

**SISTEM REKOMENDASI PEKERJAAN MENGGUNAKAN PENDEKATAN
CONTENT-BASED FILTERING**

SKRIPSI

Oleh:
ALYA FITRIA
NIM. 200605110022



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**SISTEM REKOMENDASI PEKERJAAN MENGGUNAKAN
PENDEKATAN CONTENT-BASED FILTERING**

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
ALYA FITRIA
NIM. 200605110022

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN

SISTEM REKOMENDASI PEKERJAAN MENGGUNAKAN
PENDEKATAN CONTENT-BASED FILTERING


SKRIPSI

Oleh:

ALYA FITRIA
NIM. 200605110022

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 3 Juni 2024

Pembimbing I,



Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005


Pembimbing II,



Dr. M. Ainul Yaqin, M.Kom
NIP. 19761013 200604 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachri Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

SISTEM REKOMENDASI PEKERJAAN MENGGUNAKAN
PENDEKATAN CONTENT-BASED FILTERING

SKRIPSI

Oleh:
ALYA FITRIA
NIM. 200605110022

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 12 Juni 2024

Susunan Dewan Penguji

Ketua Penguji : Dr. Totok Chamidy, M. Kom
NIP. 19691222 200604 1 001

Anggota Penguji I : Ashri Shabrina Afrah, M.T
NIP. 19900430 202012 2 003


Anggota Penguji II : Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005

Anggota Penguji III : Dr. M. Ainul Yaqin, M.Kom
NIP. 19761013 200604 1 004

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrud Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Alya Fitria
NIM : 200605110022
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Sistem Rekomendasi Pekerjaan Menggunakan Pendekatan Content-Based Filtering

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 14 Juni 2024
Yang membuat pernyataan,



Alya Fitria
NIM. 200605110022

MOTTO

“Sesungguhnya Allah tidak mengubah keadaan suatu kaum sehingga mereka mengubah keadaan yang ada pada diri mereka sendiri”

(Q.S. Ar-Ra’d: 11)

HALAMAN PERSEMBAHAN

Dengan rasa syukur yang mendalam kepada Allah SWT, yang telah memberikan rahmat dan hidayah-Nya, penulis ingin mempersembahkan skripsi ini kepada seluruh pihak yang berperan aktif menyelesaikan penelitian ini

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Puji syukur kepada Allah SWT yang telah melimpahkan rahmat, hidayah, dan kesempatan bagi penulis untuk menyelesaikan skripsi ini dengan baik dan sesuai yang telah direncanakan. Penulisan skripsi ini tidak terlepas dari dukungan dan bantuan berbagai pihak yang senantiasa memberikan dorongan, motivasi, serta bimbingan dalam proses penelitian ini. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Syahiduz Zaman, M.Kom selaku dosen pembimbing I dan Dr. M. Ainul Yaqin, M.Kom selaku dosen pembimbing II yang telah memberikan bantuan dan arahan kepada penulis, sehingga bisa menuntaskan skripsi ini.
5. Dr. Totok Chamidy, M. Kom selaku dosen penguji I dan Ashri Shabrina Afrah, M.T selaku dosen penguji II yang telah menguji serta memberikan masukan, sehingga penulis dapat menuntaskan skripsi dengan baik.
6. Segenap Dosen, Admin, Laboran dan Mahasiswa Program Studi Teknik Informatika yang telah memberi bimbingan selama pengerjaan skripsi.

7. Keluarga penulis terutama untuk kedua orang tua yaitu Abah Ahmad Amir Aziz dan Mama Layin Kholifah, serta untuk adik-adik penulis yaitu Alfita dan Alfian yang telah menjadi motivasi utama untuk dan menyelesaikan skripsi.
8. Teman-teman mahasiswa Teknik Informatika angkatan 2020 (INTEGER) yang senantiasa berbagi informasi seputar tugas sampai skripsi.
9. Teman-teman Pejuang S.Kom, Ekata, Himatif dan Sambat Kuliah yang telah menemani proses pengerjaan skripsi dan memberi pengalaman baru selama perkuliahan kepada penulis.
10. Sahabat-sahabat yang berbeda kampus dan berada jauh tetapi tetap saling mendukung dan mendoakan, yaitu Alivia, Naya, Deya, Shema dan Lana.
11. Seluruh pihak yang terlibat, yang mana namanya tidak dapat penulis sebutkan satu persatu, semoga Allah SWT balas dengan kebaikan yang tiada habisnya.
12. Terimakasih untuk diri saya sendiri yang telah kuat, tetap bertahan, menjadi berani dan menolak putus asa selama menjalani serangkaian proses pengerjaan skripsi ini.

Akhir kata, penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan. Penulis berharap skripsi ini dapat memberikan banyak manfaat kepada pembaca dan utamanya kepada penulis sendiri.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 14 Juni 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO..	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث.....	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
BAB II STUDI PUSTAKA	7
2.1 Penelitian Terdahulu	7
2.2 Sistem Rekomendasi	10
2.3 Pekerjaan	12
2.4 <i>Content Based Filtering</i>	13
2.5 TF-IDF	16
2.6 <i>Cosine Similarity</i>	17
2.7 <i>Black Box Testing</i>	19
2.8 <i>Precision</i>	19
BAB III DESAIN DAN IMPLEMENTASI	21
3.1 Tahapan Penelitian	21
3.1.1 Pengumpulan Data	21
3.1.2 <i>Preprocessing Text</i>	26
3.1.3 <i>Content Based Filtering</i>	29
3.1.4 Rencana Pengujian.....	39
3.2 Desain Sistem	39
BAB IV HASIL DAN PEMBAHASAN.....	42
4.1 Hasil	42
4.2 Pengujian.....	62
4.2.1 <i>Black Box Testing</i>	63
4.2.2 <i>Precision</i>	63
4.3 Pembahasan.....	69
BAB V KESIMPULAN DAN SARAN	75

5.1 Kesimpulan.....	75
5.2 Saran.....	75

DAFTAR PUSTAKA

LAMPIRA

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu.....	9
Tabel 3.1 Sampel data lowongan pekerjaan.....	23
Tabel 3.2 Sampel data profil pencari kerja	25
Tabel 3.3 Contoh data sesudah preprocessing	29
Tabel 3.4 Perhitungan TF.....	32
Tabel 3.5 Perhitungan IDF	33
Tabel 3.6 Perhitungan TF-IDF.....	34
Tabel 3.7 Perhitungan dot product	36
Tabel 3.8 Perhitungan magnitude	37
Tabel 3.9 Perhitungan cosine similarity	38
Tabel 4.1 Hasil rekomendasi pencari kerja index ke-0.....	56
Tabel 4.2 Pengujian fungsionalitas black box.....	63
Tabel 4.3 Nilai similarity skenario 1	64
Tabel 4.4 Nilai precision skenario 1	65
Tabel 4.5 Nilai similarity skenario 2.....	65
Tabel 4.6 Nilai precision skenario 2	66
Tabel 4.7 Nilai similarity skenario 3.....	66
Tabel 4.8 Nilai precision skenario 3	67
Tabel 4.9 Nilai similarity skenario 4.....	67
Tabel 4.10 Nilai precision skenario 4	68
Tabel 4.11 Nilai similarity skenario 5.....	68
Tabel 4.12 Nilai precision skenario 5	69
Tabel 4.13 Hasil rata-rata precision	70

DAFTAR GAMBAR

Gambar 2.1 Skema content-based filtering.....	14
Gambar 3.1 Desain penelitian	21
Gambar 3.2 Flowchart scraping lowongan pekerjaan.....	22
Gambar 3.3 Flowchart scraping pencari kerja	24
Gambar 3.4 Tahapan preprocessing	26
Gambar 3.5 Flowchart content-based filtering.....	30
Gambar 3.6 Flowchart TF-IDF	31
Gambar 3.7 Flowchart cosine similarity	35
Gambar 3.8 Desain sistem.....	39
Gambar 4.1 Dataset lowongan pekerjaan.....	42
Gambar 4.2 Dataset pencari kerja	43
Gambar 4.3 Dataset lowongan pekerjaan setelah penggabungan kolom.....	43
Gambar 4.4 Dataset pencari kerja setelah setelah penggabungan kolom	44
Gambar 4.5 Halaman dataset	44
Gambar 4.6 Halaman perhitungan metode di dataset lowongan pekerjaan	45
Gambar 4.7 Halaman perhitungan metode di dataset pencari kerja.....	47
Gambar 4.8 Halaman form rekomendasi input dataset.....	50
Gambar 4.9 Halaman form rekomendasi input manual	51
Gambar 4.10 Halaman perhitungan hasil.....	52
Gambar 4.11 Halaman hasil rekomendasi	56
Gambar 4.12 Grafik 10 current position terbanyak	64

ABSTRAK

Fitria, Alya. 2024. **Sistem Rekomendasi Pekerjaan Menggunakan Pendekatan Content-Based Filtering**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Syahiduz Zaman, M.Kom. (II) Dr. M. Ainul Yaqin, M.Kom.

Kata Kunci: *Pekerjaan, Content Based Filtering, Sistem Rekomendasi*

Pada era digital dan globalisasi saat ini, transformasi teknologi informasi telah mengubah lingkup pencarian pekerjaan, dengan *platform online* seperti *LinkedIn* menjadi alat utama bagi pencari kerja dan perusahaan. Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi pekerjaan menggunakan metode *content-based filtering*, yang mencocokkan profil pencari kerja dengan lowongan pekerjaan berdasarkan karakteristik dan preferensi individu. Data diperoleh melalui *web scraping* dari situs *JobStreet* untuk data lowongan pekerjaan dan *LinkedIn* untuk data pencari kerja, meliputi 437 data lowongan pekerjaan dan 100 data profil pencari kerja. Proses analisis melibatkan *preprocessing text*, pembobotan kata dengan TF-IDF, dan perhitungan *cosine similarity* untuk menentukan tingkat kemiripan antar dokumen. Hasil pengujian menunjukkan bahwa sistem rekomendasi yang dikembangkan dapat memberikan rekomendasi yang relevan dengan rata-rata nilai presisi sebesar 0.53. Pengujian fungsionalitas dengan metode *blackbox testing* menghasilkan kinerja sistem yang sesuai dengan fungsinya. Penelitian ini menyimpulkan bahwa pendekatan *content-based filtering* efektif dalam menciptakan rekomendasi pekerjaan yang sesuai dengan latar belakang dan keterampilan pencari kerja, memberikan solusi praktis bagi mereka dalam menemukan pekerjaan yang relevan serta mengurangi ketidaksesuaian antara kualifikasi individu dan kebutuhan pasar kerja.

ABSTRACT

Fitria, Alya. 2024. **Job Recommendation System Using a Content-Based Filtering Approach.** Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Syahiduz Zaman, M.Kom. (II) Dr. M. Ainul Yaqin, M.Kom.

In the current era of digitalization and globalization, the transformation of information technology has significantly changed the landscape of job searching, with online platforms such as LinkedIn becoming primary tools for job seekers and companies. This research aims to develop a job recommendation system using the content-based filtering method, which matches job seekers' profiles with job vacancies based on individual characteristics and preferences. Data was collected through web scraping from JobStreet for job vacancy data and LinkedIn for job seeker data, comprising 437 job vacancies and 100 job seeker profiles. The analysis process involved text preprocessing, term weighting with TF-IDF, and cosine similarity calculations to determine the degree of similarity between documents. The testing results indicate that the developed recommendation system can provide relevant recommendations with an average precision score of 0.53. Functionality testing using the blackbox testing method confirmed that the system performs according to its intended functions. This research concludes that the content-based filtering approach is effective in creating job recommendations that align with the background and skills of job seekers, offering practical solutions for finding relevant employment and reducing the mismatch between individual qualifications and labor market needs.

Keywords: *Jobs, Content Based Filtering, Recommendation System*

مستخلص البحث

فيتريا، عليا. 2024. نظام توصيات الوظائف باستخدام نهج التصنيفية المعتمد على المحتوى. أطروحة. برنامج دراسة هندسة شهيدوز (I): المعلومات، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرفون د. م. عينول يقين، ماجستير (II). زمان، ماجستير

الكلمات الرئيسية : وظائف، تصنيفية مبنية على المحتوى، نظام التوصية

في عصر الرقمنة والعولمة الحالي، أدى تحول تكنولوجيا المعلومات إلى تغيير كبير في مشهد البحث عن الوظائف، حيث أدوات رئيسية للباحثين عن العمل والشركات. تهدف هذه الدراسة إلى تطوير نظام LinkedIn أصبحت المنصات الإلكترونية مثل توصيات للوظائف باستخدام طريقة التصنيفية المعتمدة على المحتوى، والتي تطابق ملفات الباحثين عن العمل مع الوظائف الشاغرة بناءً للحصول على بيانات JobStreet على الخصائص والتفضيلات الفردية. تم جمع البيانات من خلال استخراج البيانات من موقع للحصول على بيانات الباحثين عن العمل، حيث تم جمع 437 وظيفة شاغرة و100 ملف LinkedIn الوظائف الشاغرة ومن وحسابات تشابه TF-IDF باحث عن عمل. تضمنت عملية التحليل معالجة النصوص الأولية، وتقييم المصطلحات باستخدام جيب التمام لتحديد درجة التشابه بين الوثائق. أشارت نتائج الاختبار إلى أن نظام التوصيات المطور يمكنه تقديم توصيات ذات صلة بدقة متوسطة تبلغ 0.53. أثبتت اختبارات الوظائف باستخدام طريقة اختبار الصندوق الأسود أن النظام يعمل وفقاً لوظائفه المقصودة. تستنتج هذه الدراسة أن نهج التصنيفية المعتمد على المحتوى فعال في إنشاء توصيات للوظائف تتماشى مع خلفية الباحثين عن العمل ومهاراتهم، مما يوفر حلولاً عملية لهم في العثور على وظائف ذات صلة ويقلل من عدم التوافق بين مؤهلات الأفراد واحتياجات سوق العمل.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era digital dan globalisasi saat ini, dunia pekerjaan mengalami perubahan yang signifikan (Frey & Osborne, 2017). Transformasi teknologi informasi telah mengubah lingkup pencarian pekerjaan (Xie & Mujtaba, 2008), dengan *platform online* seperti *LinkedIn*, *JobStreet*, *Indeed*, dan *Glints* memungkinkan pencari kerja untuk menjelajahi peluang pekerjaan yang lebih luas. Banyaknya individu yang mencari pekerjaan melalui *platform online* (Dillahunt et al., 2021), hal ini mencerminkan adanya perubahan dalam perilaku pencarian kerja, dari metode konvensional menjadi lebih mengandalkan teknologi dan internet sebagai sumber utama untuk menemukan pekerjaan. Di sisi lain, perusahaan juga mencari cara yang lebih efisien untuk merekrut karyawan yang sesuai dengan kebutuhan mereka (Setiani, 2013). Sehingga dibutuhkan sistem rekomendasi pekerjaan yang bertujuan untuk mencocokkan profil individu pencari kerja dengan lowongan pekerjaan yang sesuai, sehingga memberikan manfaat ganda yaitu membantu individu menemukan pekerjaan yang sesuai dengan keahlian dan minat mereka, serta membantu perusahaan menemukan karyawan yang sesuai dengan kualifikasi mereka.

Disamping itu, tingginya Tingkat Pengangguran Terbuka (TPT) di Indonesia juga menjadi urgensi dalam dilaksanakannya penelitian ini. Berdasarkan hasil Survei Angkatan Kerja Nasional Indonesia (Sakernas) yang dilaksanakan oleh Badan Pusat Statistik (BPS) pada Februari 2023, sebanyak

7.989.275 orang atau sebesar 5,45% penduduk usia kerja tidak mempunyai pekerjaan. Menurut Menteri Tenaga Kerja (Menaker) RI, Ida Fauziyah, menyatakan bahwa besarnya jumlah pengangguran disebabkan oleh tidak adanya *link and match* antara perguruan tinggi dengan kebutuhan pasar kerja. Salah satu upaya yang dapat dilakukan dalam mengurangi pengangguran adalah melalui sistem rekomendasi pekerjaan yang mempertimbangkan latar belakang pekerja, sehingga dapat meningkatkan kesesuaian bidang pekerjaan dengan latar belakang pendidikan dan keterampilan pekerja. Dalam QS. Al-Jumu'ah (62:10), Allah SWT berfirman:

فَإِذَا قُضِيَتِ الصَّلَاةُ فَانْتَشِرُوا فِي الْأَرْضِ وَابْتَغُوا مِن فَضْلِ اللَّهِ وَاذْكُرُوا اللَّهَ كَثِيرًا لَّعَلَّكُمْ تُفْلِحُونَ

"Kemudian apabila telah disudahi shalat, maka bertebaranlah kamu di muka bumi dan carilah karunia Allah, dan ingatlah Allah banyak-banyak agar kamu berhasil." (QS. Al-Jumu'ah (62:10))

Pada ayat diatas, Allah memerintahkan umat-Nya untuk bertebaran di muka bumi setelah menyelesaikan sholat. Yang dimaksud bertebaran ialah umat Islam diharapkan untuk berperan aktif dalam kehidupan dunia, salah satunya bekerja atau mencari nafkah. Ayat ini juga mengandung pesan tentang usaha dalam mencari nafkah. Ketika Allah memerintahkan umat-Nya untuk mencari karunia-Nya, ini mengingatkan kita tentang pentingnya kerja keras, kecerdasan, dan usaha yang diperlukan dalam mencari pekerjaan yang sesuai dengan kemampuan dan keterampilan kita. Hal ini juga berkaitan dengan QS. Al-Qasas (28:77):

وَابْتَغِ فِيمَا آتَاكَ اللَّهُ الدَّارَ الْآخِرَةَ ۖ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا ۗ وَأَحْسِنَ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ
وَلَا تَبْغِ الْفَسَادَ فِي الْأَرْضِ ۗ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ ۗ

"Dan carilah dalam apa yang Allah telah anugerahkan kepadamu untuk negeri akhirat, dan janganlah kamu melupakan bagianmu dari dunia, dan berbuat baiklah sebagaimana Allah telah berbuat baik kepadamu. Dan janganlah kamu mencari kerusakan di muka bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan." (QS. Al-Qasas (28:77))

Ayat diatas menjelaskan bahwa Al-Qur'an mengajarkan umatnya untuk mencari kebaikan di dunia dan akhirat menggunakan anugerah yang Allah berikan. Anugerah yang dimaksud ini mencakup keterampilan dan potensi yang dimiliki setiap orang. Ayat tersebut juga mengingatkan bahwa Allah telah memberikan kemampuan unik kepada setiap individu, dan kita bertanggung jawab untuk memanfaatkan untuk mencapai kebaikan dan menjalankan tanggung jawab selama di dunia. Ketika kita bekerja sesuai dengan kemampuan di bidang tertentu, bukan hanya dapat mencapai keberhasilan dalam karir di dunia, tetapi juga mendapatkan keberkahan.

Pentingnya penelitian ini tercermin dalam perubahan paradigma pencarian kerja yang semakin terdigitalisasi. *LinkedIn* sebagai salah satu *platform* terbesar dalam jaringan profesional, memiliki potensi besar untuk meningkatkan efisiensi pencarian pekerjaan (Li et al., 2020). Namun, hingga saat ini, masih ada kendala dalam menyajikan rekomendasi pekerjaan yang sesuai dengan profil individu dengan relevansi tinggi. Kehadiran pandemi *COVID-19* juga telah mempercepat transformasi pekerjaan ke dunia digital (Tîrnovanu et al., 2023), menjadikan penggunaan *LinkedIn* semakin penting dalam mencari peluang kerja. Penggunaan *LinkedIn* sebagai *platform* sosial profesional telah menjadi pusat informasi yang mencerminkan latar belakang dan aktivitas karir individu (Johnson, 2019).

LinkedIn menjadi sumber berharga bagi individu untuk mencari pekerjaan, berkoneksi dengan profesional, dan membangun reputasi dalam industri (Aguado et al., 2019). Selain itu, *LinkedIn* juga mengizinkan pengguna untuk mempublikasikan *portfolio* keterampilan, sertifikasi, dan pengalaman kerja mereka, yang dapat dijadikan referensi utama bagi perusahaan dalam merekrut calon karyawan yang sesuai.

Meskipun banyak penelitian yang telah dilakukan dalam bidang sistem rekomendasi pekerjaan, masih ada kesenjangan dalam hal penggunaan data profil *LinkedIn* secara lebih mendalam untuk meningkatkan relevansi rekomendasi (Cubrich et al., 2021). Selain itu, sistem rekomendasi pekerjaan dengan pendekatan *content-based filtering* yang memanfaatkan data profil *LinkedIn* sebagai sumber informasi utama dan mencocokkan dengan konten lowongan pekerjaan juga belum banyak diteliti. Oleh karena itu, penelitian ini akan fokus pada pendekatan *content-based filtering* yang memanfaatkan data profil *LinkedIn* yang lebih mendalam untuk menghasilkan rekomendasi yang lebih personal dan relevan.

Metode yang akan digunakan dalam penelitian ini adalah *content-based filtering*. *Content-based filtering* merupakan salah satu metode dalam bidang sistem rekomendasi yang berfokus pada analisis konten atau karakteristik dari item yang akan direkomendasikan (Rashid et al., 2022). Dalam penelitian ini, item yang akan direkomendasikan adalah pekerjaan, metode *content-based filtering* akan digunakan untuk mencocokkan karakteristik dari pekerjaan yang tersedia dengan profil individu pencari kerja. Atribut-atribut yang digunakan dari

pekerjaan mencakup jenis pekerjaan, lokasi, dan keterampilan yang dibutuhkan akan digunakan sebagai fitur untuk analisis. Selain itu, preferensi pencari kerja akan diidentifikasi berdasarkan karakteristik profil mereka seperti fokus pendidikan, pengalaman kerja, kemampuan atau *skill*, sertifikasi, dan pencapaian karir. Salah satu kelebihan utama dari metode *content-based filtering* adalah kemampuannya untuk mengenali preferensi individu dengan lebih baik serta dapat menghasilkan rekomendasi yang lebih akurat dan relevan dibanding dengan metode rekomendasi lainnya (Kumar, 2022).

