**EVALUASI *KMEANS CLUSTERING* PADA *PREPROCESSING* SISTEM TEMU KEMBALI INFORMASI**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV Politeknik Negeri Malang

**Oleh:**

**YUDHA PRADANA PUTRA NIM. 1641720044**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

**EVALUASI *KMEANS CLUSTERING* PADA *PREPROCESSING* SISTEM TEMU KEMBALI INFORMASI**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV Politeknik Negeri Malang

**Oleh:**

**YUDHA PRADANA PUTRA NIM. 1641720044**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

**HALAMAN PENGESAHAN**

**EVALUASI *KMEANS CLUSTERING* PADA *PREPROCESSING* SISTEM TEMU KEMBALI INFORMASI**

**Disusun oleh:**

**YUDHA PRADANA PUTRA NIM. 1641720044**

**Skripsi ini telah diuji dan disahkan pada tanggal 24 Juni 2019**

**Disetujui oleh:**

1. Penguji I : Dr.Eng. Cahya Rahmad, ST. M.Kom.

NIP. 19720202 200501 1 002 .......................

1. Penguji II : Drs. Rawansyah, M.Pd

NIP. 19590620 199403 1 001 .......................

3. Pembimbing I : Dr.Eng. Faisal Rahutomo, ST., M.Kom.

NIP. 197711162005011008 ........................

4. Pembimbing II : Yoppy Yunhasnawa, S.ST., M.Sc.

NIP. 198906212019031013 ........................

Mengetahui,

|  |  |  |
| --- | --- | --- |
| Ketua Jurusan  Teknologi Informasi |  | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, S.T., M.Cs.  NIP. 19711110 199903 1 002 |  | Imam Fahrur Rozi, ST., MT.  NIP. 19840610 200812 1 0004 |

**PERNYATAAN**

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 24 Juni 2019

Yudha Pradana Putra

# **ABSTRAK**

# **ABSTRACT**

# **KATA PENGANTAR**

Puji syukur kehadirat Allah SWT, atas limpahan Rahmat dan Karunia-Nya, sehingga penulis dapat merampungkan skripsi dengan judul “Evaluasi *Kmeans Clustering* pada *Preprocessing* Sistem Temu Kembali Informasi” untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar Sarjana pada Program Studi Teknik Informatika di Politeknik Negeri Malang.

Dalam penyusunan laporan ini, penulis menerima banyak bantuan serta bimbingan dari berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah memberikan petunjuk dan hidayahnya dalam penyusunan skripsi ini sehingga dapat berjalan dengan baik.
2. Kedua orang tua dan keluarga penulis yang telah banyak memberikan dukungan baik secara moral maupun material.
3. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi.
4. Bapak Imam Fahrur Rozi, ST., MT selaku ketua Program Studi Teknik Informatika.
5. Bapak Dr. Eng. Faisal Rahutomo, ST., M.Kom. dan Bapak Yoppy Yunhasnawa, S.ST., M.Sc. selaku dosen pembimbing skripsi, yang telah memberikan waktu, kesempatan, petunjuk dan bimbingan.
6. Dosen-dosen pengajar Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang yang telah memberikan bimbingan dan ilmunya.
7. Teman-teman yang selalu meluangkan waktunya untuk membantu dan mendukung penulis sehingga penyusunan skripsi ini dapat diselesaikan dengan baik,
8. Semua pihak yang tidak dapat disebutkan satu persatu, yang telah memberikan bantuan hingga penyusunan skripsi ini selesai

Penulis sadar bahwa hasil pengerjaan skripsi dan laporan ini masih jauh dari sempurna. Karenanya, segala kritik dan saran yang membangun sangat penulis harapkan.

Malang, 24 Juni 2019

Penulis

# **DAFTAR ISI**

[**ABSTRAK** iv](#_Toc39692067)

[**ABSTRACT** v](#_Toc39692068)

[**KATA PENGANTAR** vi](#_Toc39692069)

[**DAFTAR ISI** viii](#_Toc39692070)

[**DAFTAR GAMBAR** xi](#_Toc39692071)

[**DAFTAR TABEL** xiii](#_Toc39692072)

[**DAFTAR LAMPIRAN** xiv](#_Toc39692073)

[**1.** **BAB I** 1](#_Toc39692074)

[**PENDAHULUAN** 1](#_Toc39692075)

[1.1. Latar Belakang 1](#_Toc39692076)

[1.2. Rumusan Masalah 2](#_Toc39692077)

[1.3. Tujuan 2](#_Toc39692078)

[1.4. Batasan Masalah 3](#_Toc39692079)

[1.5. Sistematika Penulisan 3](#_Toc39692080)

[**2.** **BAB II** 6](#_Toc39692081)

[**LANDASAN TEORI** 6](#_Toc39692082)

[2.1. Penelitian Terdahulu 6](#_Toc39692083)

[2.2. Sistem Temu Kembali Informasi 7](#_Toc39692084)

[2.3. Berita 7](#_Toc39692085)

[2.4. Flask 8](#_Toc39692086)

[2.5. Clustering 8](#_Toc39692087)

[*2.6.* Text Preprocessing 9](#_Toc39692088)

[2.7. Term Frequency – Inverse Document Frequency (TF-IDF) 10](#_Toc39692089)

[*2.8.* *Doc2Vec* 11](#_Toc39692090)

[2.9. Algoritma K-Means++ 11](#_Toc39692091)

[2.10. K-Means Clustering 12](#_Toc39692092)

[2.11. Cosine Similarity 14](#_Toc39692093)

[**3.** **BAB III** 16](#_Toc39692094)

[**METODOLOGI PENELITIAN** 16](#_Toc39692095)

[3.1. Studi Literatur 16](#_Toc39692096)

[3.2. Metode Pengumpulan Data 16](#_Toc39692097)

[3.3. Metode Pengembangan Perangkat Lunak 17](#_Toc39692098)

[3.4. Metode Pengolahan Data 17](#_Toc39692099)

[3.5. Pengujian 18](#_Toc39692100)

[3.6. Kesimpulan 18](#_Toc39692101)

[**4.** **BAB IV** 19](#_Toc39692102)

[**ANALISA DAN PERANCANGAN** 19](#_Toc39692103)

[4.1. Analisis Kebutuhan Sistem 19](#_Toc39692104)

[4.1.1. Kebutuhan Perangkat Keras (*Hardware*) 19](#_Toc39692105)

[4.1.2. Kebutuhan Perangkat Lunak (Software) 19](#_Toc39692106)

[4.2. Perancangan Sistem 19](#_Toc39692107)

[4.2.1. Arsitektur Sistem 20](#_Toc39692108)

[4.2.2. Usecase Diagram 20](#_Toc39692109)

[4.2.3. Skenario Usecase 21](#_Toc39692110)

[4.2.3.1. Mengelola Data Berita 21](#_Toc39692111)

[4.2.3.2. Melihat Hasil Preprocessing Berita 23](#_Toc39692112)

[4.2.3.3. Melihat Hasil Pembobotan Kata TFIDF 24](#_Toc39692113)

[4.2.3.4. Melihat Hasil Pembobotan Kata Doc2vec 25](#_Toc39692114)

[4.2.3.5. Melihat Hasil Cluster 26](#_Toc39692115)

[4.2.3.6. Melihat Hasil Pencarian 28](#_Toc39692116)

[4.2.3.7. Melihat Hasil Pengujian 29](#_Toc39692117)

[4.2.4 Activity Diagram 30](#_Toc39692118)

[4.2.5 Entity Relationship Diagram 35](#_Toc39692119)

[4.2.6 Sequence Diagram 35](#_Toc39692120)

[**5.** **BAB V** 47](#_Toc39692121)

[**IMPLEMENTASI** 47](#_Toc39692122)

[5.1. Implementasi Basis Data (Database) 47](#_Toc39692123)

[5.2. Implementasi Proses Text Preprocessing 48](#_Toc39692124)

[5.3. Implementasi Training dan Testing 48](#_Toc39692125)

[5.4. Implementasi User Interface 51](#_Toc39692126)

[5.4.1. Tampilan Halaman Home 51](#_Toc39692127)

[5.4.2. Tampilan Halaman Labeling 52](#_Toc39692128)

[5.4.3. Tampilan Hasil Deteksi Wajah dan Landmark Point 52](#_Toc39692129)

[5.4.4. Tampilan Hasil Perhitungan Jarak 53](#_Toc39692130)

[5.4.5. Tampilan Halaman Testing 53](#_Toc39692131)

[5.4.6. Tampilan Hasil Klasifikasi 54](#_Toc39692132)

[5.4.7. Tampilan Hasil Perhitungan Jarak Fitur 55](#_Toc39692133)

[5.4.8. Tampilan Perhitungan Kernel dan Weight Data Test 55](#_Toc39692134)

[5.4.9. Perhitungan Klasifikasi f(x) 55](#_Toc39692135)

[5.4.10. Tampilan Perhitungan Normalisasi 56](#_Toc39692136)

[5.4.11. Tampilan Perhitungan Matriks Kernel 56](#_Toc39692137)

[5.4.12. Tampilan Perhitungan Matriks Hessian 57](#_Toc39692138)

[5.4.13. Tampilan Perhitungan Alpha 57](#_Toc39692139)

[5.4.14. Tampilan Perhitungan Matriks Kernel Data Training 58](#_Toc39692140)

[5.4.15. Tampilan Perhitungan Weight Data Training 58](#_Toc39692141)

[5.4.16. Tampilan Perhitungan Bias 59](#_Toc39692142)

[**6.** **BAB VI** 60](#_Toc39692143)

[**PENGUJIAN DAN PEMBAHASAN** 60](#_Toc39692144)

[6.1. Pengujian 60](#_Toc39692145)

[6.1.1. Pengujian Fungsional 60](#_Toc39692146)

[6.1.2. Pengujian Data Antropometri 61](#_Toc39692147)

[6.1.3. Pengujian Akurasi 63](#_Toc39692148)

[**7.** **BAB VII** 67](#_Toc39692149)

[**KESIMPULAN DAN SARAN** 67](#_Toc39692150)

[7.1. Kesimpulan 67](#_Toc39692151)

[7.2. Saran 67](#_Toc39692152)

[**DAFTAR PUSTAKA** 68](#_Toc39692153)

[**LAMPIRAN** 70](#_Toc39692154)

[**PROFIL PENULIS** 82](#_Toc39692155)

# **DAFTAR GAMBAR**

[Gambar 2.1 Landmark pada Wajah Berdasarkan Acuan Antropometri 8](#_Toc11133425)

[Gambar 2.2 Indeks Titik Wajah dari Dataset iBUG 9](#_Toc11133426)

[Gambar 2.3 Hyperplane pada Support Vector Machine 11](#_Toc11133427)

[Gambar 3.1 Waterfall Model 16](#_Toc11133434)

[Gambar 4.1 Work Breakdown Structure 20](#_Toc11133438)

[Gambar 4.2 Flowchart Utama Sistem 21](#_Toc11133439)

[Gambar 4.3 Flowchart Haar Cascade Classifier 22](#_Toc11133440)

[Gambar 4.4 Flowchart SVM 23](#_Toc11133441)

[Gambar 4.5 Flowchart Training SVM 24](#_Toc11133442)

[Gambar 4. 6 Flowchart Testing SVM 33](#_Toc11133443)

[Gambar 4.7 Mockup Halaman Home 35](#_Toc11133444)

[Gambar 4.8 Mockup Halaman Labeling 36](#_Toc11133445)

[Gambar 4.9 Mockup Halaman Hasil Labeling 36](#_Toc11133446)

[Gambar 4.10 Mockup Halaman Testing 37](#_Toc11133447)

[Gambar 4.11 Mockup Halaman Hasil Testing 38](#_Toc11133448)

[Gambar 5.1 Tabel Penyimpanan Data Training 39](#_Toc12950678)

[Gambar 5.2 Source Code Lokalisasi Titik Wajah 39](#_Toc12950679)

[Gambar 5.3 Source Code Pengambilan Jarak Fitur 40](#_Toc12950680)

[Gambar 5.4 Source Code Proses Normalisasi 40](#_Toc12950681)

[Gambar 5.5 Source Code Proses Aktivasi Kernel 41](#_Toc12950682)

[Gambar 5.6 Source Code Proses Perhitungan Matriks Hessian 41](#_Toc12950683)

[Gambar 5.7 Source Code Proses Perhitungan Error dan Alpha 42](#_Toc12950684)

[Gambar 5.8 Source Code Proses Klasifikasi 42](#_Toc12950685)

[Gambar 5.9 Halaman Home 43](#_Toc12950686)

[Gambar 5.10 Tampilan Data Training 43](#_Toc12950687)

[Gambar 5.11 Halaman Labeling 44](#_Toc12950688)

[Gambar 5.12 Halaman Hasil Labeling 44](#_Toc12950689)

[Gambar 5.13 Tampilan Hasil Ekstraksi Fitur 45](#_Toc12950690)

[Gambar 5.14 Halaman Tetsing 45](#_Toc12950691)

[Gambar 5.15 Halaman Hasil Testing 46](#_Toc12950692)

[Gambar 5. 16 Hasil Ekstraksi Fitur 46](#_Toc12950693)

[Gambar 5.17 Perhitungan Kernal Dataset 47](#_Toc12950694)

[Gambar 5.18 Hasil Perhitungan Klasifikasi 47](#_Toc12950695)

[Gambar 5.19 Tampilan Perhitungan Normalisasi 48](#_Toc12950696)

[Gambar 5.20 Tampilan Perhitungan Matriks Kernel 48](#_Toc12950697)

[Gambar 5.21 Tampilan Perhitungan Matriks Hessian 49](#_Toc12950698)

[Gambar 5.22 Tampilan Perhitungan Alpha 49](#_Toc12950699)

[Gambar 5.23 Tampilan Perhitungan Matriks Kernel terhadap Data x 50](#_Toc12950700)

[Gambar 5.24 Tampilan Perhitungan Nilai Bobot 50](#_Toc12950701)

[Gambar 5.25 Tampilan Nilai Bias 51](#_Toc12950702)

# **DAFTAR TABEL**

[Tabel 3.1 Tabel Fitur 15](#_Toc11133179)

[Tabel 4.1 Hasil Ekstraksi Dataset 25](#_Toc11133209)

[Tabel 4.2 Hasil Normalisasi Dataset 25](#_Toc11133210)

[Tabel 4.3 Perhitungan Kernal Gaussian RBF 26](#_Toc11133211)

[Tabel 4.4 Hasil Perhitungan Matriks Hessian 27](#_Toc11133212)

[Tabel 4.5 Perhitungan Nilai Error Iterasi 0 28](#_Toc11133213)

[Tabel 4.6 Perhitungan Nilai Alfa pada Error Iterasi 0 28](#_Toc11133214)

[Tabel 4.7 Perhitungan Nilai Error Iterasi 1 29](#_Toc11133215)

[Tabel 4.8 Perhitungan Nilai Alfa pada Error Iterasi 1 29](#_Toc11133216)

[Tabel 4.9 Perhitungan Nilai Error Iterasi 2 29](#_Toc11133217)

[Tabel 4.10 Perhitungan Nilai Alfa pada Error Iterasi 2 30](#_Toc11133218)

[Tabel 4.11 Perhitungan αi 30](#_Toc11133219)

[Tabel 4.12 Perhitungan Kernel Data Positif dan Negatif 31](#_Toc11133220)

[Tabel 4.13 Perhitungan Nilai Bobot 31](#_Toc11133221)

[Tabel 4.14 Data Test 33](#_Toc11133222)

[Tabel 4.16 Normalisasi Data Tes 34](#_Toc11133223)

[Tabel 4.17 Perhitungan Kernel Data Tes terhadap Data Positif dan Negatif 34](#_Toc11133224)

[Tabel 4.18 Perhitungan Bobot 34](#_Toc11133225)

[Tabel 6.1 Hasil Pengujian Fungsional 52](#_Toc11133226)

[Tabel 6.2 Pengujian Antropometri Kelas Negatif (Laki-laki) 54](#_Toc11133227)

[Tabel 6.3 Hasil Pengujian Antropometri pada Kelas Positif (Female) 54](#_Toc11133228)

[Tabel 6.4 Hasil Pengujian Klasifikasi 55](#_Toc11133229)

[Tabel 6.5 Confusion Matriks 57](#_Toc11133230)

[Tabel 6.6 Hasil Akurasi Klasifikasi terhadap Jenis Data Training 58](#_Toc11133231)

# **DAFTAR LAMPIRAN**

[Lampiran 1. Hasil Pengujian Akurasi Data Training Jenis A](#_Toc11133138)

[Lampiran 2. Hasil Pengujian Akurasi Data Training Jenis B](#_Toc11133139)

[Lampiran 3. Lembar Bimbingan Skripsi Pembimbing I](#_Toc11133140)

[Lampiran 4. Lembar Bimbingan Skripsi Pembimbing II](#_Toc11133141)

[Lampiran 5. Lembar Persetujuan Maju Ujian Skripsi](#_Toc11133142)

[Lampiran 6. Lembar Revisi Penguji I](#_Toc11133143)

[Lampiran 7. Lembar Revisi Penguji II](#_Toc11133144)

[Lampiran 8. Lembar Verifikasi Abstrak dan Tata Tulis](#_Toc11133145)

# **BAB I**

# **PENDAHULUAN**

## Latar Belakang

Perkembangan teknologi informasi selalu mengalami perkembangan yang sangat pesat pada setiap tahunnya. Banyak fasilitas atau kemudahan-kemudahan yang diperoleh dari perkembangan teknologi informasi yang secara langsung berdampak kepada semua kalangan masyarakat. Saat ini, masyarakat dapat mencari suatu informasi yang tersedia di internet melalui media online. Salah satu media online yang digunakan masyarakat dalam mencari informasi secara akurat, cepat dan efisien adalah berita online.

Berita online adalah berita yang tersaji dan dapat diakses melalui media internet. Berita online bertindak sebagai perantara atas suatu peristiwa yang terjadi dengan masyarakat umum. Berita online saat ini sering diakses oleh masyarakat dimana beberapa penyedia berita online juga melampirkan berita melalui media sosial berbentuk link, sehingga itu dapat memudahkan masyarakat dalam mengaksesnya. Selain itu masyarakat juga bisa langung mengakses laman web masing-masing penyedia portal berita online. Seiring dengan banyaknya berita yang diterbitkan di internet, diperlukan suatu sistem yang bisa menemukan berita berdasarkan query tertentu dengan relevan dan cepat. Sistem temu kembali dapat dimanfaatkan untuk permasalahan tersebut.

Sistem temu kembali adalah sebuah sistem yang digunakan untuk menemukan kembali (retrieve) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Namun terkadang muncul permasalahan pada efisiensi sewaktu sistem memproses data yang sangat besar. Kurang efisiensi tersebut dikarenakan waktu tunggu sistem yang menjadi lebih lama karena diperlukan waktu untuk menghitung tingkat kemiripan query dengan dokumen.

Berdasarkan pemaparan diatas, penelitian yang akan dilakukan adalah membangun sistem temu kembali yang secara otomatis dapat menemukan berita dengan relevan dan cepat berdasarkan query yang diinputkan dengan terlebih dahulu menerapkan metode clustering untuk mengelompokan dokumen. Caranya dengan mengelompokan dokumen-dokumen ke dalam cluster berdasarkan kedekatan antar dokumen. Tujuan dari pengelompokan dokumen terlebih dahulu untuk mengurangi jumlah pencocokan query dengan dokumen. Metode clustering yang digunakan adalah metode k-means clustering untuk proses klastering berita pada tahap preprocessing. Setelah itu dilakukan pencocokan query dengan dokumen yang disederhanakan menjadi pencocokan query dengan centroid klaster. Untuk data yang akan digunakan untuk penelitian ini adalah berita yang berasal dari *Indonesian news corpus* yang telah dibuat pada penelitian sebelumnya.

## Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode K-means pada preprocessing sistem temu kembali informasi?
2. Bagaimana nilai performa untuk perbandingan dari sistem temu kembali informasi yang mengimplementasikan metode K-means dan TF-IDF menggunakan precision, recall dan f-score?
3. Bagaimana nilai performa untuk perbandingan dari sistem temu kembali informasi yang mengimplementasikan metode K-means dan Doc2Vec menggunakan precision, recall dan f-score?
4. Bagaimana nilai performa untuk perbandingan dari sistem temu kembali informasi tanpa metode klastering menggunakan precision, recall dan f-score?
5. Bagaimana perbandingan nilai performa dari sistem temu kembali informasi yang mengimplementasikan metode Single Pass Clustering dan TF-IDF (Faisal Rahutomo, Dwi Puspitasari, Tri Endah Sulistyoningrum, 2019) dengan metode yang diterapkan pada point-point sebelumnya?

