



PENERAPAN METODE BM25 PADA SISTEM REKOMENDASI DOSEN PEMBIMBING DAN PENGUJI TUGAS AKHIR MAHASISWA BERBASIS WEB

Safri¹, La Ode Muh. Golok Jaya², Jumadil Nangi³

^{1,2,3} Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail: ¹safri102.031@gmail.com, ²laodemgj@gmail.com, ³jumadilnangi87@gmail.com

Abstrak

Tugas akhir merupakan salah satu mata kuliah yang harus ditempuh oleh seorang mahasiswa untuk memperoleh gelar sarjananya. Dalam menyelesaikan tugas akhir, biasanya mahasiswa telah ditentukan dosen pembimbing dan pengujinya oleh Ketua Jurusan. Dalam proses penentuan dosen pembimbing dan penguji di Jurusan Teknik Informatika Universitas Halu Oleo masih menggunakan cara konvensional dengan mengandalkan pengetahuan pribadi tentang spesifikasi keahlian dosen yang sesuai dengan topik tugas akhir mahasiswa. Penerapan metode BM25 (*Best Match 25*) sangat cocok digunakan untuk merekomendasikan dosen pembimbing dan penguji tugas akhir mahasiswa karena efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari. Pada hasil uji perbandingan penentuan dosen pembimbing dan penguji tanpa menggunakan sistem dan dengan menggunakan sistem diperoleh akurasi sebesar 58%.

Kata kunci; BM25, Rekomendasi Dosen Pembimbing dan Penguji

Abstract

This is one of the subjects that must be taken by the researcher to obtain their Bachelor degree. In completing the thesis, usually the researcher had been assigned to supervisor and examiner by the head of the department. In the process of determining supervisor and examiner in Department of informatic, Halu Oleo University still use the conventional method by relying on personal knowledge about the specification of the lecture expertise to accordance with the topic of the student thesis. The application of BM25 (Best Match 25) it is very suitable to recommend supervisor and examiner of the student assignments because it is effective and has accuracy in sorting documents based on the query being sought. In the comparative test result the determination of supervisor and examiner without using the system and by using the system obtained an accuracy about 58%.

Keywords; BM25, Recommendation of Supervisor and Examiner



1. PENDAHULUAN

Tugas akhir merupakan salah satu mata kuliah yang harus ditempuh oleh seorang mahasiswa untuk memperoleh gelar sarjananya. Dalam menyelesaikan tugas akhir, biasanya mahasiswa yang dalam proses penyusunan skripsi dibantu oleh 2 orang dosen pembimbing yang akan membimbing serta membantunya dalam penyusunan tugas akhir. Dalam menentukan calon dosen pembimbing terdapat beberapa dosen yang bisa menjadi dosen pembimbing, namun jumlah mahasiswa yang mengajukan tugas akhir lebih banyak dibanding dosen pembimbing menyebabkan proses penentuan dosen pembimbing menjadi kurang optimal. Penentuan pembimbing dan penguji tugas akhir biasanya diberikan kepada Ketua Jurusan, yang dalam proses penentuannya masih menggunakan cara konvensional yaitu dengan mengandalkan pengetahuan pribadi tentang spesifikasi keahlian dosen yang sesuai dengan topik tugas akhir mahasiswa.

Begitu pula yang terjadi di Jurusan Teknik Informatika Universitas Halu Oleo, hal ini sering menimbulkan keputusan yang kurang optimal, di mana dosen yang ditunjuk memiliki kompetensi yang kurang terhadap judul tugas akhir mahasiswa. Oleh karena itu dibutuhkan sistem yang dapat mendukung pengambilan keputusan dalam menentukan dosen pembimbing dan penguji yang sesuai dengan judul tugas akhir mahasiswa.

BM25 merupakan formula terbaik dalam kelas *Best Match*, dikarenakan formula ini efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari [1]. Media pengelola data tugas akhir yang memanfaatkan teknologi web dan penentuan dosen pembimbing dan penguji tugas akhir sangat dibutuhkan dalam hal ini.

