JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 9
No. 2
Agustus
2023

Sistem Rekomendasi Topik Skripsi Program Studi Informatika

Lukas Kurniawan[#1], Agata Filiana[#2], Gloria Virginia[#3], Bastian Surya Hartono[#4]

[#]*Duta Wacana Christian University*
Jl. Dr. Wahidin Sudirohusodo No.5-25, Kotabaru, Kec. Gondokusuman, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55224

[1]lukas.kurniawan@ti.ukdw.ac.id
[2]afiliana@ti.ukdw.ac.id
[3]virginia@ti.ukdw.ac.id
[4]bastiansurya@ti.ukdw.ac.id

***Abstrak*— Salah satu syarat kelulusan kuliah adalah lulus skripsi. Pada skripsi, mahasiswa perlu menentukan topik skripsi. Penentuan topik adalah hal yang penting pada skripsi, karena topik yang tepat akan mengurangi kendala dalam membuat skripsi [1]. Penulis melakukan analisa terhadap persentase nilai E pada nilai mata kuliah seminar dan skripsi. Hasil analisis menunjukan terdapat 21.6% dari 351 mahasiswa seminar yang mendapat nilai E dan 61.25% dari 240 mahasiswa yang belum dapat lulus skripsi. Hal tersebut menunjukan mahasiswa belum siap mengerjakan skripsi. FTI UKDW (Fakultas Teknologi Informasi Universitas Kristen Duta Wacana) ingin membuat sistem rekomendasi topik skripsi agar mahasiswa dapat memilih topik skripsi dengan tepat. Hasil penelitian, menunjukan sistem rekomendasi topik skripsi dapat dibuat menggunakan kombinasi *K-Means++*, *cosine similarity*, dan *LDA Gibbs sampling*. Sistem diimplementasikan pada *framework* Laravel.**

***Kata kunci*— *K-Means++*, *K-Means*, *LDA Gibbs Sampling*, *TF-IDF*, *Cosine similarity*, *Principal Component Analysis*, sistem rekomendasi topik skripsi, Laravel**

I. PENDAHULUAN

Skripsi adalah salah satu syarat kelulusan. Pada skripsi, mahasiswa perlu menentukan topik skripsi. Penentuan topik adalah hal yang penting pada skripsi, karena topik yang tepat akan mengurangi kendala dalam membuat skripsi [1]. Pada Prodi Informatika Universitas Kristen Duta Wacana (FTI UKDW), mahasiswa harus mengikuti tiga mata kuliah utama yaitu Riset Teknologi Informasi (RTI), Seminar, dan Skripsi. RTI adalah matakuliah yang mempersiapkan mahasiswa dalam pembuatan proposal. Sedangkan seminar adalah mata kuliah untuk menyiapkan mahasiswa mempresentasikan proposal skripsi mereka. Berdasarkan data yang diperoleh dari *data warehouse* FTI, terlihat beberapa mahasiswa yang mengulang mata kuliah Riset Teknologi Informasi (RTI), Seminar, dan Skripsi. Sebanyak 3% dari 378 mahasiswa dan 21.5% dari 351 mahasiswa mendapatkan nilai E untuk mata kuliah RTI dan Seminar secara berurutan. Sedangkan, pada mata kuliah skripsi terdapat 39.1% dari 240 mahasiswa yang belum dapat menyelesaikan skripsi pada semester pertama. Pada setiap mata kuliah di atas, rata-rata mahasiswa mengulang sebanyak 2-3 kali. Hasil analisis tersebut menunjukkan cukup banyak mahasiswa yang perlu mengulang pada 3 mata kuliah tersebut.

Data warehouse UKDW memiliki berbagai jenis data yang dapat digunakan sebagai data pelatihan model mesin, seperti data demografis, akademik dan lain-lain. Setiap jenis data memiliki informasi yang saling berbeda. Pada penelitian yang dilakukan oleh Francis, dkk [2] menunjukan data yang berkaitan dengan geografis pengguna menjadi variabel yang menurunkan akurasi model mesin yang dibuat. Tetapi, data akademik dan tingkah laku pengguna menjadi variabel yang meningkatkan akurasi model mesin. Oleh karena itu, data akademik akan direkomendasikan sebagai data latih.

Pada proses pembelajaran mesin, jika data yang dimiliki belum memiliki label, maka model mesin yang digunakan adalah *clustering*. Algortima *K-Means* adalah algoritma untuk melakukan *clustering* berdasarkan metode *non-hierarchy*, dimana data akan dipartisi lalu membentuk kelompok-kelompok karena saling memiliki kesamaan satu dengan yang lain [3]. Aubaidan, dkk [4] melakukan penelitian yang membandingkan *K-Means* dan *K-Means++* dalam klasterisasi dokumen kejahatan. Penelitian tersebut menunjukan *K-Means++* mampu mengidentifikasikan dokumen kejahatan dengan jauh lebih baik. Hal tersebut terjadi karena *K-Means++* mampu menentukan titik awal klaster dengan baik. Model *K-Means* dapat dievaluasi menggunakan tolak ukur SSE (*Sum*

Submitted 05-06-2023; Revised 04-08-2023; Accepted 24-08-2023

of Squared Errors) dan SC (*Silhouette Coefficient*) [3], [5]. Nilai tolak ukur yang baik dapat diketahui dengan menggunakan metode *elbow method* yang mencari selisih tertinggi SSE pada setiap jumlah klaster yang dilatihkan. Selisih tertinggi SSE menandakan bahwa jumlah klaster yang ditentukan adalah terbaik [6]. Hartanti [5] menggunakan *elbow method* dalam menentukan jumlah klaster yang tepat pada model *K-Means* yang dibuat.

Beberapa penelitian *K-Means* menunjukan bahwa jumlah data latih yang digunakan berjumlah banyak [7]. Pengurangan jumlah dimensi dilakukan agar kualitas model mesin menjadi baik. PCA adalah suatu metode untuk mengurangi jumlah dimensi yang menjadi bahan pelatihan bagi model pembelajaran mesin [8]. Prabhu [7] menggunakan PCA untuk mengurangi jumlah dimensi data latih. Penggunaan PCA mampu meningkatkan efisiensi dan efektivitas dari model yang dibangun.

