STUDI KOMPARASI *ENSEMBLE METHOD: BOOTSTRAP AGGREGATING (BAGGING)* DAN *ADAPTIVE BOOSTING (ADABOOST)* PADA KLASIFIKASI TERJEMAHAN AYAT AL-QUR'AN BAHASA INDONESIA

Skripsi
untuk memenuhi sebagai persyaratan
mencapai derajat Sarjana S-1
Program Studi Teknik Informatika

Disusun oleh:
Sekar Minati
16650023

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA
2020

HALAMAN PENGESAHAN

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-689/Un.02/DST/PP.00.9/02/2020

Tugas Akhir dengan judul : STUDI KOMPARASI ENSEMBLE METHOD BOOTSTRAP AGGREGATING (BAGGING) DAN ADAPTIVE BOOSTING (ADABOOST) PADA KLASIFIKASI TERJEMAHAN AYAT AL - QUR 'AN BAHASA INDONESIA

yang dipersiapkan dan disusun oleh:

Nama : SEKAR MINATI
Nomor Induk Mahasiswa : 16650023
Telah diujikan pada : Kamis, 13 Februari 2020
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR

Ketua Sidang/Penguji I

Muhammad Didik Rohmad Wahyudi, S.T., MT.
NIP. 19760812 200901 1 015

Penguji II

Dr. Agung Fatwanto, S.Si., M.Kom.
NIP. 19770103 200501 1 003

Penguji III

Rahmat Hidayat, S.Kom., M.Cs.
NIP. 19850514 201503 1 002

Yogyakarta, 13 Februari 2020
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan

Dr. Murtono, M.Si.
NIP. 19691212 200003 1 001

1/1 24/02/2020

SURAT PERSETUJUAN SKRIPSI

Universitas Islam Negeri Sunan Kalijaga FM-UINSK-BM-05-03/R0

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi
Lamp :

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Sekar Minati
NIM : 16650023
Judul Skripsi : "Studi Komparasi *Ensemble Method: Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* pada Klasifikasi Terjemahan Ayat Al-Qur'an Bahasa Indonesia"

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 4 Februari 2020
Pembimbing

M. Didik Rohmad Wahyudi, S.T., MT.
NIP. 19760812 200901 1 015

PERNYATAAN KEASLIAN SKRIPSI

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Sekar Minati

NIM : 16650023

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi saya yang berjudul **"Studi Komparasi *Ensemble Method: Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* pada Klasifikasi Terjemahan Ayat Al-Qur'an Bahasa Indonesia"** merupakan hasil penelitian saya sendiri, tidak terdapat pada karya yang pernah di ajukan untuk memperoleh gelar kesarjana di suatu perguruan tinggi, dan bukan plagiasi karya orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 5 Februari 2020

Yang menyatakan

Sekar Minati

NIM.16650023

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul **Studi Komparasi *Ensemble Method: Bootstrap Aggregating* (*Bagging*) dan *Adaptive Boosting* (*AdaBoost*) pada Klasifikasi Terjemahan Ayat Al-Qur'an Bahasa Indonesia** sebagai salah satu syarat untuk mencapai gelar sarjana program studi Teknik Informatika Universitas Islam Negeri Sunan Kalijaga Yogyakarta. Sholawat serta salam selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW. beserta seluruh keluarga dan sahabat beliau.

Penulis menyadari bahwa apa yang dilakukan dalam penyusunan laporan penelitian ini masih terlalu jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran yang berguna dalam penyempurnaan analisis ini di masa yang akan datang. Semoga apa yang telah penulis lakukan dapat bermanfaat bagi pembaca.

Tidak lupa penulis juga mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyelesaian skripsi ini, baik secara langsung maupun tidak langsung. Ucapan terimakasih penulis sampaikan kepada:

1. Bapak Prof. Drs. K.H. Yudian Wahyudi, M.A., Ph.D., selaku Rektor UIN Sunan Kalijaga Yogyakarta.

2. Bapak Dr. H. Waryono, M.Ag., selaku Wakil Rektor Bidang Kemahasiswaan dan Kerjasama UIN Sunan Kalijaga Yogyakarta.
3. Bapak Dr. Murtono, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Sumarsono, S.T., M.Kom., selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
5. Bapak Muhammad Didik Rohmad Wahyudi, S.T., MT., selaku dosen pembimbing skripsi yang telah sabar dan meluangkan waktunya untuk memberikan motivasi, koreksi dan kritik saran kepada penulis sehingga skripsi ini dapat diselesaikan.
6. Bapak Dr. Agung Fatwanto, S.Si., M.Kom., selaku Dosen Pembimbing Akademik.
7. Ibu Ade Ratnasari, S.Kom. M.T., Bapak Agus Mulyanto, S.Si., M.Kom., Bapak Aulia Faqih Rifa'i, M.Kom., Ibu Maria Ulfah Siregar, S.Kom. MIT., Ph.D., Bapak M. Mustakim, S.T, Bapak M. Taufiq Nuruzzaman, S.T., Bapak Nurochman, S.Kom., M.Kom., Bapak Rahmat Hidayat, S.Kom., M.Cs., Ibu Dr. Shofwatul 'Uyun, S.T., M.Kom., selaku dosen pengampu mata kuliah program studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta yang

telah banyak membantu sehingga penulis dapat menyusun tugas akhir.

8. Seluruh staf dan karyawan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
9. Ucapan terima kasih yang terdalam untuk kedua orang tuaku tercinta, Ibu Loeky Zuraidah dan Bapak Bambang Mulyantono yang selalu memberikan doa, perhatian, kasih sayang dan segala dukungan yang tak terhingga.
10. Seluruh Keluarga Besar Eyang Saekan Adi Partono dan Mbah Achmadun yang senantiasa mendoakan dan memberikan dukungan kepada penulis.
11. Sahabat-sahabatku Lina, Ulfa, Nadia, Yayang, Mbak Nafi, Areta, Dimas, Nabilah, Echa, Shara, Ardiyan, Husni, Aat, Hendra, Ari, Abdy, Alvri, Dio, yang telah senantiasa menemani penulis serta memberikan semangat, dorongan, dan inspirasi dalam proses pengerjaan skripsi ini.
12. Seluruh teman-teman Teknik Informatika 2016 yang tidak dapat penulis sebutkan satu per satu.
13. Teman-teman Himpunan Mahasiswa Program Studi Teknik Informatika dan seluruh Keluarga Besar Teknik Informatika UIN Sunan Kalijaga.
14. Teman-teman KKN Kelompok 102 Angkatan 99.

15. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu dan telah memberikan banyak doa dan dukungan sehingga penelitian ini dapat terselesaikan.

Yogyakarta, 3 Februari 2020

Penulis

HALAMAN PERSEMBAHAN

Skripsi ini aku persembahkan
untuk yang tersayang, Ibu dan Ayah,
yang selalu menemani, mendukung, dan menyayangiku
sejak saat pertama kali ku jejakkan kaki ini di dunia,
hingga saat dimana aku siap untuk menyongsong indahnya dunia.
Terima kasih. Aku sayang kalian.

HALAMAN MOTTO

Carpe Diem. Seize the Day.

"Live as if you were to die tomorrow. Learn as if you were to live forever."

— Mahatma Gandhi

"Aspire to inspire before we expire."

— Eugene Bell Jr.

"Be strong but not rude.

Be kind but not weak.

Be humble but not timid.

Be proud but not arrogant."

— Zig Ziglar

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN ii
SURAT PERSETUJUAN SKRIPSI iii
PERNYATAAN KEASLIAN SKRIPSI iv
KATA PENGANTAR v
HALAMAN PERSEMBAHAN ix
HALAMAN MOTTO x
DAFTAR ISI xi
DAFTAR GAMBAR xiv
DAFTAR TABEL xvi
INTISARI xviii
ABSTRACT xx
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 5
1.3. Batasan Masalah 5
1.4. Tujuan Penelitian 6
1.5. Manfaat Penelitian 6
1.6. Keaslian Penelitian 7
1.7. Sistematika Penulisan 7
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI 10
2.1. Tinjauan Pustaka 10

2.2. Landasan Teori 20
2.2.1. Kecerdasan Buatan (*Artificial Intelligence*) ... 20
2.2.2. Pembelajaran Mesin (*Machine Learning*) 21
2.2.3. *Text Mining* 22
2.2.4. *Preprocessing Text* 23
2.2.5. *Term Weighting* 25
2.2.6. *Term Frequency-Inverse Document Frequency (TF-IDF)* 29
2.2.7. Klasifikasi Teks 30
2.2.8. *Adaptive Boosting* (*Adaboost*) *Classifier* 32
2.2.9. *Bootstrap Aggregating* (*Bagging*) *Classifier* .. 34
2.2.10. Evaluasi Model 35
2.2.11. Al-Qur'an 41
2.2.12. *Python* 44
BAB III METODE PENELITIAN 47
3.1. Metode Penelitian 47
3.2. Tahapan Penelitian 47
3.2.1. Studi Pendahuluan 48
3.2.2. Pengumpulan Data 48
3.2.3. *Data Preprocessing* 49
3.2.4. *Term Frequency-Inverse Document Frequency* (*TF-IDF*) 50
3.2.5. Klasifikasi 50
3.2.6. Analisis Hasil 51
3.2.7. Visualisasi Hasil 51

3.3. Kebutuhan Sistem 51
BAB IV HASIL DAN PEMBAHASAN 53
4.1. Pengumpulan Data 54
4.2. Pengolahan Data 55
4.2.1. *Case Folding* 56
4.2.2. *Stopword Filtering* 58
4.3. Analisis 59
4.3.1. *Term Frequency-Inverse Document Frequency* (*TF-IDF*) 59
4.3.2. Klasifikasi 61
4.3.2.1. *Bootstrap Aggregating (Bagging)* 62
4.3.2.2. *Adaptive Boosting (AdaBoost)* 63
4.4. Hasil 68
4.4.1. Pengujian Performa Algoritma 68
4.4.2. Pengujian Data Artikel Islam 72
4.4.3. Persilangan Bab Tematik 75
4.4.4. Klasifikasi Data yang Tidak Terlabeli 80
BAB V PENUTUP 84
5.1. Kesimpulan 84
5.2. Saran 85
DAFTAR PUSTAKA 87
LAMPIRAN 96
CURRICULUM VITAE 115

DAFTAR GAMBAR

Gambar 2.1 Algoritma AdaBoost (Zhou, 2009) 33
Gambar 2.2 Algoritma Boosting (Zhou, 2009)..... 35
Gambar 3.1 Skema Alur Penelitian 47
Gambar 4.1 Flowchart Proses Analisis 53
Gambar 4.2 Proses Klasifikasi Bagging 63
Gambar 4.3 Diagram Scatter Training Points..... 65
Gambar 4.4 Diagram Bagging Ensemble Size pada Al-Qur'an Tematik Al-Hadi 70
Gambar 4.5 Diagram AdaBoost Ensemble Size pada Al-Qur'an Tematik Al-Hadi 71
Gambar 4.6 Diagram Bagging Ensemble Size pada Al-Qur'an Tematik Kemenag 71
Gambar 4.7 Diagram AdaBoost Ensemble Size pada Al-Qur'an Tematik Kemenag 72
Gambar 4.8 Sebaran Pembagian Bab Tematik Kemenag pada Al-Qur'an Tematik Al-Hadi 76
Gambar 4.9 Sebaran Pembagian Bab Tematik Kemenag pada Al-Qur'an Tematik Al-Hadi (Lanjutan)..... 77
Gambar 4.10 Sebaran Pembagian Bab Tematik Kemenag pada Al-Qur'an Tematik Al-Hadi (Lanjutan)..... 78
Gambar 4.11 Sebaran Pembagian Bab Tematik Al-Hadi pada Al-Qur'an Tematik Kemenag 79

Gambar 4.12 Sebaran Pembagian Bab Tematik Al-Hadi pada Al-Qur'an Tematik Kemenag (Lanjutan) 80
Gambar 4.13 Hasil Klasifikasi Ayat yang Belum Terindeks pada Kedua Bab Tematik oleh *Bagging Tree*..... 81
Gambar 4.14 Hasil Klasifikasi Ayat yang Belum Terindeks pada Kedua Bab Tematik oleh *AdaBoost*..... 82

DAFTAR TABEL

Tabel 2.1 Perbandingan Tinjauan Pustaka 14
Tabel 2.2 Perbandingan Tinjauan Pustaka (Lanjutan) 15
Tabel 2.3 Perbandingan Tinjauan Pustaka (Lanjutan) 16
Tabel 2.4 Perbandingan Tinjauan Pustaka (Lanjutan) 17
Tabel 2.5 Perbandingan Tinjauan Pustaka (Lanjutan) 18
Tabel 2.6 Perbandingan Tinjauan Pustaka (Lanjutan) 19
Tabel 4.1 Pembagian Tema Al-Qur'an Tematik 55
Tabel 4.2 Contoh Data Terjemahan Ayat Al-Qur'an 56
Tabel 4.3 Hasil Case Folding Data Terjemahan Ayat Al-Qur'an 57
Tabel 4.4 Hasil Case Folding Data Terjemahan Ayat Al-Qur'an 58
Tabel 4.5 Contoh Dokumen untuk Pembobotan TF-IDF 60
Tabel 4.6 Contoh Proses Pembobotan TF-IDF 60
Tabel 4.7 Contoh Proses Pembobotan TF-IDF (Lanjutan) 61
Tabel 4.8 Contoh Data untuk Proses Klasifikasi 65
Tabel 4.9 Bobot Training Points 66
Tabel 4.10 Nilai Error Kombinasi Decision Tree 67
Tabel 4.11 Performa Algoritma Klasifikasi pada Al-Qur'an Tematik Al-Hadi 69
Tabel 4.12 Performa Algoritma Klasifikasi pada Al-Qur'an Tematik Kemenag 69
Tabel 4.13 Contoh Data Artikel Islami 73

Tabel 4.14 Hasil Summarization..... 74
Tabel 4.15 Hasil Preprocessing 74
Tabel 4.16 Hasil Klasifikasi..... 75

Studi Komparasi *Ensemble Method: Bootstrap Aggregating* (*Bagging*) dan *Adaptive Boosting* (*AdaBoost*) pada Klasifikasi Terjemahan Ayat Al-Qur'an Bahasa Indonesia

Sekar Minati

16650023

INTISARI

Al-Qur'an adalah salah satu pedoman hidup bagi Muslim yang merupakan sumber pengetahuan, kebijakan, maupun hukum. Dalam rangka memudahkan masyarakat untuk mempelajari dan memahami Al-Qur'an dibuatlah metode penafsiran salah satunya yaitu tematik, dan untuk mempermudah dalam melakukan pencarian ayat sesuai tematik diciptakanlah suatu indeks Al-Qur'an.

Penelitian ini bertujuan untuk meningkatkan hasil pembelajaran klasifikasi Al-Qur'an tematik Kementerian Agama (Kemenag) dan Al-Qur'an Al-Hadi dengan menggunakan metode pembelajaran *ensemble* yaitu *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)*. Kedua metode tersebut diuji dengan melakukan prediksi dalam pencarian dalil yang terkait pada artikel-artikel islami.

Metode *Bagging* dan *AdaBoost* dapat meningkatkan hasil pembelajaran klasifikasi dengan *base estimator* algoritma klasifikasi *Decision Tree* dengan rata-rata hasil pengujian kedua indeks tersebut diperoleh nilai *precision* sebesar 0.49, *recall* sebesar 0.48, *f1-score* sebesar 0.49, *accuracy* 54%, dan evaluasi *cross validation* sebesar 31% pada *Bagging* dan *precision* sebesar 0.25, *recall* sebesar 0.18, *f1-score* sebesar 0.16, *accuracy* 34%, dan evaluasi *cross validation* sebesar 32% pada *AdaBoost*.

Kata kunci: *text mining*, klasifikasi, *ensemble*, *bagging*, *adaboost*, al-qur'an, tematik, artikel islami

Comparative Study of the Ensemble Method: Bootstrap Aggregating (Bagging) and Adaptive Boosting (AdaBoost) for Classification of Indonesian Al-Qur'an Translation

Sekar Minati

16650023

ABSTRACT

Al-Qur'an is a way of life for Muslims which is a source of knowledge, policy, and law. In order to make it easier for the public to learn and understand the Al-Qur'an, one method of interpretation is namely thematic, and the Al-Qur'an index was created to make it easier to search for verses according to its theme.

This research to improve the learning outcomes of Qur'an's Ministry of Religion Republic Indonesia and Al-Qur'an Al-Hadi thematic Qur'an classification using ensemble learning methods: Bootstrap Aggregating (Bagging) and Adaptive Boosting (AdaBoost). Both of these methods tested by doing prediction to searching the proof related to Islamic articles.

Bagging and AdaBoost methods can improve classification learning outcomes with the base estimator algorithm Decision Tree classification with the average test

results of the two indices obtained the value of precision 0.49, recall 0.48, f1-score 0.49, accuracy 54%, and cross-validation evaluation of 31% in Bagging method and precision 0.25, recall 0.18, f1-score of 0.16, accuracy 34%, and evaluation of cross-validation of 32% on AdaBoost.

Keywords: text mining, classification, ensemble, bagging, adaboost, al-quran, thematic, Islamic articles

BAB I
PENDAHULUAN

1.1. Latar Belakang

Association Religion Data Archives (ARDA) mendata bahwa terdapat 22,8% penganut Islam di dunia ini dari total populasi di bumi (Johnson & Grim, 2015). Data tersebut menjadikan Islam sebagai agama terbesar kedua di dunia setelah agama Kristen. Populasi penduduk di Indonesia pada tahun 2010 sekitar 237.641.326 orang dan 87,2% dari populasi tersebut adalah beragama Islam (Statistics Indonesia, 2010). Ini membuat populasi Muslim terbesar di suatu negara adalah Indonesia, negara yang menjadi rumah bagi 12,7% Muslim di dunia, diikuti oleh Pakistan (11,0%), dan India (10,9%) (Pew Research Center, 2011).

Al-Qur'an adalah salah satu pedoman hidup bagi Muslim yang di dalamnya terdiri dari 114 surat, 6.236 ayat, dan 77.477 kata yang disusun menggunakan Bahasa Arab (Osman, Hilal, & Alhawarat, 2016). Al-Qur'an diturunkan kepada Nabi Muhammad SAW. ketika beliau berumur 40 tahun selama 23 tahun. Setelah Nabi Muhammad SAW. wafat, Al-Qur'an, As-Sunnah, dan buku-buku Islam terdahulu menjadi sumber bagi para Muslim dalam mendapatkan sumber pengetahuan, kebijakan, maupun hukum. Maka dari itu, Al-

Qur'an sangat penting bagi kehidupan sehari-hari manusia khususnya umat Muslim.