Pada Peraturan Rektor Universitas Halu Oleo Nomor 1 Tahun 2019 Tentang Peraturan Akademik di lingkungan Universitas Halu Oleo BAB XVII Ujian Tugas Akhir Bagian Kesatu Tentang Pelaksanaan Ujian Tugas Akhir, Skripsi, Tesis dan Disertasi Pasal 83 No (1) Pembimbing/ Promotor/ Co- Promotor Untuk Tugas Akhir/ Skripsi/ Tesis/ Disertasi/ ditentukan Oleh Ketua Jurusan / Koordinator Program Studi berdasarkan keahlian dan kompetensinya.

Berdasarkan pemaparan pada paragraf sebelumnya dan melihat permasalahan yang

ada maka penulis menerapkan metode BM25 pada sistem rekomendasi dosen pembimbing dan penguji tugas akhir mahasiswa berbasis web.

2. METODE PENELITIAN

2.1 Tugas Akhir

Tugas akhir merupakan suatu karya ilmiah yang disusun mahasiswa untuk menyelesaikan studinya melalui proses berpikir ilmiah, kreatif, integratif, dan sesuai dengan disiplin ilmunya yang disusun untuk memenuhi persyaratan kebulatan studi dalam program dan jenjang pendidikan yang ada di lingkungan tempat studi. Tugas akhir disusun dengan tujuan memberi kesempatan kepada mahasiswa agar dapat memformulasikan ide, konsep, pola berpikir, dan kreativitasnya yang dikemas secara terpadu dan komprehensif, dan dapat mengkomunikasikan dalam format yang lazim digunakan dikalangan masyarakat ilmiah [2].

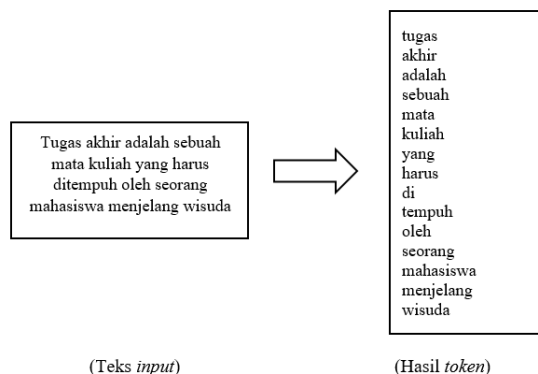
2.2 Text Mining

Text mining didefinisikan sebagai data yang berupa teks yang biasanya sumber data didapatkan dari dokumen, dengan tujuan yaitu mencari kata-kata yang dapat mewakili isi dari dokumen tersebut yang nantinya dapat dilakukan analisis hubungan antar dokumen [3]. *Text mining* dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber [4]. Pada proses *text mining* terdapat beberapa tahapan yaitu :

1) Text Preprocessing

Text Preprocessing merupakan tahapan awal dari *text mining* yang bertujuan mempersiapkan teks menjadi data yang akan mengalami pengolahan pada tahap selanjutnya. Pada *text mining*, data mentah yang berisi informasi memiliki struktur yang sembarang, sehingga diperlukan proses perubahan bentuk menjadi data yang terstruktur sesuai kebutuhan, yaitu biasanya akan menjadi nilai-nilai numerik [5], Pada tahap ini, tindakan yang dilakukan adalah menggunakan fungsi *toLowerCase*, dengan mengubah semua karakter huruf menjadi huruf kecil, dan *tokenizing* yaitu proses

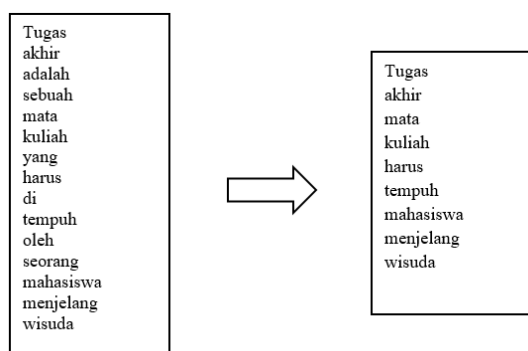
penguraian deskripsi yang semula berupa kalimat mejadi kata-kata kemudian menghilangkan delimiter-delimiter seperti tanda koma (,), tanda titik (.), spasi, dan karakter angka yang terdapat pada kata tersebut dapat dilihat pada Gambar 1.