PCA dapat digunakan untuk evaluasi model *K-Means* [9]. Afifuddin [9] melakukan evaluasi model *K-Means* dengan mengurangkan dimensi menjadi 2 dan divisualisasikan. Hasil visualisasi akan menunjukan seberapa baik model dalam mengklasterisasi data. Selain itu, Toraismaya [8] melakukan penelitian menggunakan PCA terhadap *K-Means* untuk mengurangi dimensi data latih dari 1501 menjadi 3 dimensi. Penelitian ini menghasilkan kesimpulan yaitu, penggunaan PCA dinilai cukup efektif dan efisien.

Cosine similarity dapat digunakan untuk mencari dokumen yang relevan dengan *query* / kata kunci. Tahap *preprocessing* yang dilakukan, seperti penghilangan tanda baca, *stemming*, dan penghapusan *stopword*. Pembobotan dokumen dan *query* dapat menggunaakan TD-IDF (*Term Frequency Invers Document Frequency*) [11]. Dokumen yang relevan didapat dengan membandingkan bobot *query* dengan bobot kumpulan dokumen. Pada penelitian Kurniadi, dkk [10] mengenai penggunaan *cosine similarity* dan TF-IDF untuk membuat sistem arsip. Sistem arsip yang dibangun mampu memberikan tingkat presisi 88.8% dan *recall* 76.1% dalam mencari dokumen yang relevan dengan *query* pencarian.

Dokumen-dokumen relevan yang didapat dari *cosine similarity* dapat dilakukan permodelan topik menggunakan LDA [12]. LDA memiliki dasar ide yaitu sebuah topik dapat terdiri dari kata-kata tertentu yang mana kumpulan kata tersebut dapat menyusun topik dari berbagai dokumen. LDA adalah suatu proses ekstraksi topik yang menggunakan pendekatan *unsupervised* yang memiliki karakter bahwa setiap dokumen saling berbagi kumpulan topik yang sama [13, 14, 15]. Pada penelitian yang telah dilakukan diketahui terdapat 909 judul penelitian yang dilakukan permodelan topik dengan jumlah topik sebanyak 4 [12]. Putra dan Kusumawardani [13] melakukan penelitian mengenai permodelan topik terhadap data *post* dan *tweet* pada akun media sosial Radio Suara Surabaya. Jumlah topik ditentukan menggunakan *perplexity*. Semakin rendah nilai *perplexity* menunjukan jumlah topik yang didapat adalah ideal. Selain menentukan jumlah topik, peneliti menentukan jumlah iterasi maksimal setiap permodelan topik. Penentuan jumlah iterasi maksimal melihat nilai *perplexity* mulai stabil pada iterasi ke-n. Ketika nilai mulai stabil, maka iterasi ke-n dijadikan sebagai iterasi maksimal.

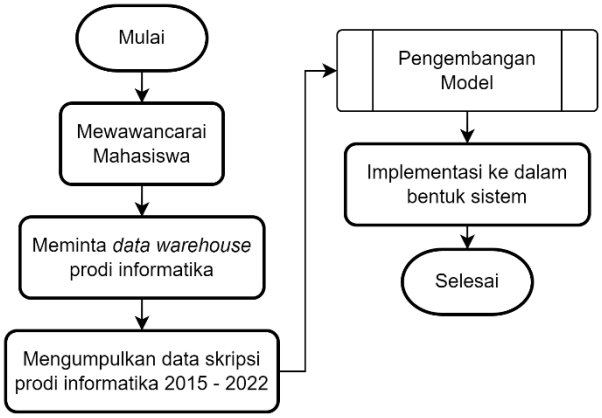
Dengan demikian sistem rekomendasi topik skripsi akan dibangun menggunakan *data warehouse* FTI UKDW yang berkaitan dengan data akademik. Sistem rekomendasi topik skripsi dibuat menggunakan kombinasi antara PCA, *K-Means++*, *cosine similarity* dan LDA (*Latent Dirichlet Allocation*) *Gibbs sampling*. Mahasiswa yang lulus skripsi dapat diklasterisasi memakai *K-Means++* berdasarkan data mata kuliah dan nilainya. Proses tersebut dapat dipakai untuk menentukan klaster dari mahasiswa pencari topik skripsi. Data skripsi yang berasal dari klaster mahasiswa pencari digunakan untuk proses selanjutnya. Kemudian, mahasiswa pencari memasukan *input* untuk mencari data skripsi yang relevan. Proses tersebut dilakukan menggunakan *cosine similarity*. Permodelan topik dilakukan menggunakan *LDA Gibbs sampling* sehingga menghasilkan topik skripsi. Sistem rekomendasi diimplementasikan ke dalam suatu *website* menggunakan *Laravel*. *Website* dapat diakses secara daring oleh mahasiswa menggunakan perangkat komputer atau laptop.

Penelitian ini memiliki harapan agar setiap mahasiswa memperoleh topik skripsi yang sesuai dengan keinginannya, sehingga mahasiswa dapat dengan mudah membuat judul skripsi yang sesuai dengan topik yang dikuasai. Selain itu, penelitian ini memiliki tujuan membuat sistem rekomendasi topik skripsi yang mampu memberikan rekomendasi tepat dengan menggunakan kombinasi *K-Means++*, LDA *Gibbs Sampling*, dan *cosine similarity*.

II. METODE PENELITIAN

A. Perancangan Penelitian

Secara umum penelitian akan dilakukan dengan tahapan pada Gambar 1.



Gambar. 1 Metodologi Penelitian

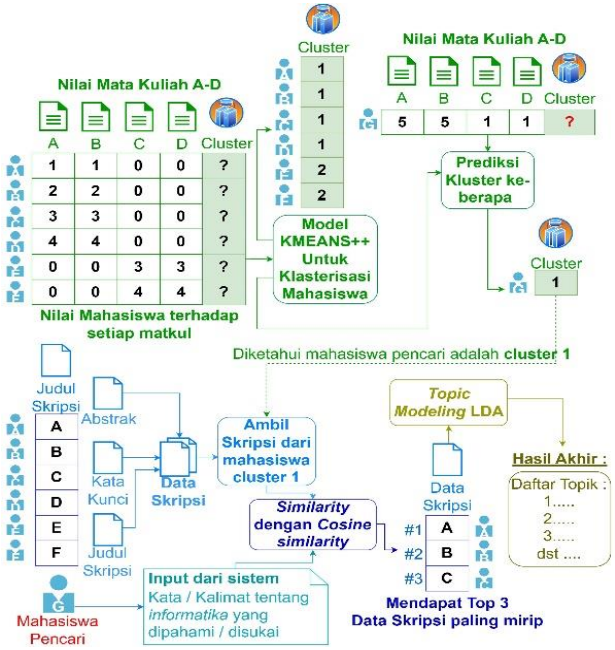
1) Mewawancari mahasiswa : Wawancara dilakukan terhadap 2 mahasiswa. Proses ini digunakan

untuk mencari berbagai faktor yang menjadi kesulitan dalam menentukan topik skripsi.