Berdasarkan uraian yang telah dijelaskan, penelitian ini bertujuan untuk mengembangkan sistem rekomendasi pekerjaan dengan metode *content-based filtering*. Dalam penelitian ini akan menghitung kedekatan fitur dari konten profil pencari kerja dengan lowongan pekerjaan yang tersedia. Hasil dari penelitian ini diharapkan dapat menjadi solusi bagi pencari kerja untuk menemukan pekerjaan yang sesuai dengan latar belakang, pengalaman, minat, dan potensi mereka.

1.2 Rumusan Masalah

Bagaimana membangun sistem rekomendasi pekerjaan menggunakan metode *content-based filtering* dan seberapa efektif sistem dalam memberikan rekomendasi yang relevan bagi pengguna?

1.3 Batasan Masalah

Berikut adalah beberapa batasan masalah dalam penelitian ini:

1. Bidang lowongan pekerjaan dan profil pencari kerja hanya berfokus pada bidang Teknologi Informasi (TI).

2. Objek penelitian yang digunakan adalah pengguna yang terdaftar di *LinkedIn*.
3. Sistem yang dikembangkan hanya untuk memberi rekomendasi kepada pencari kerja dalam menemukan pekerjaan yang sesuai dengan profilnya.

1.4 Tujuan Penelitian

Adapun tujuan dilakukannya penelitian ini adalah untuk membangun sistem rekomendasi pekerjaan dengan metode *content-based filtering* sehingga dapat memberikan rekomendasi relevan bagi pengguna. Tingkat relevansi dari hasil rekomendasi yang diberikan oleh sistem akan dinilai berdasarkan skor presisi.

1.5 Manfaat Penelitian

Berikut adalah beberapa manfaat dalam penelitian ini:

1. Secara praktis penelitian ini dapat memberikan manfaat bagi pencari kerja dalam menemukan pekerjaan yang relevan dengan latar belakangnya.
2. Secara teoritis penelitian ini dapat menambah pengetahuan tentang sistem rekomendasi pekerjaan menggunakan metode *content-based filtering* dalam memberikan rekomendasi yang relevan kepada pengguna.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terdahulu

Penelitian mengenai sistem rekomendasi pekerjaan dengan metode *content-based filtering* telah banyak dilakukan oleh peneliti sebelumnya. Seperti yang telah dilakukan oleh (Dhameliya & Desai, 2019) berjudul “*Job Recommendation System using Content and Collaborative Filtering based Techniques*” telah mengusulkan sistem rekomendasi pekerjaan dengan menggunakan metode gabungan, yaitu *Content Based Filtering* untuk memberi rekomendasi berdasarkan *skill* pengguna yang diidentifikasi dari pekerjaan sebelumnya, dan *Collaborative Filtering* untuk memberi rekomendasi berdasarkan pekerjaan yang sering dilamar pengguna atau pekerjaan yang telah dilamar oleh pengguna lain yang memiliki *skill* serupa. Pembobotan kata menggunakan model *skip-gram* dari *Word2Vec*. Dan perhitungan kemiripan diukur menggunakan *K-Nearest Neighbors* dan *Pearson Correlation Coefficient*. Data yang digunakan adalah dari situs Wuzzuf, sebuah platform pencari kerja di Mesir, yang terdiri dari 1.900 pekerjaan dan 23.000 pelamar pekerjaan. Hasil evaluasi sistem ini mendapat skor *recall* 0,6397.

Selanjutnya, penelitian oleh (Boukari et al., 2020) yang berjudul “*Huntalent: A Candidates Recommendation System for Automatic Recruitment via LinkedIn*” telah mengembangkan sistem rekomendasi pekerjaan untuk proses rekrutmen dengan menggunakan kerangka kerja *Apache Spark*. Metode *Content Based Filtering* digunakan untuk merekomendasikan dan mengidentifikasi calon-calon potensial, tingkat kemiripan antar calon diukur dengan *cosine similarity*.

Data yang digunakan terdiri dari 2.556 profil kandidat yang berasal dari jaringan sosial publik *LinkedIn* di berbagai negara dalam format JSON. Hasil evaluasi sistem ini mendapatkan skor *recall* 0,786 dan *precision* 0,433.

Penelitian lain dilakukan oleh (Mulay et al., 2022) berjudul “*Job Recommendation System Using Hybrid Filtering*” telah menggunakan metode *Content Based Filtering* dan *Collaborative Filtering* dalam mengembangkan sistem rekomendasi pekerjaan. Hasil rekomendasi akan bergantung pada pengalaman sebelumnya (*experiences, projects, skills, internships, etc*) pengguna serta data dari pengguna yang serupa, menggunakan algoritma TF-IDF, *cosine similarity*, dan *singular value decomposition*. Data yang digunakan adalah dari situs *AmbitionBox*, platform pencari kerja di India. Pengujian sistem ini mendapat hasil rata-rata skor *precision* 0,3918, *recall* 0,6362 dan *f-measure* 0,4730.

Selain itu, penelitian oleh (Rashid et al., 2022) berjudul “*Student Career Recommendation System Using Content Based Filtering Method*” mengembangkan sistem rekomendasi karir siswa dengan metode *Content Based Filtering*. Sistem ini dapat membantu mahasiswa dalam memilih karir yang sesuai dengan minat dan kepribadian mereka, dengan menggunakan algoritma TF-IDF dan *cosine similarity*. Data pekerjaan yang digunakan adalah dari situs *JobStreet*. Pengujian sistem ini dilakukan dengan pengujian fungsional *black box* dan *usability testing*, hasilnya memperoleh skor rata-rata *System Usability Scale* (SUS) sebesar 81,25%

Terakhir, penelitian oleh (Raharjo et al., 2022) berjudul “Sistem Rekomendasi *Content Based Filtering* Pekerjaan dan Tenaga Kerja Potensial

menggunakan *Cosine Similarity*” mengkaji sistem rekomendasi pekerjaan dengan pendekatan *Content Based Filtering* yang menggunakan *cosine similarity*. Dengan menggunakan data 2.484 *file resume* dan 18.892 data *job posting* yang diambil dari situs Kaggle, penelitian ini mendapatkan nilai hasil pengujian *Mean Reciprocal Rank* (MRR) sebesar 0,857 dan *Mean Average Precision* (MAP) sebesar 0,833.

Tabel 2.1 Penelitian terdahulu

Sumber	Objek	Metode			Variabel	Hasil
		Pra	Main	Post		
(Dhameliya & Desai, 2019)	Pekerjaan	Word2Vec	CF, CBF	K-NN, Pearson Correlation Coefficient	<i>skills</i>	Skor <i>recall</i> 0,6397
(Boukari et al., 2020)	Rekrutmen Pekerjaan	-	CBF	<i>Cosine Similarity</i>	4 (<i>current position, skill, education, experience</i>)	Skor <i>recall</i> 0,786 dan <i>precision</i> 0,433
(Mulay et al., 2022)	Pekerjaan	TF-IDF	CF, CBF	<i>Cosine Similarity</i>	4 (<i>experience, project, skill, internship</i>)	Skor <i>precision</i> 0,3918, <i>recall</i> 0,6362 dan <i>f-measure</i> 0,4730
(Rashid et al., 2022)	Karir	TF-IDF	CBF	<i>Cosine Similarity</i>	2 (<i>personality, interest</i>)	Skor SUS 81,25%
(Raharjo et al., 2022)	Pekerjaan dan Tenaga Kerja	TF-IDF, Word2Vec	CBF	<i>Cosine Similarity</i>	6 (<i>current position, highlight, accomplish, experience, education, skill</i>)	Skor MRR 0,857 dan MAP 0,833
(Alya, 2024)	Pekerjaan	TF-IDF	CBF	<i>Cosine Similarity</i>	7 (<i>current position, highlight, skill, education, experience, license, certification</i>)	...

Penelitian terdahulu yang telah dilakukan telah memberi banyak pengetahuan mengenai sistem rekomendasi pekerjaan dengan metode *content-based filtering*, penelitian selanjutnya akan mengusulkan sistem rekomendasi

pekerjaan *content-based filtering* yang menggunakan algoritma TF-IDF dan *cosine similarity* dengan memanfaatkan data profil pengguna *LinkedIn* di Indonesia, yang belum pernah diteliti sebelumnya. Akan diambil 7 variabel dari data tersebut, yaitu *current position*, *highlight*, *skill*, *education*, *experience*, *license*, dan *certification*. Evaluasi sistem akan dilakukan dengan menggunakan parameter dari salah satu variabel guna dapat menemukan nilai presisi yang lebih optimal dari hasil rekomendasi yang diberikan oleh sistem.

2.2 Sistem Rekomendasi

Sistem rekomendasi adalah solusi teknologi yang digunakan untuk memberikan saran dan rekomendasi yang dipersonalisasi kepada pengguna berdasarkan preferensi dan minat mereka (Jannach et al., 2010). Tujuan dari sistem rekomendasi adalah untuk memberikan rekomendasi yang sesuai dengan kebutuhan pengguna, membantu pengguna dalam mengatasi masalah kelebihan informasi (Hosseini et al., 2013). Dengan sistem rekomendasi, pengguna dapat menemukan item yang relevan dari banyaknya opsi atau pilihan yang tersedia. Sehingga dapat membantu dalam proses pengambilan keputusan dengan menyarankan opsi yang lebih relevan berdasarkan latar belakang atau pengalaman pengguna.

Penerapan sistem rekomendasi telah banyak dikembangkan dalam berbagai aplikasi, salah satunya aplikasi streaming film *Netflix*. Dengan menganalisis riwayat film yang telah ditonton oleh pengguna, sistem akan dapat memberikan rekomendasi film berdasarkan karakteristik item *film* yang disukai, dengan *genre*

serupa, ia akan merekomendasikan juga kepada pengguna dengan preferensi yang sama. Dalam penggunaannya, sistem rekomendasi yang dikembangkan penelitian ini mampu membantu seseorang dalam memberi petunjuk pekerjaan yang sesuai. Hal ini berkaitan dengan dengan prinsip ajaran Islam yang menekankan pentingnya memberikan petunjuk yang baik kepada sesama, melalui ayat Al-Qur'an berikut:

وَكُنْتُمْ خَيْرَ أُمَّةٍ أُخْرِجَتْ لِلنَّاسِ تَأْمُرُونَ بِالْمَعْرُوفِ وَتَنْهَوْنَ عَنِ الْمُنْكَرِ وَتُؤْمِنُونَ بِاللَّهِ

"Kamu (umat islam) adalah umat terbaik yang dilahirkan untuk (memberi) petunjuk kepada manusia, menyuruh kepada yang ma'ruf, dan mencegah dari yang munkar, dan beriman kepada Allah. " (QS. Al-Imran: 3:110)

Keterkaitan ayat diatas dengan sistem rekomendasi dapat dilihat dari tugas umat Islam yang dinyatakan sebagai "umat terbaik yang dilahirkan untuk memberi petunjuk kepada manusia, menyuruh kepada yang ma'ruf, dan mencegah dari yang munkar, dan beriman kepada Allah." Sistem rekomendasi bertujuan untuk memberikan panduan yang tepat kepada penggunanya, sejalan dengan bagaimana umat Islam diarahkan untuk memberi petunjuk kepada manusia. Dalam memberi rekomendasi pekerjaan, sistem ini membantu pengguna menemukan pekerjaan yang sesuai dengan keterampilan dan minat mereka, yang mencerminkan tugas menyuruh kepada yang ma'ruf (kebaikan). Selain itu, sistem ini juga berfungsi untuk menyaring dan mencegah pengguna dari rekomendasi yang tidak relevan atau merugikan, selaras dengan tugas mencegah dari yang munkar (keburukan).

2.3 Pekerjaan

Pekerjaan merujuk pada aktivitas yang dilakukan oleh individu dalam suatu organisasi atau lingkungan yang melibatkan tanggung jawab, tugas, dan fungsi tertentu. Pekerjaan dapat berkisar dari berbagai industri dan sektor, dan karakteristiknya dapat mencakup deskripsi pekerjaan, kualifikasi yang diperlukan, lokasi, tingkat pengalaman, gaji, dan banyak aspek lainnya (Van Zyl, 1999). Pentingnya melaksanakan pekerjaan dengan sebaik-baiknya dan penuh tanggungjawab merupakan anjuran dalam Islam, karena Allah menyukai usaha dan dedikasi yang tinggi dalam setiap tindakan dan pekerjaan yang dilakukan oleh umat-Nya. Hal ini disebutkan dalam hadist berikut:

مِنْ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ: "إِنَّ اللَّهَ يُحِبُّ إِذَا عَمِلَ أَحَدُكُمْ عَمَلًا أَنْ يُتَّقِنَهُ" (رواه البيهقي)

“Dari Abu Hurairah radhiyallahu ‘anhu, dia berkata, Rasulullah SAW bersabda, ‘Sesungguhnya Allah mencintai seseorang ketika dia melakukan suatu pekerjaan, dia melakukannya dengan sebaik-baiknya.’” (Hadis Riwayat al-Bayhaqi).

Korelasi hadist tersebut dengan penelitian ini adalah ketika seseorang mencintai pekerjaannya, maka ia akan mampu memberikan dedikasi tinggi dalam melakukan setiap pekerjaan. Dan untuk mencapai dedikasi yang tinggi tersebut maka diperlukan kesesuaian antara kemampuan dengan pekerjaan yang dijalani. Ketika individu mengerjakan tindakan yang disukai, maka ia akan melakukannya dengan sebaik-baiknya. Hal ini selaras dengan tujuan penelitian ini, yaitu untuk memberikan rekomendasi pekerjaan yang sesuai dengan kemampuan dan latar belakang pencari kerja. Sehingga dengan adanya sistem rekomendasi ini maka

akan dapat membatu individu dalam menemukan pekerjaan yang sesuai dengan kemampuannya.

Disamping itu, karakteristik pekerjaan telah mengalami transformasi yang signifikan di abad ke-21, dipengaruhi oleh keberadaan dan perkembangan teknologi, yang telah menciptakan tantangan baru dalam bidang karir dan pencarian pekerjaan (Fayard, 2021). Salah satu perubahan yang terlihat adalah peralihan proses pencarian pekerjaan ke ranah *online*. Dengan adanya *platform* dan situs penyedia lowongan pekerjaan, individu dapat dengan mudah mencari pekerjaan tanpa batasan geografis. Informasi detail mengenai pekerjaan dapat diakses secara online, memberikan peluang bagi pencari kerja untuk menemukan banyak pekerjaan tersedia. Selain itu, konsep *work from home* atau bekerja dari rumah juga berkembang pesat, memungkinkan individu untuk dapat bekerja tanpa harus berada di lokasi fisik tertentu. Fenomena ini mencerminkan pergeseran paradigma dalam cara pekerjaan diakses, ditemukan, dan dilakukan, yang semakin terintegrasi dengan kemajuan teknologi informasi.

2.4 Content Based Filtering

Content based filtering adalah salah satu metode yang digunakan dalam sistem rekomendasi untuk memberikan rekomendasi yang personal kepada pengguna (Ricci et al., 2011). Sistem rekomendasi berbasis konten adalah sebuah pendekatan yang difokuskan pada aspek “konten” dari item yang akan direkomendasikan kepada pengguna (Aggarwal, 1997). Terdapat dua komponen utama yang dibutuhkan dalam mengembangkan sistem rekomendasi dengan *content-based filtering*, yaitu item yang akan direkomendasikan dan profil

pengguna. Item yang dimaksud dalam penelitian ini adalah pekerjaan, representasi dari pekerjaan adalah atribut atau fitur yang tersedia pada konten atau teks deskripsi pekerjaan itu sendiri, sedangkan representasi profil pengguna adalah bentuk preferensi dan minat pengguna. Dari dua komponen tersebut kemudian akan menghasilkan rekomendasi yang dibuat berdasarkan kemiripan antara item dengan profil pengguna. Fitur dalam item ini akan disaring dan dicocokkan dengan profil pengguna, sehingga menghasilkan nilai kemiripan yang dapat memprediksi sejauh mana profil pengguna cocok dengan item tersebut. Skema dalam memberikan rekomendasi ini ditunjukkan pada gambar dibawah ini.



Gambar 2.1 Skema *content-based filtering*

Proses penyaringan dan pencocokan dalam *content-based filtering* ini berkaitan dengan sabda Allah dalam Al-Qur'an surat Al-Hujurat (49:6):

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهْلَةٍ فَتُصْحَبُوا عَلَىٰ مَا فَعَلْتُمْ
تُدْمِين

"Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu." (QS. Al-Hujurat: 49:6)

Ayat tersebut berkaitan secara konsep tentang pentingnya melakukan analisis dan verifikasi. Ayat ini mengajarkan pentingnya memeriksa kebenaran dan keaslian informasi sebelum menerima dan menyebarkannya. Hal ini mencerminkan prinsip *content-based filtering* dimana sistem harus melakukan analisis terhadap konten untuk memastikan bahwa rekomendasi yang diberikan sesuai dengan kebutuhan pengguna, akurat dan berkualitas.

Penggunaan metode *content-based filtering* dalam sistem rekomendasi adalah sebagai paradigma atau konsep umum yang mengacu pada pendekatan yang menggunakan “konten” dari item serta preferensi pengguna untuk menyediakan rekomendasi yang sesuai. Dalam implementasinya, *content-based filtering* tidak memiliki algoritma khusus untuk dirinya sendiri, pendekatan ini dapat menggunakan berbagai algoritma lainnya seperti TF-IDF, *Cosine Similarity*, *Vector Space Model*, KNN, *Naive Bayes Classifier*, *Decision Trees* dan lain sebagainya.

Salah satu kombinasi algoritma yang umum digunakan dalam *content-based filtering* adalah TF-IDF (*Term Frequency-Inverse Document Frequency*) dan *Cosine Similarity* (Adomavicius & Tuzhilin, 2005). Penggunaan dua algoritma tersebut juga akan diimplementasikan dalam penelitian ini karena cocok dengan karakteristik pada dataset yang akan digunakan. Kombinasi ini efektif dalam membuat sistem rekomendasi berbasis teks, dimana TF-IDF digunakan untuk mengukur bobot setiap kata dalam dokumen, dan *cosine similarity* digunakan untuk mengukur kesamaan antara dokumen-dokumen berdasarkan representasi vektor TF-IDF mereka.

2.5 TF-IDF

Term Frequency Inverse Document Frequency atau TF-IDF adalah sebuah teknik dasar dalam pemrosesan teks yang digunakan untuk mengukur tingkat kepentingan sebuah kata dalam suatu dokumen atau sekumpulan dokumen. Teknik ini akan mengubah dokumen teks menjadi vektor fitur, dengan setiap kata dalam dokumen dianggap sebagai fitur (Feldman & Sanger, 2006). Setiap dokumen diwakili sebagai titik dalam ruang fitur, di mana kata-kata menjadi dimensi vektor dan bobotnya dihitung menggunakan TF-IDF.