Gambar 1. Contoh Tahapan *Tokenizing*

2) Seleksi Fitur (*Feature Selection*)

Pada tahap ini akan dilakukan seleksi dengan mengurangi jumlah kata-kata yang dianggap tidak penting dalam dokumen tersebut untuk menghasilkan proses pengklasifikasian yang lebih efektif dan akurat. Tahapan ini adalah dengan melakukan penghilangan *stopword* dan juga mengubah kata-kata kedalam bentuk dasar terhadap kata yang berimbuhan, dapat dilihat pada Gambar 2.



Gambar 2 Contoh Tahap Seleksi Fitur

3) *Stemming*

Stemming adalah proses pemetaan dan penguraian berbagai bentuk dari suatu kata menjadi kata dasarnya. Tujuan dilakukannya proses *stemming* adalah menghilangkan imbuhan-imbuhan berupa *prefix*, *suffix*, maupun konfiks yang terdapat pada setiap kata.

2.3 BM25 (*Best Match 25*)

BM25 adalah model pembobotan hasil pencarian yang dikembangkan oleh City University London berdasarkan pada model probabilistik dasar yang mengurutkan dokumen dalam urutan menurun terhadap nilai relevansi sebuah dokumen terhadap informasi yang dibutuhkan. BM25 meranking dokumen berdasarkan probabilitas dan menggunakan *term frequency* untuk meranking *similarity*.

BM25 merupakan sebuah algoritma yang digunakan oleh *search engine* untuk menentukan relevansi dokumen. Algoritma ini menggunakan *tf-idf* untuk melakukan prosesnya. Tujuan akhir dari metode BM25 adalah melakukan pembobotan dokumen berdasarkan *query* (atau dokumen yang menjadi patokan) yang dimasukkan dengan menghitung koefisien kemiripan (*Similarity Coefisien*) [6].

$$\sum_{i=1}^{|q|} idf(q_i) \frac{tf(q_i, d)(k_1 + 1)}{tf(q_i, d) + k_1(1 - b + b) \frac{|d|}{avgdl}} \quad (1)$$

Dimana:

tf (q_i,d): Frekuensi term yang muncul pada dokumen.

|d| : Jumlah kata dalam dokumen d.

avgdl : Panjang rata-rata seluruh dokumen korpus.

k₁, b : *Tunning Parameter* Nilai *k₁* = 1.2 dan *b*=0.75.

idf (q_i): Bobot dokumen dari *query* dapat dihitung sebagai berikut:

$$Idf(q_i) = \log \frac{N - df(q_i) + 0.5}{df(q_i) + 0.5} \quad (2)$$

Dimana:

N: Jumlah dokumen dalam korpus.

df: Jumlah dokumen dalam korpus yang mengandung *term (q_i)*.

Contoh kasus diketahui terdapat 3 dokumen, yaitu D1, D2 dan D3 sebagai berikut:

D1: citra identifikasi *fuzzy* diagnosis deteksi logika ekstraksi pakar genetika visualisasi virtual temu bm25

D2: kriptografi keamanan robotik jaringan *wireless gateway* arduino protokol prototipe konversi *router service cloud*

D3: geografis klasifikasi *android game* informasi keputusan *ahp saw* klastering prediksi *c45 apriori cart*.

Jika terdapat *query* (Q): “Sistem Informasi Pengelohan Data Koleksi Benda-Benda Bersejarah Pada Museum Negeri Provinsi Sulawesi Tenggara Berbasis *Android*”. Tentukan daftar dokumen yang paling relevan dengan *query* tersebut.

Penyelesaian:

Untuk menyelesaikan, pertama yang harus dikerjakan adalah melakukan tahap *preprocessing* terhadap semua (N=4) dokumen yang terlibat, yaitu D1, D2, dan D3 dan Q. Tabel 1 adalah tabel hasil *preprocessing* dokumen.