2) Mengambil data warehouse dan mengumpulkan data skripsi : Data yang dijadikan sebagai proses pembelajaran didapat dari *data warehouse* FTI UKDW. Data tersebut dilengkapi dengan data skripsi seperti judul dan abstrak. Kemudian data yang telah dilengkapi diproses ke dalam tahap pengembangan model.

3) Pengembangan Model : Data yang telah dimiliki disiapkan sebagai data latih ke model pembelajaran. Tahap pengembangan model terdiri dari beberapa proses seperti *preprocessing* data skripsi, pelatihan data ke dalam model-model pembalajaran, dan evaluasi dari setiap model pembelajaran. Tahap ini mengkombinasikan PCA, *K-Means++*, *cosine similarity*, dan LDA.

4) Implementasi dan evaluasi sistem : Implementasi sistem berupa proses mengimplementasi model yang telah dibuat kedalam suatu *website*. *Website* dibuat menggunakan *framework* Laravel. Sistem yang telah dibentuk dievaluasi mengenai lama waktu dalam memberikan rekomendasi. Selain itu, *website* dapat diakses oleh pengguna secara daring menggunakan internet.



Gambar. 2 Gambaran kerja sistem dalam memberikan rekomendasi topik skripsi.

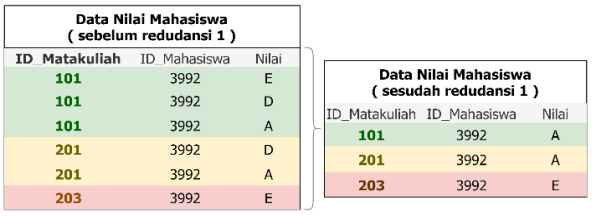
Pada gambar 2 terdapat skenario mahasiswa G akan mencari rekomendasi topik skripsi. Mahasiswa G memberi input berupa kata atau kalimat mengenai area topik yang dirasa dipahami atau disukai. Kemudian, sistem akan memproses nilai mahasiswa G, untuk mencari daftar orang pembuat skripsi dengan klaster yang sama dengan mahasiswa G. Pada kasus ini, mahasiswa G dinilai paling mirip dengan klaster 1, sehingga semua data skripsi dari mahasiswa di klaster 1 diambil. Dari data skripsi tersebut, dilakukan filterisasi dengan mencari skripsi yang relevan dengan input mahasiswa G. Skripsi yang didapat akan dilakukan permodelan topik menggunakan LDA. Hasil topik yang didapat akan direkomendasikan terhadap mahasiswa G.

B. Sumber Data

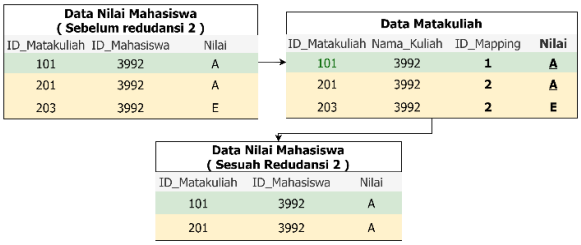
Terdapat dua data utama yang digunakan, yaitu *data warehouse* FTI UKDW dan data skripsi. Data yang berasal dari *data warehouse* berbentuk .csv dan berkaitan dengan nilai mahasiswa, seperti data mahasiswa, nilai, mata kuliah, kurikulum, dan *mapping* mata kuliah.

Data skripsi didapatkan dengan melakukan *web scraping* pada repositori menggunakan ekstensi Google Chrome Bernama “Web Scraper”. Ekstensi tersebut memungkinkan penulis untuk mengambil data yang terdapat pada repositori. Proses *scapping* dilakukan sebanyak 2 kali, pada tanggal 1 Agustus 2022 dan 25 Oktober 2022. Dilakukan proses sebanyak dua kali karena terdapat penambahan data skripsi. Kemudian, kedua data .csv digabungkan menjadi 1 data .csv yang sama.

Data yang dipakai adalah data nilai mahasiswa angkatan 2015 ke atas yang bersih. Pemilihan tahun angkatan 2015, karena kelengkapan data nilai adalah baik sejak tahun angkatan 2015. Data nilai tahun angkatan 2015 ke atas perlu dihilangkan data nilai yang mengulang. Pengulangan terjadi karena beberapa alasan seperti, mahasiswa mengambil lebih dari satu kali mata kuliah yang sama, mata kuliah yang sama berganti nama atau id, dan perubahan kurikulum. Gambar 3 dan Gambar 4 menunjukan proses penghapusan data nilai yang muncul berulang kali. Jika ada 2 nilai yang sama, maka nilai terbaik yang akan dipertahankan.



Gambar. 3 Terdapat nilai mata kuliah sama yang mengulang



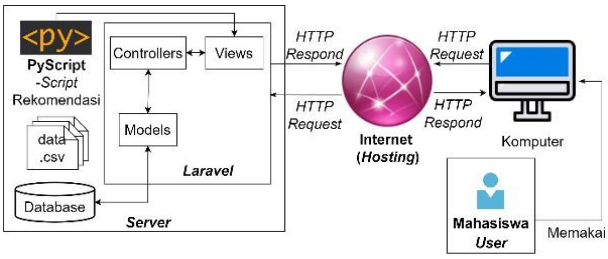
Gambar. 4 Nilai mata kuliah sama karena mengulang pada kurikulum berbeda

Data skripsi perlu dilakukan pemrosesan awal. Pemrosesan awal terdiri dari *casefolding*, *stemming*,

penghapusan *punctuation* dan *stopword* Bahasa Indonesia. Pemrosesan tersebut menggunakan *Library* dari Sastrawi. Kemudian, dilakukan pencatatan *term* dengan makna sama tetapi memiliki penulisan berbeda, seperti k-nn dan knn. *Term* yang memiliki makna, secara manual akan dicatat ke dalam *dictionary* untuk dapat dimanfaatkan sebagai pengetahuan sistem. Sebagai contoh, knn adalah algoritma untuk mengklasifikasi.

C. *Arsitektur Sistem*

Arsitektur sistem rekomendasi topik skripsi dapat dilihat pada Gambar 5.



