

# Penentuan Pembimbing Tugas Akhir Mahasiswa Jurusan Teknik Informatika Universitas Surabaya Dengan Metode Dice Coefficient

Vincentius Riandaru Prasetyo  
Jurusan Teknik Informatika  
Universitas Surabaya  
vincent@staff.ubaya.ac.id

Budi Hartanto  
Jurusan Teknik Informatika  
Universitas Surabaya  
budi@staff.ubaya.ac.id

Aldo Adhi Mulyono  
Jurusan Teknik Informatika  
Universitas Surabaya  
s160415022@student.ubaya.ac.id

**Abstrak - Tugas Akhir (TA)** adalah sebuah karya ilmiah yang harus dibuat oleh mahasiswa jurusan Teknik Informatika, Universitas Surabaya sebagai salah satu syarat kelulusan. Dalam mengajukan topik TA, mahasiswa wajib membuat dan mengumpulkan proposal TA tersebut. Setelah itu, ketua jurusan akan menentukan calon dosen pembimbing, dosen *reviewer 1*, dan dosen *reviewer 2* sesuai dengan kompetensi untuk proposal tersebut. Dalam proses pengajuan proposal tersebut, terdapat beberapa masalah yang muncul, yakni adanya kemungkinan kesalahan penentuan pembimbing karena proses penentuannya berdasarkan pengetahuan pribadi dari ketua jurusan, kesulitan memantau perkembangan proses pengajuan proposal, dan kesulitan untuk melakukan pemerataan pembimbing TA. Untuk mengatasi masalah tersebut, dibuatlah sistem untuk melayani administrasi pengajuan topik TA beserta penentuan pembimbing. Proses penentuan pembimbing akan menggunakan metode *Dice Coefficient*. Dari hasil pengujian sistem dengan metode *K-Fold Cross Validation* dengan nilai  $k = 100$  untuk 1000 data TA yang ada, didapatkan nilai akurasi sistem sebesar 36,25%. Apabila diambil 100 data TA yang memiliki dosen pembimbing yang ideal, sistem memberikan nilai akurasi yang lebih baik yaitu sebesar 45,5%

**Kata Kunci:** *Dice Coefficient*, *K-Fold Cross Validation*, Rekomendasi, Tugas Akhir.

## I. PENDAHULUAN

Tugas Akhir (TA) adalah sebuah karya ilmiah yang harus dibuat mahasiswa Universitas Surabaya sebagai salah satu syarat kelulusan, salah satunya adalah mahasiswa Teknik Informatika. Untuk mengerjakan TA, mahasiswa Teknik Informatika, Universitas Surabaya harus memenuhi beberapa syarat, yaitu: lulus mata kuliah Metodologi Penelitian dengan nilai minimal D dan telah menyelesaikan minimum 110 SKS. Setelah itu, mahasiswa akan mengajukan topik TA dengan cara membuat proposal dan menyerahkannya ke jurusan, kemudian ketua jurusan akan menentukan 1 calon dosen pembimbing dan 2 *reviewers* sesuai dengan kompetensinya.

Dalam proses pengajuan topik TA tersebut, terdapat beberapa masalah yang terjadi. Masalah pertama adalah adanya kemungkinan terjadi kekeliruan penentuan pembimbing TA karena ketua jurusan hanya mengandalkan pengetahuan pribadi tentang kompetensi dosen. Hal ini dapat dilihat seperti contoh pada Gambar 1 yang diambil dari <http://digilib.ubaya.ac.id>.



Gambar 1. Contoh Judul TA.

Pada Gambar 1 di atas dapat dilihat bahwa judul TA yang dikerjakan masuk ke dalam peminatan *enterprise system*, sedangkan untuk dosen pembimbing ke-2 termasuk dalam peminatan *intelligence system*, di mana itu tidak sesuai bidang kompetensinya. Selain itu, beberapa dosen juga memiliki kompetensi yang tidak diketahui oleh ketua jurusan, sehingga dosen bersangkutan tidak pernah ditunjuk untuk membimbing topik TA tersebut. Masalah lain adalah kesulitan dalam melakukan pemantauan data proposal TA. Hal tersebut dikarenakan proposal TA yang disimpan secara *hard copy* di Pusat Administrasi Jurusan (PAJ), sehingga proses pencarian data menjadi sulit.

*Dice coefficient* adalah metode untuk membandingkan similaritas dari dua sampel teks yang berbeda. *Dice coefficient* merupakan versi *semimetric* dari *Jaccard coefficient*. Metode ini mempertahankan akurasi pada *dataset* yang beragam dan memberi bobot yang lebih kecil untuk *dataset* yang mengandung fitur-fitur yang tidak berkaitan [1]. Kelebihan dari metode ini adalah proses komputasi yang tidak terlalu berat namun akurat dan dapat menentukan prioritas antara *precision* atau *recall* berdasarkan kebutuhan.