## Tujuan

Tujuan dari pembuatan sistem ini adalah sebagai berikut :

Mengetahui cara implementasi metode K-means pada preprocessing sistem temu kembali informasi.

Mengetahui nilai performa untuk perbandingan dari sistem temu kembali informasi yang mengimplementasikan metode K-means dan TF-IDF menggunakan precision, recall dan f-score.

Mengetahui nilai performa untuk perbandingan dari sistem temu kembali informasi yang mengimplementasikan metode K-means dan Doc2Vec menggunakan precision, recall dan f-score.

Mengetahui nilai performa untuk perbandingan dari sistem temu kembali informasi tanpa metode klastering menggunakan precision, recall dan f-score.

Mengetahui dan membandingkan nilai performa dari sistem temu kembali informasi yang mengimplementasikan metode Single Pass Clustering dan TF-IDF (Faisal Rahutomo, Dwi Puspitasari, Tri Endah Sulistyoningrum, 2019) dengan metode yang diterapkan pada point-point sebelumnya.

## Batasan Masalah

Agar skripsi penulis yang berjudul Evaluasi K-Means Clustering pada Preprocessing Sistem Temu Kembali Informasi dapat berjalan sesuai dengan rencana, maka penulis memberikan batasan-batasan masalah yaitu :

Metode yang digunakan untuk clustering berita adalah metode k-means.

Metode yang digunakan untuk menghitung kemiripan *query* dengan dokumen adalah metode cosine similarity.

Data yang akan diolah berasal dari *Indonesian News Corpus* yang digunakan pada penelitian sebelumnya yang diambil dalam rentang waktu Juli 2015 – Desember 2015, yang berkategorikan teknologi sejumlah 809 berita

Aplikasi berbasis website, Bahasa pemrograman yang digunakan pada aplikasi adalah Phyton.

## Sistematika Penulisan

Sistematika penulisan skripsi dengan judul “Penerapan Feature Based Pada Peringkasan Berita Dari Portal Berita Online” akan ditulis berdasarkan Panduan Penulisan Laporan Akhir 2020 (POLINEMA, 2019), dengan susunan sebagai berikut:

**BAB I : PENDAHULUAN**

Pendahuluan berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan. Bagian pendahuluan akan menjadi dasar penelitian yang dilakukan.

**BAB II : LANDASAN TEORI**

Berisikan teori-teori yang relevan yang melengkapi latar belakang. Sekaligus memberi review tentang pustaka yang telah dibaca selama masa pencarian solusi terhadap masalah yang diangkat dalam tugas akhir. Pustaka yang dijadikan rujukan atau referensi merupakan publikasi ilmiah. Publikasi ilmiah dapat berbentuk jurnal, prosiding, modul ajar, diktat dan lain-lain.

**BAB III : METODOLOGI PENELITIAN**

Terdiri dari langkah-langkah yang akan membimbing penulis memilih metode, teknik, prosedur apa yang tepat, dan tools apa yang akan digunakan sehingga setiap tahapan dapat dilakukan dengan tepat, termasuk desain dan perancangan sistem yang akan dibuat.

**BAB IV : ANALISIS DAN PERANCANGAN**

Pada bagian ini diuraikan dengan jelas sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan nonfungsional. Rancangan sistem meliputi rancangan model sistem, rancangan arsitektur sistem, rancangan proses, rancangan prosedural, rancangan data dan rancangan antarmuka pengguna

**BAB V : IMPLEMENTASI DAN PENGUJIAN**

Pada bagian ini, implementasi sistem dipaparkan secara detil sesuai rancangan dan komponen (tools) bahasa pemrograman yang dipakai. Implementasi rancangan proses dapat disertai dengan potongan kode pada proses yang dimaksud. Pengujian merupakan proses untuk menentukan apakah hasil dari tugas akhir sudah sesuai dengan kebutuhan sistem dan berjalan sesuai lingkungan yang diinginkan. Pengujian dapat berupa pengujian fungsional, penerimaan pengguna, ataupun performa sistem. Pada pengujian dipaparkan secara detil mengenai metode pengujian, tujuan pengujian, proses pengujian serta analisa hasil pengujian.

**BAB VI : HASIL DAN PEMBAHASAN**

Pembahasan merupakan argumentasi rasional dari penulis yang disusun secara sistematis berdasarkan fakta ilmiah yang diperoleh dari hasil pengujian. Pembahasan hasil pengujian dapat disajikan dalam bentuk uraian teoritik, baik secara kualitatif maupun kuantitatif. Secara ilmiah, hasil pengujian yang diperoleh dalam penelitian dapat berupa temuan baru atau perbaikan, penegasan atau penolakan interpretasi suatu fenomena ilmiah yang diteliti pada penelitian sebelumnya. Penyajian hasil penelitian dapat diperjelas dengan menggunakan tabel, kurva, grafik, foto atau bentuk lain dapat digunakan sesuai keperluan secara lengkap dan jelas.

**BAB VII : KESIMPULAN DAN SARAN**

Berisi uraian singkat dan jelas tentang hasil tugas akhir yang diperoleh sesuai dengan tujuan penelitian. Apabila diperlukan, saran dapat digunakan untuk menyampaikan hal-hal yang dapat diperbaiki, dikembangkan atau dijadikan penelitian lebih lanjut

# **BAB II**

# **LANDASAN TEORI**

## Penelitian Terdahulu

Pada Penelitian kasus sebelumnya telah dilakukan penelitian oleh (Faisal Rahutomo, Dwi Puspitasari, Tri Endah Sulistyoningrum, 2019) mengenai Implementasi Single Pass Clustering pada Preprocessing Temu Kembali Teks. Dalam penelitian tersebut, pada tahap preprocessing mengimplementasikan single pass clustering untuk mengklasterkan informasi yang ada terlebih dahulu serta melakukan pencocokan query dengan dokumen disederhanakan kepada pencocokan query dengan centroid cluster, mampu mencari berita dengan lebih cepat. Sedangkan pengujian efektifitas menggunakan nilai pengujian precision, recall dan f-score. Dari pengujian tersebut, didapatkan hasil jika proses pencarian paling tepat dilakukan pada cluster dengan nilai threshold 0,1.

Hasil tersebut didapatkan ketika pengujian dilakukan menggunakan *keyword* ‘iphone’, ‘transaksi online’, ‘4g lte’, ‘aplikasi whatsapp’, dan ‘virtual reality’. Dari kelima pengujian tersebut hasil terbaik didapatkan pada pengujian dengan *keyword* ‘4g lte’. Yang mana pada pengujian tersebut nilai pengujian *f-score* sebesar 0,732. Nilai tersebut adalah nilai pengujian *f-score* yang paling baik dari pengujian lainnya. Sedangkan untuk nilai pengujian *precision* sebesar 0,756 dan pengujian *recall* sebesar 0,708.

Selain itu pada penelitian sebelumnya juga telah dilakukan oleh (Sugiyanto, Bayu Surarso, Aris Sugiharto, 2014) mengenai Analisa Performa Metode Cosine dan Jacard pada Pengujian Kesamaan Dokumen. Pada penelitian ini, membangun sistem temu kembali informasi yang menggunakan algoritma *single pass clustering* untuk mengklasterkan dokumen dan mengujinya dengan pendekatan kemiripan dan kesamaan dokumen yang berbeda yaitu menggunakan metode *cosine similarity* dan *jaccard similarity*. Hasil dari penelitian tersebut, menunjukkan bahwa penggunaan pengukuran kemiripan abstrak, baik dengan cosine maupun jaccard secara rata-rata keduanya memiliki performa nilai yang tinggi, namun jika dibandingkan cosine similarity memiliki tingkat akurasi yang lebih baik dengan nilai 0,949808 sedangkan jaccard sebesar 0,949077.

Sedangkan pada penelitian lainnya, telah dilakukan oleh (Nyoman Gede Y, Made Sudarma, Wayan Gede A, 2018) mengenai Penerapan Metode Clustering Text Mining untuk Pengelompokan berita pada Unstructured Textual Data. Pada penelitian ini menggunakan algoritma K-Means untuk melakukan pengelompokan terhadap dokumen teks. Menurut penelitian ini, K-Means dinilai lebih cocok untuk melakukan pengelompokan berita karena data inputan yang akan diproses terbilang masih sederhana. Hasil ujicoba yang dilakukan menggunakan jumlah data yang berbeda yaitu 50, 100, 200, 300, 400 dan 500 data didapatkan hasil bahwa algoritma K-Means yang diterapkan untuk mengklasterkan berita mampu bekerja dan memberikan akurasi yang memuaskan, dengan rata-rata precision sebesar 70,76% sedangkan recall sebesar 70,86% sedangkan purity sebesar 0,76.

## Sistem Temu Kembali Informasi

Sistem Temu Kembali Informasi digunakan untuk menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Jadi secara konsep ada beberapa dokumen yang berisi informasi yang diorganisasikan ke dalam sebuah media penyimpanan untuk mempermudah ditemukannya kembali dokumen tersebut berdasarkan query yang dimasukkan oleh pengguna. Teknik pencarian pada sistem temu kembali informasi ini berbeda dengan sistem manajemen basis data, dimana pada sistem temu kembali informasi ini memliki dua bagian yaitu bagian pengindexan dan pencarian.

## Berita

Berita adalah suatu laporan informasi fakta terbaru dan penting mengenai peristiwa terkini yang disampaikan kepada masyarakat melalui berbagai media, seperti surat kabar, Televisi, radio, media online, dan berbagai media lainnya. Pada umumnya isi berita yang disampaikan kepada masyarakat adalah informasi terbaru terkini, atau fakta-fakta terbaru mengenai sesuatu yang terjadi di masa lalu juga dapat dijadikan berita. Selain menyampaikan informasi, tujuan lain dari berita adalah untuk mempengaruhi masyarakat terkait isu yang diangkat di dalam berita tersebut.

## Flask

Flask adalah kerangka kerja aplikasi web mikro yang ditulis dalam bahasa pemograman Python dan berdasarkan Werkzeug toolkit, berguna untuk menerima request (url) dan memberikan respon, dan template engine Jinja2, digunakan untuk menampilkan data dan menulis logika pada tampilannya, berlisensi BSD. Aplikasi yang menggunakan Flask framework antara lain adalah Pinterest, LinkedIn dan halaman web komunitas situs Flask itu sendiri. Flask disebut micro framework karena tidak membutuhkan alat-alat tertentu atau pustaka. Flask tidak memiliki database abstraction layer, validasi form, atau komponen lain dimana sudah ada pustaka pihak ketiga yang menyediakan fungsi umum. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam Flask itu sendiri. Ekstensi yang ada untuk object-relational mapper, validasi form, penanganan unggahan, berbagai teknologi otentikasi terbuka dan beberapa alat-lata yang terkait kerangka umum. Ekstensi diperbarui jauh lebih teratur daripada inti program Flask. Flask juga menyediakan libraries dan kumpulan kode yang bisa digunakan untuk membangun website tanpa perlu melakukan semuanya dari nol.

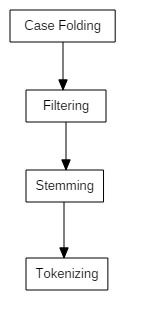
## Clustering

Clustering atau klasterisasi adalah metode pengelompokan data. Menurut Tan, 2006 clustering adalah sebuah proses untuk mengelompokan data ke dalam beberapa cluster atau kelompok sehingga data dalam satu cluster memiliki tingkat kemiripan yang maksimum dan data antar cluster memiliki kemiripan yang minimum. Clustering merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan cluster. Objek yang di dalam cluster memiliki kemiripan karakteristik antar satu sama lainnya dan berbeda dengan cluster yang lain. Partisi tidak dilakukan secara manual melainkan dengan suatu algoritma clustering. Oleh karena itu, clustering sangat berguna dan bisa menemukan group atau kelompokyang tidak dikenal dalam data. Clustering banyak digunakan dalam berbagai aplikasi seperti misalnya pada business inteligence, pengenalan pola citra, web search, bidang ilmu biologi, dan untuk keamanan (security).

Klastering dokumen dapat dilakukan sebelum atau sesudah proses temu kembali. Pada klastering dokumen yang dilakukan sebelum proses temu kembali, dokumen dikelompokkan ke dalam klaster berdasarkan kemiripan antar dokumen. Selanjutnya apabila suatu dokumen ditemukan maka seluruh dokumen yang berada dalam klaster yang sama dengan dokumen tersebut juga dapat ditemukan. (Jian Zhang, dkk., 2001). Salah satu tujuan klastering dokumen dalam sistem temu kembali informasi adalah mempercepat pemrosesan query dengan menelusuri hanya pada sejumlah kecil anggota atau wakil klaster, sehingga dapat mempercepat proses temu kembali informasi.

## Text Preprocessing

Sebelum melakukan analisa, preprocessing harus dilakukan terlebih dahulu untuk melakukan normalisasi kata-kata yang tidak diperlukan. Tahap ini bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan pada tahapan berikutnya. Pada proses text preprocessing meliputi, case folding, tokenizing, filtering dan stemming.



*Gambar 2.1 Text Preprocessing*

Case folding mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau lowercase) karena tidak semua dokumen konsisten dalam penggunaan huruf kapital. Tokenizing merupakan tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Tokenizing melakukan pemecahan sekumpulan karakter dalam suatu teks ke dalam satuan kata. Filtering adalah tahap mengambil kata-kata penting dari hasil token. Dapat menggunakan algoritma stoplist atau wordlist yakni kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Dan stemming adalah suatu proses yang mentransformasikan kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (root word). Pencarian kata dasar bahasa Indonesia dengan bahasa inggris berbeda, untuk bahasa Indonesia sendiri stemming menghilangkan imbuhan di awal, ditengah, diakhir ataupun imbuhan awal dan akhir kata.

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

Term Frequency – Inverse Document Frequency (TF-IDF) adalah metode untuk menghitung bobot pada setiap kata yang paling umum digunakan pada information retrieval. Metode ini terkenal efisien, mudah dan akurat. Metode ini akan menghitung nilai Term Frequency (TF) dan Inverse Document Frequency (IDF) pada setiap token (kata) di setiap dokumen dalam korpus. Secara sederhana, metode TF-IDF digunakan untuk mengetahui berapa sering suatu kata muncul di dalam dokumen. Metode ini pertama akan menghitung TF terlebih dahulu dengan rumus :

Dimana nilai **ft,d** adalah frekuensi **term (t)** pada **document (d)**. Jadi jika suatu kata atau term terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = 1 + log (5) = 1.699. Tetapi jika term tidak terdapat dalam dokumen tersebut, bobotnya adalah nol (0).

Kemudian menghitung IDF dimana sebuah perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Semakin sedikit jumlah dokumen yang mengandung term yang dimaksud, maka nilai IDF semakin besar. Dihitung menggunakan rumus :

IDFj = log(D/dfj)

Dimana **D** adalah jumlah semua dokumen dalam koleksi sedangkan **dfj** adalah jumlah dokumen yang mengandung **term (tj). Setelah itu menghitung Term Weighting TF-IDF untu menggabungkan formula perhitungan TF dengan formula IDF dengan cara :**

**Wij = tfij x idfj**

Dimana **Wij** adalah bobot **term (tj)** terhadap **dokumen (di).** Sedangkan**tfij**adalah jumlah kemunculan **term (tj)** dalam **dokumen (di). D** adalah adalah jumlah semua dokumen yang ada dalam database dan **dfj** adalah jumlah dokumen yang mengandung **term (tj)** (minimal ada satu kata yaitu **term (tj)**).

## *Doc2Vec*

Doc2Vec merupakan pengembangan dari implementasi metode *word embedding Word2Vec* yang bertujuan untuk mempresentasikan dokumen ke dalam bentuk vector. Dalam perkembangannya, Le dan Mikolov mengusulkan *Paragraph Vector*, teknik *unsupervised* yang mempelajari representasi vector kontinyu dan terdistribusi untuk potongan teks. Teks nya dimaksud disini panjangnya dapat bervariasi, mulai dari kalimat hingga dokumen. *Paragraph Vector* adalah untuk menekankan fakta bahwa metode ini dapat diterapkan pada potongan teks yang panjangnya bervariasi. Teknik ini dinilai cocok untuk merepresentasikan dokumen yang pada umumnya memiliki ukuran yang besar. Doc2Vec dapat melakukan ekstraksi fitur dengan menggunakan semua informasi atau kata yang ada pada dokumen, karena setiap kata yang ada pada dokumen digunakan untuk proses learning. Doc2Vec menghasilkan vector dokumen dan vector kata yang ada pada data training.

## Algoritma K-Means++

Algoritma k-means sering digunakan dalam teknik clustering untuk meminimalkan jarak kuadrat yang telah dirata-ratakan antar titik dalam cluster yang sama. Tetapi algoritma k-means memiliki kelemahan yaitu tidak bisa memberikan akurasi yang baik dan kurang tepat. Menurut Jurnal K-Means++ The Advantages of Carefull Seeding, jika k-means ditambahkan dengan metode randomized seeding technique dapat meningkatkan nilai akurasi pada algoritma k-means.

Akurasi algoritma k-means bergantung pada nilai titik centroid awalnya. Jika menggunakan nilai centroid yang berbeda, maka akan menghasilkan yang berbeda juga bahkan membutuhkan banyak iterasi untuk menentukan centroid selanjutnya. Dengan menambahkan rumus randomized seeding technique maka akan lebih menentukan nilai centroid awal. Berikut rumus dari randomized seeding technique :

Keterangan :

D(xi)2 = Jarak *Euclidean Distance*

∑x€X D(xi)2 = Jumlah Jarak *Euclidean Distance*

Rumus randomized seeding technique akan menghasilkan sebuah nilai yang akan dijadikan patokan semakin jauh nilai objek maka semakin besar kemungkinan nilai objek akan menjadi nilai centroid berikutnya.

## K-Means Clustering

*K-Means* merupakan suatu algoritma pengklasteran yang cukup sederhana yang mempartisi databest ke dalam beberapa clasteran *k.* Algoritma cukup mudah untuk diimplementasikan dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan (Wu & Kumar, 2009). Prinsip utama dari teknik ini adalah menyusun *K* buah partisi/pusat massa (*centroid*)/rata-rata (*mean*) dari sekumpulan data. *Algoritma K-Means* dimulai dengan pembentukan partisi klaster diawal kemudian secara iteraktif partisi *claster* ini diperbaiki hingga tidak terjadi perubahan yang signifikan pada partisi *claster* (Written, 2011). Pengelompokan yang dapat digunakan seperti pengelompokan non hierarki yang membagi data kedalam bentuk dua atau lebih kelompok. K-means merupakan metode analisis kelompok yang mengarah pada pembagian *N* objek pengamatan kedalam K kelompok (*cluster*) dan setiap objek pengamatan dimiliki oleh suatu kelompok dengan rata-rata (*mean*) terdekat (Prasetyo, 2012). Dasar algoritma K-means sebagai berikut :

Menentukan k sebagai jumlah cluster yang ingin dibentuk

Membangkitkan nilai random untuk pusat cluster awal (centroid)

Menghitung jarak setiap data input terhadap masing-masing centroid menggunakan rumus jarak Euclidean (Euclidean Distance) hingga ditemukan jarak yang paling dekat dari setiap data dengan centroid. Berikut persamaannya :

Dimana :

xi 🡪 data kriteria,

*μj* 🡪 centroid pada cluster ke-j

Mengelompokkan setiap data berdasarkan kedekatannya setiap data berdasarkan kedekatannya dengan centroid (jarak terkecil).

Memperbaharui nilai centroid. Nilai centroid baru diperoleh dari rata-rata cluster yang bersangkutan dengan menggunakan rumus :

Dimana :

*μj* (t+1) 🡪 centroid baru pada iterasi ke (t+1)

Nsj 🡪 banyak data pada cluster Sj

Melakukan perulangan dari langkah 2 hingga 5, sampai anggota tiap cluster tidak ada yang berubah.

Jika langkah 6 telah terpenuhi, maka nilai pusat cluster (*μj*) pada iterasi terakhir akan digunakan sebagai parameter untuk menentukan klasifikasi data.

## Cosine Similarity

*Cosine Similarity* adalah ukuran kesamaan antara dua buah vektor dalam sebuah ruang dimensi yang didapat dari nilai cosinus sudut dari perkalian dua buah vektor yang dibandingkan karena cosinus dari 00 adalah 1 dan kurang dari 1 untuk nilai sudut yang lain, maka nilai *similarity* dari dua buah vektor dikatakan mirip ketika nilai dari *cosine similarity* adalah 1.