Di samping pentingnya Al-Qur'an bagi kehidupan sehari-hari, masih banyak umat Muslim yang tidak mengerti makna dari tiap-tiap ayat yang ada pada Al-Qur'an. Maka dari itu para *mufassir* menciptakan metode penafsiran dalam rangka mengkontekstualisasikan pemaknaan ayat-ayat Al-Qur'an. Metode-metode penafsiran yang digunakan adalah metode *tahlili* (analisis), *ijmali* (global), *muqaran* (perbandingan), dan *maudui* (tematik). Metode penafsiran *maudui* (tematik) merupakan metode yang terbaik dibandingkan dengan keempat metode lainnya, dikarenakan metode ini disajikan dalam bentuk tema-tema pokok yang berhubungan langsung dengan permasalahan kehidupan yang dihadapi oleh masyarakat sehingga penyajiannya terkesan simpel dan juga menjadi kelebihan serta keunikan dari metode ini (A. S. Hidayatullah, 2009).

Dalam rangka memudahkan masyarakat dalam mengakses ayat-ayat Al-Qur'an yang telah dilakukan proses penafsiran secara tematik ini, dibuatlah indeks Al-Qur'an. Indeks Al-Qur'an berusaha memberikan informasi sebaik mungkin dalam menyajikan keberadaan ayat-ayat yang dibutuhkan oleh masyarakat dalam rangka membumikan Al-Qur'an di tengah-tengah masyarakat (A. S. Hidayatullah, 2009). Bereferensi dengan indeks Al-Qur'an pulalah para

ulama, mubaligh, pemikir, penafsir, mahasiswa, dan semua orang mempelajari dan memahami Al-Qur'an hingga dapat menciptakan karya-karya tulis yang islami guna untuk berdakwah kepada umat Muslim lainnya.

Di Indonesia sendiri terdapat berbagai ragam dan bentuk penyajian dari indeks Al-Qur'an. Berawal dari buku-buku hingga berkembang menjadi bentuk digital seiring dengan perkembangan zaman. Indeks Al-Qur'an tematik yang tersedia diantaranya indeks Al-Qur'an yang disusun oleh Kementerian Agama (Kemenag) dan Al-Qur'an Al-Hadi yang disusun oleh Pusat Kajian Hadis. Kedua indeks tersebut memiliki kriteria tersendiri dalam melakukan proses penafsiran tematik, sehingga keduanya memungkinkan untuk memiliki tema dan pengelompokkan ayat-ayat yang berbeda.

Mengingat pentingnya indeks sebagai salah satu panduan dalam proses pembelajaran hingga berdakwah, maka pencarian rujukan tiap-tiap ayat terhadap indeks Al-Qur'an ini sering dilakukan. Akan tetapi dalam proses ini masih berjalan secara manual. Mengingat banyaknya jumlah ayat dan indeks yang ada pada Al-Qur'an, maka proses pencarian ini akan memakan waktu yang cukup lama. Masalah ini menjadi tantangan bagi Informatika untuk mengeksplorasi pengetahuan melalui sistem komputasi kecerdasan buatan untuk membantu proses pencarian indeks Al-Qur'an sehingga dapat menjawab berbagai pertanyaan berdasarkan

pengetahuan dari Al-Qur'an. Hal ini menjadi latar belakang bagi penulis untuk menciptakan lingkungan komputasi pada bidang *text mining*, yaitu dengan cara melakukan klasifikasi pada tiap ayat-ayat Al-Qur'an guna mempermudah umat Muslim dalam mempelajari dan memahami Al-Qur'an.

Klasifikasi merupakan salah satu pemrosesan dalam *text mining* yang bertujuan untuk membagi sesuatu menurut kelas-kelas. *Text mining* merupakan pengembangan dari *data mining*, *data mining* biasanya digunakan untuk mengolah data yang sifatnya terstruktur seperti data dari *database*, sedangkan *text mining* digunakan untuk mengolah data yang sifatnya tidak terstruktur maupun semi-terstruktur seperti *email*, dokumen teks, dan lainnya (Gupta & Lehal, 2009). Algoritma klasifikasi yang sering digunakan pada *text mining* diantaranya *Support Vector Machine (SVM)*, *Naïve Bayes*, *k-Nearest Neighbor (KNN)*, *Decision Tree*, dan *Artificial Neural Networks (ANN)*.

Pada penelitian kali ini penulis ingin melakukan optimasi terhadap pengklasifikasian yang dilakukan menggunakan algoritma *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* yang keduanya merupakan bagian dari metode *ensemble learning*. Metode *ensemble learning* menggunakan beberapa algoritma pembelajaran untuk memperoleh hasil prediksi yang lebih baik daripada yang bisa diperoleh dari salah satu pembelajaran algoritma

konstituen saja (Polikar, 2006). Dengan hasil klasifikasi dari percobaan ini diharapkan dapat membantu umat Muslim dalam mempelajari Al-Qur'an dan mengukur seberapa optimal metode *ensemble learning* dalam meningkatkan hasil akurasi klasifikasi data yang berupa teks.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dituturkan oleh penulis, maka rumusan masalah dalam penelitian ini adalah seberapa optimal algoritma *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* dalam meningkatkan hasil akurasi klasifikasi pada data yang berupa teks yang dalam hal ini berupa indeks Al-Qur'an Kementerian Agama (Kemenag) dan Al-Qur'an Al-Hadi, serta.melakukan prediksi dalam pencarian dalil yang terkait pada artikel-artikel islami.

1.3. Batasan Masalah

Batasan masalah dalam menjawab rumusan masalah yang sebelumnya sudah dijelaskan diantaranya adalah:

1. Penerapan *text mining* menggunakan algoritma *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* dalam melakukan klasifikasi terhadap Al-Qur'an.
2. Data yang akan digunakan hanya terjemahan Al-Qur'an berbahasa Indonesia yang diterjemahkan oleh

Kementerian Agama Republik Indonesia dengan kelas yang disusun berdasarkan indeks Al-Qur'an Kementerian Agama (Kemenag) dan Al-Qur'an Al-Hadi.

3. Bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa pemrograman *Python* dengan berbagai macam *library* di dalamnya.

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk melakukan perbandingan antara algoritma *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* untuk melakukan klasifikasi pada terjemahan ayat Al-Qur'an bahasa Indonesia.

1.5. Manfaat Penelitian

Berdasarkan latar belakang dan tujuan di atas, maka penelitian ini dapat memberikan manfaat sebagai berikut:

1. Penelitian ini dapat menambah wawasan dan pengembangan bagi keilmuan di bidang *text mining*.
2. Mempermudah masyarakat, khususnya umat Muslim dalam mempelajari tematik Al-Qur'an.

1.6. Keaslian Penelitian

Penelitian mengenai klasifikasi Al-Qur'an sudah banyak dilakukan baik terjemahan Al-Qur'an dalam Bahasa Inggris maupun dalam Bahasa Indonesia. Namun, berdasarkan dari referensi dan tinjauan pustaka, metode klasifikasi yang digunakan dalam penelitian, metode *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)*, yang diajukan sebagai Tugas Akhir S1 pada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta ini belum pernah digunakan sebelumnya terutama dalam bidang klasifikasi terjemahan Al-Qur'an dalam Bahasa Indonesia.

1.7. Sistematika Penulisan

Sebagai gambaran dan kerangka yang jelas mengenai pokok bahasan setiap bab dalam penelitian ini, maka diperlukan sistematika penulisan. Penyusunan laporan tugas akhir ini memiliki sistematika penulisan yang diawali dari BAB I dan diakhiri BAB V. Berikut adalah penjelasan pada tiap-tiap bab dalam laporan penelitian ini:

BAB I PENDAHULUAN

Bab pendahuluan berisikan penjelasan mengenai latar belakang dilakukannya penelitian, rumusan masalah penelitian, batasan masalah, tujuan penelitian, manfaat

penelitian, keaslian penelitian, dan sistematika penulisan penelitian.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab tinjauan pustaka dan landasan teori berisikan mengenai penelitian terdahulu dan teori-teori dasar yang terkait dengan penelitian ini. Teori yang digunakan terdiri dari *text mining*, klasifikasi teks, metode *ensemble learning* (*adaboost* dan *bagging*), *text preprocessing*, *TF-IDF*, dan *Python*.

BAB III METODE PENELITIAN

Bab metode penelitian berisi tentang penjelasan mengenai metode ataupun algoritma yang digunakan serta tahapan-tahapan yang dilakukan untuk mencapai tujuan dan kesimpulan tugas akhir.

BAB IV HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan membahas analisis data dan hasil dari penelitian yang telah dilakukan.

BAB V PENUTUP

Bab penutup berisi tentang kesimpulan dari hasil penelitian yang telah dilakukan. Selanjutnya, kekurangan yang ada pada

penelitian dituliskan pada saran untuk pengembangan penelitian di masa yang akan datang.

BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Prasetio dan Pratiwi (2015) melakukan penelitian dengan mengkombinasi teknik *bagging* dan beberapa algoritma klasifikasi seperti *naïve bayes*, *decision tree* dan *k-nearest neighbor* untuk meningkatkan akurasi dari klasifikasi dataset medis. Penelitian ini menggunakan lima dataset medis yaitu, *breast-cancer*, *liver-disorder*, *heart-disease*, *pima-diabetes* dan *vertebral column*. Setelah diterapkan teknik *bagging* pada lima dataset medis, *naïve bayes* memiliki akurasi paling tinggi untuk dataset *breast-cancer* sebesar 96,14% dengan AUC sebesar 0,984, *heart-disease* sebesar 84,44% dengan AUC sebesar 0,911 dan *pima-diabetes* sebesar 74,73% dengan AUC sebesar 0,806. Sedangkan *k-nearest neighbor* memiliki akurasi yang paling baik untuk dataset *liver-disorder* sebesar 62,03% dengan AUC sebesar dan 0,632 dan *vertebral column* dengan akurasi sebesar 82,26% dengan AUC sebesar 0,867.

Daniati (2014) melakukan penelitian tentang analisis sentimen terhadap kata atau frasa yang digunakan sebagai data uji untuk menghasilkan beberapa kelas seperti sentimen positif, negatif, dan netral. Metode yang digunakan untuk pembobotan menggunakan kata TF-IDF, kemudian pelabelan

kalimat menggunakan *sign numbers multiplication*. Selanjutnya, *training* dan *testing* menggunakan metode *Naïve Bayes* dengan kombinasi *AdaBoost*. Hasilnya, penggunaan *AdaBoost* dapat meningkatkan akurasi *Naïve Bayes*.

Adeleke *et al.* (2017) melakukan otomatisasi pelabelan ayat Al-Qur'an dengan menerapkan tiga algoritma klasifikasi teks yaitu, *k-Nearest Neighbor*, *Support Vector Machine*, dan *Naïve Bayes*. Dataset penelitian ini yaitu terjemahan bahasa Inggris dari ayat-ayat Al-Qur'an. Terjemahan tersebut kemudian diklasifikasikan sebagai "Shahadah" (pilar pertama Islam) atau "Pray" (pilar kedua Islam). Didapatkan hasil bahwa algoritma klasifikasi teks mampu mencapai akurasi lebih dari 70% dalam memberi label pada ayat-ayat Al-Qur'an.

Selanjutnya, Rahman *et al.* (2018) juga melakukan penelitian klasifikasi pada surat Al-Baqarah dengan menggunakan algoritma klasifikasi *Support Vector Machine*, *Naive Bayes*, *J48 Decision Tree*, dan *K-Nearest Neighbours*. Label yang disiapkan untuk mengklasifikasikan ayat yang dipilih adalah Iman, Ibadah dan Akhlak. Dataset yang terdiri dari 286 kalimat dari surat Al-Baqarah, didenormalisasi dengan menggunakan *StringToWordVector* di WEKA dengan metode TF-IDF. Hasil akurasi yang diperoleh algoritma SVM dan algoritma *classifier J48* memiliki kinerja akurasi terbaik secara keseluruhan 85% sedangkan *Naïve Bayes* memiliki

hasil akurasi terakhir 71%. Ini menunjukkan bahwa tidak ada algoritma klasifikasi terbaik yang spesifik.

Al-Kabi *et al.* (2015) mengklasifikasi Hadits ke dalam lima kategori yaitu: Wudhu, Puasa, Zakat, dan Adzan menggunakan algoritma *Naive Bayes*, *Bagging*, *SVM*, dan *LogiBoost*. Dataset terdiri dari 793 Hadits Sahih Bukhari. Setiap hadits memiliki dua bagian (sanad dan matan) serta tiga bagian (kutipan, tindakan, dan laporan). Dataset dikurangi menjadi 474 hadis dengan memfilter bagian "kutipan" dari ucapan secara manual dari lima kelas yang telah ditentukan. Hasil *accuracy, recall, precision,* dan *F-measure* algoritma NB memiliki skor tertinggi dengan tingkat akurasi (56,6038%) dan tingkat kesalahan (43,3962%).

Penelitian mengenai pengukuran *similarity* tema pada juz 30 Al-Qur'an telah dilakukan oleh Supriyati dan Iqbal (2018). Pada penelitian ini dibahas kedekatan tema ayat-ayat pada juz 30 yang berkaitan dengan kiamat, hisab, surga dan neraka. Dataset yang telah dilakukan *preprocessing* selanjutnya dilakukan klasifikasi menggunakan beberapa algoritma seperti *Decision Tree*, *Support Vector Machine* dan *Naïve Bayes*. Hasil terbaik klasifikasi tersebut diperoleh algoritma *decision tree* dengan akurasi 74,34%, *SVM* dengan akurasi 75,84%, dan *Naïve Bayes* dengan akurasi 66,29%.

Selain data dan teks, klasifikasi juga dilakukan terhadap gambar. Periyasamy, Thamilselvan dan Sathiaseelan

(2015) melakukan klasifikasi terhadap gambar dengan berbagai algoritma klasifikasi diantaranya *Artificial Neural Network, K-Means*, dan *Adaboost*. Pada penelitian ini algoritma *Artificial Neural Network* memberikan hasil akurasi yang lebih tinggi dalam melakukan *image processing* dibandingkan dengan dua algoritma lainnya yaitu *K-Means* dan *Adaboost* dengan akurasi klasifikasi 98%, sensitivitas 0,95%, dan spesifisitas 0,98%.

Pada tahun sebelumnya, Rahim (2019) melakukan penelitian tentang pencarian dalil pada artikel islami dengan menerapkan *Cosine Similarity* dan *Naïve Bayes* berdasarkan indeks Al-Qur'an. Penelitian ini dilakukan dengan tahap *preprocessing* untuk artikel islami, pembobotan dengan TF-IDF, dan perhitungan dengan *Cosine Similarity*. Hasil uji coba yang dilakukan dengan menggunakan *confusion matrix* didapatkan: *accuracy* 38%, *precision* 9%, dan *recall* 8%.

Sebelumnya penulis juga telah melakukan studi komparasi terhadap algoritma pengklasifikasian terhadap terjemahan Al-Qur'an bahasa Indonesia dan bahasa Inggris (Hidayat & Minati, 2019). Data terjemahan dilakukan term weighting menggunakan TF-IDF lalu selanjutnya di implementasikan algoritma klasifikasi. Algoritma klasifikasi yang dilakukan diantaranya *Support Vector Machine, Naive Bayes, k-Nearest Neighbor,* dan *J48 Decision Tree*. Hasil klasifikasi ditentukan dengan mengukur akurasi, dan *f-*

measure, itu diuji dengan persentase split 66% sebagai data latih dan sisanya sebagai data uji. Hasil klasifikasi algoritma *classifier* SVM memiliki keseluruhan kinerja akurasi terbaik untuk terjemahan bahasa Indonesia dari 81,443% dan *classifier Naïve Bayes* memiliki akurasi terbaik untuk terjemahan bahasa Inggris, yang mencapai 78,35%.

Tabel 2.1 Perbandingan Tinjauan Pustaka

No	Judul	Peneliti	Metode	Hasil
1	*A Comparative Study of ANN, K-Means and Adaboost Algorithms for image classification*	Periyasamy. N, Thamilselvan. P, dan Dr. J. G. R. Sathiaseelan	*Artificial Neural Network, K-Means,* dan *Adaboost*	Pada penelitian ini algoritma *Artificial Neural Network* memberikan hasil akurasi yang lebih tinggi dalam melakukan *image processing* dibandingkan dengan dua algoritma lainnya yaitu *K-Means* dan *Adaboost*.

Tabel 2.2 Perbandingan Tinjauan Pustaka (Lanjutan)

No	Judul	Peneliti	Metode	Hasil
2	*Extended Topical Classification of Hadith Arabic Text*	Mohammed N. Al-Kabi, Heider A. Wahsheh, Izzat M. Alsmadi, Abdallah Moh'd Ali Al-Akhras	*Naive Bayes*, *Bagging*, *SVM*, dan *LogiBoost*	Hasil *accuracy*, *recall*, *precision*, dan *F-measure* algoritma NB memiliki skor tertinggi dengan tingkat akurasi (56,6038%) dan tingkat kesalahan (43,3962%).
3	Pengukuran Similarity Tema pada Juz 30 Al Qur'an Menggunakan Teks Klasifikasi	Endang Supriyati dan Mohammad Iqbal	*Decision Tree*, *Support Vector Machine* dan *Naïve Bayes*	Hasil terbaik klasifikasi tersebut diperoleh algoritma *decision tree* dengan akurasi 74,34%, *SVM* dengan akurasi 75,84%, dan *Naïve Bayes* dengan akurasi 66,29%.

Tabel 2.3 Perbandingan Tinjauan Pustaka (Lanjutan)

No	Judul	Peneliti	Metode	Hasil
4	*Comparative Analysis for Topic Classification in Juz Al-Baqarah*	Mohamad Izzuddin Rahman, Noor Azah Samsudin, Aida Mustapha, dan Adeleke Abdullahi	*Support Vector Machine*, *Naive Bayes*, *Decision Tree*, dan *K-Nearest Neighbours*	Hasil akurasi yang diperoleh algoritma SVM dan algoritma *J48* memiliki kinerja akurasi terbaik secara keseluruhan 85% sedangkan *Naïve Bayes* memiliki hasil akurasi terakhir 71%. Ini menunjukkan bahwa tidak ada algoritma klasifikasi terbaik yang spesifik.

Tabel 2.4 Perbandingan Tinjauan Pustaka (Lanjutan)

No	Judul	Peneliti	Metode	Hasil
5	Implementasi *Cosine Similarity* dan *Naive Bayes Classifier* Untuk Mencari Dalil Pada Artikel Islami Berdasarkan Indeks Al-Qur'an	Lana Rahim	*Cosine Similarity* dan *Naïve Bayes*	Hasil uji coba yang dilakukan dengan menggunakan *confusion matrix* didapatkan: *accuracy* 38%, *precision* 9%, dan *recall* 8%.
6	*Comparative Analysis of Text Classification Algorithms for Automated Labelling of Quranic Verses*	Abdullahi O. Adeleke, Noor A. Samsudin, Aida Mustapha, dan Nazri M. Nawi	*k-Nearest Neighbor*, *Support Vector Machine*, dan *Naïve Bayes*	Terjemahan Al-Qur'an diklasifikasikan sebagai "Shahadah" (pilar pertama Islam) atau "Pray" (pilar kedua Islam). Didapatkan hasil bahwa algoritma klasifikasi teks mampu mencapai akurasi lebih dari 70% dalam memberi label pada ayat-ayat Al-Qur'an.