Penelitian tentang penentuan atau pemilihan dosen pembimbing pernah dilakukan sebelumnya, salah satunya

yang dilakukan oleh Kasih dan Farida [2]. Pada penelitian tersebut mengimplementasikan metode klasifikasi dengan algoritma *Naive Bayes* untuk membuat sistem rekomendasi pemilihan dosen pembimbing TA pada jurusan Teknik Informatika, Universitas Nusantara PGRI Kediri. Berbeda dengan penelitian sebelumnya, Yasni, et.al. menggunakan metode *cosine similarity* dengan komposisi data judul sebesar 30%, komposisi data topik sebesar 40%, dan komposisi data abstrak sebesar 30% untuk penentuan dosen pembimbing [3]. Pada penelitian tersebut, sistem yang dibangun memiliki nilai *precision* yang didapatkan adalah 0,7 dan nilai *recall* yang didapatkan adalah 1. Salam, et.al. pada penelitiannya menggunakan algoritma Rabin-Karp dengan dua input yaitu peminatan (RPLD, IF, atau SKKD) dan topik TA yang diajukan [4]. Sistem yang dibangun akan melakukan pencocokan pola teks topik TA yang diajukan dan topik-topik TA yang ada pada basis data dan akan memberikan keluaran berupa daftar rekomendasi dosen pembimbing untuk topik TA bersangkutan yang diurutkan berdasarkan tingkat kecocokan pola tertinggi ke terendah.

Selain penelitian tentang penentuan atau pemilihan dosen pembimbing, penelitian yang mengimplementasikan metode *dice coefficient* juga pernah dilakukan sebelumnya. Nugraheni menggunakan metode tersebut untuk merancang *case based reasoning* untuk melakukan diagnosis penyakit dengan cara menginputkan gejala-gejala yang dialami seorang pasien [5]. Gejala-gejala tersebut nantinya akan dihitung similaritasnya dengan kasus yang sudah ada sebelumnya. Berbeda dengan Nugraheni, penelitian yang dilakukan oleh Thada dan Jaglan berusaha membandingkan metode *jaccard*, *dice*, dan *cosine* untuk mencari nilai *fitness* terbaik yang digunakan pada algoritma genetika [6]. Kesimpulan yang didapatkan pada penelitian tersebut adalah metode *cosine similarity* menghasilkan nilai *fitness* terbaik apabila dikombinasikan dengan algoritma genetika, diikuti oleh *dice* dan terakhir *jaccard*.

## II. METODOLOGI PENELITIAN

### A. Information Retrieval

*Information retrieval* adalah ilmu untuk mendapatkan informasi berupa teks dari kumpulan-kumpulan dokumen yang ada berdasarkan sebuah *query*. Informasi yang dikembalikan berupa daftar dokumen terkait *query* yang sudah diurutkan berdasarkan nilai kemiripan yang diukur menggunakan fungsi similaritas [7].

Pencarian dokumen dengan melakukan *query* SQL pada *database* hanya menampilkan dokumen yang mengandung *query* bersangkutan namun tidak mengukur nilai kemiripan antara dokumen dengan *query* pengguna. Dengan metode *information retrieval*, diukur nilai similaritas dokumen dengan *query*, sehingga pengguna dapat memperoleh informasi-informasi yang relevan berdasarkan *query* yang diinputkan [8].

### B. Preprocessing

*Preprocessing* merupakan suatu proses yang dilakukan untuk mendapatkan data yang bersih sehingga proses pencarian data menjadi lebih akurat [9]. *Preprocessing* terdiri dari banyak tahapan, sedangkan untuk penelitian ini hanya digunakan tahapan penghapusan *stopword*, *stemming*, dan tokenisasi.

#### B.1. Penghapusan Stopword

*Stopword* merupakan kata-kata yang terdapat pada suatu kalimat, di mana kata tersebut memiliki pengaruh yang kecil pada proses *information retrieval* [10]. Kata-kata yang termasuk ke dalam *stopword* adalah kata ganti (saya, kamu, dia) dan kata hubung (di, ke, dari, tetapi).

Tabel 1. Contoh Penghapusan *Stopword*.

Kalimat Awal	Setelah Penghapusan <i>Stopword</i>
Pembuatan Sistem Informasi Akuntansi Perusahaan 'X' dengan Metode Penyusutan Aset Service Hour	pembuatan sistem informasi akuntansi perusahaan x metode penyusutan aset service hour

Berdasarkan Tabel 1 dapat dilihat bahwa kata “pada” dan kata “dengan” akan dihapus karena termasuk dalam *stopword*. Kata-kata tersebut seringkali muncul pada suatu kalimat sehingga untuk mempercepat komputasi, kata-kata tersebut sebaiknya dihapus.

#### B.2. Stemming

*Stemming* merupakan proses untuk mencari kata dasar pada suatu kata, seperti kata “menyapu” yang mempunyai kata dasar yaitu “sapu”. Salah satu algoritma yang dapat digunakan untuk proses *stemming* dari teks berbahasa Indonesia adalah Nazief Andriani. *Stemming* memiliki 2 tahapan utama yaitu pengecekan kata dasar dan penghapusan afiks, prefiks, dan sufiks pada suatu kata [11]. Apabila hasil penghapusan *stopword* pada Tabel 1 dilakukan proses *stemming*, maka hasilnya ditunjukkan pada Tabel 2.

Tabel 2. Contoh Proses *Stemming*.

Kalimat Setelah <i>Stopword</i>	Setelah <i>Stemming</i>
pembuatan sistem informasi akuntansi perusahaan x metode penyusutan aset service hour	buat sistem informasi akuntansi usaha x metode susut aset service hour

Pada Tabel 2, dapat dilihat bahwa kata “pembuatan” berubah menjadi kata dasar “buat”. Hal tersebut berlaku juga untuk kata “perusahaan” dan “penyusutan”. Pada penelitian ini, proses *stemming* memanfaatkan *library* Sastrawi, di mana *library* tersebut mengimplementasikan algoritma Nazief Andriani.