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

Keterangan : A = vector

B = vector

Ai = bobot term i dalam blok Ai

Bi = bobot term i dalam blok Bi

i = jumlah term dalam kalimat

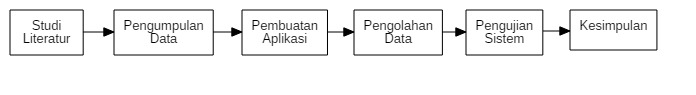
n = jumlah vector

Dimana *A* merupakan bobot setiap ciri pada *vector* A, dan B merupakan bobot setiap ciri pada B. jika dikaitkan dengan *information retrieval* maka A adalah bobot setiap istilah pada dokumen A, dan B merupakan bobot setiap istilah pada dokumen B. Pada penelitian ini digunakan *cosine similarity* karena citra merupakan salah satu data yang memiliki dimensi tinggi. Pada citra dapat dikatakan bahwa setiap *pixel* merupakan dimensi yang berbeda dan nilai warna pada setiap *pixel* tersebut merupakan nilai dari setiap dimensi. Pengukuran kemiripan dapat dilakukan dengan membandingkan dokumen 1 dengan dokumen 2 kemudian sistem akan menghitung nilai kemiripan. Ai.Bi adalah nilai yang diperoleh dari *term* A dan *term* B kemudian kedua nilai tersebut dijumlahkan, kemudian nilai Ai2 semua nilai *term* dokumen A semuan nilainya dipagkatkan dua, begitu juga (Widyaningtyas, Adiwijaya, & Faraby, 2018) dengan *term* Bi2 semua nilai yang diperoleh dipagkatkan dua kemudian semua nilai yang diperoleh dijumlahkan.

# **BAB III**

# **METODOLOGI PENELITIAN**

Dalam metode penelitian ini akan menjelaskan langkah-langkah yang dilakukan untuk merancang sistem evaluasi *kmeans clustering* pada *preprocessing* sistem temu kembal informasi adalah sebagai berikut:



## Studi Literatur

Mengumpulkan informasi mengenai data yang diperlukan untuk menganalisa sistem yang akan dibuat termasuk dataset. Selain itu juga mencari informasi dan referensi mengenai algoritma *kmeans clustering* dan *cosine similarity*. Studi literature ini dilakukan dengan cara menggunakan jurnal, internet serta buku untuk mendapatkan referensi penelitian-penelitian terdahulu mengenai metode terkait. Sumber yang diperoleh baik dari dalam maupun luar negeri.

## Metode Pengumpulan Data

Pada penelitian ini, data yang digunakan adalah berita yang berasal dari corpus. Corpus yang digunakan adalah Indonesian news corpus. Data yang diperoleh berasal dari penelitian sebelumnya yang dilakukan oleh Aad Miqdad Muadz Muzad dan Faisal Rahutomo, dalam tesis yang berjudul Korpus Berita Daring Bahasa Indonesia Dengan Depth First Focused Crawling. Corpus yang digunakan memiliki format xml dan json. Corpus berita tersebut bisa didapatkan melalui situs internet Mendeley data. Data berita online ini diambil dari beberapa situs portal berita online seperti kompas.com, republika.co.id, viva.co.id dan tribunnews.com. Data yang digunakan pada penelitian ini adalah data dari dengan kategori teknologi. Waktu berita diambil dalam rentang waktu bulan Juli hingga bulan Desember 2015. Selain itu, data yang akan diterapkan pada sistem ini bersifat dinamis, dimana pada penelitian ini menggunakan 800 dataset berita. Namun tidak menutup kemungkinan jumlah dataset tersebut bisa bertambah, berkurang atau dimodifikasi menggunakan fitur *CRUD (Create, Read, Update, Delete)* yang akan diterapkan pada sistem temu kembali informasi ini.

## Metode Pengembangan Perangkat Lunak

Metode yang digunakan dalam pengembangan keseluruhan system adalah menggunakan Metode Waterfall. Menurut Rizki Alfiasca Pascapraharastyan (2014: 140), Metode Waterfall adalah metode yang melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *requirements analysis*, *design*, *development*, dan *testing*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui pada metode ini harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement.* Pada tahap ini akan menganalisa kebutuhan sistem yang akan dibuat. Berikutnya tahap *design*, pada tahap ini dapat membantu dalam menentukan perangkat keras dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara kesuluruhan. Selanjutnya tahap *development,* sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit yang dikembangkan nantinya akan diuji pada tahap *testing*.

2

3

1

4

Gambar 7.1 Metode Waterfall

## Metode Pengolahan Data

Pada penelitian ini, diterapkan sebuah preprocessing untuk melakukan pengelompokkan berita. Hasil dari proses klastering ini digunakan untuk proses temu kembali informasi berdasarkan query yang diinputkan pengguna. Sebelum berita diklaster, dilakukan tahap *preprocessing* terlebih dahulu menggunakan proses *case folding*, *tokenization*, *filtering* dan *stemming*. Hasil tersebut kemudian akan dilakukan perhitungan untuk mendapatkan nilai vector dokumen menggunakan metode *TF–IDF* dan *Doc2Vec*. Tujuan dari tahap ini adalah untuk mendapatkan nilai vektor dokumen untuk digunakan pada tahap proses klaster menggunakan k-means clustering. Setelah itu, mendapatkan *vector centroid* untuk dilakukan perhitungan dengan *vector query* menggunakan metode *cosine similarity* untuk mencari kemiripan setiap cluster dengan query yang diinputkan.

## Pengujian

Pengujian dilakukan ketika sistem dijalankan untuk mengukur tingkat keberhasilan sistem dengan tujuan dibuatnya penelitian ini. Pengujian pada penelitian ini dilakukan menggunakan *precision, recall, f-score* dan waktu.

## Kesimpulan

Kesimpulan diambil berdasarkan hasil pengujian dari penelitian yang telah dilakukan.

# **BAB IV**

# **ANALISA DAN PERANCANGAN**

## Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis yang menjelaskan secara mendetail mengenai komponen-komponen yang dibutuhkan sistem untuk melakukan evaluasi *kmeans clustering* pada *preprocessing* sistem temu kembali informasi baik komponen perangkat keras maupun perangkat lunak dan detail-detail lainnya yang mendukung berjalannya sistem.

### Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras minimal yang dibutuhkan untuk merancang, mengimplementasikan dan mengoperasikan sistem temu kembali informasi yang menggunakan *kmeans clustering* pada *preprocessing* ini adalah laptop atau komputer dengan spesifikasi sebagai berikut:

1. *Processor* Intel(R) Core(TM) i5-6200U CPU 2.30 GHz
2. Memory RAM 8 GB
3. *Harddisk drive* dengan minimal *space* 500 GB
4. Intel(R) HD *Graphics* 4000

### Kebutuhan Perangkat Lunak (Software)

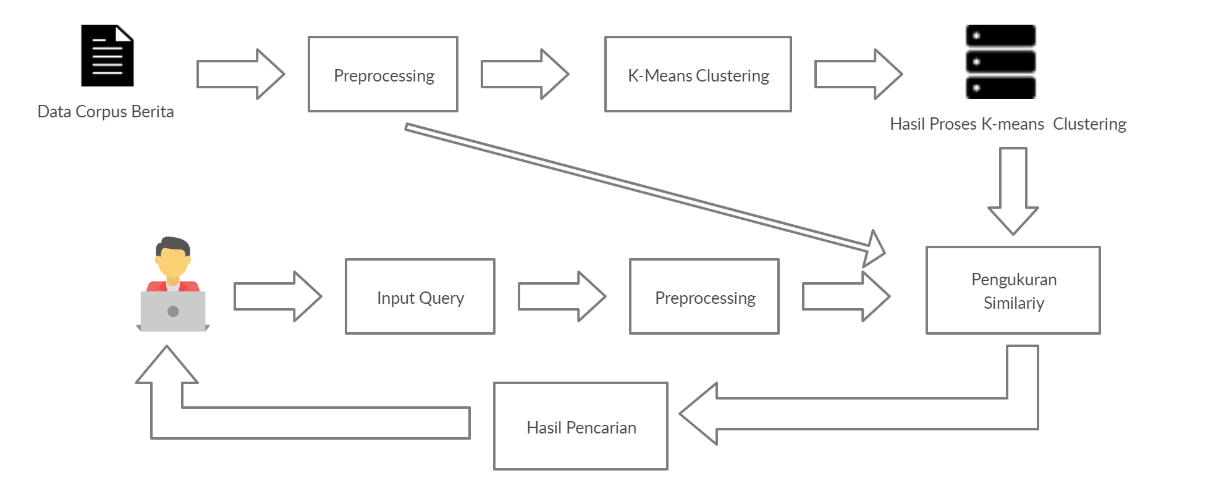
Untuk dapat merancang, menerapkan dan menjalankan sistem temu kembali informasi yang menggunakan *kmeans clustering* pada *preprocessing* ini secara tepat dan lancar, perlu disiapkan komponen perangkat lunak berikut:

1. Sistem Operasi Windows 10 64-bit
2. Pycharm
3. Web Browser Google Chrome
4. Python 3.7.7
5. *Package* *python* yang dibutuhkan: framework flask, math, datetime, os, csv, json, minidom, sastrawi, numpy, matplotlib, sklearn, gensim
6. MySQL, Web Server Apache, XAMPP

## Perancangan Sistem

Pada tahap ini, perancangan desain dari sistem yang ingin dibuat seperti apa. Pada tahap ini juga akan dijelaskan model sistem menggunakan beberapa diagram yang umum digunakan untuk pengembangan perangkat lunak. Beberapa diagram tersebut adalah sebagai berikut:

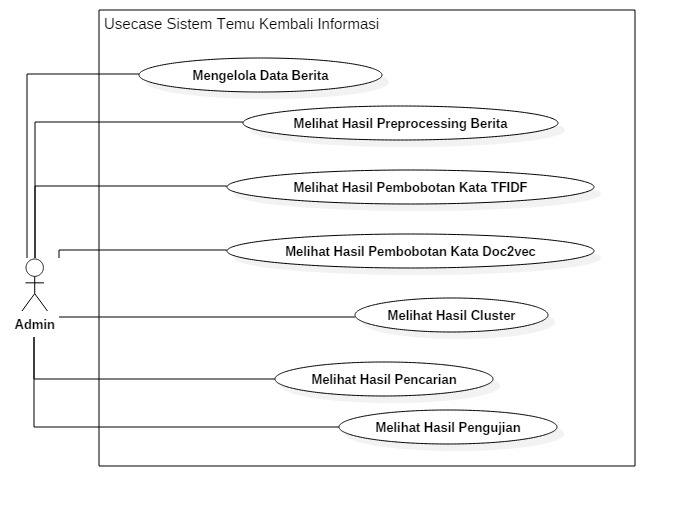
### Arsitektur Sistem

****

Gambar Arsitektur Sistem

Pada tahap perencanaan ini, peneliti menentukan arsitektur sistem dan perancangan antarmuka sistem yang akan dibuat. Pada penelitian ini diterapkan tahap preprocessing pada corpus yang berisi berita untuk diklasterkan terlebih dahulu menggunakan metode *kmeans clustering*. Sebelum diklastering, berita diproses terlebih dahulu pada *text processing* (*case folding*, *filtering*, *stemming*, *tokenizing*) dengan pembobotan kata menggunakan *TF – IDF* dan *Doc2vec*. Hasil dari proses klaster tersebut digunakan untuk proses sistem temu kembali informasi berdasarkan query yang diinputkan dan telah dilakukan tahap *preprocessing* sebelumnya. Terakhir dilakukan proses pengukuran similarity menggunakan metode cosine similarity antara *vector query* dan *vector centroid cluster*.

### Usecase Diagram



*Gambar Usecase Diagram*

Sesuai dengan usecase yang telah digambarkan, admin dapat mengelola data berita (*create, update, delete*), melihat hasil preprocessing berita (*case* *folding*, *filtering*, *stemming*, *tokenizing*), melihat hasil pembobotan kata TF-IDF dan Doc2vec, melihat hasil *cluster*, melihat hasil pencarian berita dan melihat hasil pengujian sistem.

### Skenario Usecase

Berikut adalah skenario usecase berdasarkan diagram usecase yang telah dibuat :

### Mengelola Data Berita

Berikut adalah skenario usecase dari usecase mengelola data berita :

|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Mengelola Data Berita | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat mengelola data berita | |
| Deskripsi | Admin melakukan fungsi insert, update, delete berita | |
| Pra-kondisi | Menampilkan halaman list data berita | |
| Pasca-kondisi | Data berita dimodifikasi dengan data terbaru | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Memilih menu news |  |  |
| 2. Memilih tombol *Add News* | - | - |
| 3. Menambahkan data berita pada sistem | 4. Menjalankan fungsi penyimpanan berita | 5. Sistem menyimpan data berita |
| 6. Memilih berita yang akan diubah | - | - |
| 7. Memilih ikon tombol *edit* | - | - |
| 8. Merubah data berita yang ada pada sistem | 9. Menjalankan fungsi *update* berita | 10. Sistem menyimpan perubahan data berita |
| 11. Memilih berita yang akan dihapus | - | - |
| 12. Memilih ikon tombol hapus | 13. Menjalankan fungsi hapus berita | 14. Sistem menghapus data berita |
| Kondisi Akhir | Sistem menampilkan semua data berita | |

### Melihat Hasil Preprocessing Berita

Berikut adalah skenario usecase dari usecase melihat hasil preprocessing berita :

|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Melihat Hasil *Preprocessing* Berita | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat melihat hasil *preprocessing* berita | |
| Deskripsi | Admin melakukan *preprocessing* berita | |
| Pra-kondisi | Menampilkan halaman beranda | |
| Pasca-kondisi | Menampilkan hasil *preprocessing* berita | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Memilih menu text preprocessing | 2. Menjalankan fungsi text preprocessing | 3. Mengambil data berita dari database |
| - | 4. Menjalankan proses case folding |  |
|  | 5. Menjalankan proses filtering |  |
|  | 6. Menjalankan fungsi stemming |  |
|  | 7. Menjalankan fungsi tokenizing | 8. menyimpan hasil tokenizing |
|  | 9. menampilkan hasil preprocessing berita | - |
| Kondisi Akhir | Sistem menampilkan hasil preprocessing berita | |

### Melihat Hasil Pembobotan Kata TFIDF

Berikut adalah skenario usecase dari usecase melihat hasil pembobotan kata tfidf :

|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Melihat Hasil Pembobotan Kata Tfidf | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat melihat hasil tfidf | |
| Deskripsi | Admin melakukan pembobotan kata menggunakan metode tfidf | |
| Pra-kondisi | Menampilkan halaman beranda | |
| Pasca-kondisi | Menampilkan hasil tfidf | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Memilih menu TF-IDF | 2. Menjalankan fungsi tfidf | 3. Mengambil data berita dari database |
| - | 4. Menjalankan proses TF |  |
|  | 5. Menjalankan proses idf | - |
|  | 6. Menjalankan proses tf-idf |  |
|  | 7. Menyimpan nilai tfidf |  |
|  | 8. Menampilkan hasil perhitungan tfidf |  |
| Kondisi Akhir | Sistem menampilkan hasil tfidf setiap berita | |

### Melihat Hasil Pembobotan Kata Doc2vec

Berikut adalah skenario usecase dari usecase melihat hasil pembobotan kata doc2vec :

|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Melihat Hasil Pembobotan Kata Doc2vec | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat melihat hasil doc2vec | |
| Deskripsi | Admin melakukan pembobotan kata menggunakan metode doc2vec | |
| Pra-kondisi | Menampilkan halaman beranda | |
| Pasca-kondisi | Menampilkan hasil doc2vec | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Memilih menu Doc2vec | 2. Menjalankan fungsi doc2vec | 3. Mengambil data berita dari database |
| - | 4. Menjalankan proses library doc2vec |  |
|  | 5. Menyimpan model doc2vec | - |
|  | 6. Menampilkan hasil doc2vec |  |
| Kondisi Akhir | Sistem menampilkan hasil doc2vec setiap berita | |

### Melihat Hasil Cluster

Berikut adalah skenario usecase dari usecase melihat hasil cluster :

|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Melihat Hasil Cluster | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat melihat hasil cluster | |
| Deskripsi | Admin melakukan pengelompokkan berita menggunakan kmeans | |
| Pra-kondisi | Menampilkan halaman menu clustering | |
| Pasca-kondisi | Menampilkan hasil cluster | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Memilih menu clustering | 2. Menampilkan halaman cluster |  |
| 3. Input jumlah cluster | 4. Menjalankan fungsi clustering |  |
|  | 5. Menyimpan nilai centroid cluster |  |
|  | 6. Menampilkan jumlah cluster yang terbentuk | - |
| 7. Memilih menu clustering news |  |  |
| 8. Memilih cluster yang ingin ditampilkan | 9. Menjalankan fungsi search | 10. Mengambil data berita terpilih sesuai cluster |
|  | 11. Menampilkan daftar berita dari cluster yang terpilih |  |
| Kondisi Akhir | Sistem menampilkan hasil tfidf setiap berita | |

### Melihat Hasil Pencarian

Berikut adalah skenario usecase dari usecase melihat hasil pencarian :

|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Melihat Hasil Pencarian | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat melihat hasil pencarian berita | |
| Deskripsi | Admin melakukan pencarian berita melalui query berdasarkan cluster yang telah terbentuk | |
| Pra-kondisi | Menampilkan halaman pencarian | |
| Pasca-kondisi | Menampilkan hasil pencarian berita | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Memilih menu information retrival | 2. Menampilkan halaman pencarian |  |
| 3. Menginputkan query | 4. Melakukan preprocessing query |  |
|  | 5. Menghitung pembobotan kata dari query |  |
|  | 6. Mengambil data centroid cluster |  |
|  | 7. Menghitung kemiripan vector query dan vector centroid | 8. Mengambil data berita dari cluster terpilih |
|  | 9. Menghitung pengujian sistem | 10. Menyimpan hasil pengujian |
|  | 11. Menampilkan berita berdasarkan cluster terpilih |  |
| Kondisi Akhir | Sistem menampilkan hasil pencarian berdasarkan cluster terpilih | |

### Melihat Hasil Pengujian

Berikut adalah skenario usecase dari usecase melihat hasil pengujian :

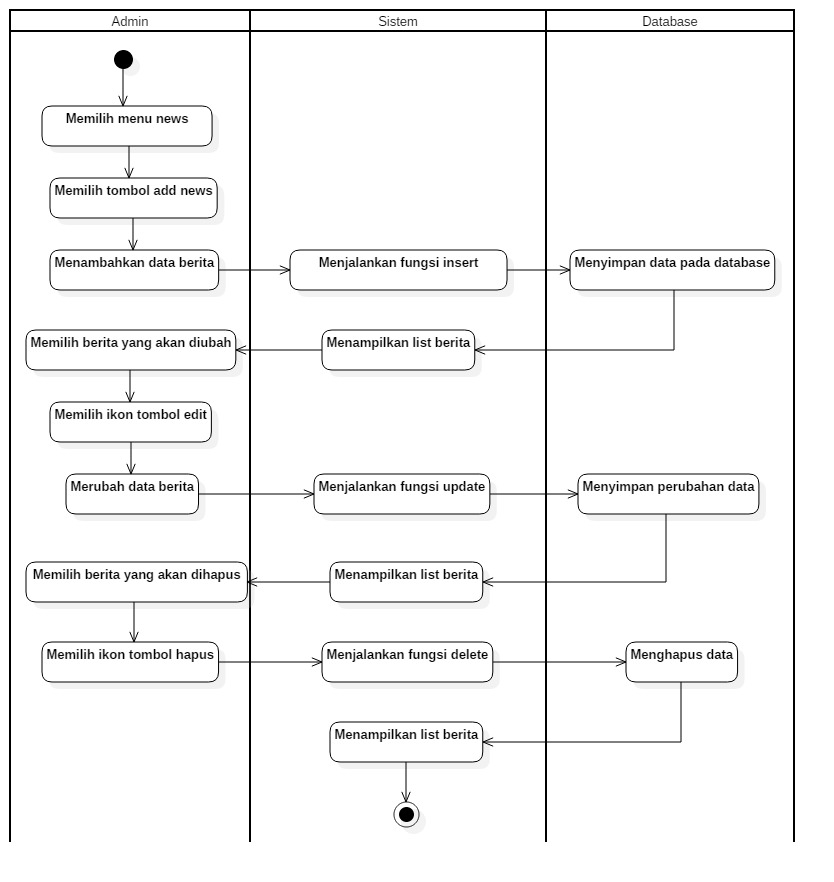
|  |  |  |
| --- | --- | --- |
| Nama *Use case* | Melihat Hasil Pengujian | |
| Aktor Utama | Admin | |
| Aktor Pendukung | - | |
| Tujuan | Admin dapat melihat hasil pengujian | |
| Deskripsi | Admin melihat hasil pengujian yang telah tersimpan di database | |
| Pra-kondisi | Menampilkan halaman beranda | |
| Pasca-kondisi | Menampilkan grafik pengujian | |
| Aliran Utama | | |
| Aksi dari Aktor | Tanggapan dari Sistem | Tanggapan dari Database |
| 1. Menekan menu testing | 2. Menjalankan fungsi testing | 3. Mengambil data testing dari database |
| - | 4. Menampilkan hasil testing |  |
| 5. Memilih menu dashboard | 6. Menampilkan data testing dalam bentuk grafik | - |
| Kondisi Akhir | Sistem menampilkan grafik testing | |

#### Activity Diagram

Sesuai dengan usecase telah dibuat desain rancangan tersebut dapat diidentifikasi lebih lanjut melalui activity diagram sebagai berikut :

##### **Mengelola Data Berita**

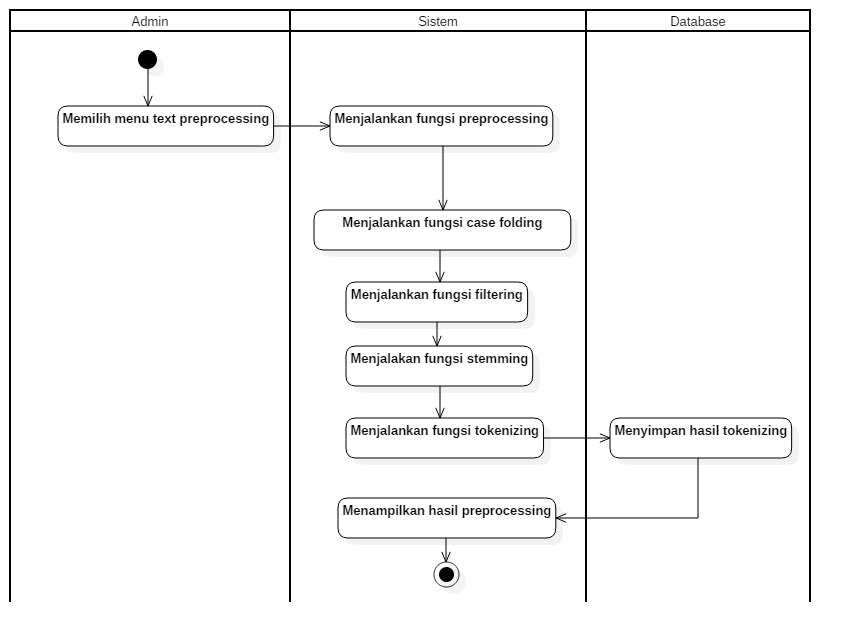
Berikut adalah activity diagram dari usecase mengelola data berita :



Pada activity diagram yang ditunjukkan oleh gambar, proses dimulai dari admin untuk memilih menu news terlebih dahulu untuk mengakses halaman data berita. Setelah itu admin dapat memilih *action* untuk menambahkan berita, mengubah berita atau menghapus berita. Setelah *action* berjalan akan otomatis tersimpan di database dan sistem akan kembali menampilkan data berita terbaru.