Tabel 1 Tahap *Preprocessing*

Dokumen	Term yang mewakili Dokumen
Q	sistem informasi pengelohan data koleksi benda-benda bersejarah pada museum negeri provinsi sulawesi tenggara berbasis android
D1	citra identifikasi fuzzy diagnosis deteksi logika ekstraksi pakar genetika visualisasi virtual temu bm25
D2	kriptografi keamanan robotik jaringan wireless gateway arduino protokol prototipe konversi router service cloud
D3	geografis klasifikasi android game informasi keputusan ahp saw klastering prediksi c45 apriori cart

Kemudian tentukan bobot untuk setiap *term* dari dokumen tersebut. Setelah itu lakukan perhitungan IDF kemudian hitung jumlah panjang *term* pada tiap dokumen |d|. Lalu hitung jumlah keseluruhan panjang *term* tersebut kemudian dibagi dengan jumlah dokumen (N) untuk mendapatkan hasil rata-rata panjang dokumen keseluruhan (*avgdl*).

Tabel 2 Proses Pembobotan *Term* pada Dokumen

No	Term	TF			DF	IDF	d			avgdl	BM25		
		D1	D2	D3			D1	D2	D3		D1	D2	D3
1	citra	1			1	0.778	13	33	13	13			
2	identifikasi	1			1	0.778							
3	fuzzy	1			1	0.778							
4	diagnosis	1			1	0.778							
5	deteksi	1			1	0.778							
6	logika	1			1	0.778							
7	ekstraksi	1			1	0.778							
8	pakar	1			1	0.778							
9	genetika	1			1	0.778							
10	visualisasi	1			1	0.778							
11	virtual	1			1	0.778							
12	temu	1			1	0.778							
13	bm25	1			1	0.778							
14	kriptografi		1		1	0.778							
15	keamanan		1		1	0.778							
16	robotik		1		1	0.778							
17	jaringan		1		1	0.778							
18	wireless		1		1	0.778							
19	gateway		1		1	0.778							
20	arduino		1		1	0.778							
21	protokol		1		1	0.778							
22	prototipe		1		1	0.778							
23	konversi		1		1	0.778							
24	router		1		1	0.778							
25	service		1		1	0.778							
26	cloud		1		1	0.778							
27	geografis			1	1	0.778							
28	klasifikasi			1	1	0.778							
29	android			1	1	0.3010					0	0	0.3292
30	game			1	1	0.778							
31	informasi			1	1	0.3010					0	0	0.3292
32	keputusan			1	1	0.778							
33	ahp			1	1	0.778							
34	saw			1	1	0.778							
35	klastering			1	1	0.778							
36	prediksi			1	1	0.778							
37	c45			1	1	0.778							
38	apriori			1	1	0.778							
39	cart			1	1	0.778							
											0	0	0.6584

Lalu hitung kemiripan *query* (Q) dengan dokumen yang ada. Kemiripan antar dokumen diproses dengan menggunakan BM25. Hasil perhitungan tersebut diperlihatkan pada Tabel 3.

Tabel 3 Hasil Akhir BM25

D1	D2	D3
0	0	0.6584

Maka, dokumen yang paling mirip dengan *query* (Q) adalah D3.

2.4 CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal [7]. *CodeIgniter* adalah *MVC framework* yang didesain untuk mempermudah penggunaannya. Adapun bagian dari MVC yaitu:

a. Model

Berisi fungsi-fungsi yang dapat digunakan untuk mengelola *database*, seperti CRUD.

b. View

Yang mengatur tampilan halaman *website*. bagian inilah yang bertugas untuk mempresentasikan data kepada *user*. *View* berbentuk struktur HTML yang berisikan variabel data yang dikirimkan oleh *Controller*.

c. Controller

Merupakan bagian yang berfungsi sebagai penghubung antara *Model* dan *View*. *Controller* berisi fungsi-fungsi untuk memproses suatu data. CodeIgniter memungkinkan untuk memasukkan *script* yang dibuat sendiri, atau bahkan mengembangkan *library* inti untuk sistem. MVC (*Model, View, Controller*) itu memisahkan antara logika pembuatan kode dengan pembuatan *template website* atau tampilan dari web. Jika menggunakan *Model-View-Controller* (MVC) akan menjadikan pembuatan sebuah *website* menjadi lebih terstruktur dan lebih singkat dalam mengerjakan pengkodean.