#### B.3. Tokenisasi

Tokenisasi adalah proses pemecahan dokumen teks menjadi token-token kata. Tiga metode yang sering

digunakan untuk melakukan tokenisasi, yaitu: Unigram (satu kata), Bigram (dua kata), dan Trigram (tiga kata) [12]. Dengan menggunakan contoh kalimat hasil *stemming* pada Tabel 2, maka proses tokenisasi akan menghasilkan token kata seperti pada Tabel 3.

Tabel 3. Contoh Hasil Tokenisasi.

Unigram	Bigram	Trigram
buat	buat sistem	buat sistem informasi
sistem	sistem informasi	sistem informasi akuntansi
infomasi	informasi akuntansi	informasi akuntansi perusahaan
akuntansi	akuntansi perusahaan	akuntansi perusahaan x
perusahaan	perusahaan x	perusahaan x metode x
x	x metode	x metode susut
metode	metode susut	metode susut aset
susut	susut aset	susut aset service
aset	aset service	aset service hour
service	service hour	
hour		

C. Ekstraksi Fitur

Setelah proses *preprocessing* dilakukan terhadap data *training*, dilakukan proses ekstraksi fitur. Ekstraksi fitur untuk data dengan tipe teks dapat dilakukan dengan metode pembobotan TF-IDF. Hasil pembobotan tersebut direpresentasikan dalam bentuk *Vector Space Model* (VSM).

C.1. Pembobotan TF-IDF

Token-token yang dihasilkan dari sebuah dokumen teks akan dilakukan pembobotan dengan metode TF-IDF. Nilai TF-IDF dapat dihitung dengan Persamaan (1).

$$tf.idf = tf \times idf \tag{1}$$

Di mana *tf* adalah *term frequency*, yaitu kemunculan suatu *term/token* pada suatu dokumen tertentu. Sedangkan *idf* adalah *inverted document frequency*, yaitu nilai log basis 10 dari jumlah dokumen *N* dibagi nilai *df*. *df* adalah *document frequency* yaitu jumlah dokumen yang memiliki suatu token tertentu [13]. Oleh karena itu, *idf* dapat dirumuskan sesuai dengan persamaan (2).

$$idf = \log\left(\frac{N}{df}\right) \tag{2}$$

C.2. Vector Space Model

*Vector Space Model* (VSM) adalah representasi kumpulan dokumen sebagai vektor dalam sebuah ruang vektor. VSM merupakan teknik dasar dalam perolehan informasi yang dapat digunakan untuk penilaian relevansi dokumen terhadap kata kunci pencarian (*query*). Kumpulan kata-kata dan dokumen direpresentasikan dalam bentuk

matriks kata-dokumen [14]. Sebagai contoh, terdapat 3 dokumen berikut:

- Doc 1 : “Sistem Informasi Penjualan”
- Doc 2 : “Sistem Penjualan dan Pembelian Barang”
- Doc 3: “Sistem Penentuan Dosen Pembimbing”

Dari ketiga contoh dokumen di atas setelah dilakukan *preprocessing* dan perhitungan *term frequency*, maka akan didapatkan VSM seperti pada Tabel 4.

Tabel 4. VSM Term Frequency.

Doc ID	token								
	sis tem	infor masi	jual	beli	ba rang	ten tu	do sen	bim bing	
1	1	1	1	0	0	0	0	0	
2	1	0	1	1	1	0	0	0	
3	1	0	0	0	0	1	1	1	

Pada Tabel 4 dapat terlihat jumlah kemunculan masing-masing token yang ada pada masing-masing dokumen. Sebagai contoh, dokumen 1 memiliki token “sistem”, “informasi”, dan “jual” dengan kemunculan masing-masing token adalah satu kali.

D. Perhitungan Similaritas

Similaritas adalah suatu fungsi yang digunakan untuk mengukur nilai kemiripan antar dua objek yang diamati. Nilai similaritas akan bernilai 0 jika kedua objek yang diamati tidak memiliki kemiripan sama sekali dan bernilai mendekati 1 jika kedua objek yang diamati semakin mirip [15].

Beberapa metode yang cukup sering digunakan untuk perhitungan similaritas adalah *cosine similarity*, *dice coefficient*, dan *jaccard coefficient*. Dilihat dari sisi kinerja yang diukur dengan metode *F-measure*, performa dari *cosine similarity* adalah yang terbaik dengan nilai *F-measure* dari *dice coefficient* dan *jaccard coefficient* mendekati nilai *F-measure cosine similarity* [16]. Keunggulan dari metode *dice coefficient* adalah dapat menentukan prioritas antara *precision* atau *recall* berdasarkan kebutuhan melalui nilai  $\alpha$  pada persamaan (3).

$$sim(q, d_j) = \frac{\sum_{k=1}^n w_{kq} w_{kj}}{\alpha \sum_{k=1}^n w_{kq}^2 + (1 - \alpha) \sum_{k=1}^n w_{kj}^2} \quad (\alpha \in [0,1]) \tag{3}$$

Di mana,  $sim(q, d_j)$  adalah nilai similaritas antara *query* (*q*) dan dokumen ke-*j* (*d<sub>j</sub>*).  $w_{kq}$  adalah nilai bobot *tf.idf* token ke-*k* pada *query*.  $w_{kj}$  adalah nilai bobot *tf.idf* token ke-*k* pada dokumen ke-*j*.  $\alpha$  adalah nilai konstanta antara 0 dan 1. Apabila nilai  $\alpha$  lebih besar dari 0,5 maka nilai *precision* akan lebih tinggi. Sedangkan jika nilai  $\alpha$  lebih kecil dari 0,5 maka nilai *recall* akan lebih tinggi. Nilai  $\alpha$  yang umum digunakan adalah 0.5. Sebagai contoh, terdapat VSM hasil pembobotan TF-IDF pada Tabel 5 berikut.