##### **Melihat Hasil Preprocessing Berita**

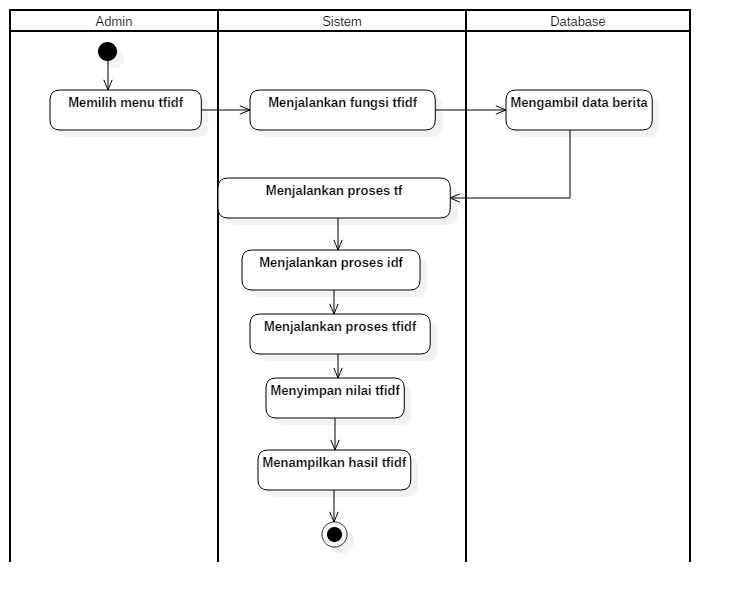
Berikut adalah activity diagram dari usecase melihat hasil preprocessing berita :



Pada activity diagram yang ditunjukkan oleh gambar, proses untuk melakukan preprocessing dimulai admin memilih menu text preprocessing. Kemudian proses case folding, filtering, stemming dan tokenizing akan berjalan dan hasilnya akan otomatis tersimpan di database.

##### **Melihat Hasil Pembobotan TF-IDF**

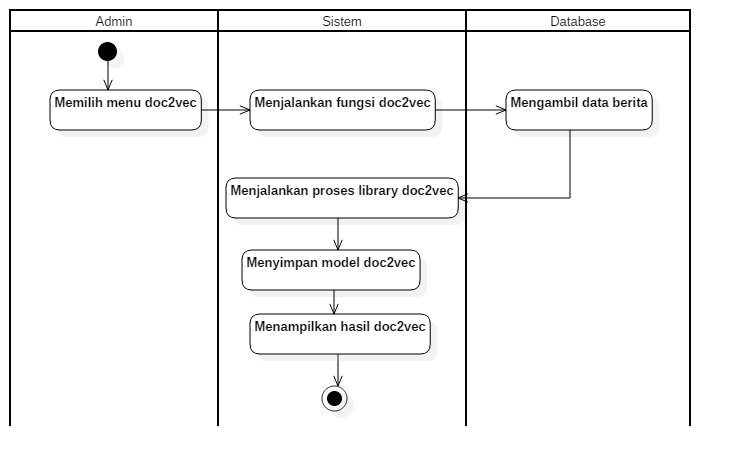
Berikut adalah activity diagram dari usecase melihat hasil pembobotan kata tfidf :



Pada activity diagram yang ditunjukkan oleh gambar, proses untuk tfidf dimulai dengan admin memilih menu tfidf. Kemudian sistem akan menjalankan fungsi tfidf dan mengambil data berita untuk perhitungan tfidf. Setelah itu hasil nilai tfidf akan disimpan oleh sistem dalam bentuk file json dan sistem akan menampilkan halaman data nilai tfidf setiap berita.

##### **Melihat Hasil Pembobotan Doc2vec**

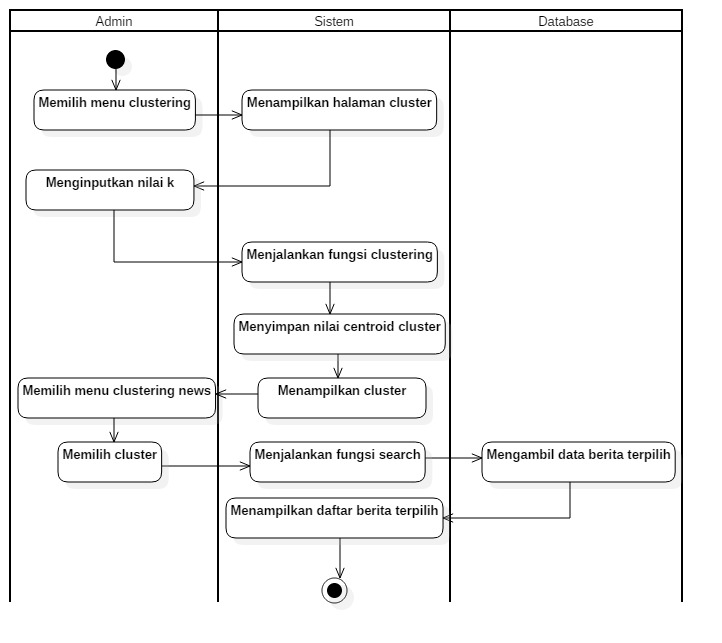
Berikut adalah activity diagram dari usecase melihat hasil pembobotan kata doc2vec :



Pada activity diagram yang ditunjukkan oleh gambar, proses untuk menjalakan fungsi doc2vec dimulai dengan admin memilih menu doc2vec sebelum sistem akan menjalankan library doc2vec. Kemudian hasil doc2vec akan disimpan oleh sistem dan sistem akan menampilkan halaman nilai doc2vec untuk setiap berita.

##### **Melihat Hasil Cluster**

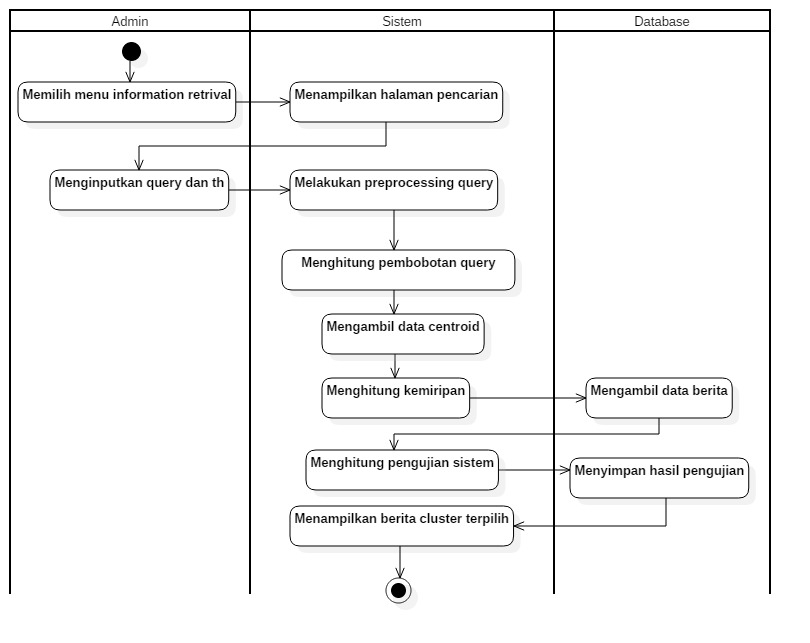
Berikut adalah activity diagram dari usecase melihat hasil cluster :



Pada activity diagram yang ditunjukkan oleh gambar, proses untuk *clustering* dimulai dengan admin memilih menu clustering terlebih dahulu. Kemudian admin menginputkan nilai k untuk menentukan jumlah cluster. Sistem akan menjalankan fungsi *clustering* dengan menyimpan nilai centroid. Untuk melihat hasil setiap clusternya admin memilih menu clustering news yang telah tersimpan dari hasil proses *clustering*.

##### **Melihat Hasil Pencarian**

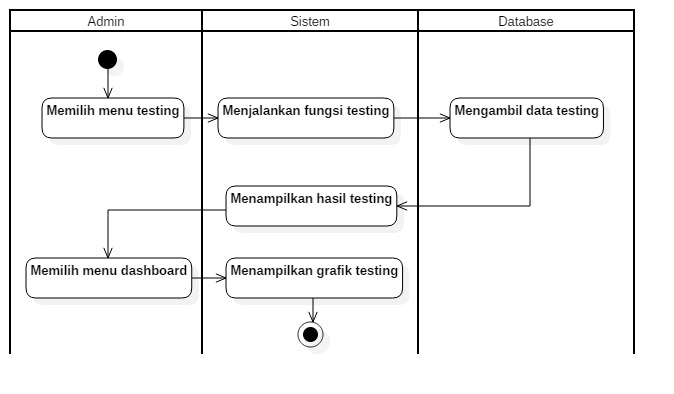
Berikut adalah activity diagram dari usecase melihat hasil pencarian :



Pada activity diagram yang ditunjukkan oleh gambar, proses untuk melakukan pencarian berita dengan menggunakan metode cosine similarity dimulai dengan admin memilih menu information retrival dan menginputkan query serta threshold. Kemudian sistem akan menjalankan preprocessing query serta menghitung kemiripan query dengan setiap cluster dan melakukan perhitungan untuk pengujiannya.

##### **Melihat Hasil Pengujian**

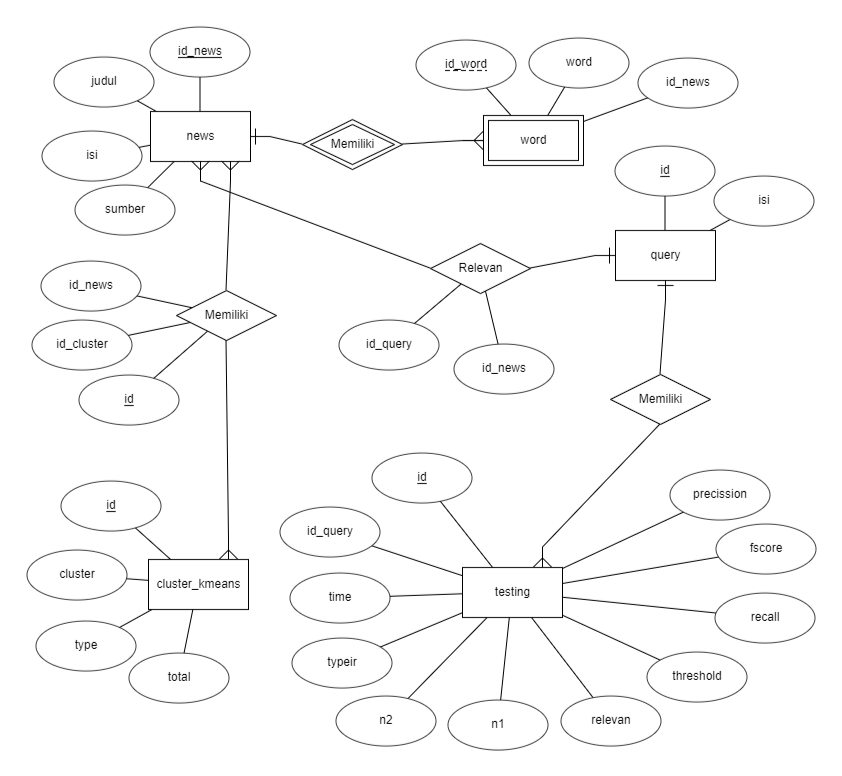
Berikut adalah activity diagram dari usecase melihat hasil pengujian :



Pada activity diagram yang ditunjukkan oleh gambar, proses untuk melihat hasil pengujian dimulai dengan admin memilih menu testing. Selain itu data testing juga ditampilkan pada menu dashboard yang disajikan dalam bentuk grafik.

#### Entity Relationship Diagram

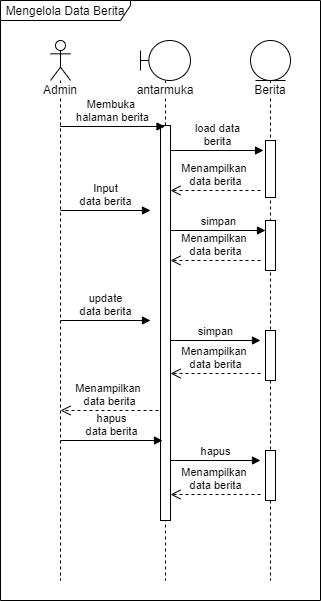
Berikut adalah pemodelan untuk menjelaskan hubungan antar data dalam database menggunakan Entity Relationship Diagram.



#### Sequence Diagram

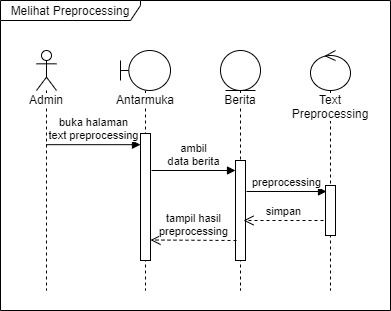
Berikut penggambaran interkasi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu dalam bentuk sequence diagram.

* + - 1. **Mengelola Data Berita**

****

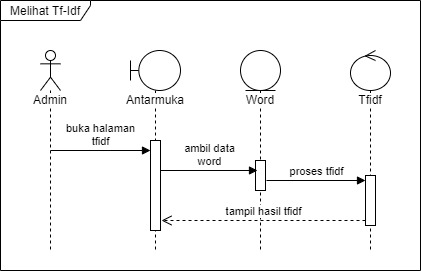
Pada gambar sequence diagram diatas, mengelola data berita dimulai dengan membuka halaman berita. Kemudian admin dapat melihat, menambahkan, mengubah dan menghapus berita.

* + - 1. **Melihat Hasil Preprocessing Berita**

****

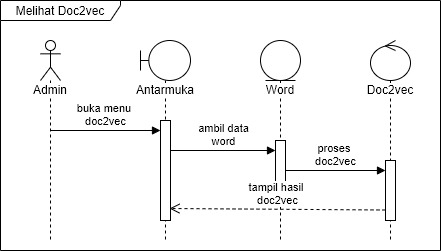
Pada gambar sequence diagram diatas, untuk melihat hasil text preprocessing admin membuka halaman text preprocessing terlebih dahulu. Setelah itu sistem akan mengambil data berita untuk dilakukan proses preprocessing dan menyimpan hasilnya serta ditampilkan hasilnya pada antarmuka.

* + - 1. **Melihat Hasil Pembobotan Tfidf**

****

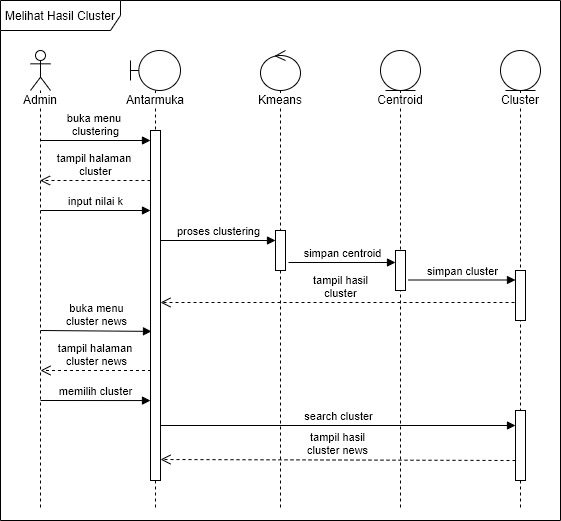
Pada gambar sequence diagram diatas, untuk melihat hasil pembobotan kata dengan metode tfidf diawali dengan admin membuka menu tfidf. Otomatis sistem akan mengambil data word atau hasil text preprocessing. Kemudian sistem akan menjalakan proses tfdidf dan menampilkannya pada antarmuka.

* + - 1. **Melihat Hasil Pembobotan Doc2vec**

****

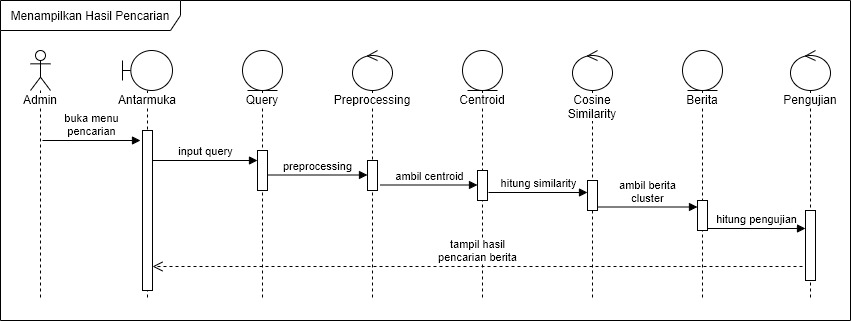
Pada gambar sequence diagram diatas, untuk melihat hasil doc2vec diawali dengan admin memilih menu doc2vec. Kemudian sistem akan otomatis mengambil data word atau hasil text preprocessing sebelumnya untuk diproses pada doc2vec. Setelah itu antarmuka akan menampilkan hasil doc2vec.