Tahapan TF-IDF melibatkan dua komponen utama. Pertama, *Term Frequency* (TF) menghitung seberapa sering sebuah kata tertentu muncul dalam sebuah dokumen, dan kedua, *Inverse Document Frequency* (IDF) menghitung seberapa jarang sebuah kata tertentu muncul dalam sebuah dokumen (Jannach et al., 2010). Untuk mengukur nilai TF atau seberapa sering kata / term (t) muncul dalam dokumen (D), digunakan rumus pada Persamaan 2.1.

$$TF_{(t,D)} = \frac{freq(t,D)}{maxOthers(t,D)} \quad (2.1)$$

Pada persamaan diatas, $freq(t,D)$ adalah frekuensi kata (t) dalam suatu dokumen (D) dan $maxOthers(t,D)$ adalah total kata dalam dokumen (D), sehingga semakin sering sebuah kata muncul maka semakin tinggi nilai TF nya. Perhitungan selanjutnya adalah perhitungan IDF, untuk mengukur seberapa penting kata (t) dalam seluruh dokumen (D). Bertujuan untuk memberikan bobot yang lebih tinggi pada kata lebih jarang muncul dalam dokumen, semakin sedikit

kata muncul, maka semakin tinggi nilai IDF nya. Perhitungan IDF dapat dilakukan dengan menggunakan rumus pada Persamaan 2.2.

$$IDF = \log \frac{N}{df(t)} \quad (2.2)$$

Pada persamaan diatas, N adalah total jumlah dokumen dalam koleksi, dan $df(t)$ adalah total jumlah dokumen yang mengandung kata (t). Langkah selanjutnya adalah perhitungan bobot pada kata, dengan mempertimbangkan seberapa sering kata tersebut muncul dalam dokumen (nilai TF) dan seberapa pentingnya kata tersebut dalam seluruh dokumen (nilai IDF). Nilai TF dan IDF akan dilakukan untuk mendapatkan nilai bobotnya. Perhitungan bobot bertujuan untuk mengidentifikasi kata yang lebih relevan dan penting dalam suatu dokumen. Perhitungan bobot dengan menggunakan rumus pada Persamaan 2.3.

$$TF-IDF_{(t,D)} = TF(t, D) \times IDF \quad (2.3)$$

2.6 *Cosine Similarity*

Cosine Similarity adalah sebuah metrik yang digunakan dalam analisis data untuk mengukur tingkat kemiripan antara dua vektor dalam ruang berdimensi. Metrik ini membandingkan kesamaan antara dua vektor berdimensi, dengan menghitung *cosinus* dari sudut diantara keduanya (Jannach et al., 2010). Hasil pengukuran dapat digunakan untuk membandingkan teks dalam dua dokumen, yang mana dokumen direpresentasikan dalam bentuk vektor. Sehingga dapat diketahui nilai kemiripan antara dua dokumen dengan *cosine similarity*. Dalam Al-Quran Surah Az-Zumar (39:75):

وَتَرَى الْمَلَائِكَةَ حَافِينَ مِنْ حَوْلِ الْعَرْشِ يُسَبِّحُونَ بِحَمْدِ رَبِّهِمْ ۖ وَقُضِيَ بَيْنَهُم بِالْحَقِّ وَقِيلَ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

"Engkau (Nabi Muhammad) akan melihat malaikat melingkar di sekeliling 'Arasy. Mereka bertasbih sambil memuji Tuhannya. (Urusan) di antara mereka (seluruh makhluk) diputuskan dengan hak (adil). (Ketika itu) dikatakan, "Segala puji bagi Allah, Tuhan semesta alam." (QS. Az-Zumar: 39:75)

Ayat diatas menekankan bahwa pada Hari Kiamat, setiap hamba Allah akan menerima keputusan yang adil mengenai amal perbuatan mereka. Allah, yang Maha Adil, akan memberikan balasan yang tepat bagi setiap individu berdasarkan amal perbuatan mereka tanpa ada yang dirugikan atau diuntungkan secara tidak adil. Hal ini berkaitan dengan konsep penggunaan *cosine similarity* dalam sistem rekomendasi untuk mengukur kesamaan antara dua vektor. Rekomendasi yang dihasilkan didasarkan pada data yang akurat dan relevan, memastikan bahwa setiap pengguna menerima rekomendasi yang adil, sesuai dengan profil mereka. Sama seperti Allah memberikan keputusan yang adil berdasarkan amal perbuatan setiap individu, *cosine similarity* digunakan untuk memberikan rekomendasi yang adil dan objektif.

Untuk mendapat nilai *cosine similarity*, langkah pertama yang dilakukan perhitungan titik (*dot product*) yaitu dengan mengalikan nilai vektor *TF-IDF* setiap kata dalam dokumen(A) dengan nilai yang sesuai dengan dokumen(B), kemudian akan dijumlahkan semuanya. Langkah kedua adalah menghitung panjang vektor *TF-IDF* dokumen(A) dan dokumen(B), yaitu dengan menghitung akar kuadrat dari jumlah kuadrat semua nilai dalam vektor. Setelah didapat hasil perhitungan *dot product* dan perhitungan *magnitude* (panjang vektor), langkah

selanjutnya adalah membagi hasil keduanya. Perhitungan *cosine similarity* dapat dilakukan dengan menggunakan rumus pada Persamaan 2.4.

$$\text{Cosine Similarity}_{(A, B)} = \frac{A \times B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.4)$$

Pada rumus diatas, A dan B adalah dokumen-dokumen yang akan dibandingkan kemiripannya. A_i adalah kata ke- i pada dokumen A , dan B_i adalah kata ke- i yang terdapat pada dokumen B . Hasil dari perhitungan *cosine similarity* akan menghasilkan bobot dokumen nilai antara 0 sampai 1, ketika bobot mendekati 1 maka dokumen tersebut akan semakin mirip.

2.7 *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak untuk melakukan validasi keseluruhan fungsionalitas pada sistem apakah dapat bekerja dengan baik (Nidhra, 2012). Pengujian dengan *black box testing* ini dilakukan tanpa memeriksa atau mengevaluasi struktur dan logika dalam kode program. *Black Box Testing* dapat berguna untuk memindai dan menemukan beberapa hal yang tidak sesuai dengan prosedur, kesalahan struktur data, kesalahan basis data hingga kesalahan antarmuka.

2.8 *Precision*

Precision adalah salah satu metrik evaluasi dasar yang paling sering digunakan dalam penentuan efektifitas *information retrieval system* maupun *recommendation system*, dengan mengukur sejauh mana sistem dapat memberikan rekomendasi positif yang relevan. *Precision* akan membandingkan antara jumlah

hasil yang didapatkan secara relevan dengan jumlah hasil keseluruhan baik yang relevan atau tidak relevan (Manning et al., 2009). Perhitungan *precision* dapat dilakukan dengan menggunakan rumus pada Persamaan 2.5.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{relevant\ item\ retrieved}{retrieved\ item} \quad (2.5)$$

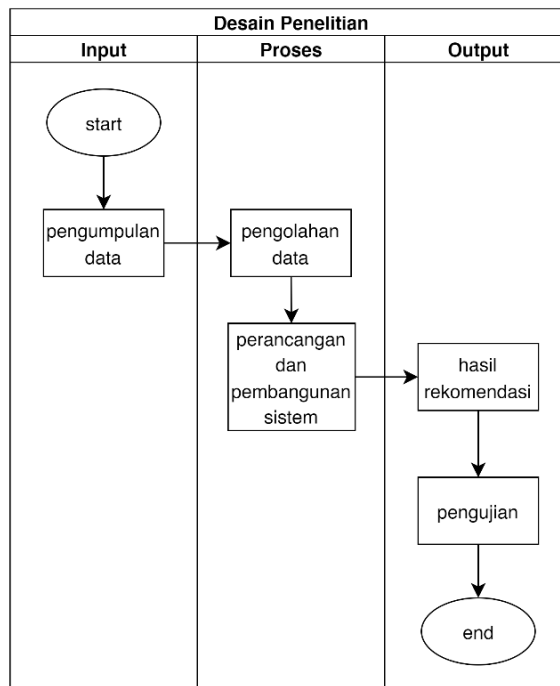
Istilah *positive* dan *negative* mengacu pada hasil rekomendasi yang diberikan oleh sistem. Sedangkan istilah *true* dan *false* mengacu pada tindakan pengguna, apakah hasil rekomendasi yang diberikan sesuai atau disukai oleh pengguna (Manning et al., 2009).

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Tahapan Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi pekerjaan yang dapat memberikan rekomendasi yang sesuai dengan latar belakang pencari kerja. Untuk mencapai tujuan tersebut, diperlukan sejumlah tahapan yang direncanakan agar penelitian berjalan dengan baik. Diantaranya terdiri dari tahap pengumpulan data, pengolahan data, perancangan dan pembangunan sistem, dan pengujian atau evaluasi. Tahapan-tahapan ini tertera pada Gambar 3.1.



Gambar 3.1 Desain penelitian

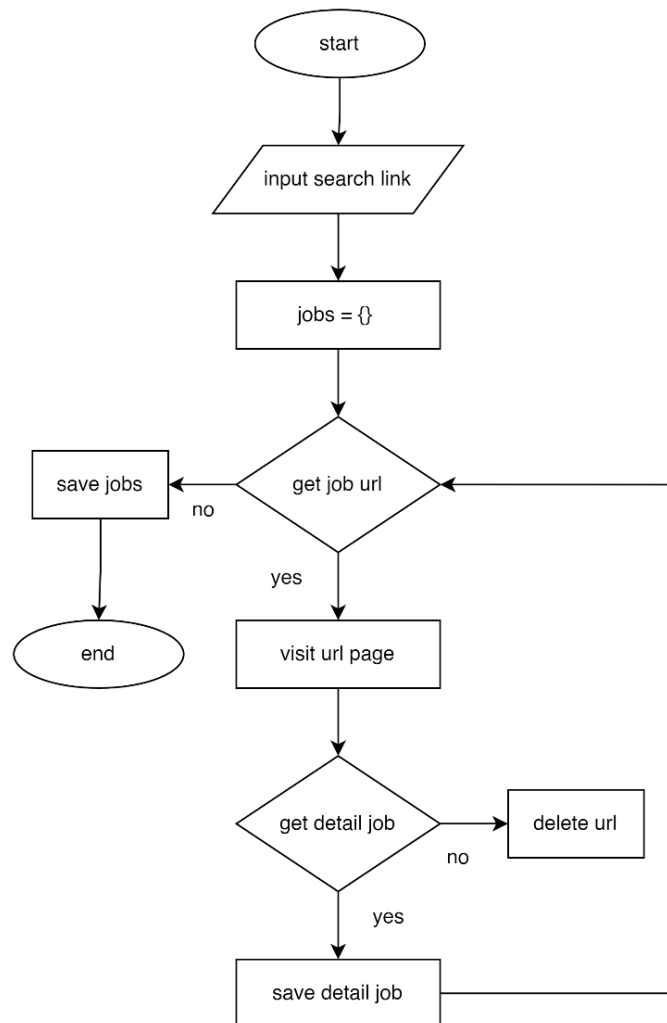
3.1.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari dua sumber utama, yaitu *LinkedIn* dan *JobStreet*. Masing-masing data memiliki kegunaan yang

berbeda sesuai dengan konteks dan tujuan penelitian. Tahapan pengumpulan data dilakukan dengan metode *web scraping*. Proses *scraping* data ini menggunakan 2 alat berbeda yang akan dijelaskan pada proses selanjutnya.

1. Data Lowongan Pekerjaan

Pengambilan data dilakukan dengan memanfaatkan *chrome extention Web Scraper* sebagai *tools* untuk melakukan *scraping* pada situs *JobStreet*. Langkah-langkah *scraping* data akan dilakukan berdasarkan *flowchart* pada Gambar 3.2.



Gambar 3.2 Flowchart *scraping* lowongan pekerjaan

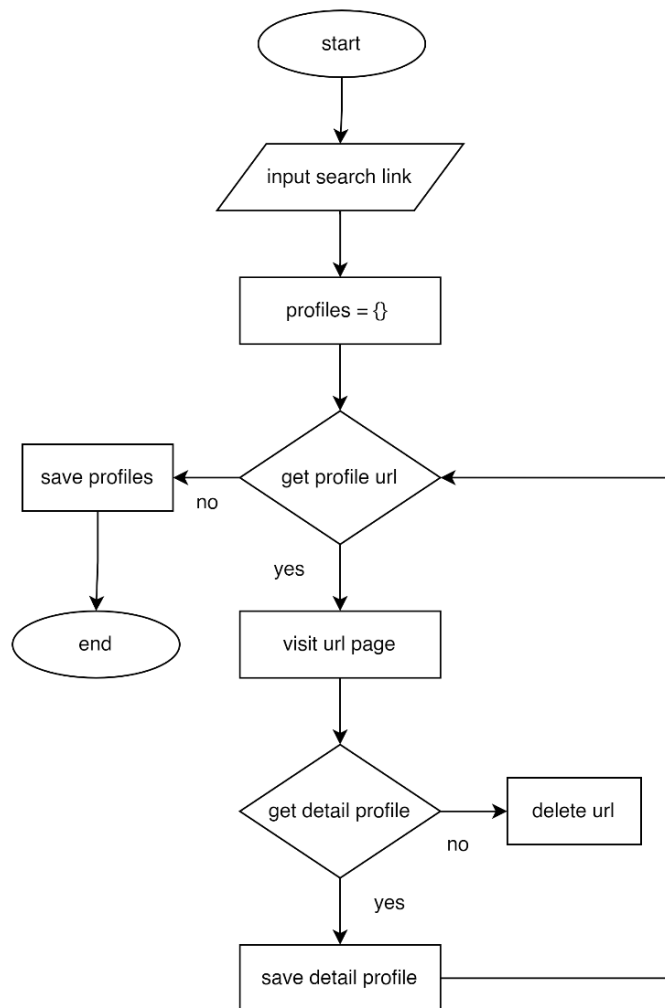
Proses *scraping* dengan ekstensi Web Scraper untuk mengambil data dari situs *JobStreet* dimulai dengan membuat *Sitemap* baru dan memasukkan *Start URL* dari halaman pencarian *JobStreet* yang telah difilter untuk bidang pekerjaan teknologi informasi, khususnya *developer* atau *programmer*. Setelah itu, ekstensi akan mengekstraksi URL dari setiap lowongan pekerjaan yang terdaftar dalam hasil pencarian. Jika ada URL yang ditemukan, ekstensi akan melanjutkan dengan mengunjungi halaman detail setiap lowongan pekerjaan. Pada halaman detail tersebut, informasi detail lowongan pekerjaan akan diekstraksi. Jika informasi berhasil diekstraksi, detail pekerjaan akan disimpan. Namun, jika tidak ada informasi yang ditemukan atau halaman tidak dapat diakses, URL tersebut akan dihapus dari daftar. Proses akan terus berlanjut, kembali ke langkah untuk mengambil URL dari pekerjaan berikutnya dalam hasil pencarian, hingga semua halaman hasil pencarian telah diproses dan informasi yang relevan telah diambil dan disimpan. Data yang berhasil dikumpulkan ekstensi selanjutnya diekspor dalam format XLSX. Detail informasi data lowongan pekerjaan yang dibutuhkan adalah title, company, location, job requirement dan job type. Pada Tabel 3.1 adalah sampel data yang berhasil diambil.

Tabel 3.1 Sampel data lowongan pekerjaan

<i>Title</i>	Staff IT
<i>Company</i>	Mini Market
<i>Location</i>	Malang, Jawa Timur
<i>Job Requirement</i>	Minimum age 25 years, GPA Min 2.75, D3/S1 Informatics, Understand Networking, Understand Windows Server, Mastering Visual FoxPro and other programming languages, Mastering SQLSERVER or MySQL Database, Minimum 2 years experience in the field, Domiciled in Malang Honest, independent, loyal and willing to work hard
<i>Job Type</i>	Kontrak

2. Data Pencari Kerja

Pengambilan data pencari kerja dilakukan dengan memanfaatkan Phantom Buster sebagai *tools* untuk melakukan *scraping* pada situs *LinkedIn*. Langkah-langkah *scraping* data akan dilakukan berdasarkan *flowchart* pada Gambar 3.3.



Gambar 3.3 Flowchart *scraping* pencari kerja

Proses *scraping* dengan Phantom Buster dimulai dengan *input* URL pencarian yang telah difilter untuk mencari pengguna dengan pekerjaan sebagai developer di situs *LinkedIn*. Selanjutnya, skrip akan mengekstraksi URL profil

dari setiap pengguna yang muncul dalam hasil pencarian, dan mengunjungi halaman detail setiap profil tersebut. Pada halaman detail, informasi yang relevan seperti nama, posisi pekerjaan, dan pendidikan akan diekstraksi. Jika informasi berhasil diambil, detail profil akan disimpan, namun jika tidak, URL profil akan dihapus dari daftar. Proses akan berlanjut dengan pengambilan URL profil dari pengguna berikutnya dalam hasil pencarian, dan berulang hingga semua halaman hasil pencarian telah diproses dan kemudian disimpan. Data yang berhasil dikumpulkan selanjutnya diekspor dalam format XLSX.

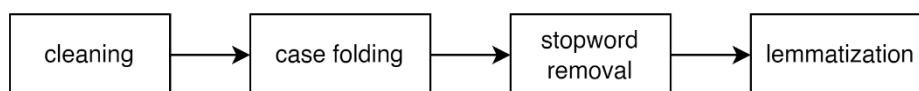
Kekurangan dalam menggunakan Phantom Buster adalah *tools* tidak dapat mengekstrak semua detail informasi yang dibutuhkan. Hal ini disebabkan oleh kebijakan dan aturan penggunaan *LinkedIn* yang membatasi pengambilan data penggunaannya untuk mengatasi penyalahgunaan data oleh pihak lain. Sehingga untuk informasi yang belum didapat dari *tools*, dilakukan pelengkapan informasi secara manual dengan mengunjungi URL profil yang telah didapat. Detail informasi data pencari kerja yang dibutuhkan adalah full name, location, about, current position, skills, education, experiences. Pada Tabel 3.2 adalah sampel data yang berhasil diambil.

Tabel 3.2 Sampel data profil pencari kerja

<i>Full Name</i>	Lutfi Widyanto
<i>Location</i>	Malang
<i>About</i>	Informatic Engineering graduate with 4+ years of experience. Very passionate in IT things especially in Programming.
<i>Current Position</i>	Web Developer
<i>Skills</i>	Laravel, Node.js, MySQL, RabbitMQ, PHP, Java, Web Development, CodeIgniter, jQuery, JavaScript, Delphi, Networking
<i>Education</i>	Bachelor's degree Information Technology
<i>Experiences</i>	Developer, Developer Staff, Information Technology Staff
<i>Licenses</i>	-

3.1.2 Preprocessing Text

Pada tahap pengolahan data dilakukan serangkaian langkah-langkah *preprocessing text* untuk mempersiapkan data mentah menjadi data yang siap diolah. Pada Gambar 3.4 adalah rincian mengenai tahapan *preprocessing*.



Gambar 3.4 Tahapan preprocessing

Berikut merupakan penjelasan tahapan *preprocessing*:

1. *Cleaning*

Langkah pertama dalam pengolahan data adalah pembersihan data. Pada tahap ini, data yang telah dikumpulkan akan dianalisis untuk mengatasi masalah seperti data yang hilang, duplikat, atau tidak relevan. Selanjutnya dilakukan pengecekan untuk memberi spasi antarkata yang berhuruf kapital, menghapus semua karakter non-huruf seperti angka dan simbol, menghapus kelebihan spasi di tengah kalimat, dan menghapus spasi diawal dan akhir kalimat. Proses ini memanfaatkan *library re* atau regex (*regular expression*).

Kode sumber 3.1 *Cleaning*

```

def cleaning(text)
    text = replace(text, '([a-z])([A-Z])', '\1 \2')
    text = replace(text, '^[a-zA-Z\s]', ' ')
    text = replace(text, '\s+', ' ')
    text = strip(text)
    Return text
  
```

2. *Case Folding*

Pengolahan data setelah *cleaning* adalah *case folding*, yaitu proses mengubah semua karakter teks menjadi huruf kecil. Untuk meningkatkan

konsistensi dan kemudahan dalam menganalisis teks. Proses ini memanfaatkan fungsi *lower()* dari *library* *pandas* dalam mengimplementasikannya.

Kode sumber 3.2 *Case folding*

```
def case_folding(text)
    text = lowercase(text)
    Return text
```

3. *Stopword Removal*

Stopwords removal adalah proses menghilangkan kata-kata umum yang tidak memberikan makna khusus dalam suatu kalimat. Tujuan proses ini adalah untuk meningkatkan relevansi, sehingga hanya fokus terhadap kata yang memiliki makna penting (Kannnan & Gurusamy, 2014). *Stopword* meliputi kata-kata preposisi (*in, on, at, with, by, etc.*), artikel (*the, a, an, etc.*), kata ganti (*I, you, he, she, it, they, etc.*), kata sambung (*and, or, but, because, etc.*), kata kerja bantu (*is, am, are, was, were, be, been, being, etc.*). Proses ini memanfaatkan *library* NLTK.

Kode sumber 3.3 *Stopword removal*

```
def stopwords_removal(text)
    words := split(text)
    filtered_words := []
    for word in words:
        if word.lower() not in stop_words:
            filtered_words.append(word)
    return ' '.join(filtered_words)
```

4. *Lemmatization*

Lematisasi adalah proses mengubah setiap kata menjadi bentuk dasarnya. Lematisasi berbeda dengan *stemming*. Lematisasi mengubah kata menjadi bentuk dasar dengan analisis lebih dalam, yaitu melalui perhitungan konteks dan makna kata, sedangkan *stemming* cenderung hanya memotong tambahan kata baik

awalan atau akhirnya, tanpa mementingkan arti atau makna kata yang sebenarnya (Weiss et al., 2005). Untuk menganalisis kata dalam bahasa Inggris menjadi bentuk dasarnya, lematisasi bekerja lebih baik daripada dengan *stemming*. Tahapannya dimulai dari memecah teks menjadi token dengan modul *word_tokenize*, kemudian dilematisasi untuk setiap tokennya. Lematisasi ini memanfaatkan modul *WordNetLemmatizer* dari library NLTK.

Kode sumber 3.4 *Lemmatization*

```
def lemmatization(text)
    words := word_tokenize(text)
    lemmatized_words := []
    for word in words:
        lemmatized_words.append(lemmatizer.lemmatize(word))
    return ' '.join(lemmatized_words)
```

Semua proses *preprocessing* tadi kemudian dijadikan menjadi fungsi *text_preprocessing*. Kelebihan penggunaan fungsi *text_preprocessing* yang menggabungkan semua tahapan *preprocessing* teks menjadi satu adalah efisiensi, konsistensi, kemudahan penggunaan dan fleksibilitas, sehingga dapat mengurangi kompleksitas kode.

Kode sumber 3.5 *Text preprocessing*

```
def text_preprocessing(text):
    text = cleaning(text)
    text = case_folding(text)
    text = stopwords_removal(text)
    text = lemmatization(text)
    return text
```

Selanjutnya *preprocessing* diimplementasikan pada kedua dataset. Untuk dataset lowongan pekerjaan yang akan memasuki *preprocessing* adalah kolom *title*, *company*, *location*, *job requirement*, *job type*. Sedangkan pada dataset

pencari kerja, kolom yang akan memasuki *preprocessing* adalah kolom *location*, *about*, *current position*, *skills*, *education*, *experiences*, *licenses*.

