



**SISTEM REKOMENDASI DOSEN PEMBIMBING  
BERDASARKAN KONTEN LATAR BELAKANG PADA NASKAH  
PROPOSAL SKRIPSI MENGGUNAKAN METODE *MULTI-CLASS  
SUPPORT VECTOR MACHINE DAN WEIGHTED PRODUCT***

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Yustinus Radityo Pradana

NIM: 155150400111016



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA

MALANG

2020

## PENGESAHAN

SISTEM REKOMENDASI DOSEN PEMBIMBING BERDASARKAN KONTEN LATAR  
BELAKANG PADA NASKAH PROPOSAL SKRIPSI MENGGUNAKAN METODE *MULTI-  
CLASS SUPPORT VECTOR MACHINE* DAN *WEIGHTED PRODUCT*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Yustinus Radityo Pradana  
NIM: 155150400111016

Skripsi ini telah diuji dan dinyatakan lulus pada  
22 Juli 2020

Telah diperiksa dan disetujui oleh:

Pembimbing 1



Dr. Eng. Ahmad Afif Supianto, S.Si.,  
M.Kom  
NIK: 201201 820623 1 001

Pembimbing 2

digitally signed by Yusi Tyroni Mursityo  
12-Aug-20

Yusi Tyroni Mursityo, S.Kom., M.AB  
NIK: 198002282006041001

Mengetahui

Ketua Jurusan Sistem Informasi



Issa Arwani, S.Kom, M.Sc  
NIP: 198309222012121003



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 Juli 2020

Yustinus Radityo Pradana  
NIM: 155150400111016



## ABSTRAK

**Yustinus Radityo Pradana, Sistem Rekomendasi Dosen Pembimbing Berdasarkan Latar Belakang Menggunakan Metode *Multi-Class Support Vector Machine* Dan *Weighted Product***

**Pembimbing: Dr. Eng. Ahmad Afif Supianto, S.Si., M.Kom dan Yusi Tyroni Mursityo, S.Kom., M.AB**

Pada Fakultas Ilmu Komputer Universitas Brawijaya, pengerjaan skripsi dimulai dengan melakukan pembuatan praproposal yang berisi latar belakang dan bidang skripsi. Dalam pengerjaan skripsi, mahasiswa butuh pendampingan oleh dosen pembimbing. Dosen pembimbing berfungsi sebagai motivator, pendamping serta pemberi arahan bagi mahasiswa yang sedang mengerjakan skripsi. Dosen pembimbing menjadi krusial dalam pengerjaan skripsi seorang mahasiswa. Oleh karena hal tersebut, pemilihan dosen pembimbing yang memiliki bidang keahlian yang sesuai dengan topik skripsi sangat penting. Topik skripsi ditentukan dari latar belakang sebuah proposal skripsi. Untuk menjawab permasalahan dibutuhkan sebuah sistem yang mampu menentukan topik skripsi dan memprediksi dosen pembimbing yang sesuai berdasarkan latar belakang proposal skripsi. Rekomendasi dosen diberikan berdasarkan kesesuaian bidang dosen dengan topik dan beberapa data dosen yang didapatkan dari unit Pengelola Sistem Informasi, Infrastruktur TI dan Kehumasan Fakultas Ilmu Komputer (PSIK-FILKOM) seperti jurusan dosen, sisa kuota bimbingan, tingkatan gelar, dan beban kerja. Hasil pengujian akurasi menghasilkan akurasi klasifikasi bidang skripsi sebesar 93,75% dan akurasi prediksi dosen pembimbing sebesar 57,14%. Hasil pengujian unit menunjukkan bahwa sistem 100% valid dan sesuai dengan kebutuhan. Hasil pengujian kompatibilitas menunjukkan bahwa sistem dapat digunakan di berbagai *web browser*, sistem operasi, dan *platform*. Hasil pengujian performa menunjukkan bahwa 85,71% bagian dari sistem telah responsive dan dapat digunakan dengan baik.

Kata kunci: rekomendasi dosen, pembimbing, *support vector machine*, *weighted product*, topik, skripsi.

**ABSTRACT**

**Yustinus Radityo Pradana, Sistem Rekomendasi Dosen Pembimbing Berdasarkan Latar Belakang Menggunakan Metode Multi-Class Support Vector Machine Dan Weighted Product**

**Supervisors: Dr. Eng. Ahmad Afif Supianto, S.Si., M.Kom and Yusi Tyroni Mursityo, S.Kom., M.AB Dr. Eng. Ahmad Afif Supianto, S.Si., M.Kom dan Yusi Tyroni Mursityo, S.Kom., M.AB**

In Fakultas Ilmu Komputer Universitas Brawijaya, thesis work started by making *preproposal* which contains the background and thesis field/topic. In the working of thesis, undergraduate student needs to be accompanied by a supervisor. Supervisor serve as motivator, companion, and guider for undergraduate students who are doing their thesis research. Supervisor roles become crucial in the working of thesis. Therefore, the selection of supervisor who have areas of expertise that matching with thesis topic is very important. The topic of thesis can be determined by the background of thesis's proposal. To answer the problem, we need a system that able to determine the topic of thesis and giving some prediction about matching supervisor based on the background of thesis's proposal. Supervisor recommendation done based on the coherency of supervisor's expertise with the thesis's topic and also based on some supervisor data that was obtained from unit Pengelola Sistem Informasi, Infrastruktur TI dan Kehumasan Fakultas Ilmu Komputer (PSIK FILKOM). The data that was obtained from unit PSIK FILKOM are supervisor's majors, remaining quota for supervising, degree level, and work load. The result of accuracy testing shows the accuracy of thesis's topic is 93.75% and accuracy of the supervisor prediction is 57.14%. The unit testing result shows that the system is 100% valid and works as expected. The compatibility test result shows that the system can be used on various web browser, operating system, and platform. The performance test result shows that 85.71% of the system works responsively and can be used properly.

**Keywords:** supervisor recommendation, guider, support vector machine, weighted product, topic, thesis.



**DAFTAR ISI**

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.1 Rumusan Masalah.....	2
1.2 Tujuan.....	3
1.3 Manfaat.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN.....</b>	<b>5</b>
2.1 Tinjauan Pustaka.....	5
2.1.1 Tinjauan Pustaka Pertama.....	5
2.1.2 Tinjauan Pustaka Kedua.....	6
2.1.3 Tinjauan Pustaka Ketiga.....	6
2.1.4 Tinjauan Pustaka Keempat.....	7
2.1.5 Tinjauan Pustaka Kelima.....	7
2.1.6 Tinjauan Pustaka Keenam.....	8
2.1.7 Tinjauan Pustaka Ketujuh.....	8
2.1.8 Tinjauan Pustaka Kedelapan.....	8
2.2 Skripsi.....	13
2.3 Dosen Pembimbing Skripsi.....	13
2.4 Jabatan Fungsional Dosen.....	14
2.5 <i>Unified Modelling Language (UML)</i> .....	14
2.6 <i>Use Case Diagram</i> .....	15







**DAFTAR GAMBAR**

Gambar 2.1 Posisi Penelitian Terhadap Penelitian Sebelumnya.....	5
Gambar 2.2 Tipe Penelitian Skripsi Fakultas Ilmu Komputer Universitas Brawijaya .....	13
Gambar 2.3 Contoh Penggunaan <i>Use Case Diagram</i> Aplikasi Daring Toko Musik Digital.....	15
Gambar 2.4 Contoh Penggunaan <i>Activity Diagram</i> Cara Memasak Kue.....	17
Gambar 2.4 Contoh Penggunaan <i>Sequence Diagram</i> Menampilkan Persegi.....	19
Gambar 2.6 Contoh Penggunaan <i>Class Diagram</i> menampilkan kelas Thoroughbred.....	21
Gambar 2.7 Alur <i>Data Preprocessing</i> .....	23
Gambar 2.8 Taksonomi dari Metode MCDM.....	27
Gambar 3.1 Diagram Alir Metodologi.....	30
Gambar 4.1 Proses Bisnis <i>As-Is</i> Pemilihan Dosen Pembimbing 1.....	33
Gambar 4.2 Proses Bisnis <i>To-Be</i> Pemilihan Dosen Pembimbing 1.....	35
Gambar 4.3 Diagram <i>Use Case</i> Sistem Rekomendasi Dosen Pembimbing.....	38
Gambar 4.4 <i>Activity Diagram</i> Login.....	40
Gambar 4.5 <i>Activity Diagram</i> Membuat Akun.....	41
Gambar 4.6 <i>Activity Diagram</i> Logout.....	41
Gambar 4.7 <i>Activity Diagram</i> Memproses Rekomendasi.....	42
Gambar 4.8 <i>Activity Diagram</i> Melihat <i>History</i> .....	43
Gambar 4.9 <i>Activity Diagram</i> Melihat Detail <i>History</i> Rekomendasi.....	43
Gambar 5.1 Arsitektur Sistem.....	44
Gambar 5.2 <i>Sequence Diagram</i> Login.....	45
Gambar 5.3 <i>Sequence Diagram</i> Membuat Akun.....	46
Gambar 5.4 <i>Sequence Diagram</i> Logout.....	46
Gambar 5.5 <i>Sequence Diagram</i> Memproses Rekomendasi.....	47
Gambar 5.6 <i>Sequence Diagram</i> Melihat <i>History</i> Rekomendasi.....	47
Gambar 5.7 <i>Sequence Diagram</i> Melihat Detail <i>History</i> Rekomendasi.....	48
Gambar 5.8 <i>Class Diagram</i> Controller.....	48
Gambar 5.9 <i>Class Diagram</i> Model.....	49
Gambar 5.10 <i>Class Diagram</i> Web Service.....	49



Gambar 5.11 Entity Relational Diagram.....	50
Gambar 5.12 Perancangan Antarmuka Halaman Login.....	53
Gambar 5.13 Perancangan Antarmuka Halaman Registrasi.....	53
Gambar 5.14 Perancangan Antarmuka Halaman List Rekomendasi.....	54
Gambar 5.15 Perancangan Antarmuka Halaman Submit Latar Belakang.....	55
Gambar 5.16 Perancangan Antarmuka Halaman List History Rekomendasi.....	55
Gambar 5.17 Perancangan Antarmuka Halaman Detail History Rekomendasi.....	56
Gambar 5.18 Tampilan Antarmuka Halaman Login.....	81
Gambar 5.19 Tampilan Antarmuka Halaman Registrasi.....	82
Gambar 5.20 Tampilan Antarmuka Halaman List Rekomendasi.....	83
Gambar 5.21 Tampilan Antarmuka Halaman Submit Latar Belakang.....	83
Gambar 5.22 Tampilan Antarmuka Halaman List History Rekomendasi.....	84
Gambar 5.23 Tampilan Antarmuka Halaman detail History Rekomendasi.....	84
Gambar 6.1 Persebaran Data Rekomendasi pada Data Uji 20%.....	87
Gambar 6.2 Persebaran Jumlah Skripsi pada 21 Dosen.....	88
Gambar 6.3 Pembagian Jumlah Skripsi Per KJFD.....	89
Gambar 6.4 Pembagian Jumlah Data Uji Per KJFD.....	89
Gambar 6.5 Akurasi Per KJFD pada Data Uji 20%.....	90
Gambar 6.6 Hasil Pengujian Kompatibilitas pada Web Browser Chrome Sistem Operasi Windows 10 Platform PC.....	95
Gambar 6.7 Hasil Pengujian Kompatibilitas pada Web Browser Chrome Sistem Operasi macOS Catalina Platform PC.....	96
Gambar 6.8 Hasil Pengujian Kompatibilitas pada Web Browser Safari Sistem Operasi macOS Catalina Platform PC.....	97
Gambar 6.9 Hasil Pengujian Kompatibilitas pada Sistem Operasi Android 9 Platform Mobile.....	98
Gambar 6.10 Hasil Pengujian Kompatibilitas pada Sistem Operasi iOS 13.4 Platform Mobile.....	99



## DAFTAR TABEL

Tabel 2.1 Daftar Tinjauan Pustaka .....	10
Tabel 2.2 Daftar Notasi <i>Use Case Diagram</i> .....	16
Tabel 2.3 Daftar Notasi <i>Activity Diagram</i> .....	17
Tabel 2.4 Daftar Notasi <i>Sequence Diagram</i> .....	19
Tabel 2.5 Daftar Notasi <i>Class Diagram</i> .....	21
Tabel 4.1 Aturan Pengodean.....	32
Tabel 4.2 Pengidentifikasian Aktor <i>As-Is</i> .....	32
Tabel 4.3 Aktifitas Proses Bisnis <i>As-Is</i> Pemilihan Dosen Pembimbing 1.....	33
Tabel 4.4 Analisis Permasalahan pada Proses Bisnis <i>As-Is</i> .....	34
Tabel 4.5 Aktifitas Proses Bisnis <i>To-Be</i> Pemilihan Dosen Pembimbing 1 .....	35
Tabel 4.6 Kebutuhan Sistem .....	37
Tabel 4.7 Deskripsi Diagram <i>Use Case</i> Sistem Rekomendasi Dosen Pembimbing .....	38
Tabel 5.1 <i>Dataset</i> Latar Belakang Skripsi.....	51
Tabel 5.2 <i>Case Folding</i> Latar Belakang.....	56
Tabel 5.3 Penghapusan Tanda Baca pada Latar Belakang.....	57
Tabel 5.4 <i>Stemming</i> pada Latar Belakang.....	59
Tabel 5.5 Penghapusan <i>Stopword</i> pada Latar Belakang.....	60
Tabel 5.6 Tokenisasi pada Latar Belakang.....	62
Tabel 5.7 Nilai <i>Term Frequency</i> , <i>Document Frequency</i> , dan <i>Inverse Document Frequency</i> .....	64
Tabel 5.8 Bobot TF-IDF Setiap <i>Term</i> Pada Data Latar Belakang .....	68
Tabel 5.9 Kriteria <i>Weighted Product</i> untuk Sistem Rekomendasi Dosen Pembimbing .....	73
Tabel 5.10 Matriks Intensitas Kriteria Menggunakan <i>Pairwise Comparison</i> .....	74
Tabel 5.11 Angka <i>Random Index</i> .....	75
Tabel 5.12 Kriteria dan Beban Kriteria <i>Weighted Product</i> .....	76
Tabel 5.13 Aturan Pembobotan Kesesuaian KJFD .....	76
Tabel 5.14 Aturan Pembobotan Jurusan Dosen .....	76
Tabel 5.15 Aturan Pembobotan Gelar Dosen .....	77
Tabel 6.1 Hasil Pengujian Akurasi SVM, BM25, dan WP Dengan Data Uji 15%....	85
Tabel 6.2 Hasil Pengujian Akurasi SVM, BM25, dan WP Dengan Data Uji 20%....	86





## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Mahasiswa adalah sebutan yang diberikan kepada orang yang menempuh pendidikan di perguruan tinggi. Sebagai persyaratan kelulusan, mahasiswa wajib menyelesaikan sebuah tugas akhir. Untuk jenjang sarjana, tugas akhir yang wajib diselesaikan bernama skripsi. Topik skripsi diambil berdasarkan mata kuliah pilihan yang pernah diikuti oleh mahasiswa tersebut. Skripsi diselesaikan dengan bantuan oleh sekurang-kurangnya satu dosen pembimbing. Setiap dosen pembimbing memiliki keahlian dan spesialisasi pada bidang tertentu, serta *track record* skripsi yang telah dibimbing.

Dalam proses pengerjaan skripsi, dosen pembimbing berfungsi sebagai motivator, pendamping, serta pemberi arahan bagi mahasiswa yang sedang mengerjakan skripsi. Oleh karena hal tersebut, peran dosen pembimbing dalam proses pengerjaan skripsi menjadi krusial. Berdasarkan hasil penelitian yang dilakukan oleh Lestari (2012), bahwa ada hubungan positif yang signifikan antara harapan/ekspektansi mahasiswa terhadap dosen pembimbing dengan motivasi dalam menyelesaikan skripsi. Sehingga hubungan antara dosen pembimbing dengan mahasiswa bimbingan menjadi sangat penting.

Pada Fakultas Ilmu Komputer Universitas Brawijaya (Filkom UB), mahasiswa dapat memilih calon dosen pembimbing dengan cara pendekatan personal atau dengan melalui rekomendasi. Pendekatan personal yang dimaksud adalah: (1) mahasiswa dapat memilih dosen yang menyediakan proyek dosen, (2) mahasiswa dapat memilih dosen yang menjadi pengajar pada mata kuliah yang pernah diikuti, (3) mahasiswa dapat memilih dosen berdasarkan pengalaman seminar atau magang. Sesuai alur skripsi pada panduan pengerjaan skripsi mahasiswa Filkom UB(2018), rekomendasi dapat diperoleh mahasiswa melalui kepala program studi atau koordinator kelompok jabatan fungsional dosen berdasarkan topik yang ingin diangkat. Namun, dikarenakan terbatasnya jumlah koordinator kelompok fungsional dosen dan kepala prodi, tidak semua mahasiswa dapat memohon rekomendasi kepada mereka. Sehingga pemilihan dosen pembimbing didominasi dengan pendekatan personal. Pengambilan keputusan pemilihan dosen pembimbing berdasarkan pendekatan personal umumnya didominasi oleh intuisi mahasiswa. Hal ini menimbulkan resiko terjadinya pemilihan calon dosen pembimbing yang tidak sesuai dengan materi/topik skripsi. Ketidaksesuaian tersebut mengakibatkan pengerjaan skripsi yang kurang efektif dan membuka kemungkinan untuk pemilihan calon dosen baru yang lebih menguasai topik/materi.

Untuk menjawab permasalahan ini, dibutuhkan sebuah sistem yang dapat membantu koordinator kelompok jabatan fungsional dosen dan kepala program studi dalam memberikan dosen pembimbing. Dosen menggunakan latar belakang pada proposal sebagai acuan untuk mengetahui topik dari calon skripsi mahasiswa. Oleh karena hal tersebut, dibutuhkan sistem yang dapat membantu



untuk mengetahui kelompok bidang KJFD dan memberi rekomendasi dosen pembimbing berdasarkan latar belakang pada proposal skripsi.

Penelitian berkaitan dengan pengelompokan tema/bidang skripsi telah dilakukan oleh Oman Somantri, Slamet Wiyono, dan Dairoh(2016). Penelitian tersebut menggunakan *k-means* dan *support vector machine*(SVM) untuk memproses judul pada prosposal skripsi menjadi tema dari skripsi tersebut. Penelitian ini menghasilkan bahwa akurasi yang didapatkan pada model SVM sebesar 85,38% dan SVM + *k-means* sebesar 86,21%. Penelitian ini berhasil menjawab permasalahan pengelompokan tema/bidang skripsi menggunakan data judul jurnal.

Penelitian berkaitan sistem rekomendasi dosen pembimbing telah dilakukan oleh Anak Agung Bagus Arisetiawan, Indriati, dan Dian Eka Ratnawati. Penelitian tersebut menggunakan data dokumen judul skripsi di bidang komputasi cerdas sebagai bahan penelitian. Data penelitian pada penelitian tersebut kemudian diolah menggunakan metode BM25. Metode BM25 adalah metode *text mining* yang digunakan untuk pemeringkatan set dokumen berdasarkan *term query* yang muncul pada setiap koleksi dokumen(Arisetiawan, Indriati dan Ratnawati, 2019). Penelitian ini berhasil menjawab permasalahan dosen pembimbing pada bidang komputasi cerdas.

Berdasarkan penjabaran sebelumnya, maka diperlukan penelitian untuk mengolah latar belakang proposal skripsi menjadi KJFD dan dosen pembimbing yang sesuai. Klasifikasi KJFD dilakukan menggunakan model *Support Vector Machine*. Perekomendasi dosen pembimbing dilakukan berdasarkan beberapa kriteria, sehingga perekomendasi menggunakan metode *Weighted Product*. Diharapkan sistem yang akan dibangun dapat mempermudah koordinator KJFD dan kepala program studi untuk memberikan rekomendasi dosen ke mahasiswa sehingga semakin banyak mahasiswa yang mempertimbangkan dosen dengan menggunakan rekomendasi. Penelitian yang dilakukan berjudul "**Sistem Rekomendasi Dosen Pembimbing Berdasarkan Latar Belakang Menggunakan Metode *Multi-Class Support Vector Machine* dan *Weighted Product***".

### 1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang dapat diambil sebagai berikut:

1. Berapa tingkat akurasi klasifikasi KJFD menggunakan algoritme *Multi-class Support Vector Machine*?
2. Berapa tingkat akurasi rekomendasi dosen pembimbing menggunakan algoritme *Weighted Product*?
3. Bagaimana hasil pengujian sistem pada sistem rekomendasi dosen pembimbing?



## 1.2 Tujuan

Berdasarkan masalah yang telah dirumuskan sebelumnya, maka tujuan dari penulisan skripsi ini antara lain:

1. Mengetahui tingkat akurasi klasifikasi Kelompok Jabatan Fungsional Dosen menggunakan algoritme *Multi-class Support Vector Machine*
2. Mengetahui tingkat akurasi rekomendasi dosen pembimbing menggunakan algoritme *Weighted Product*.
3. Mengetahui hasil pengujian sistem pada sistem rekomendasi dosen pembimbing.

## 1.3 Manfaat

Manfaat yang dapat diberikan oleh penelitian ini adalah:

1. Mengetahui tingkat akurasi klasifikasi KJFD berdasarkan latar belakang dari skripsi menggunakan pendekatan *Multi-Class Support Vector Machine*.
2. Memberikan rekomendasi dosen pembimbing kepada mahasiswa yang ingin mengambil skripsi dengan menggunakan data latar belakang proposal skripsi yang telah dibuat dan mengurangi tingkat keraguan dalam memilih dosen pembimbing.

## 1.4 Batasan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka dibutuhkan batasan masalah agar penelitian tidak melebar. Batasan masalah pada penelitian ini adalah:

1. Data latar belakang skripsi yang digunakan adalah latar belakang skripsi mahasiswa yang melaksanakan yudisium pada tahun 2018 hingga Agustus 2019.
2. Data latar belakang skripsi yang digunakan pada proses *mining* adalah skripsi mahasiswa yang berasal dari Program Studi Sistem Informasi Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.

## 1.5 Sistematika Pembahasan

Keseluruhan dari penulisan penelitian ini dibagi menjadi delapan bab yang dibagi tiap pokok bahasan. Adapun sistematika dari penelitian ini sebagai berikut:

### BAB 1 : PENDAHULUAN

Berisi tentang latar belakang penelitian, rumusan masalah, tujuan penulisan, manfaat penulisan, dan batasan masalah yang akan diselesaikan.

### BAB 2 : LANDASAN KEPUSTAKAAN

Berisi tentang dasar teori, penelitian, dan referensi yang mendasari penelitian ini.



### **BAB 3 : METODOLOGI PENELITIAN**

Berisi tentang penjelasan langkah-langkah/metodologi dalam melaksanakan kegiatan penelitian.

### **BAB 4 : REKAYASA KEBUTUHAN**

Berisi penggalan dan pemodelan kebutuhan sebagai acuan dalam pembangunan sistem yang akan dilaksanakan.

### **BAB 5 : PERANCANGAN DAN IMPLEMENTASI**

Berisi rancangan pembangunan aplikasi sesuai rekayasa kebutuhan yang telah dilakukan serta proses implementasi yang dilakukan untuk membangun sistem.

### **BAB 6 : PENGUJIAN**

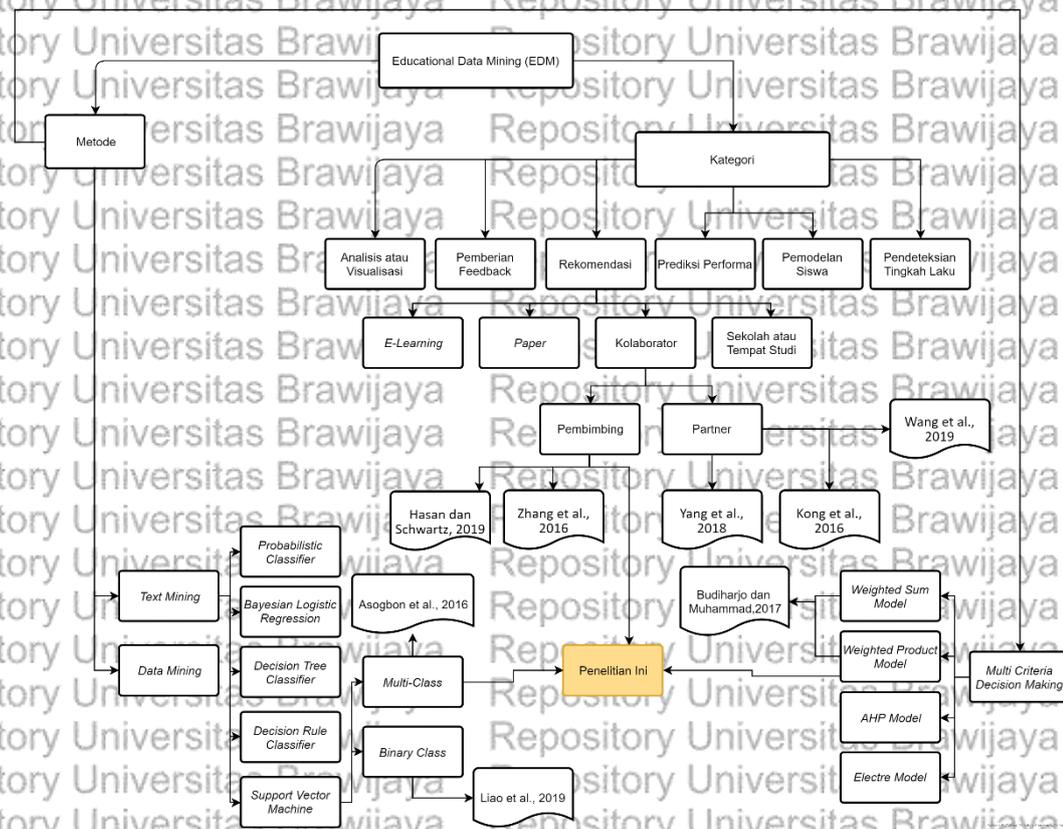
Berisi tentang pengujian mengenai sistem yang telah dibangun, pengujian dibagi menjadi pengujian akurasi dan pengujian pengguna.

### **BAB 7 : PENUTUP**

Berisi kesimpulan yang ditarik berdasarkan hasil penelitian yang telah dilakukan, serta saran untuk penelitian selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Tinjauan Pustaka



Gambar 2.1 Posisi Penelitian Terhadap Penelitian Penelitian Sebelumnya

#### 2.1.1 Tinjauan Pustaka Pertama

Penelitian yang berjudul “A Nearest Neighbor Based Personal Rank Algorithm for Collaborator Recommendation” yang dibuat oleh Yang, C. et al. dan dipublikasikan pada tahun 2018 menjadi tinjauan pustaka keempat pada penelitian ini. Jumlah peneliti yang besar dan perbedaan arah penelitian menjadikan dasar pentingnya pencarian kolaborator yang sesuai, hal ini menjadi dasar permasalahan yang diangkat pada penelitian tinjauan ini. Penelitian tinjauan ini mencoba mengembangkan algoritme *Random Walk with Restart*. Penelitian tinjauan ini mengusulkan algoritme baru bernama *Nearest Neighbor based Random Walk*. Hasil dari penelitian tinjauan ini menemukan bahwa algoritme yang diusulkan dapat memberi garansi bahwa kandidat yang diberikan memiliki jarak sosial yang lebih dekat dengan pengguna (Yang et al., 2018).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan, dimana kedua penelitian berusaha memberikan rekomendasi seorang kolaborator pada penelitian. Dosen pembimbing dapat juga disebut sebagai kolaborator, dikarenakan pada penelitian skripsi seorang dosen pembimbing ikut berkontribusi pandangan memberikan arahan pada skripsi.



Namun, pada penelitian tinjauan ini algoritma yang digunakan untuk memberikan rekomendasi adalah *Random Walk with Restart*, sementara penelitian ini menggunakan gabungan antar *Support Vector Model* dan *Weighted Product*.

### 2.1.2 Tinjauan Pustaka Kedua

Penelitian yang berjudul "*Exploiting publication contents and collaboration networks for collaborator recommendation*" menjadi tinjauan pustaka kelima pada penelitian ini. Penelitian tinjauan ini dibuat oleh Kong, X. et al. dan dipublikasikan pada tahun 2016. Penelitian tinjauan ini mengangkat permasalahan dimana para peneliti ingin mencoba berkolaborasi dengan peneliti yang tidak terhubung pada jaringan sosial mereka. Penelitian tinjauan ini mencoba menjawab permasalahan dikarenakan belum ada penelitian mengenai pemberian rekomendasi peneliti yang memberikan opsi peneliti di luar jaringan hubungan sosial pengguna. Penelitian tinjauan ini mencoba mengembangkan hasil penelitian *Collaboration Networks for Collaborators Recommendation* (CCRec) dengan membuat model *hybrid* antara CCRec dengan *topic clustering* dan *random walk*. Hasil dari penelitian tinjauan ini menyimpulkan bahwa model yang diusulkan berhasil mengalahkan performa model yang lainnya pada skor *precision*, *recall*, dan *F1* (Kong et al., 2016).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan. Diantaranya pada penelitian yang sedang dilakukan terdapat kemungkinan bahwa dosen yang direkomendasikan tidak pernah mengajar atau berhubungan dengan mahasiswa tersebut. Penelitian tinjauan ini mampu menjawab dengan menghubungkan topik yang diambil sama seperti pendekatan penelitian skripsi. Namun, penelitian tinjauan mencoba menjawab masalah dengan membangun *framework* TNERec sementara penelitian skripsi mencoba menjawab masalah dengan memproses latar belakang menjadi topik skripsi dan merekomendasikan dosen menggunakan *weighted product*.

### 2.1.3 Tinjauan Pustaka Ketiga

Penelitian yang berjudul "*Sustainable Collaborator Recommendation Based on Conference Closure*" menjadi tinjauan pustaka keenam pada penelitian ini. Penelitian tinjauan ini dibuat oleh Wang, W. et al. dan dipublikasikan pada tahun 2019. Penelitian tinjauan ini mengangkat permasalahan bahwa belum ada penelitian sebelumnya yang mencoba mempelajari bagaimana cara menemukan dan merekomendasikan kolaborator yang berkelanjutan. Penelitian tinjauan ini mencoba menjawab permasalahan dengan mengusulkan *sustainable collaborator recommendation* (SCORE). SCORE mencoba mengembangkan metode rekomendasi sebelumnya dengan menghubungkan peneliti yang mengikuti konferensi ilmiah yang sama. Hasil dari penelitian tinjauan ini menyimpulkan bahwa sistem SCORE dapat memberikan rekomendasi peneliti yang lebih beragam sementara membatasi penurunan dari akurasi (Wang et al., 2019).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan, yaitu kedua penelitian mencoba memberikan rekomendasi



kolaborator penelitian dimana dosen pembimbing dapat disebut sebagai kolaborator dalam penelitian skripsi. Namun, penelitian ini berfokus pada hubungan kolaborasi dimana pihak-pihak yang tergabung memiliki kedekatan terutama pada konferensi yang diikuti.

#### 2.1.4 Tinjauan Pustaka Keempat

Penelitian yang berjudul “*A research analytics framework-supported recommendation approach for supervisor selection*” menjadi tinjauan pustaka ketujuh pada penelitian ini. Penelitian tinjauan ini dibuat oleh Zhang, M. et al. dan dipublikasikan pada tahun 2016. Penelitian tinjauan ini mengangkat permasalahan dimana ketidaksesuaian pembimbing dengan siswa berdampak pada performa akademik siswa. Penelitian tinjauan ini mengusulkan pendekatan yang berpusat pada siswa, pendekatan tersebut dinamakan *Research Analytics Framework for Education* (RAF-E). Pendekatan ini menggabungkan teknik *content-based* dan *collaborative filtering*. Hasil dari penelitian tinjauan ini menyimpulkan bahwa metode RAF-E memiliki performa yang jauh lebih unggul dibandingkan dengan semua pendekatan yang menjadi dasar (*baseline*) perbandingan (Zhang et al., 2016).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan. Diantaranya kedua penelitian membahas mengenai perkomendasi dosen pembimbing secara spesifik. Namun, penelitian ini mencoba menggabungkan *social connection* dan publikasi.

#### 2.1.5 Tinjauan Pustaka Kelima

Penelitian yang berjudul “*A Multi-criteria Decision Support System for Ph.D. Supervisor Selection: A Hybrid Approach*” menjadi tinjauan pustaka kedelapan. Penelitian tinjauan ini mengangkat permasalahan pentingnya pemilihan pembimbing yang sesuai untuk pengerjaan penelitian disertasi. Penelitian tinjauan ini mencoba menjawab permasalahan dengan memberikan sistem rekomendasi pembimbing disertasi. Metode yang digunakan untuk memecahkan masalah pada penelitian tinjauan ini adalah dengan menguji perbandingan performa antara BM25 dengan TF-IDF. Hasil penelitian tinjauan ini adalah menemukan bahwa rekomendasi yang menggunakan BM25 memiliki performa yang lebih baik diperbandingkan dengan menggunakan TF-IDF (Hasan dan Schwartz, 2019).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan. Diantaranya kedua penelitian membahas mengenai perkomendasi dosen pembimbing secara spesifik dan menggunakan *Multi Attribute Decision Making* sebagai pendekatan untuk memberikan rekomendasi dosen. Penelitian tinjauan ini juga membahas mengenai pendekatan mengenai metode TF-IDF dan BM25. Namun, penelitian ini menggabungkan pendekatan tersebut dengan pendekatan *fuzzy*.



### 2.1.6 Tinjauan Pustaka Keenam

Penelitian yang berjudul “*A Robust Machine Learning Technique to Predict Low-performing Students*” menjadi tinjauan pustaka keenam. Penelitian ini mengangkat mengenai prediksi performa murid yang rendah menggunakan teknik *machine learning*. Penelitian ini menggunakan data *presage* dan data *in-progress*. Data *presage* adalah data objektif yang sudah ada dan tersedia, contohnya: nilai (GPA) saat sekolah, gender, umur, status sosioekonomik. Data *in-progress* adalah data saat proses berjalan, contohnya: nilai ujian tengah semester atau waktu pengumpulan tugas. Penelitian ini menggunakan metode *Binary Support Vector Machine* untuk membentuk model *machine learning*. Hasil dari penelitian ini adalah metode *Binary Support Vector Machine* dapat digunakan untuk memberikan prediksi yang dapat digunakan untuk mengurangi resiko siswa dengan performa rendah (Liao et al., 2019).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan. Diantaranya kedua penelitian menggunakan metode *Support Vector Machine* untuk melakukan prediksi kelas. Namun, penelitian tinjauan ini menggunakan *Binary Support Vector Machine* yang hanya membahas kelas berjumlah biner (dua).

### 2.1.7 Tinjauan Pustaka Ketujuh

Penelitian yang berjudul “*A Multi-class Support Vector Machine Approach for Students Academic Performance Prediction*” menjadi tinjauan pustaka ketujuh. Penelitian ini mengangkat mengenai prediksi performa murid dengan menggunakan pendekatan *Multi-class Support Vector Machine*. Penelitian ini membagi performa akademik siswa menjadi 5 kelompok, yaitu: *Distinction, Good, Average, Weak, dan Hapless*. Penelitian ini menggunakan data informasi demografi, informasi personal, informasi akademik, dan informasi keluarga. Penelitian ini menghasilkan akurasi sebesar 72,25% dalam memprediksi performa siswa (Asogbon et al., 2016).