Tabel 2.5 Perbandingan Tinjauan Pustaka (Lanjutan)

No	Judul	Peneliti	Metode	Hasil
7	*News Sentiment Analysis Using Naive Bayes And Adaboost*	Erna Daniati	*Naive Bayes* dan *Adaboost*	Proses *training* dan *testing* menggunakan metode *Naïve Bayes* dengan kombinasi *AdaBoost*. Hasilnya, penggunaan *AdaBoost* dapat meningkatkan akurasi *Naïve Bayes*.
8	*Comparative Analysis of Text Mining Classification Algorithms for English and Indonesian Qur'an Translation*	Rahmat Hidayat dan Sekar Minati	*Support Vector Machine, Naive Bayes, k - Nearest Neighbor, dan J48 Decision Tree*	Hasil klasifikasi algoritma *SVM* memiliki akurasi terbaik untuk terjemahan bahasa Indonesia 81,443% dan *Naïve Bayes* memiliki akurasi terbaik untuk terjemahan bahasa Inggris, yang mencapai 78,35%.

Tabel 2.6 Perbandingan Tinjauan Pustaka (Lanjutan)

No	Judul	Peneliti	Metode	Hasil
9	Penerapan Teknik *Bagging* pada Algoritma Klasifikasi untuk Mengatasi Ketidakse-imbangan Kelas Dataset Medis	Rizki Tri Prasetio dan Pratiwi	*Bagging*	Setelah diterapkan teknik bagging pada lima dataset medis, *naïve bayes* memiliki akurasi paling tinggi untuk dataset *breast-cancer*, *heart-disease*, dan *pima-diabetes*. Sedangkan *k-nearest neighbor* memiliki akurasi yang paling baik untuk dataset *liver-disorder* dan *vertebral column*.

Bereferensi dari penelitian-penelitian di atas yang telah dirangkum pada Tabel 2.1 hingga Tabel 2.6, penulis akan melakukan penelitian mengenai pengklasifikasian terjemahan Bahasa Indonesia ayat-ayat Al-Qur'an dengan algoritma yang relatif baru untuk pengklasifikasian teks, yaitu algoritma

Bootstrap Aggregating (Bagging) dan *Adaptive Boosting (AdaBoost)*.

2.2. Landasan Teori

2.2.1. Kecerdasan Buatan (*Artificial Intelligence*)

Secara bahasa, istilah 'kecerdasan buatan' sering digunakan untuk menggambarkan mesin atau komputer yang meniru fungsi 'kognitif' yang diasosiasikan manusia dengan pikiran manusia, seperti 'belajar' dan 'pemecahan masalah'. Tujuan dari penelitian AI meliputi penalaran, representasi pengetahuan, perencanaan, pembelajaran, pemrosesan bahasa alami, persepsi dan kemampuan untuk bergerak dan memanipulasi objek (Wikipedia Contributors, 2020a).

Konsep dasar penelitian AI berawal di bidang pembelajaran mesin atau *machine learning* (ML), yaitu studi tentang algoritma komputer yang meningkatkan 'kecerdasan' mesin secara otomatis seiring dengan banyaknya 'pengalaman' atau pembelajaran yang dilakukan. Algoritma pembelajaran mesin membangun model matematika berdasarkan data sampel, yang dikenal sebagai '*training data*', untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk melakukan tugas. Pembelajaran mesin terkait erat dengan statistik komputasi, yang berfokus pada membuat prediksi menggunakan komputer. Studi tentang

optimasi matematika memberikan metode, teori dan domain aplikasi ke bidang pembelajaran mesin.

2.2.2. Pembelajaran Mesin (*Machine Learning*)

Melakukan pembelajaran mesin melibatkan pembuatan model, yang dilatih pada beberapa data pelatihan dan kemudian dapat memproses data tambahan untuk membuat prediksi. Berbagai jenis model telah digunakan dan diteliti untuk sistem pembelajaran mesin dengan berbagai kelebihan dan kekurangannya masing-masing (Wikipedia Contributors, 2020b).

Dalam pembelajaran mesin, klasifikasi *multi-label* dan masalah yang sangat terkait dari klasifikasi *multi-output* adalah varian dari masalah klasifikasi di mana beberapa label dapat ditugaskan untuk setiap *instance*. Klasifikasi *multi-label* adalah generalisasi dari klasifikasi *multi-class*, yang merupakan masalah label tunggal dalam mengkategorikan *instance* menjadi tepat satu dari lebih dari dua kelas. Dalam masalah *multi-label*, tidak ada batasan pada berapa banyak kelas dalam suatu *instance* (Wikipedia Contributors, 2020c).

Ada beberapa metode transformasi masalah untuk klasifikasi *multi-label*, yaitu transformasi ke dalam masalah klasifikasi biner atau *multi-class* dan *ensemble method*. Beberapa algoritma atau model klasifikasi telah disesuaikan dengan data *multi-label*, diantaranya:

1. *k-Nearest Neighbors*: algoritma *ML-kNN* memperluas *classifier k-NN* ke data *multi-label*.
2. *Decision Tree*: "*Clare*" adalah algoritma *C4.5* yang diadaptasi untuk klasifikasi *multi-label*; modifikasi melibatkan perhitungan entropi. *MMC*, *MMDT*, dan *SSC* yang disempurnakan *MMDT*, dapat mengklasifikasikan data *multi-label* berdasarkan atribut *multi-valued* tanpa mengubah atribut menjadi *single-values*. Mereka juga dinamai metode klasifikasi pohon keputusan *multi-valued* dan *multi-label*.
3. Metode *kernel* untuk keluaran vector
4. *Neural Networks*: *BP-MLL* adalah adaptasi dari algoritma *back-propagation* yang populer untuk pembelajaran *multi-label*.

2.2.3. *Text Mining*

Data mining adalah bidang studi dalam pembelajaran mesin, dan berfokus pada analisis data eksplorasi melalui pembelajaran tanpa pengawasan (*unsupervised learning*) untuk menghilangkan keacakan dan menemukan pola yang tersembunyi (Wikipedia Contributors, 2020b) karena metode *data mining* ini hampir selalu intensif secara komputasi (Dataflair Team, 2018). Berdasarkan tipe datanya, *data mining* dibagi beberapa kategori diantaranya *text mining*, *data stream mining*, *graph mining*, dan sebagainya.

Text mining (Witten, 2002) atau disebut juga dengan *text data mining* merupakan bidang yang berfokus pada usaha untuk mendapatkan informasi yang bermakna melalui pemrosesan bahasa alami melalui proses menganalisis teks untuk mengekstrak informasi yang berguna untuk tujuan tertentu dari berbagai sumber tertulis. Sumber tertulis dapat berupa situs web, buku, email, ulasan, artikel. Informasi berkualitas tinggi biasanya diperoleh melalui merancang pola dan tren melalui cara seperti pembelajaran pola statistic (Wikipedia Contributors, 2020d). *Text mining* biasanya melibatkan proses penataan teks input, memperoleh pola dalam data terstruktur, dan akhirnya evaluasi dan interpretasi dari output. 'Kualitas tinggi' dalam penambangan teks biasanya merujuk pada beberapa kombinasi relevansi, kebaruan, dan minat. Tugas *text mining* yang umum meliputi kategorisasi teks, pengelompokan teks, ekstraksi konsep/entitas, produksi taksonomi granular, analisis sentimen, peringkasan dokumen, dan pemodelan hubungan entitas.

2.2.4. *Preprocessing Text*

Text mining sangat tergantung pada tahapan *preprocessing* ini, karena pada tahapan ini data yang masih berupa data mentah atau tidak terstruktur disiapkan menjadi sumber data yang terstruktur untuk digunakan pada operasi

text mining (Feldman & Sanger, 2007). Tahapan *preprocessing text* diantaranya sebagai berikut (Bestari, Saptono, & Anggrainingsih, 2018):

1. *Case folding*

   Case folding adalah proses menghilangkan karakter selain huruf abjad dan mengubah semua huruf menjadi huruf kecil (*lowercase*) (Bestari et al., 2018).

2. *Tokenization*

   Tokenization atau tokenisasi adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token/bagian-bagian tertentu (kata) yang menjadi acuan pemisah antar token dapat berupa spasi maupun tanda baca (Manning, Raghavan, & Schütze, 2009).

3. *Stopword filtering*

   Stopword atau *stoplist* merupakan istilah dari daftar kata seperti kata depan, kata sambung, kata ganti, kata sifat dan lain sebagainya yang tidak relevan dijadikan indeks karena kemunculannya tidak unik untuk sebuah dokumen tertentu. Langkah pembersihan kata-kata yang tidak relevan dijadikan indeks tersebut disebut dengan *stopword filtering* (Rahutomo & Ririd, 2019). Beberapa contoh *stopword* dalam Bahasa Indonesia yaitu, ‘yang’, ‘di’, ‘dan’, ‘itu, ‘dengan’, dan lain

sebagainya (Doyle, 2000; Tala, 2003; Wibisono, 2008).

4. *Stemming*

   Stemming adalah proses dasar yang sering digunakan untuk mengefisienkan dan mengefektifkan proses *text mining*. *Stemmers* menghapus kata-kata imbuhan menjadi kata dasar yang berasal dari *stem* atau akar yang sama; contohnya, kata 'membuka', 'terbuka', 'pembuka' terklaster dengan akar kata 'buka'. Mengidentifikasi kata-kata dari akar yang sama meningkatkan sensitivitas pengambilan dengan meningkatkan kemampuan untuk menemukan dokumen yang relevan, tetapi sering dikaitkan dengan penurunan selektivitas, di mana pengelompokan menyebabkan makna yang berguna menjadi hilang. *Stemming* diharapkan meningkatkan *recall* tetapi mungkin mengurangi *precision* (Adriani, Asian, Nazief, Tahaghoghi, & Williams, 2007).

2.2.5. *Term Weighting*

Term weighting merupakan proses memberikan nilai terhadap sebuah istilah dalam dokumen. Proses pemberian nilai ini dapat dipengaruhi oleh jumlah dokumen, panjang dokumen, seringnya sebuah istilah muncul dan berbagai faktor

lainnya. Beberapa metode dalam melakukan proses *term weighting* diantaranya:

1. *Term Frequency* (TF)

   Frekuensi setiap kata dalam dokumen (TF) adalah bobot yang tergantung pada distribusi setiap kata dalam dokumen. Ini mengungkapkan pentingnya kata dalam dokumen.

2. *Inverse Document Frequency* (IDF)

   Inverse frekuensi dokumen dari setiap kata dalam basis data dokumen (IDF) adalah bobot yang tergantung pada distribusi setiap kata dalam basis data dokumen.

3. *Term Frequency-Inverse Document Frequency* (TF-IDF)

   TF-IDF adalah teknik yang menggunakan TF dan IDF untuk menentukan bobot suatu istilah. Skema TF-IDF sangat populer di bidang klasifikasi teks dan hampir semua skema pembobotan lainnya adalah varian dari skema ini.

4. *Information Gain* (IG)

   Gagasan dasar dari metode berbasis IG yaitu berapa banyak informasi yang diperoleh istilah tentang kategori setelah dikurangi dengan perolehan informasi dari istilah yang sama dengan kategori lainnya. Dapat dikatakan bahwa semakin tinggi perbedaan antara perolehan informasi istilah dari satu kategori dan rata-

rata kategori lainnya, semakin banyak istilah yang membantu dalam memisahkan kategori positif dan negatif (Mazyad, Teytaud, & Fonlupt, 2018).

5. *Mutual Information* (MI)

   MI ini mengukur pengurangan ketidakpastian suatu variabel acak ketika kita tahu tentang variabel lain. Metrik umumnya diterapkan untuk mengidentifikasi istilah penempatan (Nanas, Uren, & De Roeck, 2004).

6. *Term Frequency-Relevance Frequency* (TF-RF)

   RF (*relevance frequency*) merupakan skema baru dalam pembobotan untuk meningkatkan daya pembeda *term* untuk kategorisasi teks. Dalam metode pembobotan TF-RF, distribusi dokumen dan jumlah sampel positif dan negatif yang mengandung istilah tersebut digunakan untuk menghitung bobot istilah dalam dokumen (Man Lan, Sam-Yuan Sung, Hwee-Boon Low, & Chew-Lim Tan, 2005).

7. *Balanced Term Weighting Scheme* (BTWS)

   Prosedur utama BTWS terdiri dari menetapkan bobot, menerapkan frekuensi dokumen terbalik dan melakukan normalisasi berdasarkan panjang dokumen, dan melakukan operasi produk dalam untuk menentukan kesamaan dokumen akhir. Setelah menetapkan bobot negatif untuk ketentuan absen, BTWS menormalkan bobot positif dan negatif secara

independen sesuai dengan kepentingan global dan lokal dari masing-masing istilah (Jung, Parky, & Du, 2007).

8. *Aspect Bernoulli* (AB)

   Aspect Bernoulli (AB) model yang diusulkan oleh Bingham, Kaban dan Fortelius (2007) ini mengusulkan skema pembobolan term termodifikasi yang memperhitungkan statistik syarat absen sambil menghitung bobot persyaratan yang ada untuk meningkatkan kinerja TC. Karya ini dimotivasi oleh pemikiran bahwa istilah-istilah tidak independen untuk muncul dalam dokumen atau tidak; beberapa istilah akan hilang ketika istilah lain ada dalam dokumen dan sebaliknya. Dengan demikian, bobot ketentuan yang ada harus dipengaruhi oleh yang hilang.

9. *Modified Term Frequency-Inverse Document Frequency* (mTF-mIDF)

   Dalam skema *IDF* yang dimodifikasi yang diusulkan (dilambangkan oleh *mIDF*), rumus *IDF* direkonstruksi dengan memasukkan jumlah dokumen di mana *term t* tidak muncul. Dengan demikian, perbedaan antara jumlah total dokumen dalam koleksi dan jumlah dokumen di mana *term t* terjadi (*N - DFt*) digunakan dalam bentuk terbalik. Karena skema modifikasi yang diusulkan (*mTF* dan *mIDF*) diekstraksi dari formula

TF dan IDF standar, skema yang diusulkan dapat digunakan sebagai pengganti formula TF dan IDF standar dalam skema pembobotan TF-IDF tradisional (Sabbah et al., 2017).

2.2.6. *Term Frequency-Inverse Document Frequency (TF-IDF)*

Tiga komponen utama yang mempengaruhi pentingnya suatu istilah dalam dokumen adalah faktor *Term Frequency* (TF), faktor *Inverse Document Frequency* (IDF) dan normalisasi panjang dokumen. Frekuensi setiap kata dalam dokumen (TF) adalah bobot yang tergantung pada distribusi setiap kata dalam dokumen. Ini mengungkapkan pentingnya kata dalam dokumen. *Inverse* frekuensi dokumen dari setiap kata dalam basis data dokumen (IDF) adalah bobot yang tergantung pada distribusi setiap kata dalam basis data dokumen. Ini mengungkapkan pentingnya setiap kata dalam database dokumen. TF-IDF adalah teknik yang menggunakan TF dan IDF untuk menentukan bobot suatu istilah. Skema TF-IDF sangat populer di bidang klasifikasi teks dan hampir semua skema pembobotan lainnya adalah varian dari skema ini (Gaigole, Patil, & Chaudhari, 2013). Rumus untuk menghitung TF-IDF dituliskan pada persamaan berikut (Nurjannah, Hamdani, & Astuti, 2013):

Sebagaimana didefinisikan, (Xia & Chai, 2011) TF adalah frekuensi *term* dalam satu dokumen. *Term* disini bisa berupa kata atau frasa. Untuk dokumen, frekuensi untuk setiap *term* dapat sangat bervariasi. Oleh karena itu, frekuensi adalah atribut penting dari istilah untuk membedakan dirinya dari *term* yang lain. Terkadang, frekuensi suatu *term* digunakan langsung sebagai nilai TF. Artinya, nilai TF dari *term* i adalah

$$TF_i = tf_{ij}$$

dimana tf_i menyatakan frekuensi dari suatu *term* i pada dokumen j.

Karena jumlah frekuensi *term* mungkin sangat besar, rumus berikut ini juga sering digunakan untuk menghitung nilai TF.

$$TF_i = \log_2(tf_{ij})$$

Adapun untuk menghitung nilai IDF menggunakan rumus sebagai berikut.

$$IDF_i = \log_2(\frac{N}{n_j}) + 1 = \log_2(N) - \log_2(n_j) + 1$$

dimana N adalah jumlah total dokumen dalam koleksi dan n_j adalah jumlah dokumen yang mengandung setidaknya satu kemunculan *term* i.

2.2.7. Klasifikasi Teks

Klasifikasi merupakan bagian dari pembelajaran yang diawasi atau *supervised learning*. Klasifikasi teks (Dalal &

Zaveri, 2011) adalah pengelompokan teks ke sekumpulan kelas yang telah ditentukan sebelumnya menggunakan teknik pembelajaran mesin (*machine learning*). Klasifikasi biasanya dilakukan berdasarkan kata-kata atau fitur-fitur penting yang diekstraksi dari dokumen teks. Sebagian besar komunikasi dan dokumentasi saat ini sudah berbentuk digital, seperti dokumen elektronik, email, blog, dan lainnya. Karena banyaknya informasi tersebut, klasifikasi yang efisien dan pengambilan konten yang relevan menjadi sangat penting.

Klasifikasi teks otomatis memiliki banyak diaplikasikan, seperti pengindeksan untuk pengambilan dokumen, mengekstraks metadata secara otomatis, disambiguasi makna kata dengan mendeteksi topik yang dicakup oleh dokumen, dan mengatur serta mengelola katalog pada *web resource*. Seperti di bidang *text mining* lainnya, hingga tahun 1990an klasifikasi teks didominasi oleh teknik *ad hoc* "rekayasa pengetahuan" yang berusaha untuk mendapatkan aturan klasifikasi dari para ahli dan mengkodekannya ke dalam sistem yang dapat menerapkannya secara otomatis ke dokumen baru. Sejak saat itu, khususnya dalam komunitas riset, pendekatan yang dominan adalah menggunakan teknik pembelajaran mesin untuk menyimpulkan kategori secara otomatis dari serangkaian pelatihan dokumen pra klasifikasi (Witten, 2002).

2.2.8. *Adaptive Boosting* (*Adaboost*) *Classifier*

Boosting (Schapire, 1990; Schapire & Freund, 1997) adalah algoritma untuk meningkatkan kemampuan memprediksi sistem pembelajaran dan metode paling umum dalam mengkoordinasikan pembelajaran. *Boosting* memiliki dua masalah utama: bagaimana menyesuaikan *training set* untuk memungkinkan *weak classifier* untuk melakukan pelatihan dan bagaimana menggabungkan *weak classifier* yang diperoleh melalui pelatihan menjadi "kuat". Untuk mengatasi masalah di atas, Schapire dan Freund memperkenalkan algoritma *Adaboost* (*adaptive boosting*) pada tahun 1997, yang bekerja dengan menyesuaikan bobot tanpa perlu pembelajaran apriori.