* + - 1. **Melihat Hasil Cluster**

****

Pada gambar sequence diagram diatas, untuk melihat hasil cluster diawali dengan admin memilih menu clustering dan menginputkan nilai k serta threshold. Setelah itu sistem akan melakukan proses clustering dan menyimpan nilai centoridnya pada sistem. Data cluster terbentuk akan ditampilkan oleh antarmuka. Untuk melihat berita setiap cluster, admin memilih menu clustering news dan memilih cluster yang ingin dilihat. Setelah itu antarmuka akan menampilkan berita dari cluster terpilih

* + - 1. **Melihat Hasil Pencarian**

****

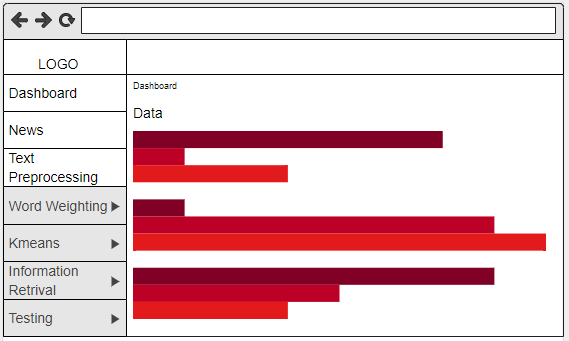
Pada gambar sequence diagram diatas, untuk melakukan pencarian berita pada sistem temu kembali informasi diawali dengan admin memilih menu information retrival dan menginputkan query dan threshold. Query akan dilakukan preprocessing terlebih dahulu sebelum dihitung similarity nya dengan cosine similarity. Kemudian hasil tersebut akan otomatis mengambil data berita dari cluster terpilih dan dilakukan perhitungan pengujian akurasi. Terakhir antarmuka akan menampilkan berita dari cluster teripilih.

* + - 1. **Melihat Hasil Pengujian**

Pada gambar sequence diagram diatas, untuk melihat hasil pengujian diawali admin memilih menu testing kemudian otomatis akan mengambil data testing dan antarmuka akan menampilkan hasilnya. Selain itu, admin juga bisa memilih menu dashboard untuk melihat data testing dalam bentuk grafik.

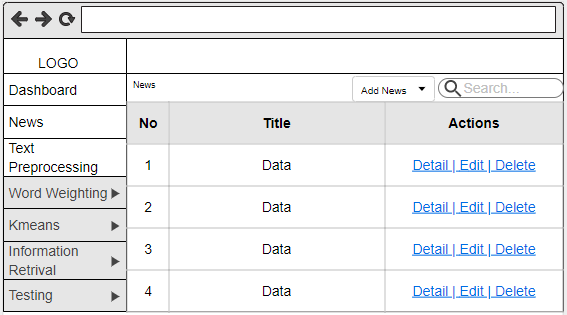
* + 1. **Mockup**
       1. **Tampilan Dashboard**

Berikut adalah rancangan desain untuk tampilan halaman dashboard. Halaman dashboard berfungsi untuk menampilkan statistic hasil pengujian yang disajikan dalam bentuk grafik.

****

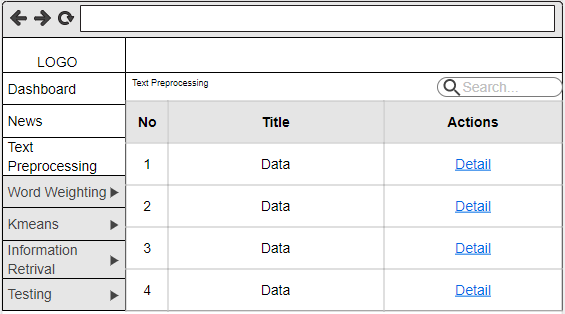
* + - 1. **Tampilan News**

Berikut adalah rancangan desain untuk tampilan halaman news. Halaman news berfungsi untuk menampilkan data berita yang tersimpan di database dan nantinya akan diolah untuk penelitian. Pada halaman ini admin juga bisa menambahkan, mengubah dan menghapus berita.



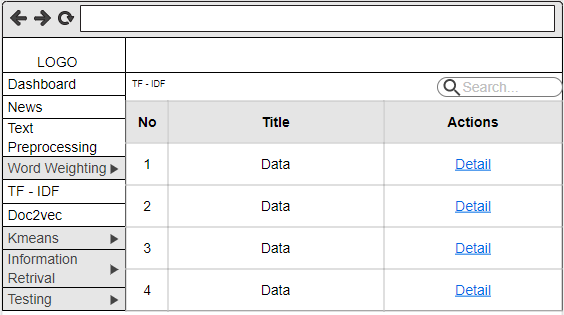
* + - 1. **Tampilan Text Preprocessing**

Berikut adalah rancangan desain untuk tampilan halaman text preprocessing. Halaman ini berfungsi untuk menampilkan hasil proses text preprocessing pada setiap berita yang telah dijalankan oleh sistem.



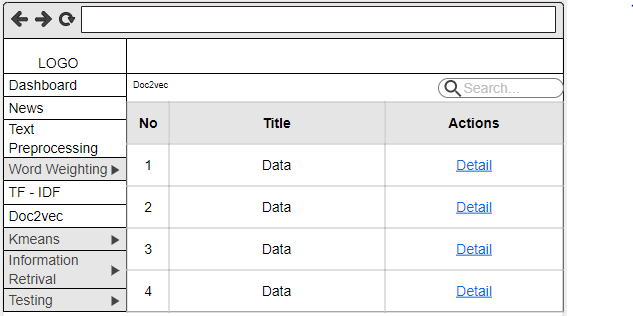
* + - 1. **Tampilan Tfidf**

Berikut adalah rancangan desain untuk tampilan halaman tfidf. Halaman ini berfungsi untuk menampilkan hasil pembobotan kata yang telah diproses menggunakan metode term frequency inverse document frequency pada setiap berita.



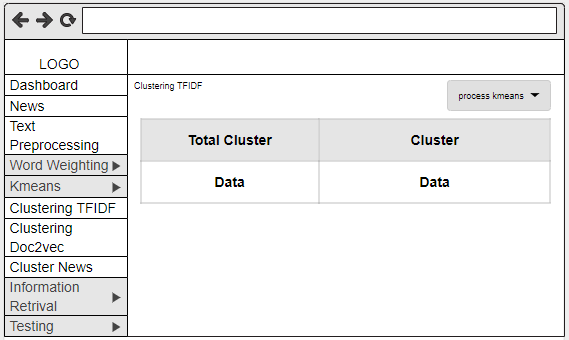
* + - 1. **Tampilan Doc2vec**

Berikut adalah rancangan desain untuk tampilan halaman doc2vec. Halaman ini berfungsi untuk menampilkan hasil pembobotan kata yang telah diproses menggunakan metode doc2vec pada setiap berita.

****

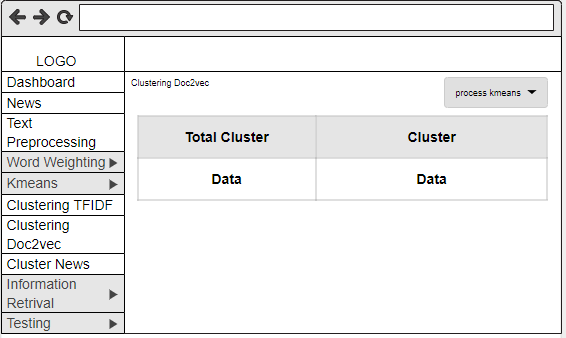
* + - 1. **Tampilan Clustering Tfidf**

Berikut adalah rancangan desain untuk tampilan halaman clustering tfidf. Halaman ini berfungsi untuk melakukan proses clustering menggunakan metode kmeans dengan pembobotan kata yang digunakan yaitu tfidf. Di halaman ini juga dapat melihat data cluster yang telah terbentuk dari hasil clustering.

****

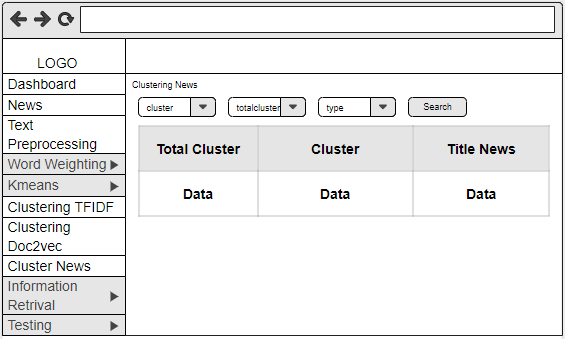
* + - 1. **Tampilan Clustering Doc2vec**

Berikut adalah rancangan desain untuk tampilan halaman clustering doc2vec. Halaman ini berfungsi untuk melakukan proses clustering menggunakan metode kmeans dengan pembobotan kata yang digunakan yaitu doc2vec. Di halaman ini juga dapat melihat data cluster yang telah terbentuk dari hasil clustering.

****

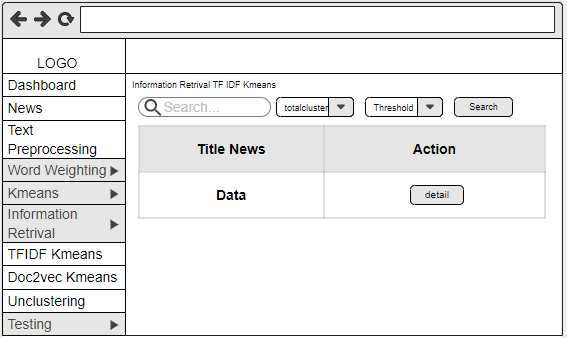
* + - 1. **Tampilan Cluster News**

Berikut adalah rancangan desain untuk tampilan halaman clustering news. Halaman ini berfungsi untuk menampilkan data cluster beserta berita yang termasuk dalam cluster tersebut.

****

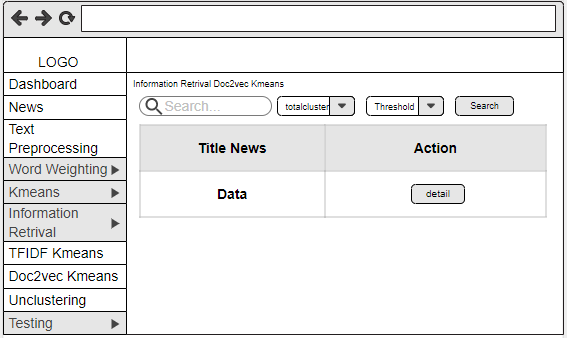
* + - 1. **Tampilan IR Kmeans-Tfidf**

Berikut adalah rancangan desain untuk tampilan halaman information retrival kmeans-tfidf. Halaman ini berfungsi untuk melakukan pencarian berita menggunakan query yang di inputkan oleh admin. Dalam proses pencarian ini menggunakan metode cosine similarity dimana data yang diproses adalah vektor query dan vektor centroid cluster yang sudah terbentuk. Kemudian sistem akan menampilkan berita yang sesuai dengan hasil similarity.

****

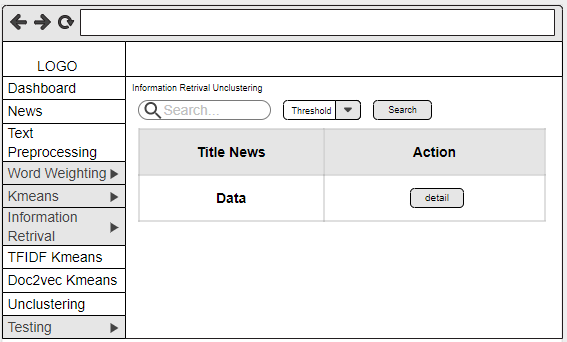
* + - 1. **Tampilan IR Kmenas-Doc2vec**

Berikut adalah rancangan desain untuk tampilan halaman information retrival kmeans-doc2vec. Halaman ini berfungsi untuk melakukan pencarian berita menggunakan query yang di inputkan oleh admin. Dalam proses pencarian ini menggunakan metode cosine similarity dimana data yang diproses adalah vektor query dan vektor centroid cluster yang sudah terbentuk. Kemudian sistem akan menampilkan berita yang sesuai dengan hasil similarity.



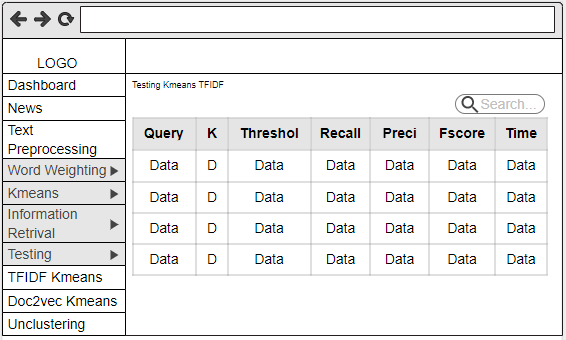
* + - 1. **Tampilan IR Unclustering**

Berikut adalah rancangan desain untuk tampilan halaman information retrival unclustering. Halaman ini berfungsi untuk melakukan pencarian berita menggunakan query yang di inputkan oleh admin. Dalam proses pencarian ini menggunakan metode cosine similarity dan tidak menggunakan metode kmeans. Kemudian sistem akan menampilkan berita yang sesuai dengan hasil similarity.

****

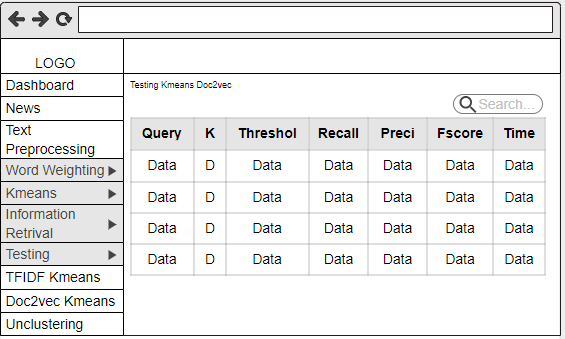
* + - 1. **Tampilan Testing Kmeans-Tfidf**

Berikut adalah rancangan desain untuk tampilan halaman testing kmeans-tfidf. Halaman ini berfungsi untuk melihat data pengujian pada sistem temu kembali informasi yang menggunakan kmeans-tfidf.



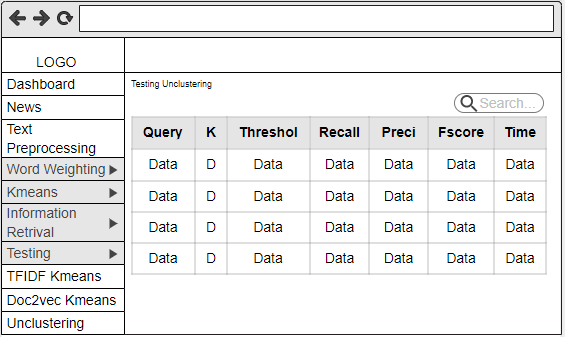
* + - 1. **Tampilan Testing Kmeans-Doc2vec**

Berikut adalah rancangan desain untuk tampilan halaman testing kmeans-tfidf. Halaman ini berfungsi untuk melihat data pengujian pada sistem temu kembali informasi yang menggunakan kmeans-doc2vec.



* + - 1. **Tampilan Testing Unclustering**

Berikut adalah rancangan desain untuk tampilan halaman testing kmeans-tfidf. Halaman ini berfungsi untuk melihat data pengujian pada sistem temu kembali informasi yang tidak menggunakan metode clustering.

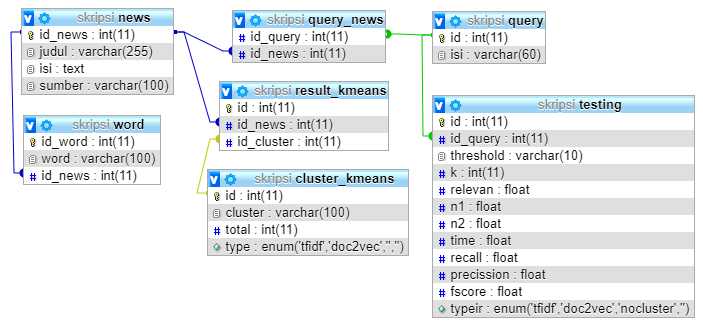


# **BAB V**

# **IMPLEMENTASI**

## Implementasi Basis Data (Database)

Implementasi database dari sistem temu kembali informasi yang menggunakan teknik klustering pada tahap preprocessingnya tertera pada gambar. Pada database ini memiliki 7 tabel yang saling berelasi dan memiliki fungsi sendiri setiap tabel. Tabel news memiliki 4 field yaitu id\_news, judul, isi dan sumber. Tabel ini berfungsi untuk menyimpan data berita yang menjadi data utama pada sistem temu kembali informasi dan dikelompokkan menggunakan teknik clustering. Tabel word memiliki 3 filed yaitu id\_word, word dan id\_news. Dimana id\_news pada tabel word memiliki relasi dengan id\_news yang ada pada tabel news. Tabel ini berfungsi menyimpan kata yang berasal dari berita yang telah dilakukan proses text preprocessing. Pada tabel cluster\_kmeans memiliki 4 field yaitu id, cluster, total dan type. Tabel ini berfungsi menyimpan data cluster yang telah terbentuk dari proses clustering. Untuk tabel result\_kmeans adalah tabel yang berfungsi untuk menyimpan hasil pengelompokkan berita pada setiap klusternya. Dimana memiliki pada fieldnya, memiliki 2 relasi tabel. Yang pertama id\_news pada result\_kmeans memiliki relasi dengan id\_news pada tabel news. Sedangkan yang kedua adalah id\_cluster memiliki relasi dengan id pada tabel cluster\_kmeans. Tabel query adalah tabel yang berfungsi menyimpan data query yang akan digunakan ujicoba pada sistem temu kembali informasi. Sedangkan tabel query\_news adalah tabel yang menyimpan data query yang relevan dengan berita atas hasil dari pencocokan secara manual yang digunakan pada penelitian sebelumnya. Dan tabel testing berfungsi untuk menyimpan hasil pengujian sistem.



Gambar 5.1. *Database*

## Implementasi Proses Text Preprocessing

|  |
| --- |
| sql = **"SELECT \* FROM news"** cursor.execute(sql) result = cursor.fetchall()  **for** row **in** result:  isi = str(row[2]) lowercase = isi.lower()translator = str.maketrans('', '', string.punctuation) delpunctuation = lowercase.replace(".", " ").translate(translator)stopwords = [line.rstrip() for line in open('uploads/stopword.txt')] cf = delpunctuation.split() stop = [a for a in cf if a not in stopwords] listToStr = ' '.join([str(elem) for elem in stop])st = stemming(listToStr)token = st.split() |

Gambar 5.2Source Code *Text Preprocessing*

Gambar 5.2 menjelaskan mengenai proses text preprocessing pada berita yang meliputi proses case folding, filtering, stemming dan tokenizing. Sebelum dilakukan proses text preprocessing, sistem terlebih dahulu mengambil data berita yang ada di database, kemudian dilakukan perulangan untuk dilakukan text preprocessing setiap berita. Dan hasilnya adalah berupa array yang meliputi *word* yang dimiliki masing-masing berita.

## Implementasi Pembobotan Kata

Dalam sistem temu kembali informasi yang menerapkan metode klustering pada preprocessing, memiliki dua metode pembobotan kata yaitu *Term Frequenct – Inverse Document Frequency* (TFIDF)dan Doc2vec.

|  |
| --- |
| **for** row **in** listberita:  id\_news = row[0]isi = row[1]  worddict = dict.fromkeys(wordset, 0)  **for** word **in** isi:  **if** word **in** worddict:  worddict[word] += 1  wd.append((id\_news, worddict)) |

Gambar 5.3Source CodeTerm Frequency

Gambar 5.3 menjelaskan mengenai proses term frequency, dimana menghitung setiap kemunculan kata yang muncul pada setiap berita. Setelah di proses nilainya akan disimpan pada variable array yaitu wd menggunakan fungsi append.

|  |
| --- |
| **def** computeIDF(docList):  N = len(docList)  idfDict = dict.fromkeys(docList[0].keys(), 0)  **for** doc **in** docList:  **for** word, val **in** doc.items():  **if** val > 0:  idfDict[word] += 1  **for** word, val **in** idfDict.items():  **if** val < 1:  idfDict[word] = 0  **else**:  idfDict[word] = (float(math.log10(N/float(val))))+1  **return** idfDict |

Gambar 5.4Source Code *IDF*

Gambar 5.4 menjelaskan mengenai proses menghitung IDF (*Inverse Document Frequency*). Fungsi ini melanjutkan proses dari perhitungan *Term Frequency* yaitu jumlah dokumen/berita dibagi dengan *Document Frequency* (DF) kemudian di logaritma. Kemudian datanya di kembalikan (*return*)dalam bentuk array.

|  |
| --- |
| **def** computeTFIDF(tfBow, idfs):  tfidf = {}  **for** word, val **in** tfBow.items():  tfidf[word] = val\*idfs[word]  **return** tfidf |

Gambar 5.5Source CodeTFIDF

Gambar 5.5 menjelaskan mengenai proses perhitungan TFIDF dimana hasil dari proses *Term Frequency* dikalikan dengan hasil *Inverse Document Frequency*. Kemudian nilainya disimpan pada variable array tfidf.

|  |
| --- |
| **for** row **in** listberita:  sentence = **""** word = row[1]  **for** row **in** word:  sentence+=row+**" "** data.append(str(sentence)) tagged\_data = [TaggedDocument(words=\_d.split(), tags=[str(i)]) **for** i, \_d **in** enumerate(data)] model = Doc2Vec(tagged\_data) model.save(**"dv2.model"**) |

Gambar 5.6Source CodeDoc2vec

Gambar 5.6 menjelaskan mengenai pembobotan kata menggunakan metode doc2vec. Dengan menggunakan library, sistem memproses dengan menginputkan parameter berupa kalimat yang telah dilakukan preprocessing. Dan hasilnya akan disimpan oleh sistem dengan attribute savedan disimpan dengan nama dv2.model.