Kode sumber 3.6 Implementasi *preprocessing* pada dataset

```
def apply_text_preprocessing(dataset)
    for row in dataset:
        for column in columns_to_process:
            row[column] = text_preprocessing(row[column])
    return dataset
```

Kolom-kolom yang telah *preprocessing* kemudian akan disatukan menjadi satu kolom baru yaitu *tags* untuk masing-masing dataset. Kolom *tags* dalam dataset lowongan pekerjaan berisi gabungan kolom *title*, *company*, *location*, *job requirement* dan *job type*. Dan pada dataset pencari kerja, kolom *tags* berisi gabungan kolom *location*, *current position*, *about*, *skills*, *education*, *experiences* dan *licenses*. Hasil akhir *preprocessing* untuk sampel data lowongan pekerjaan (tabel 3.1) dan sampel data pencari kerja (tabel 3.2) adalah pada Tabel 3.3.

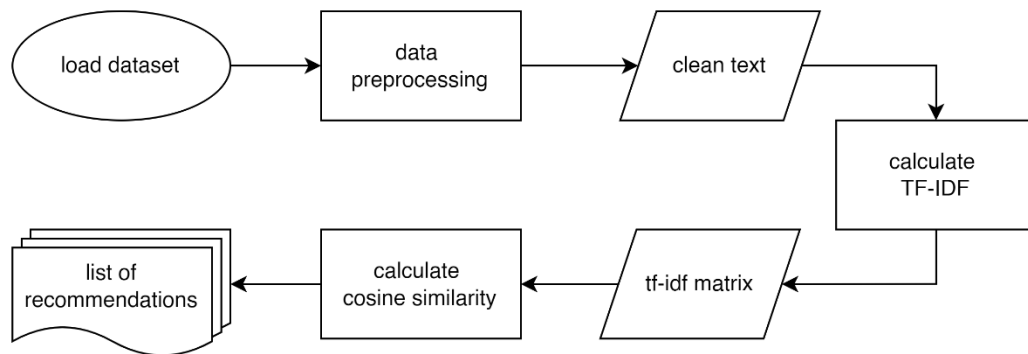
Tabel 3.3 Contoh data sesudah *preprocessing*

Sampel Data	Tags
Lowongan pekerjaan	staff mini market malang jawa timur qualification minimum age year gpa min informatics understand networking understand window server mastering visual fox pro programming language mastering sqlserver sql database minimum year experience field domiciled malang honest independent loyal willing work hard kontrak
Pencari kerja	malang informatic engineering graduate year experience passionate thing especially programming web developer laravel node j sql rabbit mq php java web development code igniter j query java script delphi networking bachelor degree information technology developer developer staff information technology staff

3.1.3 Content Based Filtering

Data yang telah melalui *preprocessing* kemudian siap untuk dianalisis lebih lanjut. Tahap selanjutnya adalah implementasi metode *content-based filtering*, dalam penelitian ini akan menggunakan algoritma TF-IDF untuk pembobotan kata dan *cosine similarity* untuk mengukur tingkat kemiripan antar data. Adapun

tahapan langkah-langkah implementasi *content-based filtering* adalah pada Gambar 3.5 dibawah ini.



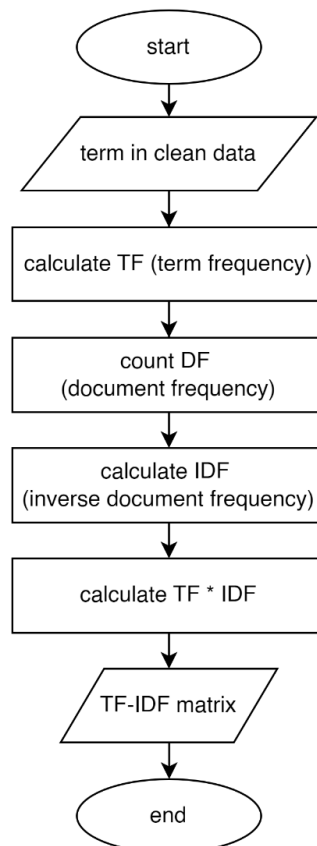
Gambar 3.5 Flowchart content-based filtering

Proses *content-based filtering* untuk sistem rekomendasi pekerjaan dimulai dengan memuat dataset yang berisi informasi terkait pekerjaan dan profil pencari kerja. Setelah dataset dimuat, tahap pertama adalah melakukan *preprocessing* data seperti yang telah dijelaskan pada bagian sebelumnya. Hasil dari *preprocessing* ini adalah data teks yang bersih dan siap untuk memasuki proses selanjutnya. Langkah berikutnya adalah menghitung nilai *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF adalah metode untuk menghitung bobot atau mengukur seberapa penting suatu kata dalam suatu dokumen terhadap semua dokumen dalam dataset. Hasil perhitungan TF-IDF adalah nilai TF-IDF berupa matriks numerik, dimana setiap baris mewakili dokumen (dalam penelitian ini, lowongan pekerjaan dan profil pencari kerja), dan setiap kolom mewakili kata-kata yang ada dalam dokumen. Setelah didapat nilai matriks TF-IDF dihitung, tahap selanjutnya adalah menghitung kesamaan antar dokumen dengan menggunakan metode *cosine similarity*. *Cosine similarity* akan mengukur seberapa mirip dua dokumen berdasarkan sudut antara vektor TF-IDF nya. Hasil

dari perhitungan ini adalah daftar rekomendasi yang menunjukkan dokumen pekerjaan mana yang paling sesuai dengan data profil pencari kerja yang diproses.

1. TF-IDF

Pada Gambar 3.6 adalah tahapan perhitungan TF-IDF secara lebih lanjut.



Gambar 3.6 Flowchart TF-IDF

Proses perhitungan TF-IDF dimulai dengan data teks yang telah dibersihkan melalui tahapan *preprocessing*. Data bersih ini akan dipecah menjadi token-token berupa kata (*term*) yang terkandung dalam dokumen. Kemudian setiap *terms* akan masuk ke dalam sistem sebagai input. Langkah pertama adalah menghitung *Term Frequency* (TF) yang berfungsi untuk mengetahui jumlah kemunculan kata (*t*) dalam sebuah dokumen (*D*), dengan cara membagi jumlah kemunculan sebuah kata

dalam dokumen dibagi dengan total jumlah kata dalam dokumen. Perhitungan TF ini dilakukan secara berulang untuk semua token atau *term* yang ada. Dengan menggunakan rumus pada Persamaan 3.1.

$$TF_{(t, D)} = \frac{freq(t, D)}{maxOthers(t, D)} \quad (3.1)$$

$$TF_{(age, D1)} = \frac{0}{38} = 0$$

$$TF_{(age, D2)} = \frac{1}{41} = 0,024$$

Hasil perhitungan nilai TF untuk masing-masing token atau *term* dari tiap-tiap dokumen sampel data yang telah melalui *preprocessing text* (tabel 3.5) dapat dilihat pada Tabel 3.4 di bawah ini.

Tabel 3.4 Perhitungan TF

No	Term	d1	d2	TF d1	TF D2
1	age	0	1	0	0,024
2	bachelor	1	0	0,026	0
3	code	1	0	0,026	0
4	database	0	1	0	0,024
5	degree	1	0	0,026	0
6	delphi	1	0	0,026	0
7	developer	3	0	0,079	0
8	development	1	0	0,026	0
9	domiciled	0	1	0	0,024
10	engineering	1	0	0,026	0
11	especially	1	0	0,026	0
12	experience	1	1	0,026	0,024
13	field	0	1	0	0,024
14	fox	0	1	0	0,024
15	gpa	0	1	0	0,024
16	graduate	1	0	0,026	0
17	hard	0	1	0	0,024
18	honest	0	1	0	0,024
19	igniter	1	0	0,026	0
20	independent	0	1	0	0,024
21	informatic	1	0	0,026	0
...
Total		38	41		

Selanjutnya, perhitungan *Document Frequency* (DF) yaitu untuk mengetahui jumlah kemunculan kata pada semua dokumen. Setelah DF, tahap

berikutnya adalah menghitung *Inverse Document Frequency* (IDF) yaitu untuk mengetahui kepentingan kata tersebut di seluruh dataset, dengan cara menghitung log total jumlah dokumen (N) dibagi dengan jumlah dokumen yang mengandung kata (df). Atau dengan menggunakan rumus pada Persamaan 3.2.

$$IDF_{(t)} = \log \frac{N}{df(t)} \quad (3.2)$$

$$IDF_{(age)} = \log \frac{2}{1} = 0,602$$

$$IDF_{(experience)} = \log \frac{2}{2} = 0,301$$

Diketahui nilai N atau jumlah dokumen adalah 2 yaitu dari tabel 3.5. Hasil perhitungan nilai DF untuk masing-masing token atau *term* dari semua dokumen sampel data yang telah melalui *preprocessing text* (tabel 3.5) dapat dilihat pada Tabel 3.5 dibawah ini.

Tabel 3.5 Perhitungan IDF

No	Term	DF	IDF
1	age	1	0,602
2	bachelor	1	0,602
3	code	1	0,602
4	database	1	0,602
5	degree	1	0,602
6	delphi	1	0,602
7	developer	1	0,602
8	development	1	0,602
9	domiciled	1	0,602
10	engineering	1	0,602
11	especially	1	0,602
12	experience	2	0,301
13	field	1	0,602
14	fox	1	0,602
15	gpa	1	0,602
16	graduate	1	0,602
17	hard	1	0,602
18	honest	1	0,602
19	igniter	1	0,602
20	independent	1	0,602
21	informatic	1	0,602
...

Setelah nilai TF dan IDF didapat, langkah berikutnya adalah perhitungan TF-IDF untuk mengetahui bobot kata dalam dokumen. Dengan cara mengalikan nilai masing-masing nilai TF pada tiap dokumen dengan nilai IDF atau dengan menggunakan rumus pada Persamaan 3.3.

$$TF-IDF_{(t,D)} = TF(t, D) \times IDF(t) \quad (3.3)$$

$$TF-IDF_{(age,D1)} = 0 \times 0,602 = 0 \quad TF-IDF_{(age,D2)} = 0,024 \times 0,602 = 0,014$$

Hasil perhitungan nilai TF-IDF untuk masing-masing token atau *term* dari tiap-tiap dokumen sampel data yang telah melalui *preprocessing text* (tabel 3.5) dapat dilihat pada Tabel 3.6 dibawah ini.

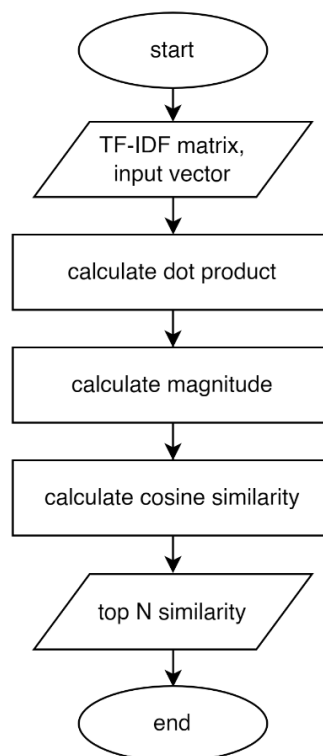
Tabel 3.6 Perhitungan TF-IDF

No	Term	TF d1 x IDF	TF d2 x IDF
1	age	0	0,014
2	bachelor	0,016	0
3	code	0,016	0
4	database	0	0,014
5	degree	0,016	0
6	delphi	0,016	0
7	developer	0,048	0
8	development	0,016	0
9	domiciled	0	0,014
10	engineering	0,016	0
11	especially	0,016	0
12	experience	0,008	0,007
13	field	0	0,014
14	fox	0	0,014
15	gpa	0	0,014
16	graduate	0,016	0
17	hard	0	0,014
18	honest	0	0,014
19	igniter	0,016	0
20	independent	0	0,014
21	informatic	0,016	0
...

Nilai TF-IDF yang didapat dari serangkaian tahapan diatas adalah berupa matriks numerik yang setiap kolomnya mewakili dokumen dan barisnya mewakili token atau *term* yang terkandung dalam sekumpulan dokumen.

2. *Cosine Similarity*

Tahapan perhitungan *cosine similarity* lebih lanjut adalah pada Gambar 3.7.



Gambar 3.7 Flowchart cosine similarity

Proses ini dimulai dengan menggunakan matriks TF-IDF (sekumpulan dokumen lowongan pekerjaan) dan vektor input (1 dokumen pencari kerja) sebagai input awal. Langkah pertama adalah menghitung *dot product* antara vektor input dan setiap vektor dokumen dalam matriks TF-IDF. *Dot product* ini dihitung dengan mengalikan nilai-nilai TF-IDF setiap *term* pada dua vektor atau

dua dokumen (dokumen A dan B) kemudian dari hasil perkalian tersebut akan dijumlahkan hasilnya, atau menggunakan rumus pada Persamaan 3.4 dibawah ini.

$$A \times B = \sum_{i=1}^n (TF - IDF_i^A) (TF - IDF_i^B) \quad (3.4)$$

$$d1 \times d2 = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0,001054 + \dots = 0,010707$$

Hasil dari perhitungan *dot product* adalah pada Tabel 3.7 dibawah ini.

Tabel 3.7 Perhitungan dot product

No	Term	Dot Product
1	age	0
2	bachelor	0
3	code	0
4	database	0
5	degree	0
6	delphi	0
7	developer	0
8	development	0
9	domiciled	0
10	engineering	0
11	especially	0
12	experience	0,001054
13	field	0
14	fox	0
15	gpa	0
16	graduate	0
17	hard	0
18	honest	0
19	igniter	0
20	independent	0
21	informatic	0
...
Total		0,010707

Selanjutnya perhitungan *magnitude* atau panjang vektor dari dokumen, yaitu dengan cara menghitung nilai kuadrat TF-IDF untuk masing-masing *term*, lalu dijumlahkan semua nilai kuadratnya dan terakhir dari total nilai kuadrat dihitung nilai akar kuadratnya, atau menggunakan rumus pada Persamaan 3.5 dibawah ini.

$$\|A\| \|B\| = \sqrt{\sum_{i=1}^n (TF - IDF_i^A)^2} \sqrt{\sum_{i=1}^n (TF - IDF_i^B)^2} \quad (3.5)$$

$$\begin{aligned}
\|d1\| &= \sqrt{0^2 + 0,016^2 + 0,016^2 + 0^2 + 0,016^2 + 0,016^2 + 0,048^2 + \dots +} \\
&= \sqrt{0 + 0,001764 + 0,001764 + 0 + 0,001764 + 0,001764 + 0,016129 + \dots +} \\
&= \sqrt{0,090242} \\
&= 0,300403063
\end{aligned}$$

$$\begin{aligned}
\|d2\| &= \sqrt{0,014^2 + 0^2 + 0^2 + 0,014^2 + 0^2 + 0^2 + 0^2 + \dots +} \\
&= \sqrt{0,001444 + 0 + 0 + 0,001444 + 0 + 0 + 0 + \dots +} \\
&= \sqrt{0,068793} \\
&= 0,262284197
\end{aligned}$$

Hasil perhitungan *magnitude* adalah pada Tabel 3.8 dibawah ini.

Tabel 3.8 Perhitungan magnitudo

No	Term	Magnitude d1	Magnitude d2
1	age	0	0,001444
2	bachelor	0,001764	0
3	code	0,001764	0
4	database	0	0,001444
5	degree	0,001764	0
6	delphi	0,001764	0
7	developer	0,016129	0
8	development	0,001764	0
9	domiciled	0	0,001444
10	engineering	0,001764	0
11	especially	0,001764	0
12	experience	0,001156	0,000961
13	field	0	0,001444
14	fox	0	0,001444
15	gpa	0	0,001444
16	graduate	0,001764	0
17	hard	0	0,001444
18	honest	0	0,001444
19	igniter	0,001764	0
20	independent	0	0,001444
21	informatic	0,001764	0
...
Total		0,090242	0,068793
Akar		0,300403063	0,262284197

Setelah didapat kedua nilai *dot product* dan *magnitude*, langkah berikutnya adalah perhitungan *cosine similarity* untuk mengetahui jarak kemiripan antara dokumen yang dibandingkan. Dengan cara membagi nilai *dot product* dengan

perkalian *magnitude* antar dokumen, atau dengan menggunakan rumus pada Persamaan 3.6.

$$\text{Cosine Similarity}_{(A, B)} = \frac{A \times B}{\|A\| \|B\|} \quad (3.6)$$

$$\text{Cosine Similarity}_{(d1, d2)} = \frac{0,010707}{(0,300403063)(0,262284197)} = 0,13589$$

Hasil perhitungan ini adalah nilai *cosine similarity* yang menunjukkan derajat kemiripan antara vektor input dan vektor dokumen. Dalam kasus percobaan perhitungan ini, didapat 1 nilai *cosine similarity* karena jumlah data yang dibandingkan hanya 2 baris data dari Tabel 3.4, yaitu 1 data lowongan pekerjaan (D1) dan 1 data pencari kerja (D2). Sehingga hanya menghasilkan 1 nilai *cosine similarity* seperti yang ditunjukkan pada Tabel 3.9 dibawah ini, yang mana hasil ini menggambarkan nilai kemiripan antara kedua data tersebut.

Tabel 3.9 Perhitungan *cosine similarity*

<i>Cosine Similarity</i>	D1
D2	0,13589

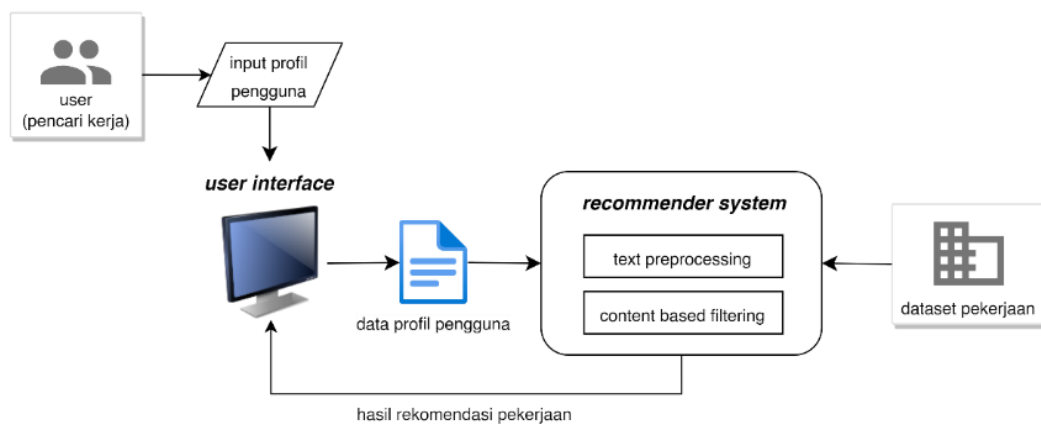
Banyak nilai *cosine similarity* yang dihasilkan bergantung pada jumlah data yang diproses. Dalam pengembangan sistem dan implementasi metode nantinya, data yang digunakan adalah semua data dalam dataset lowongan pekerjaan yang dibandingkan dengan 1 data pencari kerja yang dipilih sebagai *input*. Sehingga menghasilkan nilai kemiripan antara data pencari kerja dengan semua data lowongan pekerjaan, yang selanjutnya akan diambil 10 data lowongan pekerjaan dengan nilai kemiripan tertinggi.

3.1.4 Rencana Pengujian

Rencana pengujian dalam penelitian ini akan menggunakan 2 tahap yaitu pengujian fungsionalitas dengan *blackbox testing* dan pengujian hasil rekomendasi dengan metrik evaluasi presisi. Skenario pengujian hasil rekomendasi akan memperhitungkan variabel posisi. Pengujian akan dilakukan pada beberapa sampel pengguna pada variabel yang sama dari kedua dataset, yaitu variabel *current position* pada data pencari kerja dengan variabel *title* pada data lowongan pekerjaan. Untuk pemilihan skenario yang digunakan diambil dari jumlah posisi pekerjaan terbanyak pada kedua dataset. Dari 10 hasil rekomendasi, akan dilakukan analisis pada setiap pekerjaan yang direkomendasikan untuk mendapat nilai relevansi. Dari nilai relevansi akan dapat dihitung presisinya. Nilai presisi dari setiap skenario pengujian kemudian dihitung rata-ratanya, sehingga mendapat nilai presisi keseluruhan dari hasil sistem rekomendasi.

3.2 Desain Sistem

Desain sistem rekomendasi yang akan dikembangkan dalam penelitian ini adalah pada Gambar 3.8 dibawah ini.



Gambar 3.8 Desain sistem

Pada penelitian ini akan mengembangkan sistem berbasis *web application* untuk memberikan rekomendasi lowongan pekerjaan yang relevan atau memiliki kemiripan dengan data profil pengguna. Untuk memudahkan proses pengembangan sistem beserta integrasinya dengan mesin rekomendasi, tampilan antarmuka sistem ini dibangun menggunakan *framework* Streamlit dari Python. Untuk mesin rekomendasinya dikembangkan juga dengan bahasa pemrograman Python, dengan bantuan beberapa *library* untuk mendukung pemrosesan teks seperti *Natural Language Toolkit* (NLTK) untuk data *preprocessing*, *Scikit-learn* (sklearn) untuk perhitungan TF-IDF dan *cosine similarity*, serta *pandas* untuk manipulasi data.