Tabel 5. Contoh VSM TF-IDF.

Dokumen ID	Token				
	sistem	informasi	jual	putus	barang
1	0,5	0,35	0,3	0	0,4
2	0,5	0	0	0,6	0,4

Apabila terdapat *query* “Sistem Penjualan Barang” yang telah melalui *preprocessing* menjadi “sistem”, “jual”, “barang”, maka hasil pembobotan TF-IDF dapat dilihat pada Tabel 6.

Tabel 6. VSM TF-IDF Ditambah *Query*.

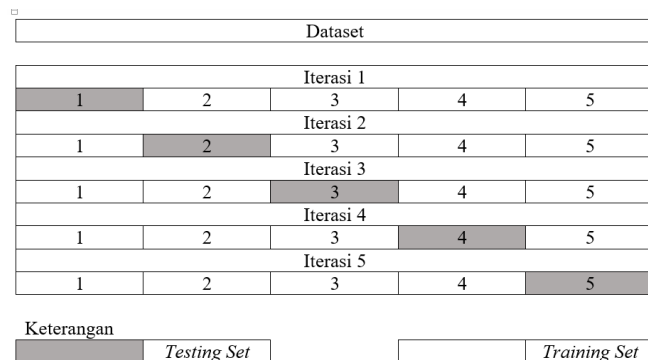
Dokumen ID	Token				
	sistem	informasi	jual	putus	barang
1	0,5	0,35	0,3	0	0,4
2	0,5	0	0	0,6	0,4
<i>Query</i>	0,5	0	0,3	0	0,4

Berdasarkan Tabel 6 di atas, maka dapat dihitung nilai similaritas antara *query* dengan dokumen 1 dan *query* dengan dokumen 2 menggunakan persamaan (3). Dari hasil perhitungan yang telah dilakukan, didapatkan nilai similaritas *query* dengan dokumen 1 adalah 1 dan nilai similaritas *query* dengan dokumen 2 adalah 0,901. Dapat dilihat bahwa nilai similaritas *query* dengan dokumen 1 lebih besar dibandingkan nilai similaritas *query* dengan dokumen 2. Oleh karena itu dokumen 1 akan ditampilkan pada urutan pertama dan dokumen 2 akan ditampilkan pada urutan kedua pada hasil pencarian.

E. *K-Fold Cross Validation*

*K-Fold Cross Validation* adalah salah satu metode validasi sebuah model *dataset*. Pada metode ini, *dataset* dibagi menjadi *K* kelompok yang memiliki jumlah data yang sama. Setelah itu, dilakukan iterasi untuk setiap kelompok tersebut sebagai data *testing*. Kelompok yang tidak berperan sebagai data *testing* akan berperan sebagai data *training*. Data *training* akan bertindak sebagai model, sedangkan data *testing* akan diklasifikasikan berdasarkan model tersebut dan dihitung akurasi atau nilai *error*. [17].

Sebagai contoh, jika dilakukan *k-fold* dengan  $k = 5$ , maka *dataset* akan dibagi menjadi 5 kelompok dengan jumlah yang sama. Pada iterasi pertama, kelompok 1 bertindak sebagai data *testing* sementara kelompok 2, 3, 4, dan 5 bertindak sebagai data *training*. Setelah itu pada iterasi kedua, kelompok 2 bertindak sebagai data *testing* sementara kelompok 1, 3, 4, dan 5 menjadi data *training*. Hal ini berlaku juga hingga iterasi kelima. Ilustrasi proses tersebut ditunjukkan pada Gambar 2.



Gambar 2. Ilustrasi *K-Fold Cross Validation*.

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Pengumpulan data dilakukan dengan cara mengunduh berkas dengan ekstensi JSON pada situs *repository.ubaya.ac.id*. Setelah itu diambil 1000 data pertama pada berkas tersebut untuk disimpan pada basis data.

Pada setiap iterasi 1000 data tersebut, diambil pula atribut *official\_url* pada data bersangkutan. Nilai dari atribut tersebut adalah sebuah *string* tautan yang menampilkan judul TA beserta dosen pembimbingnya. Sistem akan mengambil data dosen pembimbing pada halaman tersebut untuk disimpan pada basis data. Proses ini dibantu dengan *library simple\_html\_dom* yang dapat diunduh melalui tautan <https://sourceforge.net/projects/simplehtmldom/files/latest/download>.

B. Implementasi Sistem

Sistem yang dibuat dengan menggunakan *framework* PHP Laravel versi 5.6.38. *Framework* tersebut mengimplementasikan konsep MVC (*Model*, *View*, dan *Controller*). *Model* adalah representasi sebuah tabel pada *database* dalam bentuk sebuah *class*. *View* adalah tampilan yang dilihat pengguna dalam menggunakan sistem. *Controller* adalah komponen berisi fungsi-fungsi yang digunakan untuk mengakses *database* melalui *model* dan kemudian ditampilkan pada *view*.

Untuk menggunakan sistem yang dibangun, *user* harus melakukan *login* terlebih dahulu. Untuk melakukan *login*, pengguna perlu memasukkan *username* dan *password* pada kolom yang tersedia seperti pada Gambar 3. *Username* dan *password* yang digunakan adalah sama dengan *username* dan *password* yang digunakan untuk login pada *my.ubaya.ac.id*. *User* yang dapat menggunakan sistem ini dibagi menjadi 4 yaitu mahasiswa, dosen, ketua jurusan, dan admin sistem.