## Implementasi Klustering

|  |
| --- |
| **with** open(**'tfidf.json'**, **'r'**) **as** openfile:json\_object = json.load(openfile)  **for** row **in** json\_object:  listval = []  **del** row[**'idnews'**]  **for** word, val **in** row.items():  listval.append(float(val))  matrix.append(listval) array = np.array(matrix)  kmeans = KMeans(n\_clusters=int(total), init=**'k-means++'**, random\_state=1) kmeans.fit(array) resultcluster = kmeans.labels\_ centroid = kmeans.cluster\_centers\_ |

Gambar 5.6Source CodeKmeans TFIDF

Gambar 5.6 menjelaskan mengenai metode kmeans clustering menggunakan library dari sklearn. Parameter yang dibutuhkan adalah nilai tfidf yang telah disimpan pada file json yang diproses pada tahap sebelumnya. Setelah berhasil membaca file json, nilai tfidf disimpan pada variable array untuk kemudian diproses pada fungsi Kmeans. Variable resultcluster digunakan untuk menyimpan hasil cluster, sedangkan variable centroid digunakan untuk menyimpan nilai centroid setiap cluster.

|  |
| --- |
| model = Doc2Vec.load(**"dv2.model"**)  N = len(model.docvecs) vecdoc = [] **for** row **in** range(N):  vecdoc.append(model.docvecs[row]) array = np.array(vecdoc) kmeans = KMeans(n\_clusters=int(total), init=**'k-means++'**, random\_state=1) kmeans.fit(array) resultcluster = kmeans.labels\_ centroidcluster = kmeans.cluster\_centers\_ |

Gambar 5.7Source CodeKmeans Doc2vec

Gambar 5.7 menjelaskan mengenai proses kmeans clustering menggunakan nilai doc2vec yang telah diproses pada tahap sebelumnya. Attribute load pada Doc2vec digunakan untuk memanggil nilai doc2vec yang telah tersimpan dan disimpan dalam variable array untuk menjadi parameter fungsi KMeans.

## Implementasi Sistem Temu Kembali Informasi

Pada penelitian ini, sistem temu kembali yang dikerjakan ada 3 macam yaitu sistem temu kembali informasi yang menggunakan metode kmeans dan tfidf, sistem temu kembali informasi yang menggunakan metode kmeans dan doc2vec dan sistem temu kembali informasi yang tidak menggunakan metode klustering.

|  |
| --- |
| lowercase = query.lower() translator = str.maketrans(**''**, **''**, string.punctuation) delpunctuation = lowercase.replace(**"."**, **" "**).translate(translator) stopwords = [line.rstrip() **for** line **in** open(**'uploads/stopword.txt'**)] cf = delpunctuation.split() stop = [a **for** a **in** cf **if** a **not in** stopwords] listToStr = **' '**.join([str(elem) **for** elem **in** stop]) st = stemming(listToStr) token = st.split() |

Gambar 5.8Source CodeText Preprocessing Query

Gambar 5.8 menjelaskan mengenai tahap awal dari proses sistem temu kembali informasi, dimana query yang diinputkan oleh admin dilakukan text preprocessing (*case folding, filtering, stemming, tokenizing*).

|  |
| --- |
| **with** open(**"centroidkmeans"**+total+**".json"**, **"r"**) **as** openfile:  *# Reading from json file* json\_object = json.load(openfile)  **for** row **in** json\_object:  cluster = row[**'idcluster'**]  **del** row[**'idcluster'**]  listcentroid.append((cluster, row)) **for** row **in** dataquery:  token = row[1]  worddict = dict.fromkeys(wordset, 0)  **for** rows **in** token:  **if** rows **in** worddict:  worddict[rows]+=1  listcentroid.append((**'q'**, worddict)) listworddict = [] **for** row **in** listcentroid:  worddict = row[1]  listworddict.append(worddict) idfword = computeIDF(listworddict)  **for** row **in** listcentroid:  wordtfidf = computeTFIDF(row[1], idfword)  **if** row[0] == **'q'**:  qtfidf.append((row[0], wordtfidf))  **if** j == (len(listcentroid)-1):  listcentroid.pop(j)  j+=1 |

Gambar 5.9Source CodeVektor Query Tfidf

Gambar 5.9 menjelaskan mengenai proses menjadikan query sebagai nilai tfidf. Pertama, query akan melakukan proses term frequency. Setelah itu dilakukan perhitungan IDF dan TFIDF bersama centroid cluster dikarenakan nantinya akan dilakukan perhitungan vektor query dengan vektor centroid cluster menggunakan metode cosine similarity untuk menghasilkan nilai tfidf untuk query tersebut.

|  |
| --- |
| model = Doc2Vec.load(**"dv2.model"**)  vecquery = model.infer\_vector(token, steps=20) |

Gambar 5.10Source CodeVektor Query Doc2vec

Gambar 5.10 menjelaskan mengenai proses menjadikan query sebagai nilai doc2vec, dimana model doc2vec yang telah disimpan tadi dipanggil dan dicari query vektornya berdasarkan model tersebut.

|  |
| --- |
| **def** computeCrossScalar(tfidf):  N = len(tfidf)  i = 0  Q = []  doc = []  **for** row **in** tfidf:  **if** i == (N-1):  Q.append(row[1])  **else**:  doc.append(row[1])  i+=1  result = []  **for** row **in** doc:  wdDict = dict.fromkeys(doc[0].keys(), 0)  **for** row2 **in** Q:  **for** word, val **in** row2.items():  wdDict[word]+=float(val)  **for** word, val **in** row.items():  wdDict[word]\*=float(val)  result.append(wdDict)  **return** result |

Gambar 5.11Source Code Perkalian Skalar

Gambar 5.11 menjelaskan mengenai perkalian skalar yang menjadi awal perhitungan metode cosine similarity untuk mencari kemiripan query dengan cluster. Perkalian dilakukan antara nilai vektor query dengan vektor centroid setiap cluster.

|  |
| --- |
| **def** longvector(tfidf):  result = []  **for** row **in** tfidf:  value = row[1]  wdDict = dict.fromkeys(row[1].keys(), 0)  **for** word, val **in** value.items():  wdDict[word] = math.pow(val, 2)  W = 0  **for** word, val **in** wdDict.items():  W+=float(val)  akar = math.sqrt(W)  result.append(akar)  **return** result |

Gambar 5.12Source Code Panjang Vektor

Gambar 5.12 menjelaskan mengenai menghitung panjang vektor dengan melakukan perkalian kuadrat setelah itu dijumlahkan setiap vektornya dan hasilnya dilakukan operasi akar kuadrat.

|  |
| --- |
| **def** computeSumScalar(computeScalar):  sum = []  **for** row **in** computeScalar:  W = 0  **for** word, val **in** row.items():  W+=float(val)  sum.append(W)  **return** sum |

Gambar 5.13Source Code Penjumlahan Hasil Skalar

Gambar 5.13 menjelaskan mengenai proses penjumlahan dari hasil perhitungan perkalian skalar pada tahap sebelumnya.

|  |
| --- |
| **for** row **in** joinsumscalardatalovector\_transpose:  sumScalar = row[0]  longvec = row[1]  C = sumScalar/(qlovector\*longvec)  cos.append((i, C))  c.append(C)  i+=1 |

Gambar 5.14Source Code Perhitungan Similarity

Gambar 5.14 menjelaskan mengenai tahap akhir perhitungan similarity menggunakan metode cosine similarity. Dimana nilai similarity didapatkan dari variable sumScalar sebagai total nilai jumlah perkalian skalar dibagi dengan hasil perkalian antara variable qlovector sebagai hasil perhitungan panjang vektor query dengan variable longvec sebagai hasil perhitungan panjang vektor centroid. Perhitungan ini dilakukan dengan perulangan sebanyak jumlah cluster yang ada. Dan semua nilai similarity setiap cluster disimpan pada variable array c dan cos.

## Implementasi Pengujian

|  |
| --- |
| sql = **"SELECT id FROM query WHERE isi=%s"** t = (query) cursor.execute(sql, t) idquery = cursor.fetchall() idquery = str(idquery).replace(**"("**, **""**).replace(**","**, **""**).replace(**")"**, **""**) sql = **"SELECT \* FROM query\_news WHERE id\_query=%s"** t = (idquery) cursor.execute(sql, t) n1 = cursor.fetchall() count = 0  **for** row **in** final:  id = row[0]  **for** rows **in** n1:  id2 = rows[1]  **if** id == id2:  count += 1 recall = float(count / len(n1)) precission = float(count / len(final)) fscore = float(2 \* (recall \* precission) / (recall + precission)) |

Gambar 5.15Source Code Perhitungan Similarity

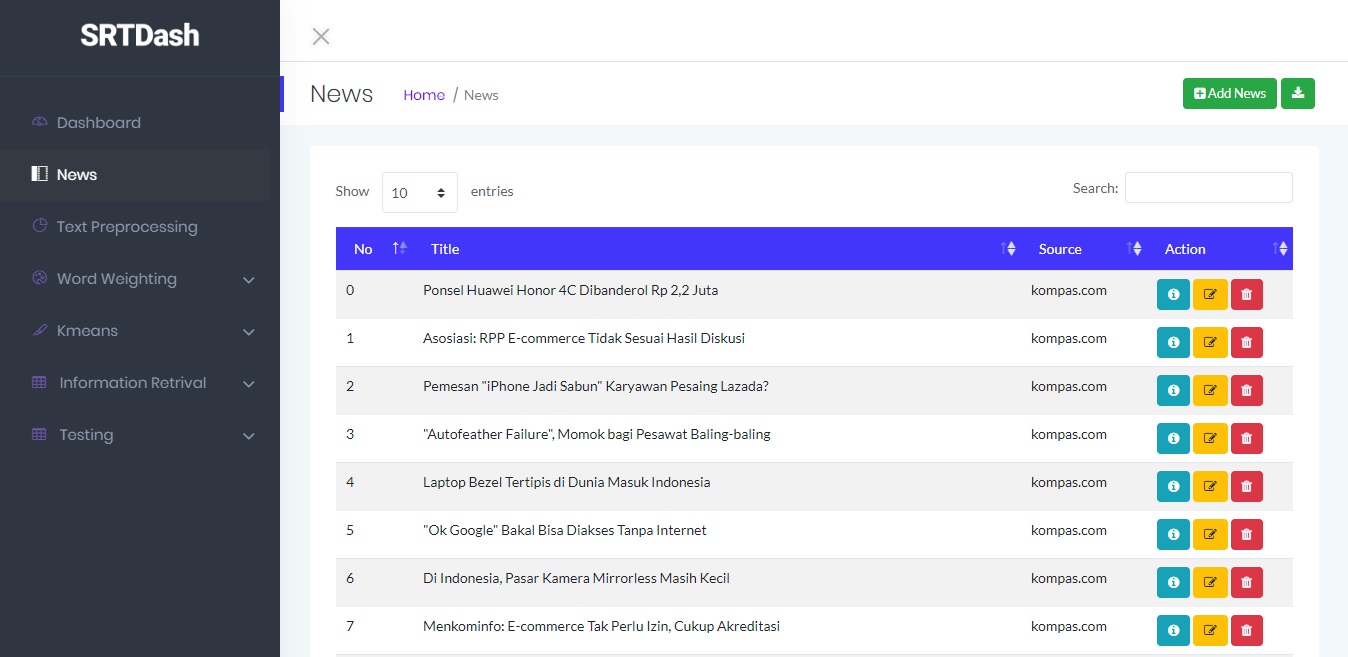
Gambar 5.15 menjelaskan mengenai perhitungan pengujian recall, precission dan fscore yang termasuk pada sistem. Dimana variable n1 menyimpan data query yang relevan dengan news yang tersimpan pada database, sedangkan variable final menyimpan data berita terpilih dari cluster yang mempunyai nilai kemiripan dengan query. Dan variable count menyimpan jumlah berita yang sama antara variable n1 dengan variable final. Setelah itu menghitung recall dengan membagi variable count dan jumlah data pada variable n1 dan disimpan pada variable recall. Untuk menghitung precission yaitu dengan membagi variable count dengan jumlah data pada variable final dan disimpan pada variable precission. Dan menghitung fscore yaitu membagi nilai perkalian anatara recall dan precission dengan penjumlahan recall dan precission setelah itu dikali dua dan disimpan pada variable fscore.

## Implementasi User Interface

Pada sub bab ini, dijelaskan penerapan user interface dari sistem temu kembali informasi yang menggunakan metode klastering pada tahap preprocessing yang sebelumnya telah dirancang menggunakan mockup. Sistem ini dibangun menggunakan HTML dan Flask dengan back-end Python. Templating yang digunakan antara flask dengan HTML adalah jinja2. Berikut merupakan implementasi user interface dari system klasifikasi jenis kelamin berdasarkan citra wajah dari sistem temu kembali informasi yang menggunakan metode klastering pada tahap preprocessing:

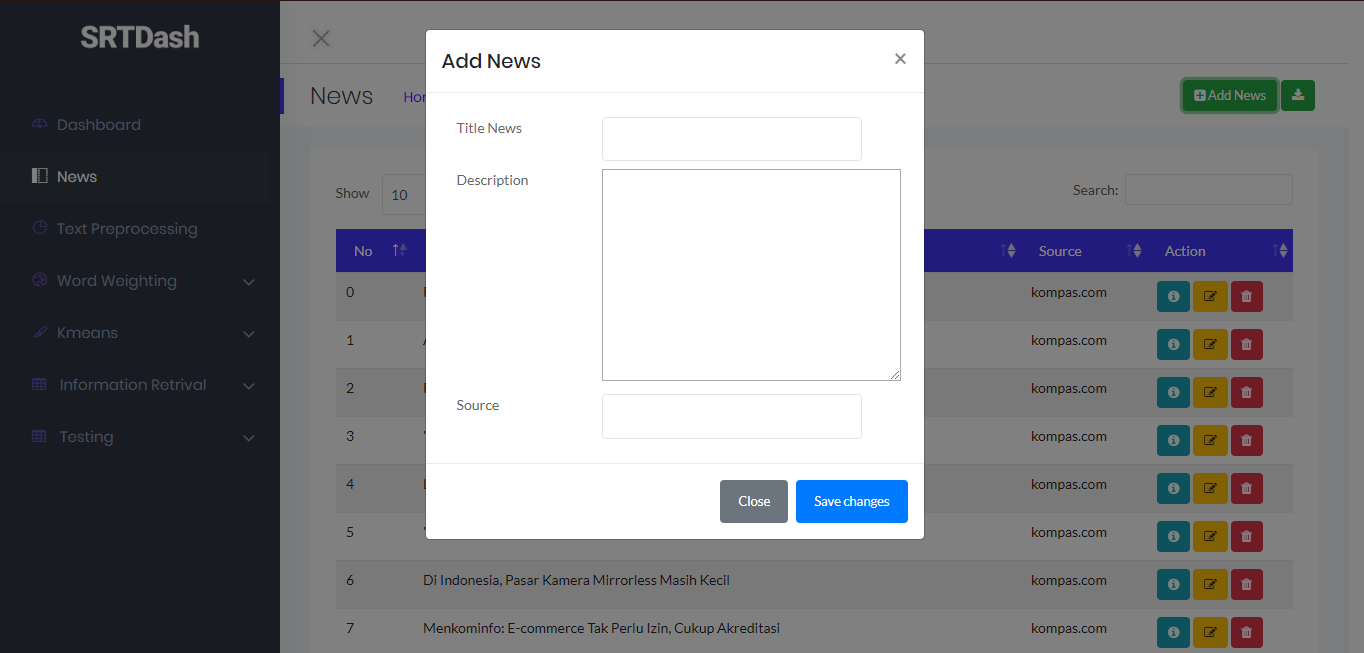
### Tampilan Halaman Dashboard

### Tampilan Halaman News



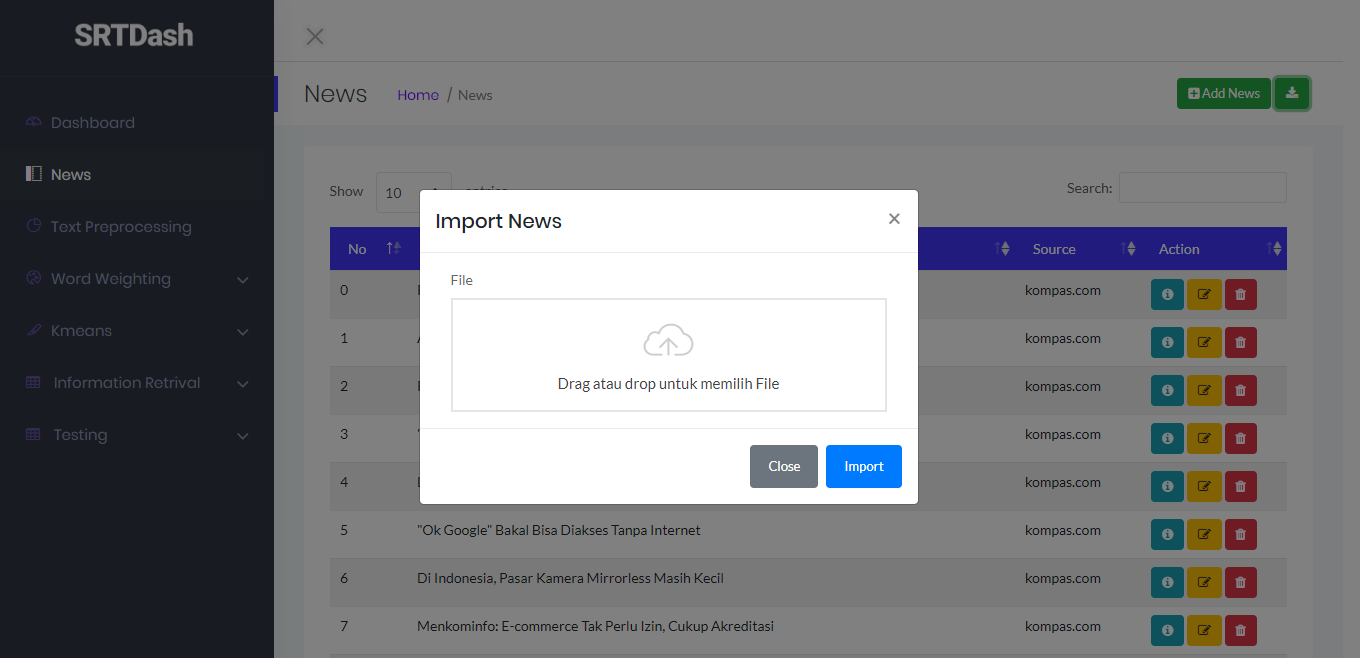
Gambar 5.16 Tampilan News

Pada gambar 5.16 menunjukkan halaman yang menampilkan semua data berita yang tersimpan di database. Data yang ditampilkan adalah judul berita, sumberberita dan isi berita. Pada halaman ini admin bisa menambahkan berita melalui tombol *Add News*.