AdaBoost (*Adaptive Boosting*), yang pertama kali diperkenalkan oleh Schapire dan Freund (1997), adalah algoritma *ensemble* populer yang meningkatkan algoritma *boosting* sederhana melalui proses berulang. Gagasan utama di balik algoritma ini adalah untuk memberikan lebih banyak fokus pada pola yang lebih sulit untuk diklasifikasi. Jumlah fokus diukur dengan bobot yang ditetapkan untuk setiap pola dalam set pelatihan. Awalnya, bobot yang sama ditetapkan untuk semua pola. Dalam setiap iterasi, bobot dari semua *instance* yang salah diklasifikasikan meningkat sedangkan bobot dari *instance* yang diklasifikasikan dengan benar berkurang. Sebagai akibatnya, *weak learner* dipaksa untuk

fokus pada contoh-contoh sulit dari pelatihan yang ditetapkan dengan melakukan iterasi tambahan dan menciptakan lebih banyak pengklasifikasi. Selanjutnya, bobot diberikan ke setiap *classifier* individu. Bobot ini mengukur akurasi keseluruhan *classifier* dan merupakan fungsi dari total bobot pola yang diklasifikasikan dengan benar. Dengan demikian, bobot yang lebih tinggi diberikan untuk pengklasifikasi yang lebih akurat. Bobot ini digunakan untuk klasifikasi pola baru (Rokach, 2010).

Studi dan penerapan algoritma *AdaBoost* berfokus pada masalah klasifikasi dalam memecahkan masalah dua kelas, label tunggal *multi-class*, multi-klasifikasi dan multi-label, kelas label tunggal, dan regresi (Chengsheng, Huacheng, & Bing, 2017).

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:
$D_1(i) = 1/m$. % Initialize the weight distribution
for $t = 1, \cdots, T$:
$h_t = \mathcal{L}(\mathcal{D}, D_t)$; % Train a base learner h_t from $\mathcal{D}$ using distribution D_t
$\epsilon_t = \Pr_{i \sim D_t}[h_t(\boldsymbol{x}_i) \neq y_i)]$; % Measure the error of h_t
$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$; % Determine the weight of h_t

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases} = \frac{D_t(i) \exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

% Update the distribution, where Z_t is a normalization % factor which enables D_{t+1} to be a distribution

end.

Output: $H(\boldsymbol{x}) = \text{sign}\left(f\left(\boldsymbol{x}\right)\right) = \text{sign} \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$

Gambar 2.1 Algoritma *AdaBoost* (Zhou, 2009)

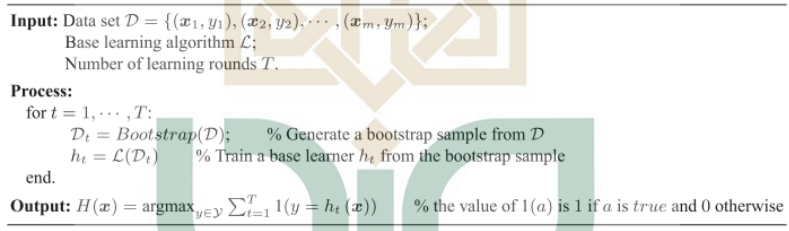
2.2.9. *Bootstrap Aggregating* (*Bagging*) *Classifier*

Bagging (Opitz & Maclin, 1999) adalah "*bootstrap*" metode *ensemble* yang menciptakan individu untuk ansambelnya dengan melatih setiap *classifier* pada redistribusi acak dari *training set*. Setiap *training set* pada *classifier* dihasilkan melalui penggambaran secara acak dengan penggantian, contoh N - di mana N adalah ukuran untuk *training set* asal; contoh asal dapat diulang dalam *training set* yang dihasilkan sementara yang lain mungkin ditinggalkan. Setiap *classifier* dalam ansambel dihasilkan dengan sampling acak berbeda dari *training set*.

Sama seperti *boosting*, teknik *bagging* meningkatkan akurasi *classifier* dengan menghasilkan model komposit yang menggabungkan beberapa *classifier* yang semuanya berasal dari *inducer* yang sama. Kedua metode mengikuti pendekatan pemungutan suara, yang diimplementasikan secara berbeda, untuk menggabungkan output dari pengklasifikasi yang berbeda. Dalam *boosting*, berbeda dengan *bagging*, masing-masing pengklasifikasi dipengaruhi oleh kinerja yang dibangun sebelum konstruksinya. Secara khusus, *classifier* baru lebih memperhatikan kesalahan klasifikasi yang dilakukan oleh pengklasifikasi yang dibangun sebelumnya di mana jumlah perhatian ditentukan sesuai dengan kinerjanya. Dalam *bagging*, setiap *instance* dipilih dengan probabilitas yang sama, sedangkan dalam *boosting*, *instance* dipilih dengan

probabilitas yang proporsional dengan nilai *weight*. Seperti yang disebutkan di atas, *bagging* membutuhkan pembelajar yang tidak stabil sebagai *inducer* dasar, sementara *boosting* instabilitas *inducer* tidak diperlukan, hanya saja tingkat kesalahan setiap *classifier* dijaga di bawah 0,5 (Rokach, 2010).

Breiman (1996) menunjukkan bahwa *bagging* efektif pada algoritma pembelajaran "tidak stabil" seperti *neural networks* dan *decision tree*, di mana perubahan kecil dalam *training set* menghasilkan perubahan besar dalam prediksi.


Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:
for $t = 1, \cdots, T$:
$\mathcal{D}_t = Bootstrap(\mathcal{D})$; % Generate a bootstrap sample from $\mathcal{D}$
$h_t = \mathcal{L}(\mathcal{D}_t)$ % Train a base learner h_t from the bootstrap sample
end.

Output: $H(\boldsymbol{x}) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} 1(y = h_t(\boldsymbol{x}))$ % the value of $1(a)$ is 1 if a is $true$ and 0 otherwise

Gambar 2.2 Algoritma *Boosting* (Zhou, 2009)

2.2.10. Evaluasi Model

Mengevaluasi model atau algoritma pembelajaran mesin sangat penting untuk setiap proyek. Evaluasi model bertujuan untuk memperkirakan akurasi generalisasi model pada data yang akan datang (tidak terlihat/keluar-sampel). Ada banyak jenis metode evaluasi yang tersedia untuk menguji suatu model, salah satunya *cross-validation*. *Cross-validation* adalah teknik yang melibatkan mempartisi dataset observasi asli ke dalam set pelatihan, digunakan untuk melatih model,

dan set independen yang digunakan untuk mengevaluasi analisis (Singh, 2019).

Teknik validasi silang yang paling umum adalah *k-fold cross-validation*, di mana dataset asli dipartisi menjadi *k* dengan ukuran sampel yang sama, yang disebut lipatan. *K* adalah angka yang ditentukan *user*, biasanya dengan 5 atau 10 sebagai nilai pilihannya. Ini diulang *k* kali, sehingga setiap kali, salah satu himpunan bagian *k* digunakan sebagai set uji/set validasi dan himpunan *k-1* lainnya disatukan untuk membentuk set pelatihan. Estimasi kesalahan dirata-rata pada semua percobaan *k* untuk mendapatkan efektivitas total dari model yang dibuat (Singh, 2019).

Seperti yang dapat dilihat, setiap titik data akan berada di set tes tepat satu kali dan akan berada di set pelatihan *k-1* kali. Ini secara signifikan mengurangi bias, karena model ini menggunakan sebagian besar data untuk pemasangan, dan itu juga secara signifikan mengurangi varians, karena sebagian besar data juga digunakan dalam set uji. Saling menukar pelatihan dan set tes juga menambah efektivitas metode ini (Singh, 2019).

Evaluation metrics (metrik evaluasi) model juga diperlukan untuk mengukur kinerja model. Pilihan metrik evaluasi bergantung pada tugas pembelajaran mesin yang diberikan (seperti *classification, regression, ranking, clustering, topic modelling*, antara lain). Beberapa metrik,

seperti *recall-precision*, berguna untuk banyak tugas (Agarwal, 2019). Beberapa metrik yang digunakan pada pembelajaran mesin diantaranya:

1. *Accuracy, Precision,* dan *Recall*

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

a. *Accuracy*

Akurasi (Agarwal, 2019) adalah metrik klasifikasi klasik. Sangat mudah dimengerti dan mudah cocok untuk masalah klasifikasi biner maupun multi kelas. Perhitungan nilai akurasi yaitu:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

Akurasi adalah proporsi hasil yang benar di antara jumlah total kasus yang diperiksa. Akurasi adalah pilihan evaluasi yang valid untuk masalah klasifikasi dengan ketidakseimbangan kelas.

b. *Precision*

Dalam tugas klasifikasi (Agarwal, 2019), *precision* untuk suatu kelas adalah jumlah *true positive* (yaitu jumlah item yang dilabeli dengan benar sebagai milik kelas positif) dibagi dengan jumlah total elemen yang dilabeli sebagai milik

kelas positif (yaitu jumlah dari *true positive* dan *false positive*, yang merupakan item yang salah diberi label sebagai milik kelas).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Precision yang tinggi berarti suatu algoritma mengembalikan hasil yang jauh lebih relevan daripada yang tidak relevan. *Precision* adalah pilihan metrik evaluasi yang valid ketika ingin sangat yakin dengan prediksi yang dilakukan.

c. *Recall*

Recall didefinisikan sebagai jumlah *true positive* dibagi dengan jumlah total elemen yang benar-benar milik kelas positif .

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall yang tinggi berarti suatu algoritma mengembalikan sebagian besar hasil yang relevan. *Recall* adalah pilihan metrik evaluasi yang valid ketika ingin menangkap sebanyak mungkin hasil positif (Agarwal, 2019).

2. *F1 Score*

F1 score adalah angka antara 0 dan 1 dan merupakan rata-rata harmonik dari *precision* dan *recall*. Karena

rata-rata harmonis dari daftar angka cenderung kuat terhadap unsur-unsur paling sedikit dari daftar, cenderung (dibandingkan dengan rata-rata aritmatika) untuk mengurangi dampak outlier besar dan memperparah dampak yang kecil:

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

F1 score adalah pilihan metrik evaluasi yang valid ketika ingin memiliki model dengan *precision* dan *recall* yang baik. Secara sederhana, semacam ***F1 score*** mempertahankan keseimbangan antara *precision* dan *recall* untuk *classifier* yang dilakukan. Jika ***precision*** rendah, *F1 score* juga akan rendah dan jika ***recall*** rendah pula, maka *F1 score* rendah (Agarwal, 2019).

3. *Categorical Crossentropy*

Logarithmic loss juga digeneralisasikan ke masalah multikelas. Pengklasifikasi dalam pengaturan multi-kelas harus menetapkan probabilitas untuk setiap kelas untuk semua contoh. Jika ada sampel *N* milik kelas *M*, maka *categorical crossentropy* adalah penjumlahan dari nilai -*ylogp*:

$$LogarithmicLoss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \times \log\,(p_{ij})$$

Dengan y_{ij} adalah 1 jika sampel *i* milik kelas *j*, selain itu 0. Lalu, p_{ij} adalah probabilitas *classifier* yang digunakan memprediksi sampel *i* milik kelas *j*.

Categorical crossentropy adalah pilihan metrik evaluasi yang valid ketika output dari *classifier* adalah probabilitas prediksi multikelas yang biasanya digunakan dalam kasus *Neural Networks*. Secara umum, meminimalkan *categorical crossentropy* memberikan akurasi yang lebih besar untuk pengklasifikasian (Agarwal, 2019).

4. *AUC*

AUC adalah area di bawah kurva ROC. AUC ROC menunjukkan seberapa baik probabilitas dari kelas positif dipisahkan dari kelas negatif. Kurva karakteristik operasi penerima (*receiver operating characteristic*), atau kurva ROC (Singh, 2019), adalah plot grafis yang menggambarkan kemampuan diagnostik sistem klasifikasi biner karena ambang diskriminasinya beragam.

Kurva ROC dibuat dengan memplot *true positive rate* (TPR) terhadap *false positive rate* (FPR) di berbagai pengaturan ambang batas. Tingkat *true positive* juga dikenal sebagai sensitivitas, *recall*, atau probabilitas deteksi dalam pembelajaran mesin. Tingkat *false*

positive juga dikenal sebagai kejatuhan atau probabilitas *false alarm*.

$$TPR = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

$$FPR = \frac{\text{FP}}{(\text{FP} + \text{TN})}$$

AUC adalah skala-invarian. Ini mengukur seberapa baik peringkat peringkat, dan bukan nilai absolutnya. Manfaat lain dari menggunakan AUC adalah bahwa itu adalah *classification-threshold-invariant* seperti *logarithmic loss*. Ini mengukur kualitas prediksi model terlepas dari apa *classification-threshold* yang dipilih, tidak seperti *F1 score* atau *accuracy* yang tergantung pada pilihan *threshold* (Singh, 2019).

2.2.11. Al-Qur'an

Al-Qur'an adalah kitab suci umat Islam yang berisi firman Allah yang diturunkan kepada Nabi Muhammad saw. dengan perantaraan malaikat Jibril untuk dibaca, dipahami, dan diamalkan sebagai petunjuk atau pedoman hidup bagi umat manusia (Tim Penyusun Bahasa Kamus Pusat, 2008). Al-Qur'an difirmankan langsung oleh Allah kepada Nabi Muhammad saw. melalui Malaikat Jibril, berangsur-angsur selama 22 tahun, 2 bulan dan 22 hari atau rata-rata selama 23 tahun, di mulai sejak tanggal 17 Ramadan, saat Nabi Muhammad saw. berumur 40 tahun hingga wafat pada tahun

632. Al-Qur'an dihormati oleh umat Muslim sebagai sebuah mukjizat terbesar Nabi Muhammad saw., sebagai salah satu tanda dari kenabian, dan merupakan puncak dari seluruh pesan suci (wahyu) yang diturunkan oleh Allah sejak Nabi Adam as. dan diakhiri dengan Nabi Muhammad saw. (Kontributor Wikipedia, 2019).

Al-Qur'an terdiri dari 114 surat, 6.236 ayat, dan 77.477 kata yang disusun menggunakan Bahasa Arab (Osman et al., 2016). Secara umum, Al-Qur'an terbagi menjadi 30 bagian yang dikenal dengan nama *juz*. Pembagian *juz* memudahkan mereka yang ingin menuntaskan pembacaan Al-Qur'an dalam kurun waktu 30 hari. Terdapat pembagian lain yang disebut *manzil*, yang membagi Al-Qur'an menjadi 7 bagian (Kontributor Wikipedia, 2019).

Upaya-upaya untuk mengetahui isi dan maksud Al-Qur'an telah menghasilkan proses penerjemahan (literal) dan penafsiran (lebih dalam, mengupas makna) dalam berbagai bahasa. Namun hasil usaha tersebut dianggap sebatas usaha manusia dan bukan usaha untuk menduplikasi ataupun mengganti teks yang asli dalam bahasa Arab, sebab teks yang asli memiliki ciri kebahasaan dan berbagai istilah khusus yang tidak ditemui dalam terjemahan bahasa lain. Dengan demikian, kedudukan terjemahan dan tafsir yang dihasilkan tidaklah sama dengan Al-Qur'an itu sendiri (Leaman, 2006).

Para *mufassir* menciptakan metode penafsiran dalam rangka mengkontekstualisasikan pemaknaan ayat-ayat Al-Qur'an. Metode-metode penafsiran yang digunakan adalah metode *tahlili* (analisis), *ijmali* (global), *muqaran* (perbandingan), dan *maudui* (tematik). Metode penafsiran *maudui* (tematik) merupakan metode yang terbaik dibandingkan dengan keempat metode lainnya, dikarenakan metode ini disajikan dalam bentuk tema-tema pokok yang berhubungan langsung dengan permasalahan kehidupan yang dihadapi oleh masyarakat sehingga penyajiannya terkesan simpel dan juga menjadi kelebihan serta keunikan dari metode ini (A. S. Hidayatullah, 2009). Dalam rangka memudahkan masyarakat dalam mengakses ayat-ayat Al-Qur'an yang telah dilakukan proses penafsiran secara tematik ini, dibuatlah indeks Al-Qur'an. Indeks Al-Qur'an berusaha memberikan informasi sebaik mungkin dalam menyajikan keberadaan ayat-ayat yang dibutuhkan oleh masyarakat dalam rangka membumikan Al-Qur'an di tengah-tengah masyarakat (A. S. Hidayatullah, 2009).

Indeks adalah sebuah daftar yang sistematis, mengandung istilah atau frase yang dilengkapi dengan petunjuk ke isi satu atau serangkaian dokumen, ke lokasi istilah atau frase itu dapat ditemukan. Oleh sebab itu, indeks dan pengindeksasian biasanya dikaitkan dengan pembuatan katalog atau klasifikasi. Namun, berbeda dengan katalogisasi

dan klasifikasi yang lebih terkonsentrasi pada "tentang apa"-nya dokumen (*aboutness*), maka pengindeksan lebih terkonsentrasi pada ekstraksi atau pengambilan kata atau istilah yang ada di dalam dokumen di halaman tertentu, lalu menempatkannya dalam daftar indeks secara terstruktur. Struktur inilah yang memungkin sebuah buku "merujuk" pembacanya ke bagian-bagian dari teks yang relevan untuk keperluannya. Indeks Al-Qur'an berfungsi untuk memudahkan masyarakat dalam pencarian ayat-ayat Al-Qur'an dengan menunjukkan lokasi ayat sesuai dengan kebutuhan mereka (A. S. Hidayatullah, 2009).

2.2.12. *Python*

Python adalah bahasa pemrograman tingkat tinggi (*high-level*) yang diciptakan oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991. Filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi-spasi yang signifikan. Konstruk bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu *programmer* menulis kode yang jelas dan logis untuk proyek skala kecil dan besar (Kuhlman, 2012).

Python dikatakan sebagai bahasa pemrograman tingkat tinggi dikarenakan (Kuhlman, 2012):

- Kode *Python* secara otomatis dikompilasi ke kode *byte* dan dieksekusi, *Python* cocok untuk digunakan sebagai

bahasa *scripting*, bahasa implementasi aplikasi web, dll.

- *Python* dapat diperluas dalam C dan C++ dan dapat memberikan kecepatan yang dibutuhkan bahkan untuk tugas komputasi intensif.
- Struktur konstruksinya yang kuat (*nested code blocks, functions, classes, modules,* dan *packages*) dan penggunaan objek dan pemrograman berorientasi objek yang konsisten, *Python* memungkinkan kita untuk menulis aplikasi yang jelas dan logis untuk tugas-tugas kecil dan besar.

