

**SISTEM REKOMENDASI LOWONGAN PEKERJAAN
MENGUNAKAN METODE COSINE SIMILARITY
BERBASIS WEBSITE**

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



DISUSUN OLEH :

BAGAS AFZA JOKO ARIYANTO

NIM 32601900008

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2022

FINAL PROJECT
JOB RECOMMENDATION SYSTEM
USING COSINE SIMILARITY METHOD
BASED ON WEBSITE

Proposed to complete the requirement to obtain a bachelor's degree (S1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University



MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2022

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **“Sistem Rekomendasi Lowongan Pekerjaan Menggunakan Metode *Cosine Similarity* Berbasis Website”** ini disusun oleh :

Nama : Bagas Afza Joko Ariyanto

NIM : 32601900008

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari :

Tanggal :

Mengesahkan,

Pembimbing I

Pembimbing II

Sam Farisa Chærul Haviana, ST.M.Kom

Dedy Kurniadi, ST.M.Kom

NIDN. 0628028602

NIDN. 0622058802

Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Sultan Agung

Ir. Sri Mulyono, M.Eng

NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan Tugas Akhir dengan judul **“Sistem Rekomendasi Lowongan Pekerjaan Menggunakan Metode *Cosine Similarity* Berbasis Website”** ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari :

Tanggal :

TIM PENGUJI

Penguji I



Andi Rianisyah, ST.M.Kom

NIDN. 0609108802

Penguji II



Ghufroon, ST, M.Kom

NIDN. 0602079005



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008
Judul Tugas Akhir : SISTEM REKOMENDASI LOWONGAN
PEKERJAAN MENGGUNAKAN METODE
COSINE SIMILARITY BERBASIS WEBSITE

Dengan ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 25 Februari 2023

Yang Menyatakan,



Bagas Afza Joko Ariyanto

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Bagas Afza Joko Ariyanto
NIM : 32601900008
Program Studi : Teknik Informatika
Fakultas : Teknologi Industri
Alamat Asal : Desa Batu RT 03 RW 01 Kecamatan Karangtengah
Kabupaten Demak

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **SISTEM REKOMENDASI LOWONGAN PEKERJAAN MENGGUNAKAN METODE COSINE SIMILARITY BERBASIS WEBSITE** Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 25 Februari 2023

Yang menyatakan,



Bagas Afza Joko Ariyanto

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Sistem Rekomendasi Lowongan Pekerjaan Menggunakan Metode *Cosine Similarity* Berbasis Website” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Sam Farisa Chaerul Haviana, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu.
4. Dosen pembimbing II penulis Dedy Kurniadi, S.T., M.Kom yang memberikan banyak nasehat dan saran.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini dan masa mendatang.

Semarang, 25 Februari 2023



Bagas Afza Joko Ariyanto

DAFTAR ISI

SISTEM REKOMENDASI LOWONGAN PEKERJAAN MENGGUNAKAN METODE COSINE SIMILARITY BERBASIS WEBSITE

JOB RECOMMENDATION SYSTEM USING COSINE SIMILARITY METHOD BASED ON WEBSITE

LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
ABSTRAK	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	7
2.2.1 <i>Curriculum Vitae</i> (CV)	7
2.2.2 Glint	7
2.2.3 <i>Text Mining</i>	8
2.2.4 <i>Text Preprocessing</i>	9
2.2.5 <i>Web Scraping</i>	12
2.2.6 <i>Cosine Similarity</i>	12

2.2.7	<i>Natural Language Processing</i>	13
2.2.8	<i>Flask</i>	13
BAB III METODOLOGI PENELITIAN		14
3.1	Metode Penelitian.....	14
3.1.1	<i>Study Literature</i>	14
3.1.2	<i>Data Collecting</i>	14
3.1.3	Perancangan Arsitektur Sistem	15
3.1.4	Gambaran Sistem	18
3.1.5	Indentifikasi Perangkat Lunak	18
3.1.6	Perancangan <i>User Interface</i>	21
BAB IV HASIL DAN ANALISIS PENELITIAN		23
4.1	Hasil Perhitungan	23
4.2	Contoh Dokumen CV	23
4.3	Implementasi <i>Text Preprocessing</i>	25
4.4	Hasil Implementasi Sistem.....	31
4.5	Hasil Pengujian Sistem.....	33
4.6	Validasi Implementasi Algoritma.....	36
BAB V KESIMPULAN DAN SARAN		37
5.1	Kesimpulan.....	37
5.2	Saran.....	37
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR TABEL