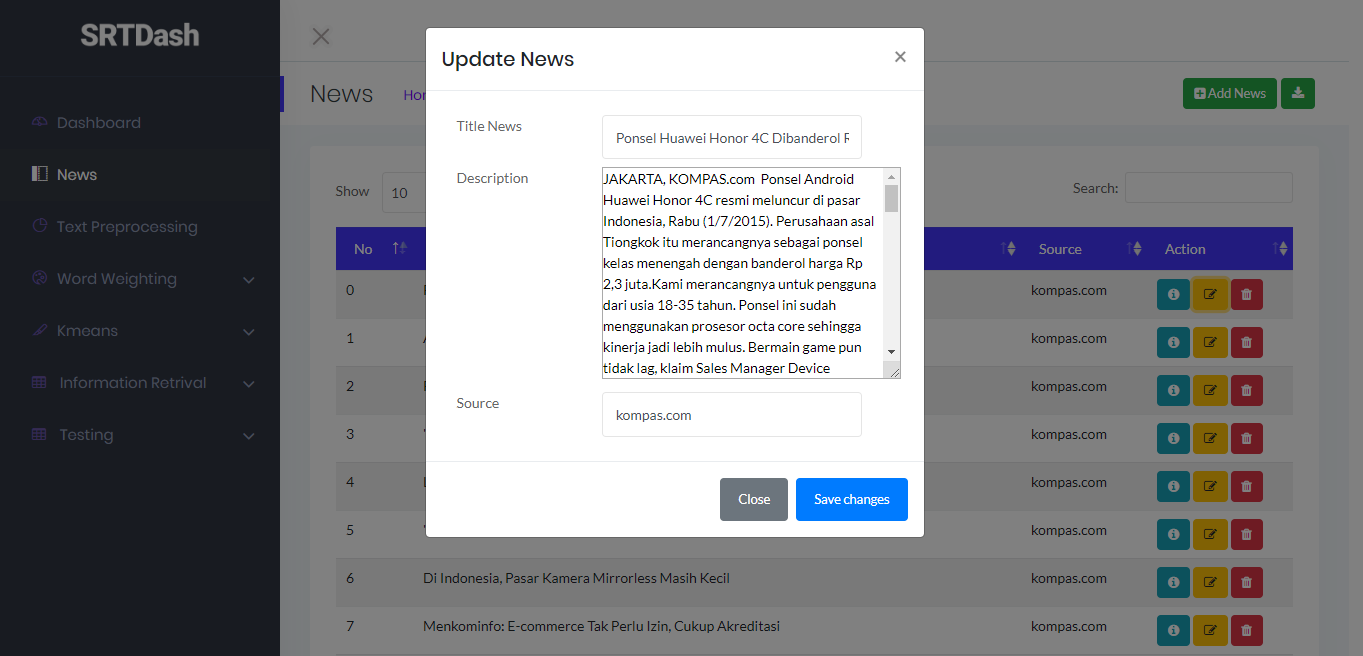
Gambar 5.17 Tampilan Form Add News

Pada gambar 5.17 menunjukkan tampilan form untuk menambahkan berita secara manual yang terdiri dari judul, isi dan sumber berita. Selain itu juga bisa menambahkan melalui format file json atau csv atau xls dengan melalui tombol *icon* import.



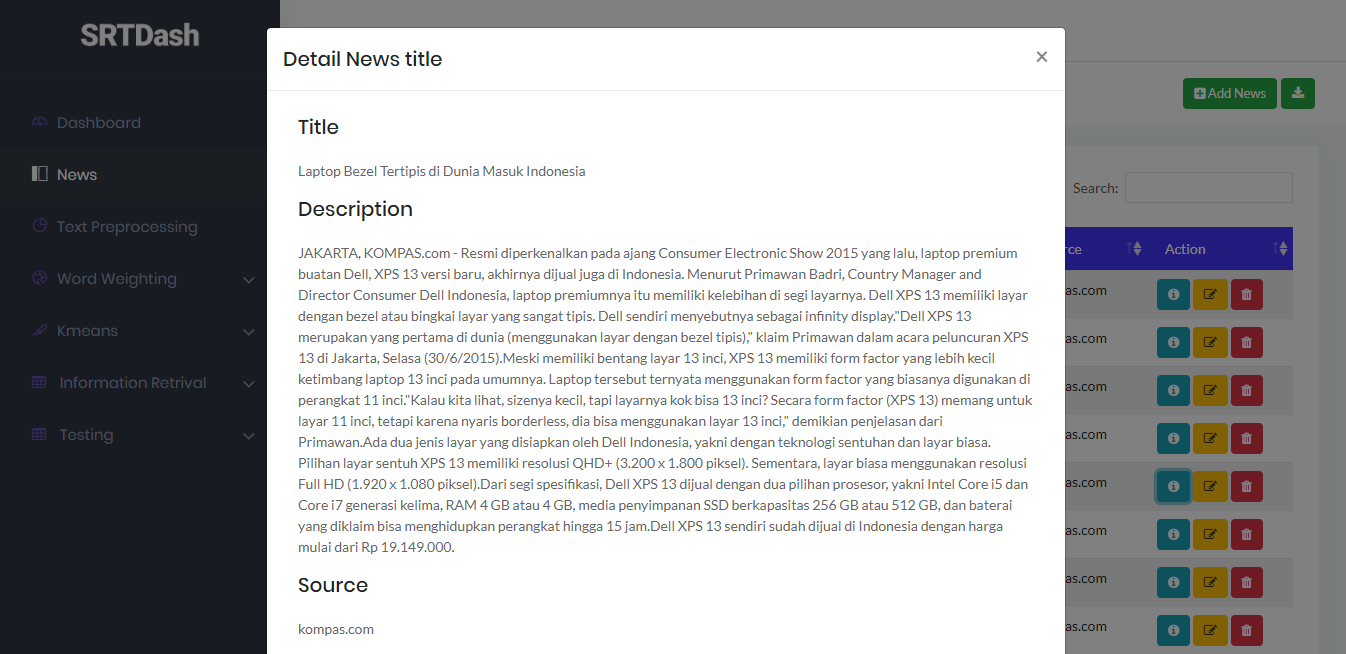
Gambar 5.18 Tampilan Form Import News

Pada gambar 5.18 menunjukkan tampilan form untuk menambahkan berita melalui import file. Selain itu juga bisa mengubah data berita yang ada di database dengan memilih button berwarna kuning pada setiap berita.



Gambar 5.19 Tampilan Form Update News

Pada gambar 5.19 menunjukkan tampilan form untuk mengubah berita mulai dari judul, isi dan sumber berita. Selain itu bisa juga menghapus berita dengan memilih tombol warna merah yang ada pada setiap berita. Untuk melihat detail berita dapat memilih tombol berwarna biru pada setiap berita.



Gambar 5.20 Tampilan Detail Berita

Pada gambar 5.20 menunjukkan tampilan detail berita, mulai dari judul, isi dan sumber berita.

### Tampilan Hasil Text Preprocessing

### Tampilan Hasil TF-IDF

### Tampilan Hasil Doc2vec

### Tampilan Hasil Kluster Kmeans - TFIDF

### Tampilan Hasil Kluster Kmeans - Doc2vec

### Tampilan Hasil Clustering Berita

### Tampilan Information Retrival Kmeans - TFIDF

### Tampilan Information Retrival Kmeans – Doc2vec

### Tampilan Information Retrival Unclustering

### Tampilan Testing Kmeans - TFIDF

### Tampilan Testing Kmeans – Doc2vec

### Tampilan Testing Unclustering

# **BAB VI**

# **PENGUJIAN DAN PEMBAHASAN**

## Pengujian

Pengujian adalah proses yang bertujuan untuk memastikan apakah semua fungsi sistem bekerja dengan baik dan mencari kesalahan yang mungkin terjadi pada sistem. Terdapat dua jenis pengujian yang dilakukan pada penelitian ini yaitu pengujian fungsional dan pengujian akurasi.

### Pengujian Fungsional

Pengujian fungsional adalah pengujian yang dilakukan untuk memastikan apakah fitur-fitur yang ada pada system yang dibuat sudah berjalan dengan benar tanpa adanya error dan telah sesuai spesifikasi yang dirancang. Tabel 6.1 menunjukkan hasil pengujian fungsional yang telah dilakukan.

*Tabel 6.1 Hasil Pengujian Fungsional*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Pengujian | Hasil yang Diharapkan | Hasil |
| 1. | Citra yang diinputkan dapat ditampilkan kembali ketika perhitungan. | Citra dapat ditampilkan dengan proses disimpan terlebih dahulu ke dalam storage | Berhasil |
| 2. | Sistem dapat mendeteksi adanya wajah atau tidak dalam citra inputan | Sistem dapat mendeteksi wajah dan apabila terdeteksi lebih dari satu wajah, maka system akan memberikan output berupa pesan pemberitahuan | Berhasil |
| 3. | Wajah yang terdeteksi dapat dilakukan cropping pada area wajah saja | Proses cropping dapat dilakukan dengan benar dan hasil dari cropping disesuaikan dengan aturan system | Berhasil |
| 4. | Area wajah dapat dilakukan proses pendeteksian landmark point | Landmark point dari wajah yang dideteksi dapat ditampilkan dan dapat dihitung | Berhasil |
| 5. | Perhitungan klasifikasi dapat berjalan hingga keseluruhan proses | Proses klasifikasi dapat berjalan dengan benar | Berhasil |
| 6. | Proses route tiap url dapat berjalan sesuai url masing-masing menu | Setiap menu yang ada dapat melakukan berpindah sesuai url masing-masing | Berhasil |
| 7. | Citra yang diinputkan dapat disimpan ke dalam storage | Citra yang dibutuhkan berhasil disimpan ke dalam storage sehingga dapat diolah kembali | Berhasil |
| 8. | Hasil ekstraksi fitur pada proses labeling data dapat disimpan ke dalam database | Setiap fitur yang dibutuhkan dapat disimpan ke dalam database | Berhasil |
| 9. | Radio button dan field untuk upload data dapat digunakan | Komponen pada form dapat diberikan inputan dengan baik | Berhasil |
| 10. | Tombol collapse untuk menyembunyikan data | Data yang terlalu banyak dapat disembunyikan menggunakan tombol collapse | Berhasil |

### Pengujian Data Antropometri

Data antropometri yang didapatkan dibandingkan dengan hasil ektraksi data training yang digunakan. Hasil tersebut menunjukkan apakah nilai hasil ekstraksi sudah memenuhi range yang telah ditetapkan pada aturan antropometri. Tabel 6.2 menjelaskan hasil pengujian data antropometri pada kelas negatif, sedangkan pada tabel 6.3 menjelaskan hasil pengujian data antropometri pada kelas positif.

*Tabel 6.2 Pengujian Antropometri Kelas Negatif (Laki-laki)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fitur |  | Antropometri | Riil | Hasil |
| Tinggi Bibir Bawah | Min | 10.94258 | 8.46666 | Tidak Sesuai |
| Max | 25.95993 | 18.52083 | Sesuai |
| Tinggi Bibir Atas | Min | 16.28089 | 5.82083 | Tidak Sesuai |
| Max | 30.21653 | 12.69999 | Sesuai |
| Jarak Mata | Min | 23.98284 | 40.21666 | Sesuai |
| Max | 41.31963 | 54.50416 | Tidak Sesuai |
| Lebar Hidung | Min | 30.51525 | 31.22083 | Sesuai |
| Max | 46.86625 | 42.86249 | Sesuai |
| Lebar Nasal Root | Min | 8.130884 | 15.34583 | Sesuai |
| Max | 40.90964 | 21.43124 | Sesuai |

*Tabel 6.3 Hasil Pengujian Antropometri pada Kelas Positif (Female)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fitur |  | Antropometri | Riil | Hasil |
| Tinggi Bibir Bawah | Min | 10.21712 | 10.05416 | Sesuai |
| Max | 22.55754 | 17.72708 | Sesuai |
| Tinggi Bibir Atas | Min | 13.59849 | 6.34999 | Tidak Sesuai |
| Max | 24.94134 | 11.37708 | Sesuai |
| Jarak Mata | Min | 25.50702 | 38.62916 | Sesuai |
| Max | 37.89224 | 56.09166 | Tidak Sesuai |
| Lebar Hidung | Min | 28.01161 | 26.98749 | Tidak Sesuai |
| Max | 42.1081 | 42.06874 | Sesuai |
| Lebar Nasal Root | Min | 7.756878 | 13.49374 | Sesuai |
| Max | 31.36896 | 21.69583 | Sesuai |

### Pengujian Akurasi

Pengujian akurasi merupakan pengujian untuk melakukan pengecekan apakah metode yang dignakan meberikan hasil sesuai harapan atau tidak. Pada pengujian akurasi, disiapkan 20 image wajah yang terdiri dari 10 image laki-laki dan 10 image perempuan. Tabel 6.2 menunjukkan hasil klasifikasi menggunakan metode Support Vector Machine dengan 60 data training yang terdiri dari 30 image laki-laki dan 30 image perempuan.

*Tabel 6.4 Hasil Pengujian Klasifikasi*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Harapan | Hasil |
| 1 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkbili.jpg | Laki-laki | Perempuan |
| 2 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmark27.JPG | Laki-laki | Perempuan |
| 3 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME017.jpg | Laki-laki | Laki-laki |
| 4 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME018.jpg | Laki-laki | Laki-laki |
| 5 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME019.jpg | Laki-laki | Perempuan |
| 6 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME020.jpg | Laki-laki | Laki-laki |
| 7 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME021.jpg | Laki-laki | Laki-laki |
| 8 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME022.JPG | Laki-laki | Perempuan |
| 9 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkfaizin.jpg | Laki-laki | Laki-laki |
| 10 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkvico.jpg | Laki-laki | Laki-laki |
| 11 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkami.jpg | Perempuan | Perempuan |
| 12 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkayu.jpg | Perempuan | Perempuan |
| 13 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkcindi.jpg | Perempuan | Laki-laki |
| 14 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkdewi.jpg | Perempuan | Perempuan |
| 15 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkfaizah.jpg | Perempuan | Laki-laki |
| 16 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkismi.jpg | Perempuan | Perempuan |
| 17 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkjunita.jpg | Perempuan | Perempuan |
| 18 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarklaeily.jpg | Perempuan | Perempuan |
| 19 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarknindi.jpg | Perempuan | Perempuan |
| 20 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkrosy.jpg | Perempuan | Perempuan |

Dari hasil testing yang ditunjukkan pada tabel 6.2, dapat dilihat bahwa terdapat 6 dari 20 data yang hasilnya tidak sesuai harapan. Enam data tersebut terdiri dari 2 perempuan dan 4 laki-laki yang tidak berhasil diklasifikasikan sesuai harapan. Dengan hasil perhitungan yang ada, dapat dibuat confusion matrix yang ditunjukkan pada tabel 6.5.

*Tabel 6.5 Confusion Matriks*

|  |  |  |
| --- | --- | --- |
|  | Terklasifikasi Positif | Terklasifikasi Negatif |
| Positif | 8 | 2 |
| Negatif | 4 | 6 |

*(6.1)*

%

Dari perhitungan confusion matrix pada persamaan 6.1. Dapat dilihat bahwa akurasi klasifikasi yang dilakukan dalam penelitian ini adalah 70%. Selain itu, pengujian juga dilakukan menggunakan beberapa jenis data training yaitu data training jenis A yaitu data training berupa citra wajah siswa SMK Negeri 8 Malang dan data training jenis B yaitu data training berupa citra wajah mahasiswa Politeknik Negeri Malangyang didapat dari penelitian Bella Sita Andjani. Hasil perhitungan akurasi dapat dilihat pada tabel 6.4.

*Tabel 6.6 Hasil Akurasi Klasifikasi terhadap Jenis Data Training*

|  |  |  |  |
| --- | --- | --- | --- |
| Data Training | Accuracy | Precision | Recall |
| Data A | 40% | 30% | 37.5% |
| Data B | 65% | 70% | 64% |
| Data A dan B | 70% | 80% | 67% |

# **BAB VII**

# **KESIMPULAN DAN SARAN**

## Kesimpulan

Berdasarkan pembahasan pada bab 1 hingga bab 6 dapat diambil kesimpulan sebagai berikut:

1. Sistem yang dibuat dapat mengklasifikasikan jenis kelamin berdasarkan citra wajah.
2. Sistem klasifikasi jenis kelamin berdasarkan citra wajah menggunakan Haar Cascade untuk mendeteksi wajah, dlib Regression Tree untuk ekstraksi fitur, dan Support Vector Machine untuk klasifikasi.
3. Akurasi perhitungan klasifikasi masih belum bagus dikarenakan fitur yang digunakan memiliki nilai yang dekat satu sama lain sehingga sulit untuk diklasifikasikan. Dengan seluruh data yang digunakan, akurasi klasifikasi yang dilakukan dalam penelitian ini adalah 70%.
4. Terdapat beberapa ukuran antropometri yang tidak sesuai dalam penerapan ekstraksi fitur pada wajah.

## Saran

Berdasarkan penelitian yang dilakukan beserta masalah yang terjadi dapat diberikan saran sebagai berikut:

1. Ditampilkan scatter plot multidimensional berdasarkan fitur dari data yang digunakan sebagai training dan ditunjukkan garis pemisah (hyperplane) antar dua kelas.
2. Melakukan pencarian fitur yang nilainya tidak berdekatan antara kelas satu dengan kelas lainnya agar hasil klasifikasi menjadi lebih baik.
3. Menggunakan metode deep learning.

# **DAFTAR PUSTAKA**

[1] Elizabeth, 2013, Pengembangan Sistem Identifikasi Biometrik Wajah Menggunakan Metode Neural Network dan Pattern Matching, Depok: Universitas Indonesia.

[2] P. Choirina dan R. A. Asmara, 2016, Deteksi Jenis Kelamin berdasarkan Citra Wajah Jarak Jauh dengan Metode Haar Cascade Classifier, Malang: Politeknik Negeri Malang.

[3] R. A. Asmara, B. S. Andjani, U. D. Rosiani, P. Choirina, 2017, Klasifikasi Jenis Kelamin pada Citra Wajah Menggunakan Metode Naive Bayes, Malang: Politeknik Negeri Malang.

[4] Z. Xu, L. Lu, and P. Shi, 2008, *A Hybrid Approach to Gender Classification from Face Images*. Shanghai: Shanghai Jiao Tong University.

[5] U. Elfiah, I. L. Putri, M. R. Hutagalung, D. S. Perdanakusuma, and T. Kosbandriati, 2011, Variasi Antropometri, Wajah Indonesia dan Sefalometri sebagai Data Dasar pada Rekonstruksi Trauma Maksilofasial, Surabaya: Journal of Emergency Vol. 1. No. 1 Desember 2011.

[6] D. Rathod Ȧ, Ȧ S. S., dan S. Natarajan Ȧ Ȧ, 2014, Facial Landmark Localization-A Literature Survey, India: International Journal of Current Engineering and Technology, Vol.4, No.3.

[7] Y. Ferik, H. Octavianto, and D. H. Wahyu, 2012, Deteksi Wajah Menggunakan Algoritma Viola-Jones, Jember: Universitas Muhammadiyah Jember.

[8] B. Santosa, 2008, Tutorial Support Vector Machines, Surabaya: Institut Sepuluh Nopember Surabaya.

[9] N. Neneng, K. Adi, and R. Isnanto, 2016, Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co-Occurrence Matrices (GLCM), Semarang: Jurnal Sistem Informasi Bisnis.

[10] R. S. Pressman, 2010, *Software Engineering*. New York: McGraw-Hill.