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan. Diantaranya kedua penelitian menggunakan metode *Multi-class Support Vector Machine* untuk melakukan prediksi kelas. Namun, penelitian tinjauan menggunakan studi kasus performa siswa sementara penelitian ini menggunakan studi kasus bidang topik skripsi.

### 2.1.8 Tinjauan Pustaka Kedelapan

Penelitian yang berjudul “*Comparison of Weighted Sum Model and Multi Attribute Decision Making Weighted Product Methods in Selecting the Best Elementary School in Indonesia*” menjadi tinjauan pustaka kedelapan. Penelitian ini mengangkat mengenai pemilihan sekolah dasar yang terbaik bagi calon siswa dengan memperbandingkan pendekatan *Weighted Sum Model* dan *Weighted Product*. Data yang digunakan diantaranya: nilai siswa, tingkat disiplin, tingkah laku, tingkat Pendidikan guru, sertifikasi, jumlah laboratorium, jumlah perpustakaan, dan kelengkapan sekolah. Penelitian ini menghasilkan bahwa baik



*Weighted Sum Model* ataupun *Weighted Product* dapat digunakan untuk pengambilan keputusan sekolah terbaik (Budiharjo dan Muhammad, 2017);

Terdapat persamaan antara penelitian tinjauan ini dengan penelitian yang sedang dilakukan. Diantaranya kedua penelitian menggunakan metode pengambilan keputusan *multi-attribute weighted product* untuk melakukan prediksi. Namun, penelitian ini membahas mengenai penilaian sekolah terbaik dan memperbandingkan metode *weighted product* dengan *weighted sum model*. Daftar tinjauan pustaka dapat dilihat pada tabel 2.1.



Tabel 2.1 Daftar Tinjauan Pustaka

No	Judul Jurnal	Nama Peneliti	Tujuan	Metode	Hasil
1	<i>A Nearest Neighbor Based Personal Rank Algorithm for Collaborator Recommendation</i> (2018), <i>2018 15th International Conference on Service Systems and Service Management (ICSSSM)</i>	C. Yang, T. Liu, L. Liu et al.	Menjawab permasalahan jumlah peneliti dan perbedaan tema penelitian pada sistem rekomendasi kolaborator	<i>Random Walk Algorithm Based on Nearest Neighbor (NRRW)</i>	Metode NRRW berhasil memastikan kandidat yang lebih pendek sehingga memiliki performa lebih dari algoritme <i>Random Walk</i>
2	<i>Exploiting publication contents and collaboration networks for collaborator recommendation</i> (2016), <i>PLoS ONE</i>	X. Kong, H. Jiang, Z. Yang et al.	Menggabungkan model rekomendasi berdasarkan <i>content-based</i> dengan <i>collaboration network</i>	<i>CCRec</i>	Menggunakan data dalam 500 domains, <i>CCRec</i> mendapatkan <i>precision, recall</i> , dan skor <i>F1</i> terbaik.
3	<i>Sustainable Collaborator Recommendation Based on Conference Closure</i> (2019), <i>IEEE Transactions on Computational Social Systems</i>	W. Wang, J. Liu, Z. Yang et al.	Menjawab pentingnya kebutuhan kolaborator yang berkelanjutan	<i>Sustainable Collaborator Recommendation (SCORE)</i>	SCORE berhasil merekomendasikan kolaborator yang berkelanjutan sementara membatasi penurunan akurasi
4	<i>A research analytics framework-supported recommendation approach for supervisor selection</i>	M. Zhang, J. Ma, Z. Liu et al.	Menjawab permasalahan dimana ketidaksesuaian pembimbing dengan	<i>Research Analytics Framework for Education (RAF-E)</i>	RAF-E memiliki performa yang jauh lebih tinggi dibandingkan dengan metode acuan ( <i>baseline</i> )

No	Judul Jurnal	Nama Peneliti	Tujuan	Metode	Hasil
	(2016), <i>British Journal of Educational Technology</i>		siswa berdampak pada performa akademik siswa		
5	<i>A Multi-criteria Decision Support System for Ph.D. Supervisor Selection: A Hybrid Approach</i> (2019), <i>Proceedings of the 52nd Hawaii International Conference on System Sciences</i>	M. Hasan, D. Schwartz	Menjawab pentingnya pembimbing disertasi dengan memberikan sistem yang dapat memberi rekomendasi	Fuzzy dan BM25	Penggunaan Fuzzy dan BM25 memberikan hasil yang lebih baik diperbandingkan dengan TF/IDF
6	<i>A Robust Machine Learning Technique to Predict Low-performing Students</i>	Liao et al.	Sistem dapat memprediksi murid dengan resiko secara akurat dan koleksi data ringan sehingga pengajar ingin menggunakan sistem	<i>Binary Support Vector Machine</i>	Akurasi AUC meningkat kurva ROC menghalus sehingga dapat disimpulkan sistem dapat digunakan untuk memprediksi murid dengan performa rendah
7	<i>A Multi-class Support Vector Machine Approach for Students Academic Performance Prediction</i>	Asogbon et al.	Sistem dapat memprediksi performa siswa dan meletakkan siswa pada program fakultas yang sesuai	<i>Multi-class Support Vector Machine</i>	Mendapatkan akurasi sebesar 72,25% untuk seluruh tipe kelas.

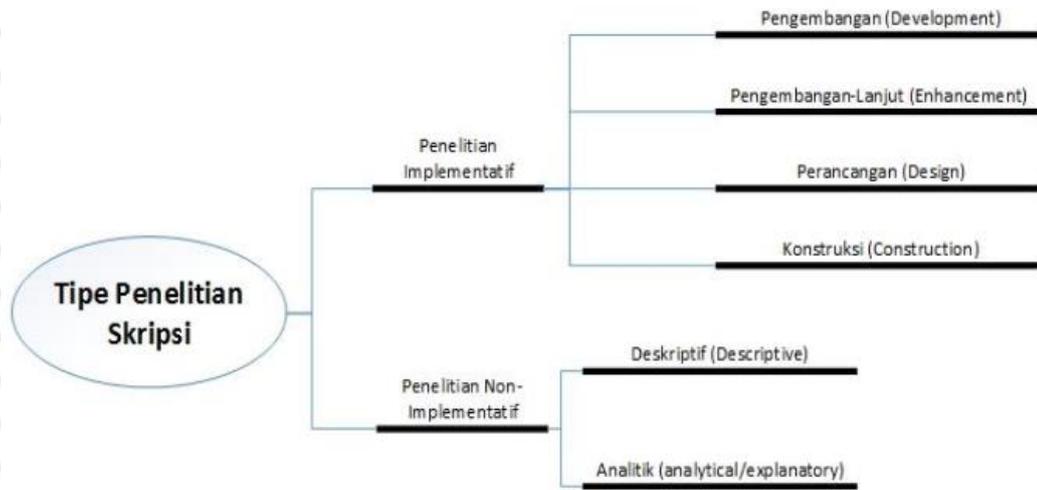
No	Judul Jurnal	Nama Peneliti	Tujuan	Metode	Hasil
8	<i>Comparison of Weighted Sum Model and Multi Attribute Decision Making Weighted Product Methods in Selecting the Best Elementary School in Indonesia</i>	A. Budiharjo dan A. Muhammad	Menguji apakah metode <i>Weighted Sum Model</i> dan <i>Weighted Product</i> dapat digunakan untuk pembobotan dalam membantu pemilihan sekolah dasar	<i>Weighted Sum Model</i> dan <i>Weighted Product</i>	<i>Weighted Sum Model</i> dan <i>Weighted Product</i> dapat digunakan untuk membantu pemilihan sekolah dasar.



## 2.2 Skripsi

Skripsi adalah bentuk karya ilmiah dikarang oleh mahasiswa yang mengikuti pendidikan program sarjana(S1) sebagai wujud dari kajian pengetahuan dan/atau penerapan teknologi berdasarkan kaidah ilmiah dalam keminatan ilmu atau studi yang telah dipelajari sebagai hasil pembelajaran(FILKOM UB, 2018). Skripsi menjadi salah satu persyaratan bagi mahasiswa untuk dapat menuntaskan Pendidikan program sarjana(S1). Sebagai salah satu persyaratan, skripsi perlu memenuhi kriteria-kriteria karya tulis ilmiah dan kesesuaian dengan standar program studi.

Pada Fakultas Ilmu Komputer Universitas Brawijaya, tipe penelitian skripsi dibagi menjadi dua, yaitu implementatif dan nonimplementatif(FILKOM UB, 2018). Penelitian implementatif adalah tipe penelitian skripsi yang menghasilkan produk/artefak utama sebagai solusi terhadap permasalahan yang diangkat. Penelitian nonimplementatif menitikberatkan pada investigasi terhadap fenomena atau situasi tertentu, atau analisis terhadap hubungan antar fenomena yang sedang dikaji untuk kemudian menghasilkan hasil investigasi atau hasil analisis ilmiah sebagai produk/artefak utamanya. Pembagian tipe penelitian skripsi pada Fakultas Ilmu Komputer Universitas Brawijaya digambarkan pada gambar 2.2.



**Gambar 2.2 Tipe Penelitian Skripsi Fakultas Ilmu Komputer Universitas Brawijaya**

Sumber: FILKOM UB (2018)

## 2.3 Dosen Pembimbing Skripsi

Dosen pembimbing skripsi adalah seseorang dengan kualifikasi tertentu yang ditugaskan untuk mengarahkan mahasiswa tingkat sarjana dalam menyelesaikan tugas akhir berupa skripsi, baik yang berasal dari kalangan dosen maupun profesional. Untuk pengerjaan skripsi pada Fakultas Ilmu Komputer



Brawijaya, maksimal pembimbing yang diperbolehkan untuk membimbing seorang mahasiswa berjumlah 2 (FILKOM UB, 2018).

Pembagian kewenangan pembimbing skripsi secara umum dibagi pada aspek materi skripsi dan aspek penulisan skripsi. Untuk skripsi dengan pembimbing berjumlah satu, maka pembimbing bertugas secara penuh untuk mengarahkan pada aspek materi dan aspek penulisan. Untuk skripsi dengan pembimbing berjumlah dua, maka pembimbing 1 bertugas untuk mengarahkan secara dominan pada aspek materi, sementara pembimbing 2 bertugas untuk mengarahkan secara dominan pada aspek penulisan (FILKOM UB, 2018).

## 2.4 Jabatan Fungsional Dosen

Menurut Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Nomor 17 Tahun 2013 tentang Jabatan Fungsional Dosen dan Angka Kreditnya, Jabatan Fungsional Dosen adalah kedudukan yang menunjukkan tugas, tanggung jawab, wewenang, dan hak seseorang Dosen dalam suatu satuan pendidikan tinggi yang dalam pelaksanaannya didasarkan pada keahlian tertentu serta bersifat mandiri. Beberapa Dosen yang memiliki Jabatan Fungsional Dosen dikumpulkan untuk membentuk Kelompok Jabatan Fungsional Dosen dan memiliki seorang Koordinator Kelompok Jabatan Fungsional Dosen.

Pada Fakultas Ilmu Komputer Universitas Brawijaya, Koordinator Jabatan Fungsional Dosen (KKJFD) memiliki kewenangan dalam proses birokrasi skripsi. Mahasiswa dapat berkonsultasi topik/judul skripsi dengan KKJFD. Jikalau mahasiswa berdiskusi dengan KKJFD, maka KKJFD berkewajiban mengarahkan mahasiswa kepada Calon Pembimbing yang sesuai/bersedia (FILKOM UB, 2018).

## 2.5 Unified Modelling Language (UML)

*Unified Modelling Language* atau UML adalah standar bahasa yang umum digunakan dalam melakukan pemodelan kerangka *blueprint* dari sebuah perangkat lunak. *System Designer* menciptakan diagram UML untuk membantu pengembang perangkat lunak membangun perangkat lunak (Pressman dan Maxim, 2015). Diagram UML yang menjadi standar memudahkan proses pemindahtanganan perangkat lunak dikarenakan pengembang atau pihak selanjutnya yang berhubungan dengan perangkat lunak dapat mengerti alur dan isi perangkat lunak tanpa meneliti setiap baris kode. Diagram UML juga memiliki keuntungan untuk memperjelas arsitektur perangkat lunak dan memudahkan hubungan perangkat lunak dengan perangkat lainnya.

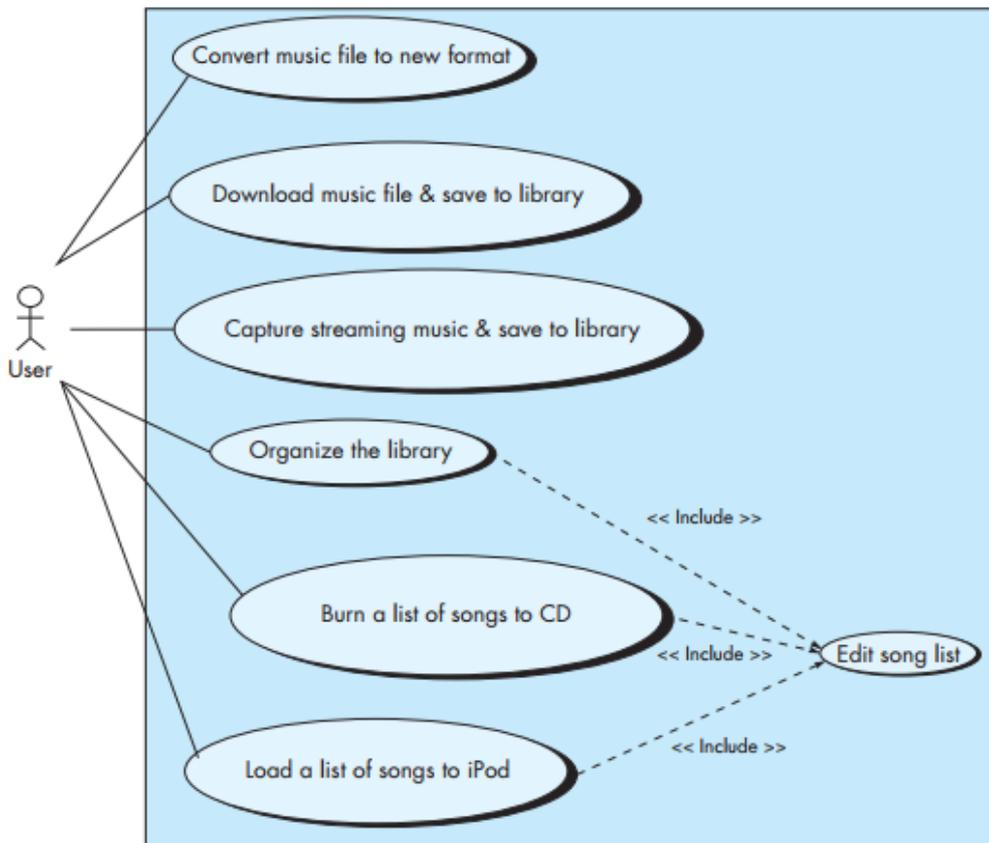
UML dikembangkan oleh *Grady Boch*, *Jim Rumbaugh*, dan *Ivar Jacobson* pada pertengahan tahun 1990 (Pressman dan Maxim, 2015). Pada tahun 1997, UML diajukan kepada lembaga Object Management Group, konsorsium nonprofit yang bekerja untuk memelihara spesifikasi yang digunakan pada bidang industri komputer. Standar yang digunakan saat ini adalah UML 2.5.1, standar ini telah termasuk ke dalam standar ISO. UML Diagram berisi kumpulan diagram, namun pada penelitian ini berfokus pada diagram *use case*, *activity*, *sequence*, dan *class*.



Jumlah diagram yang terdapat pada UML memberikan ruang bagi *software designer* untuk menampilkan segala aspek penting sistem (Pressman dan Maxim, 2015).

## 2.6 Use Case Diagram

*Use case diagram* digunakan sebagai alat bantu dalam menentukan fungsionalitas dan fitur dari sebuah perangkat lunak berdasarkan sudut pandang pengguna. *Use case* mendeskripsikan bagaimana pengguna berinteraksi dengan sistem dengan mendefinisikan langkah yang diperlukan untuk mencapai sebuah tujuan. *Use case diagram* menjadi gambaran atas semua *use case* dan bagaimana setiap *use case* saling berelasi. *Use case diagram* memberikan gambaran besar atas fungsionalitas dari sebuah sistem (Pressman dan Maxim, 2015). Berikut contoh pemodelan *use case diagram* pada gambar 2.3.



Gambar 2.3 Contoh Penggunaan *Use Case Diagram* Aplikasi Daring Toko Musik Digital

Sumber: Pressman dan Maxim (2015)

Menurut Pressman dan Maxim (2015), pada *use case diagram* terdapat beberapa notasi yang diperlukan untuk membuat diagram, diantaranya dijelaskan pada tabel 2.2.



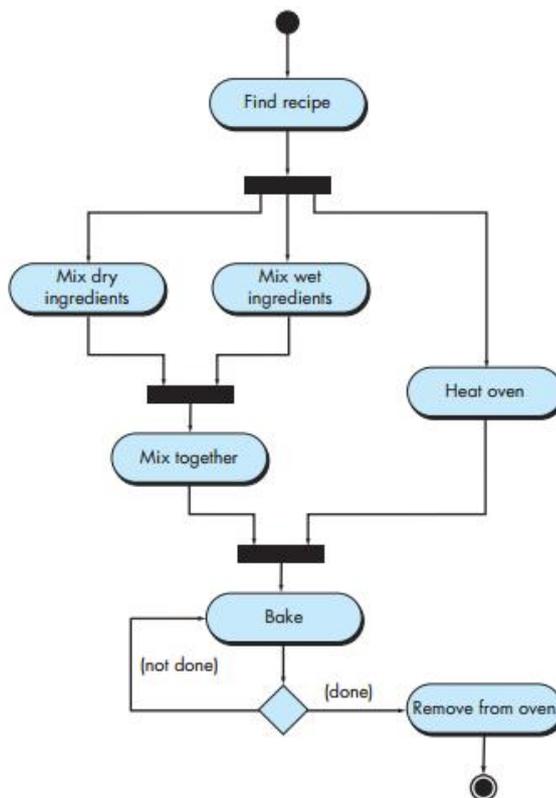
Tabel 2.2 Daftar Notasi *Use Case Diagram*

Nama	Notasi	Fungsi
Aktor	 User	Aktor adalah satu kategori pengguna atau elemen interaksi lain yang memiliki hubungan dengan <i>use case</i>
<i>Use Case</i>	 Use Case	<i>Use case</i> melambangkan fungsional atau kebutuhan yang dibutuhkan oleh pengguna.
Garis Penghubung Asosiasi		Garis penghubung berfungsi untuk menampilkan hubungan antar aktor dengan <i>use case</i>
Garis Penghubung Include	 <<Include>>	Garis penghubung <i>include</i> berfungsi untuk menghubungkan <i>use case</i> duplikasi yang termasuk ke dalam proses <i>use case</i> lain.

Sumber: Pressman dan Maxim (2015)

### 2.7 Activity Diagram

*Activity diagram* menjelaskan tingkah laku dinamis dari sebuah sistem atau bagian sistem melalui alur kontrol antara aksi-aksi yang terdapat pada sistem. *Activity diagram* memiliki kemiripan dengan *flowchart*. Namun, *activity diagram* dapat menunjukkan alur yang berjalan secara paralel dan bersamaan (Pressman dan Maxim, 2015). Berikut contoh pemodelan *activity diagram* pada gambar 2.4.



Gambar 2.4 Contoh Penggunaan Activity Diagram Cara Memasak Kue

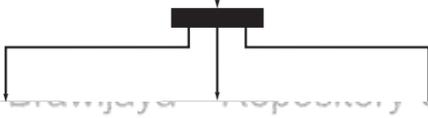
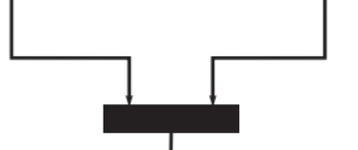
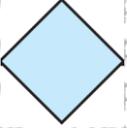
Sumber: Pressman dan Maxim (2015)

Menurut Pressman dan Maxim(2015), pada *activity diagram* terdapat beberapa notasi yang diperlukan untuk membuat diagram, diantaranya dijelaskan pada tabel 2.3.

Tabel 2.3 Daftar Notasi Activity Diagram

Nama	Notasi	Fungsi
Action Node		Action Node merepresentasikan sebuah task yang dilakukan oleh sistem perangkat lunak
Anak Panah		Anak panah merepresentasikan alur kontrol antar satu action node dengan yang lain
Initial & End Node		Initial node merepresentasikan titik mulai dari sebuah aktivitas. End node merepresentasikan titik

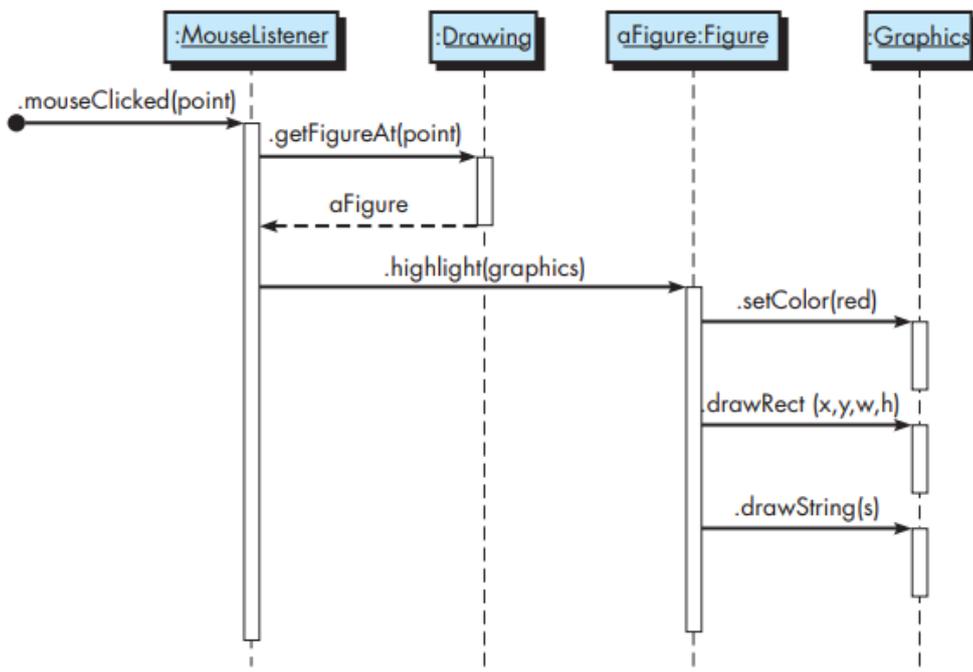


Nama	Notasi	Fungsi
		akhir dari sebuah aktivitas.
Fork		Fork merepresentasikan pemisahan alur aktivitas menjadi dua atau lebih alur aktivitas yang dijalankan secara paralel dan bersamaan
Join		Join merepresentasikan penggabungan dua atau lebih alur aktivitas yang berjalan secara paralel menjadi satu.
Decision Node		Decision node berhubungan dengan cabang alur kontrol berdasarkan sebuah kondisi

Sumber: Pressman dan Maxim (2015)

## 2.8 Sequence Diagram

Sequence diagram digunakan untuk menampilkan komunikasi dinamis antar objek pada saat eksekusi sebuah task. Penggunaan sequence diagram bertujuan untuk menunjukkan interaksi dalam satu use case atau satu skenario dari sebuah sistem perangkat lunak (Pressman dan Maxim, 2015). Berikut contoh pemodelan sequence diagram pada gambar 2.5.



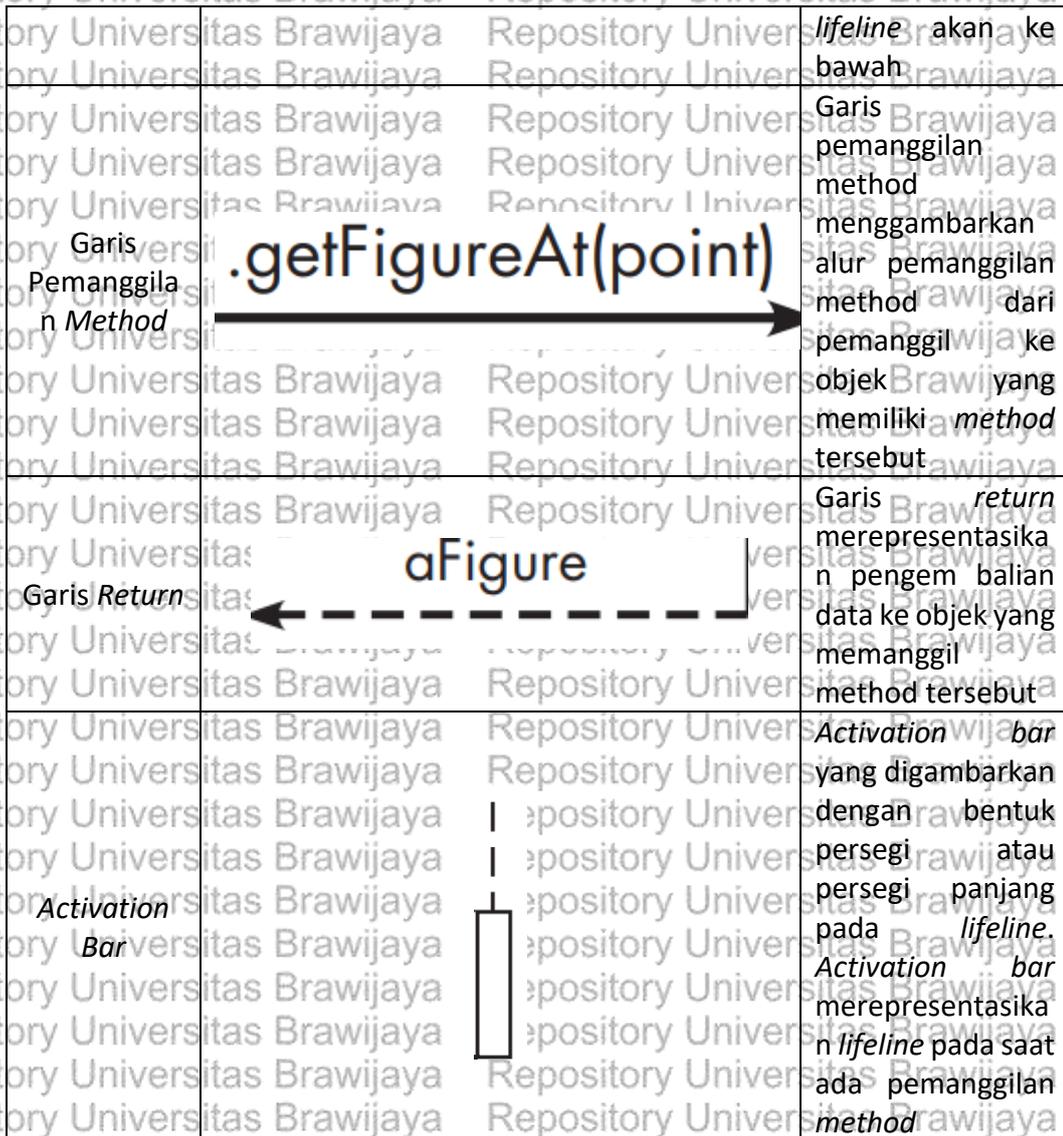
Gambar 2.4 Contoh Penggunaan Sequence Diagram Menampilkan Persegi

Sumber: Pressman dan Maxim (2015)

Menurut Pressman dan Maxim(2015), pada *activity diagram* terdapat beberapa notasi yang diperlukan untuk membuat diagram, diantaranya dijelaskan pada tabel 2.5.

Tabel 2.4 Daftar Notasi Sequence Diagram

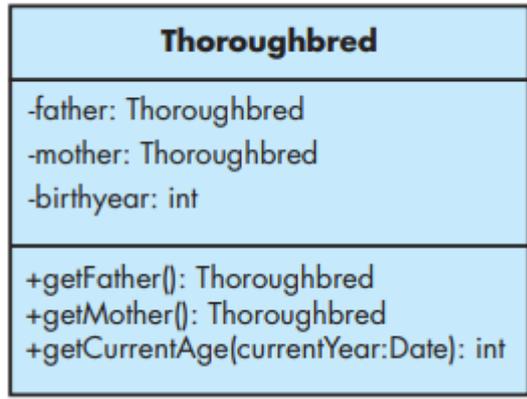
Nama	Notasi	Fungsi
Object/Entitiy		Entity atau object merepresentasikan objek yang berkaitan dengan use-case yang dibahas.
Lifeline		Lifeline digambarkan dengan garis putus-putus vertikal merepresentasikan waktu kehidupan objek. Semakin lama waktu yang digunakan, maka semakin panjang



Sumber: Pressman dan Maxim. (2015)

## 2.9 Class Diagram

*Class diagram* memberikan pandangan statis atau struktural dari sebuah sistem. *Class diagram* digunakan untuk memodelkan *class* termasuk atribut, operasi, dan hubungan dan asosiasi dengan *class* lainnya. *Class diagram* pada dasarnya berbentuk kotak yang dibagi bagi menjadi beberapa bagian horizontal (Pressman dan Maxim, 2015). Berikut contoh pemodelan *sequence diagram* pada gambar 2.6.



Gambar 2.6 Contoh Penggunaan *Class Diagram* menampilkan kelas Thoroughbred

Sumber: Pressman dan Maxim (2015)

Menurut Pressman dan Maxim(2015), pada *class diagram* terdapat beberapa notasi yang diperlukan untuk membuat diagram, diantaranya dijelaskan pada tabel 2.5

Tabel 2.5 Daftar Notasi *Class Diagram*

Nama	Notasi	Fungsi
Nama Kelas		Nama Kelas berada di bagian paling atas pada kelas. Ditulis dengan menggunakan huruf tebal
Attribute		Attribute atau variable pada kelas berada pada bagian tengah. Diawali dengan kode visibilitas kemudian nama variabel, titik dua, dan diakhiri dengan tipe data variabel
Method		Method pada kelas berada pada bagian bawah. Diawali dengan kode visibilitas kemudian nama method dan attribut, titik dua dan



Nama	Notasi	Fungsi
Garis Inheritance		diakhiri dengan tipe data <i>return</i> . Garis <i>inheritance</i> merepresentasikan alur <i>inheritance</i> /turunan dari sebuah <i>class</i> . Class yang meng- <i>inherit</i> atau meng- <i>extend</i> menunjuk kelas yang di- <i>inherit</i> .

Sumber: Pressman dan Maxim (2015)

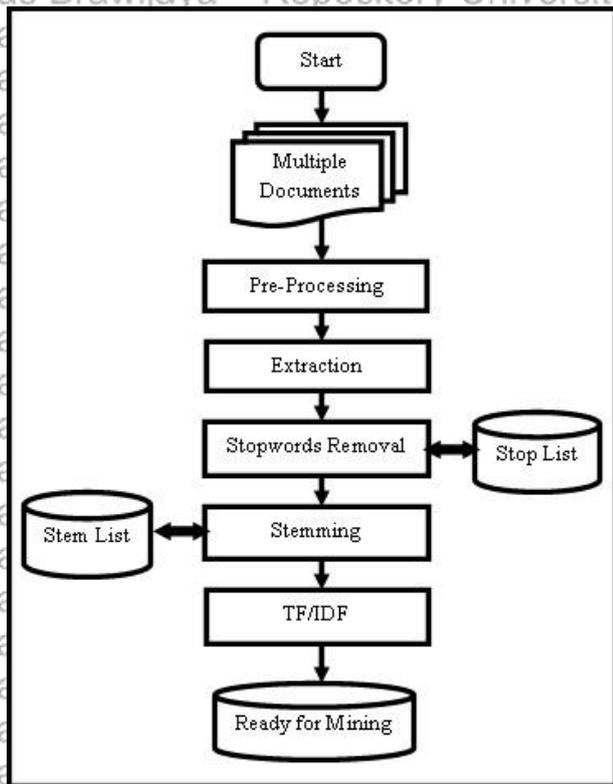
### 2.10 Text Mining

*Text mining* adalah proses ekstraksi informasi yang berguna dari sumber data yang bersifat data tekstual yang tidak terstruktur. *Text mining* juga dapat digunakan untuk mencari *pattern* menarik yang tidak ditemukan pada *database records* melainkan pada data tekstual yang tidak terstruktur. *Text mining* didasari oleh *data mining* dalam mencari *pattern* menarik dalam kumpulan data. Dikarenakan pada *data mining* diasumsikan data telah tersimpan dalam format yang terstruktur, maka *preprocessing* berfokus pada *scrubbing* dan *normalisasi data*. Namun, pada *text mining* kegiatan *preprocessing* berfokus pada identifikasi dan ekstraksi dari fitur yang representatif untuk dokumen dengan *natural language* (Feldman dan Sanger, 2007).

*Text mining* mengikuti model umum yang diberikan oleh *data mining* sehingga secara kasar dapat dibagi menjadi empat area utama, yaitu: *preprocessing task*, *core mining operations*, *presentation layer components* dan *browsing functionality*, dan *refinement technique*. *Preprocessing task* berisi segala kegiatan, proses dan *methods* yang dibutuhkan dalam menyiapkan data untuk operasi utama *text mining*. *Core Mining Operations* adalah inti dari sistem *text mining*, berisi *pattern discovery*, *trend analysis*, dan *incremental knowledge discovery*. *Presentation Layer Components* berisi GUI dan fungsionalitas *pattern browsing* yang dimana sebagai akses terhadap bahasa *query* (Feldman dan Sanger, 2007).

### 2.11 Text Preprocessing

*Text preprocessing* adalah langkah awal untuk mengubah data mentah menjadi data yang dapat dipahami. Data nyata pada umumnya memiliki kekurangan dan tidak konsisten. *Text preprocessing* untuk proses *text mining* terbagi menjadi beberapa tahapan, diantaranya: *case folding*, penghapusan tanda baca, *stemming*, penghapusan *stopword*, tokenisasi (Vijayarani, Ilamathi dan Nithya, 2015). Berikut alur *data preprocessing* pada gambar 2.7.



Gambar 2.7 Alur Data Preprocessing

Sumber: Vijayarani, et al. (2015)

2.11.1 Case Folding

Case Folding adalah bagian proses dari preprocessing untuk menormalisasikan data. Case Folding adalah sebuah teknik untuk mengecilkan semua huruf menjadi huruf kecil agar perbandingan case-insensitive dapat dilakukan (Shubham Bauskar et al., 2019). Jikalau terdapat dua kata yang memiliki arti dan konteks yang sama namun memiliki format yang berbeda, maka algoritme penyamaan akan melihat kedua kata tersebut sebagai entitas yang berbeda. Sebagai contoh kata “Makan” dan “makan” memiliki makna yang sama. Namun, dikarenakan penggunaan huruf besar pada huruf “M”, sistem memisahkan kedua kata tersebut. Oleh karena hal tersebut, case folding mengubah kata “Makan” menjadi “makan” agar sistem melihat dua kata tersebut sebagai entitas yang sama.

Case folding memiliki kerugian dimana beberapa kata benda spesifik diturunkan dari kata benda umum (Shubham Bauskar et al., 2019). Kata benda spesifik yang memiliki arti lain sering ditemui pada judul dan penamaan sesuatu. Sebagai contoh, nama perusahaan “Gudang Garam” terbagi atas kata umum “gudang” dan “garam” yang sama sekali tidak ada hubungannya secara konteks dengan perusahaan Gudang Garam sebagai perusahaan pengolah tembakau.