Tabel 3. 1 Contoh hasil <i>CountVectorizer</i>	20
Tabel 4. 1 Rincian Contoh Dokumen CV	23
Tabel 4. 2 Pengujian Performa Sistem	33
Tabel 4. 3 Pengujian <i>Black Box</i>	34
Tabel 4. 4 Pengujian <i>Translate to English</i>	35



DAFTAR GAMBAR

Gambar 2. 1 <i>Flowchart</i> Alur <i>Text Preprocessing</i>	10
Gambar 3. 1 <i>Flowchart</i> Rancangan Arsitektur Sistem	16
Gambar 3. 2 Rancangan Antar Muka Halaman Utama.....	21
Gambar 3. 3 Rancangan Antar Muka Halaman Rekomendasi	22
Gambar 4. 1 <i>Parsing</i>	26
Gambar 4. 2 <i>Case Folding</i>	27
Gambar 4. 3 <i>Translate To English</i>	28
Gambar 4. 4 <i>Tokenization</i>	28
Gambar 4. 5 <i>Filtering</i>	29
Gambar 4. 6 <i>Stemming</i>	30
Gambar 4. 7 Penggabungan Hasil <i>Text Preprocessing</i>	30
Gambar 4. 8 Halaman Utama.....	31
Gambar 4. 9 Tampilan Saat Format CV Yang Dimasukan Salah.....	32
Gambar 4. 10 Halaman Rekomendasi.....	32



ABSTRAK

Curriculum Vitae (CV) merupakan sebuah dokumen yang sering digunakan untuk melamar pekerjaan di sebuah perusahaan, yang mana isi dari CV meliputi biodata diri, kualitas diri yang biasanya terdapat kemampuan yang relevan dengan posisi pekerjaan yang ingin dilamar dan juga menjadi dokumen untuk mempertimbangkan apakah layak untuk penempatan posisi yang dilamar. Oleh karena itu, dibutuhkan teknik yang dapat memberikan rekomendasi lowongan pekerjaan yang relevan dengan CV yang dimiliki. Metode *cosine similarity* digunakan untuk menghitung *similarity* atau kemiripan dokumen CV, dengan tujuan untuk menghitung seberapa besar tingkat *similarity*-nya dengan data lowongan pekerjaan sehingga nantinya dapat digunakan sebagai parameter rekomendasi lowongan yang cocok bagi pelamar pekerjaan. Pada perhitungan metode *cosine similariy* semakin mendekati angka 1 maka kecocokan CV pelamar dengan data lowongan pekerjaan semakin besar potensi diterima dalam perusahaan yang bersangkutan dan sebaliknya jika mendekati angka 0 maka dinyatakan sebaliknya. Saat melakukan uji sistem, performa sistem bergantung kepada jumlah kata yang ada pada CV semakin kaya kata yang terdapat pada CV, proses pada *text preprocessing* akan semakin lama. Sistem ini bergantung kepada koneksi internet. Uji *test* yang dilakukan pada kecepatan koneksi internet rata-rata 100 Kbps sampai 200 Kbps program tidak memiliki *error* saat dijalankan. Saat sistem melalui uji *parsing* data dan *translate to english* terdapat kondisi pengujian yang tidak sesuai hasil. Yaitu pengujian terhadap *parsing* CV dengan bahasa campuran yaitu bahasa Indonesia dan bahasa Inggris dengan hasil ketika dilakukan penerjemah beberapa kata tidak bisa selaras kedalam satu bahasa.

Kata Kunci: CV (*Curriculum Vitae*), *Cosine Similarity*

ABSTRACT

Curriculum Vitae (CV) is a document that is often used to apply for a job in a company, where the contents of the CV include personal biodata, self-qualities which usually have abilities that are relevant to the job position you want to apply for and also become a document to consider whether it is feasible to apply for a job. the position applied for. Therefore, a technique is needed that can provide recommendations for job vacancies that are relevant to the CV you have. The cosine similarity method is used to calculate the similarity or similarity of CV documents, with the aim of calculating how much the level of similarity is to the job vacancy data so that later it can be used as a parameter for recommendation of suitable vacancies for job applicants. In the calculation of the cosine similariy method, the closer to number 1, the suitability of the applicant's CV with the job vacancy data, the greater the potential to be accepted in the company concerned and vice versa if it is close to 0, the opposite is stated. When testing the system, system performance depends on the number of words in the CV, the richer the words in the CV, the longer the text preprocessing process will be. This system depends on internet connection. The tests were conducted at an average internet connection speed of 100 Kbps to 200 Kbps, the program had no errors when it was run. When the system goes through data parsing tests and translating to English, there are test conditions that don't match the results. That is testing of CV parsing with mixed languages, namely Indonesian and English with the result that when the translation is done, some words cannot be aligned into one language.