Gambar 3. Halaman *Login*.

Apabila *user* yang login adalah mahasiswa, maka sistem menampilkan halaman pengajuan topik TA seperti pada Gambar 4. Setelah mahasiswa memasukkan data judul, abstrak, peminatan, dan proposal TA, sistem akan melakukan validasi terhadap data tersebut. Jika terdapat kolom yang masih kosong atau data yang diinputkan sudah ada sebelumnya, maka sistem akan memberikan pesan *error*.

Gambar 4. Halaman Awal Mahasiswa.

Setelah data topik TA yang diajukan oleh mahasiswa sudah valid, maka sistem akan langsung menentukan rekomendasi calon dosen pembimbing bagi mahasiswa tersebut. Hasil rekomendasi yang dihasilkan oleh sistem tidak akan langsung diberikan kepada mahasiswa tetapi harus melalui persetujuan ketua jurusan.

Ketua jurusan yang telah login ke dalam sistem, dapat melihat daftar mahasiswa beserta judul, proposal yang telah mahasiswa unggah sebelumnya, dan hasil rekomendasi calon dosen pembimbing, seperti yang ditunjukkan pada Gambar 5. Setelah itu, ketua jurusan dapat melakukan beberapa aksi

yaitu menyetujui hasil rekomendasi calon dosen pembimbing, mengganti hasil rekomendasi, atau menolak topik TA yang diajukan mahasiswa.

ID	Judul	Mahasiswa	Proposal	Pengajuan Ke	Cadobing	Aksi
1	Pembuatan Sistem Pakar Diagnosa Penyakit Kanker Payudara	ALDO ADHI MULYONO	[Icon]	1	Susana Limanto, S.T., M.Si.	Tolak Topik, Detail, Confirm

Gambar 5. Halaman Penentuan Calon Pembimbing.

Setelah calon pembimbing ditentukan, sistem akan memberikan pesan ke mahasiswa dan dosen yang bersangkutan. Dosen dapat melihat dan membaca topik, abstrak, dan proposal yang telah dibuat sebelumnya. Dosen dapat menyetujui ataupun memberikan saran perbaikan untuk proposal TA yang diajukan oleh mahasiswa, seperti yang dapat dilihat pada Gambar 6 berikut.

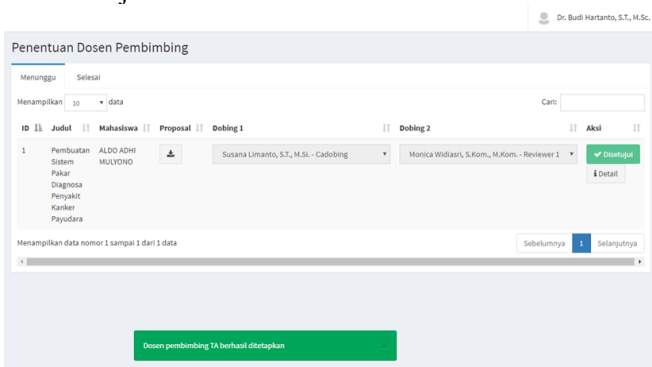
ID	Judul	Mahasiswa	Proposal	Status	Aksi
1	Pembuatan Sistem Pakar Diagnosa Penyakit Kanker Payudara	ALDO ADHI MULYONO	[Icon]	Pengajuan Baru	Beri Saran, Setujui, Detail

Gambar 6. Halaman Persetujuan Proposal TA.

Sistem akan memberikan rekomendasi 2 *reviewers* proposal TA kepada ketua jurusan apabila proposal TA sudah mendapatkan persetujuan dari calon pembimbing. Sama seperti saat menentukan calon dosen pembimbing, ketua jurusan dapat menyetujui hasil rekomendasi 2 *reviewers* atau mengganti hasil rekomendasi. Setelah itu, sistem akan mengirimkan pesan ke mahasiswa dan 2 dosen *reviewers* untuk proses *review* proposal yang diajukan. Dosen *reviewer* yang telah menyetujui proposal TA, akan diberikan pilihan apakah bersedia membimbing topik TA tersebut atau tidak.

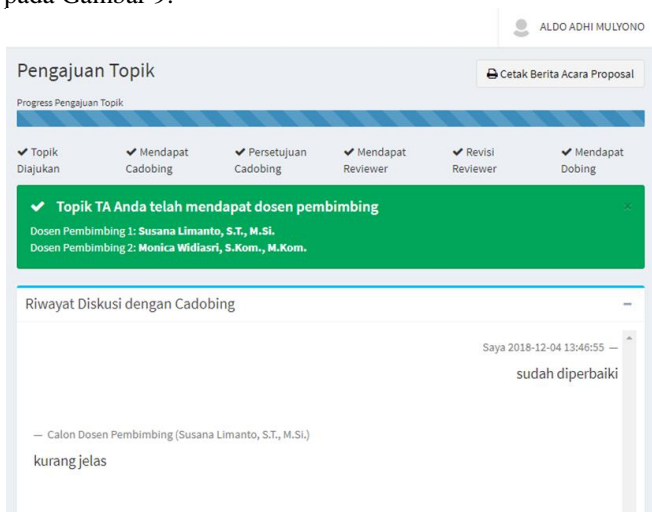
Setelah kedua *reviewers* menyatakan persetujuannya atau hasil revisi proposal berdasarkan saran *reviewer* telah disetujui cadobing, ketua jurusan bertugas untuk menentukan dosen pembimbing 1 dan 2 pada data TA bersangkutan, seperti yang ditunjukkan pada Gambar 7. Ketua jurusan dapat mengubah formasi dosen pembimbing 1 dan 2 untuk topik TA tersebut. Secara *default*, pembimbing 1 adalah calon pembimbing yang telah dipilih sebelumnya dan