2.11.2 Penghapusan Tanda Baca

Tanda baca atau karakter spesial tidak memiliki signfiikansi saat content-based matching dan harus dihilangkan sebelum mengaplikasikan algoritme



klasifikasi (Shubham Bauskar et al., 2019). Meskipun secara konteks penghilangan karakter spesial seperti "\$" dapat menghilangkan konteks, penghilangan karakter "\$" tidak menghilangkan secara konten. Sebagai contoh pada kalimat "ayam tersebut berharga \$5" dengan penghapusan tanda baca menjadi "ayam tersebut berharga 5". Konteks mata uang US Dollar menghilang, namun secara konten tetap 5. Tanda baca yang tetap dipertahankan dapat menjadi data *outlier*.

### 2.11.3 Stemming

*Stemming* adalah metode yang digunakan untuk mengidentifikasi kata dasar (batang) dari sebuah kata (Vijayarani, Ilamathi dan Nithya, 2015). Proses *stemming* berusaha menghapus imbuhan yang tertempel pada sebuah kata dasar. Tujuan dari *stemming* adalah untuk menghapus beragam akhiran/imbuhan, mengurangi jumlah kata, menyamakan kata dasar secara akurat, dan untuk mengurangi waktu dan penggunaan memori. Sebagai contoh terdapat dua kata yaitu "memakan" dan "dimakan", konteks kedua kata tersebut berhubungan dengan kegiatan makan. Dengan melakukan *stemming*, kedua kata tersebut diubah menjadi "makan" sehingga kedua kata tersebut dibaca oleh sistem sebagai satu entitas yang sama.

Pada bahasa pemrograman Python, terdapat sebuah *library* yang dapat digunakan untuk melakukan *stemming* bernama PySastrawi. PySastrawi dapat diakses melalui <https://github.com/har07/PySastrawi/>. PySastrawi merupakan hasil porting dari *library* PHP bernama Sastrawi. *Stemming* menggunakan Sastrawi berpatok pada kata dasar yang terdaftar pada situs <http://kateglo.com/>.

### 2.11.4 Penghapusan Stopword

*Stopword* adalah kata yang paling umum pada sebuah bahasa. Pada Bahasa Indonesia, kata yang dapat termasuk *stopword* sebagai contoh adalah: "yang", "untuk", "dan", "ke", dan lain lain. Kata-kata *stopword* tidak memiliki kekuatan pembeda dan harus dihilangkan sebelum membentuk model *bag of words*, dikarenakan *stopword* mengambil lebih banyak memori dan meningkatkan waktu proses tanpa memberikan relevansi yang cukup (Shubham Bauskar et al., 2019).

Pada *library* Sastrawi terdapat fitur penghapusan *stopword* yang sesuai dengan bahasa Indonesia. *Library* Sastrawi menyediakan 139 kata *stopword* pada Bahasa Indonesia yang dapat dilihat pada halaman web berikut: <https://github.com/sastrawi/sastrawi/blob/master/src/Sastrawi/StopWordRemover/StopWordRemoverFactory.php>. *Library* Sastrawi menyediakan fitur untuk menambah *stopword* sesuai dengan kebutuhan, sebagai contoh untuk bidang IT kata "internet" berada pada mayoritas penulisan skripsi dan dapat ditambahkan ke dalam *stopword*.

### 2.12 Pembobotan Kata

*Classifier* umum dan algoritme *learning* tidak dapat secara langsung memproses dokumen teks pada bentuk aslinya. Oleh karena itu, pada tahapan *preprocessing* dokumen diubah menjadi representasi yang lebih dapat diatur.



Secara umum, dokumen direpresentasikan menggunakan *feature vector*. Sebuah *feature* adalah sebuah *entity* tanpa struktur internal (Feldman dan Sanger, 2007).

Model *bag-of-words* yang paling umum menggunakan semua kata pada sebuah dokumen menjadi fitur, sehingga dimensi dari *feature space* sama dengan jumlah kata yang berbeda pada semua dokumen. Metode pemberian bobot pada *feature* beragam. Salah satu metode pemberian bobot adalah *Term frequency and inverse document* (Feldman dan Sanger, 2007).

### 2.12.1 Term Frequency and Inverse Document Frequency

*Term frequency and inverse document frequency* (TF-IDF) adalah sebuah statistik numerik yang menunjukkan seberapa penting sebuah kata pada sebuah dokumen di kumpulan dokumen. TF-IDF umumnya digunakan sebagai faktor pembobotan pada *information retrieval* dan *text mining*. TF-IDF adalah hasil dari dua statistik, yaitu *term frequency* dan *inverse document frequency*. Frekuensi kata (*term frequency*) dalam sebuah dokumen dihitung untuk mendapatkan jumlah kata pada dokumen. *Inverse document frequency* adalah sebuah pembobotan statistik yang digunakan untuk mengukur pentingnya sebuah *term* dalam kumpulan dokumen teks (Vijayarani, Ilamathi dan Nithya, 2015). Berikut rumus perhitungan *Term Frequency* pada persamaan 2.1.

$$tf - idf_{t,d} = (1 + \log \left( \frac{N}{df_t} \right)) \quad (2.1)$$

### 2.13 Klasifikasi

Pembahasan paling umum dalam menganalisa data yang kompleks adalah klasifikasi atau kategorisasi dari elemen. Secara abstrak, klasifikasi adalah sebuah *task* untuk mengklasifikasikan *instance data* yang ada untuk menjadi beberapa kumpulan set kategori yang sudah terspesifikasikan terlebih dahulu. Terdapat sebuah *task* bernama *text categorization* (TC) dimana diberikan sebuah kumpulan kategori (subyek atau topik) dan sebuah kumpulan dokumen teks dan proses pencarian topik yang sesuai dari setiap dokumen (Feldman dan Sanger, 2007).

Dengan berkembangnya *Artificial Intelligence*, terdapat dua pendekatan utama pada *text categorization/classification*. Yang pertama adalah pendekatan *knowledge engineering* dimana pengetahuan ahli mengenai kategori/kelas secara langsung dikodekan ke dalam sistem. Yang kedua adalah pendekatan *machine learning* (ML) dimana sebuah proses induktif umum dalam membentuk sebuah *classifier* dengan mempelajari dari kumpulan set contoh yang telah terklasifikasi. Sebagian besar penilitan pada *categorization/classification* terkonsentrasikan pada pendekatan ML yang hanya membutuhkan sebuah kumpulan dari *training instance* yang telah diberi kelas secara manual (Feldman dan Sanger, 2007). Salah satu algoritme pendekatan ML pada masalah klasifikasi adalah *Support Vector Machine*.



## 2.14 Multi-Class Support Vector Machine

Algoritme *Support Vector Machine* (SVM) sangat cepat dan efektif untuk menjawab permasalahan klasifikasi teks. Dalam bahasa geometris, sebuah *classifier* SVM biner dapat dilihat sebagai sebuah *hyperplane* dalam sebuah *feature space* membagi antara titik yang merepresentasikan *instance* positif dari kategori dari titik yang merepresentasikan *instance* negatif. SVM *hyperplanes* sepenuhnya ditentukan oleh secara relatif beberapa himpunan bagian kecil dari *training instance*, dimana hal tersebut disebut *support vectors* (Feldman dan Sanger, 2007).

Dikarenakan beragam kompleksitas yang ada, sebuah solusi langsung untuk menjawab masalah *multiclass* menggunakan formula *single* SVM secara umum dihindari. Metode-metode yang populer digunakan untuk menjawab permasalahan *multiclass* adalah metode *one-versus-all* dan *one-versus-one*. Pada permasalahan *multiclass*,  $M$  akan menunjukkan jumlah dari kelas dan  $\omega_i$ ,  $i = 1, \dots, M$  dimana  $M$  akan menunjukkan jumlah kelas  $M$  (Duan dan Keerthi, 2005).

Metode *one-versus-all* membentuk  $M$  klasifier biner. Keluaran klasifier ke- $i$  fungsi  $\rho_i$  dilatih untuk mengambil contoh dari  $\omega_i$  sebagai positif dan contoh dari kelas lain sebagai negatif. Sebagai contoh baru  $x$ , *one-versus-all* menempatkan hal tersebut kepada kelas dengan nilai  $\rho_i$  terbesar (Duan dan Keerthi, 2005).

Metode *one-versus-one* membentuk sebuah klasifier biner untuk setiap pasang kelas yang berbeda sampai semua klasifier biner  $M(M-1)/2$  terbentuk. Klasifier biner  $C_{ij}$  dilatih untuk mengambil contoh dari  $\omega_i$  sebagai positif dan contoh dari  $\omega_j$  sebagai negatif (Duan dan Keerthi, 2005).

LIBSVM adalah sebuah *library* untuk untuk *Support Vector machine*. LIBSVM mendukung task-task sebagai berikut: (1) SVC, *support vector classification* (dua kelas dan multi kelas); (2) SVR, *support vector regression*; (3) SVM satu kelas. Penggunaan LIBSVM secara umumnya melibatkan dua tahapan: pertama, *training* sebuah kumpulan data untuk mendapatkan sebuah model dan kedua, menggunakan model yang telah didapatkan untuk memprediksi informasi dari sebuah kumpulan data *testing* (Chang dan Lin, 2011).

## 2.15 BM25

Pendekatan Okapi *best matching* 25 (BM25) didasarkan pada *framework* pengambilan probabilistic yang dikembangkan pada tahun 1970-an hingga 1980-an oleh Robertson et al. BM25 digunakan untuk menghitung kesamaan antara sebuah *query* pengguna dan sebuah dokumen (Goyal, Behera dan McGinnity, 2013). Berikut rumus perhitungan nilai BM25 dapat dilihat pada persamaan 2.2.

$$rsv_q = \sum_{t \in q} \log \left( \frac{N}{df_t} \right) \cdot \frac{(k_1 + 1) \cdot tf_{td}}{k_1 \cdot \left( 1 - b + b \cdot \left( \frac{L_d}{L_{avg}} \right) \right) + tf_{td}} \quad (2.2)$$

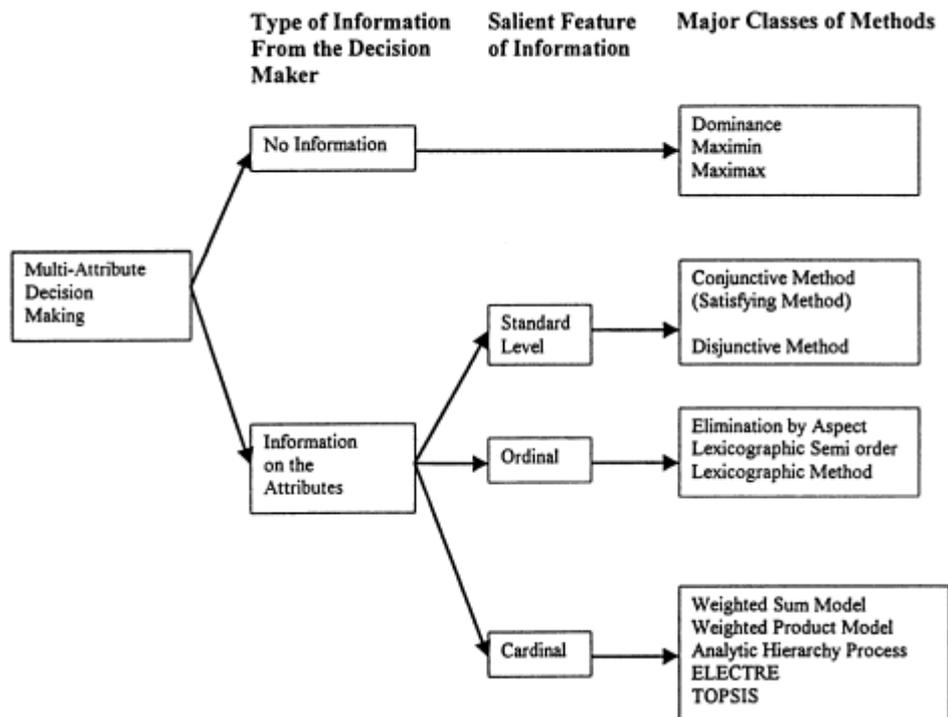
Dokumen yang sangat panjang *query term* tidak mendapatkan nilai yang pantas oleh BM25 dan dokumen yang sangat tersebut dapat mendapatkan

penalize(hukuman poin) yang berlebihan. BM25L dan BM25+ menjadi pengembangan sederhana namun efektif dari BM25 untuk menjawab permasalahan dimana BM25 memiliki tendensi untuk memberi hukuman poin berlebihan pada dokumen yang sangat panjang(Lv dan Zhai, 2011).

### 2.16 Multi-Criteria Decision Making

Multi-criteria decision making (MCDM) adalah sebuah cabang *decision making* (pembuatan keputusan) yang paling terkenal. Secara umum, MCDM dan *multi-attribute decision making* (MADM) digunakan untuk menjelaskan model yang sama. MCDM berfokus pada permasalahan dengan ruang keputusan yang berlainan. Pada masalah tersebut, kumpulan dari alternatif keputusan telah ditentukan terlebih dahulu(Triantaphyllou, 2000).

Meskipun metode MCDM bermacam-macam, banyak dari mereka memiliki beberapa aspek umum yaitu istilah alternatif dan atribut. Alternatif merepresentasikan pilihan berbeda yang tersedia untuk pengambil keputusan. Atribut yang dapat direferensikan sebagai *goals* atau kriteria keputusan, merepresentasikan dimensi-dimensi yang berbeda dimana dapat dilihat(Triantaphyllou, 2000). Taksonomi dari metode-metode pada MCDM dapat dilihat pada gambar 2.8.



Gambar 2.8 Taksonomi dari Metode MCDM

Sumber: Triantaphyllou (2000)



### 2.16.1 Weighted Product Model (WPM)

Weighted product model (WPM) adalah bagian dari metode *Multi Criteria Decision Making* (MCDM). WPM digunakan untuk pengambilan keputusan disaat data pada atribut bersifat angka kardinal. Angka kardinal adalah angka yang digunakan dalam perhitungan atau penjumlahan. WPM merupakan proses modifikasi WSM untuk menjawab kelemahan dari WSM (Triantaphyllou, 2000).

WPM memperbandingkan setiap alternatif dengan memperkalikan rasio pada setiap kriteria yang dimiliki. Keuntungan metode WPM adalah dapat menggunakan nilai relatif. Salah satu pendekatan alternative WPM adalah dengan hanya menggunakan *product* tanpa rasio (Triantaphyllou, 2000). Berikut rumus pendekatan WPM menggunakan *product* tanpa rasio pada persamaan 2.3

$$P(A_k) = \prod_{j=1}^n (a_{kj})^{w_j} \quad (2.3)$$

### 2.17 Pairwise Comparisons

*Pairwise comparison* adalah metode yang digunakan untuk menganalisa beberapa populasi secara berpasangan untuk mendeterminasi apakah mereka berbeda secara signifikan antar yang lain. *Pairwise comparison* juga dapat menjadi metode dimana pembuat keputusan memperbandingkan antar satu kriteria (atribut) dengan yang lainnya untuk menentukan tingkat preferensi untuk setiap pasang kriteria. Penggunaan skala ordinal (1-9) diadopsi untuk membantu menentukan nilai preferensi dari sebuah atribut dengan yang lainnya (Odu, 2019).

Menurut Odu (2019), penentuan beban atribut berdasarkan metode *pairwise comparison* memiliki tiga Langkah utama yang dapat diimplementasikan sebagai berikut:

1. Membentuk sebuah matriks dengan memperbandingkan kriteria. Nilai intensitas digunakan untuk mengisi matriks seperti (1,3,5,7,9) digunakan untuk merepresentasikan kepentingan yang sama, kepentingan cukup atau sedang diperbandingkan dengan yang lain, kepentingan yang lebih kuat, kepentingan yang sangat kuat, dan kepentingan yang ekstrim. Dimana skala ordinal 2,4,6, dan 8 digunakan untuk nilai antara antar angka ganjil;
2. Menghitung beban kriteria yang juga dapat disebut nilai prioritas atau *principal eigenvector*. Hal ini dapat dilakukan cara menjumlahkan nilai pada setiap kolom kemudian dibagi dengan setiap elemen berdasarkan total kolom, dan membagi jumlah dari nilai yang sudah ternormalisasi pada setiap baris dengan jumlah kriteria.
3. Memperkirakan konsistensi untuk analisis sensitivitas yang disebut dengan rasio konsistensi (CR). Jikalau rasio konsistensi lebih rendah dari 0,1 maka rasio tersebut mengindikasikan tingkat konsistensi layak pada *pairwise comparison*, akan tetapi jikalau CR lebih besar dari 0,1 hal tersebut menunjukkan bahwa *pairwise comparison* yang telah dilakukan tidak konsisten saat pengambilan keputusan.



## 2.18 Pengujian Sistem

Pengujian sistem adalah proses melatih atau menguji sebuah sistem atau komponen sistem secara manual ataupun otomatis dengan maksud untuk memverifikasi apakah hal tersebut sesuai dengan kebutuhan yang dispesifikasikan. (Chopra, 2018) Salah satu strategi pengujian adalah dengan melakukan *Verification and Validation (V&V)*. Verifikasi mengacu pada kumpulan *task* yang dimana memastikan bahwa sebuah perangkat lunak telah mengimplementasikan sebuah fungsi spesifik secara benar. Validasi mengacu pada kumpulan *task* berbeda yang dimana memastikan bahwa perangkat lunak yang telah dibangun dapat ditarik kepada kebutuhan pengguna. Pengujian kompatibilitas pada *client side* berfokus pada kompatibilitas aplikasi yang dengan menguji sistem operasi, *platform*, *web browser* yang digunakan untuk penggunaan sistem (Pressman dan Maxim, 2015).

### 2.18.1 Pengujian Unit

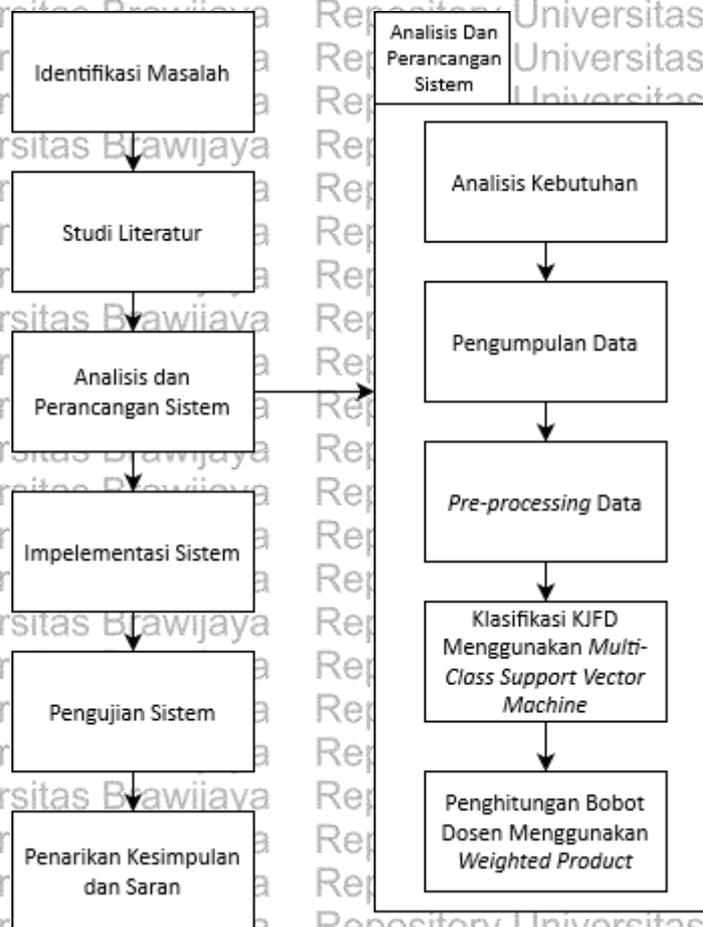
Pengujian unit memfokuskan verifikasi pada bagian terkecil dari sebuah perangkat lunak, komponen perangkat lunak atau modul perangkat lunak. Pengujian unit berfokus pada logika proses internal dan struktur data di dalam batasan sebuah komponen. Pada perangkat lunak yang berorientasi objek (OO), konsep sebuah unit berubah. Sebuah kelas yang terenkapsulasi secara umum menjadi focus dari pengujian unit. Namun, operasi atau metode dalam sebuah kelas menjadi unit terkecil yang dapat diuji. Sehingga pengujian unit pada perangkat lunak berorientasi objek (OO) didorong oleh operasi/metode yang dienkapsulasi di dalam sebuah kelas (Pressman dan Maxim, 2015).

### 2.18.2 Pengujian Performa

Pengujian performa digunakan untuk menemukan permasalahan performa yang dapat disebabkan oleh kurangnya sumberdaya *server-side*, *bandwidth* jaringan yang tidak sesuai, kapabilitas basis data yang tidak mencukupi, kesalahan atau lemahnya kapabilitas sistem operasi, fungsionalitas aplikasi yang dirancang dengan buruk, dan masalah hardware dan software lainnya yang dapat menyebabkan menurunnya performa *client-server*. Pengujian performa didesain untuk mensimulasikan kondisi pada dunia nyata. Dua pengujian performa yang dapat dilakukan adalah: (1) pengujian *load* untuk memeriksa performa pada kondisi dunia nyata, dan (2) pengujian *stress* memaksa *loading* hingga ke titik puncak untuk mengetahui berapa kapasitas yang dapat dilayani oleh aplikasi (Pressman dan Maxim, 2015).



## BAB 3 METODOLOGI



Gambar 3.1 Diagram Alir Metodologi

### 3.1 Identifikasi Masalah

Pada tahapan identifikasi masalah, peneliti mencari dan mengobservasi kondisi pemilihan dosen pembimbing. Penulis kemudian mengobservasi akibat dari permasalahan yang terjadi di lapangan.

### 3.2 Studi Literatur

Pada tahapan studi pustaka, peneliti mencari dan mengumpulkan referensi yang dapat membantu pelaksanaan penelitian. Referensi dapat ditemukan dari buku, jurnal penelitian, e-book, dan internet. Referensi menjadi sarana untuk mengumpulkan teori, prosedur, dan *tools* yang dapat digunakan dalam melaksanakan penelitian.



### 3.3 Analisis dan Perancangan Sistem

Pada tahapan ini, peneliti menganalisa kebutuhan pengguna yang perlu diterapkan pada sistem yang akan dibangun. Kemudian, peneliti melakukan pengumpulan data latar belakang skripsi. Data yang telah didapatkan akan mendapatkan perlakuan *pre-processing*. Kemudian data tersebut menjadi bahan bagi *text mining* yang dilakukan menggunakan metode *multi-class support vector machine*. Hasil *text mining* digunakan menjadi salah satu atribut kriteria pada *weighted product*.

### 3.4 Implementasi Sistem

Pada tahapan ini, peneliti mengimplementasikan metode *multi-class support vector machine* ke dalam baris kode aplikasi. Pengaplikasian metode *multi-class support vector machine* dan *weighted product* akan diimplementasikan sebagai *web service*. *Web service* akan menerima data latar belakang, memproses, dan mengirimkan data rekomendasi dosen pembimbing ke aplikasi *frontend web*. Aplikasi *frontend web* berguna untuk menampilkan data yang telah diproses oleh *web service*.

### 3.5 Pengujian Sistem dan Analisis

Pada tahapan ini, peneliti melakukan pengujian terhadap hasil penelitian yang telah dilakukan. Pengujian dilakukan kepada dua hal utama, yaitu kepada algoritme yang telah digunakan dan sistem yang telah dibangun. Pengujian algoritme dilakukan untuk menghitung akurasi hasil keluaran dengan hasil yang sesuai dengan kondisi sebenarnya. Pengujian sistem dilakukan untuk mengetahui apakah sistem yang telah dibangun sesuai dengan hasil analisis dan perancangan sebelumnya.

### 3.6 Penarikan Kesimpulan dan Saran

Pada tahapan ini, peneliti melakukan penarikan kesimpulan sesuai dengan hasil dari penelitian ini. Kemudian peneliti memberikan saran yang dapat digunakan untuk pengembangan dan pertimbangan pada penelitian selanjutnya.

## BAB 4 REKAYASA KEBUTUHAN

### 4.1 Aturan Pengodean

Aturan pengodean digunakan untuk menstandarkan penggunaan angka dan kode sehingga meningkatkan *traceability* dalam perancangan. Pada tabel 4.1 dijelaskan aturan pengodean dan contoh penggunaan dari aturan tersebut dimana XX akan digantikan dengan angka.

Tabel 4.1 Aturan Pengodean

Artefak	Aturan Pengodean	Contoh Penggunaan
Data Sampel	SD-XX	SD-01
Proses Bisnis	PB-Jenis-XX	PB-AI-01 atau PB-TB-01
Kebutuhan Sistem	RQ-XX	RQ-01
<i>Use Case</i>	UC-XX	UC-01

### 4.2 Pemodelan Proses Bisnis

Pemodelan proses bisnis dibutuhkan untuk memodelkan aktivitas pemilihan dosen pembimbing skripsi pada Program Studi Sistem Informasi Fakultas Ilmu Komputer UB. Proses bisnis terbagi menjadi dua, yaitu: *as-is* dan *to-be*. Proses bisnis *as-is* menggambarkan mengenai kondisi aktivitas pemilihan dosen pembimbing skripsi yang sedang berjalan atau yang sedang digunakan saat ini. Informasi mengenai alur aktivitas pemilihan dosen pembimbing skripsi *as-is* didapatkan melalui wawancara dengan Kepala Program Studi Sistem Informasi Fakultas Ilmu Komputer UB. Proses bisnis *to-be* menggambarkan mengenai kondisi aktivitas pemilihan dosen pembimbing skripsi setelah hasil penelitian ini diterapkan.

### 4.3 Pengidentifikasian Aktor As-Is

Pengidentifikasian aktor berguna untuk memetakan aktor atau fungsi jabatan apa saja yang terlibat dalam proses pemilihan dosen pembimbing skripsi pada Program Studi Sistem Informasi UB. Hasil dari pengidentifikasian aktor dijelaskan pada tabel 4.2.

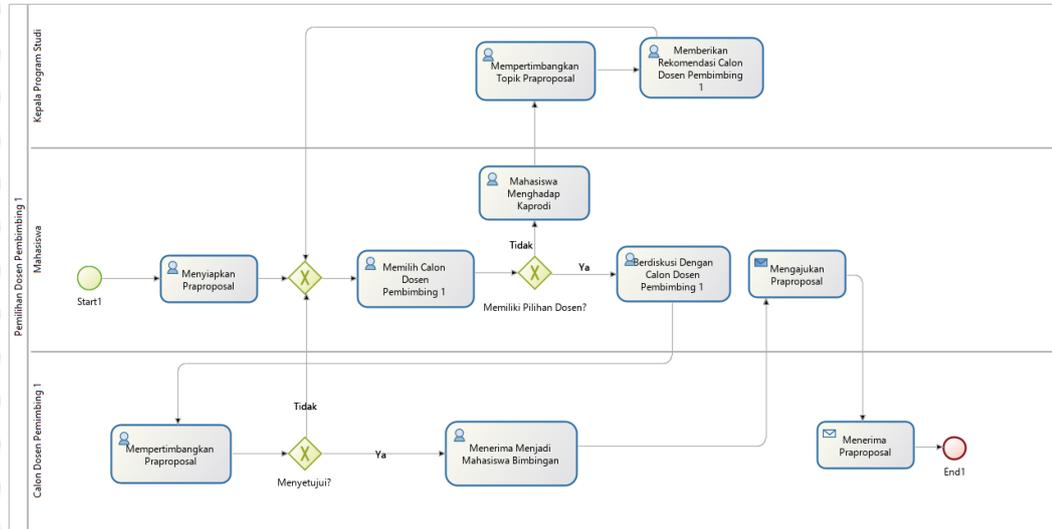
Tabel 4.2 Pengidentifikasian Aktor As-Is

Nama Aktor	Deskripsi
Mahasiswa	Aktor yang berperan memilih dosen pembimbing skripsi
Kepala Program Studi	Aktor yang berperan mengkaji dan menyetujui pilihan dosen pembimbing skripsi
Calon Dosen Pembimbing	Aktor yang menerima <i>request</i> pemilihan dosen pembimbing skripsi



#### 4.4 Proses Bisnis As-Is

Berdasarkan wawancara dengan Kepala Program Studi Sistem Informasi UB, proses pemilihan dosen pembimbing oleh mahasiswa hanya untuk memilih dosen pembimbing 1. Bisnis proses pemilihan dosen pembimbing 1 saat ini (*as-is*) dijelaskan pada gambar 4.1.



**Gambar 4.1 Proses Bisnis As-Is Pemilihan Dosen Pembimbing 1**

Alur proses bisnis pemilihan dosen pembimbing 1 dimulai dengan mahasiswa memilih calon dosen pembimbing yang diinginkan. Mahasiswa menghadap calon dosen pembimbing untuk berdiskusi mengenai rancangan topik skripsi yang ingin diambil oleh mahasiswa tersebut. Jikalau calon dosen pembimbing menyetujui untuk menjadi dosen pembimbing. Setiap aktivitas dijelaskan lebih lanjut pada tabel 4.3.

**Tabel 4.3 Aktifitas Proses Bisnis As-Is Pemilihan Dosen Pembimbing 1**

Kode Aktivitas	Nama Aktor	Aktivitas	Deskripsi
PB-AI-01	Mahasiswa	Menyiapkan Praproposal	Mahasiswa menyiapkan praproposal yang berisi latar belakang dan bidang penelitian
PB-AI-02	Mahasiswa	Memilih Calon Dosen Pembimbing	Mahasiswa memilih dosen pembimbing yang diinginkan
PB-AI-03	Mahasiswa	Berdiskusi Dengan Calon Dosen Pembimbing 1	Mahasiswa berdiskusi dengan calon dosen pembimbing mengenai praproposal yang telah dibuat
PB-AI-04	Calon Dosen Pembimbing 1	Mempertimbangkan Praproposal	Calon dosen pembimbing mempertimbangkan



			kesesuaian praproposal dengan bidang keahlian serta keinginan untuk membimbing
PB-AI-05	Calon Dosen Pembimbing 1	Menerima Menjadi Mahasiswa Bimbingan	Calon Dosen Pembimbing menerima mahasiswa mahasiswanya
PB-AI-06	Mahasiswa	Mengajukan Praproposal	Mahasiswa mengajukan praproposal untuk diterima secara resmi oleh calon dosen pembimbing
PB-AI-07	Calon Dosen Pembimbing 1	Menerima Praproposal	Calon Dosen Pembimbing menerima praproposal mahasiswa secara resmi
PB-AI-08	Mahasiswa	Mahasiswa Menghadap Kaprodi	Jikalau mahasiswa ditolak oleh calon dosen pembimbing, mahasiswa dapat menghadap kaprodi untuk berdiskusi mengenai praproposal
PB-AI-09	Kepala Prodi	Mempertimbangkan Topik Praproposal	Kaprodi mempertimbangkan kualitas dan isi dari praproposal
PB-AI-10	Kepala Prodi	Memberikan Rekomendasi Calon Dosen Pembimbing 1	Kaprodi memberikan rekomendasi calon dosen pembimbing berdasarkan pertimbangan dari praproposal

#### 4.4.1 Analisis Permasalahan Proses Bisnis As-Is

Hasil wawancara dengan Kepala Program Studi Sistem Informasi UB kemudian dianalisis permasalahannya. Analisis permasalahan dilakukan untuk menemukan permasalahan yang terjadi pada kondisi saat ini sehingga dapat dirumuskan solusi untuk menghadapi permasalahan tersebut. Hasil analisis permasalahan proses bisnis *as-is* dijelaskan pada tabel 4.4.

**Tabel 4.4 Analisis Permasalahan pada Proses Bisnis As-Is**

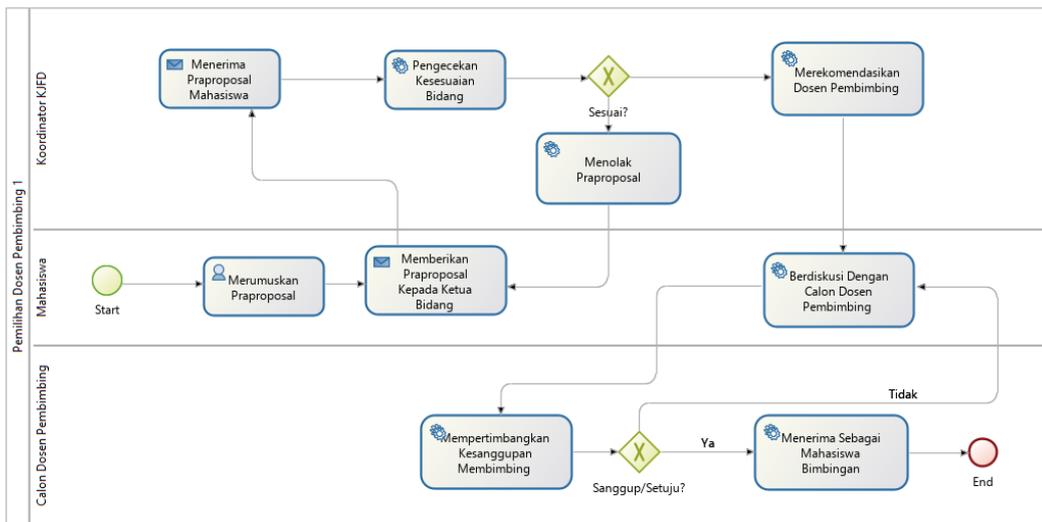
Permasalahan	Pihak Terkait	Dampak	Solusi
Mahasiswa dapat memilih dosen sesuai dengan keinginannya	Mahasiswa dan Calon Dosen Pembimbing	Ada kemungkinan terjadinya ketidaksesuaian pemilihan dosen pembimbing	Diperlukan adanya pengecekan apakah pemilihan dosen



		dengan topik skripsi	tersebut sesuai dengan topik skripsi
Mahasiswa tidak memiliki target calon dosen pembimbing	Mahasiswa	Mahasiswa mengalami kebingungan dalam memilih dosen pembimbing	Diperlukan adanya sebuah sistem yang memberikan rekomendasi calon dosen pembimbing

### 4.5 Proses Bisnis To-Be

Proses bisnis *to-be* merupakan proses bisnis yang diusulkan untuk menjawab permasalahan yang terdapat pada proses bisnis *as-is*. Proses bisnis *to-be* dirumuskan berdasarkan analisis permasalahan yang terdapat pada proses bisnis *as-is*. Proses bisnis *to-be* dimodelkan menggunakan notasi BPMN yang digambarkan pada gambar 4.2.