Keunggulan lainnya dari *Python* yaitu merupakan salah satu bahasa pemrograman yang mendukung beberapa teknologi terkini seperti *machine learning* (pembelajaran mesin). Ada begitu banyak modul dan *library* yang dapat digunakan untuk menerapkan *machine learning* di dalam *Python*. Salah satu modulnya yang terkenal yaitu *scikit-learn*.

Scikit-learn adalah modul *Python* yang mengintegrasikan berbagai algoritma pembelajaran mesin yang canggih untuk masalah berskala menengah baik yang *supervised* dan *unsupervised*. Paket ini berfokus pada membawa pembelajaran mesin ke non-spesialis menggunakan *general-purpose* bahasa tingkat tinggi. Pembelajaran mesin yang terdapat dalam *scikit-learn* diantaranya *classification*,

regression, *clustering*, *dimensionality reduction*, *model selection*, dan *preprocessing*.

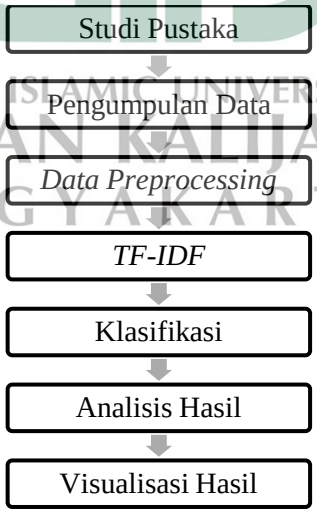
BAB III
METODE PENELITIAN

3.1. Metode Penelitian

Penelitian ini menggunakan metode *ensemble learning* yaitu *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* untuk meningkatkan hasil akurasi klasifikasi setiap terjemahan ayat Al-Qur'an berdasarkan tema-tema yang sudah ada.

3.2. Tahapan Penelitian

Untuk mencapai tujuan dari penelitian ini, terdapat beberapa tahapan yang dilakukan. Tahapan-tahapan tersebut digambarkan pada Gambar 3.1.



Gambar 3.1 Skema Alur Penelitian

3.2.1. Studi Pendahuluan

Tahap studi pendahuluan dilakukan untuk mengetahui konsep secara mendalam tentang teori-teori yang dipakai dalam penelitian. Peneliti mempelajari literatur-literatur yang pernah dilakukan oleh peneliti sebelumnya, yaitu dilakukan dengan membaca dan memahami informasi dasar, rumusan penelitian, hingga hasil penelitian dari berbagai jurnal, buku, tugas akhir, maupun situs-situs yang terkait penelitian. Selain itu, dalam tahap ini juga diperlukan pengkajian data ataupun informasi yang dibutuhkan sebagai referensi pendukung terkait objek penelitian.

3.2.2. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah Al-Qur'an terjemahan Bahasa Indonesia yang diterjemahkan oleh Kementrian Agama Republik Indonesia. Terjemahan tersebut didapatkan dari situs www.qurandatabase.org. Pembagian kelas tema untuk setiap ayatnya berdasarkan 2 (dua) indeks tematik, yaitu indeks tematik yang disusun oleh Kementerian Agama Republik Indonesia (Kemenag) dan indeks tematik Al-Qur'an Al-Hadi yang ditulis oleh DR. Ahmad Lutfi Fathullah yang terdapat pada situs https://Al-Qur'analhadi.com. Selain data Al-Qur'an, penelitian ini menggunakan data artikel islami yang bersumber dari beberapa web contohnya seperti

Replubika.co.id yang di dalamnya terdapat beberapa dalil atau ayat Al-Qur'an terkait.

3.2.3. *Data Preprocessing*

Pengolahan data dilakukan menggunakan bahasa pemrograman *Python*. Setelah dataset disiapkan, dilakukan *text preprocessing* untuk mengubah data teks yang tidak terstruktur atau sembarang menjadi data yang terstruktur. Tahapan proses yang dilakukan yaitu *case-folding*, *tokenizing*, dan *filtering stopword*. *Case-folding* yaitu penyamaan *case* penulisan menjadi huruf kecil semua. *Tokenizing* yaitu penghilangan tanda baca seperti titik (.), koma (,), tanda tanya (?), dan sebagainya. *Filtering stopword* yaitu penghapusan kata hubung seperti "yang", "dan", "di", "dari", dan lainnya. Pada penelitian ini tidak dilakukan proses stemming, dikarenakan stemming tidak signifikan mempengaruhi tingkat akurasi hasil klasifikasi yang dilakukan (A. F. Hidayatullah, 2015).

Pada bahasa pemrograman *Python* terdapat banyak *library* yang dapat digunakan pada tahapan *data preprocessing* ini diantaranya *regular expression* (*regex*) ataupun *scikit-learn*.

3.2.4. *Term Frequency-Inverse Document Frequency* (*TF-IDF*)

Setelah dilakukan tahap *data preprocessing*, selanjutnya dilakukan proses *TF-IDF* (*Term Frequency-Inverse Document Frequency*). Di dalam proses ini dilakukan pembobotan kata pada setiap dokumen dalam korpus. Bobot token (kata) semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen (Robertson, 2004). Proses *TF-IDF* dapat dilakukan menggunakan *library* pada *Python* diantaranya yaitu dengan *scikit-learn* maupun *gensim*.

3.2.5. Klasifikasi

Tahapan selanjutnya dilakukan pengklasifikasian dataset dua indeks tematik yang telah terlabeli menggunakan algoritma *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* yang keduanya pada Python terdapat pada modul *scikit-learn* pada bagian *library* pembelajaran mesin *supervised – ensemble methods*. lalu dilakukan analisis terhadap kedua hasil yang didapatkan algoritma tersebut.

Dalam *scikit-learn* (Pedregosa et al., 2011), metode *Bagging* diberikan sebagai satuan *meta-estimator BaggingClassifier* dengan input *base estimators* yang ditentukan *user* bersama dengan parameter yang menentukan strategi untuk menggambar himpunan bagian acak. Pada

modul ini juga terdapat metode boosting yang terkenal yaitu *AdaBoost* yang dikenalkan oleh Freund dan Schapire (1996). Modul *AdaBoostClassifier* diimplementasikan untuk klasifikasi *multi-class*.

3.2.6. Analisis Hasil

Setelah tahapan pengolahan data selesai, selanjutnya dilakukan analisis hasil penelitian. Analisis hasil ini dilakukan untuk mengetahui hasil pengimplementasian algoritma *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* dalam melakukan analisis berdasarkan kelas menggunakan berbagai evaluasi model. Dalam tahap ini akan didapatkan hasil evaluasi model algoritma *Bagging* dan *AdaBoost* dalam melakukan analisis yang dijalankan dengan modul *scikit-learn*.

3.2.7. Visualisasi Hasil

Hasil analisis evaluasi model divisualisasikan dalam bentuk tabel dan berbagai jenis diagram dengan menunjukkan hasil klasifikasi masing-masing algoritma yang telah dilakukan.

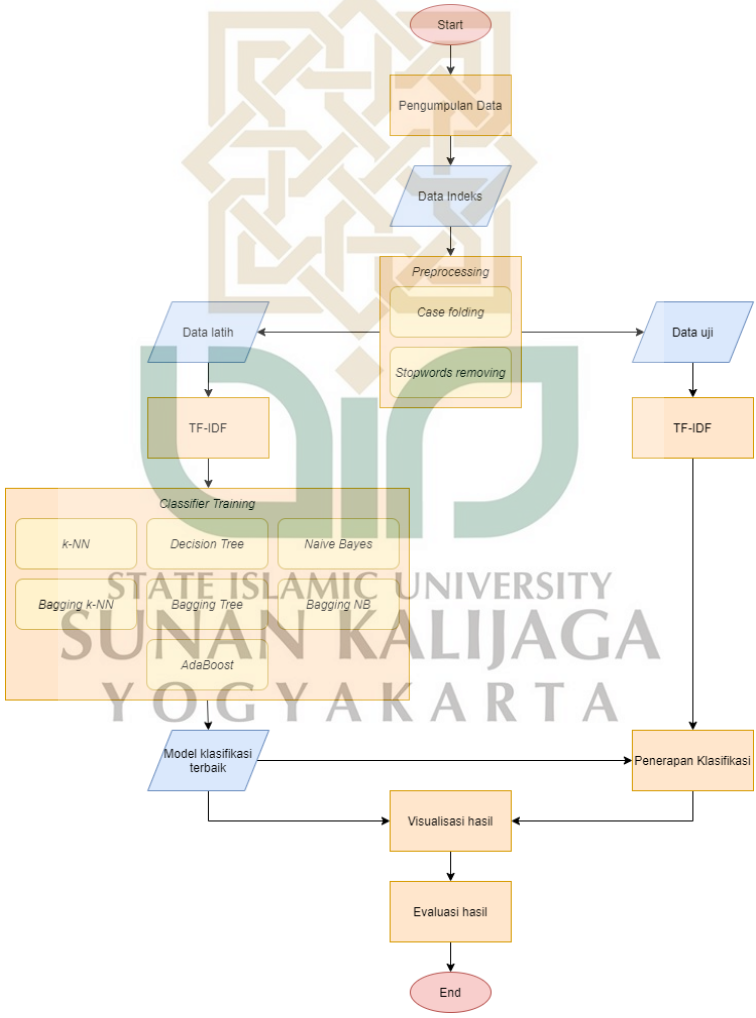
3.3. Kebutuhan Sistem

Penelitian ini membutuhkan beberapa instrumen seperti perangkat keras dan perangkat lunak. Perangkat lunak

yang dibutuhkan diantaranya sistem operasi *Windows*, dan *Anaconda* yang didalamnya lengkap dengan *tools* yang dapat digunakan untuk melakukan komputasi pembelajaran mesin dengan bahasa pemrograman *Python*. Sedangkan perangkat keras yang dibutuhkan yaitu laptop atau PC dengan penyimpanan kurang lebih 1GB dan RAM minimal 4GB.

BAB IV
HASIL DAN PEMBAHASAN

Pada bab sebelumnya telah dipaparkan mengenai alur penelitian secara umum, dan berikutnya pemaparan mengenai proses analisis secara rinci ditunjukan pada Gambar 4.1.



Gambar 4.1 *Flowchart* Proses Analisis

4.1. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah Al-Qur'an terjemahan Bahasa Indonesia yang diterjemahkan oleh Kementrian Agama Republik Indonesia dengan total jumlah data terjemahan ayat sebanyak 6236 ayat. Pembagian kelas tema untuk setiap ayatnya berdasarkan 2 (dua) Al-Qur'an tematik, yaitu Al-Qur'an tematik yang disusun oleh Kementerian Agama Republik Indonesia (Kemenag) dan Al-Qur'an tematik Al-Qur'an Al-Hadi yang ditulis oleh DR. Ahmad Lutfi Fathullah yang terdapat pada situs https://Al-Qur'analhadi.com. Kedua Al-Qur'an tematik tersebut memiliki jumlah pembagian tema yang berbeda, Al-Qur'an tematik Kemenag memiliki 15 (lima belas) pembagian tema dengan 5508 ayat bertemakan dan Al-Qur'an tematik Al-Qur'an Al-Hadi memiliki 8 (delapan) pembagian tema dengan 4699 ayat bertemakan. Pembagian tema kedua Al-Qur'an tematik disebutkan pada Tabel 4.1.

Selain data Al-Qur'an, penelitian ini menggunakan data artikel islami yang bersumber dari beberapa *web* contohnya seperti Republika.co.id. Artikel islami yang digunakan yaitu di dalamnya tercantum minimal 3 (tiga) dalil atau ayat Al-Qur'an terkait guna sebagai acuan prediksi. Data artikel islami ini digunakan untuk melakukan uji klasifikasi tema serta prediksi dalil yang terkait menggunakan algoritma pembelajaran klasifikasi yang sudah ditentukan.

Tabel 4.1 Pembagian Tema Al-Qur'an Tematik

Kemenag	Al-Hadi
1. Arkanul Islam 2. Iman 3. Al Quran 4. Cabang-cabang Ilmu 5. Amal 6. Dakwah Kepada Allah 7. Jihad 8. Manusia dan hubungan kemasyarakatan 9. Akhlak 10. Peraturan tentang Harta 11. Hukum 12. Negara dan Kemasyarakatan 13. Pertanian dan Perdagangan 14. Sejarah dan Kisah-kisah 15. Agama-agama	1. Aqidah 2. Syariah 3. Akhlak 4. Ilmu 5. Kisah 6. Alam Dunia 7. Alam Ghaib 8. Alam Akhirat

4.2. Pengolahan Data

Data yang digunakan pada penelitian ini adalah Al-Qur'an terjemahan Bahasa Indonesia dengan total jumlah data terjemahan ayat sebanyak 6236 ayat. Terjemahan ayat-ayat tersebut dilakukan *preprocessing* sebelum memasuki tahap analisa. Tahapan *preprocessing* yang dilakukan yaitu *case folding*. Contoh data terjemahan ayat dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Contoh Data Terjemahan Ayat Al-Qur'an

No.	Terjemahan
1	Maka jika kamu tidak dapat membuat(nya) -- dan pasti kamu tidak akan dapat membuat(nya), peliharalah dirimu dari neraka yang bahan bakarnya manusia dan batu, yang disediakan bagi orang-orang kafir.
2	Allah Pencipta langit dan bumi, dan bila Dia berkehendak (untuk menciptakan) sesuatu, maka (cukuplah) Dia hanya mengatakan kepadanya: "Jadilah!" Lalu jadilah ia.
3	Dan sesungguhnya Allah telah mengambil perjanjian (dari) Bani Israil dan telah Kami angkat diantara mereka 12 orang pemimpin dan Allah berfirman: "Sesungguhnya Aku beserta kamu, sesungguhnya jika kamu mendirikan shalat dan menunaikan zakat serta beriman kepada rasul-rasul-Ku dan kamu bantu mereka dan kamu pinjamkan kepada Allah pinjaman yang baik sesungguhnya Aku akan menutupi dosa-dosamu. Dan sesungguhnya kamu akan Kumasukkan ke dalam surga yang mengalir air didalamnya sungai-sungai. Maka barangsiapa yang kafir di antaramu sesudah itu, sesungguhnya ia telah tersesat dari jalan yang lurus.

4.2.1. *Case Folding*

Pada tahap c*ase folding* dilakukan proses menghilangkan karakter selain huruf abjad dan tanda hubung (-) tunggal dan mengubah semua huruf menjadi huruf kecil (*lowercase*), karena besar (*uppercase*) atau kecilnya huruf dapat mempengaruhi hasil dari perhitungan TF (*Term Frequency*). Tanda hubung tidak dihilangkan dikarenakan tanda hubung digunakan untuk menyambung unsur kata ulang sehingga maknanya berbeda. Selain itu, kata 'tidak' pada teks ditambahkan karakter tanda hubung tepat setelahnya sehingga

kata ‘tidak’ menjadi satu kesatuan dengan kata setelahnya, hal ini dilakukan karena kata ‘tidak’ memberikan makna negasi sehingga makna ayat akan berubah jika kata ‘tidak’ berdiri sendiri. Hasil dari proses *case folding* dapat dilihat pada Tabel 4.4 dan Tabel 4.5.

Tabel 4.3 Hasil *Case Folding* Data Terjemahan Ayat Al-Qur’an

No.	Terjemahan
1	maka jika kamu tidak-dapat membuatnya dan pasti kamu tidak-akan dapat membuatnya peliharalah dirimu dari neraka yang bahan bakarnya manusia dan batu yang disediakan bagi orang-orang kafir
2	allah pencipta langit dan bumi dan bila dia berkehendak untuk menciptakan sesuatu maka cukuplah dia hanya mengatakan kepadanya jadilah lalu jadilah ia
3	dan sesungguhnya allah telah mengambil perjanjian dari bani israil dan telah kami angkat diantara mereka orang pemimpin dan allah berfirman sesungguhnya aku beserta kamu sesungguhnya jika kamu mendirikan shalat dan menunaikan zakat serta beriman kepada rasul-rasul-ku dan kamu bantu mereka dan kamu pinjamkan kepada allah pinjaman yang baik sesungguhnya aku akan menutupi dosa-dosamu dan sesungguhnya kamu akan kumasukkan ke dalam surga yang mengalir air didalamnya sungai-sungai maka barangsiapa yang kafir di antaramu sesudah itu sesungguhnya ia telah tersesat dari jalan yang lurus

Proses *case folding* di atas dapat dilakukan dengan menggunakan *library regular expression* (*regex*) pada *Python*.

4.2.2. *Stopword Filtering*

Tahapan *stopword filtering* dilakukan proses menghilangkan kata-kata yang tidak relevan dijadikan indeks seperti kata ‘yang’, ‘di’, ‘dan’, ‘itu, ‘dengan’, dan lain sebagainya.

Tabel 4.4 Hasil *Case Folding* Data Terjemahan Ayat Al-Qur’an

No.	Terjemahan
1	jika kamu tidak-dapat membuatnya pasti kamu tidak-akan membuatnya peliharalah dirimu neraka bahan bakarnya manusia batu disediakan orang-orang kafir
2	allah pencipta langit bumi bila berkehendak menciptakan maka cukuplah hanya mengatakan kepadanya jadilah lalu jadilah
3	sesungguhnya allah mengambil perjanjian bani israil telah angkat diantara orang pemimpin allah berfirman sesungguhnya aku beserta kamu sesungguhnya kamu mendirikan shalat menunaikan zakat beriman rasul-rasul-ku kamu bantu kamu pinjamkan allah pinjaman baik sesungguhnya aku menutupi dosa-dosamu dan sesungguhnya kamu kumasukkan dalam surga mengalir air didalamnya sungai-sungai barangsiapa kafir antaramu itu sesungguhnya telah tersesat jalan lurus

Proses *stopword filtering* di atas dapat dilakukan dengan menggunakan *library Sastrawi* pada *Python* dengan menggunakan fungsi *StopWordRemover*.

4.3. Analisis

4.3.1. *Term Frequency-Inverse Document Frequency* (*TF-IDF*)

Metode *term weighting* yang dilakukan pada penelitian ini adalah *TF-IDF* (*Term Frequency-Inverse Document Frequency*) yaitu proses pembobotan kata pada setiap dokumen dalam korpus. Bobot token (kata) semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. Langkah-langkah dalam melakukan proses *TF-IDF* sebagai berikut:

1. Hitung nilai *TF* (*term frequency*) yang merupakan jumlah kemunculan suatu *term* pada suatu dokumen.
2. Hitung *DF* (*document frequency*) yaitu jumlah dokumen yang mengandung suatu term tertentu.
3. Hitung IDF (*inverse document frequency*) dengan rumus berikut:

   $$idf = \log \frac{N}{df}$$

   dengan N merupakan banyaknya dokumen dan df (*document frequency*) yang merupakan nilai pada langkah ke-2.
4. Kalikan nilai *TF* dari masing-masing *term* pada masing-masing dokumen dengan nilai *IDF*. Hasil inilah yang merupakan bobot dari setiap *term* pada dokumen.