Gambar. 5 Arsitektur sistem rekomendasi topik skripsi

Sistem rekomendasi topik skripsi merupakan *website* yang dibangun menggunakan *framework Laravel. Framework* Laravel dipilih karena menerapkan MVC (*Model–view–controller*) yang memudahkan dalam menghubungkan beberapa elemen, seperti:

- *Database*, dihubungkan melalui *model*.
- Tampilan, dihubungkan melalui *view*.
- *PyScript*, dihubungkan melalui *view*.
- Komunikasi, diatur pada *controllers*.

Website dapat diakses oleh mahasiswa melalui internet karena sistem akan dihostingkan dengan menyewa penyedia layanan *hosting*.

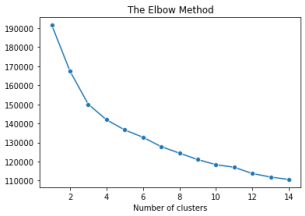
III. HASIL DAN PEMBAHASAN

Pembahasan mencakup analisis implementasi *K-Means++*, PCA, LDA *Gibbs Sampling*, *cosine similarity*, dan TF-IDF ke dalam sistem. Terdapat juga analisis terhadap sistem yang telah dibuat. Analisa akan membahas lama waktu sistem dalam memberikan rekomendasi.

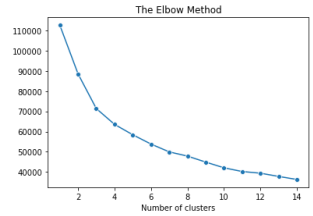
A. *Implementasi PCA dan Model K-Means++*

Model *K-Means++* dibuat menggunakan *Library Scikit-learn*. Data latih menggunakan data nilai yang sudah dipersiapkan. Proses penentuan jumlah klaster pada model *K-Means++* dapat dilakukan dengan menggunakan *elbow method* dan visualisasi klasterisasi data. Pada Gambar 6 dan Gambar 7 dilakukan percobaan melihat nilai SSE untuk jumlah klaster 1 hingga 14 dengan data yang telah di kurangi jumlah dimensi dan tanpa dikurangi jumlah dimensi.

Pada Gambar 6 dan Gambar 7 diketahui penggunaan PCA tidak memberikan dampak berbeda dengan tanpa PCA. Selain itu, selisih tertinggi SSE ditemukan pada jumlah klaster 3 dan diikuti pada jumlah klaster 4. Diantara kedua jumlah klaster tersebut dilakukan visualisasi hasil klasterisasi data. Pada Gambar 10 dan Gambar 11 dilakukan percobaan untuk melihat hasil klasterisasi data oleh *K-Means++* dengan PCA dan tanpa PCA dengan jumlah klaster 3 dan 4. Setiap warna menunjukan kelompok klaster yang saling berbeda.

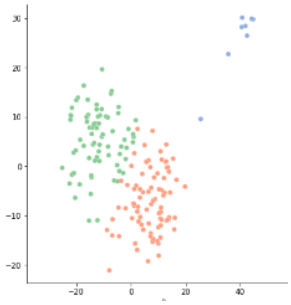


Gambar. 6 Nilai SSE pada setiap percobaan jumlah klaster

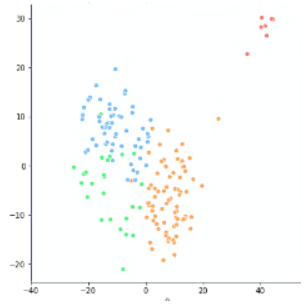


Gambar. 7 Nilai SSE pada model yang tidak memakai PCA

Hasil klasterisasi menunjukan bahwa jumlah klaster 4 masih dapat mengklasterisasi data ke dalam kelompok yang saling berbeda. Jumlah klaster yang lebih banyak memungkinkan data klaster untuk dapat terkelompokan lebih banyak. Sehingga, jumlah klaster ditentukan dengan



Gambar. 8 Jumlah Klaster 3



Gambar. 9 Jumlah Klaster 4

nilai 4. Hasil 4 klaster digunakan untuk mengklastersisasi skripsi menjadi 4, karena data nilai mahasiswa yang sudah skripsi telah diklasterisasi menggunakan *K-Means++*.

B. *Implementasi Cosine Similarity*

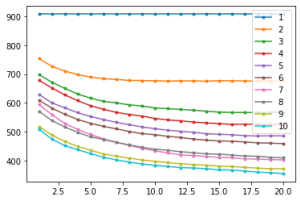
Pembobotan dokumen skripsi yang telah dilakukan digunakan sebagai pembanding terhadap bobot *query* yang diberikan pengguna. *Query* yang diberikan penggunakan dibobotkan menggunakan vektorisasi *Tfidfvektorizer* dari *Library Scikit-learn.* Sebelum dilakukan pembobotan, dilakukan pra-pemrosesan terhadap *query*, seperti *stemming*, *casefolding*, dan penghapus *stopword* menggunakan Sastrawi. Pada Tabel 1 terdapat pengujian kualitas dari hasil implementasi dengan mencari nilai presisi dan *recall* menggunakan 4 *query* yang berbeda.

TABEL I
HASIL PRESISI DAN RECALL

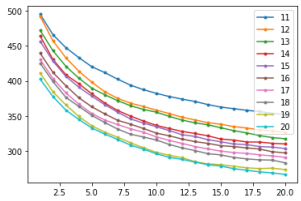
Query	**Presisi**	***Recall***
pengembangan website	74.3%	100%
pengembangan aplikasi mobile android	76.1%	100%
machine learning	28.5%	100%
pembuatan dashboard dw	100%	92.8%

C. *Implementasi LDA Gibbs Sampling*

Model LDA *Gibbs Sampling* yang digunakan dibuat mandiri menggunakan bahasa pemrograman *Python*. Model tersebut tidak dibuat menggunakan *Library* permodelan topik, seperti *Scikit-learn* dan *Gensim*. Permodelan LDA *Gibbs Sampling* memerlukan nilai awal dalam menentukan jumlah topik yang akan dihasilkan. Selain itu, LDA *Gibbs Sampling* perlu menentukan jumlah iterasi yang ideal agar model dapat menghasilkan topik skripsi yang baik, tetapi tidak memakan waktu lama. Pada Gambar 10 dan Gambar 11 terdapat percobaan dengan jumlah topik 1-20 dengan iterasi dari 1 hingga 20. Pada percobaan ini, dicari nilai *perplexity* yang mulai stabil. Sumbu horizontal menunjukan jumlah iterasi dan sumbu vertikal menunjukan nilai *perplexity* serta setiap warna menunjukan jumlah topik skripsi yang diujikan.