Gambar 4.2 Proses Bisnis To-Be Pemilihan Dosen Pembimbing 1

Alur proses bisnis *to-be* pemilihan dosen pembimbing 1 dimulai oleh mahasiswa merumuskan praproposal yang kemudian akan diberikan kepada Kepala Bidang, untuk dilakukan pengecekan apakah topik praproposal tersebut sesuai dengan bidang tersebut. Hal ini dilakukan untuk menjawab dampak permasalahan yang mengakibatkan terjadinya mahasiswa yang dapat memilih dosen yang tidak sesuai dengan bidang skripsi yang akan diambil. Kemudian terdapat langkah pemberian rekomendasi oleh kepala bidang. Hal ini dilakukan untuk menjawab permasalahan mahasiswa yang tidak mengetahui dosen yang sesuai dengan topik skripsi yang akan diambil oleh mahasiswa tersebut. Setiap aktivitas dijelaskan lebih lanjut pada Tabel 4.5

Tabel 4.5 Aktifitas Proses Bisnis To-Be Pemilihan Dosen Pembimbing 1



Kode Aktivitas	Nama Aktor	Aktivitas	Deskripsi
PB-TB-01	Mahasiswa	Merumuskan Praproposal	Mahasiswa merumuskan proposal awal mengenai topik dan studi kasus yang akan diangkat dalam skripsinya
PB-TB-02	Mahasiswa	Memberikan Praproposal kepada koordinator KJFD	Mahasiswa memberikan preproposal kepada koordinator KJFD yang sesuai dengan bidang penelitian skripsi
PB-TB-03	Koordinator KJFD	Menerima Praproposal Mahasiswa	Koordinator KJFD menerima praproposal yang diberikan oleh mahasiswa untuk dipertimbangkan
PB-TB-04	Koordinator KJFD	Pengecekan Kesesuaian Bidang	Koordinator KJFD melakukan pengecekan apakah praproposal yang telah diterima sesuai dengan bidang tersebut.
PB-TB-05	Koordinator KJFD	Merekomendasikan Dosen Pembimbing	Koordinator KJFD merekomendasikan dosen pembimbing yang sesuai dengan topik dan studi kasus.
PB-TB-06	Koordinator KJFD	Menolak Praproposal	Jikalau praproposal yang diberikan tidak sesuai dengan bidang tersebut, maka koordinator KJFD menolak praproposal dan mengembalikan kepada mahasiswa
PB-TB-07	Mahasiswa	Berdiskusi dengan Calon Dosen Pembimbing	Mahasiswa berdiskusi dengan calon dosen pembimbing mengenai topik dan studi kasus skripsi yang diambil
PB-TB-08	Calon Dosen Pembimbing	Mempertimbangkan Kesanggupan Membimbing	Calon dosen pembimbing mempertimbangkan kesanggupan untuk membimbing mahasiswa tersebut



PB-TB-09	Calon Dosen Pembimbing	Menerima Sebagai Mahasiswa Bimbingan	Jikalau calon dosen pembimbing menyatakan sanggup, maka dosen menerima mahasiswa tersebut sebagai mahasiswa bimbingannya
----------	------------------------	--------------------------------------	--

#### 4.6 Penarikan Kebutuhan

Berdasarkan hasil perumusan bisnis proses *to-be*, maka dapat ditarik beberapa kebutuhan yang diperlukan pada sistem rekomendasi dosen. Prioritas kebutuhan dibagi menjadi kebutuhan utama dan kebutuhan pendukung. Kebutuhan sistem dijabarkan pada tabel 4.6.

Tabel 4.6 Kebutuhan Sistem

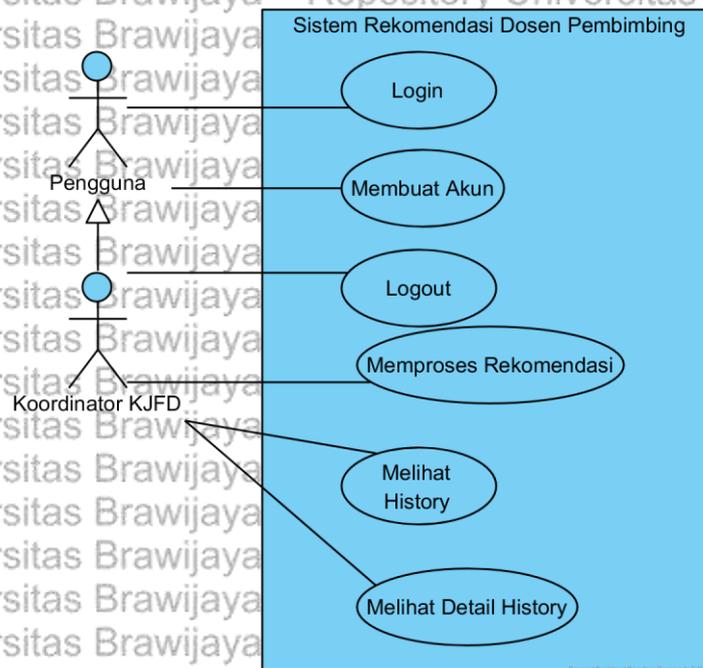
Kode Kebutuhan	Nama Kebutuhan	Deskripsi	Prioritas
RQ-01	Sistem dapat memberikan rekomendasi dosen	Sistem diharapkan mampu memberikan rekomendasi dosen berdasarkan latar belakang proposal	Kebutuhan utama
RQ-02	Pengguna dapat melihat history rekomendasi	Sistem diharapkan mampu memberikan list dan detail data historis dari rekomendasi dosen yang telah diberikan	Kebutuhan utama
RQ-03	Pengguna yang dapat menggunakan hanya koordinator KJFD dan kepala program studi	Sistem diharapkan mampu melakukan autentikasi akun koordinator KJFD dan kepala program studi	Kebutuhan utama
RQ-04	Pengguna dapat membuat akun	Sistem diharapkan mampu memproses permintaan pembuatan akun baru	Kebutuhan pendukung
RQ-05	Pengguna dapat keluar dari dalam sistem	Sistem diharapkan mampu menerima permintaan keluar dari sistem dan memastikan akun tersebut tidak	Kebutuhan pendukung



	dapat digunakan sebelum masuk kembali
--	---------------------------------------

### 4.7 Use Case Diagram

Berdasarkan penarikan kebutuhan terdapat 5 kebutuhan yang harus dipenuhi dalam sistem rekomendasi dosen. Penarikan hubungan aktor dan kebutuhan (*use case*) digambarkan pada diagram *Use Case* pada gambar 4.3.



Gambar 4.3 Diagram *Use Case* Sistem Rekomendasi Dosen Pembimbing

Sistem rekomendasi dosen pembimbing ini memiliki 6 *use case* dan 2 aktor sebagai pelaku dari *use case* tersebut, penjelasan mengenai *use case* diagram tersebut dideskripsikan pada tabel 4.7.

Tabel 4.7 Deskripsi Diagram *Use Case* Sistem Rekomendasi Dosen Pembimbing

Kode <i>Use Case</i>	Kode Kebutuhan	Nama <i>Use Case</i>	Aktor	Deskripsi
UC-01	RQ-03	Login	Pengguna	<i>Use Case</i> ini menjelaskan kebutuhan bahwa pengguna dapat masuk ke dalam sistem melalui autentikasi login
UC-02	RQ-04	Membuat Akun	Pengguna	<i>Use Case</i> ini menjelaskan kebutuhan pengguna untuk dapat membuat sebuah akun untuk mengakses sistem



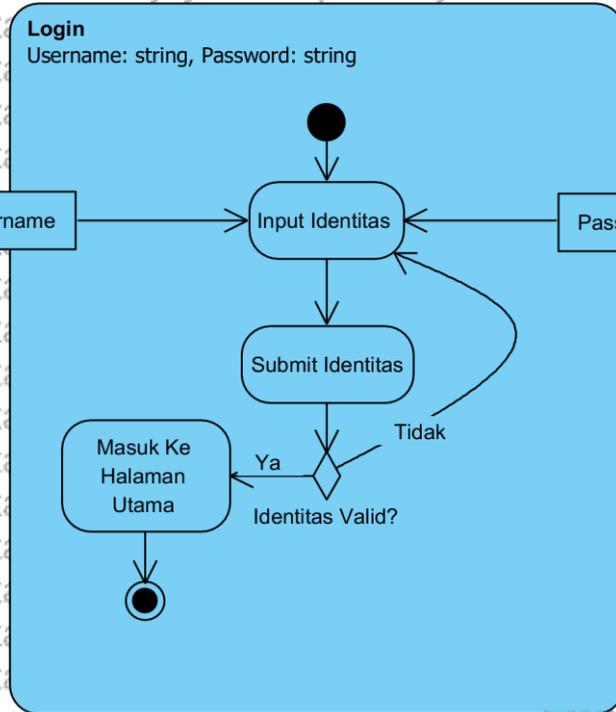
UC-03	RQ-05	Logout	Kepala KJFD	<i>Use Case</i> ini menjelaskan kebutuhan KKJFD untuk dapat keluar dari sistem.
UC-04	RQ-01	Memproses Rekomendasi	Kepala KJFD	<i>Use Case</i> ini menjelaskan kebutuhan KKJFD untuk dapat menggunakan sistem sebagai alat untuk menghasilkan rekomendasi dosen pembimbing.
UC-05	RQ-02	Melihat <i>History</i> Rekomendasi	Kepala KJFD	<i>Use Case</i> ini menjelaskan kebutuhan KKJFD untuk dapat melihat list <i>history</i> rekomendasi yang telah dilakukan oleh KKJFD tersebut.
UC-06	RQ-02	Melihat Detail <i>History</i> Rekomendasi	Kepala KJFD	<i>Use Case</i> ini menjelaskan kebutuhan KKJFD untuk dapat melihat detail <i>history</i> rekomendasi yang telah dilakukan oleh KKJFD tersebut.

#### 4.8 Activity Diagram

Berdasarkan *use case* yang telah dijelaskan, maka alur aktivitas setiap *use case* dijelaskan menggunakan *activity diagram*.

##### 4.8.1 Activity Diagram Login

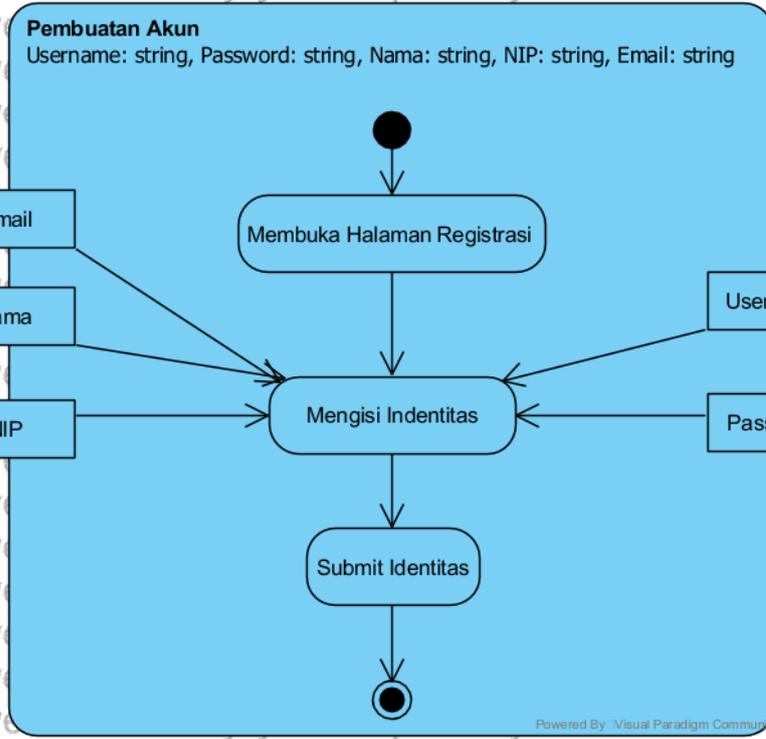
Aktivitas login dimulai pada pengguna memasukkan identitas berupa username dan password ke dalam sistem. Kemudian pengguna mensubmit identitas tersebut ke dalam sistem. Sistem akan melakukan pengecekan mengenai validitas dan kesamaan identitas tersebut dengan yang tersimpan pada *database*. Jikalau data pengguna sesuai dan valid, maka sistem akan menampilkan pengguna halaman utama. *Activity Diagram* login digambarkan pada gambar 4.4.



Gambar 4.4 Activity Diagram Login

#### 4.8.2 Activity Diagram Membuat Akun

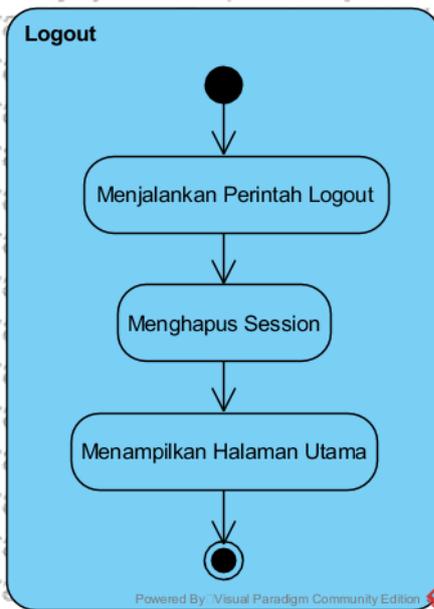
Aktivitas membuat akun dimulai dari pengguna membuka halaman registrasi. Kemudian pengguna mengisi identitas berupa NIP, Nama, Email, *Username*, dan *Password*. Setelah seluruh identitas terisi pengguna melakukan submit data tersebut untuk disimpan oleh sistem pada *database*. *Activity Diagram* membuat akun dijelaskan pada gambar 4.5.



Gambar 4.5 Activity Diagram Membuat Akun

### 4.8.3 Activity Diagram Logout

Aktivitas *Logout* menjelaskan langkah-langkah keluar dari sistem. Langkah ini diawali dengan kepala KJFD sebagai pengguna menjalankan perintah *logout*, kemudian sistem akan menghapus *session* dan menampilkan halaman login kepada pengguna. *Activity diagram logout* dijelaskan pada gambar 4.6.

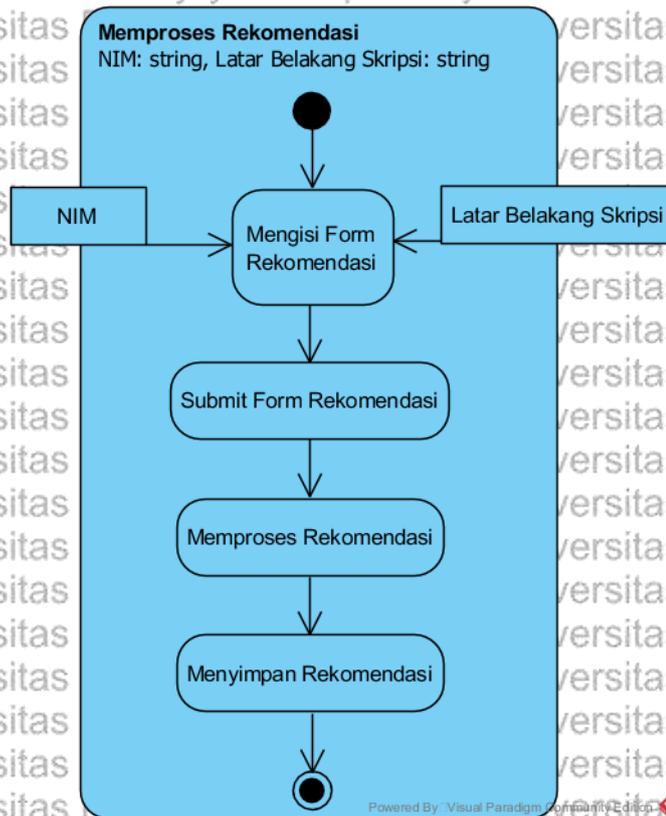


Gambar 4.6 Activity Diagram Logout



#### 4.8.4 Activity Diagram Memproses Rekomendasi

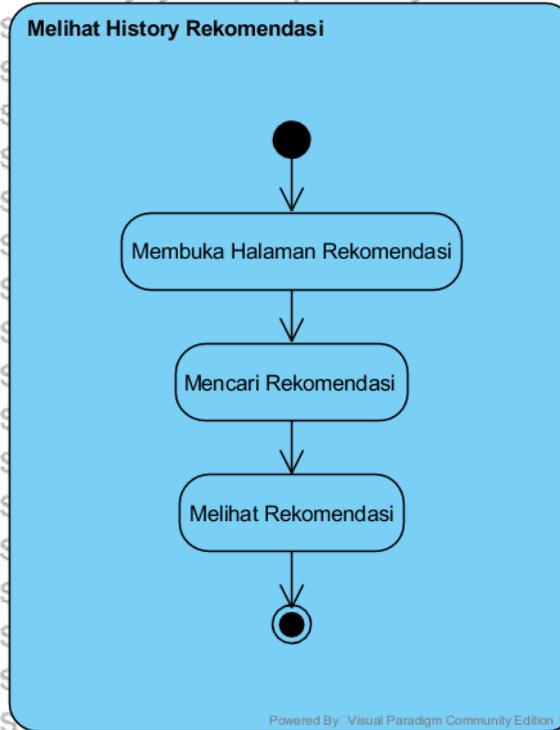
Aktivitas memproses rekomendasi dimulai dari kepala KJFD atau Kaprodi mengisi form rekomendasi dengan nim dan latar belakang skripsi/proposal. Setelah mengisi kebutuhan form, koordinator KJFD atau Kaprodi mensubmit form rekomendasi tersebut. Sistem akan memproses latar belakang menjadi prediksi kjd yang sesuai dan rekomendasi dosen. Hasil proses latar belakang dan nim akan disimpan ke dalam *database* agar dapat diakses kembali. *Activity diagram* memproses rekomendasi digambarkan pada gambar 4.7.



Gambar 4.7 Activity Diagram Memproses Rekomendasi

#### 4.8.5 Activity Diagram Melihat History Rekomendasi

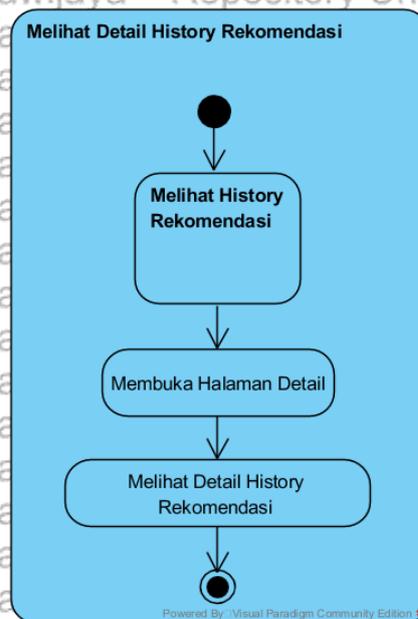
Aktivitas melihat *history* dimulai dari kepala KJFD atau kaprodi membuka halaman *history* rekomendasi. Kemudian pengguna dapat mencari *history* yang diinginkan dan melihat *history* rekomendasi tersebut. *Activity Diagram* melihat *history* digambarkan pada gambar 4.8.



Gambar 4.8 Activity Diagram Melihat History

#### 4.8.6 Activity Diagram Melihat Detail History Rekomendasi

Aktivitas melihat detail *history* rekomendasi dimulai dengan Kepala KJFD atau Kaprodi melakukan aktivitas melihat *history* rekomendasi terlebih dahulu. Kemudian pengguna membuka halaman detail dari *history* yang terpilih dan melihat detail *history* rekomendasi. Activity diagram melihat detail *history* rekomendasi digambarkan pada gambar 4.9.



Gambar 4.9 Activity Diagram Melihat Detail History Rekomendasi



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan Perangkat Lunak

Perancangan perangkat lunak menjelaskan bagaimana sistem akan dibangun. Perancangan mencoba untuk memberikan cetak biru yang akan digunakan untuk membangun sebuah sistem. Perancangan terdiri atas diagram dan perancangan mengenai data dan bagaimana data akan diolah untuk menghasilkan rekomendasi dosen.

#### 5.1.1 Perancangan Arsitektur Sistem

Sistem per Rekomendasi dosen pembimbing dibangun berdasarkan arsitektur *Model-View-Controller* (MVC) dengan bantuan *Web Service*. Arsitektur sistem per Rekomendasi dosen pembimbing dijelaskan pada gambar 5.1.



Gambar 5.1 Arsitektur Sistem

Pada gambar 5.1 dijelaskan bahwa sistem per Rekomendasi dosen pembimbing diakses menggunakan browser dikarenakan sistem dibangun berdasarkan *web-based*. Sistem utama dibangun menggunakan bahasa pemrograman PHP dengan *framework codeigniter*. *Codeigniter* adalah *framework* yang mengaplikasikan arsitektur MVC. Sistem rekomendasi berhubungan dengan *web service* sebagai pemroses rekomendasi. *Web service* dibangun menggunakan bahasa *python* dan berfungsi untuk melakukan pemrosesan kata latar belakang menjadi rekomendasi nama dosen pembimbing. Pada bahasa pemrograman *python*, terdapat library bernama *Natural Language Toolkit* (NLTK). NLTK berfungsi sebagai *library* untuk melakukan pemrosesan teks latar belakang menjadi sesuatu yang dapat dipahami oleh sistem yang disebut *text preprocessing*. NLTK juga berisi metode untuk pemrosesan menggunakan *multi-class support vector machine*.



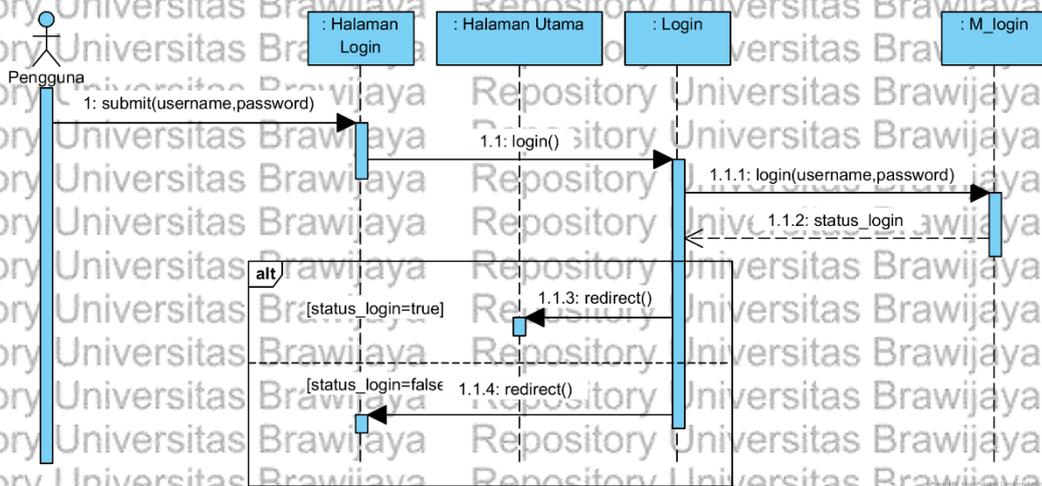
Oleh karena hal tersebut, arsitektur dipisahkan antara sistem utama dengan *web service*.

### 5.1.2 Sequence Diagram

Alur pengiriman data, pemanggilan fungsi, serta hubungan antar entitas diperjelas pada *sequence diagram*. *Sequence diagram* dibentuk berdasarkan runtutan aktivitas yang telah digambarkan pada *activity diagram*. *Sequence diagram* untuk sistem rekomendasi dosen pembimbing berjumlah 6. Jumlah tersebut didasarkan pada jumlah fungsi yang telah dijelaskan pada *use case diagram*.

#### 5.1.2.1 Sequence Diagram Login

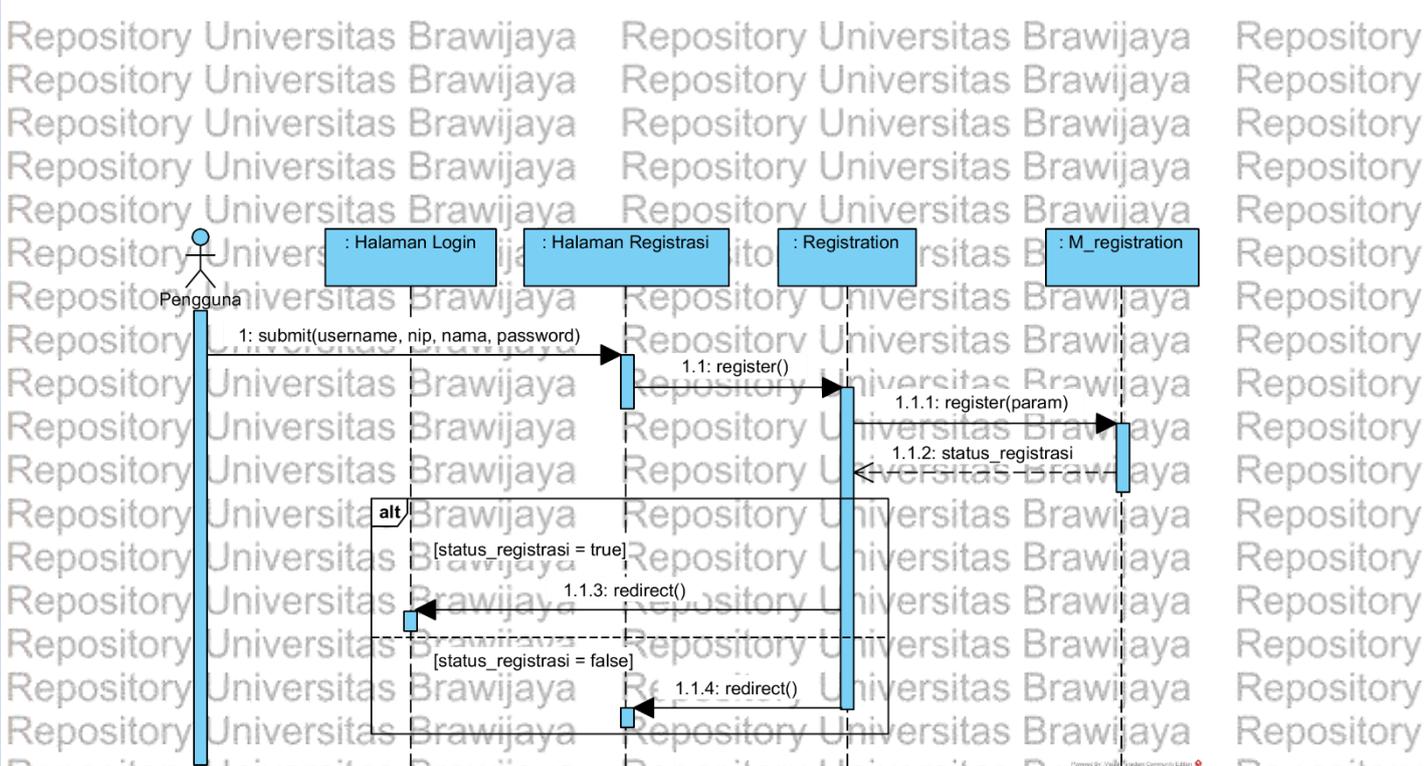
Pada *sequence diagram login* terdapat 1 aktor, yaitu pengguna. Pengguna yang dimaksud adalah semua yang dapat mengakses sistem, baik manusia ataupun *web service*. Terdapat 4 entitas yang terlibat, diantaranya: Halaman Login, Halaman utama, *Controller Login*, dan *Model M login*. Terdapat 1 *decision* yang dimana alurnya terbagi berdasarkan nilai *boolean* dari status login. *Sequence diagram login* digambarkan pada gambar 5.2



Gambar 5.2 Sequence Diagram Login

#### 5.1.2.2 Sequence Diagram Membuat Akun

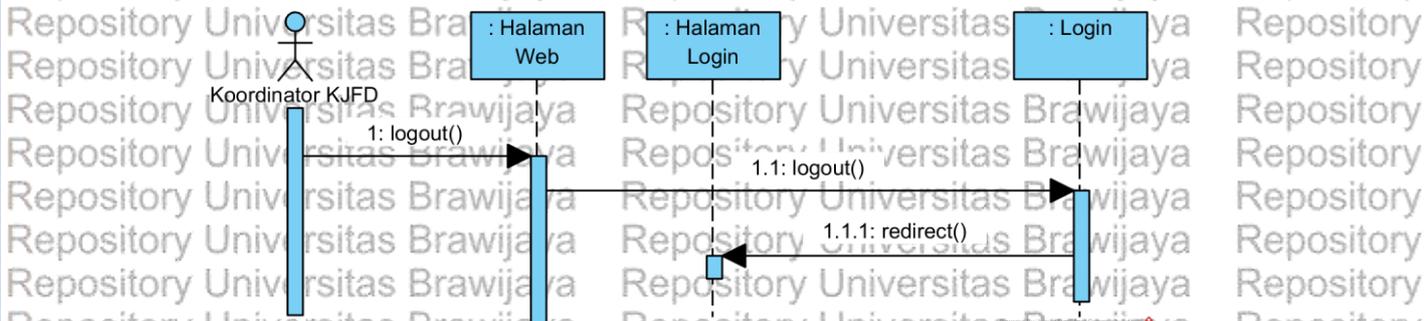
Pada *sequence diagram* membuat akun terdapat 1 aktor, yaitu pengguna. Pengguna yang dimaksud adalah semua yang dapat mengakses sistem, baik manusia ataupun *web service*. Terdapat 4 entitas yang terlibat, diantaranya: Halaman Login, Halaman Registrasi, *Controller Registration*, *Model M\_registration*. Terdapat 1 *decision* yang dimana alurnya terbagi berdasarkan nilai *boolean* dari status\_registrasi. *Sequence diagram* membuat akun digambarkan pada gambar 5.3



Gambar 5.3 Sequence Diagram Membuat Akun

5.1.2.3 Sequence Diagram Logout

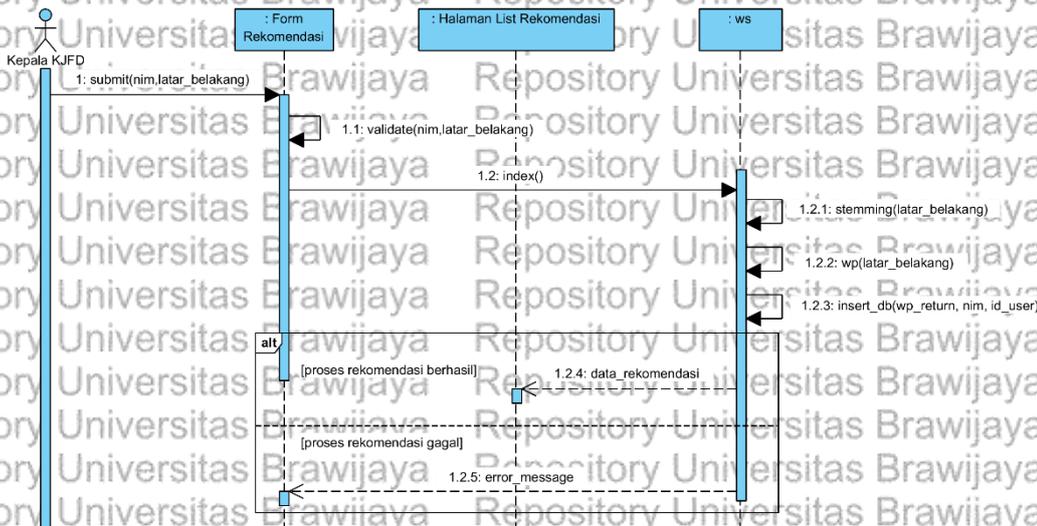
Pada sequence diagram logout terdapat 1 aktor, yaitu Koordinator KJFD/Kaprodi. Terdapat 3 entitas yang terlibat, diantaranya: Halaman Web, Halaman Login, Controller Login,. Halaman Web yang dimaksud adalah segala tampilan halaman yang berada pada sistem setelah pengguna masuk. Sequence diagram logout digambarkan pada gambar 5.4



Gambar 5.4 Sequence Diagram Logout

5.1.2.4 Sequence Diagram Memproses Rekomendasi

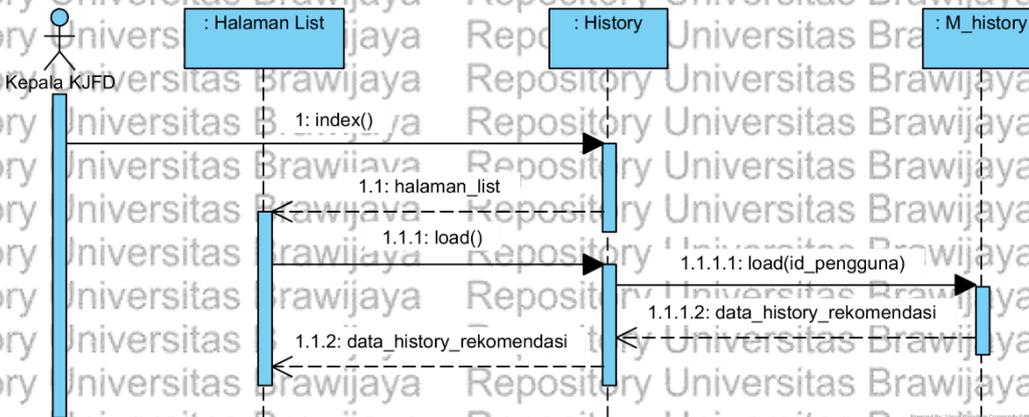
Pada sequence diagram memproses rekomendasi terdapat 1 aktor, yaitu Koordinator KJFD/Kaprodi. Terdapat 3 entitas yang terlibat, diantaranya: Form Rekomendasi, Halaman List Rekomendasi, dan Model ws. Model ws adalah sebuah web service yang dibangun berdasarkan python. Web service ws berfungsi sebagai pemroses dari latar belakang menjadi rekomendasi. Sequence diagram memproses rekomendasi digambarkan pada gambar 5.5



**Gambar 5.5 Sequence Diagram Memproses Rekomendasi**

**5.1.2.5 Sequence Diagram Melihat History Rekomendasi**

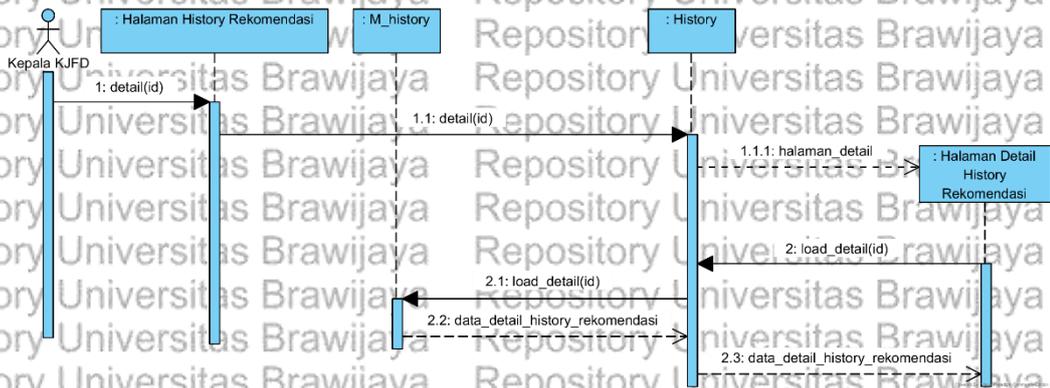
Pada *sequence diagram* melihat *history* rekomendasi terdapat 1 aktor, yaitu Koordinator KJFD/Kaprodi. Terdapat 3 entitas yang terlibat, diantaranya: Halaman *List History*, *Controller History*, dan *Model M\_history*. *Sequence diagram* melihat *history* rekomendasi digambarkan pada gambar 5.6



**Gambar 5.6 Sequence Diagram Melihat History Rekomendasi**

**5.1.2.6 Sequence Diagram Melihat Detail History Rekomendasi**

Pada *sequence diagram* melihat *history* rekomendasi terdapat 1 aktor, yaitu Koordinator KJFD/Kaprodi. Terdapat 3 entitas yang terlibat, diantaranya: Halaman *List History*, *Controller History*, dan *Model M\_history*. *Sequence diagram* melihat *history* rekomendasi digambarkan pada gambar 5.7



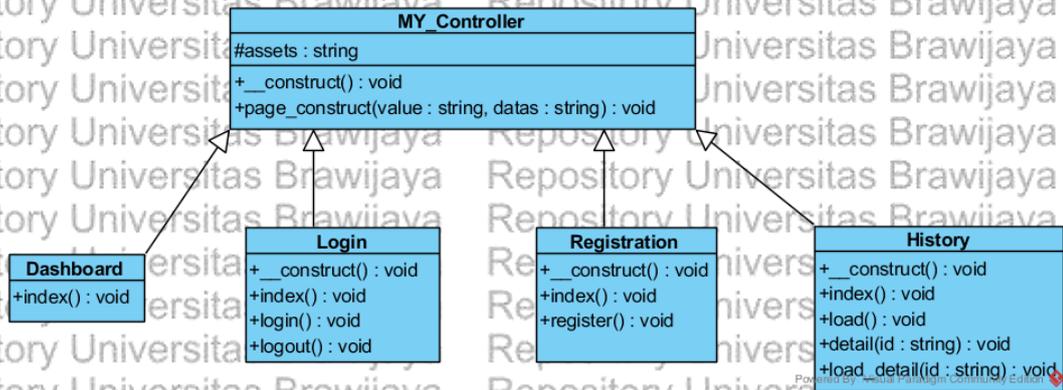
Gambar 5.7 Sequence Diagram Melihat Detail History Rekomendasi

### 5.1.3 Class Diagram

Hubungan antar objek dan class dijelaskan melalui class diagram. Pada sistem rekomendasi dosen pembimbing terdapat 3 kelompok class, yaitu: Class Controller, Class Model, dan Class Web Service.