Untuk alur berjalannya sistem, pengguna dapat memilih untuk menggunakan data dari dataset pencari kerja atau dengan menginputkan data profilnya secara manual dengan mengisikan *form* melalui tampilan antarmuka. Dataset pencari kerja tersebut didapat melalui proses *web scraping* dari situs LinkedIn (linkedin.com), dengan memanfaatkan tools Phantom Buster. Kemudian data inputan tersebut akan memasuki mesin rekomendasi yang diawali dengan proses *text preprocessing* yang mencakup langkah-langkah seperti *cleaning*, *case folding*, *stopword removal*, dan *lemmatization* agar data profil yang didapat memiliki kualitas yang baik. Untuk data lowongan pekerjaan juga didapatkan melalui proses *web scraping* pada salah satu *website* penyedia lowongan yaitu *JobStreet* (jobsreet.co.id) dengan memanfaatkan *chrome extention* bernama *Web Scraper*. Dataset lowongan pekerjaan juga akan melalui proses *text preprocessing*

agar data yang didapatkan memiliki kualitas yang baik dan untuk mempercepat memprosesan teks.

Kedua data tersebut kemudian akan memasuki proses inti dari sistem rekomendasi ini yaitu *content-based filtering*, yang berfokus pada perhitungan kesamaan konten antara profil pencari kerja dan dataset lowongan pekerjaan. Fitur-fitur yang didapat dari konten profil dan pekerjaan masing-masing akan dihitung bobot tiap kata nya terlebih dahulu dengan menggunakan TF-IDF. Kemudian perhitungan kesamaan akan dilakukan menggunakan metode *cosine similarity*. Hasil dari perhitungan kesamaan tersebut akan ditampilkan melalui tampilan antarmuka, data lowongan pekerjaan yang memiliki tingkat kemiripan yang tinggi dengan data profil pengguna akan ditampilkan kepada pengguna sebagai lowongan pekerjaan yang direkomendasikan oleh sistem.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil dari penelitian ini adalah mengembangkan sistem rekomendasi pekerjaan yang mengimplementasikan metode *content-based filtering*. Sistem berhasil dikembangkan dengan menggunakan bahasa pemrograman Python untuk mesin rekomendasi dan *framework* Steamlit untuk tampilan antarmuka. Jumlah data yang digunakan dalam penelitian ini adalah sebanyak 437 data lowongan pekerjaan yang didapat dari proses *web scraping* pada situs *JobStreet*, dapat dilihat pada Gambar 4.1 dibawah ini.

	title	company	location	job-type	links-href	job-req
0	.NET Developer	PT Lawencon Internasional	Jakarta Raya	Full time	https://www.jobstreet.co.id/id/job/75549445?ty...	Responsibilities: Develops code and creates cu...
1	.Net Developer	PT Glim Sinergi Kreatifa	Jakarta Timur, Jakarta Raya	Full time	https://www.jobstreet.co.id/id/job/75467237?ty...	Join us and be part of a company where your co...
2	.Net Developer	PT Lippo General Insurance Tbk	Tangerang, Banten	Full time	https://www.jobstreet.co.id/id/job/75401411?ty...	Responsibilities: Create program code, debuggi...
3	.NET Developer	PT Kamoro Maxima Integra	Jakarta Selatan, Jakarta Raya	Full time	https://www.jobstreet.co.id/id/job/75339477?ty...	Requirements: Minimum Bachelor Degree with min...
4	.NET Developer - IT Solutions (A-105243)	RGF HR Agent Indonesia	Jakarta Selatan, Jakarta Raya	Full time	https://www.jobstreet.co.id/id/job/74920482?ty...	About The Company: The working venue is in Sou...
...
432	Web Programmer (Pasaman, Sumatera Barat)	PT Jasamedika Saranatama	Sumatera Barat	Full time	https://www.jobstreet.co.id/id/job/75252541?ty...	Qualifications: D3 or S1 education (Informatic...
433	Web Programmer (Trenggalek)	PT Jasamedika Saranatama	Jawa Timur	Full time	https://www.jobstreet.co.id/id/job/75230431?ty...	Job Description Qualifications: D3 or Bachelor...
434	Web Shopify Developer and e-Commerce Specialis...	MIRA South Sea pearls	Jakarta Raya	Paruh waktu	https://www.jobstreet.co.id/id/job/75501863?ty...	Shopify Developer Job Description Template/Bri...
435	Website Developer	PT In Fashion	North Kuta, Bali	Full time	https://www.jobstreet.co.id/id/job/75502268?ty...	Responsibilities :Develop and maintain Shopify...
436	Website Developer	Company Confidential	Denpasar, Bali	Full time	https://www.jobstreet.co.id/id/job/74501479?ty...	Company Description/n/n Company that provides ...

437 rows x 6 columns

Gambar 4.1 Dataset lowongan pekerjaan

Data pencari kerja juga digunakan pada penelitian ini, sebanyak 100 data pencari kerja yang didapat dari hasil proses *web scraping* pada situs *LinkedIn*, dapat dilihat pada Gambar 4.2 dibawah ini.

	fullName	location	currentPosition	about	skills	education	experiences	licenses	linkedinProfileUri
0	Lutfi Widyanto	Malang	Web Developer	Informatic Engineering graduate with 4+ years ...	Laravel, Node.js, MySQL, RabbitMQ, PHP, Java, ...	Bachelor's degree Information Technology	Developer, Developer Staff, Information Techno...	-	https://linkedin.com/in/lutfi-widyanto
1	Sulthan Rafif	Malang	Software Developer	I am a Software Developer who is able to use J...	TensorFlow, PyTorch, Pandas, Linux, Jupyter, H...	Bachelor's degree Information Technology, Mast...	Back End Developer, Graduation Exam Assessor, ...	IT Support Google Professional Certificate, Ce...	https://linkedin.com/in/sulthan-rafif-b66662193
2	Thoriq Achmad Ramadhan	Malang	Web Developer	I am a graduate majoring in informatics engine...	Full-Stack Development, Back-End Web Developme...	Bachelor's degree Computer Science	Fullstack Developer, Interactive Systems Progr...	Go Certificate Course, Learn to Create Web App...	https://linkedin.com/in/thoriqar
3	Roffy Ansyakur Assya	Malang	Mobile Developer	Hello! I'm Roffy, a passionate Mobile develope...	Scrum Mobile Applications Android Android Deve...	Bachelor's degree, Informatics Engineering	Mobile Developer Flutter Developer Mobile Appl...	Certified Junior Mobile Programmer Training En...	https://linkedin.com/in/roffy-ansyakur
4	Muhammad Irfan Jazuli	Surabaya	Backend Developer	Helpful, talented, and dedicated are the princ...	Node.js Express.js Laravel MySQL Postman API C...	Bachelor's degree, Information Systems	Back-end Developer Cloud Computing Cohort Bang...	Application Development with Cloud Run Deploy ...	https://linkedin.com/in/mirfanjazuli

Gambar 4.2 Dataset pencari kerja

Data yang berhasil dikumpulkan kemudian memasuki proses terjemah untuk menyalarkan semua data menjadi bahasa inggris, agar lebih konsisten ketika memasuki proses *text preprocessing*. Proses penerjemahan dilakukan dengan memanfaatkan fungsi *google translate* pada *google spreadsheet*, yang diimplementasikan pada kolom-kolom dalam dataset. Selanjutnya dilakukan *text preprocessing* sesuai dengan tahapan yang telah dijelaskan pada bab 3. Hasil dari proses *text preprocessing* adalah data yang bersih dan siap diolah. Dataset lowongan pekerjaan yang sudah melalui *preprocessing text* dapat dilihat pada Gambar 4.3 dibawah ini.

	title	tags
0	net developer	net developer pt lawencon internasional jakart...
1	net developer	net developer pt glm sinergi kreatifa jakarta ...
2	net developer	net developer pt lippo general insurance tbk t...
3	net developer	net developer pt kamoro maximum integra jakart...
4	net developer solution	net developer solution rgf hr agent indonesia ...

Gambar 4.3 Dataset lowongan pekerjaan setelah penggabungan kolom

Terjemah dataset dan *preprocessing text* juga diterapkan dalam dataset pencari kerja, sehingga hasilnya dapat dilihat pada Gambar 4.4 dibawah ini.

	fullName	tags
0	lutfi widyanto	malang informatic engineering graduate year ex...
1	sulthan rafif	malang software developer able use java java s...
2	thoriq achmad ramadhan	malang graduate majoring informatics engineeri...
3	roffy ansyakur assya	malang hello roffy passionate mobile developer...
4	muhammad irfan jazuli	surabaya helpful talented dedicated principle ...

Gambar 4.4 Dataset pencari kerja setelah setelah penggabungan kolom

Dari data yang dimiliki, selanjutnya dibangun tampilan antarmuka sistem rekomendasi yang mengimplementasikan metode *content-based filtering* dalam memberikan rekomendasi. Terdapat beberapa bagian yang ditampilkan pada bagian antarmuka yaitu halaman menampilkan dataset, perhitungan metode, form rekomendasi, perhitungan hasil dan hasil rekomendasi.

The screenshot shows a web application with a navigation bar at the top containing links for 'Dataset', 'Perhitungan Metode', 'Form Rekomendasi', 'Perhitungan Hasil', and 'Hasil Rekomendasi'. The main content area is titled 'Dataset' and contains two tables.

Dataset Lowongan Pekerjaan

	title	company	location
0	.NET Developer	PT Lawencon Internasional	Jakarta Raya
1	.Net Developer	PT Glim Sinergi Kreatifa	Jakarta Timur
2	.Net Developer	PT Lippo General Insurance Tbk	Tangerang, B
3	.NET Developer	PT Kamoro Maxima Integra	Jakarta Selat
4	.NET Developer - IT Solutions (A-105243)	RGF HR Agent Indonesia	Jakarta Selat
5	.Net Developer - Property Company	PT. SIGMA GLOBAL TEKNOLOGI (SIGMATECH)	Tangerang Di
6	.Net Developer (Software Developer)	PT. Sinergi Karir Cipta	Jakarta Raya
7	.Net Engineer	PT GARUDA DATA TEKNOLOGI	Surabaya, Ja
8	ABAPER	PT Forisa Nusapersada	Jakarta Utara
9	AI Engineer (Anime)	IZUMO	Jakarta Raya

Dataset Pencari Kerja

	fullName	location	currentPosition
0	Lutfi Widyanto	Malang	Web Developer
1	Sulthan Rafif	Malang	Software Developer
2	Thoriq Achmad Ramadhan	Malang	Web Developer
3	Roffy Ansyakur Assya	Malang	Mobile Developer
4	Muhammad Irfan Jazuli	Surabaya	Backend Developer
5	Akmal Baihaqi	Malang	Backend Developer
6	Toton Dwi Antoko	Malang	Frontend Developer at Dipa Inhouse
7	Farras Hafish Zidane	Surabaya	Internal Developer At Telkom Indonesia ACE Certifie
8	Krisna Putra	Malang	ReactJS Developer
9	Alvan Maulana	Malang	Web developer

Gambar 4.5 Halaman dataset

Pada Gambar 4.5 adalah tampilan halaman dataset ditampilkan seluruh data yang digunakan dalam sistem, terdiri dari dataset lowongan pekerjaan dan dataset pencari kerja. Tujuan ditampilkan data lowongan pekerjaan adalah agar pengguna dapat membaca semua informasi terkait pekerjaan yang tersedia, sedangkan untuk data pencari kerja digunakan sebagai acuan atau contoh bagi pengguna dalam mengisi *form* rekomendasi secara manual pada halaman berikutnya.

The screenshot shows a web application interface with a navigation bar at the top containing links for 'Dataset', 'Perhitungan Metode', 'Form Rekomendasi', 'Perhitungan Hasil', and 'Hasil Rekomendasi'. The main content area is titled 'Perhitungan Metode' and displays 'Dataset Lowongan Pekerjaan' with a feature count of 5051. Below this, there are two tables:

Nilai TF-IDF

↑	abacus	abadi	abap	abaper	ability	able	abnormal	abreast	abroad	absolute	abu
0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0.072	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	
8	0	0	0.2361	0.2058	0.0351	0.0346	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	

Nilai Cosine Similarity

id	0	1	2	3	4	5	6	7	8	9	10
0	1	0.2184	0.1969	0.1796	0.2656	0.3029	0.3097	0.2526	0.1566	0.0392	0.0696
1	0.2184	1	0.1537	0.1979	0.2105	0.2469	0.3446	0.2251	0.1064	0.0401	0.0446
2	0.1969	0.1537	1	0.1516	0.1678	0.2761	0.2505	0.1601	0.1027	0.0079	0.007
3	0.1796	0.1979	0.1516	1	0.2322	0.2047	0.1858	0.2193	0.071	0.0553	0.043
4	0.2656	0.2105	0.1678	0.2322	1	0.2489	0.2778	0.2314	0.1671	0.0338	0.0351
5	0.3029	0.2469	0.2761	0.2047	0.2489	1	0.3432	0.2915	0.0843	0.0181	0.0153
6	0.3097	0.3446	0.2505	0.1858	0.2778	0.3432	1	0.248	0.0443	0.0158	0.0107
7	0.2526	0.2251	0.1601	0.2193	0.2314	0.2915	0.248	1	0.1364	0.0449	0.0564
8	0.1566	0.1064	0.1027	0.071	0.1671	0.0843	0.0443	0.1364	1	0.0451	0.0367
9	0.0392	0.0401	0.0079	0.0553	0.0338	0.0181	0.0158	0.0449	0.0451	1	0.211

Gambar 4.6 Halaman perhitungan metode di dataset lowongan pekerjaan

Pada Gambar 4.6 adalah tampilan halaman perhitungan metode yang menampilkan detail proses implementasi metode *content-based filtering*, meliputi perhitungan TF-IDF dan *cosine similarity* yang diterapkan pada dataset lowongan pekerjaan. Adapun data yang ditampilkan pada halaman perhitungan TF-IDF adalah jumlah total fitur dan nilai TF-IDF untuk setiap fitur. Total jumlah fitur atau kata kunci diperoleh dari proses tokenisasi yang dilakukan oleh TF-IDF berdasarkan dataset yang dipilih, yang pada dataset ini didapat sebanyak 5.051 fitur didalamnya. Nilai TF-IDF juga didapat untuk setiap fitur, sejumlah total fitur yaitu 5.051, ini menggambarkan nilai bobot kepentingan untuk masing-masing kata dalam dokumen. Pada gambar terlihat untuk kata kunci “*able*” ditemukan dalam data index ke-6 dengan nilai TF-IDF atau bobotnya 0.072. Selain itu ditemukan juga kata kunci “*abap*” dengan bobot 0.2361, “*abaper*” dengan bobot 0.2058, “*ability*” dengan bobot 0.0351, “*able*” dengan bobot 0.0346, pada data index ke-8. Dalam tabel nilai TF-IDF, kolom mewakili fitur atau kata kunci dan baris mewakili *index* data.

Berikutnya data yang ditampilkan dari perhitungan *cosine similarity* adalah nilai kemiripan antar data dalam dataset lowongan pekerjaan itu sendiri. Nilai *cosine similarity* yang didapat berupa matriks, karena hasil perkalian matriks TF-IDF dengan matriks. Pada gambar terlihat, pada baris pertama yaitu data index ke-0 memiliki nilai kemiripan sebesar 1 dengan data index ke-0, 0.2184 jika dibandingkan dengan data index-1, 0.1969 jika dibandingkan dengan data index ke-2, 0.1796 jika dibandingkan dengan data index ke-3, 0.2656 jika dibandingkan dengan data index ke-4, 0.3029 jika dibandingkan dengan data index ke-5, 0.3097

jika dibandingkan dengan data index ke-6, 0.2526 jika dibandingkan dengan index ke-7, 0.1566 jika dibandingkan dengan data index ke-8, 0.0392 jika dibandingkan dengan data index ke-9, 0.0696 jika dibandingkan dengan data index ke-10, dan seterusnya. Dalam tabel nilai *cosine similarity*, kolom dan baris masing-masing mewakili index data dalam dataset.

Dataset Pencari Kerja

Jumlah fitur (kata kunci): 1563

Nilai TF-IDF

	abap	ability	able	abroad	academic	academy	access	accomplishment	according
0	0	0	0	0	0	0	0	0	0
1	0	0	0.0583	0	0.0645	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0.0283	0	0	0
5	0	0	0	0	0	0	0	0.1696	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Nilai Cosine Similarity

id	0	1	2	3	4	5	6	7	8	9	10
0	1	0.1504	0.1788	0.1557	0.0561	0.2013	0.1205	0.0913	0.0598	0.1798	0.1588
1	0.1504	1	0.2238	0.2466	0.0555	0.1935	0.1322	0.1608	0.0898	0.1179	0.1892
2	0.1788	0.2238	1	0.2006	0.2752	0.1726	0.3287	0.3758	0.0699	0.1804	0.3796
3	0.1557	0.2466	0.2006	1	0.0912	0.15	0.1135	0.1611	0.0867	0.1641	0.385
4	0.0561	0.0555	0.2752	0.0912	1	0.0684	0.1407	0.5825	0.0201	0.0319	0.1059
5	0.2013	0.1935	0.1726	0.15	0.0684	1	0.1219	0.1901	0.0667	0.1451	0.1556
6	0.1205	0.1322	0.3287	0.1135	0.1407	0.1219	1	0.2802	0.1108	0.2152	0.3524
7	0.0913	0.1608	0.3758	0.1611	0.5825	0.1901	0.2802	1	0.0736	0.0925	0.2491
8	0.0598	0.0898	0.0699	0.0867	0.0201	0.0667	0.1108	0.0736	1	0.1032	0.0651
9	0.1798	0.1179	0.1804	0.1641	0.0319	0.1451	0.2152	0.0925	0.1032	1	0.147

Gambar 4.7 Halaman perhitungan metode di dataset pencari kerja

Pada Gambar 4.7 adalah lanjutan tampilan halaman perhitungan metode yang menampilkan detail proses implementasi metode *content-based filtering*,

meliputi perhitungan TF-IDF dan *cosine similarity* yang diterapkan pada dataset pencari kerja. Adapun data yang ditampilkan pada halaman perhitungan TF-IDF adalah jumlah total fitur dan nilai TF-IDF untuk setiap fitur. Total jumlah fitur atau kata kunci diperoleh dari proses tokenisasi yang dilakukan oleh TF-IDF berdasarkan dataset yang dipilih, yang pada dataset ini didapat sebanyak 1.563 fitur didalamnya. Nilai TF-IDF juga didapat untuk setiap fitur, sejumlah total fitur yaitu 1.563, ini menggambarkan nilai bobot kepentingan untuk masing-masing kata dalam dokumen. Pada gambar terlihat untuk kata kunci “*able*” mendapat bobot 0.0583, “*academic*” dengan bobot 0.0645, pada data index ke-1. Selain itu ditemukan juga kata kunci “*academy*” dengan bobot 0.0283 pada data index ke-4, “*accomplishment*” dengan bobot 0.1692 pada data index ke-5. Dalam tabel nilai TF-IDF, kolom mewakili fitur atau kata kunci dan baris mewakili *index* data.

Berikutnya data yang ditampilkan dari perhitungan *cosine similarity* adalah nilai kemiripan antar data dalam dataset lowongan pekerjaan itu sendiri. Nilai *cosine similarity* yang didapat berupa matriks, karena hasil perkalian matriks TF-IDF dengan matriks. Pada gambar terlihat, pada baris pertama yaitu data index ke-0 memiliki nilai kemiripan sebesar 1 dengan data index ke-0, 0.1504 jika dibandingkan dengan data index-1, 0.1788 jika dibandingkan dengan data index ke-2, 0.1557 jika dibandingkan dengan data index ke-3, 0.0561 jika dibandingkan dengan data index ke-4, 0.2013 jika dibandingkan dengan data index ke-5, 0.1205 jika dibandingkan dengan data index ke-6, 0.0913 jika dibandingkan dengan index ke-7, 0.0598 jika dibandingkan dengan data index ke-8, 0.1798 jika dibandingkan dengan data index ke-9, 0.1588 jika dibandingkan dengan data index ke-10, dan

seterusnya. Dalam tabel nilai *cosine similarity*, kolom dan baris masing-masing mewakili index data dalam dataset. Nilai hasil perhitungan pada tampilan diatas didapat dari implementasi kode dibawah ini.

Kode sumber 4.1 *TF-IDF* dan *cosine similarity*

```
def tfidf_cosine(nama, dataset, titles)
    tfidf_vectorizer := initialize TfIdfVectorizer
    tfidf_mtrx := tfidf_vectorizer.fit_transform(dataset['tags'])
    result_cosine := calculate cosine similarity between
    tfidf_mtrx and tfidf_mtrx
    feature_names := tfidf_vectorizer.get_feature_names_out()
    num_features := get length of feature_names
    display 'Dataset '+nama as subheader
    display 'Jumlah fitur (kata kunci): ' + num_features
    display 'Nilai TF-IDF'
    display DataFrame of tfidf_mtrx with columns as
    tfidf_vectorizer.get_feature_names_out()
    display 'Nilai Cosine Similarity' as subheader
    display DataFrame of result_cosine with index as titles and
    columns as titles
```

Kode sumber 4.1 adalah implementasi perhitungan TF-IDF dalam sistem ini, yang menerapkan modul TfIdfVectorizer dari *library* Scikit-Learn. Pertama-tama dilakukan inisialisasi TfIdfVectorizer dan disimpan dalam “*tfidf_vectorizer*”. Vectorizer ini berguna untuk melakukan tokenisasi dari teks menjadi token-token atau memecah teks per kata. Kemudian dilakukan transformasi teks dengan fungsi *fit_transform* dari kolom *tags* pada dataset menjadi matriks TF-IDF yang berisi bobot kata untuk setiap token atau kata dari hasil tokensasi dan disimpan dalam “*tfidf_mtrx*”. Selanjutnya memasuki perhitungan kemiripan antar data dengan *cosine similarity*, yang memperhitungkan kemiripan berdasarkan nilai pada *tfidf_mtrx* dan disimpan dalam “*result_cosine*”. Dalam perhitungan dataset ini, *tfidf* matriks dikalikan dengan *tf-idf* matriks yang sama sehingga didapat nilai kemiripan antar data. Untuk mendapatkan data semua fitur dalam sebuah dataset, digunakan fungsi *get_features_names_out* dan hasilnya disimpan dalam

“*features_name*”. Kemudian untuk mendapatkan nilai total jumlah fitur, digunakan fungsi *len* dan disimpan dalam “*num_features*”.