Keywords: CV (*Curriculum Vitae*), *Cosine Similarity*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin berkembangnya teknologi *modern*, banyak sektor yang menerima dampaknya baik dalam sektor industri, ekonomi, transportasi, militer, kesehatan, bahkan dalam sektor pertanian. Semakin kesini teknologi mulai mendominasi dengan mulai banyaknya teknologi yang dapat menyerupai pola pikir bahkan perilaku dari manusia. Dengan pemanfaatan yang bijak teknologi tersebut dapat menghemat tenaga, waktu, dan biaya jika dibandingkan dengan menggunakan tenaga manusia.

Dari pemaparan diatas penulis ingin membuat sebuah sistem yang dapat membantu kehidupan manusia khususnya dalam bidang pencarian pekerjaan. Sistem yang akan dibuat yaitu sistem yang mampu memberikan rekomendasi lowongan pekerjaan berdasarkan CV (*Curriculum Vitae*) yang dimiliki pelamar. Dengan adanya sistem ini diharap masyarakat dapat mempermudah dalam mencari pekerjaan yang sesuai dengan keahlian yang dimiliki, sehingga dapat meningkatkan rasio diterimanya menjadi karyawan.

Dapat disimpulkan dari pernyataan diatas bahwa pada Proposal Tugas Akhir ini, penulis akan berfokus untuk mengambil studi kasus yaitu mengambil data dari CV dan mengambil data lowongan pekerjaan dari sebuah situs yang menyediakan lowongan pekerjaan. CV sendiri merupakan dokumen umum yang harus dibawa ketika ingin melamar pekerjaan di sebuah perusahaan. Dari hasil penelitian yang dilakukan oleh (Dewi dkk., 2021) tentang “*Pelaksanaan Rekrutmen, Seleksi dan Penempatan Kerja Karyawan (Studi pada Karyawan PT Gemala Kempa Daya)*” terdapat 265 pelamar yang melamar di PT. Gemala Kempa Daya, pada tahap seleksi berkas yang memenuhi kualifikasi terdapat 192 pelamar. Masih dalam seleksi berkas sudah ada sekitar 73 pelamar yang gugur karena berkas yang mereka bawa tidak memenuhi kualifikasi. Dari hasil penelitian (Suhandi dkk., 2020) tentang “*Dinamika Permasalahan Ketenagakerjaan Dan Pengangguran di Indonesia*” menjelaskan bahwa daya saing tenaga kerja Indonesia relatif masih rendah

dibandingkan dengan daya saing negara tetangga. Rendahnya daya saing ini disebabkan karena rendahnya mutu SDM (Sumber Daya Manusia) sebagai akibat dari rendahnya tingkat pendidikan, rendahnya kompetensi kerja dan kecocokan data CV dengan pekerjaan yang ingin dilamar.

Dari kedua hasil penelitian diatas penulis memiliki beberapa kesimpulan diantaranya *Human Resources Development* atau yang lebih dikenal dengan HRD *first impression* mereka pada pelamar didapatkan melalui berkas yang dikirim pada saat pelamar melamar pekerjaan, yang pastinya salah satu berkas yang dikirim adalah CV yang didalamnya mencakup *skill*, data diri, dan pengalaman si pelamar. Tidak hanya itu tolak ukur HRD yang kedua adalah mengenai korelasi CV dari si pelamar dengan posisi atau *job* yang akan diambil. Semakin relevan isi CV dengan *job* yang akan diambil semakin besar pula kesempatan si pelamar diterima dalam perusahaan yang bersangkutan.

Oleh karena itu, penulis bertujuan membuat sistem yang mampu merekomendasikan lowongan pekerjaan yang relevan dengan data CV yang dikirimkan. Dengan begitu masyarakat bisa menerima data lowongan pekerjaan yang sesuai dengan keahlian yang dimiliki, sehingga dapat meningkatkan rasio diterima pada sebuah perusahaan yang bersangkutan.