#### 5.1.3.1 Class Diagram Controller

Class Controller pada sistem rekomendasi dosen pembimbing berisi kumpulan class yang berfungsi sebagai pengatur jalannya logika pada framework Codeigniter. Segala Controller yang dibuat umumnya menurunkan fitur dan fungsi dari class CI\_Controller. Pada sistem ini, terdapat sebuah class MY\_Controller yang berfungsi sebagai penambahan fitur dan fungsi dasar yang akan digunakan oleh sejumlah children class yang akan dibuat. Children class untuk class controller pada sistem ini berjumlah 4, yaitu: Dashboard, Login, Registration, dan History. Children class tersebut melakukan ekstensi kepada satu superclass yaitu class My\_Controller. Class diagram controller pada sistem ini digambarkan pada gambar 5.8



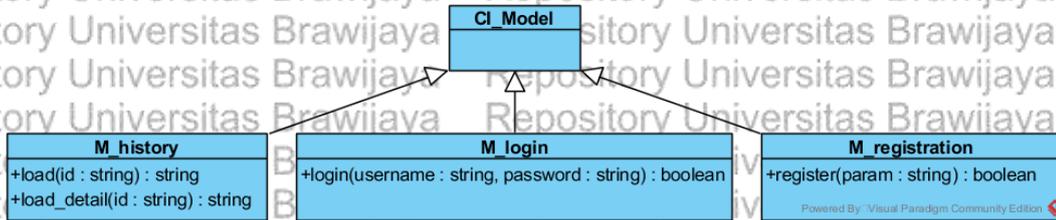
Gambar 5.8 Class Diagram Controller

#### 5.1.3.2 Class Diagram Model

Class model pada sistem rekomendasi dosen pembimbing berisi kumpulan class yang berfungsi untuk berhubungan dengan entitas entitas seperti entitas database. Fungsi class model tersebut sesuai dengan fungsi class model pada



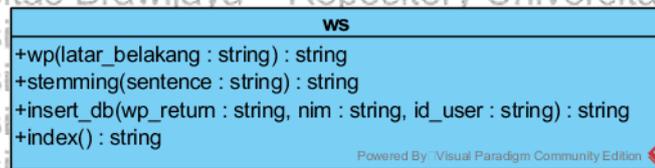
framework Codeigniter. Setiap class model pada framework Codeigniter pada dasarnya menjadi children class dari CI\_Model. Children class untuk class model pada sistem ini berjumlah 3, diantaranya: M\_history, M\_login, dan M\_registration. Class diagram model pada sistem ini digambarkan pada gambar 5.9



Gambar 5.9 Class Diagram Model

### 5.1.3.3 Class Diagram Web Service

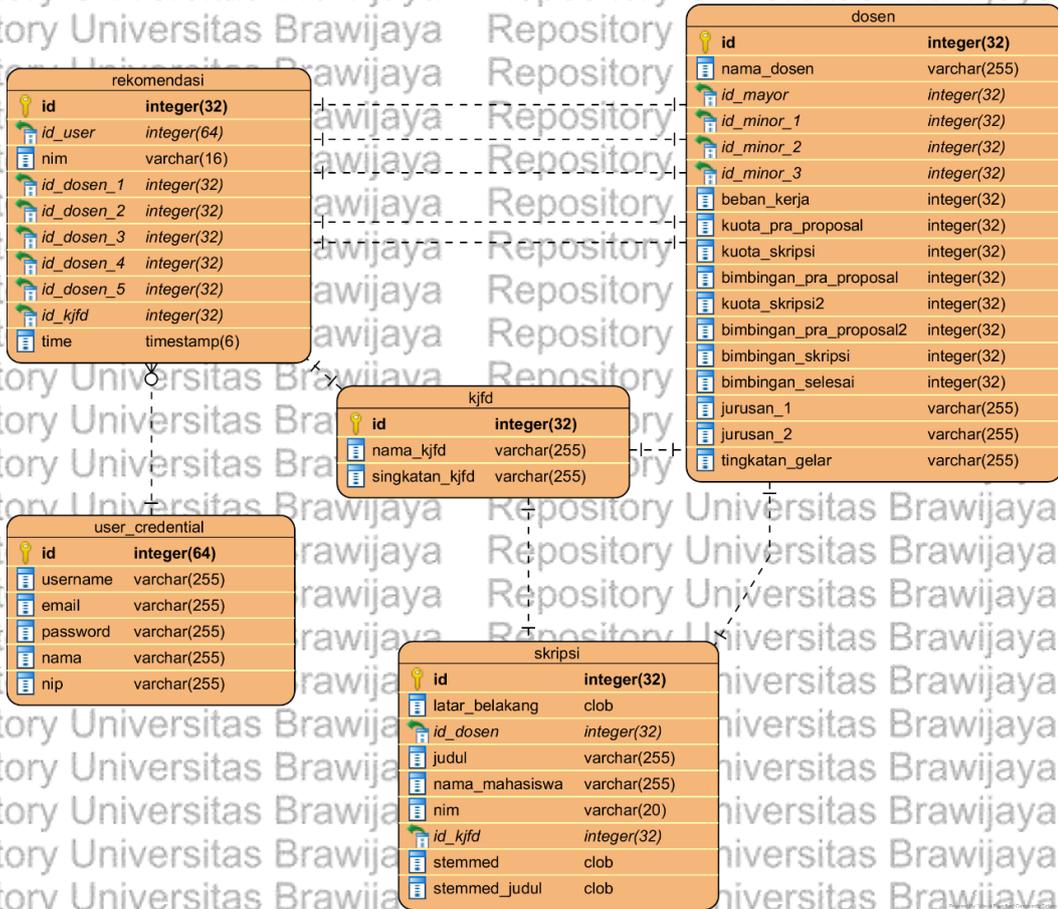
Class web service pada sistem rekomendasi pembimbing adalah program berbahasa python yang berfungsi sebagai web service untuk memproses latar belakang menjadi prediksi KJFD dan rekomendasi dosen pembimbing. Class diagram web service pada sistem ini digambarkan pada gambar 5.10



Gambar 5.10 Class Diagram Web Service

### 5.1.4 Perancangan Entity Relationship Diagram

Entity relationship diagram (ERD) menjelaskan bagaimana skema logikal pada database saling berhubungan. Skema logikal menjelaskan struktur database seperti hubungan antar tabel, view, dan procedure. ERD sistem perekomendasi dosen pembimbing dijelaskan pada gambar 5.11



**Gambar 5.11 Entity Relational Diagram**

Pada gambar 5.11 terdapat 5 tabel yang saling berhubungan. Kelima tabel tersebut adalah tabel rekomendasi, user\_credential, kjfd, skripsi, dan dosen. Jenis hubungan yang terdapat pada diagram ini adalah *one-to-one* dan *one-to-many*. Hubungan *one-to-many* terdapat pada hubungan antar tabel kjfd dan skripsi, kjfd dan rekomendasi, kjfd dan dosen, rekomendasi dan dosen. Hubungan *many to one* terdapat pada tabel user\_credential dan rekomendasi.

### 5.1.5 Perancangan Data

Proses *mining* merupakan proses pengolahan data, oleh karena itu perancangan dan pengolahan data menjadi prioritas pada penelitian ini. Pada perancangan data terdapat beberapa tahapan diantaranya: pengumpulan data dan perancangan *dataset* latar belakang skripsi. Pengumpulan data menjelaskan asal data dan bagaimana data terkumpul. Perancangan *dataset* latar belakang skripsi menjelaskan bagaimana data latar belakang skripsi terhubung dengan kelas KJFD.

#### 5.1.5.1 Pengumpulan Data

Data latar belakang didapatkan melalui data latar belakang pada jurnal lulusan Sistem Informasi Fakultas Ilmu Komputer UB. Data jurnal didapatkan melalui situs



web JPTIHK yang beralamatkan <http://j-ptiik.ub.ac.id/index.php/j-ptiik>. Situs JPTIHK berisi jurnal mahasiswa yang berhubungan dengan teknologi informasi dan ilmu komputer.

Data yang didapatkan dari sebuah jurnal adalah: Latar belakang jurnal, judul jurnal, nama mahasiswa, dan nama dosen pembimbing. Latar belakang jurnal didapatkan pada bab pendahuluan sebuah jurnal. Nama dosen pembimbing yang digunakan adalah nama dosen pembimbing satu. Nama dosen pembimbing satu tertera pada nama penulis 2. Nama mahasiswa tertera pada nama penulis 1.

Data yang diambil didasarkan pada mahasiswa Sistem Informasi Fakultas Ilmu Komputer UB yang telah melaksanakan yudisium. Periode masa yudisium yang digunakan adalah yudisium yang dilaksanakan sejak tahun 2018 hingga Agustus 2019. Jumlah mahasiswa yang mengikuti yudisium pada masa tersebut sejumlah 476 orang. Jumlah dosen yang membimbing mahasiswa pada periode tersebut sejumlah 26 dosen.

#### 5.1.5.2 Perancangan *Dataset* Latar Belakang Skripsi

Perancangan *dataset* latar belakang skripsi merupakan tahap dimana setiap data latar belakang skripsi diberi label bidang/kjfd secara manual. Contoh *dataset* latar belakang dapat dilihat pada tabel 5.1

**Tabel 5.1 *Dataset* Latar Belakang Skripsi**

Kode	ID_Dosen	Potongan Latar Belakang	KJFD
SD-01	16	Tujuan dari penelitian ini adalah mengembangkan sistem informasi yang dapat mengatasi pencarian data pasien yang saat ini lama (berdasarkan analisis proses bisnis). Adapun batasan masalah dari penelitian ini adalah hanya pada BNN Kota Malang Seksi Rehabilitasi saja, dalam penerapannya penulis menggunakan kerangka kerja bahasa pemrograman PHP yaitu Codelgniter. Penggunaan strategi yang cocok sesuai dengan situasi dan kondisi organisasi berperan penting dalam berhasilnya pengembangan sistem informasi, peneliti menggunakan siklus pengembangan iterative - incremental, pendekatan berorientasi objek dan implementasi menggunakan pattern mvc dengan pengujian skenario.	Pengembangan Sistem Informasi
SD-02	22	Dalam melakukan penilaian terhadap permasalahan yang terjadi maka perlu melakukan evaluasi	Tata Kelola dan Manajemen SI



		<p>terhadap layanan TI yang ada dalam perusahaan PT. Aerofood Indonesia dengan framework COBIT 5, dalam hal ini evaluasi mampu untuk meningkatkan kualitas sistem serta tercapainya kesesuaian kualitas yang diharapkan oleh perusahaan sendiri. Tentu dalam melakukan penelitian ini hasil yang didapatkan dijadikan sebagai panduan dan rekomendasi untuk dapat menciptakan keputusan yang sejajar dengan tujuan utama perusahaan PT. Aerofood Indonesia.</p>	
<p>SD-03</p>	<p>10</p>	<p>Akan tetapi, proses merancang jadwal mata kuliah tidaklah mudah dan seringkali menimbulkan beberapa permasalahan diantaranya kelas yang tersedia tidak dapat memenuhi jumlah mahasiswa yang berminat dan kelas yang tersedia cukup banyak namun sepi peminat. Akibatnya, pihak akademik selalu menambah waktu pengisian KRS untuk mengakomodir mahasiswa yang kehabisan kelas. Sehingga dibutuhkan suatu sistem yang dapat melakukan prediksi apakah mahasiswa mengambil mata kuliah atau tidak. Tujuan dari penelitian ini yaitu mengimplementasikan data mining dengan algoritme Naive Bayes untuk melakukan klasifikasi terhadap mahasiswa pengambil mata kuliah berdasarkan atribut - atribut mahasiswa yaitu Nilai, IP, IPK, SKS, SKS dan Semester serta mengetahui tingkat akurasi Naive Bayes jika diterapkan untuk klasifikasi mahasiswa pengambil mata kuliah.</p>	<p>Manajemen Data dan Informasi</p>

**5.1.6 Perancangan Desain Antarmuka Pengguna**

Antarmuka diperlukan untuk mempermudah pengguna dalam berkomunikasi dengan sistem. Antarmuka pengguna akan berfungsi untuk menerima masukan pengguna dan hasil dari proses sistem. Desain antarmuka dibangun untuk mempermudah dan memberi gambaran mengenai tampilan sistem yang akan dibangun.



### 5.1.6.1 Halaman Login

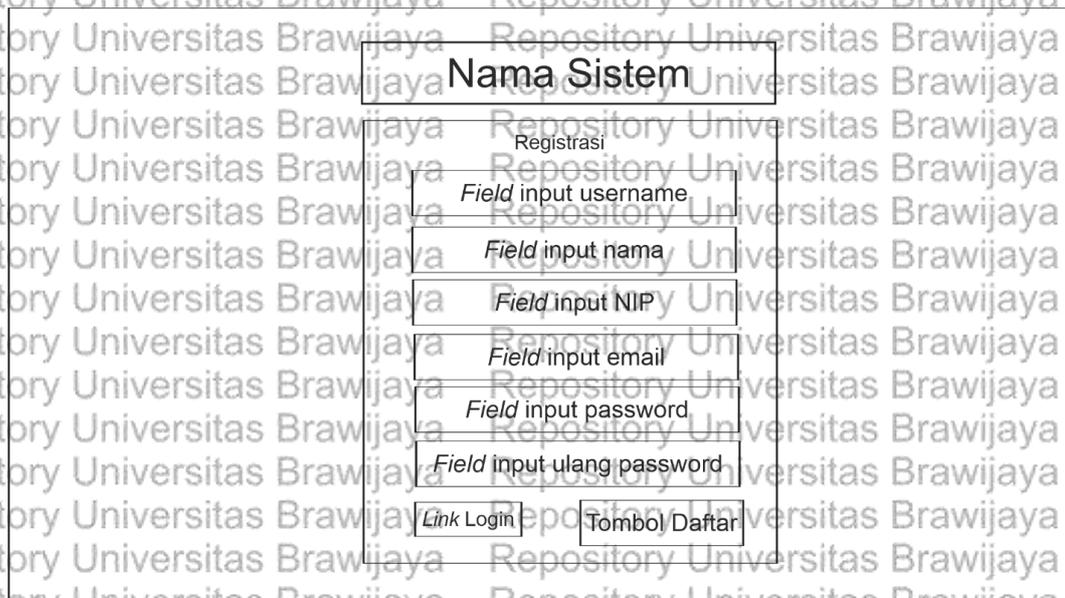
Perancangan desain halaman login berisi peletakan *field* inputan dan tombol untuk melakukan proses login. Halaman ini berisi 3 *field* inputan, 1 tombol, dan 1 *link*. Perancangan desain halaman login dapat dilihat pada gambar 5.12



Gambar 5.12 Perancangan Antarmuka Halaman Login

### 5.1.6.2 Halaman Registrasi

Perancangan desain halaman registrasi berisi peletakan *field* inputan dan tombol untuk melakukan proses membuat akun. Halaman ini berisi 6 *field* inputan, 1 tombol, dan 1 *link*. Perancangan desain halaman registrasi dapat dilihat pada gambar 5.13



Gambar 5.13 Perancangan Antarmuka Halaman Registrasi



### 5.1.6.3 Halaman *List Rekomendasi*

Perancangan desain halaman *list* rekomendasi berisi peletakan *field* inputan dan tabel untuk melihat *list* dosen yang direkomendasikan. Halaman ini berisi 1 *field* inputan, 1 tombol, dan 1 tabel. Perancangan desain halaman *list* rekomendasi dapat dilihat pada gambar 5.14



Gambar 5.14 Perancangan Antarmuka Halaman *List Rekomendasi*

### 5.1.6.4 Halaman *Submit Latar Belakang*

Perancangan desain halaman *submit* latar belakang berisi peletakan *field* inputan dan tombol untuk mengirim data latar belakang agar diproses menjadi rekomendasi dosen pembimbing. Halaman ini berbentuk *modal* yang tampil di depan halaman *list* rekomendasi. Halaman ini berisi 2 *field* inputan, dan 1 tombol. Perancangan desain halaman *submit* latar belakang dapat dilihat pada gambar 5.15



**Gambar 5.15 Perancangan Antarmuka Halaman *Submit Latar Belakang***

**5.1.6.5 Halaman *List History* Rekomendasi**

Perancangan desain halaman *list history* rekomendasi berisi peletakan *field* inputan dan tabel untuk melihat *list* data rekomendasi yang pernah dilakukan. Halaman ini berisi 1 *field* inputan, dan 1 tabel. Perancangan desain halaman *list history* rekomendasi dapat dilihat pada gambar 5.16



**Gambar 5.16 Perancangan Antarmuka Halaman *List History* Rekomendasi**

**5.1.6.6 Halaman *Detail History* Rekomendasi**

Perancangan desain halaman detail *history* rekomendasi berisi peletakan data dan *list* dosen yang pernah direkomendasikan. Halaman ini berisi 1 tabel. Perancangan desain halaman detail *history* rekomendasi dapat dilihat pada gambar 5.17



**Gambar 5.17 Perancangan Antarmuka Halaman Detail *History* Rekomendasi**

### 5.1.7 Text Preprocessing

*Text preprocessing* adalah proses untuk mempersiapkan data teks agar memiliki kualitas kualitas yang lebih tinggi. Pada *text preprocessing*, data teks diolah agar memiliki makna dan pengaruh pada proses klasifikasi.

#### 5.1.7.1 Case Folding

*Case folding* adalah proses perubahan teks menjadi huruf kecil. *Case folding* dilakukan untuk mengurangi perbedaan kata yang diakibatkan oleh perbedaan kapitalisasi huruf. Pada tabel 5.2 ditampilkan hasil proses *case folding* pada data sampel.

**Tabel 5.2 Case Folding Latar Belakang**

Kode	ID_Dosen	Potongan Latar Belakang	KJFD
SD-01	16	tujuan dari penelitian ini adalah mengembangkan sistem informasi yang dapat mengatasi pencarian data pasien yang saat ini lama (berdasarkan analisis proses bisnis). adapun batasan masalah dari penelitian ini adalah hanya pada bnn kota malang seksi rehabilitasi saja, dalam penerapannya penulis menggunakan kerangka kerja bahasa pemrograman php yaitu codeigniter . penggunaan strategi yang cocok sesuai dengan situasi dan kondisi organisasi berperan penting dalam berhasilnya pengembangan sistem informasi, peneliti menggunakan siklus pengembangan iterative - incremental , pendekatan berorientasi objek dan implementasi menggunakan pattern mvc dengan pengujian skenario.	Pengembangan Sistem Informasi
SD-02	22	dalam melakukan penilaian terhadap permasalahan yang terjadi maka perlu melakukan evaluasi terhadap layanan ti yang ada dalam perusahaan pt. aerofood indonesia dengan framework cobit 5, dalam hal ini evaluasi mampu untuk meningkatkan kualitas sistem serta tercapainya kesesuaian kualitas yang diharapkan oleh perusahaan sendiri. tentu dalam melakukan penelitian ini hasil yang	Tata Kelola dan Manajemen Si



		didapatkan dijadikan sebagai panduan dan rekomendasi untuk dapat menciptakan keputusan yang sejajar dengan tujuan utama perusahaan pt. aerofood indonesia.	
SD-03	10	akan tetapi, proses merancang jadwal mata kuliah tidaklah mudah dan seringkali menimbulkan beberapa permasalahan diantaranya kelas yang tersedia tidak dapat memenuhi jumlah mahasiswa yang berminat dan kelas yang tersedia cukup banyak namun sepi peminat. akibatnya, pihak akademik selalu menambah waktu pengisian krs untuk mengakomodir mahasiswa yang kehabisan kelas. sehingga dibutuhkan suatu sistem yang dapat melakukan prediksi apakah mahasiswa mengambil mata kuliah atau tidak. tujuan dari penelitian ini yaitu mengimplementasikan data mining dengan algoritme naive bayes untuk melakukan klasifikasi terhadap mahasiswa pengambil mata kuliah berdasarkan atribut - atribut mahasiswa yaitu nilai, ip, ipk, sks, dan semester serta mengetahui tingkat akurasi naive bayes jika diterapkan untuk klasifikasi mahasiswa pengambil mata kuliah.	Manajemen Data dan Informasi

**5.1.7.2 Penghapusan Tanda Baca**

Tanda baca pada kalimat tidak memiliki pengaruh pada proses *mining* teks. Oleh karena hal tersebut, tanda baca seperti [!"#\$%&'()\*+,-./:;<=>@[\\]^\_`{|}~] tidak diperlukan dan perlu dihapus. Pada tabel 5.3 ditampilkan hasil proses penghapusan tanda baca pada data sampel.

**Tabel 5.3 Penghapusan Tanda Baca pada Latar Belakang**

Kode	ID_Dosen	Potongan Latar Belakang	KJFD
SD-01	16	tujuan dari penelitian ini adalah mengembangkan sistem informasi yang dapat mengatasi pencarian data pasien yang saat ini lama berdasarkan analisis proses bisnis adapun batasan masalah dari penelitian ini adalah hanya pada	Pengembangan Sistem Informasi



		<p>bnn kota malang seksi rehabilitasi saja dalam penerapannya penulis menggunakan kerangka kerja bahasa pemrograman php yaitu codeigniter penggunaan strategi yang cocok sesuai dengan situasi dan kondisi organisasi berperan penting dalam berhasilnya pengembangan sistem informasi peneliti menggunakan siklus pengembangan iterative incremental pendekatan berorientasi objek dan implementasi menggunakan pattern mvc dengan pengujian skenario</p>	
SD-02	22	<p>dalam melakukan penilaian terhadap permasalahan yang terjadi maka perlu melakukan evaluasi terhadap layanan ti yang ada dalam perusahaan pt aerofood indonesia dengan framework cobit 5 dalam hal ini evaluasi mampu untuk meningkatkan kualitas sistem serta tercapainya kesesuaian kualitas yang diharapkan oleh perusahaan sendiri tentu dalam melakukan penelitian ini hasil yang didapatkan dijadikan sebagai panduan dan rekomendasi untuk dapat menciptakan keputusan yang sejajar dengan tujuan utama perusahaan pt aerofood indonesia</p>	Tata Kelola dan Manajemen SI
SD-03	10	<p>akan tetapi proses merancang jadwal mata kuliah tidaklah mudah dan seringkali menimbulkan beberapa permasalahan diantaranya kelas yang tersedia tidak dapat memenuhi jumlah mahasiswa yang berminat dan kelas yang tersedia cukup banyak namun sepi peminat akibatnya pihak akademik selalu menambah waktu pengisian krs untuk mengakomodir mahasiswa yang kehabisan kelas sehingga dibutuhkan suatu sistem yang dapat melakukan prediksi apakah mahasiswa mengambil mata kuliah atau tidak tujuan dari penelitian ini yaitu</p>	Manajemen Data dan Informasi



		mengimplementasikan data mining dengan algoritme naive bayes untuk melakukan klasifikasi terhadap mahasiswa pengambil mata kuliah berdasarkan atribut atribut mahasiswa yaitu nilai ip ipk sks sks dan semester serta mengetahui tingkat akurasi naive bayes jika diterapkan untuk klasifikasi mahasiswa pengambil mata kuliah	
--	--	--	--

### 5.1.7.3 Stemming

*Stemming* adalah proses untuk mengidentifikasi akar dari sebuah kata yang biasa disebut dengan kata dasar. Proses *stemming* berusaha menghapus imbuhan pada kata dan mensimplifikasi kata yang berulang. Pada tabel 5.4 ditampilkan hasil proses *stemming* pada data sampel.

**Tabel 5.4 Stemming pada Latar Belakang**

Kode	ID_Dosen	Potongan Latar Belakang	KJFD
SD-01	16	tujuan dari teliti ini adalah kembang sistem informasi yang dapat atas cari data pasien yang saat ini lama dasar analisis proses bisnis adapun batas masalah dari teliti ini adalah hanya pada bnn kota malang seksi rehabilitasi saja dalam terap tulis guna kerangka kerja bahasa pemrograman php yaitu codeigniter guna strategi yang cocok sesuai dengan situasi dan kondisi organisasi peran penting dalam hasil kembang sistem informasi teliti guna siklus kembang iterative incremental dekat orientasi objek dan implementasi guna pattern mvc dengan uji skenario	Pengembangan Sistem Informasi
SD-02	22	dalam laku nilai hadap masalah yang jadi maka perlu laku evaluasi hadap layan ti yang ada dalam usaha pt aerofood indonesia dengan framework cobit dalam hal ini evaluasi mampu untuk tingkat kualitas sistem serta capa sesuai kualitas yang harap oleh usaha sendiri tentu dalam laku teliti ini hasil yang dapat jadi bagai pandu dan rekomendasi untuk dapat cipta putus yang sejajar dengan tuju utama usaha pt aerofood indonesia	Tata Kelola dan Manajemen SI



SD-03	10	akan tetapi proses rancang jadwal mata kuliah tidak mudah dan seringkali timbul beberapa permasalahan antara kelas yang sedia tidak dapat penuh jumlah mahasiswa yang minat dan kelas yang sedia cukup banyak namun sepi minat akibat pihak akademik selalu tambah waktu isi krs untuk mengakomodir mahasiswa yang habis kelas sehingga butuh suatu sistem yang dapat laku prediksi apakah mahasiswa ambil mata kuliah atau tidak tuju dari teliti ini yaitu implementasi data mining dengan algoritme naive bayes untuk laku klasifikasi hadap mahasiswa ambil mata kuliah dasar atribut atribut mahasiswa yaitu nilai ip ipk sks sks dan semester serta tahu tingkat akurasi naive bayes jika terap untuk klasifikasi mahasiswa ambil mata kuliah	Manajemen Data dan Informasi
-------	----	---	------------------------------

#### 5.1.7.4 Penghapusan *Stop Word*

*Stop word* merupakan kata yang umum digunakan sehingga muncul dalam jumlah besar namun tidak memiliki makna secara signifikan. Pada Bahasa Indonesia, *stop word* dapat ditemui seperti “dan”, “yang”, “dari”, “ke”, “maka”, dan lain lain. Kemunculan kata-kata tersebut menghasilkan noise yang mengakibatkan algoritme klasifikasi menganggap kata tersebut menjadi kata yang sering muncul dan penting. Pada penelitian ini, proses penghapusan stop word dilakukan menggunakan bantuan library Sastrawi. Pada tabel 5.5 ditampilkan hasil proses penghapusan *stop word* pada data sampel.

Tabel 5.5 Penghapusan *Stopword* pada Latar Belakang

Kode	ID_Dosen	Potongan Latar Belakang	<i>Stopword</i> Terhapus	KJFD
SD-01	16	tuju teliti kembang sistem informasi atas cari data pasien lama dasar analisis proses bisnis adapun batas masalah teliti bnn kota malang seksi rehabilitasi terap tulis kerangka kerja bahasa pemrograman php codeigniter strategi cocok sesuai situasi kondisi organisasi peran penting	'guna', 'saja', 'dengan', 'dan', 'ini', 'yang', 'hanya', 'yaitu', 'dari', 'adalah', 'saat', 'dapat',	Pengembangan Sistem Informasi



		hasil kembang sistem informasi teliti siklus kembang iterative incremental dekat orientasi objek implementasi pattern mvc uji skenario	'dalam', 'pada'	
SD-02	22	laku nilai hadap masalah jadi perlu laku evaluasi hadap layanan ti usaha pt aerofood indonesia framework cobit evaluasi mampu tingkat kualitas sistem capa sesuai kualitas harap usaha sendiri laku teliti hasil jadi sebagai pandu rekomendasi cipta putus sejajar tuju utama usaha pt aerofood indonesia	'yang', 'dan', 'tentu', 'dengan', 'serta', 'maka', 'untuk', 'oleh', 'ini', 'dapat', 'dalam', 'hal', 'ada'	Tata Kelola dan Manajemen SI
SD-03	10	proses rancang jadwal mata kuliah mudah seringkali timbul beberapa permasalahan kelas sedia penuh jumlah mahasiswa minat kelas sedia cukup banyak sepi minat akibat pihak akademik selalu tambah waktu isi krs mengakomodir mahasiswa habis kelas butuh suatu sistem laku prediksi mahasiswa ambil mata kuliah tuju teliti implementasi data mining algoritme naive bayes laku klasifikasi hadap mahasiswa ambil mata kuliah dasar atribut atribut mahasiswa nilai ipk sks sks semester tahu tingkat akurasi naive bayes terap klasifikasi mahasiswa ambil mata kuliah	'jika', 'dengan', 'dari', 'antara', 'namun', 'yang', 'apakah', 'atau', 'akan', 'ini', 'sehingga', 'serta', 'dan', 'untuk', 'tetapi', 'yaitu', 'tidak', 'dapat'	Manajemen Data dan Informasi



### 5.1.7.5 Tokenisasi

Tokenisasi adalah pemisahan kata pada dokumen menjadi *token-token*. Dalam implementasinya setiap token akan dikumpulkan dalam bentuk *array*. Pada tabel 5.6 ditampilkan hasil proses tokenisasi pada data sampel

**Tabel 5.6 Tokenisasi pada Latar Belakang**

Kode	Potongan Latar Belakang Sebelum Tokenisasi	Potongan Latar Belakang Setelah Tokenisasi
SD-01	<p>tujuan dari teliti ini adalah kembang sistem informasi yang dapat atas cari data pasien yang saat ini lama dasar analisis proses bisnis adapun batas masalah dari teliti ini adalah hanya pada bnn kota malang seksi rehabilitasi saja dalam terap tulis guna kerangka kerja bahasa pemrograman php yaitu codeigniter guna strategi yang cocok sesuai dengan situasi dan kondisi organisasi peran penting dalam hasil kembang sistem informasi teliti guna siklus kembang iterative incremental dekat orientasi objek dan implementasi guna pattern mvc dengan uji skenario</p>	<p>['tujuan', 'dari', 'teliti', 'ini', 'adalah', 'kembang', 'sistem', 'informasi', 'yang', 'dapat', 'atas', 'cari', 'data', 'pasien', 'yang', 'saat', 'ini', 'lama', 'dasar', 'analisis', 'proses', 'bisnis', 'adapun', 'batas', 'masalah', 'dari', 'teliti', 'ini', 'adalah', 'hanya', 'pada', 'bnn', 'kota', 'malang', 'seksi', 'rehabilitasi', 'saja', 'dalam', 'terap', 'tulis', 'guna', 'kerangka', 'kerja', 'bahasa', 'pemrograman', 'php', 'yaitu', 'codeigniter', 'guna', 'strategi', 'yang', 'cocok', 'sesuai', 'dengan', 'situasi', 'dan', 'kondisi', 'organisasi', 'peran', 'penting', 'dalam', 'hasil', 'kembang', 'sistem', 'informasi', 'teliti', 'guna', 'siklus', 'kembang', 'iterative', 'incremental', 'dekat', 'orientasi', 'objek', 'implementasi', 'guna', 'pattern', 'mvc', 'dengan', 'uji', 'skenario']</p>
SD-02	<p>dalam laku nilai hadap masalah yang jadi maka perlu laku evaluasi hadap layanan ti yang ada dalam usaha pt aerofood indonesia dengan framework cobit dalam hal ini evaluasi mampu untuk tingkat kualitas sistem serta capa sesuai kualitas yang harap oleh usaha sendiri tentu dalam laku teliti ini hasil yang dapat jadi bagai pandu dan rekomendasi untuk dapat cipta putus yang sejajar dengan tuju utama usaha pt aerofood indonesia</p>	<p>['dalam', 'laku', 'nilai', 'hadap', 'masalah', 'yang', 'jadi', 'maka', 'perlu', 'laku', 'evaluasi', 'hadap', 'layanan', 'ti', 'yang', 'ada', 'dalam', 'usaha', 'pt', 'aerofood', 'indonesia', 'dengan', 'framework', 'cobit', 'dalam', 'hal', 'ini', 'evaluasi', 'mampu', 'untuk', 'tingkat', 'kualitas', 'sistem', 'serta', 'capa', 'sesuai', 'kualitas', 'yang', 'harap', 'oleh', 'usaha', 'sendiri', 'tentu', 'dalam', 'laku', 'teliti', 'ini', 'hasil', 'yang', 'dapat', 'jadi', 'bagai', 'pandu', 'dan', 'rekomendasi', 'untuk', 'dapat', 'cipta', 'putus', 'yang', 'sejajar', 'dengan', 'tju', 'utama', 'usaha', 'pt', 'aerofood', 'indonesia']</p>



Kode	Potongan Latar Belakang Sebelum Tokenisasi	Potongan Latar Belakang Setelah Tokenisasi
		'utama', 'usaha', 'pt', 'aerofood', 'indonesia']
SD-03	akan tetapi proses rancang jadwal mata kuliah tidak mudah dan seringkali timbul beberapa permasalahan antara kelas yang sedia tidak dapat penuh jumlah mahasiswa yang minat dan kelas yang sedia cukup banyak namun sepi minat akibat pihak akademik selalu tambah waktu isi krs untuk mengakomodir mahasiswa yang habis kelas sehingga butuh suatu sistem yang dapat laku prediksi apakah mahasiswa ambil mata kuliah atau tidak tuju dari teliti ini yaitu implementasi data mining dengan algoritme naive bayes untuk laku klasifikasi hadap mahasiswa ambil mata kuliah dasar atribut atribut mahasiswa yaitu nilai ip ipk sks sks dan semester serta tahu tingkat akurasi naive bayes jika terap untuk klasifikasi mahasiswa ambil mata kuliah	['proses', 'rancang', 'jadwal', 'mata', 'kuliah', 'mudah', 'seringkali', 'timbul', 'beberapa', 'permasalahan', 'kelas', 'sedia', 'penuh', 'jumlah', 'mahasiswa', 'minat', 'kelas', 'sedia', 'cukup', 'banyak', 'sepi', 'minat', 'akibat', 'pihak', 'akademik', 'selalu', 'tambah', 'waktu', 'isi', 'krs', 'mengakomodir', 'mahasiswa', 'habis', 'kelas', 'butuh', 'suatu', 'sistem', 'laku', 'prediksi', 'mahasiswa', 'ambil', 'mata', 'kuliah', 'tuju', 'teliti', 'implementasi', 'data', 'mining', 'algoritme', 'naive', 'bayes', 'laku', 'klasifikasi', 'hadap', 'mahasiswa', 'ambil', 'mata', 'kuliah', 'dasar', 'atribut', 'atribut', 'mahasiswa', 'nilai', 'ip', 'ipk', 'sks', 'sks', 'semester', 'tahu', 'tingkat', 'akurasi', 'naive', 'bayes', 'terap', 'klasifikasi', 'mahasiswa', 'ambil', 'mata', 'kuliah']

### 5.1.8 Pembobotan Kata

Proses *text mining* mencari seberapa besar pengaruh sebuah kata dalam kumpulan dokumen. Pengaruh sebuah kata dapat diketahui melalui bobot kata tersebut. Pembobotan kata dilakukan dengan menggunakan metode pembobotan *Term Frequency – Inverse Document Frequency* (TF-IDF). Jumlah kemunculan kata (*term frequency*) pada sebuah dokumen meningkatkan pengaruh kata pada dokumen tersebut. Perhitungan *inverse document frequency* menampilkan keunikan kata tersebut pada kumpulan dokumen. Pada tabel 5.7 ditampilkan hasil perhitungan TF, DF, dan IDF pada data sampel.