Gambar 4.8 Halaman *form* rekomendasi *input* dataset

Pada Gambar 4.8 adalah tampilan halaman *form* rekomendasi dengan menggunakan opsi *input* data dari dataset. Untuk mendapatkan rekomendasi pekerjaan dari input data ini, pengguna cukup memilih *index* data pada *field* ‘*select people*’, dimana *field* tersebut sudah terhubung dengan dataset pencari kerja. Kemudian sistem akan memproses atau *load data* dari *index* yang dipilih dan menampilkan *tags* apa saja yang terdapat dalam baris data tersebut. Isi dari *tags* adalah gabungan data *location*, *current position*, *about*, *skills*, *education*, *experiences* dan *licenses*. Setelah berhasil *load data* pada *field tags*, selanjutnya dapat diklik button “*Get Recommendation*” untuk mengirim data *input* menuju mesin rekomendasi dan diproses datanya. Ketika data berhasil diinputkan, maka akan muncul pesan *success* yang berisi “*Data selected from dataset*”. Selain input data dari dataset, terdapat opsi penggunaan data yang diinput secara manual melalui form yang disediakan pada gambar berikut ini.

The screenshot shows a web application interface for a manual input recommendation form. The navigation bar at the top includes links for 'Dataset', 'Perhitungan Metode', 'Form Rekomendasi' (highlighted), 'Perhitungan Hasil', and 'Hasil Rekomendasi'. The main heading is 'Form Rekomendasi Pekerja'. Below the heading, there are radio buttons for 'Choose data source:', with 'Manual Input' selected. The form fields are: 'Name' (Alya Fitria), 'Location' (Malang), 'Description' (a paragraph about technology and collaboration), 'Current Job' (Web Developer), 'Tech Skills' (HTML, CSS, Bootstrap, Laravel, Python), and 'Education' (Bachelor Degree). There is an 'Experiences' section with an 'Add Experience' button and one experience entry: 'Experience 1' with title 'Frontend Developer Intern' and description 'Developed and maintained responsive websites using HTML, CSS, and JavaScript.' At the bottom, there is a 'Get Recommendation' button and a green success message: 'Your Data Saved!'.

Gambar 4.9 Halaman *form* rekomendasi *input* manual

Pada Gambar 4.9 adalah tampilan halaman *form* rekomendasi dengan menggunakan opsi *input* data manual. Disediakan *form* yang berisi berbagai *field* untuk mendapatkan data pengguna seperti *name*, *location*, *description*, *current job*, *tech skills*, *education* dan *experiences* yang wajib diisi. Ketika semua *field*

telah terisi dan pengguna mengklik *button* “*Get Recommendation*” maka akan muncul pesan *success* yang berisi “*Your Data Saved!*”. Sedangkan jika ada salah satu atau lebih *field* yang belum diisi, dan diklik *button* “*Get Recommendation*” maka sistem akan menampilkan pesan error yang berisi “*All fields must be filled in!*”.

Setelah data *input* terkirim, selanjutnya nilai perhitungan hasil *input* akan ditampilkan pada halaman berikutnya. Pada gambar dibawah ini adalah tampilan dan nilai hasil perhitungan untuk data input dari dataset dengan *index* ke-0.

The screenshot shows a web interface for 'Perhitungan Hasil' (Result Calculation). It displays TF-IDF values for user input and recommended documents, Cosine Scores for all data, and Cosine Scores for the top 10 results.

Perhitungan Hasil

Nilai TF-IDF dari input pengguna:

	bachelor	code	degree	delphi	developer	development	engineering	especially	exper
0	0.0797	0.0835	0.0776	0.3032	0.2046	0.0656	0.0921	0.1763	0

Nilai TF-IDF dari 10 dokumen pekerjaan yang direkomendasikan:

	ability	able	according	administer	advanced	advantage	advisory	age	agile	aj
0	0	0	0	0	0	0	0	0.1241	0	
1	0	0.0775	0	0	0	0	0	0	0.106	
2	0	0	0	0	0	0	0	0.0997	0	
3	0	0.0699	0	0	0	0	0	0	0	
4	0	0.072	0	0	0.1387	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	
7	0.0627	0.0618	0	0	0	0	0	0	0.0846	
8	0	0	0.0652	0.1237	0	0.0677	0.1158	0	0	0
9	0	0	0	0	0	0	0	0	0	

Cosine Scores All Data

	0	1	2	3	4	5	6	7	8	9	10	:
0	0.0808	0.0496	0.0617	0.1078	0.0625	0.0561	0.0875	0.0714	0.0381	0.026	0.013	f

Cosine Scores Top 10

	386	52	217	431	426	118	281	109	218	193
similarity	0.2206	0.197	0.1928	0.1826	0.1786	0.1692	0.1655	0.1616	0.1614	0.1595

Gambar 4.10 Halaman perhitungan hasil

Pada Gambar 4.10 adalah tampilan halaman perhitungan hasil akan menampilkan detail perhitungan metode *content-based filtering* berdasarkan data yang telah diinputkan dari halaman *form* rekomendasi. Halaman ini hanya akan menampilkan hasil ketika halaman *form* rekomendasi telah diisi, jika belum maka akan muncul pesan *error* yang berisi “*Please fill the form*”. Nilai TF-IDF yang ditampilkan terdiri dari 2 macam data, yaitu nilai TF-IDF dari data yang diinputkan pengguna dan dari 10 data lowongan pekerjaan yang direkomendasikan. Nilai TF-IDF dari data *input* pengguna memiliki bobot untuk setiap kata kunci yang sama dengan kata kunci yang telah tersimpan atau kata kunci yang telah dibangun dari dataset lowongan pekerjaan. Nilai TF-IDF yang didapat ini bersifat dinamis terhadap jumlah dokumen atau dataset yang dibandingkan. Pada gambar terlihat, kata kunci “*bachelor*” mendapat bobot 0.0797, “*code*” mendapat bobot 0.0835, “*degree*” mendapat bobot 0.0776, dan seterusnya. Kemudian untuk nilai TF-IDF dari 10 data yang direkomendasikan, kolomnya berjumlah sebanyak fitur yang terdapat dalam 10 data yang direkomendasi, barisnya berjumlah 10 data yang sudah diurutkan dari data dengan kemiripan tertinggi. Pada gambar terlihat, kata kunci “*age*” mendapat bobot 0.1241 pada index rekomendasi ke-1, “*able*” mendapat bobot 0.0775 pada index rekomendasi ke-2, “*advanced*” mendapat bobot sebesar 0.1387 pada index rekomendasi ke-4, dan seterusnya.

Sedangkan untuk nilai *cosine similarity* yang ditampilkan adalah nilai kemiripan pada semua lowongan pekerjaan dalam dataset dan 10 nilai dengan *similarity* tertinggi. Nilai *cosine similarity* ini didapat dari perbandingan 1 data

input oleh pengguna yang dibandingkan dengan semua baris data dalam dataset lowongan pekerjaan. Pada gambar terlihat, nilai kemiripan antara data input dengan data lowongan pekerjaan index ke-0 adalah sebesar 0.080, dengan data index ke-1 sebesar 0.0496, dengan data index ke-2 sebesar 0.0617, dan seterusnya. Kemudian dari semua nilai kemiripan akan diurutkan dan diambil 10 nilai kemiripan tertinggi, sehingga didapat data input paling mirip dengan data index ke-386 sebesar 0.2206, dengan data index ke-52 sebesar 0.197, dengan data index ke-217 sebesar 0.1928, dan seterusnya. Nilai hasil perhitungan berdasarkan data *input* pada tampilan diatas didapat dari implementasi kode yang akan dijelaskan dibawah ini.

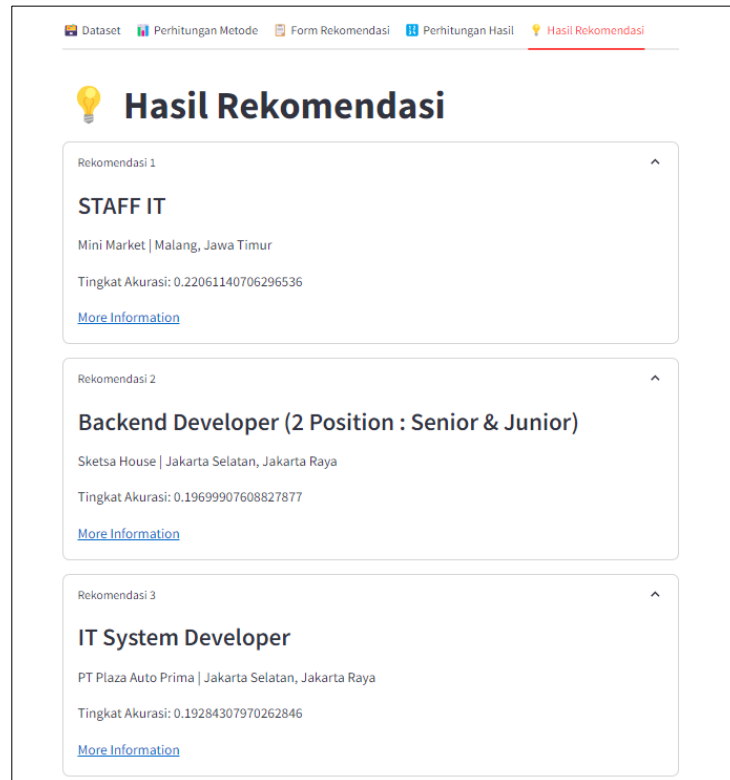
Kode sumber 4.2 *Get recommendation*

```
def get_recommendations(**kwargs):
    kwargs_str = {key: str(value) for key, value in
kwargs.items()}
    input_text = " ".join(kwargs_str.values())
    input_vector = tfidf.transform([input_text])
    cosine_scores = cosine_similarity(input_vector,
tfidf_matrix).flatten()
    sim_indices = cosine_scores.argsort()[::-11:-1]
    recommended_jobs = dataset_loker.iloc[sim_indices]
    recommended_jobs['similarity'] = cosine_scores[sim_indices]
    recommended_job_vectors = tfidf_matrix[sim_indices]
    return recommended_jobs, tfidf.get_feature_names_out(),
input_vector.toarray(), recommended_job_vectors.toarray(),
cosine_scores
```

Kode sumber 4.2 diatas berguna untuk memproses data input dari halaman *form* rekomendasi dengan mengimplementasikan *content-based filtering*, kode ini dijalankan setelah *button* “*Get Recommendation*” diklik. Pertama-tama fungsi *get_recommendations* menerima *input* berupa *keyword arguments* dan disimpan dalam *input_text*. Selanjutnya *input_text* akan memasuki proses transformasi teks dengan fungsi *transform* menjadi vektor TF-IDF yang memberi bobot kata untuk

setiap kata kunci dari *input_text* yang ada pada matriks TF-IDF (representasi dataset lowongan pekerjaan) yang tersimpan sebelumnya, dan disimpan dalam “*input_vector*”. Selanjutnya memasuki perhitungan kemiripan antar data dengan *cosine similarity*. Yang memperhitungkan kemiripan antara *input_vector* (representasi data input pencari kerja) dengan *tfidf_matrix* (representasi dataset lowongan pekerjaan), dan memanggil fungsi *flatten* untuk menyamakan dimensi data antara vektor input dan matriks tf-idf, kemudian nilainya disimpan dalam *cosine_scores*. Selanjutnya dilakukan perangkingan nilai *cosine similarity* dengan fungsi *argsort* untuk mengambil *index* data lowongan pekerjaan dengan 10 nilai kemiripan tertinggi, dan disimpan dalam *sim_indices*. Dari *index* yang didapat selanjutnya diambil data dari dataset lowongan pekerjaan dan disimpan dalam *recommended_jobs*. Kemudian nilai *similarity* dari 10 *index* tersebut disimpan dalam kolom baru bernama *similarity* yang ditambahkan dalam dataframe *recommended_jobs*. Pada gambar dibawah ini adalah tampilan dan hasil rekomendasi pekerjaan untuk data *input* dari dataset dengan *index* ke-0.

Pada Gambar 4.11 adalah halaman hasil rekomendasi, halaman ini akan menampilkan 10 hasil rekomendasi pekerjaan dengan nilai *similarity* yang tertinggi. Detail lowongan pekerjaan yang akan ditampilkan adalah *title* atau posisi pekerjaan, *company* atau perusahaan yang membuka lowongan, lokasi perusahaan, nilai *similarity* dan tautan *more information* yang dapat diakses menuju ke halaman *eksternal* situs penyedia lowongan pekerjaan.



Gambar 4.11 Halaman hasil rekomendasi

Pada Tabel 4.1 adalah semua hasil rekomendasi pekerjaan untuk user index ke-0 dari dataset pencari kerja.

Tabel 4.1 Hasil rekomendasi pencari kerja index ke-0

Job ID	Title	Similarity
386	Staff IT	0.2206
52	Backend Developer (2 position: senior & junior)	0.1970
217	IT System Developer	0.1928
431	Web Programmer Frontend	0.1826
426	Web Programmer	0.1786
118	Full Stack Engineer	0.1692
281	PHP Java Script Developer	0.1655
109	Full Stack Developer	0.1616
218	IT System Developer Staff	0.1614
193	PHP Developer	0.1595

Hasil rekomendasi 1 didapat rekomendasi pekerjaan dengan posisi Staff IT pada *index* ke-386 yang mendapat nilai *similarity* 0.2206. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada

dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel lokasi (dibuktikan dengan kesamaan pada kata “malang” sebesar 0.4204), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “networking” sebesar 0.1799, “sql” sebesar 0.0674), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “informatic” sebesar 0.1141, “year” sebesar 0.0979, “programming” sebesar 0.0665, “experience” sebesar 0.0419) dan berdasarkan pengalaman (dibuktikan dengan kesamaan pada kata “staff” sebesar 0.1454).

Hasil rekomendasi 2 didapat rekomendasi pekerjaan dengan posisi Backend Developer pada *index* ke-52 yang mendapat nilai *similarity* 0.1970. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “web” sebesar 0.0662 dan “developer” sebesar 0.0777), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “programming” sebesar 0.0936, “experience” sebesar 0.1027, “year” sebesar 0.0459), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “sql” sebesar 0.1265, “laravel” sebesar 0.1084, “PHP” sebesar 0.0872, “development” sebesar 0.0249), berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “information” sebesar 0.1454, “bachelor” sebesar 0.0303, “technology” sebesar 0.0297, “degree” sebesar 0.0295) dan berdasarkan pengalaman (dibuktikan dengan kesamaan pada kata “staff” sebesar 0.2731).

Hasil rekomendasi 3 didapat rekomendasi pekerjaan dengan posisi IT System Developer pada *index* ke-217 yang mendapat nilai *similarity* 0.1928. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*web*” sebesar 0.0566 dan “*developer*” sebesar 0.133), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*programming*” sebesar 0.0534, “*engineering*” sebesar 0.1198, “*experience*” sebesar 0.0337, “*year*” sebesar 0.0786), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*rabbit*” sebesar 0.135, “*mq*” sebesar 0.1311, “*sql*” sebesar 0.1083, “*laravel*” sebesar 0.0927, “*PHP*” sebesar 0.0746, “*script*” sebesar 0.0721, “*java*” sebesar 0.0565) dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*information*” sebesar 0.1216, “*bachelor*” sebesar 0.0518, “*degree*” sebesar 0.0505).

Hasil rekomendasi 4 didapat rekomendasi pekerjaan dengan posisi Web Programmer Frontend pada *index* ke-431 yang mendapat nilai *similarity* 0.1826. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*web*” sebesar 0.3585, “*developer*” sebesar 0.0468), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*experience*” sebesar 0.071, “*programming*” sebesar 0.0563, “*year*” sebesar 0.0415), berdasarkan *skills*

(dibuktikan dengan kesamaan pada kata “*query*” sebesar 0.1074, “*script*” sebesar 0.0761, “*java*” sebesar 0.0596, “*development*” sebesar 0.045) dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*information*” sebesar 0.1924, “*technology*” sebesar 0.1073, “*bachelor*” sebesar 0.0547, “*degree*” sebesar 0.0532).

Hasil rekomendasi 5 didapat rekomendasi pekerjaan dengan posisi Web Programmer pada *index* ke-426 yang mendapat nilai *similarity* 0.1786. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*web*” sebesar 0.1846), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*experience*” sebesar 0.0732, “*year*” sebesar 0.0427), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*query*” sebesar 0.1106, “*laravel*” sebesar 0.1008, “*php*” sebesar 0.0811, “*script*” sebesar 0.0784, “*java*” sebesar 0.0614, “*sql*” sebesar 0.0588, “*development*” sebesar 0.0463) dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*information*” sebesar 0.1982, “*technology*” sebesar 0.1658, “*bachelor*” sebesar 0.0563, “*degree*” sebesar 0.0548).

Hasil rekomendasi 6 didapat rekomendasi pekerjaan dengan posisi Full Stack Engineer pada *index* ke-118 yang mendapat nilai *similarity* 0.1692. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan

ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*web*” sebesar 0.0637, “*developer*” sebesar 0.0997), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*engineering*” sebesar 0.202, “*experience*” sebesar 0.1514, “*graduate*” sebesar 0.1121, “*informatics*” sebesar 0.1031, “*programming*” sebesar 0.06, “*year*” sebesar 0.0442), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*sql*” sebesar 0.1217, “*node*” sebesar 0.1048, “*development*” sebesar 0.0479) dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*technology*” sebesar 0.2288, “*information*” sebesar 0.0684).

Hasil rekomendasi 7 didapat rekomendasi pekerjaan dengan posisi PHP Java Script Developer pada *index* ke-281 yang mendapat nilai *similarity* 0.1655. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*developer*” sebesar 0.1009), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*graduate*” sebesar 0.1135, “*engineering*” sebesar 0.0681, “*year*” sebesar 0.0447, “*experience*” sebesar 0.0383), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*java*” sebesar 0.1927, “*php*” sebesar 0.1698, “*script*” sebesar 0.1641, “*sql*” sebesar 0.1232, “*laravel*” sebesar 0.1055, “*development*” sebesar 0.097), dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*degree*” sebesar 0.1148, “*bachelor*” sebesar 0.059).

Hasil rekomendasi 8 didapat rekomendasi pekerjaan dengan posisi Full Stack Developer pada *index* ke-109 yang mendapat nilai *similarity* 0.1616. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*developer*” sebesar 0.1241, “*web*” sebesar 0.1057), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*engineering*” sebesar 0.1118, “*informatics*” sebesar 0.0856, “*experience*” sebesar 0.0942, “*programming*” sebesar 0.0498, “*year*” sebesar 0.0367), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*development*” sebesar 0.1592, “*java*” sebesar 0.1054, “*sql*” sebesar 0.101, “*node*” sebesar 0.087, “*script*” sebesar 0.0673), dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*information*” sebesar 0.1702, “*technology*” sebesar 0.0475).

Hasil rekomendasi 9 didapat rekomendasi pekerjaan dengan posisi IT System Developer Staff pada *index* ke-218 yang mendapat nilai *similarity* 0.1614. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*developer*” sebesar 0.0278), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*experience*” sebesar 0.0633, “*year*” sebesar 0.0247), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*delphi*” sebesar

0.1237, “*development*” sebesar 0.107, “*java*” sebesar 0.1063, “*sql*” sebesar 0.0679, “*php*” sebesar 0.0468), berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*information*” sebesar 0.0763, “*technology*” sebesar 0.0638) dan berdasarkan pengalaman (dibuktikan dengan kesamaan pada kata “*staff*” sebesar 0.1466).

Hasil rekomendasi 10 didapat rekomendasi pekerjaan dengan posisi PHP Developer pada *index* ke-193 yang mendapat nilai *similarity* 0.1595. Jika ditinjau berdasarkan kata kunci yang ada di data *input* dan sama dengan kata yang ada dalam data rekomendasi pekerjaan ini, maka dapat dikatakan bahwa pekerjaan ini relevan dengan pengguna. Relevansi ditemukan berdasarkan kecocokan pada variabel posisi saat ini (dibuktikan dengan kesamaan pada kata “*developer*” sebesar 0.1225), berdasarkan deskripsi diri (dibuktikan dengan kesamaan pada kata “*graduate*” sebesar 0.1378, “*year*” sebesar 0.0979, “*experience*” sebesar 0.1214, “*engineering*” sebesar 0.0827, “*year*” sebesar 0.0543), berdasarkan *skills* (dibuktikan dengan kesamaan pada kata “*php*” sebesar 0.3094, “*laravel*” sebesar 0.1281) dan berdasarkan pendidikan (dibuktikan dengan kesamaan pada kata “*degree*” sebesar 0.2092, “*information*” sebesar 0.084, “*bachelor*” sebesar 0.0716, “*technology*” sebesar 0.0703).

4.2 Pengujian

Pengujian dalam penelitian ini dilakukan pada uji fungsionalitas pada fungsi sistem dan uji hasil rekomendasi untuk mendapat nilai presisi pada beberapa skenario pengujian.