Berikut merupakan contoh proses perhitungan *TF-IDF* beberapa terjemahan ayat yang digunakan untuk proses klasifikasi ditunjukkan pada Tabel 4.5 dan 4.6 berikut.

Tabel 4.5 Contoh Dokumen untuk Pembobotan TF-IDF

Dokumen	Terjemahan
D1	menyebut nama allah maha pemurah maha penyayang
D2	engkaulah sembah hanya engkaulah kami meminta pertolongan
D3	tiadakah kamu mengetahui kerajaan langit bumi kepunyaan allah tiada bagimu allah seorang pelindung maupun seorang penolong

Tabel 4.6 Contoh Proses Pembobotan TF-IDF

Term	Dokumen			*df*	*idf*	*TF-IDF*		
	D1	D2	D3			D1	D2	D3
allah	1	0	2	2	0.18	0.18	0	0.35
bagimu	0	0	1	1	0.48	0	0	0.48
bumi	0	0	1	1	0.48	0	0	0.48
engkaulah	0	2	0	1	0.48	0	0.95	0
hanya	0	1	0	1	0.48	0	0.48	0
kami	0	1	0	1	0.48	0	0.48	0
kamu	0	0	1	1	0.48	0	0	0.48
kepunyaan	0	0	1	1	0.48	0	0	0.48
kerajaan	0	0	1	1	0.48	0	0	0.48
langit	0	0	1	1	0.48	0	0	0.48
maha	2	0	0	1	0.48	0.95	0	0
maupun	0	0	1	1	0.48	0	0	0.48
meminta	0	1	0	1	0.48	0	0.48	0
mengetahui	0	0	1	1	0.48	0	0	0.48

Tabel 4.7 Contoh Proses Pembobotan TF-IDF (Lanjutan)

Term	Dokumen			*df*	*idf*	*TF-IDF*		
	D1	**D2**	**D3**			**D1**	**D2**	**D3**
menyebut	1	0	0	1	0.48	0.48	0	0
nama	1	0	0	1	0.48	0.48	0	0
pelindung	0	0	1	1	0.48	0	0	0.48
pemurah	1	0	0	1	0.48	0.48	0	0
penolong	0	0	1	1	0.48	0	0	0.48
penyayang	1	0	0	1	0.48	0.48	0	0
pertolongan	0	1	0	1	0.48	0	0.48	0
sembah	0	1	0	1	0.48	0	0.48	0
seorang	0	0	2	1	0.48	0	0	0.95
tiada	0	0	1	1	0.48	0	0	0.48
tiadakah	0	0	1	1	0.48	0	0	0.48

Proses pembobotan di atas secara otomatis dilakukan pada penelitian ini menggunakan fungsi *TfidfVectorizer* yang terdapat pada *library sklearn* pada bahasa pemrograman *Python*.

4.3.2. Klasifikasi

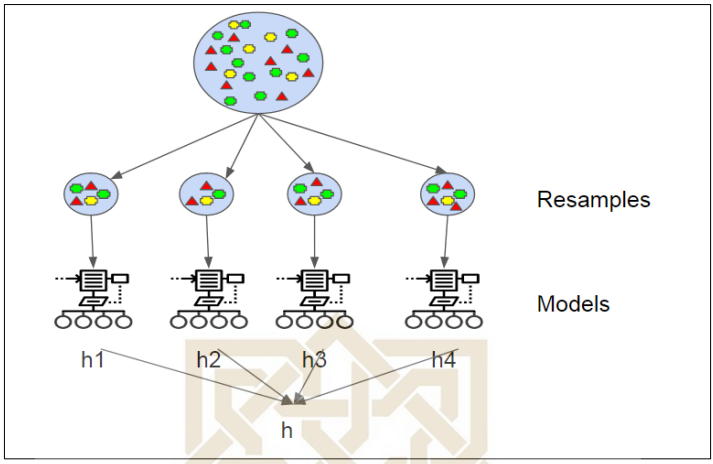
Klasifikasi atau pengelompokan teks yang dilakukan pada penelitian ini berdasarkan kelas dari bab indeks tematik yaitu Al-Qur'an Al-Hadi dan Kemenag. Masing-masing indeks dilakukan *training* klasifikasi dengan beberapa algoritma klasifikasi dan selanjutnya dilakukan optimasi

menggunakan metode *ensemble learning: Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)*.

4.3.2.1. *Bootstrap Aggregating (Bagging)*

Algoritma pembelajaran *ensemble Bagging* menciptakan beberapa model klasifikasi dari algoritma klasifikasi dasar dengan beberapa data dari data *training*. Beberapa algoritma dasar yang digunakan yaitu algoritma klasifikasi *Naive Bayes*, *Decision Tree*, dan *K-Nearest Neighbours* (*KNN*). Tahapan proses algoritma *Bagging* diantaranya:

1. Memilih beberapa data dari data training untuk dijadikan model klasifikasi.
2. Dataset yang terpilih dilakukan pengklasifikasian menggunakan algoritma klasifikasi dasar sehingga terciptalah sebuah model klasifikasi.
3. Selanjutnya lakukan kembali proses nomor satu dan dua sehingga banyak model klasifikasi yang tercipta.
4. Ambillah beberapa data untuk dilakukan pengujian pada masing-masing model yang telah tersedia.
5. Hasil klasifikasi dari setiap model lalu di *voting* dan jumlah voting terbanyak merupakan kelas hasil klasifikasi metode *Bagging*.



Gambar 4.2 Proses Klasifikasi *Bagging*

Proses klasifikasi metode *Bagging* di atas secara otomatis dilakukan pada penelitian ini menggunakan fungsi *BaggingClassifier* yang terdapat pada *library sklearn* pada bahasa pemrograman *Python*.

4.3.2.2. *Adaptive Boosting (AdaBoost)*

Metode *AdaBoost* digunakan untuk menciptakan beberapa *weak classifiers* menjadi *classifier* yang 'kuat'. Berikut adalah langkah proses dari algoritma *AdaBoost*:

1. Inisialisasi bobot ke semua *training points*. Setiap training pada *AdaBoost classifier* ditentukan dengan nilai bobot yang merupakan kebalikan dari jumlah total *point* yang mempresentasikan dataset *training*.

$$\omega = \frac{1}{N}$$

2. Menghitung nilai *error* untuk semua *weak classifiers* yang diberikan.

$$\epsilon = \sum_{wrong} \omega_i$$

3. Pilihlah *classifier* dengan nilai *error* yang terendah.
4. Menghitung *voting power* dengan menggunakan rumus berikut:

$$\alpha = \frac{1}{2}\log\frac{1-\epsilon}{\epsilon}$$

Hal ini menunjukkan seberapa besar kekuatan *weak classifiers* dalam hal seberapa akurat ia jika dibandingkan dengan *weak classifiers* yang lain.

5. Menambahkan *classifier* pada *ensemble classifier* dan periksa apakah *classifier* yang dilakukan cukup bagus.

$$f(x_i) = \sum_{t=1}^{T} \alpha_t h_t(x_i)$$

6. Setelah *classifier* dirasa sudah baik, perbaruilah nilai bobot pada setiap *training points* berdasarkan yang sudah terklasifikasi dengan benar atau salah oleh *classifier* sebelumnya. Jika sudah terklasifikasi dengan benar, nilai *training points* menurun dan jika salah nilai *training points* meningkat d jika terklasifikasi dengan benar
jika terklasifikasi dengan salah
berikut:

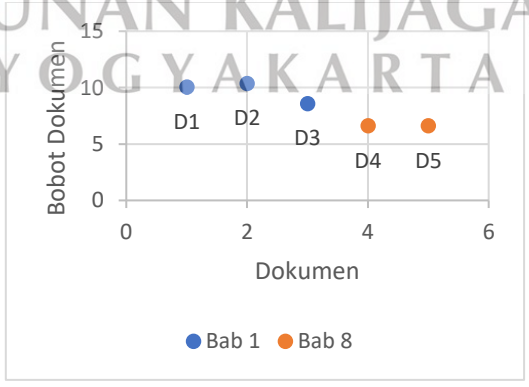
$$w_{new} = \begin{cases} \frac{w_{old}}{2(1-\epsilon)} \\ \frac{w_{old}}{2\epsilon} \end{cases}$$

7. Selanjutnya ulangi proses dengan melakukan perhitungan pada nilai *error* untuk setiap kemungkinan *classifier* hingga sekiranya menemukan *classifier* yang terbaik.

Tabel 4.8 Contoh Data untuk Proses Klasifikasi

Dokumen	Terjemahan	Bab Indeks
D1	menyebut nama allah maha pemurah maha penyayang	1
D2	engkaulah sembah hanya engkaulah kami meminta pertolongan	1
D3	taatilah allah rasul kamu diberi rahmat	1
D4	memasuki api sangat panas neraka	8
D5	aku memasukkannya dalam neraka saqar	8

Data ayat terjemahan yang disajikan pada Tabel 4.8 divisualisasikan menggunakan diagram *scatter* pada Gambar 4.3.



Gambar 4.3 Diagram *Scatter Training Points*

Nilai bobot diinisialisasikan sesuai dengan langkah pertama, dimana nilai N pada contoh ini yaitu 5.

Tabel 4.9 Bobot *Training Points*

Point	**Bobot**
W_{D1}	1/5
W_{D2}	1/5
W_{D3}	1/5
W_{D4}	1/5
W_{D5}	1/5

Selanjutnya, menghitung nilai *error* untuk semua *training points* dengan mempertimbangkan setiap *decision stump* yaitu *decision tree* yang hanya memiliki satu *split*. Beberapa contoh kombinasi *decision tree* diantaranya:

- Semua point yang berada di bawah 6 adalah Bab 8.
- Semua point yang berada di bawah 9 adalah Bab 8.
- Semua point yang berada di bawah 12 adalah Bab 1.
- Semua point yang berada di atas 6 adalah Bab 1.
- Semua point yang berada di atas 9 adalah Bab 1.
- Semua point yang berada di atas 12 adalah Bab 1.

Tabel 4.10 Nilai *Error* Kombinasi *Decision Tree*

Kombinasi	*Wrong*	*Error*
$x < 6$	D4, D5	2/5
$x < 9$	D3	1/5
$x < 12$	D4, D5	2/5
$x > 6$	D4, D5	2/5
$x > 9$	D3	1/5
$x > 12$	D4, D5	2/5

Dari tabel di atas dapat dilihat bawah kombinasi ke-2 dan ke-5 yang memiliki nilai *error* terendah yaitu dengan satu *point* yang salah terklasifikasi dengan nilai *error* 1/5. Selanjutnya dilanjutkan dengan menghitung nilai *voting power* sebagai berikut:

$$\epsilon = 1/5 \qquad (1)$$

$$\alpha = \frac{1}{2}\log\frac{1-\frac{1}{5}}{\frac{1}{5}} = \frac{1}{2}\log 4 \qquad (2)$$

$$h(x) = \frac{1}{2}\log 4 * F(x < 9) \qquad (3)$$

Jika *classifier* sudah cukup baik, maka hentikan perhitungan *AdaBoost* ini. Namun jika belum mendapatkan *classifier* yang baik maka ulangi langkah-langkah di atas dengan mengganti nilai bobot yang sesuai persamaan pada langkah nomor tujuh.

Proses klasifikasi metode *AdaBoost* di atas secara otomatis dilakukan pada penelitian ini menggunakan fungsi

AdaBoostClassifier yang terdapat pada *library sklearn* pada bahasa pemrograman *Python*.

4.4. Hasil

4.4.1. Pengujian Performa Algoritma

Proses klasifikasi dilakukan terhadap kedua indeks yaitu Al-Qur'an Al-Hadi dan Kemenag yang masing-masing memiliki jumlah bab yang berbeda yaitu 8 untuk Al-Hadi dan 15 untuk Kemenag. Klasifikasi dilakukan menggunakan algoritma pembelajaran *ensemble* yaitu *Bagging* dan *AdaBoost*. Dalam pengujiannya kedua algoritma tersebut menggunakan *base estimator* berupa algoritma klasifikasi dasar yaitu *Decision Tree, K-NN,* dan *Naïve Bayes* untuk *Bagging* dan *AdaBoost* dengan *base estimator* tetapnya yaitu *Decision Tree*. Uji klasifikasi dilakukan dengan perhitungan *precision, recall, f1-score, accuracy,* dan evaluasi *cross validation* yang hasilnya ditampilkan dalam tabel-tabel berikut.

Tabel 4.11 Performa Algoritma Klasifikasi pada Al-Qur'an Tematik Al-Hadi

Algoritma	Precision	Recall	F1-Score	Accuracy	CV
KNN	0.46	0.37	0.39	0.47	0.32
Bagging KNN	0.46	0.42	0.43	0.49	0.31
Naïve Bayes	0.44	0.24	0.23	0.41	0.33
Bagging NB	0.44	0.24	0.23	0.41	0.32
Decision Tree	0.74	0.46	0.49	0.58	0.26
Bagging Tree	0.55	0.54	0.54	0.58	0.32
AdaBoost	0.32	0.26	0.24	0.36	0.33

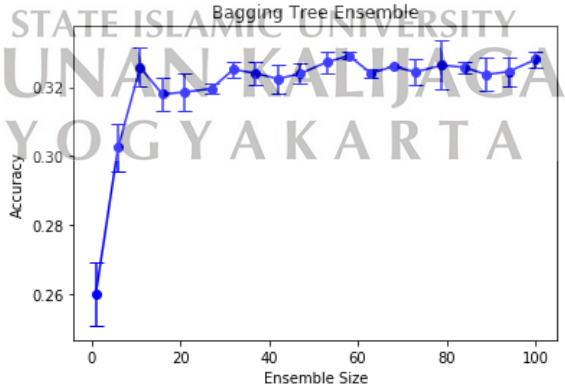
Tabel 4.12 Performa Algoritma Klasifikasi pada Al-Qur'an Tematik Kemenag

Algoritma	Precision	Recall	F1-Score	Accuracy	CV
KNN	0.41	0.22	0.26	0.43	0.32
Bagging KNN	0.38	0.30	0.33	0.45	0.30
Naïve Bayes	0.29	0.09	0.08	0.37	0.32
Bagging NB	0.29	0.10	0.08	0.37	0.32
Decision Tree	0.63	0.25	0.30	0.50	0.27
Bagging Tree	0.43	0.41	0.42	0.50	0.30
AdaBoost	0.17	0.10	0.08	0.32	0.31

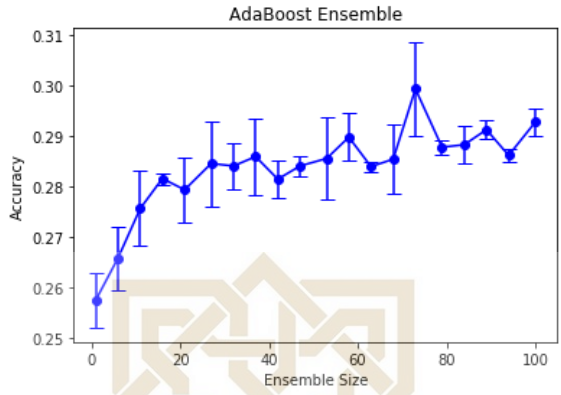
Dari kedua tabel performa di atas dapat dilihat bahwa *Bagging* dan *AdaBoost* dapat meningkatkan akurasi algoritma pembelajaran dasar *Decision Tree* dengan masing-masing 32% dan 33% pada indeks Al-Hadi dan 30% dan 31% pada

indeks Kemenag. Pembelajaran *ensemble Bagging* pada *Decision Tree* mencapai akurasi yang lebih tinggi dibandingkan dengan pembelajaran ensemble *Bagging* pada *k-NN* dan *Naïve Bayes* karena *k-NN* dan *Naïve Bayes* kurang sensitif terhadap gangguan pada sampel *training*, dan karenanya mereka disebut *stable learners*. Mengkombinasikan *stable learners* pada pembelajaran *ensemble* kurang menguntungkan karena *ensemble* tidak akan membantu meningkatkan kinerja generalisasi.

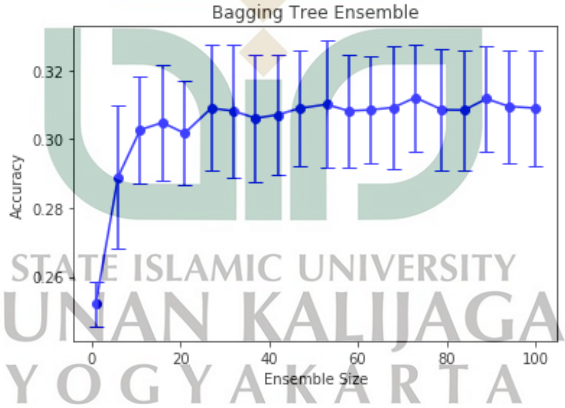
Selain menentukan algoritma untuk *base estimator*, banyaknya estimator atau banyaknya *ensemble* yang dilakukan juga dapat menentukan nilai akurasi dari algoritma pembelajaran *ensemble* ini. Pengujian dalam menentukan banyaknya *estimator* dengan nilai akurasi evaluasi *cross validation* ditunjukkan pada diagram berikut.



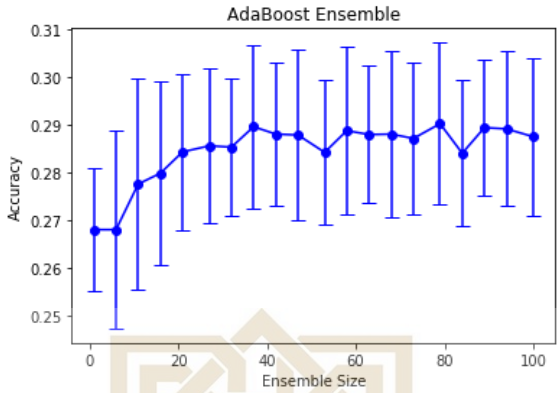
Gambar 4.4 Diagram *Bagging Ensemble Size* pada Al-Qur'an Tematik Al-Hadi



Gambar 4.5 Diagram *AdaBoost Ensemble Size* pada Al-Qur'an Tematik Al-Hadi



Gambar 4.6 Diagram *Bagging Ensemble Size* pada Al-Qur'an Tematik Kemenag



Gambar 4.7 Diagram *AdaBoost Ensemble Size* pada Al-Qur'an Tematik Kemenag

Gambar-gambar di atas menunjukkan bagaimana akurasi tes meningkat dengan besarnya *ensemble*. Berdasarkan hasil *cross validation*, kita dapat melihat keakuratan meningkat hingga sekitar 10 *estimator* dasar dan kemudian dataran tinggi sesudahnya. Dengan demikian, menambahkan *estimator* dasar di atas 10 hanya akan meningkatkan kompleksitas komputasi tanpa perolehan akurasi yang lebih baik pada kedua data, baik tema pada Al-Qur'an tematik Al-Hadi maupun tema pada Al-Qur'an tematik Kemenag.