2.5 PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web tetapi juga bisa digunakan sebagai bahasa pemrograman umum [8]. PHP merupakan bahasa pemrograman web yang bersifat *server-side HTML=embedded scripting*, dimana *script* nya menyatu dengan HTML dan berada pada sisi *server*. Artinya adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan HTML biasa. PHP dikenal sebagai bahasa *scripting* yang menyatu dengan tag HTML, dieksekusi di *server* dan digunakan untuk membuat halaman web yang dinamis seperti ASP (*Active Server Pages*) dan JSP (*Java Server Pages*).

Kelebihan yang dimiliki bahasa pemrograman PHP dibandingkan bahasa pemrograman lain, yaitu:

- 1) Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- 2) Web *Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.

- 3) Dalam sisi pengembangan lebih mudah, karena banyaknya *developer* yang siap membantu dalam pengembangan.
- 4) Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
- 5) PHP adalah bahasa *open-source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.
- 6) PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini.

PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (*software* dibelakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java SunMicrosystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla, Postnuke, Xaraya dan lain-lain.

3. HASIL DAN PEMBAHASAN

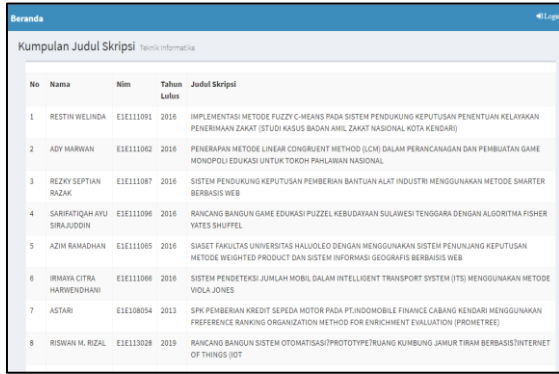
Pembuatan program dan hasil uji coba program yang telah dirancang kemudian diimplementasikan dengan bahasa pemrograman PHP. Uji coba dilakukan untuk mengetahui apakah program dapat berjalan sebagaimana mestinya dengan lingkungan uji coba yang telah ditentukan serta dilakukan sesuai dengan skenario uji coba.

3.1 Implementasi Tampilan Sistem

Implementasi rancangan antarmuka terbagi menjadi beberapa bagian utama, yaitu :

1) Menu Beranda *User*

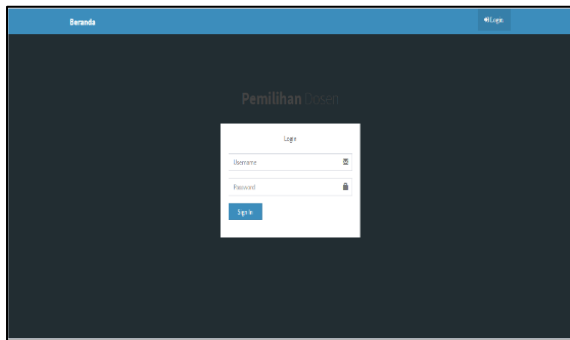
Implementasi dari desain menu utama menampilkan kumpulan skripsi yang ada di Jurusan Teknik Informatika Universitas Halu Oleo, sesuai dengan Gambar 3.



Gambar 3 Menu Beranda User

2) Halaman Login

Pada halaman ini merupakan halaman login dari sistem. Untuk masuk ke halaman utama aplikasi, pengguna aplikasi harus memasukkan *username* dan *password* terlebih dahulu. Tampilan Halaman Login dapat dilihat pada Gambar 4.



Gambar 4 Halaman Login

3) Menu Beranda Admin

Implementasi dari desain menu beranda admin merupakan halaman beranda sistem pemilihan dosen pembimbing dan pengujian sesuai pada Gambar 5.