4.2.1 *Black Box Testing*

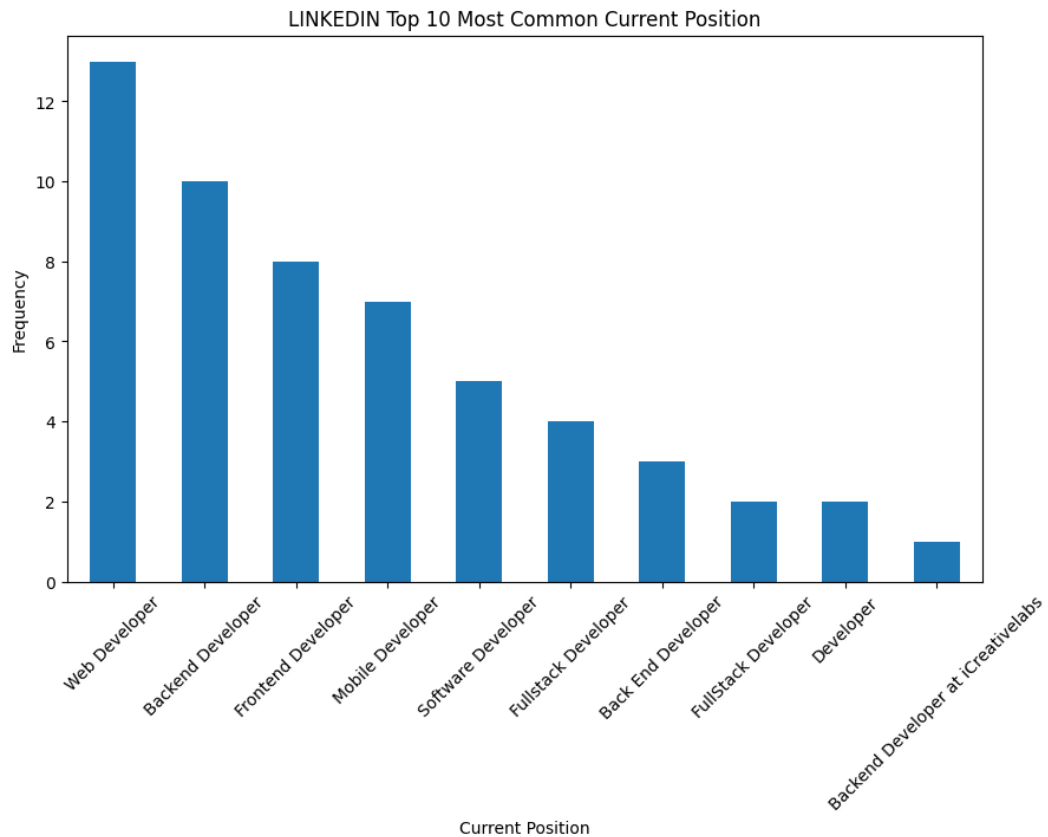
Pengujian pertama adalah uji fungsionalitas sistem untuk mengetahui apakah fungsi sistem dapat berjalan dan menampilkan hasil yang sesuai. Rencana pengujiannya adalah menguji *input*, hasil yang diharapkan dan *output output* yang ditampilkan oleh sistem. Hasil pengujian *black box* adalah pada Tabel 4.2.

Tabel 4.2 Pengujian fungsionalitas *black box*

<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Kesimpulan
Mengambil data dari dataset pencari kerja	Menampilkan rekomendasi lowongan pekerjaan	Sesuai	Berhasil
Mengisi semua data secara manual pada <i>form</i>	Menampilkan rekomendasi lowongan pekerjaan	Sesuai	Berhasil
Tidak mengisi salah satu data secara manual pada <i>form</i>	Menampilkan alert 'semua data harus diisi' dan tidak menampilkan rekomendasi	Sesuai	Berhasil
Tidak mengisi semua data secara manual pada <i>form</i>	Menampilkan alert 'semua data harus diisi' dan tidak menampilkan rekomendasi	Sesuai	Berhasil

4.2.2 *Precision*

Pengujian selanjutnya adalah menguji nilai presisi dari hasil rekomendasi yang telah diberikan sistem untuk mengetahui seberapa relevan hasil rekomendasi pekerjaan yang diberikan pada data yang diujikan. Pada pengujian ini mengambil 5 sampel data pencari kerja dari 5 *current position* yang sama untuk diujikan. Pemilihan 5 *current position* ini ditentukan dari jumlah terbanyak sesuai pada Gambar 4.12. Adapun 5 posisi yang akan dijadikan skenario pengujian adalah *Backend Developer*, *Frontend Developer*, *Fullstack Developer*, *Mobile Developer*, dan *Web Developer*. Pengujian dilakukan dengan menganalisis secara manual hasil dari 10 lowongan pekerjaan yang direkomendasikan oleh sistem.



Gambar 4.12 Grafik 10 current position terbanyak

1. Skenario Pengujian 1

Pada skenario 1 digunakan 5 sampel pencari kerja dengan posisi *Backend Developer* untuk diujikan hasil rekomendasi yang diperoleh dari sistem. Pada Tabel 4.3 adalah hasil rekomendasi pekerjaan, berupa 10 nilai *similarity* tertinggi yang didapat dari tiap-tiap sampel pengguna yang diujikan.

Tabel 4.3 Nilai *similarity* skenario 1

ID	Nilai <i>Similarity</i> Skenario 1									
	1	2	3	4	5	6	7	8	9	10
32	0.3091	0.1827	0.1725	0.1693	0.15	0.1473	0.1466	0.1413	0.1348	0.1327
42	0.2547	0.2479	0.2376	0.2324	0.2235	0.222	0.2211	0.2174	0.2129	0.208
49	0.264	0.2261	0.2207	0.2067	0.1987	0.1976	0.1819	0.1755	0.1655	0.1631
68	0.3168	0.2359	0.2337	0.2258	0.2227	0.2225	0.1969	0.1888	0.1842	0.1839
85	0.218	0.2133	0.1945	0.1859	0.1835	0.1783	0.1755	0.1722	0.1714	0.1701

Dari hasil rekomendasi tersebut, dilakukan analisis kecocokan antara posisi pencari kerja saat ini dengan posisi lowongan pekerjaan yang direkomendasikan. Nilai 1 untuk hasil rekomendasi yang relevan dan 0 untuk hasil yang tidak relevan. Hasil rekomendasi dianggap relevan ketika posisi yang direkomendasikan sama dengan posisi yang diujikan yaitu *Backend Developer*, jika mendapat posisi selain itu maka dianggap tidak relevan. Nilai presisi dihitung dengan rumus pada Persamaan 2.5, yaitu dengan membagi jumlah item relevan dengan semua item yang direkomendasikan. Hasil analisis relevansi dapat dilihat pada Tabel 4.4.

Tabel 4.4 Nilai *precision* skenario 1

ID	Relevansi Hasil Rekomendasi Skenario 1										Presisi
	1	2	3	4	5	6	7	8	9	10	
32	0	1	1	1	0	0	0	0	0	1	4/10
42	0	1	1	1	0	0	0	1	0	0	4/10
49	1	0	0	0	0	0	1	0	1	0	3/10
68	1	0	0	0	0	1	0	0	1	1	4/10
85	1	0	1	0	0	0	0	0	0	1	3/10
Rata-rata presisi skenario 1											18/50 = 0.36

2. Skenario Pengujian 2

Pada skenario 1 digunakan 5 sampel pencari kerja dengan posisi *Frontend Developer* untuk diujikan hasil rekomendasi yang diperoleh dari sistem. Pada Tabel 4.5 adalah hasil rekomendasi pekerjaan, berupa 10 nilai *similarity* tertinggi yang didapat dari tiap-tiap sampel pengguna yang diujikan.

Tabel 4.5 Nilai *similarity* skenario 2

ID	Nilai <i>Similarity</i> Skenario 2									
	1	2	3	4	5	6	7	8	9	10
18	0.2964	0.2536	0.2275	0.2189	0.214	0.2075	0.2064	0.2064	0.2059	0.2022
27	0.2719	0.2694	0.2488	0.2415	0.199	0.1966	0.1878	0.1743	0.1705	0.1667
40	0.2174	0.1942	0.1921	0.1902	0.184	0.1823	0.1814	0.1783	0.1782	0.1776
57	0.2247	0.2126	0.1896	0.1885	0.1866	0.1851	0.1836	0.1796	0.1719	0.155
97	0.2595	0.2209	0.2179	0.217	0.2108	0.2076	0.2063	0.1954	0.1945	0.1875

Dari hasil rekomendasi tersebut, dilakukan analisis kecocokan antara posisi pencari kerja saat ini dengan posisi lowongan pekerjaan yang direkomendasikan. Nilai 1 untuk hasil rekomendasi yang relevan dan 0 untuk hasil yang tidak relevan. Hasil rekomendasi dianggap relevan ketika posisi yang direkomendasikan sama dengan posisi yang diujikan yaitu *Frontend Developer*, jika mendapat posisi selain itu maka dianggap tidak relevan. Nilai presisi dihitung dengan rumus pada Persamaan 2.5, yaitu dengan membagi jumlah item relevan dengan semua item yang direkomendasikan. Hasil analisis relevansi dapat dilihat pada Tabel 4.6.

Tabel 4.6 Nilai *precision* skenario 2

ID	Relevansi Hasil Rekomendasi Skenario 2										Presisi
	1	2	3	4	5	6	7	8	9	10	
18	1	1	0	1	1	0	0	0	1	0	5/10
38	1	1	0	0	1	0	0	1	0	0	4/10
40	1	0	1	0	0	1	0	1	0	0	4/10
56	1	0	0	0	0	1	1	1	1	1	6/10
66	1	1	0	1	0	1	0	0	0	1	5/10
Rata-rata presisi skenario 2											24/50 = 0.48

3. Skenario Pengujian 3

Pada skenario 1 digunakan 5 sampel pencari kerja dengan posisi *Fullstack Developer* untuk diujikan hasil rekomendasi yang diperoleh dari sistem. Pada Tabel 4.7 adalah hasil rekomendasi pekerjaan, berupa 10 nilai *similarity* tertinggi yang didapat dari tiap-tiap sampel pengguna yang diujikan.

Tabel 4.7 Nilai *similarity* skenario 3

ID	Nilai <i>Similarity</i> Skenario 3									
	1	2	3	4	5	6	7	8	9	10
21	0.3084	0.2641	0.238	0.1975	0.1919	0.1911	0.1872	0.1834	0.1834	0.1797
47	0.2824	0.2548	0.2102	0.1968	0.1928	0.192	0.1915	0.1866	0.1763	0.1756
55	0.2079	0.178	0.1753	0.1673	0.1668	0.1667	0.1646	0.1642	0.158	0.157
59	0.2517	0.226	0.2139	0.2129	0.2111	0.2103	0.2037	0.2026	0.2005	0.1928
96	0.3119	0.2982	0.2825	0.2509	0.2443	0.2434	0.2364	0.2283	0.2165	0.2154

Dari hasil rekomendasi tersebut, dilakukan analisis kecocokan antara posisi pencari kerja saat ini dengan posisi lowongan pekerjaan yang direkomendasikan. Nilai 1 untuk hasil rekomendasi yang relevan dan 0 untuk hasil yang tidak relevan. Hasil rekomendasi dianggap relevan ketika posisi yang direkomendasikan sama dengan posisi yang diujikan yaitu *Fullstack Developer*, jika mendapat posisi selain itu maka dianggap tidak relevan. Nilai presisi dihitung dengan rumus pada Persamaan 2.5, yaitu dengan membagi jumlah item relevan dengan semua item yang direkomendasikan. Hasil analisis relevansi dapat dilihat pada Tabel 4.8.

Tabel 4.8 Nilai *precision* skenario 3

ID	Relevansi Hasil Rekomendasi Skenario 3										Presisi
	1	2	3	4	5	6	7	8	9	10	
21	1	1	0	0	0	0	1	0	0	0	3/10
47	0	0	1	0	1	1	1	1	0	1	6/10
55	0	0	1	1	1	1	0	1	1	0	6/10
59	1	0	0	1	0	0	0	0	0	1	3/10
96	0	1	0	1	0	1	0	1	0	1	5/10
Rata-rata presisi skenario 3											23/50 = 0.46

4. Skenario Pengujian 4

Pada skenario 1 digunakan 5 sampel pencari kerja dengan posisi *Mobile Developer* untuk diujikan hasil rekomendasi yang diperoleh dari sistem. Pada Tabel 4.9 adalah hasil rekomendasi pekerjaan, berupa 10 nilai *similarity* tertinggi yang didapat dari tiap-tiap sampel pengguna yang diujikan.

Tabel 4.9 Nilai *similarity* skenario 4

ID	Nilai <i>Similarity</i> Skenario 4									
	1	2	3	4	5	6	7	8	9	10
3	0.3503	0.3391	0.3011	0.2924	0.2923	0.2802	0.2798	0.273	0.263	0.2585
15	0.2672	0.2102	0.2063	0.2052	0.2049	0.2026	0.1902	0.184	0.1822	0.1797
23	0.3324	0.2562	0.2288	0.2175	0.2147	0.2091	0.2082	0.203	0.203	0.2028
31	0.3232	0.3098	0.2883	0.2736	0.2683	0.2639	0.2591	0.252	0.2506	0.2442
61	0.4077	0.3588	0.3451	0.3378	0.3356	0.3183	0.3035	0.3032	0.2758	0.2735

Dari hasil rekomendasi tersebut, dilakukan analisis kecocokan antara posisi pencari kerja saat ini dengan posisi lowongan pekerjaan yang direkomendasikan. Nilai 1 untuk hasil rekomendasi yang relevan dan 0 untuk hasil yang tidak relevan. Hasil rekomendasi dianggap relevan ketika posisi yang direkomendasikan sama dengan posisi yang diujikan yaitu *Mobile Developer*, jika mendapat posisi selain itu maka dianggap tidak relevan. Nilai presisi dihitung dengan rumus pada Persamaan 2.5, yaitu dengan membagi jumlah item relevan dengan semua item yang direkomendasikan. Hasil analisis relevansi dapat dilihat pada Tabel 4.10.

Tabel 4.10 Nilai *precision* skenario 4

ID	Relevansi Hasil Rekomendasi Skenario 4										Presisi
	1	2	3	4	5	6	7	8	9	10	
3	1	1	1	0	1	1	1	0	0	1	7/10
15	0	0	1	1	0	1	1	1	1	0	6/10
23	1	0	1	1	1	1	0	1	0	0	6/10
31	1	1	1	1	1	1	0	0	1	1	8/10
61	1	1	1	0	1	1	1	1	1	0	8/10
Rata-rata presisi skenario 4											35/50 = 0.7

5. Skenario Pengujian 5

Pada skenario 1 digunakan 5 sampel pencari kerja dengan posisi *Web Developer* untuk diujikan hasil rekomendasi yang diperoleh dari sistem. Pada Tabel 4.11 adalah hasil rekomendasi pekerjaan, berupa 10 nilai *similarity* tertinggi yang didapat dari tiap-tiap sampel pengguna yang diujikan.

Tabel 4.11 Nilai *similarity* skenario 5

ID	Nilai <i>Similarity</i> Skenario 5									
	1	2	3	4	5	6	7	8	9	10
13	0.2954	0.284	0.261	0.2494	0.2467	0.2371	0.2336	0.2312	0.2257	0.2116
17	0.1878	0.1809	0.1803	0.1679	0.1658	0.1655	0.1649	0.164	0.1607	0.1574
24	0.3196	0.2362	0.2313	0.2244	0.2174	0.1998	0.1927	0.1857	0.1766	0.1736
29	0.2412	0.2179	0.2167	0.1965	0.1958	0.1934	0.1881	0.1835	0.1789	0.175
71	0.314	0.3045	0.261	0.2547	0.254	0.253	0.2468	0.2448	0.2402	0.2381

Dari hasil rekomendasi tersebut, dilakukan analisis kecocokan antara posisi pencari kerja saat ini dengan posisi lowongan pekerjaan yang direkomendasikan. Nilai 1 untuk hasil rekomendasi yang relevan dan 0 untuk hasil yang tidak relevan. Hasil rekomendasi dianggap relevan ketika posisi yang direkomendasikan sama dengan posisi yang diujikan yaitu *Web Developer*, jika mendapat posisi selain itu maka dianggap tidak relevan. Nilai presisi dihitung dengan rumus pada Persamaan 2.5, yaitu dengan membagi jumlah item relevan dengan semua item yang direkomendasikan. Hasil analisis relevansi dapat dilihat pada Tabel 4.12.

Tabel 4.12 Nilai *precision* skenario 5

ID	Relevansi Hasil Rekomendasi Skenario 5										Presisi
	1	2	3	4	5	6	7	8	9	10	
13	1	1	0	0	1	0	0	0	0	0	3/10
17	0	1	0	1	1	0	0	1	0	1	5/10
24	1	1	1	0	1	1	1	1	0	1	8/10
29	1	1	1	1	1	1	1	1	1	1	10/10
71	1	1	0	1	1	1	0	0	1	1	7/10
Rata-rata presisi skenario 5											33/50 = 0.6

4.3 Pembahasan

Sistem rekomendasi *content-based filtering* adalah sebuah metode dalam sistem rekomendasi yang menggunakan pendekatan berbasis konten yang ada pada objek penelitian untuk menghasilkan rekomendasi relevan bagi pengguna. Pada penelitian ini dilakukan pengumpulan data dengan metode *web scraping* sebanyak 437 data dari situs *Jobstreet* untuk data lowongan pekerjaan dan 100 data dari situs *LinkedIn* untuk data pencari kerja. Dari data yang telah didapat kemudian dilakukan *preprocessing text* agar data siap diolah, diberi bobot per kata dengan TF-IDF dan dihitung nilai *cosine similarity* untuk mengetahui tingkat

kemiripannya. Berdasarkan hasil pengujian yang telah dilakukan pada semua skenario dari hasil rekomendasi yang diberikan sistem untuk nilai presisi. Didapat rata-rata nilai presisi dari keseluruhan skenario adalah pada Tabel 4.13.

Tabel 4.13 Hasil rata-rata *precision*

Skenario Pengujian	<i>Current Position</i>	<i>Precision</i>
1	<i>Backend Developer</i>	0.36
2	<i>Frontend Developer</i>	0.48
3	<i>Fullstack Developer</i>	0.46
4	<i>Mobile Developer</i>	0.7
5	<i>Web Developer</i>	0.66
Rata-rata presisi		0.53

Hasil diatas menunjukkan nilai presisi untuk setiap skenario pengujian serta nilai rata-rata presisi semua skenario, yaitu 0.53 yang merepresentasikan hasil rekomendasi relevan yang diberikan. Hasil ini dapat dikatakan baik karena nilai presisi yang didapat lebih tinggi daripada presisi pada penelitian-penelitian terdahulu. Nilai presisi tertinggi didapat pada skenario pengujian ke-4 dengan posisi *Mobile Developer* dan terendah didapat pada skenario pengujian ke-1 dengan posisi *Backend Developer*.

Berdasarkan hasil analisis relevansi yang telah dilakukan pada Lampiran 1, untuk skenario pengujian 1, 2, 3 dan 5 terbilang rendah karena cenderung memunculkan posisi yang tidak sama dengan posisi yang diujikan. Munculnya hasil posisi yang tidak sama ini dapat disebabkan oleh kesamaan yang tinggi pada variabel selain posisi (*skills, experiences, licenses* ataupun *job requirement*) sehingga nilai *similarity* yang didapat tetap tinggi dan masuk dalam hasil rekomendasi. Dari fenomena tersebut, dapat disimpulkan bahwa untuk posisi pada skenario pengujian 1, 2, 3 dan 5 cenderung memiliki kesamaan pada *skills* yang diperlukan, hal ini menyebabkan posisi skenario A muncul pada skenario B

maupun sebaliknya, sehingga mempengaruhi nilai presisi menjadi lebih rendah. Sedangkan untuk posisi skenario pengujian 4, ia memiliki kebutuhan *skills* berbeda dengan posisi lain, hal ini menyebabkan posisi tersebut menjadi unik dan mendapat nilai presisi yang lebih tinggi dan hasil yang lebih relevan.

Hasil penelitian ini menunjukkan bahwa telah berhasil dikembangkan sistem rekomendasi pekerjaan dengan metode *content-based filtering* yang menganalisis kemiripan pencari kerja dari variabel *location, about, current position, skills, education, experiences* dan *licenses* dengan lowongan pekerjaan dari variabel *title, company, job requirement* dan *job type*. Untuk sistem yang telah dikembangkan, dilakukan pengujian fungsionalitas dengan metode *blackbox testing* dan mendapat hasil bahwa sistem telah berjalan sesuai fungsinya. Dan berdasarkan hasil uji coba pada 5 skenario pengujian dengan posisi berbeda, didapat rata-rata nilai presisi yang cukup baik, yaitu pada 0.53. Hal ini menunjukkan bahwa sistem rekomendasi pekerjaan cukup efektif dalam memberikan rekomendasi yang relevan.

Dalam kaitannya dengan islam atau integrasi islam, Al-Qur'an tidak secara langsung menjelaskan tentang sistem rekomendasi pekerjaan. Dalam implementasinya pada penelitian ini, sistem rekomendasi pekerjaan bertujuan untuk memberikan rekomendasi lowongan pekerjaan yang paling relevan dengan latar belakang pencari kerja. Hal ini kemudian memiliki keterkaitan dengan qur'an surah Al-Baqarah ayat 267.