pembimbing 2 adalah *reviewer* yang telah menyatakan kesediaannya untuk membimbing. Apabila kedua *reviewers* menyatakan tidak bersedia membimbing TA, maka sistem menampilkan daftar semua dosen yang tersedia untuk dipilih oleh ketua jurusan.



Gambar 7. Halaman Penentuan Dosen Pembimbing.

Sistem akan memberikan notifikasi kepada mahasiswa apabila dosen pembimbing 1 dan 2 sudah ditentukan, seperti yang terlihat pada Gambar 8. Mahasiswa dapat mencetak berita acara pengajuan proposal TA dan mengunduh berita acara pengajuan topik TA berupa berkas PDF, seperti contoh pada Gambar 9.



Gambar 8. Notifikasi Dosen Pembimbing.

**Berita Acara Proses Proposal Tugas Akhir  
Fakultas Teknik Jurusan Teknik Informatika  
Universitas Surabaya**

NRP : 160415022  
 Nama : ALDO ADHI MULYONO  
 Tanggal Pengajuan : Selasa, 04 Desember 2018. 03:10:43

Judul TA:  
**Pembuatan Sistem Pakar Diagnosa Penyakit Kanker Payudara**

---

Calon Dosen Pembimbing : **Susana Limanto, S.T., M.Si.**  
 Disetujui Tanggal : **Selasa, 04 Desember 2018. 20:21:33**

---

**Reviewer 1 : Monica Widiasri, S.Kom., M.Kom.**  
 Saran: tambahkan penjelasan pada gambaran sistem proses inti sistem pakar secara singkat.

---

Status : **Menyetujui Topik, Bersedia Membimbing**  
 Tanggal : Selasa, 04 Desember 2018. 13:44:05

---

**Reviewer 2 : Daniel Hary Prasetyo, S.Kom.**  
 Saran: -

---

Status : **Menolak Topik**  
 Tanggal : Selasa, 04 Desember 2018. 13:34:39

Kesimpulan:  
 Dosen Pembimbing 1 : **Susana Limanto, S.T., M.Si.**  
 Dosen Pembimbing 2 : **Monica Widiasri, S.Kom., M.Kom.**  
 Tanggal Penentuan : **Tuesday, 04 December 2018. 13:58:33**

Gambar 9. Berita Acara Pengajuan Proposal TA.

C. Evaluasi Sistem

*Activity diagram* Hasil implementasi sistem dievaluasi menggunakan metode *K-Fold Cross Validation* dengan nilai  $k = 10$  untuk 1000 data TA yang ada sebelumnya. Urutan proses evaluasi sistem yang dilakukan adalah sebagai berikut:

1. Inisialisasi  $x = 1$
2. Inisialisasi  $akurasi = 0$ ,  $total\_akurasi = 0$
3. Ambil data TA nomor  $x$  sampai  $(x+9)$  sebagai data testing
4. Ambil data TA selain data *testing* sebagai data *training*
5. Untuk masing-masing data testing dilakukan:
  - Perhitungan *dice coefficient* untuk mendapatkan rekomendasi dosen pembimbing 1 dan dosen pembimbing 2.
  - Jika dosen pembimbing 1 riil sama dengan dosen pembimbing 1 atau 2 hasil rekomendasi, tambah akurasi dengan 1
  - Jika dosen pembimbing 2 riil sama dengan dosen pembimbing 1 atau 2 hasil rekomendasi, tambah akurasi dengan 1
6. Set  $akurasi = akurasi / 20$ .
7. Set  $total\_akurasi = total\_akurasi + akurasi$
8. Lakukan *increment*  $x$  dengan 10 dan kembali ke langkah nomor 3

9. Set  $total\_akurasi = total\_akurasi / 100$

Pada iterasi pertama, didapatkan hasil seperti yang terlihat pada Tabel 7. Setelah 100 iterasi dijalankan, didapatkan rata-rata nilai akurasi pada setiap iterasi tersebut adalah 36,25%. Apabila semakin tinggi nilai similaritas antara data *training* dan *testing*, maka semakin tinggi pula peluang data TA tersebut memiliki rekomendasi yang akurat. Hal tersebut dibuktikan pada Tabel 8. Data TA yang mendapatkan rekomendasi yang akurat memiliki nilai similaritas 46,54% dengan data *training*, sedangkan untuk nilai 1 sebesar 44,68% dan nilai 0 sebesar 41,36%.