Gambar. 10 Nilai *perplexity* untuk topik 1-10



Gambar. 11 Nilai *perplexity* untuk topik 11-20

Visualisasi Gambar 10 dan Gambar 11 menunjukkan bahwa nilai *perplexity* mulai stabil pada iterasi 10 hingga 15. Kemudian diputuskan bahwa nilai 10 sebagai iterasi maksimal agar waktu komputasi tidak lama.

Penelitian ini melakukan percobaan untuk menentukan jumlah topik dari semua dokumen skripsi yang ada. Untuk menyingkat waktu percobaan, penelitian melakukan pengecekan nilai *perplexity* pada 6 jumlah topik skripsi, yaitu 10,15,20,25,30, dan 35.

TABEL II
HASIL PERPLEXITY TOPIK 10,15,20,25,30, DAN 35

	10	15	20	25	30	35
Perplexity	396.7	323.6	287	281	248	244

Jumlah topik skripsi 35 menjadi nilai *perplexity* terendah dibandingkan yang lain. Hal tersebut sehingga dilakukan percobaan mencari jumlah topik skripsi yang lebih detail dari 31 hingga 35.

TABEL III
HASIL PERPLEXITY TOPIK 31,32,33,34, DAN 35

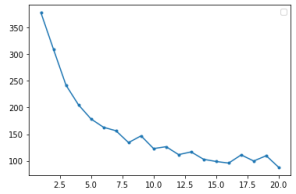
	31	32	33	34	35
Perplexity	246.27	244.3	242.9	243.1	238.78

Berdasarkan nilai *perplexity* terendah pada Tabel 3, diketahui permodelan topik menghasilkan jumlah topik terbaik dengan jumlah topik 35.

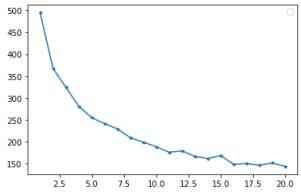
Dilakukan percobaan untuk melihat jumlah topik skripsi terbanyak yang mungkin dihasilkan pada kumpulan data skripsi pada setiap klaster yang dihasilkan pada *K-Means++*. Penulis melakukan percobaan untuk melihat nilai *perplexity* pada topik dengan jumlah 1-20.

Hasil percobaan pada Gambar 12 hingga Gambar 15 menunjukkan nilai *perplexity* cenderung stabil pada jumlah topik mendekati 20. Hal ini dapat menunjukkan jumlah topik terbanyak pada setiap klaster adalah 20. Sehingga ditentukan bahwa jumlah maksimal topik yang dapat dibentuk adalah 20 dan jumlah iterasi maksimal adalah 10.

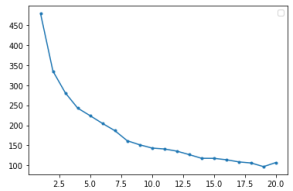
Penelitian ini tidak melakukan percobaan terhadap nilai *alpha* dan *beta*. Tetapi besarnya nilai *alpha* dan *beta* ditentukan berbanding terbalik dengan jumlah topik yang dibuat [14].



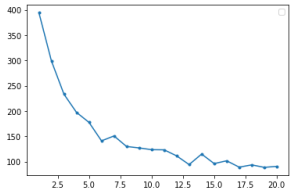
Gambar. 12 Nilai *perplexity* data skripsi pada klaster 1



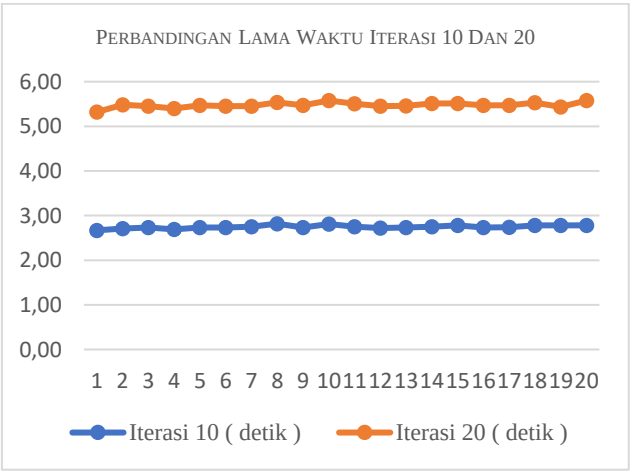
Gambar. 13 Nilai *perplexity* data skripsi pada klaster 2



Gambar. 14 Nilai *perplexity* data skripsi pada klaster 3



Gambar. 15 Nilai *perplexity* data skripsi pada klaster 4



Gambar 16 Perbandingan lama waktu iterasi 10 dan 20

Lama waktu permodelan topik memerluhkan waktu. Sehingga, dilakukan percobaan untuk mencari waktu permodelan topik dapat dilakukan pada iterasi 10 dan 20 untuk jumlah topik 1 hingga 20 yang dapat dilihat pada Gambar 16. Gambar 16 menunjukan bahwa 20 iterasi membutuhkan waktu kurang lebih 2 kali lebih lama dari 10 iterasi. Pada Gambar 10 terlihat bahwa 10 iterasi sudah memiliki nilai *perplexity* yang mulai stabil, maka dari itu ditentukan 10 sebagai iterasi maksimal. Meskipun, pada gambar 11 terlihat 10 iterasi belum cukup stabil, tetapi penentuan 10 iterasi menjadi iterasi maksimal karena lama

waktu yang dibutuhkan untuk menjalankan 20 iterasi adalah hampir dua kali lebih lama dari 10 iterasi dan lama waktu untuk 20 iterasi berkisar di 5 hingga 6 detik yang dapat menyebabkan *user* menunggu terlalu lama.

Selain itu, dilakukan percobaan terhadap lama permodelan topik dari 1 hingga 20 selesai dilakukan terhadap data skripsi pada setiap klaster. Percobaan dilakukan dengan nilai iterasi maksimal sebanyak 10. Data percobaan dapat dilihat pada Tabel 4.