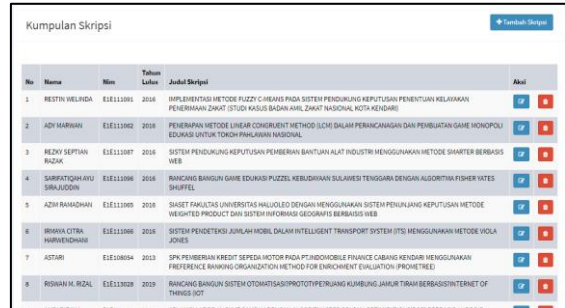


Gambar 5 Menu Beranda Admin

4) Menu Kumpulan Skripsi

Implementasi dari desain menu ini merupakan tampilan menu kumpulan skripsi, admin dapat menginputkan skripsi baru dan

mengedit atau menghapus skripsi, dapat dilihat pada Gambar 6.



Gambar 6 Menu Kumpulan Skripsi

5) Menu Pemilihan Dosen

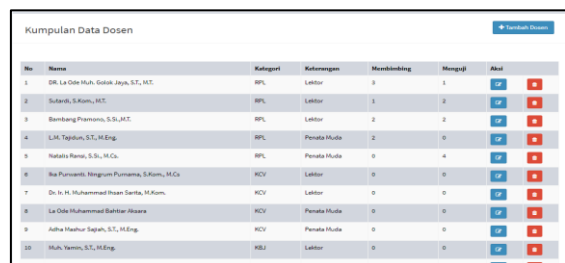
Pada menu ini admin dapat menambah rekomendasi dosen pembimbing dan pengujian, menghapus judul tugas akhir dan melihat detail dosen pembimbing dan pengujian, dapat dilihat pada Gambar 7.



Gambar 7 Menu Pemilihan Dosen

6) Menu Daftar Dosen

Pada menu ini merupakan acuan penentuan dosen pembimbing dan pengujian berdasarkan judul tugas akhir mahasiswa. Pada menu daftar dosen terdapat kategori peminatan masing-masing dosen yang terdiri dari 3 kategori yaitu RPL, KBJ, KCV, jabatan fungsional dosen, jumlah membimbing dan menguji sesuai dengan Gambar 8.

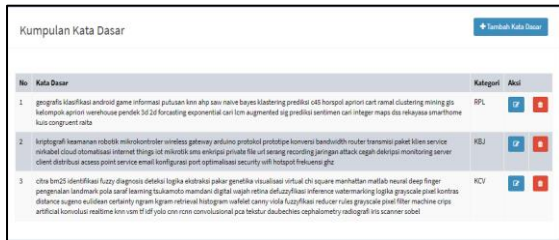


Gambar 8 Menu Daftar Dosen

7) Menu Daftar Kata Dasar

Pada menu ini akan dijadikan sebagai data latih untuk rekomendasi dosen pembimbing dan pengujian. Apabila dalam judul

tugas akhir mengandung kata-kata yang ada disalah satu kategori data latih, pada saat proses pemilihan dosen akan menampilkan kategori peminatan dosen dapat dilihat pada Gambar 9.

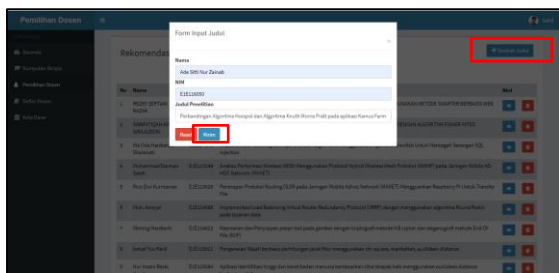


Gambar 9 Menu Daftar Kata Dasar

3.2 Pengujian Sistem

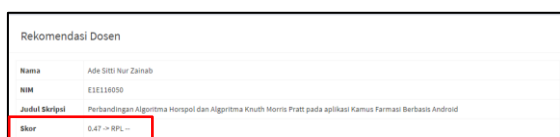
Pengujian sistem merupakan tahapan yang sangat penting dalam pembuatan aplikasi. Pengujian berfungsi untuk memastikan bahwa aplikasi yang dibangun telah berjalan dengan baik. Pengujian ini dilakukan dengan cara memasukkan nama, nim, judul tugas akhir mahasiswa dan sistem akan memberi rekomendasi dosen pembimbing dan penguji tugas akhir berdasarkan judul tugas akhir mahasiswa.