Tabel 7. Hasil K-Fold Cross Validation Iterasi I

ID TA	Dobing 1	Dobing 2	Rekomendasi		Nilai
			Dobing 1	Dobing 2	
1	Liliana	Lisana	Bambang	Ellysa	0
2	Bambang	Lisana	Bambang	Hendra	1
3	Monica	Ferdi	Melissa	Ongko	0
4	Endah	Melissa	Dhiani	Lisana	0
5	Daniel	Fitri	Fitri	Dhiani	1
6	Budi	Lisana	Ellysa	Budi	1
7	Lisana	Andre	Andre	Ferdi	1
8	Susana	Ellysa	Daniel	Bambang	0
9	Melissa	Ongko	Melissa	Ongko	2
10	Ellysa	Liliana	Ellysa	Liliana	2
<b>Akurasi</b>					<b>8/20 = 0,4</b>

Tabel 8. Rata-rata Nilai Similaritas Tiap Skor

Skor	Jumlah Data	Rata-rata Nilai Similaritas Tertinggi
0	318	0,413635627
1	385	0,446864893
2	297	0,465405536

Analisis juga dilakukan terhadap data TA yang mendapat rekomendasi yang kurang akurat. Setelah analisis dilakukan didapatkan beberapa penyebab kegagalan penentuan rekomendasi yang akurat adalah sebagai berikut:

1. Sistem tidak berhasil menemukan data TA yang cukup mewakili/mirip dengan data TA yang diinputkan. Hal tersebut memperbesar peluang ketidakakuratan rekomendasi sistem.
2. Sistem menemukan data TA yang cukup mirip dengan data TA yang diinputkan, tetapi data TA yang ditemukan tersebut memiliki dosen pembimbing yang berbeda dengan dosen pembimbing riil data TA yang diinputkan. Faktor ini juga dapat dipengaruhi oleh faktor nomor 3.
3. Data TA yang diinputkan atau ditemukan memiliki dosen pembimbing yang kurang ideal.
4. Ekstraksi fitur dengan metode TF-IDF kurang cocok diterapkan, hal ini dikarenakan dalam pembobotan TF-IDF fitur-fitur yang sering muncul akan memiliki nilai bobot yang semakin kecil sehingga mempengaruhi nilai similaritas.

Untuk mengatasi masalah kurang idealnya data dosen pembimbing riil, maka dilakukan *filtering* mengambil data TA dengan dosen pembimbing yang ideal. *Filtering* dilakukan dengan cara mengambil data TA yang memiliki peminatan sama dengan lab atau keahlian dua dosen pembimbing dengan *query* SQL. Setelah itu, diambil 100 data TA dari hasil *filtering* dengan proporsi 33 data dengan peminatan ESANET, 33 data dengan peminatan *Intelligent System*, dan 34 data dengan peminatan *Multimedia Computing*.

Dari 100 data tersebut, dilakukan validasi dengan metode *K-Fold Cross Validation* dengan nilai  $k = 100$  (*Leave-One-Out Cross Validation*). Sehingga setiap iterasi validasi diambil satu data TA sebagai data *testing*, sementara 99 data lainnya berperan sebagai data *training*. Pada iterasi pertama data TA pertama berperan sebagai data *testing*, sedangkan data TA ke-2 hingga ke-100 berperan sebagai data *training*. Pada iterasi kedua, data TA kedua berperan sebagai data *testing*, sedangkan data TA ke-1 dan ke-3 hingga ke-100 berperan sebagai data *training*. Iterasi tersebut dilanjutkan hingga data TA ke-100. Setelah 100 iterasi dijalankan, didapatkan akurasi sistem sebesar 45,5%.

Sama seperti pada ujicoba dengan 1000 data TA, semakin tinggi nilai similaritasnya maka akan semakin tinggi peluang data TA tersebut memiliki rekomendasi yang akurat. Hal tersebut dibuktikan pada Tabel 9, di mana perbedaan nilai similaritas tersebut cukup signifikan bila dibandingkan dengan hasil ujicoba pada 1000 data TA sebelum proses *filtering*.

Tabel 9. Rata-Rata Similaritas Setelah *Filtering*.

Skor	Jumlah Data	Rata-rata Nilai Similaritas Tertinggi
0	4	0,17148040924015576
1	49	0,379346186791362
2	47	0,4957814364820072

IV. KESIMPULAN

Dari hasil evaluasi yang dilakukan dengan metode *K-Fold Cross Validation*, akurasi sistem yang dihasilkan hanya 36,25% untuk nilai  $k = 100$ . Berdasarkan data akurasi tersebut dapat disimpulkan bahwa metode *dice coefficient* kurang cocok diterapkan dalam kasus penentuan rekomendasi pembimbing TA berdasarkan 1000 data TA terbaru. Hal tersebut terjadi karena data TA yang memiliki similaritas tertinggi dengan data TA yang diinputkan belum tentu dapat memberikan rekomendasi yang akurat. Pembimbing TA yang tepat belum tentu berada pada data TA dengan similaritas yang tertinggi, tetapi semakin kecil similaritasnya maka semakin kecil pula peluang ditemukan pembimbing yang tepat. Selain itu, terdapat faktor keterbatasan data *training* yang mirip dengan data *testing* menyebabkan rekomendasi yang kurang akurat. Kurang idealnya data dosen pembimbing riil pada data *training* juga dapat menyebabkan kegagalan penentuan rekomendasi.

Setelah dilakukan *filtering* untuk mengambil 100 data TA yang memiliki dosen pembimbing ideal saja, sistem memberikan hasil akurasi yang lebih baik yaitu sebesar 45,5%. Hal tersebut memperkuat bukti bahwa nilai akurasi dipengaruhi oleh faktor kurang idealnya data dosen pembimbing riil pada data TA. Akan tetapi menurut Ketua Jurusan Teknik Informatika, Universitas Surabaya, sistem ini dapat membantu dalam memberikan gambaran terkait dosen pembimbing yang akan dipilih, sehingga akan mempercepat waktu penentuan dosen tersebut dibandingkan harus menentukan secara manual. Selain itu, sistem ini akan membantu PAJ, ketua jurusan, calon dosen pembimbing, mahasiswa, dan reviewer dalam memantau perkembangan proposal TA yang diajukan.