4.4.2. Pengujian Data Artikel Islam

Pada bagian ini artikel-artikel islami yang telah dikumpulkan akan dijadikan sebagai *data test*, lalu diuji dengan algoritma *Bagging Decision Tree*. Sebelum melakukan prediksi menggunakan algoritma klasifikasi, artikel-artikel

tersebut dilakukan proses *summarization* (pengambilan intisari artikel) dan *preprocessing*.

Tabel 4.13 Contoh Data Artikel Islami

Artikel	Dalil Artikel
"Namun, tahukah Anda ada tiga sifat jiwa dalam diri yang disebutkan dalam Alquran? Berikut tiga sifat dalam Alquran sebagaimana Republika.co.id rangkum dari buku Belajar Mudah Memahami Hikmah karya Abinya Nasha. 1. Al Ammarah bi suuâ€™, yaitu suka menyuruh kepada keburukan. Kata tersebut bermakna bahwa jiwa pada dasarnya memiliki sifat yang cenderung melakukan keburukan. Maka dari itu, setiap orang pada dasarnya memiliki sifat untuk melakukan hal yang buruk. 2 Lawwamah, yaitu menyesali diri. Dalam sifat ini, manusia sangat diwajarkan ketika merasa menyesal atas diri sendiri dan cenderung mencela dirinya. Seperti yang dijelaskan dalam firman Allah Annafsullawwamah, yaitu suatu keadaan di mana jiwa menyesali keadaan diri karena merasa kurang melakukan kebaikan dan menyesal atas keburukan yang dilakukan. Dalam hal ini, jiwa memiliki kesadaran akan hal itu. 3. Muthmainnah, yaitu sifat jiwa yang memperoleh ketenangan. Menurut Ibnu Qayyim dalam kitab Ighatsat al-Lahfan min Masyayidisy Syaithan, apabila jiwa merasa tenteram kepada Allah SWT tenang dengan mengingat-Nya, dan bertobat kepada-Nya, rindu bertemu dengan-Nya, dan menghibur diri dengan dekat kepada-Nya, maka ialah jiwa yang dalam keadaan muthmainnah. Maka demikianlah sesungguh jiwa memiliki kecenderungan untuk berbuat buruk karena setiap jiwa punya hawa nafsu. Namun, permasalahannya adalah bagaimaan kita menahan diri utuk tidak dituntut oleh keburukan tersebut"	[12:53] [75:2] [89:27] [89:28] [89:29] [89:30]

Tabel 4.14 Hasil *Summarization*

Artikel
Maka dari itu, setiap orang pada dasarnya memiliki sifat untuk melakukan hal yang buruk.Annafsullawwamah, yaitu suatu keadaan di mana jiwa menyesali keadaan diri karena merasa kurang melakukan kebaikan dan menyesal atas keburukan yang dilakukan.

Tahapan *summarization* di atas dapat dilakukan dengan menggunakan *library* pada bahasa pemrograman *Python* yaitu *gensim*.

Tabel 4.15 Hasil *Preprocessing*

Artikel
dari setiap orang dasarnya memiliki sifat melakukan burukannafsullawwamah suatu keadaan mana jiwa menyesali keadaan diri merasa kurang melakukan kebaikan menyesal atas keburukan yang dilakukan

Setelah artikel melalui tahap *summarization* dan *preprocessing*, selanjutnya dilakukan klasifikasi untuk menentukan bab dan dalil dengan menggunakan metode *Bagging Tree* pada tema di Al-Qur'an Al-Hadi. Artikel diklasifikasikan terhadap tema yang ada untuk memprediksi pada bab mana artikel tersebut terklasifikasikan, lalu hasil tema yang didapat menjadi acuan untuk memilih ayat-ayat dan artikel diklasifikasikan ke dalam ayat-ayat tersebut. Sehingga didapatkanlah hasil prediksi bab berdasarkan tema pada indeks

Al-Qur'an beserta dalil yang sekiranya sesuai pada bab tersebut.

Tabel 4.16 Hasil Klasifikasi

Artikel	Bab Prediksi	Dalil Prediksi	Dalil Artikel
dari setiap orang dasarnya memiliki sifat melakukan burukannafsullawwamah suatu keadaan mana jiwa menyesali keadaan diri merasa kurang melakukan kebaikan menyesal atas keburukan yang dilakukan	5	[58:4]	[12:53] [75:2] [89:27] [89:28] [89:29] [89:30]

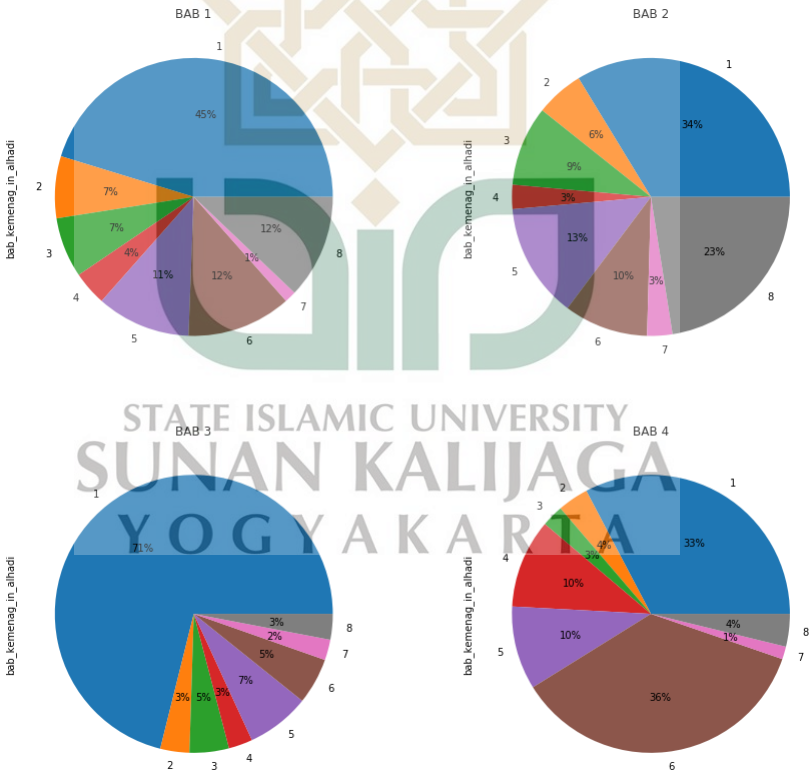
Dari hari klasifikasi di atas dapat dilihat bahwa prediksi dalil tidak sesuai dengan dalil yang terdapat pada artikel. Dalil pada artikel menunjukkan bahwa ayat-ayat tersebut berada pada tema 1 dan 3 pada Al-Qur'an Al-Hadi, sedangkan hasil dari algoritma *Bagging Tree* memprediksi masuk ke dalam tema 5. Hal ini menunjukkan bahwa algoritma ini kurang mampu dalam memprediksi tema maupun dalil pada artikel islami. Tetapi hal tersebut juga perlu dilakukan peninjauan ulang oleh para pakar dibidang hal ini seperti pakar tafsir Al-Qur'an.

4.4.3. Persilangan Bab Tematik

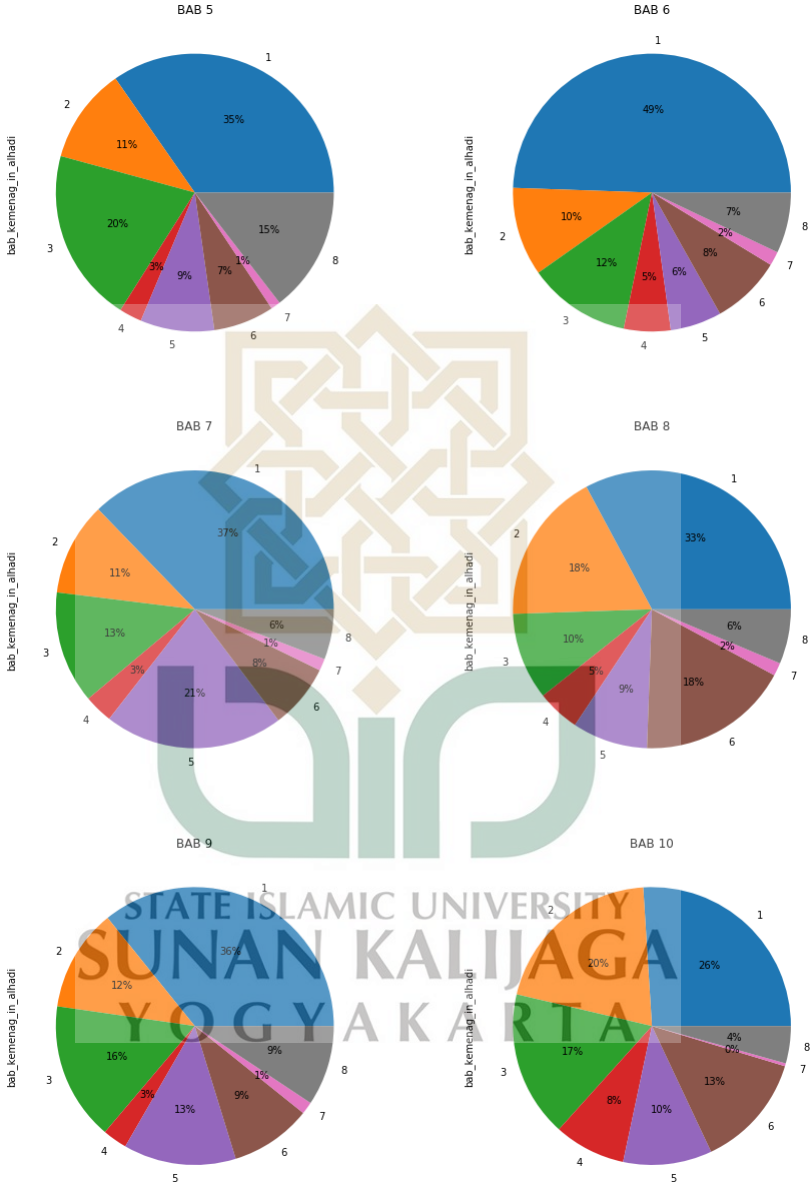
Al-Qur'an Al-Hadi dan Kemenag masing-masing memiliki jumlah bab yang berbeda yaitu 8 untuk Al-Hadi dan

15 untuk Kemenag yang rinciannya terdapat pada Tabel 4.1 sebelumnya. Dengan perbedaan jumlah bab yang sangat signifikan ini maka peneliti ingin menguji bagaimana sebaran pembagian tema jika tema pada bab pertama diuji dengan tema pada bab kedua begitupun sebaliknya.

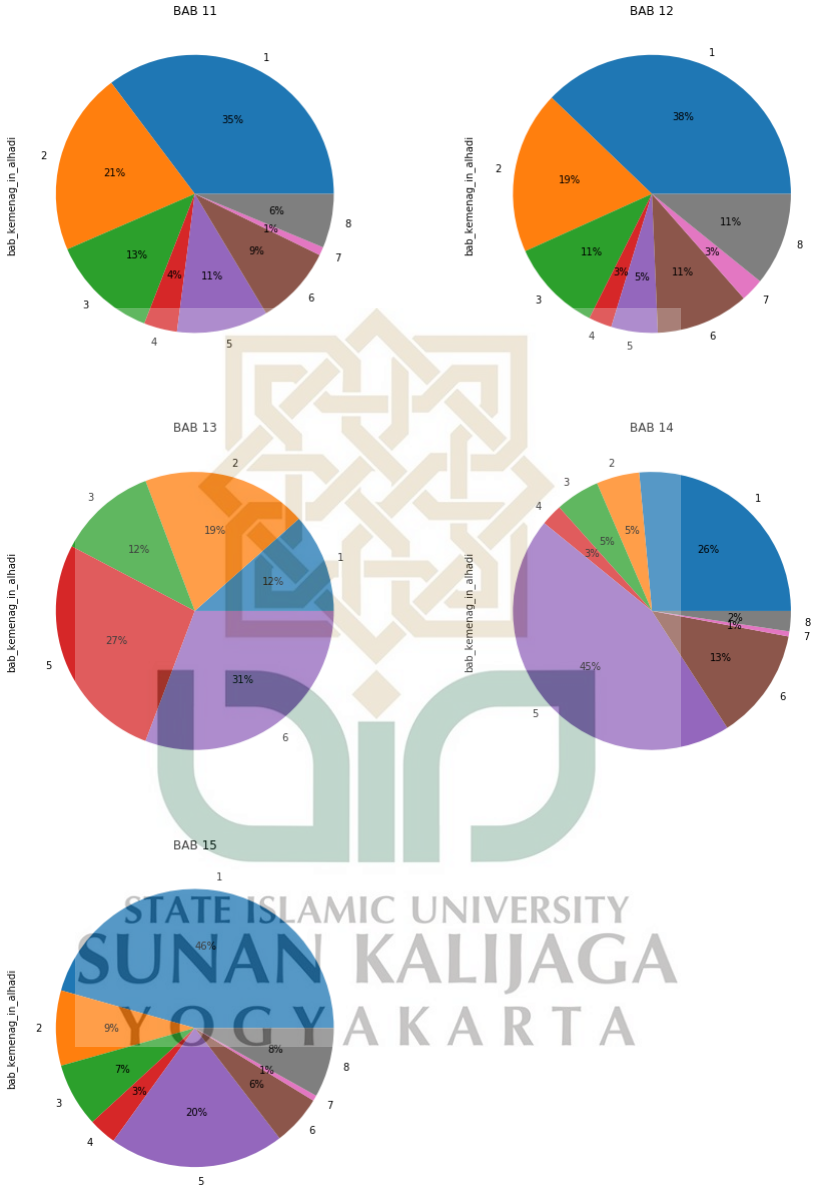
Berikut merupakan pembagian bab masing-masing tema dengan *data train* tema pada Al-Qur'an tematik Al-Hadi dan *data test* tema pada Al-Qur'an tematik Kemenag.



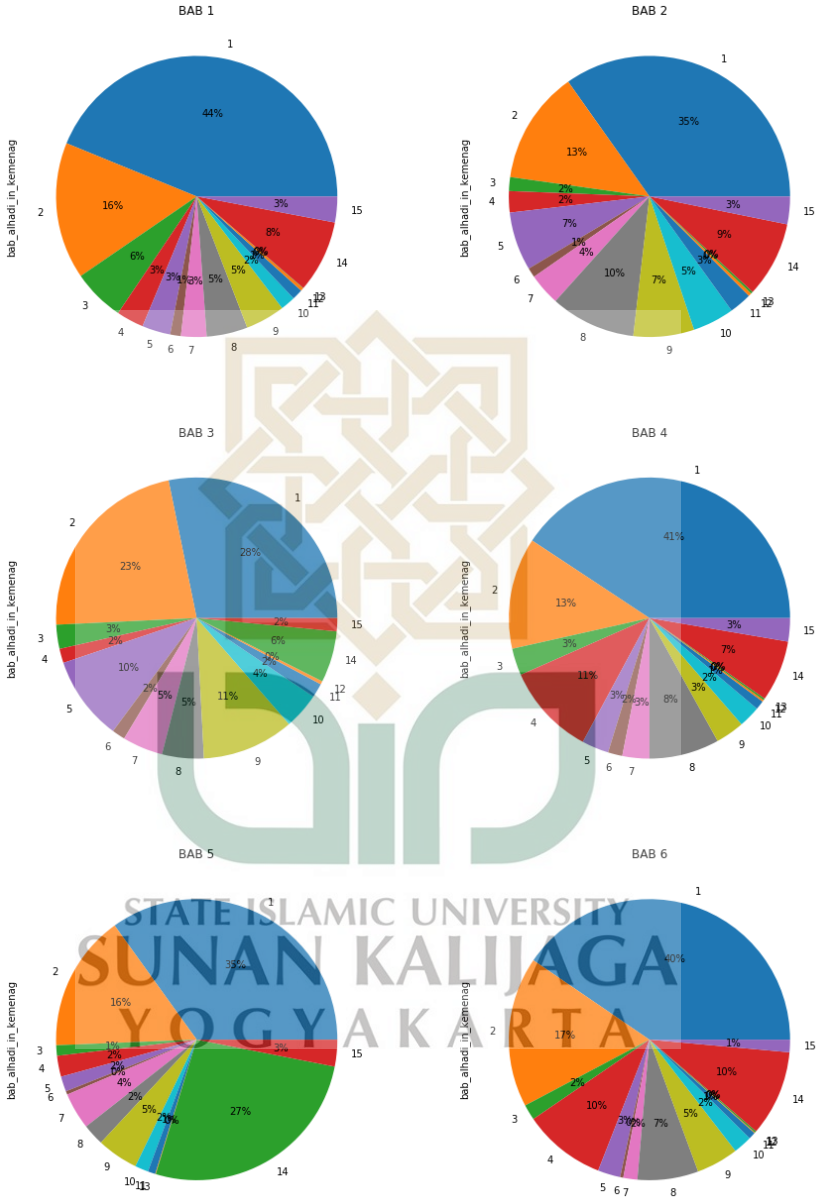
Gambar 4.8 Sebaran Pembagian Bab Tematik Kemenag pada Al-Qur'an Tematik Al-Hadi



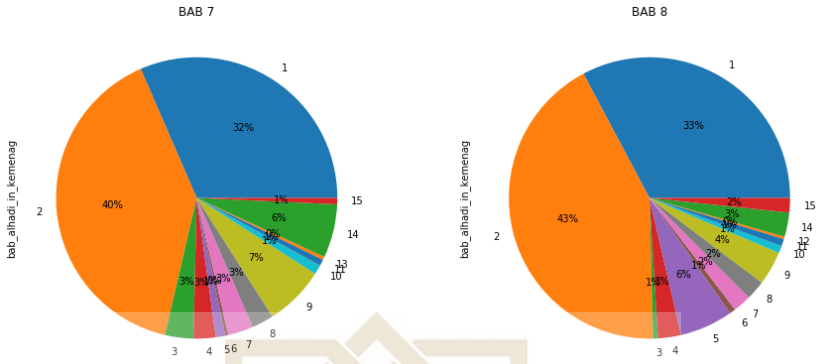
Gambar 4.9 Sebaran Pembagian Bab Tematik Kemenag pada Al-Qur'an Tematik Al-Hadi (Lanjutan)



Gambar 4.10 Sebaran Pembagian Bab Tematik Kemenag pada Al-Qur'an Tematik Al-Hadi (Lanjutan)



Gambar 4.11 Sebaran Pembagian Bab Tematik Al-Hadi pada Al-Qur'an Tematik Kemenag



Gambar 4.12 Sebaran Pembagian Bab Tematik Al-Hadi pada Al-Qur'an Tematik Kemenag (Lanjutan)

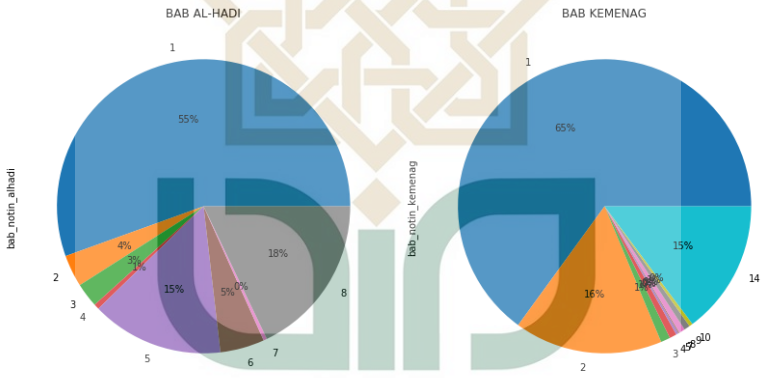
Berdasarkan hasil sebaran, baik dari pengklasifikasian yang dilakukan oleh *Bagging Tree* dan *AdaBoost*, pembagian 15 bab tematik Kemenag pada tematik Al-Hadi dan 8 bab tematik Al-Hadi pada tematik Kemenag, distribusi bab asal yang dihasilkan oleh *classifier* merata pada setiap bab targetnya. Untuk mengetahui apakah penyilangan bab tematik menggunakan klasifikasi ini hasilnya tepat atau tidak, diperlukan pengkajian lain baik dengan pembelajaran lain maupun seorang pakar untuk meninjaunya.