**Tabel 5.7 Nilai Term Frequency, Document Frequency, dan Inverse Document Frequency**

	SD-01	SD-02	SD-03	DF	IDF LOG(N/DF)
Adapun	1	0	0	1	0.477121
Aerofood	0	2	0	1	0.477121
Akademik	0	0	1	1	0.477121
Akibat	0	0	1	1	0.477121
Akurasi	0	0	1	1	0.477121
Algoritme	0	0	1	1	0.477121
Ambil	0	0	3	1	0.477121
Analisis	1	0	0	1	0.477121
Atas	1	0	0	1	0.477121
Atribut	0	0	2	1	0.477121
Bagai	0	1	0	1	0.477121
Bahasa	1	0	0	1	0.477121
Banyak	0	0	1	1	0.477121
Batas	1	0	0	1	0.477121
Bayes	0	0	2	1	0.477121
Beberapa	0	0	1	1	0.477121
Bisnis	1	0	0	1	0.477121
Bnn	1	0	0	1	0.477121
Butuh	0	0	1	1	0.477121
Capa	0	1	0	1	0.477121
Cari	1	0	0	1	0.477121
Cipta	0	1	0	1	0.477121
Cobit	0	1	0	1	0.477121
Cocok	1	0	0	1	0.477121
Codeigniter	1	0	0	1	0.477121
Cukup	0	0	1	1	0.477121
Dasar	1	0	1	2	0.176091
Data	1	0	1	2	0.176091



Dekat	1	0	0	1	0.477121
Evaluasi	0	2	0	1	0.477121
Framework	0	1	0	1	0.477121
Habis	0	0	1	1	0.477121
Hadap	0	2	1	2	0.176091
Harap	0	1	0	1	0.477121
Hasil	1	1	0	2	0.176091
Implementasi	1	0	1	2	0.176091
Incremental	1	0	0	1	0.477121
Indonesia	0	2	0	1	0.477121
Informasi	2	0	0	1	0.477121
Ip	0	0	1	1	0.477121
lpk	0	0	1	1	0.477121
Isi	0	0	1	1	0.477121
Iterative	1	0	0	1	0.477121
Jadi	0	2	0	1	0.477121
Jadwal	0	0	1	1	0.477121
Jumlah	0	0	1	1	0.477121
Kelas	0	0	3	1	0.477121
Kembang	3	0	0	1	0.477121
Kerangka	1	0	0	1	0.477121
Kerja	1	0	0	1	0.477121
Klasifikasi	0	0	2	1	0.477121
Kondisi	1	0	0	1	0.477121
Kota	1	0	0	1	0.477121
Krs	0	0	1	1	0.477121
kualitas	0	2	0	1	0.477121
Kuliah	0	0	4	1	0.477121
Laku	0	3	2	2	0.176091
Lama	1	0	0	1	0.477121
Layan	0	1	0	1	0.477121



Mahasiswa	0	0	6	1	0.477121
Malang	1	0	0	1	0.477121
Mampu	0	1	0	1	0.477121
Masalah	1	1	0	2	0.176091
Mata	0	0	4	1	0.477121
Mengakomodir	0	0	1	1	0.477121
Minat	0	0	2	1	0.477121
Mining	0	0	1	1	0.477121
Mudah	0	0	1	1	0.477121
Mvc	1	0	0	1	0.477121
Naive	0	0	2	1	0.477121
Nilai	0	1	1	2	0.176091
Objek	1	0	0	1	0.477121
Organisasi	1	0	0	1	0.477121
Orientasi	1	0	0	1	0.477121
Pandu	0	1	0	1	0.477121
Pasien	1	0	0	1	0.477121
Pattern	1	0	0	1	0.477121
Pemrograman	1	0	0	1	0.477121
Penting	1	0	0	1	0.477121
Penuh	0	0	1	1	0.477121
Peran	1	0	0	1	0.477121
Perlu	0	1	0	1	0.477121
Permasalahan	0	0	1	1	0.477121
Php	1	0	0	1	0.477121
Pihak	0	0	1	1	0.477121
Prediksi	0	0	1	1	0.477121
Proses	1	0	1	2	0.176091
PT	0	2	0	1	0.477121
Putus	0	1	0	1	0.477121
Rancang	0	0	1	1	0.477121



Rehabilitasi	1	0	0	1	0.477121
Rekomendasi	0	1	0	1	0.477121
Sedia	0	0	2	1	0.477121
Sejajar	0	1	0	1	0.477121
Seksi	1	0	0	1	0.477121
Selalu	0	0	1	1	0.477121
Semester	0	0	1	1	0.477121
Sendiri	0	1	0	1	0.477121
Sepi	0	0	1	1	0.477121
Seringkali	0	0	1	1	0.477121
Sesuai	1	1	0	2	0.176091
Siklus	1	0	0	1	0.477121
Sistem	2	1	1	3	0
Situasi	1	0	0	1	0.477121
Skenario	1	0	0	1	0.477121
Sks	0	0	2	1	0.477121
Strategi	1	0	0	1	0.477121
Suatu	0	0	1	1	0.477121
Tahu	0	0	1	1	0.477121
Tambah	0	0	1	1	0.477121
Teliti	3	1	1	3	0
Terap	1	0	1	2	0.176091
Ti	0	1	0	1	0.477121
Timbul	0	0	1	1	0.477121
Tingkat	0	1	1	2	0.176091
Tuju	1	1	1	3	0
Tulis	1	0	0	1	0.477121
Uji	1	0	0	1	0.477121
Usaha	0	3	0	1	0.477121
Utama	0	1	0	1	0.477121
Waktu	0	0	1	1	0.477121



Setelah ditemukan nilai *term frequency* dan *inverse document frequency* maka tahapan dilanjutkan dengan menghitung bobot TF-IDF. Bobot TF-IDF didapatkan melalui hasil perkalian antara nilai *term frequency* dengan nilai *inverse document frequency*. Contoh hasil perhitungan bobot TF-IDF pada kata 'usaha' pada SD-02 dijabarkan pada persamaan x.y sebagai berikut:

$$tf_{td} \times idf_t = 3 * 0.477121 = 1.431363$$

(x.y)

Pada tabel 5.8 ditampilkan hasil pembobotan TF-IDF setiap term pada data sampel

**Tabel 5.8 Bobot TF-IDF Setiap Term Pada Data Latar Belakang**

	SD-01	SD-02	SD-03
Adapun	0.477121	0	0
Aerofood	0	0.954242	0
Akademik	0	0	0.477121
Akibat	0	0	0.477121
Akurasi	0	0	0.477121
Algoritme	0	0	0.477121
Ambil	0	0	1.431363
Analisis	0.477121	0	0
Atas	0.477121	0	0
Atribut	0	0	0.954242
Bagai	0	0.477121	0
Bahasa	0.477121	0	0
Banyak	0	0	0.477121
Batas	0.477121	0	0
Bayes	0	0	0.954242
Beberapa	0	0	0.477121
Bisnis	0.477121	0	0
Bnn	0.477121	0	0
Butuh	0	0	0.477121
Capa	0	0.477121	0
Cari	0.477121	0	0
Cipta	0	0.477121	0



Cobit	0	0.477121	0
Cocok	0.477121	0	0
Codeigniter	0.477121	0	0
Cukup	0	0	0.477121
Dasar	0.176091	0	0.176091
Data	0.176091	0	0.176091
Dekat	0.477121	0	0
Evaluasi	0	0.954242	0
Framework	0	0.477121	0
Habis	0	0	0.477121
Hadap	0	0.352182	0.176091
Harap	0	0.477121	0
Hasil	0.176091	0.176091	0
Implementasi	0.176091	0	0.176091
Incremental	0.477121	0	0
Indonesia	0	0.954242	0
Informasi	0.954242	0	0
Ip	0	0	0.477121
Ipk	0	0	0.477121
Isi	0	0	0.477121
Iterative	0.477121	0	0
Jadi	0	0.954242	0
Jadwal	0	0	0.477121
Jumlah	0	0	0.477121
Kelas	0	0	1.431363
Kembang	1.431363	0	0
Kerangka	0.477121	0	0
Kerja	0.477121	0	0
Klasifikasi	0	0	0.954242
Kondisi	0.477121	0	0
Kota	0.477121	0	0



Krs	0	0	0.477121
kualitas	0	0.954242	0
Kuliah	0	0	1.908484
Laku	0	0.528273	0.352182
Lama	0.477121	0	0
Layan	0	0.477121	0
Mahasiswa	0	0	2.862726
Malang	0.477121	0	0
Mampu	0	0.477121	0
Masalah	0.176091	0.176091	0
Mata	0	0	1.908484
Mengakomodir	0	0	0.477121
Minat	0	0	0.954242
Mining	0	0	0.477121
Mudah	0	0	0.477121
Mvc	0.477121	0	0
Naive	0	0	0.954242
Nilai	0	0.176091	0.176091
Objek	0.477121	0	0
Organisasi	0.477121	0	0
Orientasi	0.477121	0	0
Pandu	0	0.477121	0
Pasien	0.477121	0	0
Pattern	0.477121	0	0
Pemrograman	0.477121	0	0
Penting	0.477121	0	0
Penuh	0	0	0.477121
Peran	0.477121	0	0
Perlu	0	0.477121	0
Permasalahan	0	0	0.477121
Php	0.477121	0	0



Pihak	0	0	0.477121
Prediksi	0	0	0.477121
Proses	0.176091	0	0.176091
PT	0	0.954242	0
Putus	0	0.477121	0
Rancang	0	0	0.477121
Rehabilitasi	0.477121	0	0
Rekomendasi	0	0.477121	0
Sedia	0	0	0.954242
Sejajar	0	0.477121	0
Seksi	0.477121	0	0
Selalu	0	0	0.477121
Semester	0	0	0.477121
Sendiri	0	0.477121	0
Sepi	0	0	0.477121
Seringkali	0	0	0.477121
Sesuai	0.176091	0.176091	0
Siklus	0.477121	0	0
Sistem	0	0	0
Situasi	0.477121	0	0
Skenario	0.477121	0	0
Sks	0	0	0.954242
Strategi	0.477121	0	0
Suatu	0	0	0.477121
Tahu	0	0	0.477121
Tambah	0	0	0.477121
Teliti	0	0	0
Terap	0.176091	0	0.176091
Ti	0	0.477121	0
Timbul	0	0	0.477121
Tingkat	0	0.176091	0.176091



Tuju	0	0	0
Tulis	0.477121	0	0
Uji	0.477121	0	0
Usaha	0	1.431363	0
Utama	0	0.477121	0
Waktu	0	0	0.477121

### 5.1.9 Klasifikasi Bidang Skripsi

Pemberian rekomendasi dosen pembimbing mempertimbangkan kesesuaian bidang pra-skripsi dengan bidang calon dosen. Analisis bidang skripsi didapatkan dari proses klasifikasi bidang skripsi yang sesuai dengan latar belakang. Untuk melakukan perkiraan kelas bidang skripsi dibutuhkan dua data terlebih dahulu. Data pertama adalah data latar belakang skripsi yang telah diproses melalui *pre-processing* dan diberi bobot per kata menggunakan TF-IDF. Data kedua adalah kelas bidang/kjfd yang menampilkan bidang/kjfd dari latar belakang skripsi tersebut. Kelas bidang/kjfd disesuaikan dengan kjfd yang terdapat pada Program Studi Sistem Informasi Universitas Brawijaya, diantaranya: Pengembangan Sistem Informasi (PSI), Tata Kelola dan Manajemen SI (TKMSI), Manajemen Data dan Informasi (MDI), Integrasi Teknologi Informasi (ITI), Sistem Informasi Geografis (SIG).

Bidang skripsi diklasifikasikan berdasarkan kelompok kelas yang telah ada. Pengelompokan terhadap kelas yang telah ada dilakukan dengan menggunakan pendekatan *supervised learning*. Pendekatan *supervised learning* pada penelitian ini dilakukan dengan mengimplementasikan algoritme *Multi-Class Support Vector Machine* pada bahasa pemrograman *Python* menggunakan *library* untuk *machine learning* bernama *scikit-learn*. *Multi-Class Support Vector Machine* pada *scikit-learn* mengimplementasikan pendekatan "*one-against-one*" untuk klasifikasi dengan kelas yang berjumlah lebih dari 2. Klasifikasi dibagi menjadi dua langkah, langkah pertama adalah pemilihan *model* menggunakan *GridSearchCV* dan langkah kedua adalah pengklasifikasian menggunakan *Support Vector Machine*.

#### 5.1.9.1 Pemilihan Model

Pemilihan model dilakukan untuk menganalisa dan memilih model yang paling sesuai dengan data yang dimiliki. Pemilihan model menggunakan *method GridSearchCV* yang telah disediakan oleh *library sklearn*. *GridSearchCV* mencari secara *exhaustive* nilai parameter terbaik untuk estimator atau klasifikator yang akan digunakan. *GridSearchCV* menguji setiap parameter dengan data yang diberikan. Dalam proses pengujian, *GridSearchCV* mengaplikasikan metode *Cross Validation* untuk memvalidasi akurasi dengan pembagian data yang berbeda.



### 5.1.9.2 Klasifikasi Bidang Skripsi Menggunakan *Support Vector Machine*

Klasifikasi bidang atau kjfd skripsi dilakukan menggunakan *Support Vector Machine*. *Support Vector Machine* mengklasifikasikan bidang skripsi berdasarkan data latar belakang yang telah diubah menggunakan TF-IDF. *Support Vector Machine* (SVM) yang digunakan adalah SVM dengan *kernel radial basis function* (rbf), C sebesar 1000, dan *gamma* sebesar 0,001.

### 5.1.10 Penentuan Nilai BM25

Penentuan nilai BM25 dilakukan untuk menjadi salah satu kriteria pertimbangan dalam pembuatan rekomendasi dosen pembimbing. Nilai bm25 untuk setiap alternatif dosen ditentukan oleh *web service* yang diimplementasikan menggunakan bahasa *python* dan menggunakan *library* `rank_bm25`. *Library* `rank_bm25` dapat diakses pada [https://github.com/dorianbrown/rank\\_bm25](https://github.com/dorianbrown/rank_bm25).

*Library* ini mengimplementasikan algoritme BM25 yang terdapat pada paper "*Improvements to BM25 and Language Models Examined*" yang dikarang oleh Trotman, Puurula dan Burgess pada tahun 2014.

*Library* `rank_bm25` menerima *query* yang telah ditokenisasi sebagai input *query* dan latar belakang skripsi sebagai *corpus* atau kumpulan dokumen yang dinilai kedekatannya dengan *query*. Metode yang digunakan pada penelitian ini adalah metode `get_scores`. Metode `get_scores` menerima input *query* yang kemudian akan diperbandingkan dengan latar belakang yang ada untuk dinilai kedekatannya dan akan mengembalikan *array* berisi nilai kedekatan dari setiap dokumen yang diperbandingkan.

Hasil nilai BM25 yang dimiliki setiap skripsi yang telah didapatkan menggunakan metode `get_score` kemudian dikelompokkan menurut dosen. Nilai tertinggi pada setiap kelompok dosen akan merepresentasikan nilai BM25 dari dosen tersebut. Nilai BM25 dosen tersebut yang kemudian akan menjadi salah satu kriteria pada pengambilan keputusan.

### 5.1.11 Penentuan Bobot Kriteria

Penelitian ini menggunakan *Weighted Product* untuk melakukan penentuan pembimbing skripsi. *Weighted product* mempertimbangkan atribut atau kriteria yang diberi beban secara subyektif sesuai dengan tingkat kepentingan dari sebuah atribut atau kriteria. Salah satu metode yang dapat digunakan untuk menentukan beban kriteria adalah *pairwise comparison*. Dengan *pairwise comparison*, setiap pasang atribut diperbandingkan untuk mendapatkan tingkat kepentingan antar satu dengan yang lain. Berikut kriteria *Weighted Product* untuk sistem rekomendasi dosen pembimbing dijelaskan pada tabel 5.9

**Tabel 5.9 Kriteria *Weighted Product* untuk Sistem Rekomendasi Dosen Pembimbing**

Kode	C1	C2	C3	C4	C5	C6
Nama Variabel	Kesesuaian KJFD	Nilai BM25	Jurusan Dosen	Sisa Kuota Bimbingan	Tingkatan Gelar	Beban Kerja



**5.1.11.1 Pembuatan Matriks dan Pemberian Nilai Intensitas**

Langkah pertama pada proses *pairwise comparisons* adalah pembuatan matriks atribut atau kriteria yang diisi dengan nilai intensitas antar atribut. Untuk mempermudah pembuatan matriks, nama kriteria perlu untuk dipersingkat. Kesesuaian KJFD direpresentasikan menjadi KKJFD. Nilai BM25 direpresentasikan menjadi BM25. Jurusan Dosen direpresentasikan menjadi JD. Sisa kuota bimbingan direpresentasikan menjadi SKKB. Tingkatan gelar dosen direpresentasikan menjadi TG. Beban kerja dosen direpresentasikan menjadi BK. Berikut matriks intensitas kriteria menggunakan *pairwise comparison* pada tabel 5.10

**Tabel 5.10 Matriks Intensitas Kriteria Menggunakan *Pairwise Comparison***

	KKJFD	BM25	JD	SKKB	TG	BK
KKJFD	1	2	5	6	7	9
BM25	1/2	1	3	3	5	7
JD	1/5	1/3	1	2	2	3
SKB	1/6	1/3	1/2	1	2	1
TG	1/7	1/5	1/2	1/2	1	1
BK	1/9	1/7	1/3	1	1	1
TOTAL	2.12	4.01	10.33	13.5	18	22

**5.1.11.2 Perhitungan Beban Kriteria**

Perhitungan beban kriteria atau yang disebut juga nilai prioritas atau *principal eigenvector* dilakukan berdasarkan matriks intensitas kriteria yang telah dibuat. Perhitungan beban kriteria dilakukan dengan cara menjumlahkan setiap kolom yang kemudian dibagi dengan nilai total setiap kolom, kemudian jumlah nilai yang telah ternormalisasi untuk setiap barisnya dibagi dengan jumlah kriteria. Berikut perhitungan beban untuk setiap kriteria:

- a. **KKJFD** :  $\left( \frac{1}{2.12} + \frac{2}{4.01} + \frac{5}{10.33} + \frac{6}{13.5} + \frac{7}{18} + \frac{9}{22} \right) / 6 = 0.44948376 = \mathbf{0,45}$
- b. **BM25** :  $\left( \frac{1/2}{2.12} + \frac{1}{4.01} + \frac{1}{10.33} + \frac{3}{13.5} + \frac{5}{18} + \frac{7}{22} \right) / 6 = 0.26563728 = \mathbf{0,27}$
- c. **JD** :  $\left( \frac{1/5}{2.12} + \frac{1/3}{4.01} + \frac{1}{10.33} + \frac{2}{13.5} + \frac{2}{18} + \frac{3}{22} \right) / 6 = 0.11164890 = \mathbf{0,11}$
- d. **SKB** :  $\left( \frac{1/6}{2.12} + \frac{1/3}{4.01} + \frac{1/2}{10.33} + \frac{1}{13.5} + \frac{2}{18} + \frac{1}{22} \right) / 6 = 0.07346405 = \mathbf{0,07}$
- e. **TG** :  $\left( \frac{1/7}{2.12} + \frac{1/6}{4.01} + \frac{1/2}{10.33} + \frac{1/2}{13.5} + \frac{1}{18} + \frac{1}{22} \right) / 6 = 0.04923300 = \mathbf{0,05}$
- f. **BK** :  $\left( \frac{1/9}{2.12} + \frac{1/7}{4.01} + \frac{1/3}{10.33} + \frac{1}{13.5} + \frac{1}{18} + \frac{1}{22} \right) / 6 = 0.04923146 = \mathbf{0,05}$

**5.1.11.3 Perhitungan Rasio Konsistensi (CR)**

Perhitungan rasio konsistensi dibutuhkan untuk mengetahui apakah kriteria yang telah dibuat memiliki konsistensi dalam proses pembuatan beban. Langkah



pertama yang perlu dilakukan dalam penghitungan rasio konsistensi adalah dengan memperkalikan nilai setiap baris pada matriks dengan nilai beban kriteria atau nilai prioritas kemudian jumlah hasil perkalian setiap baris tersebut dibagi dengan beban mereka. Hasil perhitungan langkah pertama sebagai berikut:

- a. **KKJFD** :  $(1 \times 0,45 + 2 \times 0,27 + 5 \times 0,11 + 6 \times 0,07 + 7 \times 0,05 + 9 \times 0,05) / 0,45 = 6,13$
- b. **BM25** :  $(1/2 \times 0,45 + 1 \times 0,27 + 3 \times 0,11 + 3 \times 0,07 + 5 \times 0,05 + 7 \times 0,05) / 0,27 = 6,05$
- c. **JD** :  $(1/5 \times 0,45 + 1/3 \times 0,27 + 1 \times 0,11 + 2 \times 0,07 + 2 \times 0,05 + 3 \times 0,05) / 0,11 = 6,18$
- d. **SKB** :  $(1/6 \times 0,45 + 1/3 \times 0,27 + 1/2 \times 0,11 + 1 \times 0,07 + 2 \times 0,05 + 1 \times 0,05) / 0,07 = 6,28$
- e. **TG** :  $(1/7 \times 0,45 + 1/6 \times 0,27 + 1/2 \times 0,11 + 1/2 \times 0,07 + 1 \times 0,05 + 1 \times 0,05) / 0,05 = 5,98$
- f. **BK** :  $(1/9 \times 0,45 + 1/7 \times 0,27 + 1/3 \times 0,11 + 1 \times 0,07 + 1 \times 0,05 + 1 \times 0,05) / 0,05 = 5,9$

Langkah kedua yang perlu dilakukan dalam perhitungan rasio konsistensi adalah dengan cara menghitung rata-rata dari hasil perhitungan pada langkah pertama. Hasil perhitungan langkah kedua sebagai berikut:

$$\frac{6.13 + 6.05 + 6.18 + 6.28 + 5.98 + 5.9}{6} = 6.08$$

Langkah ketiga yang perlu dilakukan dalam perhitungan rasio konsistensi adalah mencari indeks konsistensi (CI). Hasil perhitungan rata-rata pada langkah kedua merepresentasikan  $\lambda_{max}$ . Rumus indeks konsistensi (CI) dapat dilihat pada persamaan 4.1

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{4.1}$$

Berikut hasil perhitungan indeks konsistensi:

$$\frac{6,08 - 6}{6 - 1} = 0,016$$

Langkah keempat yang perlu dilakukan dalam perhitungan rasio konsistensi adalah mencari *Random Index* (RI). *Random Index* yang diperlukan adalah yang urutannya berada sesuai dengan jumlah kriteria yang menjadi pertimbangan pada *pairwise comparison*. Angka *Random Index* dapat dilihat pada tabel 5.11

**Tabel 5.11 Angka Random Index**

Order	1	2	3	4	5	6	7	8	9	10
RI	0,00	0,00	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

Sumber: Saaty dan Vargas (2001)



Berdasarkan tabel angka random index, maka didapatkan random index sebesar **1,24**. Langkah kelima adalah menghitung Rasio Konsistensi. Rumus rasio konsistensi (CR) dapat dilihat pada persamaan 4.2

$$CR = \frac{CI}{RI} \tag{4.2}$$

Berikut hasil perhitungan rasio konsistensi:

$$\frac{0,016}{1,24} = 0,012$$

Dikarenakan hasil perhitungan rasio konsistensi (CR) menghasilkan nilai dibawah 0,1 maka *pairwise comparison* yang dilakukan telah menghasilkan nilai yang konsisten dan beban yang telah diberikan dapat digunakan.

### 5.1.12 Perhitungan *Weighted Product*

Penentuan pembimbing dosen termasuk dalam permasalahan pengambilan keputusan multi atribut/kriteria. *Weighted Product* adalah salah satu metode *Multi-attribute Decision Making* (MCDM) yang dapat diterapkan. *Weighted Product* memiliki beban yang telah ditentukan pada subbab 4.13.5. Berikut Kriteria dan Beban yang digunakan *weighthed product* pada tabel 5.12

**Tabel 5.12 Kriteria dan Beban Kriteria *Weighted Product***

Kode	C1	C2	C3	C4	C5	C6
<b>Nama Variabel</b>	Kesesuaian KJFD	Nilai BM25	Jurusan Dosen	Sisa Kuota Bimbingan	Tingkatan Gelar	Beban Kerja
<b>Beban</b>	0,45	0,27	0,11	0,07	0,05	0,05

Keterangan dan nilai dari masing-masing kriteria:

C1: Kesesuaian antara KJFD/bidang dengan bidang penelitian mahasiswa, prediksi penelitian mahasiswa berasal dari klasifikasi pada tahapan sebelumnya. Aturan pembobotan untuk C1 dapat dilihat pada tabel 5.13. Bobot C1 kemudian dinormalisasikan dengan bobot tertinggi.

**Tabel 5.13 Aturan Pembobotan Kesesuaian KJFD**

Nilai	Bobot
Sesuai Mayor	20
Sesuai Minor	10
Tidak Sesuai Dengan Mayor atau Minor	0

C2: Nilai BM25 tertinggi pada setiap dosen. Nilai BM25 yang didapatkan setiap dosen dinormalisasikan dengan nilai BM25 dosen tertinggi.

C3: Jurusan dosen tersebut. Aturan pembobotan untuk C3 dapat dilihat pada tabel 5.14. Bobot C3 kemudian dinormalisasikan dengan nilai C3 tertinggi.

**Tabel 5.14 Aturan Pembobotan Jurusan Dosen**

Nilai	Bobot
-------	-------



Sistem Informasi	20
Lainnya	10

C4: Sisa kuota bimbingan dosen. Aturan pembobotan untuk C4 mengikuti nilai dari data sebenarnya yang dinormalisasi dengan sisa kuota bimbingan tertinggi.

C5: Tingkat gelar dosen. Aturan pembobotan untuk C5 dapat dilihat pada tabel 5.15. Kemudian bobot C5 dinormalisasikan dengan bobot C5 tertinggi.

**Tabel 5.15 Aturan Pembobotan Gelar Dosen**

Nilai	Bobot
Doktor	10
Master	7

C6: Beban kerja dosen(sks). Aturan pembobotan untuk C6 mengikuti nilai dari data sebenarnya yang kemudian dinormalisasikan dengan nilai C6 tertinggi. Dikarenakan C6 adalah kriteria beban, maka bobot bernilai negatif.

$$P(A_k) = \prod_{j=1}^n (a_{kj})^{w_j}$$

Perhitungan *weighted product* setiap alternatif didasarkan pada persamaan 2.3. Dikarenakan jumlah dan bobot setiap kriteria telah diketahui, maka perhitungan *product* setiap alternatif dosen sebagai berikut:

$$P(A_k) = C1^{0,45} \times C2^{0,27} \times C3^{0,11} \times C4^{0,07} \times C5^{0,05} \times C6^{-0,05}$$

Dari hasil perhitungan *product*, diambil lima alternatif dengan nilai *product* tertinggi. Kelima alternatif dosen pembimbing tersebut yang akan menjadi rekomendasi yang dapat digunakan oleh Koordinator KJFD atau Kaprodi untuk calon mahasiswa skripsi.

## 5.2 Implementasi Sistem

Sistem ini diimplementasikan menggunakan Bahasa Pemrograman Python dan PHP. Bahasa PHP digunakan untuk implementasi halaman web sistem rekomendasi dosen pembimbing. Bahasa Python digunakan untuk implementasi *web service*.

### 5.2.1 Implementasi Text Processing

*Text processing* menjadi langkah pertama dalam pemrosesan rekomendasi dosen pembimbing. Implementasi ini dilakukan pada *web service*. Bahasa yang digunakan untuk implementasi ini adalah Python. Metode *text\_proc* membutuhkan sebuah variabel argumen bernama *sentence* yang berisi text latar belakang yang masih belum mengalami pemrosesan teks. Berikut baris kode metode *text\_proc* sebagai pemroses *text processing*:

<pre> Algorithm Web Service: Metode text_proc                 </pre>
--



```

1 def text_proc(sentence):
2     sentence = sentence.lower()
3     sentence = re.sub(r"d+", "", sentence)
4     sentence = sentence.translate(str.maketrans("", "", string.punctuation))
5     sentence = sentence.strip()
6     sentence = "".join(sentence.split())
7     stemmingFactory = StemmerFactory()
8     stemmer = stemmingFactory.create_stemmer()
9     sentence = stemmer.stem(sentence)
10    difference_length = 1
11    difference = []
12    filterFactory = StopWordRemoverFactory()
13    stopwordsRemover = filterFactory.create_stop_word_remover()
14    while difference_length>0:
15        sentenceWithoutStopword = stopwordsRemover.remove(sentence)
16        sentenceToken = nltk.tokenize.word_tokenize(sentence)
17        withoutStopwordToken =
18            nltk.tokenize.word_tokenize(sentenceWithoutStopword)
19        sentence = sentenceWithoutStopword
20        difference.extend(list(set(sentenceToken) -
21            set(withoutStopwordToken)))
21        difference_length = len(list(set(sentenceToken) -
22            set(withoutStopwordToken)))
23        return sentence

```

Proses *text processing* dibantu menggunakan *library* Sastrawi dan Natural Language Toolkit (NLTK). Objek *StemmerFactory* pada baris 7 adalah objek yang disediakan oleh Sastrawi untuk melakukan proses *stemming*. Objek *StopWordRemoverFactory* pada baris 12 adalah objek yang disediakan oleh Sastrawi untuk melakukan proses penghapusan *stopword*. Metode *word\_tokenize* pada baris 16 dan 17 adalah metode yang disediakan oleh NLTK untuk melakukan proses tokenisasi kata.

### 5.2.2 Implementasi Prediksi

*Prediksi* menjadi langkah utama dalam pemrosesan rekomendasi dosen pembimbing. Pada implementasi prediksi terdapat 3 bagian besar, yaitu: pencarian nilai BM25, klasifikasi KJFD menggunakan *Support Vector Machine*, dan prediksi dosen pembimbing menggunakan *Weighted Product*. Implementasi prediksi dilakukan pada *web service*. Implementasi dilakukan menggunakan bahasa Python. Metode *predict* membutuhkan sebuah variabel argument bernama *latar\_belakang* yang berisi latar belakang telah ditokenisasi pada metode *text\_proc*. Berikut baris kode metode *predict* sebagai pemroses prediksi dosen pembimbing skripsi:

Algoritme *Web Service*: Metode *predict*

```

1 def predict(latar_belakang) :
2     np.random.seed(500)

```



```

3 Corpus = pd.read_csv("./skripsi.csv", sep=";")
4 lb_tokenized = [word_tokenize(str(entry)) for entry in
Corpus['stemmed']]
5 #BM25
6 bm25 = BM25Plus(lb_tokenized)
7 tokenized_query = latar_belakang.split(" ")
8 bm25_score = bm25.get_scores(tokenized_query)
9 for index,entry in enumerate(Corpus['stemmed']):
10 Corpus.loc[index,'bm_25'] = bm25_score[index]
11 dosen_bm25 = Corpus.groupby('id_dosen', sort=True)['bm_25'].max()
12 dosen_bm25_max = dosen_bm25.max()
13 Corpus['stemmed'] = [entry.lower() for entry in Corpus['stemmed']]
14 Corpus['stemmed'] = [word_tokenize(str(entry)) for entry in
Corpus['stemmed']]
15 Corpus['stemmed'] = [str(entry) for index,entry in
enumerate(Corpus['stemmed'])]
16 #Support Vector Machine
17 Test_X = [latar_belakang]
18 Test_X = pd.DataFrame(Test_X, columns=["stem"])
19 Train_X = Corpus['stemmed']
20 Train_Y = Corpus['kjfd']
21 Encoder = LabelEncoder()
22 Train_Y = Encoder.fit_transform(Train_Y)
23 Tfidf_vect = TfidfVectorizer(max_features=1000)
24 Tfidf_vect.fit(Corpus['stemmed'])
25 Train_X_Tfidf = Tfidf_vect.transform(Train_X)
26 Test_X_Tfidf = Tfidf_vect.transform(Test_X['stem'])
27 smote = SMOTE('minority')
28 x_sm, y_sm = smote.fit_sample(Train_X_Tfidf,Train_Y)
29 final_model = svm.SVC(C=1000,kernel='rbf',gamma=0.001)
30 final_model.fit(x_sm,y_sm)
31 Y_pred = final_model.predict(Test_X_Tfidf)
32 Y_pred_label = list(Encoder.inverse_transform(Y_pred))
33 conn =
psycopg2.connect(host=db_host,user=db_username,password=db_pass,
database=db_name)
34 cur = conn.cursor()
35 cur.execute("SELECT * FROM dosen")
36 tuples = cur.fetchall()
37 headers = [entry[0] for entry in cur.description]
38 Dosen = pd.DataFrame(list(tuples),columns=headers)
39 cur.close()
40 conn.close()
41 skb_max = Dosen['kuota_skripsi'].max()
42 bk_max = Dosen['beban_kerja'].max()

```



```

43 for index, entry in enumerate(dosen_bm25):
44     Dosen.loc[index,'bm_25'] = entry
45     dosen_product = []
46     #Weighted Product
47     for dosen_index, dosen_row in Dosen.iterrows():
48         bm25_value_normalized = dosen_row['bm_25']/dosen_bm25_max
49         Dosen.loc[dosen_index,'bm_25'] = bm25_value_normalized
50         #C1 = Kesesuaian KJFD
51         minor = []
52         minor.append(dosen_row['id_minor1']) if not
math.isnan(dosen_row['id_minor1']) else minor.append(0)
53         minor.append(dosen_row['id_minor_2']) if not
math.isnan(dosen_row['id_minor_2']) else minor.append(0)
54         minor.append(dosen_row['id_minor_3']) if not
math.isnan(dosen_row['id_minor_3']) else minor.append(0)
55         C1 = 0
56         if Y_pred_label[0] == dosen_row['id_mayor']:
57             C1=1
58         elif Y_pred_label[0] in minor:
59             C1=0.5
60         #C2 = BM25 Dosen
61         C2 = bm25_value_normalized
62         #C3 = Jurusan Dosen
63         C3 = 0.5
64         if dosen_row['jurusan_1'] == 'Sistem Informasi' or
dosen_row['jurusan_2'] == 'Sistem Informasi':
65             C3 = 1
66         #C4 = Sisa Kuota Bimbingan
67         skb_normalized = dosen_row['kuota_skripsi']/skb_max
68         C4 = skb_normalized
69         #C5 = Tingkatan Gelar Dosen
70         C5 = 0.7
71         if dosen_row['tingkatan_gelar'] == "Doktor":
72             C5 = 1
73         elif dosen_row['tingkatan_gelar'] == "Master":
74             C5 = 0.7
75         #C6 = Beban Kerja
76         bk_normalized = dosen_row['beban_kerja']/bk_max
77         C6 = bk_normalized
78         bobot_relatif_kriteria = [0.45,0.27,0.11,0.07,0.05,0.05]
79         nama_bobot =['kkjfd','bm25','jd','skkb','tg','bk']
80         W1 = C1*bobot_relatif_kriteria[0]
81         W2 = C2*bobot_relatif_kriteria[1]
82         W3 = C3*bobot_relatif_kriteria[2]
83         W4 = C4*bobot_relatif_kriteria[3]

```



```

84 W5 = C5**bobot_relatif_kriteria[4]
85 W6 = C6**bobot_relatif_kriteria[5] if C6!=0 else 0
86 product = W1 * W2 * W3 * W4 * W5 * W6
87 Dosen.loc[dosen_index,'product'] = product
88 Dosen = Dosen.sort_values(by=['product'],
89 ascending=False).head(5).reset_index()
90 Dosen_json = Dosen.to_json(orient='records')
91 Dosen_array = json.loads(Dosen_json)
92 returnVal = {}
93 returnVal['kjfd'] = int(Y_pred_label[0])
94 returnVal['dosen'] = Dosen_array
return returnVal

```

Proses prediksi dibantu menggunakan *library* rank-bm25, sklearn, dan psycpg2. Objek BM25Plus pada baris 6 adalah objek yang disediakan oleh rank-bm25 untuk melakukan perhitungan BM25 menggunakan algoritme BM25+. Metode SVC pada objek svm pada baris 29 adalah metode yang disediakan oleh NLTK untuk melakukan klasifikasi *Multi-class Support Vector Machine*. Objek psycpg2 pada baris 35 adalah objek yang disediakan *library* psycpg2 untuk mengakses *database* PostgreSQL pada bahasa Python.