يَا أَيُّهَا الَّذِينَ آمَنُوا أَنْفِقُوا مِنْ طَيِّبَاتِ مَا كَسَبْتُمْ وَمِمَّا أَخْرَجْنَا لَكُمْ مِنَ الْأَرْضِ ۗ وَلَا تَيَمَّمُوا الْخَبِيثَ مِنْهُ تُنْفِقُونَ وَلَسْتُمْ
بِأَحْدِيثِهِ إِلَّا أَنْ تُعْمِضُوا فِيهِ ۗ وَاعْلَمُوا أَنَّ اللَّهَ عَنِّي حَمِيدٌ

"Wahai orang-orang yang beriman, infakkanlah sebagian dari hasil usahamu yang baik-baik dan sebagian dari apa yang Kami keluarkan dari bumi untukmu. Janganlah kamu memilih yang buruk untuk kamu infakkan, padahal kamu tidak mau mengambilnya, kecuali dengan memicingkan mata (enggan) terhadapnya. Ketahuilah bahwa Allah Mahakaya lagi Maha Terpuji." (QS. Al-Baqarah (2:267))

Menurut tafsir Kementrian Agama RI, infakkanlah sebagian dari hasil usahamu yang baik-baik dan diperoleh dengan cara yang halal, sebab Allah itu baik dan hanya menerima yang baik-baik. Dan sedekahkanlah sebagian dari apa yang kami keluarkan dari bumi berupa hasil pertanian, tambang, dan lainnya, untukmu. Pilihlah yang baik-baik dari apa yang kamu nafkahkan itu, walaupun tidak harus semuanya baik, tetapi janganlah kamu memilih secara sengaja yang buruk untuk kamu keluarkan guna disedekahkan kepada orang lain, padahal kamu sendiri kalau diberi yang buruk-buruk seperti itu tidak mau mengambilnya melainkan dengan memicingkan mata karena rasa enggan terhadapnya. Jika kamu tidak mau menerima yang buruk-buruk, mengapa kamu berikan yang seperti itu kepada orang lain. Dan ketahuilah dan yakinkan bahwa Allah maha kaya, tidak membutuhkan sedekah kamu, baik pemberian untuknya maupun untuk makhluk-makhluk-Nya, sebab dia bisa memberi secara langsung.

Ayat diatas menjadi pedoman moral yang kuat untuk dapat menyajikan opsi lowongan pekerjaan yang berkualitas tinggi dan sesuai dengan kebutuhan pengguna. Semakin berkualitas hasil yang diberikan, maka semakin bermanfaat juga kebergunaannya bagi pengguna. Hasil dari penelitian ini adalah untuk memberikan rekomendasi pekerjaan yang terbaik bagi para pencari kerja, bukan hanya menawarkan pekerjaan yang tersedia secara acak atau tanpa pertimbangan.

Dalam ayat lain juga menjelaskan tentang pentingnya menunaikan amanat kepada pemiliknya dan pentingnya keadilan dalam memutuskan perkara, yaitu pada Al-Qur'an surah An-Nisa ayat 58:

إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ ۗ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا

“Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat.” (QS. An-Nisa (4:58))

Menurut tafsir Al-Mukhtasar, sesungguhnya Allah menyuruh kalian menunaikan amanat kepada pemiliknya. Dan Dia menyuruh kalian, apabila kalian memutuskan perkara di antara manusia dalam semua urusan mereka, maka putuskanlah perkara mereka dengan adil, jangan memihak atau zalim dalam memutuskan. Sesungguhnya Allah mengingatkan dan memberi bimbingan yang sebaik-baiknya ke arahnya (menjaga amanat) dalam setiap kondisi kalian. Sesungguhnya Allah Maha Mendengar ucapan-ucapan kalian dan Maha Melihat perbuatan-perbuatan kalian.

Ayat ini menegaskan pentingnya menunaikan amanat kepada pemiliknya. Dalam konteks sistem rekomendasi pekerjaan, ini menggarisbawahi pentingnya kepercayaan dan tanggung jawab dalam memberikan rekomendasi pekerjaan kepada pencari kerja. Sistem rekomendasi harus dijalankan dengan itikad baik dan integritas, memastikan bahwa rekomendasi yang diberikan sesuai dengan kebutuhan dan kualifikasi pencari kerja tanpa adanya manipulasi atau kepentingan

yang tersembunyi. Ayat ini juga menekankan pentingnya keadilan dalam memutuskan perkara di antara manusia. Pada sistem rekomendasi pekerjaan, hal ini mencerminkan pentingnya memberikan rekomendasi yang adil dan merata kepada semua pencari kerja, memastikan bahwa setiap pencari kerja diperlakukan secara adil dan memiliki kesempatan yang sama untuk mendapatkan rekomendasi pekerjaan yang sesuai dengan kemampuan mereka.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada bab sebelumnya, pengembangan sistem rekomendasi pekerjaan menggunakan pendekatan *content-based filtering* telah berhasil diimplementasikan. Sehingga dapat disimpulkan sebagai berikut:

1. Berhasil mengembangkan sistem rekomendasi pekerjaan menggunakan pendekatan metode *content-based filtering*
2. Pengujian sistem dengan menggunakan *black box testing* menghasilkan nilai sebesar 100% dengan keberhasilan memberi rekomendasi bagi pengguna.
3. Hasil penerapan *content-based filtering* pada sistem rekomenaasi pekerjaan mendapatkan nilai presisi sebesar 0.53

5.2 Saran

Berdasarkan hasil yang didapat dari penelitian ini, ditemukan masih banyak kekurangan sehingga diberikan saran untuk penelitian kedepan, sebagai berikut:

1. Menerapkan metode rekomendasi lain seperti *collaborative filtering*, untuk meningkatkan hasil rekomendasi.
2. Data pekerjaan terhubung langsung dengan situs *JobStreet* melalui API, untuk memastikan bahwa lowongan pekerjaan masih tersedia.
3. Mengembangkan sistem rekomendasi calon pencari kerja potensial untuk perusahaan yang sedang membuka lowongan.

DAFTAR PUSTAKA

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Aggarwal, C. C. (1997). Recommender Systems. In *Communications of the ACM* (Vol. 40, Issue 3). Springer Cham. <https://doi.org/10.1145/245108.245121>
- Aguado, D., Andrés, J. C., García-Izquierdo, A. L., & Rodríguez, J. (2019). LinkedIn “Big Four”: Job performance validation in the ICT sector. *Revista de Psicología Del Trabajo y de Las Organizaciones*, 35(2), 53–64. <https://doi.org/10.5093/jwop2019a7>
- Asian, J., Nazief, B., Williams, H. E., & Tahaghoghi, S. M. M. (2007). Stemming Indonesian: A confix-stripping approach. *Conferences in Research and Practice in Information Technology Series*, 38(January), 307–314. <https://doi.org/10.1145/1316457.1316459>
- Boukari, S., Fayeche, S., & Faiz, R. (2020). Huntalent: A candidates recommendation system for automatic recruitment via linkedin. *2020 7th International Conference on Social Network Analysis, Management and Security, SNAMS 2020*. <https://doi.org/10.1109/SNAMS52053.2020.9336532>
- Cubrich, M., King, R. T., Mracek, D. L., Strong, J. M. G., Hassenkamp, K., Vaughn, D., & Dudley, N. M. (2021). Examining the criterion-related validity evidence of LinkedIn profile elements in an applied sample. *Computers in Human Behavior*, 120(December 2020), 106742. <https://doi.org/10.1016/j.chb.2021.106742>
- Dhameliya, J., & Desai, N. (2019). Job Recommendation System using Content and Collaborative Filtering based Techniques. *International Journal of Soft Computing and Engineering*, 9(3), 8–13. <https://doi.org/10.35940/ijscce.c3266.099319>
- Dillahunt, T. R., Israni, A., & Lu, A. J. (2021). Examining the use of online platforms for employment: A survey of u.s. job seekers. *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3411764.3445350>
- Fayard, A. L. (2021). Notes on the Meaning of Work: Labor, Work, and Action in the 21st Century. *Journal of Management Inquiry*, 30(2), 207–220. <https://doi.org/10.1177/1056492619841705>
- Feldman, R., & Sanger, J. (2006). The Text Mining Handbook. In *The Text Mining Handbook*. <https://doi.org/10.1017/cbo9780511546914>
- Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible

- are jobs to computerisation? *Technological Forecasting and Social Change*, 114, 254–280. <https://doi.org/10.1016/j.techfore.2016.08.019>
- Hossein, A., Rafsanjani, N., Salim, N., & Aghdam, A. R. (2013). Recommendation Systems: a review. *International Journal of Computational Engineering Research*, 3(5), 47–52.
- Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press; Illustrated edition.
- Johnson, D. (2019). What is LinkedIn. *University of Montana*. umt.edu/elcs
- Kannan, S., & Gurusamy, V. (2014). Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, February 2015, 7–16.
- Kumar, A. A. (2022). Job Recommendation System Based on Skill Sets. *International Journal of Creative Research Thoughts (IJCRT)*, 10(8), 770–785.
- Li, S., Shi, B., Yang, J., Yan, J., Wang, S., Chen, F., & He, Q. (2020). Deep Job Understanding at LinkedIn. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2145–2148. <https://doi.org/10.1145/3397271.3401403>
- Lin, W., Alvarez, S. A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1), 83–105. <https://doi.org/10.1023/A:1013284820704>
- Manning, C. D., Raghavan, P., & Hinrich, S. (2009). *An Introduction to Information Retrieval* (Issue c). <http://www-nlp.stanford.edu/IR-book/>
- Mulay, A., Sutar, S., Patel, J., Chhabria, A., & Mumbaikar, S. (2022). Job Recommendation System Using Hybrid Filtering. *ITM Web of Conferences*, 44, 02002. <https://doi.org/10.1051/itmconf/20224402002>
- Nidhra, S. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, 2(2), 29–50. <https://doi.org/10.5121/ijesa.2012.2204>
- Raharjo, P. N., Handojo, A., & Juwiantho, H. (2022). Sistem Rekomendasi Content Based Filtering Pekerjaan dan Tenaga Kerja Potensial menggunakan Cosine Similarity. *Jurnal Invra*, 10(2), 1–6.
- Rashid, A. H. A., Mohamad, M., Masrom, S., & Selamat, A. (2022). Student Career Recommendation System Using Content-Based Filtering Method. *2022 3rd International Conference on Artificial Intelligence and Data Sciences: Championing Innovations in Artificial Intelligence and Data Sciences for Sustainable Future, AiDAS 2022 - Proceedings, December*, 60–65. <https://doi.org/10.1109/AiDAS56890.2022.9918766>
- Ricci, F., Shapira, B., Rokach, L., & Kantor, P. B. (2011). Recommender Systems

Handbook. In *Recommender Systems Handbook*. Springer New York, NY. <https://doi.org/10.1007/978-0-387-85820-3>

Setiani, B. (2013). Dalam Proses Rekrutmen Tenaga Kerja Di Perusahaan. *Manajemen Sumber Daya Manusia*, 1(1), 41.

Tîrnovanu, A. C., Năstase, M., Şişu, J. A., & Mujaya, J. (2023). Digital Transformation during COVID-19 Pandemic – a Bibliometric Analysis. *Proceedings of the International Conference on Business Excellence*, 17(1), 2070–2084. <https://doi.org/10.2478/picbe-2023-0181>

Van Zyl, E. (1999). Personnel Psychology. In *SA Journal of Industrial Psychology* (Vol. 25, Issue 1). <https://doi.org/10.4102/sajip.v25i1.663>

Weiss, S. M., Indurkha, N., Zhang, T., & Damerau, F. J. (2005). Text mining: Predictive Methods for Analyzing Unstructured Information. In *Text Mining: Predictive Methods for Analyzing Unstructured Information* (1st ed., Vol. 36, Issue 1C). Springer New York, NY. <https://doi.org/10.1007/978-0-387-34555-0>

Xie, Y., & Mujtaba, B. G. (2008). Strategies For Information Systems Candidates And Job Seekers In The Twenty-First Century Workplace. *Review of Business Information Systems (RBIS)*, 12(1), 7–16.

LAMPIRAN

Lampiran 1. Hasil rekomendasi

Skenario	User ID	Job ID	title	similarity
1	32	73	dev ops engineer	0.3091
		32	back end developer	0.1827
		52	backend developer position senior junior	0.1725
		269	node j backend developer hybrid	0.1693
		97	frontend engineer	0.1500
		131	fullstack developer backend focus	0.1473
		414	web developer	0.1466
		164	magang intern	0.1413
		100	frontend engineer react	0.1348
		50	backend developer	0.1327
	42	121	full stack programmer	0.2547
		45	backend developer	0.2479
		50	backend developer	0.2376
		34	back end developer	0.2324
		132	fullstack developer contract month	0.2235
		112	full stack developer	0.2220
		268	next j developer full stack indonesia	0.2211
		32	back end developer	0.2174
		281	php java script developer di jakarta	0.2129
	165	net developer	0.2080	
	49	32	back end developer	0.2640
		126	full stack developer	0.2261
		115	full stack developer angular springboot	0.2207
		73	dev ops engineer	0.2067
		100	frontend engineer react	0.1987
		176	developer	0.1976
		44	backend developer	0.1819
		366	software development	0.1755
		52	backend developer position senior junior	0.1655
	89	front end engineer java script vue j	0.1631	
	68	176	developer	0.3168
		118	full stack engineer	0.2359
		387	staff back end android specialist	0.2337
		121	full stack programmer	0.2258
		342	senior fullstack developer	0.2227
		330	senior back end engineer java	0.2225
		431	web programmer frontend	0.1969
		231	java programmer	0.1888
		32	back end developer	0.1842
		53	back end developer engineer	0.1839
	85	32	back end developer	0.2180
		145	go lang developer	0.2133
		52	backend developer position senior junior	0.1945
		198	programmer	0.1859
		165	net developer	0.1835

		149	golang programmer	0.1783
		111	full stack developer	0.1755
		146	go lang developer	0.1722
		115	full stack developer angular springboot	0.1714
		34	back end developer	0.1701
2	18	96	frontend developer officer	0.2964
		87	front end developer uni lah student app	0.2536
		220	web developer	0.2275
		101	frontend mobile developer	0.2189
		85	front end developer	0.2140
		109	full stack developer	0.2075
		176	developer	0.2064
		406	web developer	0.2064
		88	front end developer executive downstream	0.2059
		121	full stack programmer	0.2022
	27	96	frontend developer officer	0.2719
		94	frontend developer	0.2694
		115	full stack developer angular springboot	0.2488
		132	fullstack developer contract month	0.2415
		101	frontend mobile developer	0.1990
		424	web developer staff	0.1966
		118	full stack engineer	0.1878
		93	frontend developer	0.1743
		6	net developer software developer	0.1705
		112	full stack developer	0.1667
	40	96	frontend developer officer	0.2174
		250	mobile app developer	0.1942
		42	backend frontend web developer net core golang laravel nest j nuxt j	0.1921
		402	ux writer	0.1902
		259	mobile developer	0.1840
		234	jr frontend programmer	0.1823
		107	full stack developer	0.1814
		99	frontend engineer flutter	0.1783
		138	fullstack mobile programmer supervisor flutter laravel	0.1782
		164	magang intern	0.1776
	57	96	frontend developer officer	0.2247
		132	fullstack developer contract month	0.2126
		121	full stack programmer	0.1896
		118	full stack engineer	0.1885
		386	staff	0.1866
		94	frontend developer	0.1851
		93	frontend developer	0.1836
		85	front end developer	0.1796
		101	frontend mobile developer	0.1719
		97	frontend engineer	0.1550

	97	96	frontend developer officer	0.2595
		100	frontend engineer react	0.2209
		406	web developer	0.2179
		234	jr frontend programmer	0.2170
		412	web developer	0.2108
		102	frontend react j developer sde	0.2076
		282	php developer	0.2063
		420	web developer java springboot asap banking	0.1954
		417	web developer banking industry	0.1945
		336	senior frontend developer	0.1875
3	21	126	full stack developer	0.3084
		115	full stack developer angular springboot	0.2641
		32	back end developer	0.2380
		394	technical consultant dev ops	0.1975
		52	backend developer position senior junior	0.1919
		44	backend developer	0.1911
		163	full stack developer	0.1872
		235	junior c software developer	0.1834
		100	frontend engineer react	0.1834
		150	group technology center senior manager	0.1797
	47	100	frontend engineer react	0.2824
		363	software architect developer simulation training unreal engine unity	0.2548
		268	next j developer full stack indonesia	0.2102
		102	frontend react j developer sde	0.1968
		342	senior fullstack developer	0.1928
		115	full stack developer angular springboot	0.1920
		135	full stack developer programmer	0.1915
		114	full stack developer node j react j	0.1866
		421	web developer senior	0.1763
		112	full stack developer	0.1756
	55	96	frontend developer officer	0.2079
		234	jr frontend programmer	0.1780
		107	full stack developer	0.1753
		341	senior full stack web developer	0.1673
		113	full stack developer java spring boot angular	0.1668
		112	full stack developer	0.1667
		417	web developer banking industry	0.1646
		109	full stack developer	0.1642
		121	full stack programmer	0.1580
		219	trainer	0.1570
	59	132	fullstack developer contract month	0.2517
		281	php java script developer di jakarta	0.2260
		96	frontend developer officer	0.2139
135		full stack developer programmer	0.2129	
417		web developer banking industry	0.2111	

		420	web developer java springboot asap banking	0.2103
		412	web developer	0.2037
		423	web developer java springboot banking	0.2026
		406	web developer	0.2005
		163	full stack developer	0.1928
	96	176	developer	0.3119
		115	full stack developer angular springboot	0.2982
		268	next j developer full stack indonesia	0.2825
		113	full stack developer java spring boot angular	0.2509
		32	back end developer	0.2443
		121	full stack programmer	0.2434
		421	web developer senior	0.2364
		129	fullstack developer angular net	0.2283
		417	web developer banking industry	0.2165
		111	full stack developer	0.2154
		4	3	82
252	mobile app developer react native			0.3391
262	mobile developer react j flutter			0.3011
19	app developer			0.2924
259	mobile developer			0.2923
263	mobile development			0.2802
138	fullstack mobile programmer supervisor flutter laravel			0.2798
16	android programmer			0.2730
378	software engineer mobile			0.2630
357	senior web mobile programmer			0.2585
15	207		programmer apps	0.2672
	328		senior android front end developer	0.2102
	262		mobile developer react j flutter	0.2063
	263		mobile development	0.2052
	16		android programmer	0.2049
	252		mobile app developer react native	0.2026
	79		flutter kotlin mobile apps developer	0.1902
	259		mobile developer	0.1840
	82		flutter mobile developer	0.1822
	13		android developer banking industry	0.1797
23	252		mobile app developer react native	0.3324
	16		android programmer	0.2562
	138		fullstack mobile programmer supervisor flutter laravel	0.2288
	263		mobile development	0.2175
	82		flutter mobile developer	0.2147
	259		mobile developer	0.2091
	13		android developer banking industry	0.2082
	191		mobile application developer	0.2030
291	programmer		0.2030	

	31	81	flutter developer	0.2028
		252	mobile app developer react native	0.3232
		262	mobile developer react j flutter	0.3098
		82	flutter mobile developer	0.2883
		259	mobile developer	0.2736
		263	mobile development	0.2683
		138	fullstack mobile programmer supervisor flutter laravel	0.2639
		207	programmer apps	0.2591
		378	software engineer mobile	0.2520
		266	mobile technical executive downstream	0.2506
		255	mobile apps programmer	0.2442
	61	82	flutter mobile developer	0.4077
		252	mobile app developer react native	0.3588
		266	mobile technical executive downstream	0.3451
		16	android programmer	0.3378
		262	mobile developer react j flutter	0.3356
		192	mobile banking developer	0.3183
		250	mobile app developer	0.3035
		251	mobile app developer	0.3032
	191	mobile application developer	0.2758	
	19	app developer	0.2735	
5	13	406	web developer	0.2954
		412	web developer	0.2840
		176	developer	0.2610
		268	next j developer full stack indonesia	0.2494
		421	web developer senior	0.2467
		88	front end developer executive downstream	0.2371
		121	full stack programmer	0.2336
		129	fullstack developer angular net	0.2312
		239	junior mobile developer swift	0.2257
		113	full stack developer java spring boot angular	0.2116
	17	3	net developer	0.1878
		431	web programmer frontend	0.1809
		342	senior fullstack developer	0.1803
		417	web developer banking industry	0.1679
		420	web developer java springboot asap banking	0.1658
		52	backend developer position senior junior	0.1655
		44	backend developer	0.1649
		232	java web developer call center team	0.1640
		4	net developer solution	0.1607
		412	web developer	0.1574
	24	424	web developer staff	0.3196
		418	web developer digital content creator	0.2362
		416	web developer	0.2313
		109	full stack developer	0.2244

		436	website developer	0.2174	
		406	web developer	0.1998	
		412	web developer	0.1927	
		417	web developer banking industry	0.1857	
		17	angular developer	0.1766	
		420	web developer java springboot asap banking	0.1736	
	29		417	web developer banking industry	0.2412
			424	web developer staff	0.2179
			420	web developer java springboot asap banking	0.2167
			341	senior full stack web developer	0.1965
			418	web developer digital content creator	0.1958
			222	web developer supervisor	0.1934
			423	web developer java springboot banking	0.1881
			431	web programmer frontend	0.1835
			406	web developer	0.1789
			412	web developer	0.1750
	71		406	web developer	0.3140
			412	web developer	0.3045
			268	next j developer full stack indonesia	0.2610
			416	web developer	0.2547
			417	web developer banking industry	0.2540
			220	web developer	0.2530
			96	frontend developer officer	0.2468
			112	full stack developer	0.2448
			431	web programmer frontend	0.2402
			436	website developer	0.2381