4.4.4. Klasifikasi Data yang Tidak Terlabeli

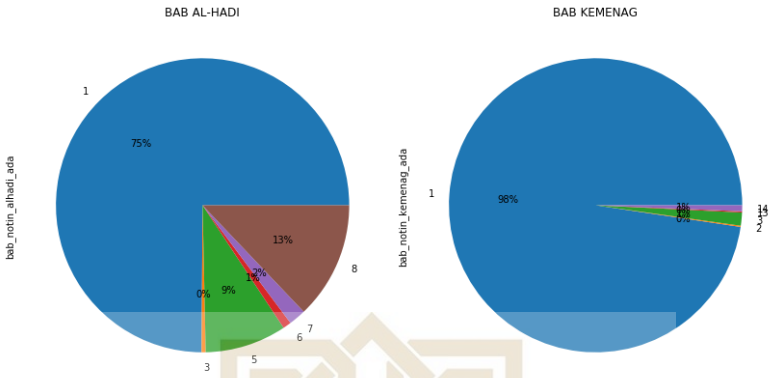
Jumlah ayat pada Al-Qur'an sebanyak 6236 ayat. Al-Qur'an tematik Kemenag memiliki 15 pembagian tema dengan 5508 ayat bertemakan dan Al-Qur'an tematik Al-Qur'an Al-Hadi memiliki 8 pembagian tema dengan 4699 ayat

bertemakan. Dengan demikian, terdapat 728 ayat yang belum bertemakan pada Al-Qur'an tematik Kemenag dan 1537 ayat yang belum bertemakan pada Al-Qur'an tematik Al-Hadi.

Seluruh data pada kedua bab tematik, masing-masing dilakukan *train* menggunakan algoritma klasifikasi. Selanjutnya data ayat Al-Qur'an yang belum terindeks diuji menggunakan pembelajaran klasifikasi *Bagging Tree* pada masing-masing bab tematik tersebut.



Gambar 4.13 Hasil Klasifikasi Ayat yang Belum Terindeks pada Kedua Bab Tematik oleh *Bagging Tree*



Gambar 4.14 Hasil Klasifikasi Ayat yang Belum Terindeks pada Kedua Bab Tematik oleh *AdaBoost*

Hasil pembagian bab yang dilakukan oleh algoritma klasifikasi *Bagging Tree* pada ayat yang tidak terdapat pada bab tematik Al-Hadi distribusinya tersebar keseluruh 8 bab yang ada pada bab tematik Al-Hadi tersebut, dengan 55% ayat terklasifikasi pada Bab 1 tentang Aqidah, 18% ayat pada Bab 8 tentang Alam Akhirat, 15% ayat pada Bab 5 tentang Kisah, dan lain sebagainya. Sedangkan pada Al-Qur'an tematik Kemenag distribusi ayatnya menyebar tetapi tidak ke keseluruhan bab yang ada pada bab tematik. Bab yang tidak terklasifikasi yaitu diantaranya, Bab 6 tentang Dakwah Kepada Allah, Bab 11 tentang Hukum, Bab 12 tentang Negara dan Kemasyarakatan, Bab 13 tentang Pertanian dan Perdagangan, dan Bab 15 tentang Agama-Agama.

Berbeda dengan *Bagging Tree*, pengklasifikasian yang dilakukan oleh algoritma *AdaBoost* distribusinya cenderung

terpusat pada suatu bab. Pada Al-Qur'an tematik Al-Hadi, ayat yang belum bertemakan terklasifikasi pada 6 bab dengan 75% ayat berada pada Bab 1 yaitu tentang Aqidah. Dua bab lainnya yang tidak terklasifikasi yaitu Bab 2 tentang Syariah dan Bab 4 tentang Ilmu. Sedangkan pada Al-Qur'an tematik Kemenag dari total bab sebanyak 15, hanya 5 bab yang terklasifikasikan dengan pembagian sebagian 98% ayat berada pada Bab 1 yaitu tentang Arkanul Islam.

Dalam melakukan evaluasi apakah pengklasifikasian yang dilakukan oleh algoritma pembelajaran klasifikasi *Bagging Tree* terhadap data ayat yang belum bertemakan ini hasilnya tepat atau tidak, diperlukan pengkajian langsung oleh seorang pakar dalam bidang tafsir Al-Qur'an untuk meninjaunya.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa metode *ensemble* algoritma klasifikasi pembelajaran *Bootstrap Aggregating (Bagging)* dan *Adaptive Boosting (AdaBoost)* dapat meningkatkan nilai akurasi klasifikasi pembelajaran dengan *base estimator* algoritma klasifikasi *Decision Tree*. Hasil rata-rata klasifikasi pada kedua tema indeks Al-Qur'an baik Al-Hadi maupun Kemenag didapatkan nilai *precision* sebesar 0.49, *recall* sebesar 0.48, *f1-score* sebesar 0.49, *accuracy* 54%, dan evaluasi *cross validation* sebesar 31% pada *Bagging Tree* dan *precision* sebesar 0.25, *recall* sebesar 0.18, *f1-score* sebesar 0.16, *accuracy* 34%, dan evaluasi *cross validation* sebesar 32% pada *AdaBoost*.

Dari segi *precision* dan *recall*, algoritma *Bagging* memiliki hasil yang lebih baik dibandingkan dengan *AdaBoost*. Hal ini menunjukkan bahwa hasil klasifikasi setiap ayat terhadap tema dalam kedua Al-Qur'an tematik yang dilakukan algoritma *Bagging* lebih valid dan meyakinkan dibandingkan dengan algoritma *AdaBoost* walaupun hasil dari akurasi *AdaBoost* lebih baik.

Meskipun kedua algoritma tersebut dapat meningkatkan hasil akurasi klasifikasi, dengan melihat hasil evaluasi model, kedua algoritma tersebut belum mampu melakukan klasifikasi yang baik pada dataset terjemahan Al-Qur'an bahasa Indonesia. Hal tersebut dilihat dari besarnya nilai akurasi yang didapatkan yaitu masih relatif rendah (± 30%). Hasil tersebut cenderung rendah dikarenakan satu ayat dalam Al-Qur'an dapat berada dalam beberapa bab tematik Al-Qur'an sehingga dapat menurunkan tingkat akurasi pengklasifikasian yang dilakukan.

5.2. Saran

Pada penelitian ini masih banyak sekali kekurangan. Maka dari itu penulis menyarankan beberapa hal untuk penelitian selanjutnya, diantaranya:

1. Penelitian selanjutnya dapat melakukan percobaan dengan menggunakan metode *term weighting* yang lainnya yang lebih sesuai untuk dataset dengan *multi-class* dan *multi-label*.
2. Membandingkan kembali dengan metode pembelajaran mesin lainnya yang mendukung khususnya untuk dataset dengan *multi-class* dan *multi-label* sehingga dapat meningkatkan hasil pengklasifikasian pada dataset Al-Qur'an tematik.

3. Mengevaluasi hasil klasifikasi yang dihasilkan oleh algoritma pembelajaran mesin kepada pakar-pakar di bidangnya, dalam hal ini pakar Al-Qur'an dan Tafsir.

DAFTAR PUSTAKA

Adeleke, A. O., Samsudin, N. A., Mustapha, A., & Nawi, N. (2017). Comparative Analysis of Text Classification Algorithms for Automated Labelling of Quranic Verses. *International Journal on Advanced Science Engineering and Information Technology*, *7*(4), 1419–1427. https://doi.org/10.18517/ijaseit.7.4.2198

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., & Williams, H. E. (2007). Stemming Indonesian : A Confix-Stripping Approach. *ACM Transactions on Asian Language Information Processing*, *6*(4). https://doi.org/10.1145/1316457.1316459.

Agarwal, R. (2019). The 5 Classification Evaluation metrics every Data Scientist must know. Diambil 20 Februari 2020, dari https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226

Al-Kabi, M. N., Wahsheh, H. A., Alsmadi, I. M., & Al-Akhras, A. M. A. (2015). Extended Topical Classification of Hadith Arabic Text. *International Journal on Islamic Applications in Computer Science and Technology*, *3*(3), 13–23.

Bestari, D. P. B., Saptono, R., & Anggrainingsih, R. (2018). ACADEMIC ARTICLES CLASSIFICATION USING

NAIVE BAYES CLASSIFIER (NBC) METHOD. *Jurnal Ilmiah Teknologi dan Informasi*, *7*(2), 74–81.

Bingham, E., Kaban, A., & Fortelius, M. (2007). The aspect Bernoulli model: multiple causes of presences and absences. *Pattern analysis & applications*, *12*(1), 55–78. https://doi.org/10.1007/s10044-007-0096-4

Breiman, L. (1996). Stacked Regressions. *Machine Learning*, *24*(1), 49–64. https://doi.org/10.1007/bf00117832

Chengsheng, T., Huacheng, L., & Bing, X. (2017). AdaBoost typical Algorithm and its application research. In *MATEC Web of Conferences* (Vol. 139). https://doi.org/10.1051/matecconf/201713900222

Dalal, M. K., & Zaveri, M. A. (2011). Automatic Text Classification: A Technical Review. *International Journal of Computer Applications*, *28*(2), 37–40.

Daniati, E. (2014). News Sentiment Analysis Using Naive Bayes And Adaboost. In *Information Technology and Security* (hal. 149–158). Malang: Lembaga Penelitian dan Pengabdian pada Masyarakat Sekolah Tinggi Informatika dan Komputer Indonesia.

Dataflair Team. (2018). Data Mining Tutorial – Introduction to Data Mining (Complete Guide). Diambil 18 Februari 2020, dari https://data-flair.training/blogs/data-mining-tutorial/

Doyle, D. (2000). Indonesian (Malay) Stopwords. Diambil 18

Januari 2020, dari https://www.ranks.nl/stopwords/indonesian

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook*. New York: Cambridge University Press.

Freund, Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. In *Machine Learning: Proceedings ofthe Thirteenth International Conference*.

Gaigole, P. C., Patil, L. H., & Chaudhari, P. M. (2013). Preprocessing Techniques in Text Categorization. In *National Conference on Innovative Paradigms in Engineering & Technology* (hal. 1–3). International Journal of Computer Applications.

Gupta, V., & Lehal, G. S. (2009). A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, *1*(1), 60–76. https://doi.org/10.4304/jetwi.1.1.60-76

Hidayat, R., & Minati, S. (2019). Comparative Analysis of Text Mining Classification Algorithms for English and Indonesian Qur'an Translation. *IJID International Journal on Informatics for Development*, *8*(1), 47–51.

Hidayatullah, A. F. (2015). The Influence of Stemming on Indonesian Tweet Sentiment Analysis. In *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2015)* (hal. 19–20). Palembang.

Hidayatullah, A. S. (2009). *INDEKS AL-QUR'AN DI INDONESIA (Study Komparatif Buku-Buku Indeks Al-Qur'an di Indonesia 1984-2007)*. UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH.

Johnson, T. M., & Grim, B. J. (2015). Religious Adherents. Diambil 18 Maret 2019, dari http://www.thearda.com/internationalData/

Jung, Y., Parky, H., & Du, D.-Z. (2007). A Balanced Term-weighting Scheme for Improved Document Comparison and Classiication.

Kontributor Wikipedia. (2019). Al-Qur'an. Diambil dari https://id.wikipedia.org/w/index.php?title=Al-Qur%27an&oldid=16360686

Kuhlman, D. (2012). *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. Platypus Global Media.

Leaman, O. (2006). *The Qur'an: an Encyclopedia*. Oxon: Routledge.

Man Lan, Sam-Yuan Sung, Hwee-Boon Low, & Chew-Lim Tan. (2005). A comparative study on term weighting schemes for text categorization. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (Vol. 1, hal. 546–551 vol. 1). https://doi.org/10.1109/IJCNN.2005.1555890

Manning, C. D., Raghavan, P., & Schütze, H. (2009).

Introduction to Information Retrieval. In *Introduction to Information Retrieval* (hal. 19–47). New York: Cambridge University Press.

Mazyad, A., Teytaud, F., & Fonlupt, C. (2018). Information Gain Based Term Weighting Method for Multi-label Text Classification Task. In *Intelligent Systems Conference (IntelliSys) 2018*. London, United Kingdom.

Nanas, N., Uren, V., & De Roeck, A. (2004). A comparative evaluation of term weighting methods for information filtering. In *Proceedings. 15th International Workshop on Database and Expert Systems Applications, 2004.* (hal. 13–17).

Nurjannah, M., Hamdani, & Astuti, I. F. (2013). Penerapan Algoritma Term Frequency-Inverse Document Frequency (Tf-Idf) untuk Text Mining. *Jurnal Informatika Mulawarman*, *8*(3), 110–113.

Opitz, D., & Maclin, R. (1999). Popular ensemble learning: an empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Osman, M., Hilal, A., & Alhawarat, M. (2016). Fine-Grained Quran Dataset. *International Journal of Advanced Computer Science and Applications*, *6*(12), 308–313. https://doi.org/10.14569/ijacsa.2015.061241

Pedregosa, F., Grisel, O., Andreas, M., Weiss, R., Passos, A., & Brucher, M. (2011). Scikit-learn: Machine Learning in

Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Periyasamy, N., Thamilselvan, P., & Sathiaseelan, J. G. R. (2015). A Comparative Study of ANN, K- Means and Adaboost Algorithms for image classification. In *International Conference on Innovations in Information, Embedded and Communication Systems*. https://doi.org/10.1109/ICIIECS.2015.7193067

Pew Research Center. (2011). The Future of the Global Muslim Population. Diambil dari http://www.pewforum.org/The-Future-of-the-Global-Muslim-Population.aspx

Polikar, R. (2006). Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–44. https://doi.org/10.1109/MCAS.2006.1688199

Prasetio, R. T., & Pratiwi. (2015). Penerapan Teknik Bagging pada Algoritma Klasifikasi untuk Mengatasi Ketidakseimbangan Kelas Dataset Medis. *INFORMATIKA*, *II*(2), 395–403.

Rahim, L. (2019). *Implementasi Cosine Similarity dan Naive Bayes Classifier Untuk Mencari Dalil Pada Artikel Islami Berdasarkan Indeks Al-Qur'an*. Universitas Islam Negeri Sunan Kalijaga.

Rahman, M. I., Samsudin, N. A., Mustapha, A., & Abdullahi, A. (2018). Comparative Analysis for Topic Classification

in Juz Al-Baqarah. *Indonesian Journal of Electrical Engineering and Computer Science*, *12*(1), 406–411. https://doi.org/10.11591/ijeecs.v12.i1.pp406-411

Rahutomo, F., & Ririd, A. R. T. H. (2019). Evaluasi Daftar Stopword Bahasa Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *6*(Februari), 41–48. https://doi.org/10.25126/jtiik.201861226

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, *60*(5), 503–520. https://doi.org/10.1108/00220410410560582

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*, 1–39. https://doi.org/10.1007/s10462-009-9124-7

Sabbah, T., Selamat, A., Selamat, M. H., Al-Anzi, F., Herrera-Viedma, E., Krejcar, O., & Fujita, H. (2017). Modified Frequency-Based Term Weighting Schemes for Text Classification. *Applied Soft Computing*, *58*. https://doi.org/10.1016/j.asoc.2017.04.069

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine learning*, *5*, 197–227. https://doi.org/10.1002/nbm.1810

Schapire, R. E., & Freund, Y. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*,

55, 119–139. https://doi.org/10.1088/0034-4885/55/7/004

Singh, B. (2019). Evaluation Metrics for Machine Learning Models. Diambil 20 Februari 2020, dari https://heartbeat.fritz.ai/evaluation-metrics-for-machine-learning-models-d42138496366

Statistics Indonesia. (2010). Results of 2010 Population Census. Diambil dari https://sp2010.bps.go.id/

Supriyati, E., & Iqbal, M. (2018). Pengukuran Similarity Tema pada Juz 30 Al Qur'an Menggunakan Teks Klasifikasi. *Jurnal SIMETRIS*, *9*(1), 361–370.

Tala, F. Z. (2003). *A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*. *Institute for Logic, Language and Computation Universiteit van Amsterdam*. Netherlands. https://doi.org/10.22146/teknosains.26972

Tim Penyusun Bahasa Kamus Pusat. (2008). *Kamus Bahasa Indonesia*. Jakarta: Pusat Bahasa.

Wibisono, Y. (2008). Stop words untuk Bahasa Indonesia. Diambil 18 Januari 2020, dari https://yudiwbs.wordpress.com/2008/07/23/stop-words-untuk-bahasa-indonesia/

Wikipedia Contributors. (2020a). Artificial Intelligence. Diambil 18 Februari 2020, dari https://en.wikipedia.org/w/index.php?title=Artificial_intelligence&oldid=941699610

Wikipedia Contributors. (2020b). Machine Learning. Diambil 18 Februari 2020, dari https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=941663327

Wikipedia Contributors. (2020c). Multi-label Classification. Diambil 24 Februari 2020, dari https://en.wikipedia.org/w/index.php?title=Multi-label_classification&oldid=940543026

Wikipedia Contributors. (2020d). Text Mining. Diambil 18 Februari 2020, dari https://en.wikipedia.org/w/index.php?title=Text_mining&oldid=939761470

Witten, I. H. (2002). *Text Mining*. Hamilton.

Xia, T., & Chai, Y. (2011). An Improvement to TF-IDF: Term Distribution based Term Weight Algorithm. *Journal of Software*, *6*(3), 413–420. https://doi.org/10.4304/jsw.6.3.413-420

Zhou, Z. (2009). Ensemble Learning. In *Encyclopedia of Biometrics*. Boston: Springer. https://doi.org/https://doi.org/10.1007/978-0-387-73003-5