### 5.2.3 Implementasi Tampilan Antarmuka

Implementasi tampilan antarmuka merepresentasikan *view* pada arsitektur MVC. Tampilan antarmuka dibangun berdasarkan template AdminLTE dengan menggunakan Bootstrap sebagai *guidelinenya*.

#### 5.2.3.1 Halaman Login

Implementasi halaman *login* dibangun dengan memodifikasi template halaman *login* AdminLTE. Halaman ini berisi *field input email/username* dengan tipe teks, *field input password* dengan tipe *password*, dan tombol *login* untuk melakukan *submit* form. Implementasi tampilan antarmuka halaman *login* dapat dilihat pada gambar 5.18

Gambar 5.18 Tampilan Antarmuka Halaman Login



### 5.2.3.2 Halaman Registrasi

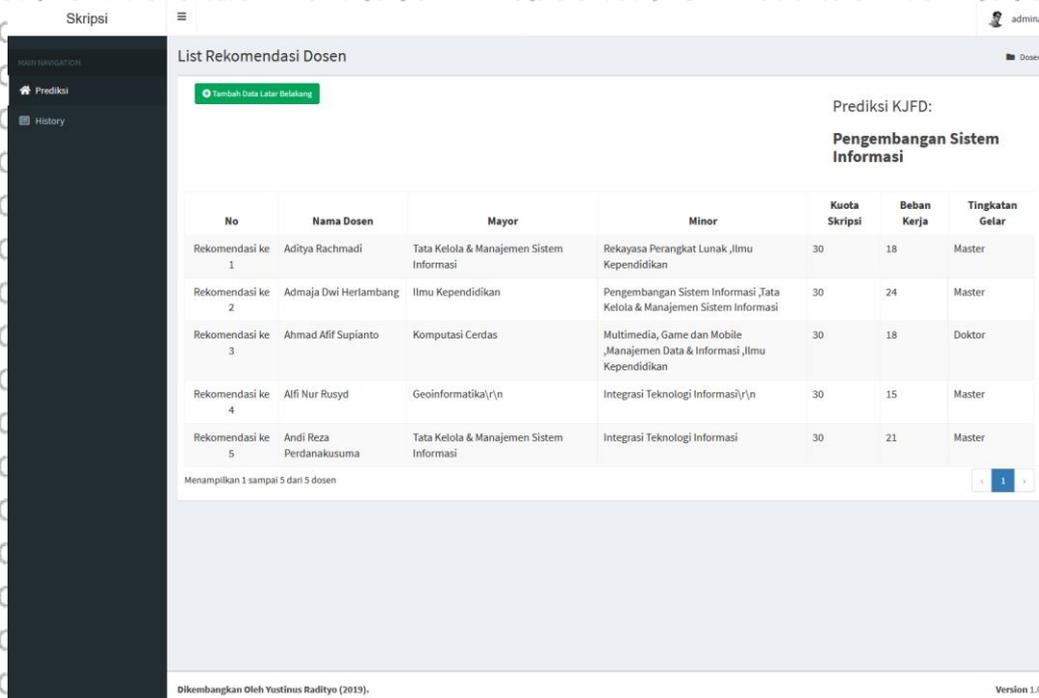
Implementasi halaman *registrasi* dibangun dengan memodifikasi template halaman *login* AdminLTE. Halaman ini berisi *field input username* dengan tipe teks, *field input* nama dengan tipe teks, *field input* NIP dengan tipe teks, *field input email* dengan tipe *email*, *field input password* dengan tipe *password*, *field input* ulang kembali *password* dengan tipe *password*, dan tombol *login* untuk melakukan *submit form*. Implementasi tampilan antarmuka halaman registrasi dapat dilihat pada gambar 5.19

The screenshot shows a registration form titled "Rekomendasi Dosen". The form is titled "Registrasi" and contains the following fields: Username, Nama, NIP, Email, Password, and Ulang kembali password. There is a "Register" button and a link for "Login disini" if the user already has an account.

Gambar 5.19 Tampilan Antarmuka Halaman Registrasi

### 5.2.3.3 Halaman List Rekomendasi

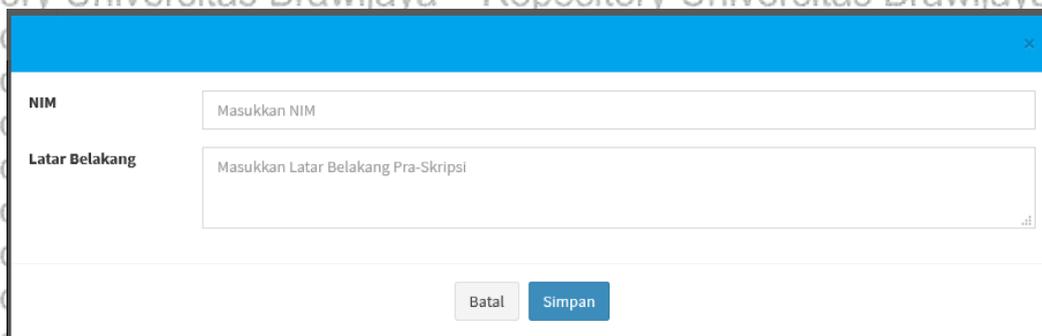
Implementasi halaman *list* rekomendasi dibangun dengan memodifikasi template halaman tabel pada AdminLTE. Halaman ini berisi tombol tambah latar belakang dan tabel yang menunjukkan hasil rekomendasi dari latar belakang yang telah ter-*submit*. Implementasi tampilan antarmuka halaman *list* rekomendasi dapat dilihat pada gambar 5.20



Gambar 5.20 Tampilan Antarmuka Halaman **List Rekomendasi**

### 5.2.3.4 Halaman **Submit** Latar Belakang

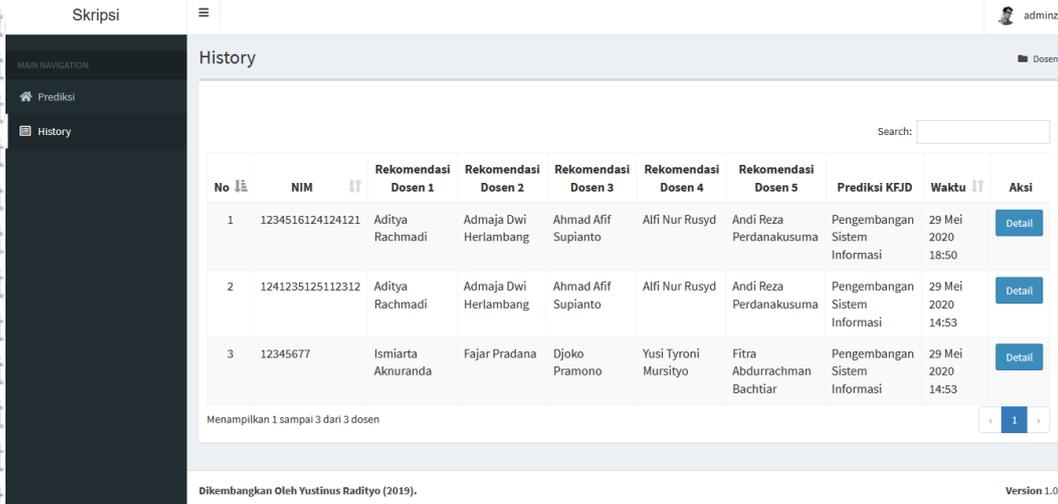
Implementasi halaman *registrasi* dibangun dengan memodifikasi template *modal* pada AdminLTE. Halaman ini berisi *field input* NIM, *field input* latar belakang, dan tombol Simpan. Implementasi tampilan antarmuka halaman login dapat dilihat pada gambar 5.21



Gambar 5.21 Tampilan Antarmuka Halaman **Submit** Latar Belakang

### 5.2.3.5 Halaman **List History** Rekomendasi

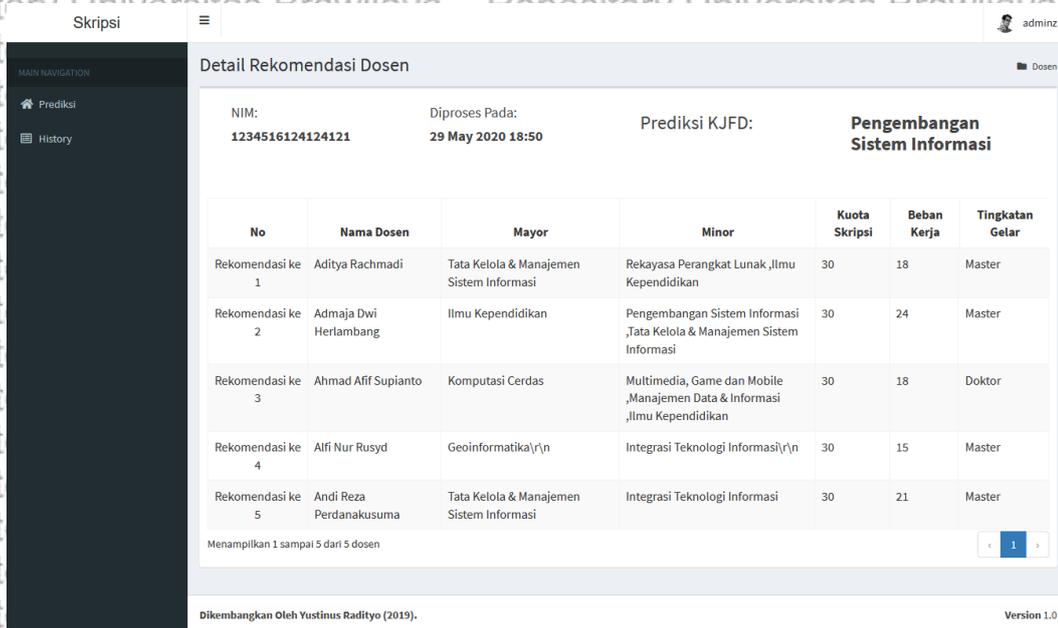
Implementasi halaman *list history* rekomendasi dibangun dengan memodifikasi template halaman tabel pada AdminLTE. Halaman ini berisi *field input* pencarian dan tabel yang menunjukkan *history* hasil rekomendasi dari latar belakang yang telah ter-*submit*. Implementasi tampilan antarmuka halaman *history list* rekomendasi dapat dilihat pada gambar 5.22



Gambar 5.22 Tampilan Antarmuka Halaman *List History* Rekomendasi

### 5.2.3.6 Halaman Detail *History* Rekomendasi

Implementasi halaman detail *history* rekomendasi dibangun dengan memodifikasi template halaman tabel pada AdminLTE. Halaman ini berisi data detail rekomendasi dan tabel yang menunjukkan detail *history* hasil rekomendasi dari latar belakang yang telah ter-*submit*. Implementasi tampilan antarmuka halaman *detail history* rekomendasi dapat dilihat pada gambar 5.23



Gambar 5.23 Tampilan Antarmuka Halaman detail *History* Rekomendasi



## BAB 6 PENGUJIAN

### 6.1 Pengujian Akurasi

Pengujian akurasi adalah hasil dari kesesuaian antara prediksi dengan data uji. Pengujian akurasi terbagi menjadi dua garis besar, yaitu: akurasi pada data uji 15% dan akurasi pada data uji 20%. Terdapat 3 hal yang diuji pada setiap pengujian akurasi, yaitu:

1. Akurasi *Support Vector Machine*. Pengujian akurasi SVM dilakukan untuk melihat tingkat akurasi antara prediksi klasifikasi jenis KJFD pada SVM yang telah dibangun dengan jenis KJFD pada data uji.
2. Akurasi BM25. Pengujian akurasi BM25 dilakukan untuk melihat tingkat akurasi antara prediksi lima nama dosen pembimbing dengan nilai BM25 tertinggi dengan nama dosen pembimbing pada data uji.
3. Akurasi *Weighted Product*. Pengujian akurasi WP dilakukan untuk melihat tingkat akurasi antara prediksi lima nama dosen pembimbing dengan nilai WP tertinggi dengan nama dosen pembimbing pada data uji.

Pengujian akurasi dilakukan dengan memperbandingkan jumlah dosen, terdapat 4 kelompok jumlah dosen yang diuji dengan keterangan sebagai berikut:

1. 26 Dosen. Jumlah 26 dosen adalah seluruh dosen yang membimbing Skripsi mahasiswa pada periode yudisium Januari 2018 hingga Agustus 2019;
2. 24 Dosen. Jumlah 24 dosen adalah dosen yang membimbing Skripsi mahasiswa pada periode yudisium Januari 2018 hingga Agustus 2019 dan jumlah bimbingan skripsinya lebih dari satu;
3. 23 Dosen. Jumlah 23 dosen adalah dosen yang membimbing Skripsi mahasiswa pada periode yudisium Januari 2018 hingga Agustus 2019 yang aktif dalam pembimbingan skripsi tahun ajaran 2019/2020;
4. 21 Dosen. Jumlah 21 dosen adalah dosen yang membimbing Skripsi mahasiswa pada periode yudisium Januari 2018 hingga Agustus 2019 yang aktif dalam pembimbingan skripsi tahun ajaran 2019/2020 dan jumlah bimbingan skripsinya lebih dari satu;

#### 6.1.2 Pengujian Akurasi Data Uji 15%

Pada pengujian ini, data uji dan data latih terbagi secara persentase. Pembagian data uji dan data latih sebanyak 15% data uji dan 85% data latih. Pembagian tersebut telah terstratifikasi berdasarkan data KJFD. Berikut hasil pengujian akurasi SVM, BM25, dan WP dengan data uji 15% dapat dilihat pada tabel 6.1

Tabel 6.1 Hasil Pengujian Akurasi SVM, BM25, dan WP Dengan Data Uji 15%

No	Jumlah Dosen	Akurasi SVM	Akurasi BM25	Akurasi WP
1	26 Dosen	93.05%	48.61%	45.83%



2	24 Dosen	93.05%	50%	48.61%
3	23 Dosen	92.75%	47.82%	52.17%
4	21 Dosen	89.85%	53.62%	55.07%

Berdasarkan hasil pengujian akurasi pada tabel 6.1 dapat dianalisa bahwa:

1. Semakin sedikit jumlah dosen, maka semakin tinggi tingkat akurasi *Weighted Product* yang menjadi acuan dalam merekomendasikan dosen;
2. Penurunan jumlah dosen tidak mempengaruhi akurasi BM25. Namun, penghilangan data outlier (dosen dengan bimbingan kurang dari 2) meningkatkan akurasi BM25;
3. Semakin sedikit jumlah dosen, maka semakin rendah tingkat akurasi *Support Vector Machine* yang menjadi acuan dalam prediksi KJFD;

### 6.1.3 Pengujian Akurasi Data Uji 20%

Pada pengujian ini, data uji dan data latih terbagi secara presentase. Pembagian data uji sebesar 20% dari data seluruhnya dan data latih sebesar 80% dari data seluruhnya. Pembagian dilakukan dengan menstratifikasi data berdasarkan KJFD. Berikut hasil pengujian akurasi SVM, BM25, dan WP dengan data uji 20% dapat dilihat pada tabel 6.2

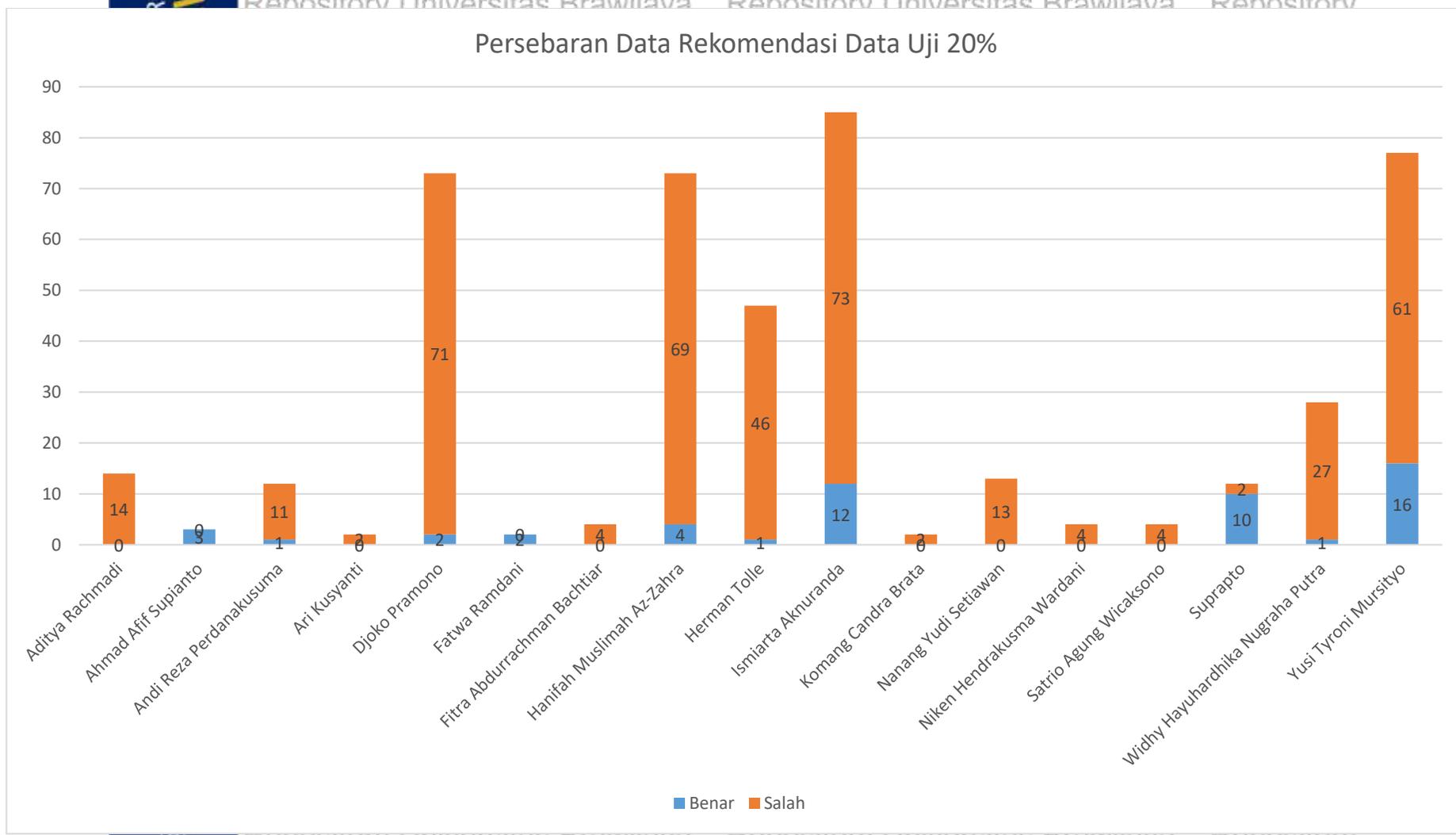
**Tabel 6.2 Hasil Pengujian Akurasi SVM, BM25, dan WP Dengan Data Uji 20%**

No	Jumlah Dosen	Akurasi SVM	Akurasi BM25	Akurasi WP
1	26 Dosen	93.75%	54.16%	47.91%
2	24 Dosen	91.58%	47.36%	49.47%
3	23 Dosen	90.21%	53.26%	52.17%
4	21 Dosen	91.20%	52.74%	57.14%

Berdasarkan hasil pengujian akurasi pada tabel 6.2 dapat dianalisa bahwa:

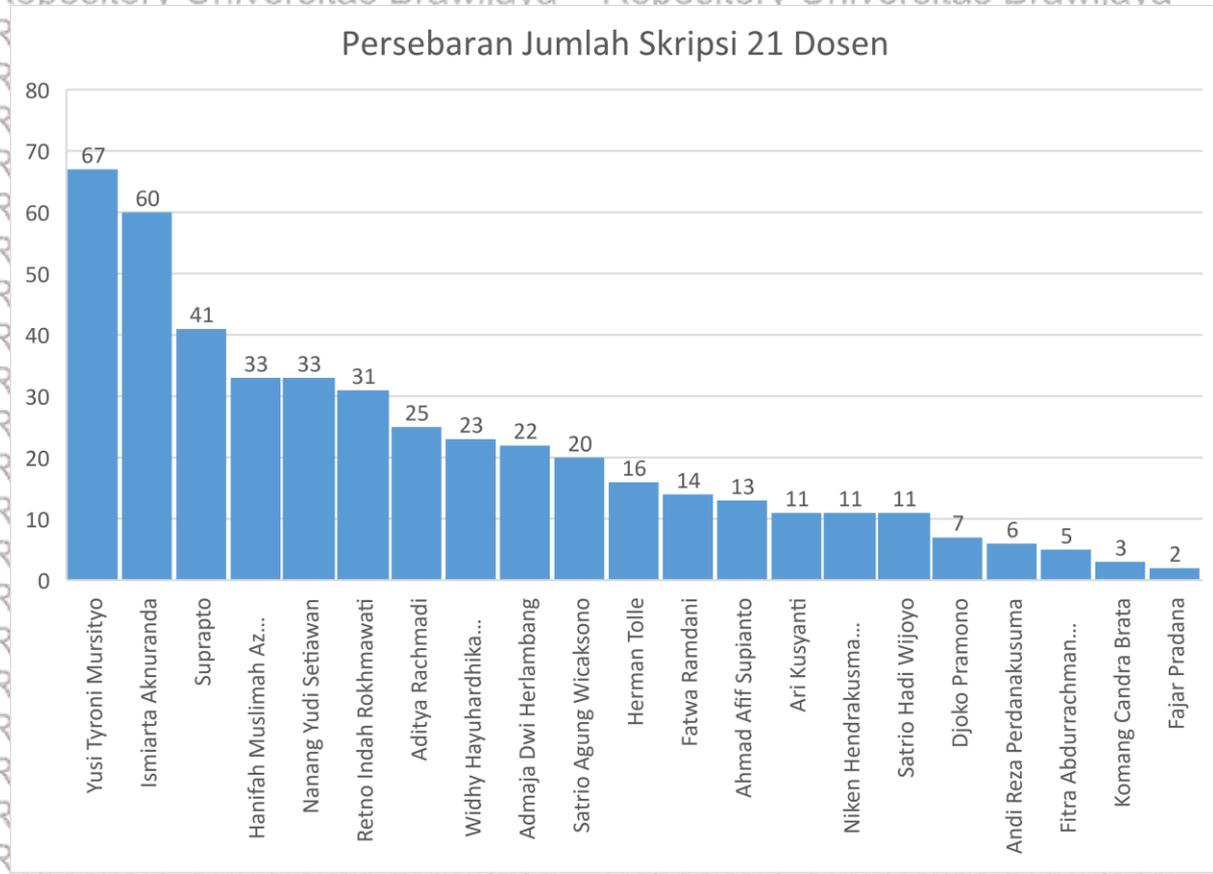
1. Akurasi *Weighted Product* yang menjadi akurasi prediksi dosen pembimbing pada data uji 20% lebih tinggi diperbandingkan dengan akurasi pada data uji 15%
2. Penurunan jumlah dosen meningkatkan akurasi *Weighted Product* yang menjadi acuan dalam merekomendasikan dosen;

Dikarenakan data uji 20% memiliki tingkat akurasi *weighted product* yang lebih tinggi, maka dapat dilakukan analisis yang lebih mendalam. Persebaran data rekomendasi yang diberikan dapat dilihat pada gambar 6.1.



**Gambar 6.1 Persebaran Data Rekomendasi pada Data Uji 20%**

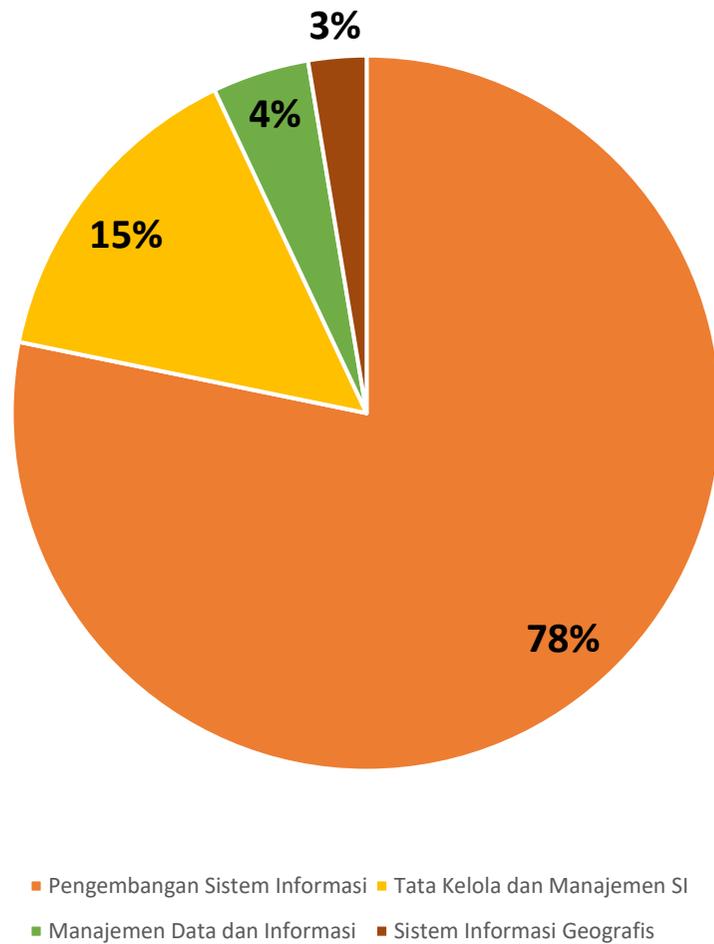
Pada gambar 6.1 dijelaskan persebaran data rekomendasi dosen pada data uji sebanyak 20% dari jumlah data. Data ini tidak menunjukkan akurasi, melainkan seberapa banyak dosen tersebut muncul dalam hasil rekomendasi. Dari 26 nama yang diperbandingkan, 17 nama muncul dalam rekomendasi dosen pembimbing. Terdapat penumpukan kumpulan nama yang direkomendasikan lebih dari 30 kali, diantaranya: Djoko Pramono, Hanifah Muslimah, Herman Tolle, Ismiarta Aknuranda, Widhy Wahyuhardika, dan Yusi Tyroni.



**Gambar 6.2 Persebaran Jumlah Skripsi pada 21 Dosen**

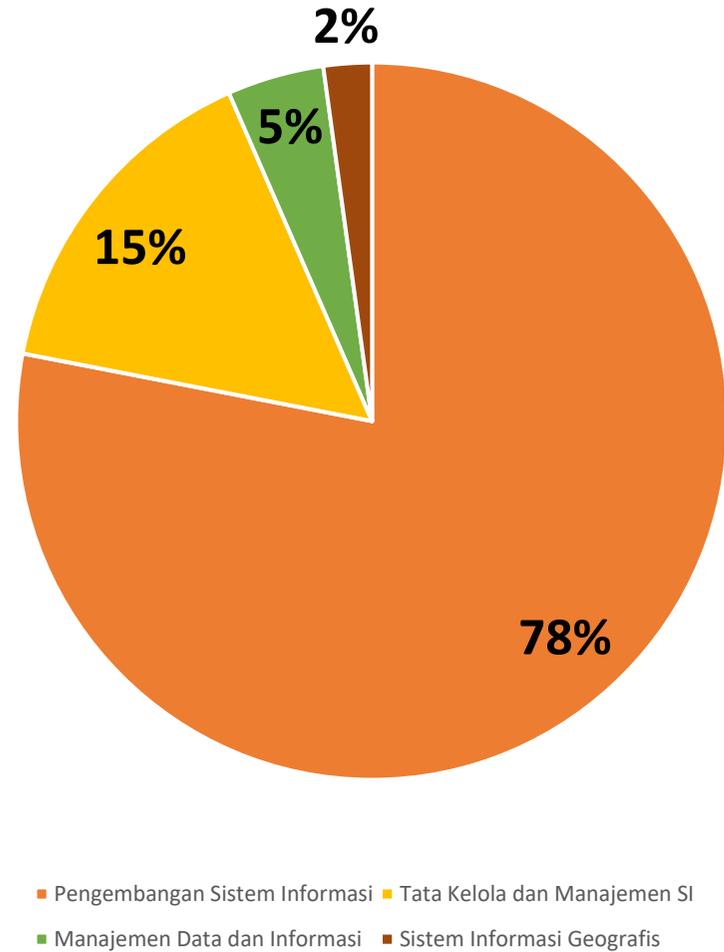
Pada gambar 6.2 dijelaskan persebaran jumlah skripsi yang dibimbing dosen dengan jumlah dosen 21. Data menunjukkan penumpukan jumlah skripsi pada Yusi Tyroni, Ismiarta Aknuranda, Suprpto, Hanifah Muslimah, Nanang Yudi, dan Retno Indah. Namun, data Suprpto, Retno Indah, dan Nanang Yudi tidak mendominasi data dosen yang direkomendasikan. Data Djoko Pramono yang mendominasi data rekomendasi terletak pada jajaran bawah jumlah skripsi yang dibimbing. Berdasarkan data tersebut dapat disimpulkan bahwa jumlah data bimbingan skripsi tidak mempengaruhi jumlah rekomendasi.

Pembagian Jumlah Skripsi Per KJFD



Gambar 6.3 Pembagian Jumlah Skripsi Per KJFD

Pembagian Jumlah Data Uji Per KJFD

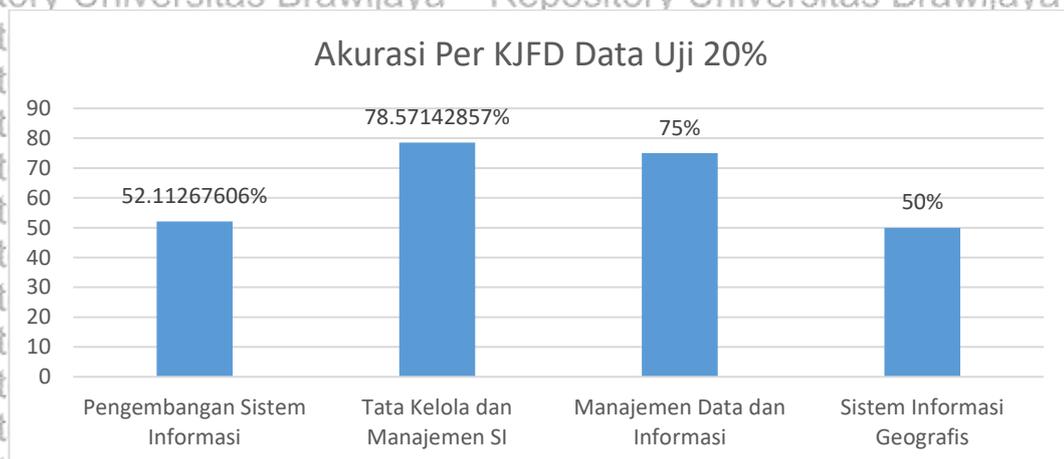


Gambar 6.4 Pembagian Jumlah Data Uji Per KJFD



Gambar 6.3 menjelaskan persentase pembagian jumlah data skripsi per bidang KJFD yang ada. Pengembangan Sistem Informasi menempati angka tertinggi dengan 78% dari jumlah data, sementara Sistem Informasi Geografis menempati angka terendah dengan 3% dari jumlah data. Grafik yang serupa juga digambarkan pada gambar 6.4 dengan perbedaan data pada Sistem Informasi Geografis dan Manajemen Data dan Informasi dikarenakan angka yang ganjil. Gambar 6.3 dan 6.4 menjelaskan bahwa pembagian data uji telah terstratifikasi sesuai dengan pembagian data seluruhnya.

Gambar 6.4 menjelaskan dominasi jumlah data pada bidang Pengembangan Sistem Informasi hal ini berkaitan dengan data pada Gambar 6.1. Keenam data dosen yang mendominasi hasil rekomendasi memiliki kesamaan, yaitu mereka membimbing skripsi dengan bidang Pengembangan Sistem Informasi. Djoko Pramono, Ismiarta Aknuranda, dan Yusi Tyroni memiliki bidang Pengembangan Sistem Informasi sebagai bidang *mayor*. Hanifah Muslimah, Herman Tolle, dan Widhy Hayuhardhika memiliki bidang Pengembangan Sistem Informasi sebagai bidang *minor*. Disini dapat dianalisa bahwa jumlah rekomendasi dipengaruhi oleh jumlah data pada bidang.



**Gambar 6.5 Akurasi Per KJFD pada Data Uji 20%**

Gambar 6.5 menjelaskan mengenai akurasi pada setiap bidang KJFD pada data uji 20% dari jumlah seluruh data. Akurasi tertinggi diraih oleh bidang Tata Kelola dan Manajemen SI. Namun, jumlah akurasi keseluruhan adalah 57,14%. Akurasi ini dipengaruhi oleh dominasi data Pengembangan Sistem Informasi yang memiliki akurasi 52,11%.

Berdasarkan data yang tertera pada gambar 6.1, 6.2, 6.3, 6.4, dan 6.5 dapat dianalisa bahwa:

1. Pada grafik 6.5 dapat dilihat bahwa akurasi tertinggi didapatkan oleh KJFD TKMSI dengan akurasi sebesar 78,57%. Namun, akurasi gabungan pada data uji 20% sebesar 57,14%. Akurasi ini lebih rendah dikarenakan KJFD PSI memiliki akurasi 52,1%. KJFD PSI berkontribusi sebesar 78% dari jumlah data, sehingga memiliki pengaruh lebih tinggi kepada akurasi sistem.



2. Pada grafik 6.1 dapat dilihat bahwa persebaran rekomendasi dosen menumpuk pada enam dosen. Enam dosen tersebut adalah Djoko, Hanifah, Herman, Ismiarta, Widhy, dan Yusi. Djoko, Ismiarta, dan Yusi memiliki kesamaan yaitu memiliki Mayor KJFD PSI. Hanifah dan Widhy memiliki kesamaan yaitu memiliki Minor KJFD PSI.
3. Persebaran data rekomendasi yang menumpuk pada enam dosen tidak dipengaruhi oleh jumlah skripsi per dosen yang dapat dilihat pada grafik 6.2
4. Persebaran data rekomendasi yang menumpuk pada enam dosen dipengaruhi oleh jumlah skripsi per KJFD. Pengaruh ini dapat dilihat bahwa keenam dosen tersebut membimbing skripsi dengan bidang KJFD PSI yang dimana pada grafik 6.3 dan 6.4 mengambil 78% dari jumlah data skripsi.