# **LAMPIRAN**

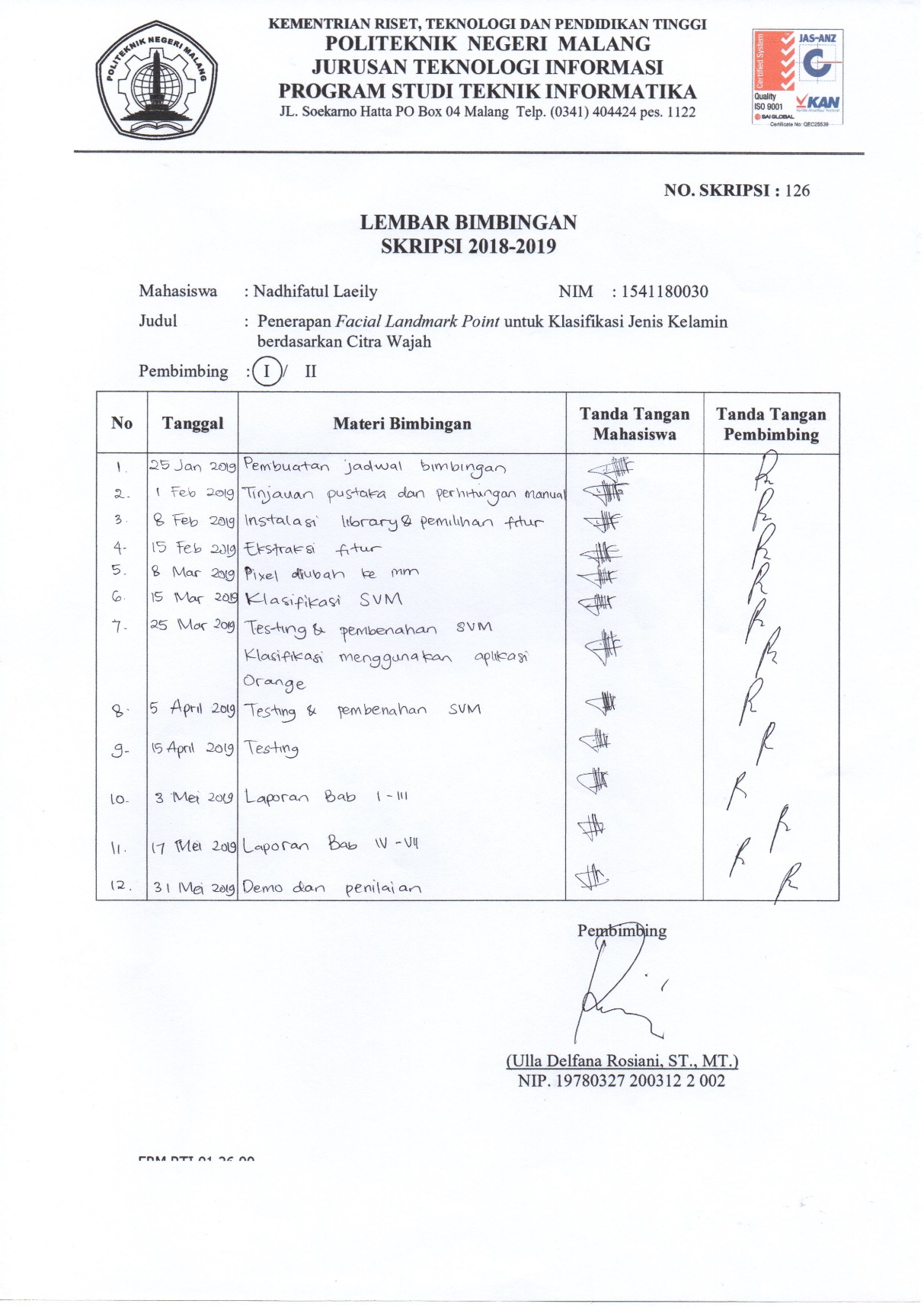
*Lampiran 1. Hasil Pengujian Akurasi Data Training Jenis A*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Harapan | Hasil |
| 1 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkbili.jpg | Laki-laki | Laki-laki |
| 2 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmark27.JPG | Laki-laki | Laki-laki |
| 3 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME017.jpg | Laki-laki | Perempuan |
| 4 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME018.jpg | Laki-laki | Perempuan |
| 5 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME019.jpg | Laki-laki | Laki-laki |
| 6 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME020.jpg | Laki-laki | Perempuan |
| 7 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME021.jpg | Laki-laki | Perempuan |
| 8 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME022.JPG | Laki-laki | Laki-laki |
| 9 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkfaizin.jpg | Laki-laki | Laki-laki |
| 10 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkvico.jpg | Laki-laki | Perempuan |
| 11 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkami.jpg | Perempuan | Laki-laki |
| 12 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkayu.jpg | Perempuan | Laki-laki |
| 13 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkcindi.jpg | Perempuan | Perempuan |
| 14 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkdewi.jpg | Perempuan | Laki-laki |
| 15 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkfaizah.jpg | Perempuan | Perempuan |
| 16 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkismi.jpg | Perempuan | Laki-laki |
| 17 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkjunita.jpg | Perempuan | Laki-laki |
| 18 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarklaeily.jpg | Perempuan | Laki-laki |
| 19 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarknindi.jpg | Perempuan | Perempuan |
| 20 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkrosy.jpg | Perempuan | Laki-laki |

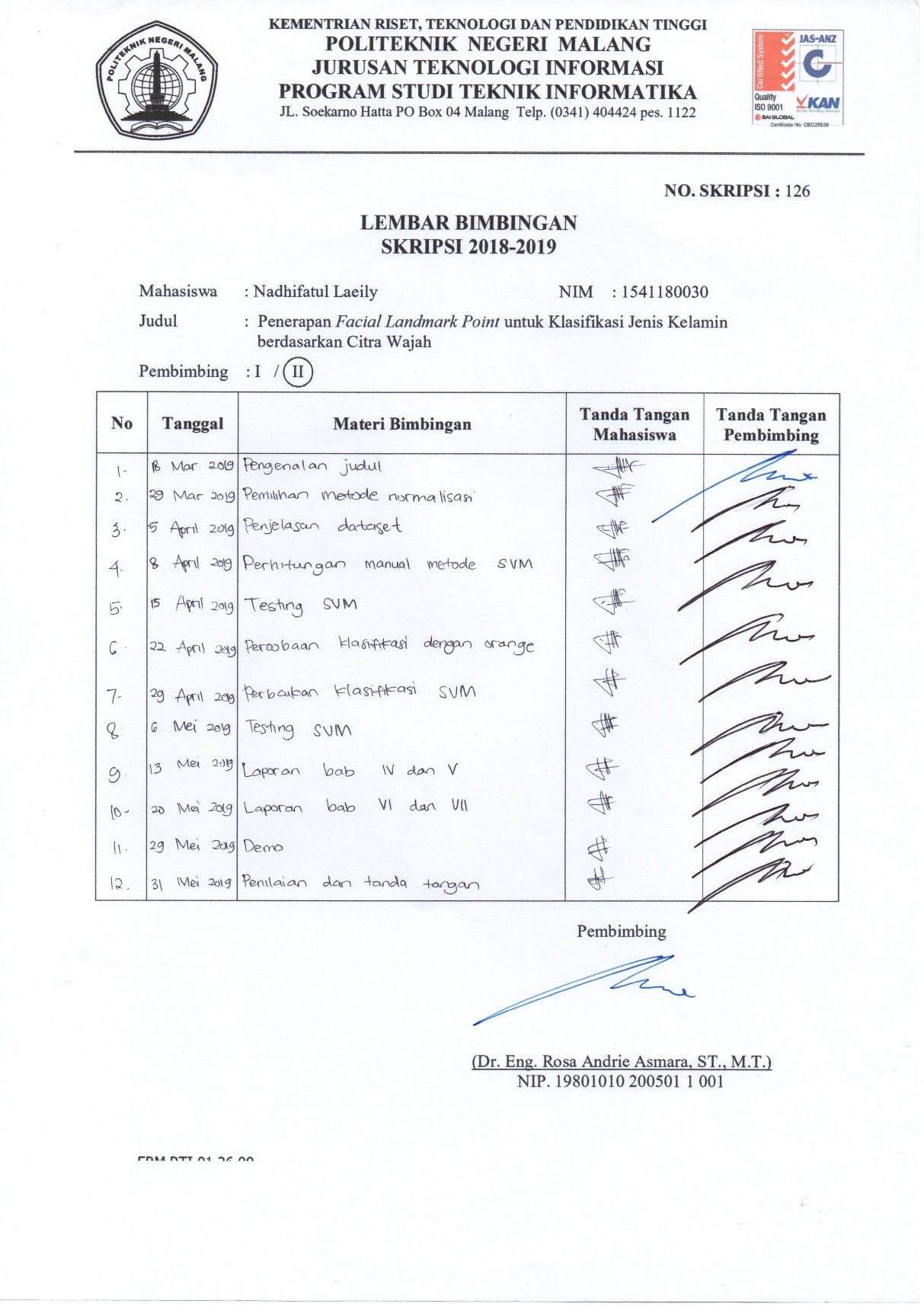
*Lampiran 2. Hasil Pengujian Akurasi Data Training Jenis B*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Harapan | Hasil |
| 1 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkbili.jpg | Laki-laki | Perempuan |
| 2 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmark27.JPG | Laki-laki | Perempuan |
| 3 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME017.jpg | Laki-laki | Laki-laki |
| 4 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME018.jpg | Laki-laki | Laki-laki |
| 5 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME019.jpg | Laki-laki | Perempuan |
| 6 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME020.jpg | Laki-laki | Laki-laki |
| 7 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME021.jpg | Laki-laki | Laki-laki |
| 8 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkME022.JPG | Laki-laki | Perempuan |
| 9 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkfaizin.jpg | Laki-laki | Laki-laki |
| 10 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkvico.jpg | Laki-laki | Laki-laki |
| 11 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkami.jpg | Perempuan | Perempuan |
| 12 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkayu.jpg | Perempuan | Perempuan |
| 13 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkcindi.jpg | Perempuan | Laki-laki |
| 14 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkdewi.jpg | Perempuan | Perempuan |
| 15 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkfaizah.jpg | Perempuan | Laki-laki |
| 16 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkismi.jpg | Perempuan | Perempuan |
| 17 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkjunita.jpg | Perempuan | Perempuan |
| 18 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarklaeily.jpg | Perempuan | Perempuan |
| 19 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarknindi.jpg | Perempuan | Laki-laki |
| 20 | E:\7\SKRIPSI\facial-landmarks\myflask\static\images\landmarkrosy.jpg | Perempuan | Perempuan |

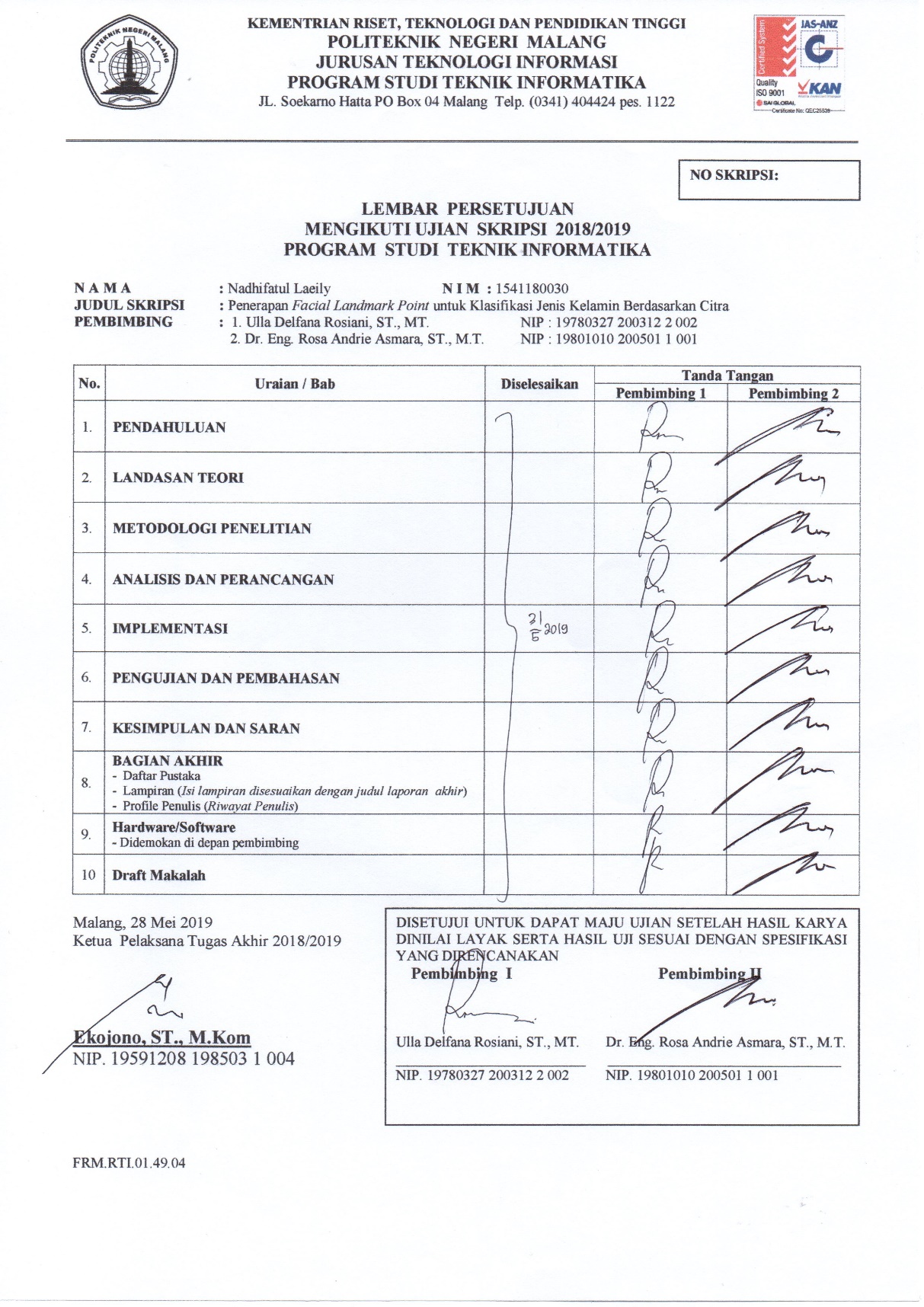
*Lampiran 3. Lembar Bimbingan Skripsi Pembimbing I*



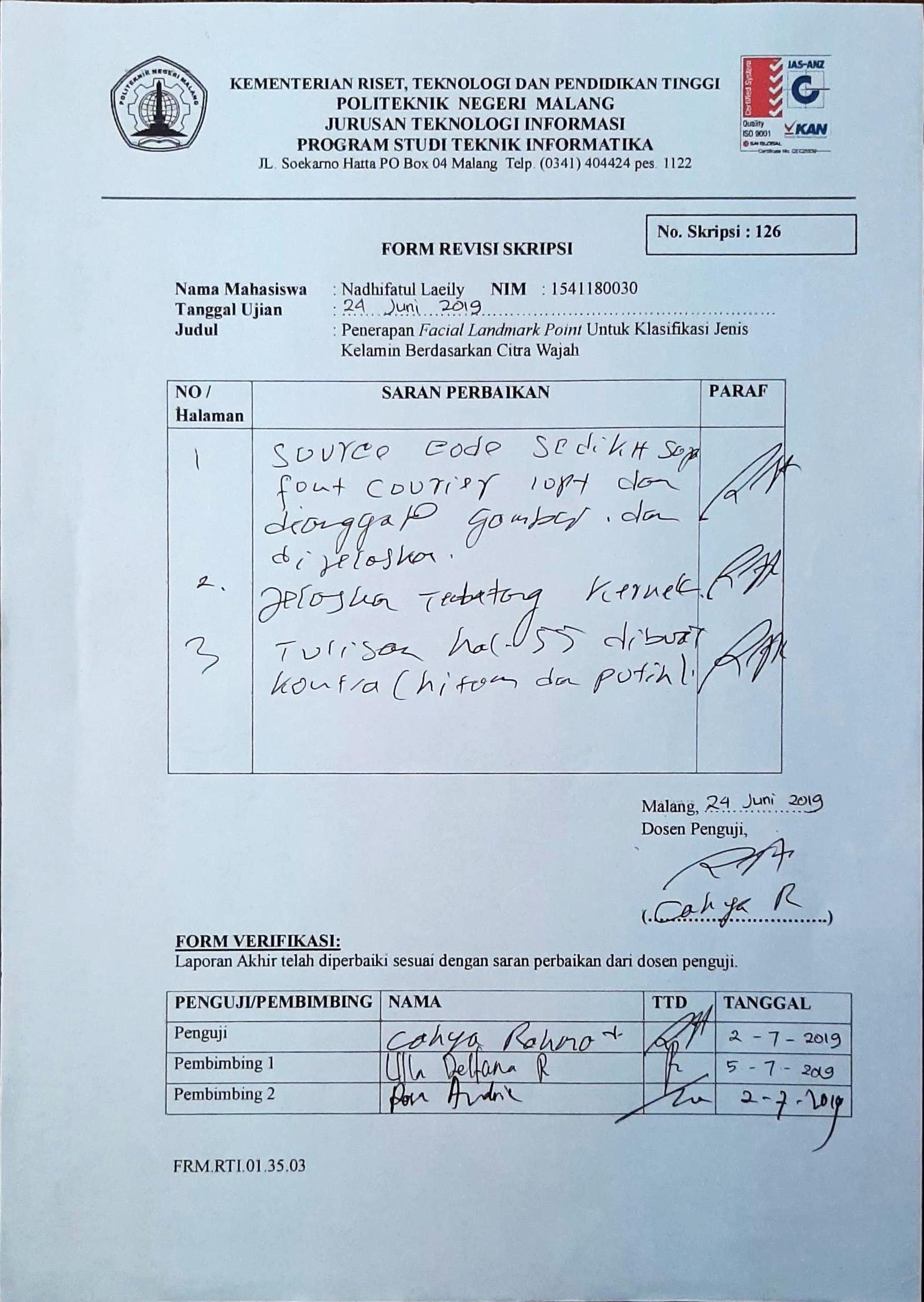
*Lampiran 4. Lembar Bimbingan Skripsi Pembimbing II*



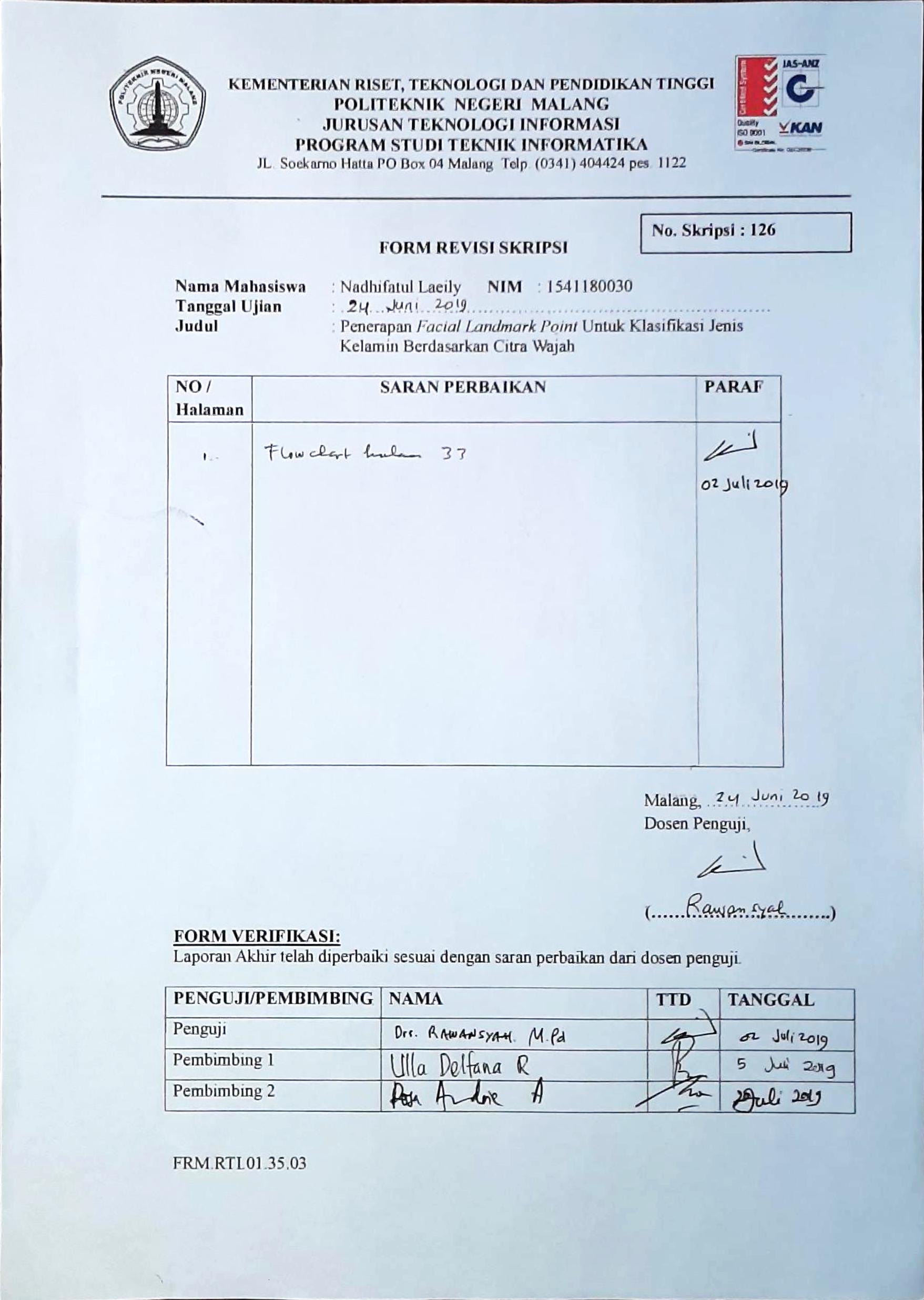
*Lampiran 5. Lembar Persetujuan Maju Ujian Skripsi*



*Lampiran 6. Lembar Revisi Penguji I*



*Lampiran 7. Lembar Revisi Penguji II*



*Lampiran 8. Lembar Verifikasi Abstrak dan Tata Tulis*



# **PROFIL PENULIS**



**DATA PRIBADI**

|  |  |  |
| --- | --- | --- |
| Nama | : | Nadhifatul Laeily |
| Tempat, Tanggal Lahir | : | Malang, 30 November 1996 |
| Jenis Kelamin | : | Perempuan |
| Alamat | : | Jl. Gondang Selatan 1/5, Randuagung, Singosari, Kab. Malang |
| Email | : | lely0330@gmail.com |
| No. Telepon | : | 081216283258 |

**DATA PENDIDIKAN**

|  |  |  |
| --- | --- | --- |
| **Nama** | **Tahun** | **Jurusan** |
| MI Almaarif 09 Randuagung | 2003-2009 | - |
| SMP Negeri 1 Singosari | 2009-2012 | - |
| SMK Negeri 8 Malang | 2012-2015 | Teknik Komputer Jaringan |
| Politeknik Negeri Malang | 2015-Sekarang | Teknik Informatika |