TABEL IV
LAMA PERMODELAN TOPIK 1 HINGGA 20 PADA DATA SKRIPSI SETIAP KLASTER

Data skripsi klaster ke-n	Lama waktu (detik)	Jumlah dokumen
1	9.867	19
2	6.685	34
3	9.961	27
4	7.1344	21

Pada data skripsi klaster ke-1, diketahui jumlah dokumen yang ada berjumlah 19. Waktu permodelan topik yang dibutuhkan pada data tersebut adalah 9.8 detik. Sedangkan pada data skripsi klaster ke-2 dibutuhkan waktu 6.68 detik untuk memodelkan topik dari 34 dokumen. Berdasarkan hal tersebut, diputuskan bahwa jumlah dokumen terbanyak yang dapat dilakukan permodelan topik adalah 29, agar lama proses permodelan topik tidak terlalu lama. Tetapi, topik skripsi yang dihasilkan tetap dapat bersumber dari banyak dokumen.

D. Implementasi Pencatatan Kata

Pada pemrosesan data teks dilakukan pencatatan kata / *term* yang memiliki penulisan yang berbeda tapi maknanya sama. Selain itu, dilakukan pencatatan keterangan terhadap *term* yang memiliki definisi tertentu sebagai pengetahuan bagi sistem yang dibuat. Pada Tabel 5 dapat dilihat contoh hasil pecatatan tersebut.

TABEL V
HASIL PEMROSESAN KATA YANG BERBEDA

Kata	Tulisan lain	Keterangan
knn	k-nn, k nn, k-nearest	algoritma
webiste	website	*typo*
…	…	…

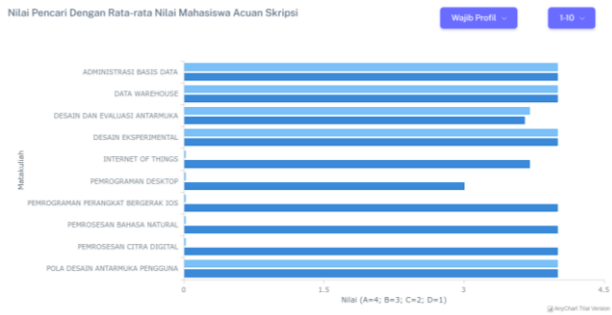
Proses tersebut dapat digunakan untuk menyamakan *query* yang diberikan mahasiswa agar tetap memiliki makna sama meskipun *query* yang diberikan berbeda. Selain itu, pencatatan *term* dimanfaatkan untuk menyamakan kata yang berbeda tapi maknanya sama pada semua dokumen skripsi yang dijadikan pelatihan dan membenarkan kesalahan penulisan yang terdapat pada dokumen skripsi.

E. Hasil dan Fitur Filterisasi Sistem

Sistem dibuat ke dalam bentuk *website* menggunakan *framework Laravel.* Sistem memiliki beberapa fitur yang berguna bagi mahasiswa pencari topik skripsi. Terdapat fitur visualisasi hasil rekomendasi ke dalam bentuk *wordcloud* yang dapat dilihat pada Gambar 17. Fitur tersebut berguna untuk memahami kata yang memiliki probabilitas tertinggi pada suatu topik skripsi.

Gambar. 17 Tampilan visualisasi hasil rekomendasi pada sistem

Terdapat fitur untuk melihat perbandingan nilai mahasiswa pencari dengan rata-rata mahasiswa yang skripsinya dijadikan sumber acuan skripsi. Fitur tersebut diberikan agar mahasiswa pencari mengetahui perbandingan nilainya dengan mahasiswa yang membuat skripsi yang menjadi acuan permodelan topik skripsi. Fitur ini dapat dilihat pada Gambar 18. Warna biru muda menunjukan nilai mahasiswa pencari. Warna biru tua menunjukan nilai rata-rata mahasiswa yang dijadikan acuan.



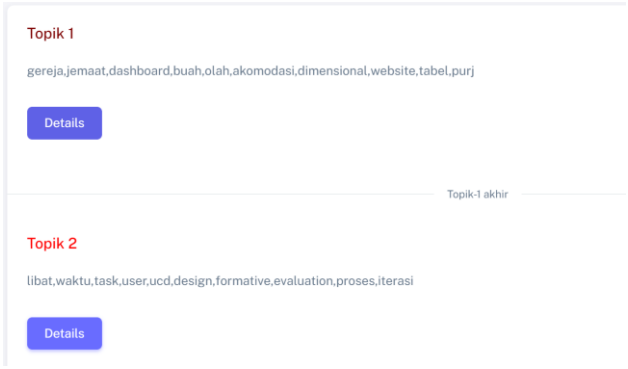
Gambar. 18 Perbandingan nilai mahasiswa pencari dengan rata-rata mahasiswa yang dijadikan acuan permodelan skripsi

Gambar 19 menunjukan terdapat fitur lain untuk melihat sumber skripsi yang digunakan dalam membuat permodelan topik. Fitur tersebut berguna agar mahasiswa pencari dapat mengetahui dan membaca sumber skripsi yang dijadikan permodelan topik.

Gambar. 19 Sunber skripsi untuk permodelan topik

Mahasiswa dapat melihat detail setiap topik yang dihasilkan. Gambar 20 dan Gambar 21 menunjukan setiap topik dapat memiliki informasi lebih detail, seperti *wordcloud* untuk topik itu sendiri dan keterangan setiap

kata serta masih dapat dicari keterangan yang lebih detail pada Google dan sistem itu sendiri.



Gambar. 20 Daftar topik yang dihasilkan

Gambar. 21 Informasi lebih detail terhadap suatu topik

Salah satu fitur yang berguna adalah teknik filterisasi sumber data skripsi. Secara umum terdapat 3 cara filterisasi. Cara pertama, sistem melakukan permodelan topik dari data skripsi klaster mahasiswa pencari. Cara kedua, sistem melakukan permodelan topik pada data skripsi selain klaster yang didapat dari mahasiswa pencari topik skripsi. Cara terakhir, sistem melakukan permodelan topik dari seluruh data skripsi yang ada.

F. Pengujian dan Analisa Sistem

Pengujian yang dilakukan berfokus terhadap lama sistem dalam memberikan rekomendasi topik skripsi. Proses perhitungan lama waktu sistem memberikan rekomendasi topik skripsi dihitung menggunakan *stopwatch*. Pengujian akan dilakukan dengan skenario halaman sedang diakses untuk pertama kali dan halaman sudah pernah diakses, dengan masing-masing menggunakan teknik filterisasi yang berbeda. Hasil pengujian dapat dilihat pada Tabel 6.