## 6.2 Pengujian Unit

Pengujian unit menguji validitas dari setiap metode yang terdapat pada sistem. Dengan menggunakan pengujian unit, hasil keluaran dari setiap sistem disesuaikan dengan hasil yang diharapkan.

### 6.2.1 Alat Pengujian

Alat yang digunakan untuk pengujian unit adalah PHPUnit dan unittest. PHPUnit digunakan untuk menguji *frontend* sistem perkomendasi dosen yang dibangun menggunakan bahasa PHP. Unittest digunakan untuk menguji *web service* sistem perkomendasi dosen yang dibangun menggunakan bahasa Python. PHPUnit dan unittest berisi fitur yang digunakan untuk memastikan hasil pemrosesan sebuah metode sesuai dengan hasil yang diinginkan oleh pengembang sistem. PHPUnit dapat diakses melalui situs <https://phpunit.de/>. Unittest dapat langsung diakses dengan meng-*import library* yang telah disediakan pada bahasa pemrograman Python.

### 6.2.2 Prosedur Pengujian

Pengujian unit dilakukan dengan menguji hasil keluaran dari setiap metode pada sistem perkomendasi dosen pembimbing. Pengujian unit pertama dilakukan pada *frontend* dari sistem perkomendasi dosen pembimbing. Metode PHPUnit yang digunakan adalah `assertContains`, `assertRedirect`, dan `assertEquals`. Metode `assertContains` memastikan bahwa tampilan berbentuk HTML yang dikembalikan oleh metode tersebut berisi potongan kode tertentu. Metode `assertRedirect` memastikan bahwa metode yang diuji memanggil metode `Redirect` yang mengarah ke sebuah metode pada kelas tertentu. Metode `assertEquals` memastikan bahwa data kembalian yang dikembalikan oleh metode tersebut sama dengan data yang telah ditentukan.

Pengujian unit kedua dilakukan pada *web service* dari sistem perkomendasi dosen pembimbing. Metode unittest yang digunakan adalah `assertEqual`. Metode

assertEqual memastikan hasil keluaran dari metode pada *web service* sesuai dengan hasil yang telah ditentukan.

### 6.2.3 Hasil Pengujian

Hasil pengujian unit pada *frontend* dijelaskan pada tabel 6.3 sebagai berikut.

**Tabel 6.3 Hasil Pengujian Unit Frontend**

No	Nama Kelas	Nama Metode	Hasil Pengujian
1	(Controller) Dashboard	index	Valid
2	(Controller) History	index	Valid
3	(Controller) History	load	Valid
4	(Controller) History	detail	Valid
5	(Controller) History	load_detail	Valid
6	(Controller) Login	index	Valid
7	(Controller) Login	login	Valid
8	(Controller) Login	logout	Valid
9	(Controller) Registration	index	Valid
10	(Controller) Registration	register	Valid
11	(Model) M_history	load	Valid
12	(Model) M_history	load_detail	Valid
13	(Model) M_login	login	Valid
14	(Model) M_registration	register	Valid

Berdasarkan hasil pengujian unit *frontend* dapat dilihat bahwa pengujian unit menghasilkan validitas sebesar 100% yang berarti semua metode berjalan sesuai dengan hasil yang diharapkan. Hasil pengujian unit pada *web service* dijelaskan pada tabel 6.4 sebagai berikut.

**Tabel 6.4 Hasil Pengujian Unit Web Service**

No	Nama Kelas	Nama Metode	Hasil Pengujian
1	Ws	Stemming	Valid
2	Ws	wp	Valid
3	Ws	Insert_db	Valid

Berdasarkan hasil pengujian unit *web service* dapat dilihat bahwa pengujian unit menghasilkan validitas sebesar 100% yang dimana semua metode pada *web service* berjalan sesuai dengan hasil yang diharapkan. Sehingga seluruh sistem mendapatkan nilai validitas sebesar 100%.



### 6.3 Pengujian Performa

Pengujian performa mencoba menjawab kemampuan dari sistem yang akan digunakan. Pengujian akan dibagi menjadi dua, yaitu pengujian *loading time* dan pengujian *stress*.

#### 6.3.1 Alat Pengujian

Alat yang digunakan untuk pengujian performa adalah Apache JMeter dan HttpWatch. Apache JMeter dapat diakses melalui <https://jmeter.apache.org/> dan digunakan untuk melakukan pengujian *stress*. Apache JMeter dapat mereplikasi *request* pada halaman *web* hingga mencapai titik kegagalan sistem. HttpWatch dapat diakses melalui <https://www.httpwatch.com/> dan digunakan untuk melakukan pengujian *loading time*. HttpWatch menghitung waktu yang dibutuhkan oleh *web browser* untuk mengambil data dari server hingga selesai memproses tampilan.

#### 6.3.2 Pengujian Loading Time

Pengujian *loading time* menghitung seberapa lama waktu yang dibutuhkan untuk sebuah website dapat dimuat dan ditampilkan. Secara umum *guideline* yang digunakan untuk proses *loading time* sebuah halaman website membutuhkan waktu kurang dari 8 detik (Chopra, 2018). Hasil pengujian *loading time* pada sistem yang telah dibangun dapat dilihat pada tabel 6.5

Tabel 6.5 Hasil Pengujian Loading Time Sistem

No	Nama Proses	Lama Proses
1	Load Halaman Login	2.705 ms (2,7 detik)
2	Load Halaman Registrasi	1.452 ms (1,5 detik)
3	Proses Registrasi dan Load Halaman Login	1.617 ms (1,6 detik)
4	Proses Login dan Load Halaman Utama	5.948 ms (5,9 detik)
5	Proses Latar Belakang menjadi Rekomendasi Dosen	50.586 ms (50,5 detik)
6	Load Halaman History	2.509 ms (2,5 detik)
7	Load Halaman Detail History	1.699 ms (1,7 detik)
8	Proses Logout dan Load Halaman Login	1.242 ms (1,2 detik)

Dapat dilihat pada tabel 6.5 bahwa Sebagian besar hasil pengujian loading time sistem telah memenuhi *guideline*. Namun, proses mengubah latar belakang menjadi rekomendasi dosen membutuhkan waktu selama 50,5 detik. Hal ini



terjadi dikarenakan proses perubahan rekomendasi dosen menjadi latar belakang berisi proses kategorisasi dan *weighted product* yang membutuhkan waktu proses yang cukup lama. Oleh karena hal tersebut, 50,5 detik menjadi waktu yang wajar.

### 6.3.3 Pengujian Stress

Pengujian *stress* dilakukan untuk mengetahui batasan dari sebuah sistem. Pada penelitian ini pengujian *stress* dibagi menjadi dua pendekatan. Pendekatan pertama adalah pendekatan jumlah *sequential traffic* yang dimana akan mengukur kekuatan sistem dalam melayani *request* dan pemrosesan yang berurutan secara sekuensial. Pendekatan kedua adalah pendekatan jumlah *concurrent traffic* yang dimana akan mengukur kekuatan sistem dalam melayani *request* dan pemrosesan yang datang dalam waktu bersamaan.

#### 6.3.3.1 Pengujian Sequential Traffic

Pengujian *sequential traffic* dilakukan dengan memberikan satu *request* yang berulang secara terus menerus hingga sistem berhenti melayani atau hingga mencapai 150 *request*. Pengujian dilakukan pada *frontend* dan *web service*. Hasil pengujian *stress* dengan pendekatan *sequential traffic* dapat dilihat pada tabel 6.6 berikut.

Tabel 6.6 Hasil Pengujian Stress Dengan Pendekatan *Sequential Traffic*

No	Nama	Jumlah Traffic Sebelum Crash	Hasil
1	Frontend Login	>150	Berhasil
2	Frontend Registrasi	>150	Berhasil
3	Web Service Pemrosesan Latar Belakang	>150	Berhasil

#### 6.3.3.2 Pengujian Concurrent Traffic

Pengujian *concurrent traffic* dilakukan dengan mengirimkan *request* dalam satu waktu dengan jumlah *thread request* yang meningkat pada setiap percobaan. *Thread request* merepresentasikan jumlah pengguna yang mengakses secara bersamaan. Pengujian dilakukan pada *frontend* dan *web service*. Hasil pengujian *stress* dengan pendekatan *concurrent traffic* dapat dilihat pada tabel 6.7 berikut.

Tabel 6.7 Hasil Pengujian Stress Dengan Pendekatan *Concurrent Traffic*

No	Nama	Jumlah Thread Sebelum Crash	Hasil
1	Frontend Login	>150	Berhasil
2	Frontend Registrasi	>150	Berhasil
3	Web Service Pemrosesan Latar Belakang	3	Gagal

Pada tabel 6.7 dapat dilihat bahwa hasil pemrosesan latar belakang mengalami crash saat digunakan 4 *thread* secara bersamaan. Hal ini disebabkan batasan *timeout* untuk respon yang diberikan oleh Heroku sebagai penyedia layanan, yaitu



selama 30 detik. Saat terdapat 4 *thread* secara bersamaan, waktu proses memakan lebih dari 30 detik.

## 6.4 Pengujian Kompatibilitas

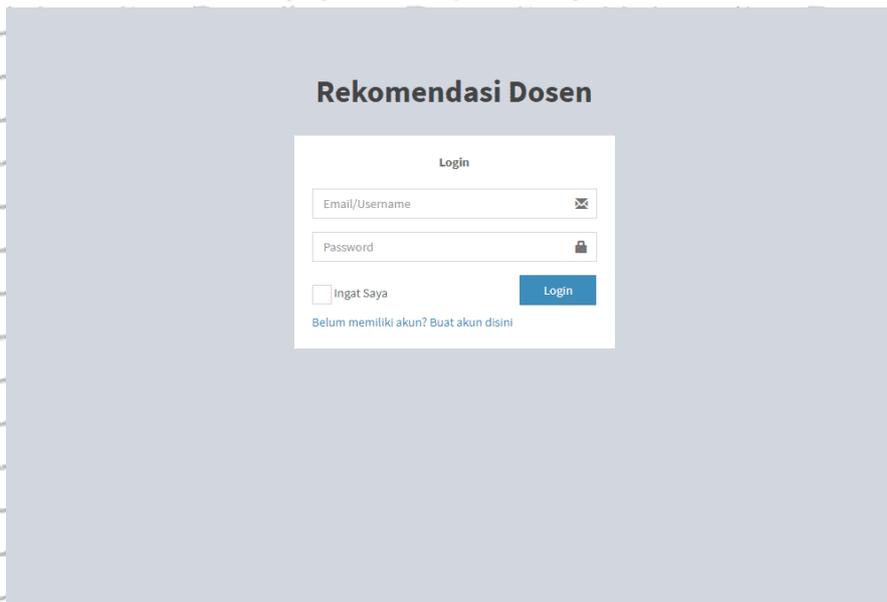
Pengujian kompatibilitas yang biasa juga disebut sebagai pengujian konfigurasi. Pengujian ini berusaha menguji kompatibilitas sistem berdasarkan sistem operasi, *web browser*, dan *platform*.

### 6.4.1 Alat Pengujian

Alat yang digunakan untuk melakukan pengujian kompatibilitas adalah LambdaTest pada [www.lambdatest.com](http://www.lambdatest.com). LambdaTest menjadi alat pengujian dikarenakan memiliki berbagai fitur untuk melakukan pengujian kompatibilitas, diantaranya: pengujian secara *real time* dan pengujian visual. Pengujian secara *realtime* memberikan pengguna akses untuk dapat mengakses halaman *website* dan menggunakannya secara langsung pada sistem operasi, *web browser*, dan *platform* yang berbeda. Pengujian visual memberikan pengguna *screenshot* tampilan *website* pada sistem operasi, *web browser*, dan *platform* yang berbeda dan responsivitas halaman *website* saat diakses menggunakan perangkat *mobile*.

### 6.4.2 Prosedur Pengujian

Pengujian dilakukan dengan menggunakan pengujian visual pada LambdaTest. Pengujian visual dilakukan pada halaman *Login* sistem. Pengujian kompatibilitas pertama dilakukan pada *web browser* Chrome versi 80 dengan sistem operasi Windows 10 menggunakan platform PC. Berdasarkan hasil pengujian, sistem rekomendasi dosen dapat digunakan pada *web browser* Chrome versi 80 dengan sistem operasi Windows 10 menggunakan *platform* PC. Tampilan hasil pengujian dapat dilihat pada gambar 6.6 berikut.



**Gambar 6.6 Hasil Pengujian Kompatibilitas pada Web Browser Chrome Sistem Operasi Windows 10 Platform PC**



Pengujian kompatibilitas kedua dilakukan pada *web browser* Chrome versi 80 dengan sistem operasi macOS Catalina menggunakan *platform* PC. Berdasarkan hasil pengujian, sistem per Rekomendasi dosen dapat digunakan pada *web browser* Chrome Chrome versi 80 dengan sistem operasi macOS Catalina menggunakan *platform* PC. Tampilan hasil pengujian dapat dilihat pada gambar 6.7 berikut.

**Rekomendasi Dosen**

Login

Email/Username

Password

Ingat Saya

Login

[Belum memiliki akun? Buat akun disini](#)

**Gambar 6.7 Hasil Pengujian Kompatibilitas pada Web Browser Chrome Sistem Operasi macOS Catalina Platform PC**

Pengujian kompatibilitas ketiga dilakukan pada *web browser* Safari versi 13 dengan sistem operasi macOS Catalina menggunakan *platform* PC. Berdasarkan hasil pengujian, sistem per Rekomendasi dosen dapat digunakan pada *web browser* Safari versi 13 dengan sistem operasi macOS Catalina menggunakan *platform* PC. Tampilan hasil pengujian dapat dilihat pada gambar 6.8 berikut.



## Rekomendasi Dosen

Login

Email/Username

Password

Ingat Saya

Login

[Belum memiliki akun? Buat akun disini](#)

**Gambar 6.8 Hasil Pengujian Kompatibilitas pada Web Browser Safari Sistem Operasi macOS Catalina Platform PC**

Pengujian kompatibility keempat dilakukan pada sistem operasi Android 9 menggunakan *platform mobile* pada perangkat Google Pixel 3. Berdasarkan hasil pengujian, sistem perkomendasi dosen dapat digunakan pada sistem operasi Android 9 menggunakan *platform mobile* pada perangkat Google Pixel 3. Tampilan hasil pengujian dapat dilihat pada gambar 6.9 berikut.



## Rekomendasi Dosen

Login

Email/Username

Password

Ingat Saya

[Belum memiliki akun? Buat akun disini](#)

**Gambar 6.9 Hasil Pengujian Kompatibilitas pada Sistem Operasi Android 9 Platform Mobile**

Pengujian kompatibility keempat dilakukan pada sistem operasi iOS 13,4 menggunakan *platform mobile* pada perangkat iPhone 8. Berdasarkan hasil pengujian, sistem perkomendasi dosen dapat digunakan pada sistem operasi iOS 13,4 menggunakan *platform mobile* pada perangkat iPhone 8. Tampilan hasil pengujian dapat dilihat pada gambar 6.10 berikut.



## Rekomendasi Dosen

Login

Ingat Saya

Belum memiliki akun? Buat akun disini

**Gambar 6.10 Hasil Pengujian Kompatibilitas pada Sistem Operasi iOS 13.4 Platform Mobile**

### 6.4.3 Hasil Pengujian

Berdasarkan pengujian kompatibilitas yang telah dilakukan, seluruh *web browser*, sistem operasi, dan jenis *platform* perangkat dapat digunakan untuk mengakses sistem rekomendasi dosen. Sehingga dapat disimpulkan hasil dari pengujian sistem rekomendasi dosen adalah valid.

### 6.5 Analisis Hasil Pengujian

Terdapat 4 jenis pengujian yang telah dilakukan. Setiap hasil pengujian yang telah dilakukan dirangkum dan kemudian dianalisis. Analisis pengujian dilakukan untuk mengetahui tingkat kesiapan sistem.

#### 6.5.1 Analisis Hasil Pengujian Akurasi

Berdasarkan pengujian akurasi yang telah dilakukan pada subbab 6.1, hasil pengujian akurasi *support vector model* yang bertujuan untuk memprediksi bidang KJFD yang sesuai dengan latar belakang tersebut memiliki hasil akurasi tertinggi pada data uji sebesar 20% dengan jumlah dosen sebanyak 21. Akurasi *support*



*vector model* tertinggi yang dapat diraih adalah 93,75%. Pengujian akurasi menampilkan bahwa prediksi bidang KJFD cukup akurat dikarenakan bidang KJFD berjumlah 5 bidang sehingga perbedaan dan keunikan antar bidang dapat ditemukan oleh sistem lebih baik dibandingkan dengan dosen pembimbing yang berjumlah di atas 20.

Hasil pengujian akurasi *weighted product* menampilkan bahwa hasil akurasi *weighted product* tertinggi pada data uji 20% dengan jumlah dosen sebanyak 21. Akurasi *weighted product* tertinggi yang dapat diraih adalah 57,14%. Pengujian akurasi menampilkan bahwa terdapat 6 dosen yang mendominasi kemunculan dalam prediksi dosen. Keenam dosen tersebut memiliki kesamaan yaitu memiliki *mayor* atau *minor* di bidang KJFD Pengembangan Sistem Informasi. Pengujian akurasi juga menampilkan bahwa pengaruh terbesar pada akurasi *weighted product* adalah bidang KJFD Pengembangan Sistem Informasi. Hal ini disebabkan dominasi data bidang KJFD Pengembangan Sistem informasi sebesar 78% dari jumlah data.

### 6.5.2 Analisis Hasil Pengujian Unit

Berdasarkan pengujian unit yang telah dilakukan pada subbab 6.2, didapatkan bahwa pengujian unit dilakukan pada unit *frontend* dan unit *web service*. Pada unit *frontend*, pengujian unit dilakukan pada 7 kelas dan 14 metode. Pengujian unit *frontend* mendapatkan hasil validasi sebesar 100%. Hasil validasi 100% menampilkan bahwa semua metode pada unit *frontend* telah berjalan sesuai dengan ekspektasi dan mengembalikan kembalian yang benar.

Pada unit *web service*, pengujian unit dilakukan pada 1 kelas dan 3 metode. Pengujian unit *web service* mendapatkan hasil validasi sebesar 100%. Hasil validasi 100% menampilkan bahwa semua metode pada unit *web service* telah berjalan sesuai dengan ekspektasi. Hal ini menjadi penting dikarenakan *web service* adalah bagian utama pemrosesan latar belakang untuk menjadi klasifikasi KJFD dan prediksi dosen pembimbing yang sesuai. Sehingga didapatkan validitas sistem sebesar 100% yang berarti sistem berhasil memproses inputan pengguna sehingga berjalan sesuai dengan ekspektasi.

### 6.5.3 Analisis Hasil Pengujian Performa

Berdasarkan pengujian performa yang telah dilakukan pada subbab 6,3 didapatkan bahwa pengujian performa dilakukan dengan dua pendekatan, yaitu pengujian *loading time* dan pengujian *stress*. Pengujian *loading time* dilakukan pada 8 proses *loading* tampilan *website*. Hasil pengujian *loading time* menampilkan bahwa 1 proses tidak mencukupi *guideline* 8 detik dengan mendapatkan waktu 50,5 detik. Proses yang lama ini dikarenakan waktu pemrosesan latar belakang menjadi rekomendasi yang lama. Namun, ke-7 proses lainnya mendapat rata-rata waktu pemrosesan sebesar 2,4 detik. Waktu ini menampilkan bahwa sebagian besar dari sistem memiliki waktu *loading time* yang cepat dan jauh di bawah *guideline*.



Untuk mendapatkan pandangan yang lebih mendalam, pengujian *stress* dilakukan dengan dua pendekatan, yaitu pengujian *sequential traffic* dan *concurrent traffic*. Kedua pengujian ini dilakukan pada 3 bagian sistem yang melakukan pemrosesan data sebagai bagian yang menggunakan sumber daya terbesar. Pengujian dilakukan hingga melebihi 150 *request*. Hasil pengujian *sequential traffic* menampilkan bahwa ketiga bagian sistem telah mampu melayani *request* berjumlah lebih dari 150. Sementara, pengujian *concurrent traffic* menampilkan bahwa terdapat 1 bagian yang mencapai titik kegagalan disaat *traffic* berjumlah 4. Titik kegagalan terjadi dikarenakan kurangnya sumber daya *memory* yang disediakan oleh penyedia *hosting*.

#### 6.5.4 Analisis Hasil Pengujian Kompatibilitas

Berdasarkan pengujian kompatibilitas yang dilakukan pada subbab 6.4, didapatkan bahwa pengujian kompatibilitas dilakukan pada sistem operasi, *web browser*, dan *platform* yang berbeda. Pengujian dilakukan pada *web browser* yang menjadi mayoritas penggunaan pada tahun 2020 hingga Mei 2020 diantaranya: Chrome 80 dan Safari 13 (StatCounter, 2020a). Pada *platform* PC, pengujian dilakukan pada sistem operasi yang menjadi mayoritas penggunaan pada tahun 2020 hingga Mei 2020, diantaranya: Windows dan OS X atau macOS (StatCounter, 2020b). Pada *platform mobile*, pengujian dilakukan pada sistem operasi yang menjadi mayoritas penggunaan pada tahun 2020 hingga Mei 2020 diantaranya: Android dan iOS (StatCounter, 2020c).

Pengujian kompatibilitas pada setiap *web browser* mendapatkan hasil kompatibilitas sebesar 100%. Pengujian kompatibilitas pada sistem operasi macOS dan Windows 10 dengan *platform* PC mendapatkan hasil kompatibilitas sebesar 100%. Sementara pengujian kompatibilitas pada sistem operasi Android dan iOS dengan *platform mobile* mendapatkan hasil kompatibilitas sebesar 100%. Hasil ini menunjukkan bahwa sistem siap untuk digunakan pada *web browser* dan sistem operasi yang mendominasi tahun 2020, serta sistem dapat digunakan dengan responsif pada *platform* PC maupun *mobile*.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil penelitian pengembangan sistem informasi perkomendasi dosen pembimbing berdasarkan latar belakang, dapat diambil kesimpulan sebagai berikut:

1. Tingkat akurasi klasifikasi Kelompok Jabatan Fungsional Dosen menggunakan algoritme *Multi-class Support Vector Machine* sebesar 93,75% pada jumlah data uji sebesar 20% dari seluruh data. Akurasi klasifikasi Kelompok Jabatan Fungsional Dosen cukup tinggi dan akurat, sehingga dapat disimpulkan bahwa klasifikasi Kelompok Jabatan Fungsional Dosen siap untuk digunakan;
2. Tingkat akurasi rekomendasi dosen pembimbing menggunakan algoritme *Weighted Product* sebesar 57,14% pada jumlah data uji sebesar 20% dari seluruh data. Akurasi rekomendasi dosen pembimbing memiliki performa akurasi yang cukup, sehingga sistem rekomendasi dosen pembimbing mampu untuk digunakan;
3. Pengujian sistem yang dilakukan adalah pengujian unit, pengujian performa, dan pengujian kompatibilitas. Pengujian unit menghasilkan nilai 100% valid. Pengujian performa menampilkan bahwa sebagian besar dari sistem memiliki performa yang responsif. Pengujian kompatibilitas menampilkan bahwa sistem dapat digunakan pada *web browser*, sistem operasi, dan *platform* yang menjadi mayoritas penggunaan pada tahun 2019. Hasil dari pengujian sistem ini menampilkan bahwa sistem mampu dan siap untuk digunakan.

### 7.2 Saran

Berdasarkan penelitian pengembangan sistem informasi perkomendasi dosen pembimbing berdasarkan latar belakang, terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya, antara lain:

1. Perlu dilakukan penelitian lebih lanjut mengenai metode lain yang dapat digunakan untuk meningkatkan akurasi rekomendasi dosen pembimbing skripsi menggunakan latar belakang
2. Perlu dilakukan penelitian lebih lanjut untuk mempercepat proses klasifikasi dan prediksi dosen pembimbing, sehingga pengguna dapat menggunakan sistem yang lebih responsif.

**DAFTAR REFERENSI**

- Amami, M., Pasi, G., Stella, F. dan Faiz, R., 2016. An LDA-Based Approach to Scientific Paper Recommendation. In: *Natural Language Processing and Information Systems*, hal.200–210.
- Arisetiawan, A.A.B., Indriati dan Ratnawati, D.E., 2019. Sistem Rekomendasi Dosen Pembimbing Berdasarkan Dokumen Judul Skripsi di Bidang Komputasi Cerdas Menggunakan Metode BM25. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(6), hal.5832–5836.
- Asogbon, M.G., Samuel, O.W., Omisore, M.O. dan Ojokoh, B.A., 2016. A Multi-class Support Vector Machine Approach for Students Academic Performance Prediction. *International Journal of Multidisciplinary and Current Research*, 4(March), hal.210–215.
- Atmadja, A.R., 2018. Recommendation System For Selecting Universities In Indonesia Using Google Maps API Services. *MATEC Web of Conferences*, 197, hal.8–12.
- Budiharjo, A.P.W. dan Muhammad, A., 2017. Comparison of Weighted Sum Model and Multi Attribute Decision Making Weighted Product Methods in Selecting the Best Elementary School in Indonesia. *International Journal of Software Engineering and Its Applications*, 11(4), hal.69–90.
- Chang, C.C. dan Lin, C.J., 2011. LIBSVM: A Library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), hal.27:1–27:27.
- Chopra, R., 2018. *Software Testing: A Self-Teaching Introduction*. Dulles: Mercury Learning and Information LLC.
- Duan, K.-B. dan Keerthi, S.S., 2005. Which Is the Best Multiclass SVM Method? An Empirical Study. In: *Lecture Notes in Computer Science*, hal.278–285.
- Feldman, R. dan Sanger, J., 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge: Cambridge University Press.
- FILKOM UB, 2018. *Buku Panduan Skripsi Fakultas Ilmu Komputer. Fakultas Ilmu Komputer Universitas Brawijaya*, Malang: Fakultas Ilmu Komputer Universitas Brawijaya.
- Goyal, P., Behera, L. dan McGinnity, T.M., 2013. A novel neighborhood based document smoothing model for information retrieval. *Information Retrieval*, 16(3), hal.391–425.
- Hasan, M.A. dan Schwartz, D., 2019. A Multi-criteria Decision Support System for Ph.D. Supervisor Selection: A Hybrid Approach. *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 6, hal.1823–1832.
- Kong, X., Jiang, H., Yang, Z., Xu, Z., Xia, F. dan Tolba, A., 2016. Exploiting publication contents and collaboration networks for collaborator recommendation. *PLoS ONE*, 11(2), hal.1–13.



Lestari, N.A., 2012. Hubungan Ekspektansi Terhadap Dosen Pembimbing Dengan Motivasi Menulis Skripsi. *Educational Psychology Journal*, 1(1), hal.1–8.

Liao, S.N., Zingaro, D., Thai, K., Alvarado, C., Griswold, W.G. dan Porter, L., 2019. A Robust Machine Learning Technique to Predict Low-performing Students. *ACM Transactions on Computing Education*, 19(3), hal.1–19.

Lv, Y. dan Zhai, C., 2011. When documents are very long, BM25 fails! In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*. New York, New York, USA: ACM Press.hal.1103.

Odu, G.O., 2019. Weighting methods for multi-criteria decision making technique. *Journal of Applied Sciences and Environmental Management*, 23(8), hal.1449.

Pressman, R.S. dan Maxim, B.R., 2015. *Software Engineering: A Practitioners' Approach*. New York: McGraw-Hill Education.

Saaty, T.L. dan Vargas, L.G., 2001. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. International Series in Operations Research & Management Science. Boston, MA: Springer US.

Shubham Bauskar, Badole, V., Jain, P. dan Chawla, M., 2019. Natural Language Processing based Hybrid Model for Detecting Fake News Using Content-Based Features and Social Features. *International Journal of Information Engineering and Electronic Business*, 11(4), hal.1–10.

Somantri, O., Wiyono, S. dan Dairoh, D., 2016. Metode K-Means untuk Optimasi Klasifikasi Tema Tugas Akhir Mahasiswa Menggunakan Support Vector Machine (SVM). *Scientific Journal of Informatics*, 3(1), hal.34–45.

Sommer, T., Bach, U., Richert, A. dan Jeschke, S., 2016. A Web-Based Recommendation System for Engineering Education E-Learning Solutions. In: *Engineering Education 4.0*. Cham: Springer International Publishing.hal.293–302.

StatCounter, 2020a. *Browser Market Share Worldwide: Jan - May 2020*. [daring] Tersedia pada: <<https://gs.statcounter.com/browser-market-share#monthly-202001-202005>> [Diakses 20 Jun 2020].

StatCounter, 2020b. *Desktop Operating System Market Share Worldwide: Jan - May 2020*. [daring] Tersedia pada: <<https://gs.statcounter.com/os-market-share/desktop/worldwide#monthly-202001-202005>> [Diakses 20 Jun 2020].

StatCounter, 2020c. *Mobile & Tablet Operating System Market Share Worldwide: Jan - May 2020*. [daring] Tersedia pada: <<https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide#monthly-202001-202005>> [Diakses 20 Jun 2020].

Triantaphyllou, E., 2000. *Multi-criteria Decision Making Methods: A Comparative Study*. Applied Optimization. Boston, MA: Springer US.

Trotman, A., Puurula, A. dan Burgess, B., 2014. Improvements to BM25 and language models examined. *ACM International Conference Proceeding Series*, 27-28-Nove, hal.58–65.



Vijayarani, S., Ilamathi, J. dan Nithya, 2015. Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, 5(1), hal.7–16.

Wang, W., Liu, J., Yang, Z., Kong, X. dan Xia, F., 2019. Sustainable Collaborator Recommendation Based on Conference Closure. *IEEE Transactions on Computational Social Systems*, 6(2), hal.311–322.

Yang, C., Liu, T., Liu, L. dan Chen, X., 2018. A Nearest Neighbor Based Personal Rank Algorithm for Collaborator Recommendation. In: *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE.hal.1–5.

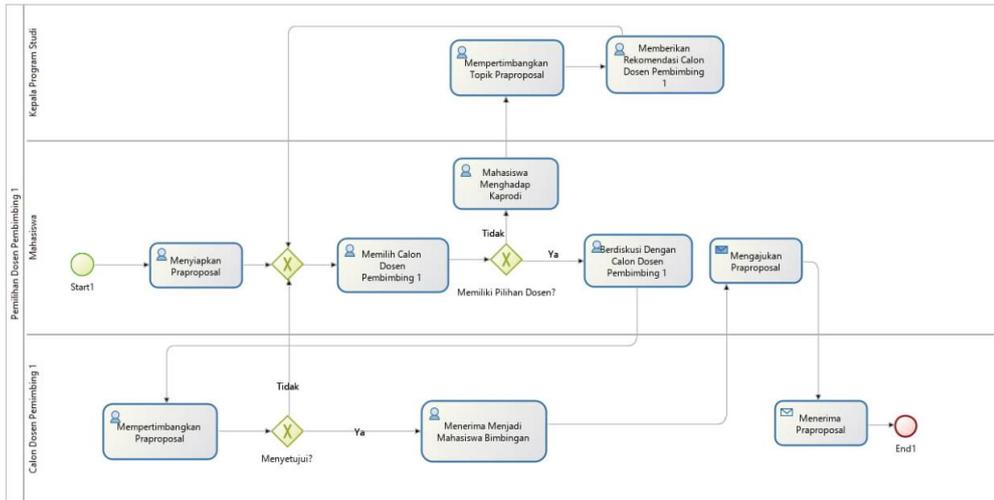
Zhang, M., Ma, J., Liu, Z., Sun, J. dan Silva, T., 2016. A research analytics framework-supported recommendation approach for supervisor selection. *British Journal of Educational Technology*, 47(2), hal.403–420.



# LAMPIRAN A VALIDASI KOORDINATOR KELOMPOK JABATAN FUNSIONAL DOSEN

## A.1 Validasi Proses Bisnis As-Is

### VALIDASI PROSES BISNIS PEMILIHAN DOSEN PEMBIMBING 1 AS-IS



Koordinator KJFD Manajemen Data dan Informasi

Nanang Yudi Setiawan, S.T., M.Kom.  
NIP. 197606192006041001



### A.3 Validasi Aktor As-Is

#### VALIDASI AKTOR PEMILIHAN DOSEN PEMBIMBING 1 AS-IS

No	Aktor
1	Mahasiswa
2	Kepala Program Studi
3	Calon Dosen Pembimbing 1

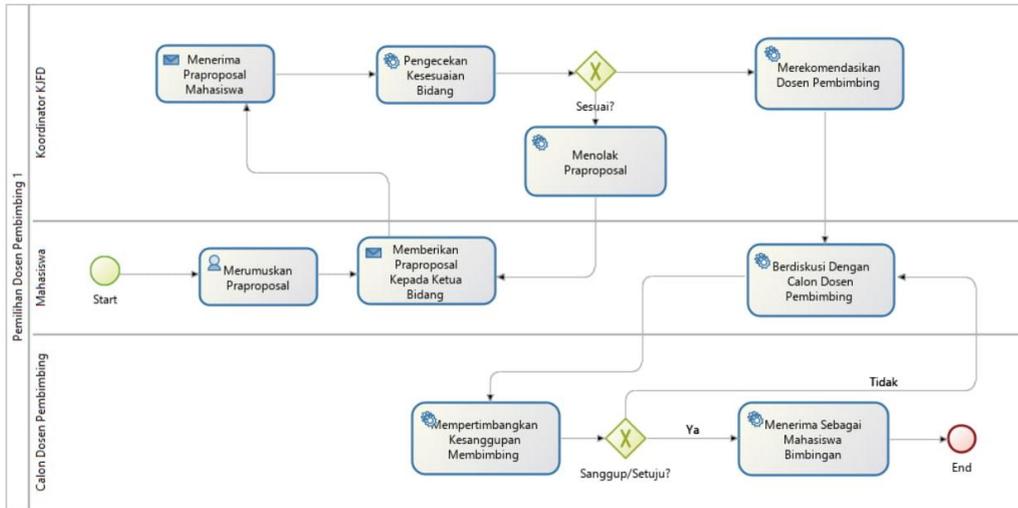
Koordinator KJFD Manajemen Data  
dan Informasi

Nanang Yudi Setiawan, S.T., M.Kom.  
NIP. 197606192006041001



### A.2 Validasi Proses Bisnis To-Be

#### VALIDASI PROSES BISNIS PEMILIHAN DOSEN PEMBIMBING 1 TO-BE



Koordinator KJFD Manajemen Data dan Informasi

Nanang Yudi Setiawan, S.T., M.Kom.  
NIP. 197606192006041001



### A.4 Validasi Aktor To-Be

#### VALIDASI AKTOR PEMILIHAN DOSEN PEMBIMBING 1 TO-BE

No	Aktor
1	Mahasiswa
2	Koordinator KJFD
3	Calon Dosen Pembimbing 1

Koordinator KJFD Manajemen Data dan Informasi

Nanang Yudi Setiawan, S.T., M.Kom.  
NIP. 197606192006041001



**A.5 Validasi Beban Variabel Weighted Product**

**VALIDASI BEBAN VARIABEL WEIGHTED PRODUCT PEMILIHAN  
DOSEN PEMBIMBING 1**

Kode	C1	C2	C3	C4	C5	C6
<b>Nama Variabel</b>	Kesesuaian KJFD	Nilai BM25	Jurusan Dosen	Sisa Kuota Bimbingan	Tingkatan Gelar	Beban Kerja
<b>Beban</b>	0,45	0,27	0,11	0,07	0,05	0,05

Koordinator KJFD Manajemen Data dan Informasi

Nanang Yudi Setiawan, S.T., M.Kom.  
NIP. 197606192006041001