Untuk penelitian lebih lanjut, dapat ditambahkan beberapa pertimbangan lain dalam menentukan rekomendasi pembimbing TA, seperti: keahlian dosen, proyek yang pernah dikerjakan dosen, rumusan masalah, tujuan penelitian, dan percobaan penggabungan dengan metode selain similaritas. Selain itu, perlu dilakukan uji coba penelitian dengan pembobotan memakai *term frequency*. Penelitian lebih lanjut juga dapat menyaring lagi data *training* dengan hanya menggunakan judul dan kata-kata kunci yang terdapat pada abstraksi. Hal tersebut juga berlaku untuk inputan sistem. Sebagai tambahan, untuk meningkatkan akurasi sistem yaitu digunakan suatu metode untuk mendeteksi struktur semantik dari suatu kalimat.

#### REFERENSI

- [1] Chahal, M. (2016). Information Retrieval using Dice Similarity Coefficient. *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 6, Issue 6, pp.72-75.
- [2] Kasih, P & Farida, I.N. (2017). Sistem Bantu Pemilihan Dosen Pembimbing Tugas Akhir Berdasarkan Kategori Pilihan dan Keahlian Dosen menggunakan Naïve Bayes. *Prosiding Seminar Nasional Teknologi, Informasi, Komunikasi dan Aplikasinya*, Vol. 4, pp.62-68.
- [3] Yasni, L., Subroto, I.M.I. & Haviana, S.F.C. (2018). Implementasi Cosine Similarity Matching dalam Penentuan Dosen Pembimbing Tugas Akhir. *Transmisi*, Vol. 20, No. 1.
- [4] Salam, A., Wicaksana, V.P. & Hastuti, K. (2015). Sistem Rekomendasi Penentuan Dosen Pembimbing Tugas Akhir Dengan Menggunakan Algoritma Rabin-Karp. *Techno. Com*, Vol. 14, No. 3, pp. 225-233.
- [5] Nugraheni, M. (2012). Rancangan Case-Based Reasoning Menggunakan Sorenson Coefficient. *Jurnal Informatika*, Vol. 6, No. 1, pp. 612-616.
- [6] Thada, V & Jaglan, V. (2013). Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm. *International Journal of Innovations in Engineering and Technology*, Vol. 4, Issue 4, pp. 202-205.
- [7] Zain, M.Y.Y. & Suswati, S.S. (2016). Information Retrieval System Pada Pencarian File Dokumen Berbasis Teks Dengan Metode Vector Space Model Dan Algoritma Ecs Stemmer. *Insand Comtech: Information Science and Computer Technology Journal*, Vol. 1, No. 1.
- [8] Mas'udia, P.E., Atmaja, M.D. & Mustafa, L.D. (2017). Information Retrieval Tugas Akhir dan Perhitungan Kemiripan Dokumen Mengacu Pada Abstrak Menggunakan Vector Space Model. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, Vol. 8, No. 1, pp. 355-362.
- [9] Prasetyo, V.R. (2018). Searching Cheapest Product On Three Different Ecommerce Using K-Means Algorithm. *Proceeding of International Seminar on Intelligent Technology and Its Application (ISITIA) 2018*. Bali, Indonesia.
- [10] Firdaus, A., Ernawati, E. & Arie, V. (2014). Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani Dan Metode Cosine Similarity. *Jurnal Teknologi Informasi*, Vol. 10, No. 1.
- [11] Prasetyo, V.R. & Winarko, E. (2016). Rating Of Indonesian Sinetron Based On Public Opinion In Twitter Using Cosine Similarity. *Proceeding of 2016 2nd International Conference on Science and Technology-Computer (ICST)*. Yogyakarta, Indonesia.
- [12] Matulatuwa, F.M., Sedyono, E. & Iriani, A. (2017). Text Mining dengan Metode Lexicon Based untuk Sentiment Analysis Pelayanan PT. Pos Indonesia Melalui Media Sosial Twitter. *Jurnal Masyarakat Informatika Indonesia*, Vol. 2, No. 3.
- [13] Karmayasa, O. (2012). Implementasi Vector Space Model Dan Beberapa Notasi Metode Term Frequency Inverse Document Frequency (Tf-Idf) Pada Sistem Temu Kembali Informasi. *Jurnal Elektronik Ilmu Komputer Udayana*, Vol. 1, No. 1.
- [14] Prasetyo, V.R. (2016). Penggolongan Program Sinetron Berdasarkan Opini Masyarakat Di Twitter Dengan Cosine Similarity. *Tesis Program Studi S2 Ilmu Komputer*. Universitas Gadjah Mada.
- [15] Baxla, M.A. (2014). Comparative Study of Similarity Measures for Item Based Top n Recommendation. *Thesis National Institute of Technology Rourkela*. Odisha.
- [16] Hamzah, A., Soesianto, F., Susanto, A. & Istiyanto, J. E. (2015). Studi Kinerja Fungsi-Fungsi Jarak dan Similaritas Dalam Clustering Dokumen Teks Berbahasa Indonesia. *Prosiding Seminar Nasional Informatika (SEMNASIF)*, Vol. 1, No. 1.
- [17] James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). K-Fold Cross-Validation. *An Introduction to Statistical Learning: With Applications in R*. New York: Springer.