- 1) Admin mengklik menu tambah Judul yang ada dipojok kanan sistem kemudian menginputkan nama, nim dan judul tugas akhir mahasiswa selanjutnya menekan tombol proses untuk melihat skor dari BM25 dapat dilihat pada Gambar 10.



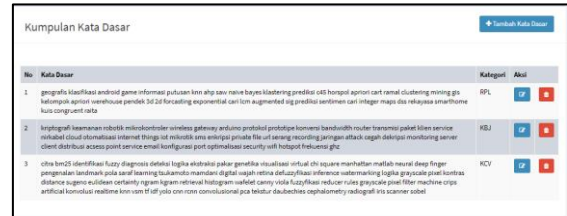
Gambar 10 Tambah Pemilihan

Setelah admin menginputkan nama, nim, judul tugas akhir mahasiswa dan menekan tombol proses, kemudian sistem menampilkan skor BM25 dari judul tugas akhir mahasiswa dapat dilihat pada Gambar 11.



Gambar 11 Skor BM25

Setelah skor BM25 diketahui, maka proses selanjutnya yaitu sistem akan mengecek judul tugas akhir mahasiswa dengan kecocokan data latih pada kumpulan kata dasar dapat dilihat pada Gambar 12.



Gambar 12 Daftar Data Latih

Karena judul inputan “Perbandingan Algoritma Horspol dan Algoritma Knuth Morris Pratt pada aplikasi Kamus Farmasi Berbasis Android” banyak mengandung data latih untuk kategori RPL maka proses selanjutnya yaitu sistem menampilkan rekomendasi dosen berdasarkan peminatan RPL dapat dilihat pada Gambar 13.



Gambar 13 Output Dosen Pembimbing dan penguji

4. KESIMPULAN

Berdasarkan penelitian dan hasil pengujian yang dilakukan pada metode BM25, maka disimpulkan sebagai berikut:

- 1) Metode BM25 dapat diterapkan dalam menentukan rekomendasi dosen pembimbing dan penguji tugas akhir mahasiswa berdasarkan judul tugas mahasiswa.
- 2) Keluaran dari sistem rekomendasi dosen pembimbing dan penguji ada 5. Urutan 1 dan 2 merupakan dosen pembimbing sedangkan urutan 3 sampai 5 merupakan dosen penguji.

5. SARAN

Saran yang dapat diberikan dalam pengembangan sistem selanjutnya adalah untuk pengembangan sistem rekomendasi dosen pembimbing dan penguji selanjutnya yaitu dapat memperluas studi kasus bukan hanya di Jurusan Teknik Informatika Universitas Halu Oleo.

DAFTAR PUSTAKA

- [1] Bintana, R.R and Agustian Surya, "Penerapan Model OKAPI BM25 Pada Sistem Temu Kembali Informasi", 273-279, 2012.
 - [2] Yudanto,"Materi PKI Pengertian Tugas Akhir", 2009. [Online]. Available : <http://staff.uny.ac.id/sites/default/files/pendidikan/Yudanto,%20S.Pd.%20Jas.%20M.Pd./MATERI%20PKI%20PEN>. [Accessed: 07-Mei-2020].
 - [3] Harlian, Milkha, "Text Mining", 2006. [Online]. Available : <http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf> . [Accessed: 04-Nov-2019].
 - [4] Hearst, Marti, "What is Text Mining",2003. [Online]. Available : <https://doi.org/10.1002/9781119282105.ch1> [Accessed: 04-Nov-2019].
 - [5] Triawati, C, "Metode Pembobotan Statistical Concept Based untuk Klustering dan Kategorisasi Dokumen Berbahasa Indonesia", Institut Teknologi Telkom Bandung, 2009.
 - [6] Sebastian, D., Rachmat, A., & Sudiarto Raharjo, W."Implementasi Algoritma Okapi BM25 dan K-Means untuk Mencari Relevansi Artikel".2011.
 - [7] Hakim, Lukmanul, *Membangun Web Berbasis PHP dengan Framework CodeIgniter*, Yogyakarta: Lokomedia, 2010.
 - [8] Kustiyaningsih, D. R. Anamisa, *Pemrograman Basis Data Berbasis Web Menggunakan PHP dan MySQL*, Yogyakarta: Graha Ilmu, 2011.
-