Pada Tabel 6 diketahui proses memberikan rekomendasi pada *load* halaman pertama akan membutuhkan waktu yang lama [15]. Halaman paling lama harus tampil kurang dari 15 detik. Jika halaman baru dapat diakses lebih dari 15 detik, maka pengguna tidak dapat menerima hal tersebut. Waktu yang lama dalam mendapatkan rekomendasi disebabkan karena penggunaan *Pyscript* untuk mengeksekusi *script Python* pada sistem. Pada saat *Pyscript* digunakan pertama kali, maka akan diunduh beberapa *file* pendukung agar *Pyscript* dapat bekerja. Proses tersebut memakan waktu di awal sistem berjalan. Tetapi, hal ini tidak akan dirasakan pada penggunaan sistem selanjutnya.

TABEL VI
LAMA WAKTU MENDAPATKAN REKOMENDASI DENGAN VARIASI YANG DIBERIKAN

Query	**Teknik filterisasi**	***Load* halaman ke-n**	**Lama waktu (S)**
dashboard	Mirip saya	1	49.43
dashboard	Mirip saya	2	6.28

IV. KESIMPULAN

Sistem rekomendasi topik skripsi berhasil dibangun dan memberikan rekomendasi yang tepat menggunakan kombinasi model yang telah dirancang. Sistem dibangun menggunakan data latih nilai mahasiswa yang sudah skripsi dengan jumlah sebanyak 168 data nilai. Data nilai tersebut telah diproses melalui tahap persiapan data. Selain data nilai, data skripsi telah diproses pada tahap persiapan data skripsi. Proses persiapan data membuat data nilai dan skripsi menjadi siap dilatihkan.

Model *K-Means++* yang dibuat menghasilkan jumlah klasterisasi sebanyak 4 berdasarkan analisa yang telah dilakukan. Model *K-Means++* dilatih menggunakan data nilai yang telah di PCA. Penggunaan PCA dilakukan karena perbedaan diantara menggunakan dan tidak menggunakan PCA tidak mempengaruhi signifikan hasil model *K-Means++* yang dibuat.

Cosine similarity digunakan untuk mencari dokumen yang relevan dengan *query* yang diberikan oleh pengguna. Terdapat proses untuk menyamakan *term* yang memiliki makna sama, tapi ditulis berbeda. Proses penyamaan ini akan membuat hanya terdapat 1 variasi pernulisan terhadap suatu *term* tertentu.

LDA *Gibbs Sampling* yang dipakai akan membuat permodelan topik dari 1 hingga sebanyak 20. Setiap proses pembuatan topik, model akan membandingkan nilai *perplexity*, jumlah topik dengan nilai *perplexity* terendah adalah jumlah topik dimana hasil dari permodelan topik akan digunakan sebagai rekomendasi topik skripsi. Nilai iterasi maksimal pada setiap permodelan topik adalah 10. Selain itu, Parameter *alpha* dan *beta* memiliki nilai berlawanan dengan jumlah topik skripsi yang dibuat. Jika jumlah topik adalah 4, maka parameter *alpha* dan *beta* bernilai ¼. LDA yang diimplementasikan ke dalam sistem dibatasi maksimal memproses 29 dokumen. Hal ini bertujuan mempercepat proses komputasi dan memastikan *query* yang diberikan pengguna sudah cukup spesifik menyaring dokumennya.

Sistem menggunakan *Pyscript* untuk menjalankan *script* Python memberikan dampak buruk dengan lamanya proses eksekusi *script* Python di awal proses. Hal ini terjadi karena komputer *client* yang mengakses sistem ini perlu mengunduh *file* pendukung agar *Pyscript* dapat mengeksekusi *script* rekomendasi.

Pada penelitian selanjutnya sistem dapat dikembangkan menggunakan *framework Django* yang dirancang untuk mengeksekusi *script* Python. Selain itu, penentuan nilai parameter *alpha* dan *beta* perlu diteliti lebih baik dan LDA yang digunakan dapat menggunakan *library* permodelan topik yang lebih dikenal, seperti *Scikit-learn* atau *Gensim*. Selain itu, jumlah dimensi yang dikurangi menggunakan PCA perlu untuk diuji pada berbagai macam jumlah. Pada penelitian ini hanya menguji dengan jumlah 20 saja.

UCAPAN TERIMA KASIH

Penelitian ini didanai dan didukung oleh Fakultas Teknologi Informasi Universitas Kristen Duta Wacana.

REFERENSI

[1] G. Janura and Ahyanuardi, "Analisis Kendala Mahasiswa dalam Penyelesaian Skripsi pada Masa Pandemi Covid-19," *Jurnal Pendidikan Teknik Elektro*, vol. 2, no. 2, pp. 97-102, 2021.

[2] B. K. Francis and S. S. Babu, "Predicting Academic Performance of Students Using a Hybrid Data Mining Approach," *Journal of Medical Systems*, vol. 43, p. 162, 2019.

[3] Haviluddin, S. J. Patandianan, G. M. Putra, N. Puspitasari and H. S. Pakpahan, "Implementasi Metode K-Means untuk Pengelompokkan Rekomendasi," *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, 2021.

[4] B. Aubaidan, M. Mohd and M. Albared, "Comparative Study of K-means and K-means++ Clustering on Crime Dromain," *Journal of Computer Science*, vol. 10, no. 7, pp. 1197-1206, 2014.

[5] N. T. Hartanti, "Metode Elbow dan K-Means Guna Mengukur Kesiapan Siswa SMK Dalam Ujian Nasional," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 6, no. 2, pp. 82-89, 2020.

[6] N. P. E. Merliana, Ernawati and A. J. Santoso, "Analisa Penentuan Jumlah Cluster Terbaik Pada Metode K-Means Clustering," in *PROSIDING SEMINAR NASIONAL MULTI DISIPLIN ILMU*, Yogyakarta.

[7] P. Prabhu and N. Anbazhagan, "Improving the Performance of K-Means Clustering For High Dimensional Data Set," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 6, pp. 2317-2322, 2011.

[8] A. Toraismaya, L. R. Sasongko and F. S. Rondonuwu, "Principal Component Dan K-Means Cluster Analysis Untuk Data Spektrum Black Tea Grades Guna Penilaian Kualitas Alternatif," *Journal of Fundamental Mathematics and Applications (JFMA)*, vol. 3, no. 2, 2020.

[9] R. N. Afifuddin and D. Nurjanah, "Sistem Rekomendasi Pemilihan Mata kuliah Peminatan Menggunakan Algoritma K-means dan Apriori (studi kasus: Jurusan S1 Teknik Informatika Fakultas Informatika)," in *e-Proceeding of Engineering*, 2019.

