/wiki/Stemming-Bahasa-Indonesia>> [diakses 18 April 2019]
- [15]. Jing, Li-Ping, Hou-Kuan Huang, Hong-Bo Shi, “Improved Feature Selection Approach TFIDF in Text Minig”, Proceedings of the First International Conference on Machine Learning and Cybernetics, 2002