[10] D. Kurniadi, S. F. C. Haviana and A. Novianto, "Implementasi Algoritma Cosine Similarity pada sistem arsip dokumen di Universitas Islam Sultan Agung," *Transformatika*, vol. 17, no. 2, pp. 124-132, 2020.

[11] Apriani, H. Zakiyudin and K. Marzuki, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta," *Jurnal Bumigora Information Technology (BITe)*, vol. 3, no. 1, pp. 19-27, 1 June 2021.

[12] A. Nurlayli and M. A. Nasichuddin, "Topic Modeling Penelitian Dosen JPTEI UNY pada Google Scholar Menggunakan Latent Dirichlet Allocation," *ELINVO (Electronics, Informatics, and Vocational Education)*, vol. 4, no. 2, pp. 154-161, 2019.

[13] I. M. K. B. Putra and n. R. P. Kusumawardani, "Analisis Topik Informasi Publik Media Sosial di Surabaya Menggunakan Pemodelan Latent Dirichlet Allocation (LDA)," *Jurnal Teknik ITS*, vol. 6, no. 2, 2017.

[14] M. F. A. Bashiri, "Analisa Sentimen Menggunakan Latent Dirichlet Allocation dan Visualisasi Topic Popularity Wordcloud," Ungaran, 2017.

[15] A. Haryoso, "Analisis Website Performance Milik Kementerian di Indonesia Menggunakan Metode Pembobotan Entropi Dan Metode Pemeringkatan Electre," https://eprints.uny.ac.id/54806/, Yogyakarta, 2017.

[16] I. N. S. W. Wijaya, I. B. A. I. Iswara and I. N. A. Arsana, "Analisis dan Evaluasi Pengalaman Pengguna PaTik Bali Dengan Metode User Experience Questionnaire (UEQ)," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 2, pp. 217-226, 2021.

[17] R. Y. Sari, H. Oktavianto and H. W. Sulistyo, "Algoritma K-Means Dengan Metode Elbow Untuk Mengelompokkan Kabupaten/Kota Di Jawa Tengah Berdasarkan Komponen Pembentuk Indeks Pembangunan Manusia," *Jurnal Smart Teknologi*, vol. 3, no. 2, pp. 104-108, 2022.

[18] T. Santika, "Evaluasi Perplexity Untuk Pemodelan Topik Diskusi Agama Islam di Media Sosial Twitter Indonesia Tahun 2006-2018 Menggunakan Latent Dirichlet Allocation," *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 3, pp. 122-129, 2021.

[19] Y. Sahria and D. H. Fudholi, "Analisis Topik Penelitian Kesehatan di Indonesia Menggunakan Metode Topic Modeling LDA (Latent Dirichlet Allocation)," *Jurnal Rekayasa Sistem dan Teknologi Informasi*, vol. 4, no. 2, p. 336–344, 2020.

[20] A. M. Rukmi and I. M. Iqbal, "Using k-means++ algorithm for researches clustering," in *AIP Conference Proceedings*, 2017.

[21] A. Riyani, M. Z. Nafan and A. B. Hanuddin, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," *Journal Linguistik Komputasional*, vol. 2, no. 1, 2019.

[22] A. Rahman, R. B. Waskitho, M. F. A. U. Nuha and N. A. Rakhmawati, "Klasterisasi Topik Konten Channel Youtube Gaming Indonesia Menggunakan Latent Dirichlet Allocation," *Journal Information Engineering and Educational Technology*, vol. 5, no. 2, 2021.

[23] D. Purwitasari, A. Muflichah, N. A. Hasanah and A. Z. Arifin, "Pemodelan Topik dengan LDA untuk Temu Kembali Informasi dalam Rekomendasi Tugas Akhir," *Jurnal Rekayasa Sistem dan Teknologi Informasi*, vol. 5, no. 3, pp. 421-428, 2021.

[24] N. Nugroho and F. D. Adhinata, "Penggunaan Metode K-Means dan K-Means++ Sebagai Clustering Data Covid-19 di Pulau Jawa," *Teknika*, vol. 11, no. 3, pp. 170-179, 2022.

[25] A. Y. Nugraha and F. F. Abdulloh, "Optimasi Naive Bayes dan Cosine Similarity Menggunakan Particle," *Jurnal Media Informatika Budidarma*, vol. 6, no. 3, pp. 1444-1451, 2022.

[26] N. P. E. Merliana, Ernawati and A. J. Santoso, "Analisa Penentuan Jumlah Cluster Terbaik Pada Metode K-Means Clustering," in *Prosiding Seminar Nasional Multi Disiplin Ilmu*, Yogyakarta.

[27] I. W. Jepriana and S. Hanief, "Metode item-based Collaborative Filtering Untuk Model Sistem Rekomendasi Konsentrasi Penjurusan di STMIK STIKOM Bali," *Jurnal Teknologi Informasi dan Komputer*, vol. 6, no. 1, pp. 22-28, 2020.

[28] F. O. Isinkaye, Y. O. Yolajimi and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261-273, 2015.

[29] S. R. Henim and R. P. Sari, "Evaluasi User Experience Sistem Informasi Akademik Mahasiswa pada Perguruan Tinggi Menggunakan User Experience Questionnaire," *Jurnal Komputer Terapan*, vol. 6, no. 1, p. 69–78, 2020.

[30] J. Boyd-Graber, Applications of Topic Models, Boulder: now publishers, 2017, pp. 1-154.

[31] S. Barber, "How Fast Does a Website Need To Be?," 2009. [Online]. Available: http://www.perftestplus.com/resources/how_fast.pdf.

[32] N. Anbazhagan, "Improving the Performance of K-Means," *International Journal on Computer Science and Engineering (IJCSE),* vol. 3, no. 6, pp. 2317-2322, 2011.

[33] A. I. Alfanzar, Khalid and I. S. Rozas, "Topic Modelling Skripsi Menggunakan Metode Latent Dirichlet Allocation," *Jurnal Sistem Informasi,* vol. 7, no. 1, pp. 7-13, March 2020.

[34] F. O. Isinkaye, Y. O. Yolajimi and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal,* pp. 261-273, 2015.

[35] I. W. Jepriana and S. Hanief, "Metode item-based Collaborative Filtering Untuk Model Sistem Rekomendasi Konsentrasi Penjurusan di STMIK STIKOM Bali," *Jurnal Teknologi Informasi dan Komputer,* 2020.