1.2 Perumusan Masalah

Bagaimana metode *Cosine Similarity* yang diimplementasikan dalam sistem berbasis *website* dapat menghitung kesamaan antara dua buah objek dan memberikan rekomendasi lowongan pekerjaan dari data CV yang sudah diolah menggunakan metode *Natural Language Processing*.

1.3 Pembatasan Masalah

Dalam penelitian kali ini ada beberapa batasan masalah yang ditetapkan. Adapun batasan masalah dalam tugas akhir ini adalah:

1. *Dataset* yang digunakan berasal dari sebuah situs pencarian lowongan pekerjaan yaitu glints (<https://glints.com/id>). Glints adalah platform penemuan karir yang menjembatani kesenjangan antara pendidikan dan pekerjaan dengan menjadi

platform profesional bagi kaum muda yang berfokus pada penemuan dan pengembangan karir. Glints sendiri bekerjasama dengan universitas dan politeknik untuk menjangkau lulusan dan mencocokkan mereka dengan perusahaan berkualitas.

2. Dari situs glints akan dilakukan proses *Web Scraping* untuk mengambil data lowongan pekerjaan yang berupa nama lowongan, nama perusahaan, asal perusahaan, kategori pekerjaan, *skills* yang dibutuhkan, dan deskripsi pekerjaan.
3. Jumlah data yang diambil dari situs glints terdiri dari 14 kategori lowongan pekerjaan yaitu, *software, data science, design, operations, marketing, product management, bussiness, engineering, management, finance, human resource development (HRD), media dan komunikasi, konsultan, dan kategori lainnya*. Dengan setiap kategori mengambil kurang lebih 30 data lowongan pekerjaan.
4. CV yang dikirimkan dapat berupa *file* berformat .pdf dengan CV tidak berupa gambar yang di-*convert* menjadi PDF, teks didalamnya dapat dilakukan proses *copy paste*, setidaknya mengandung deskripsi diri, *skill* atau keahlian, bidang yang digeluti, riwayat karir atau proyek yang pernah dikerjakan, dan maksimal halaman CV 3 lembar.
5. Penelitian ini hanya membahas seberapa besar tingkat kesamaan CV dengan data lowongan pekerjaan, sedangkan untuk menentukan diterimanya pelamar dalam sebuah perusahaan yang bersangkutan dikembalikan kepada perusahaan tempat pelamar melamar.

1.4 Tujuan

Tujuan tugas akhir ini adalah melakukan uji kecocokan atau kemiripan antara data lowongan pekerjaan dengan data CV yang dikirim yang nantinya data CV tersebut akan diolah oleh sistem, kemudian akan memunculkan rekomendasi lowongan pekerjaan yang relevan dengan CV yang dikirim.

1.5 Manfaat

Berdasarkan permasalahan dan tujuan penelitian, maka penulis mengharapkan penelitian ini dapat memberikan manfaat yang besar bagi

penggunanya guna memudahkan pengguna dalam mencari lowongan pekerjaan yang sesuai dengan kualitas diri melalui CV, sehingga diharap dapat meningkatkan rasio diterimanya pelamar dalam lowongan yang direkomendasikan.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang akan dipakai penulis dalam pembuatan laporan tugas akhir ini adalah sebagai berikut:

BAB I	:	PENDAHULUAN
		Pada bab ini penulis mengutarakan latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, serta sistematika penulisan.
BAB II	:	TINJAUAN PUSTAKA DAN DASAR TEORI
		Pada bab ini memuat penelitian-penelitian sebelumnya dan dasar teori yang berguna untuk membantu penulis memahami bagaimana teori yang berhubungan dengan metode <i>Cosine Similarity</i> dan metode <i>Natural Language Processing</i> untuk penelitian ini.
BAB III	:	METODE PENELITIAN
		Pada bab ini mengungkapkan proses tahapan-tahapan penelitian dimulai dari mendapatkan data hingga proses uji kemiripan atau kesamaan data yang ada.
BAB IV	:	HASIL DAN ANALISIS PENELITIAN
		Pada bab ini penulis mengungkapkan hasil penelitian yaitu uji kemiripan atau kesamaan menggunakan metode <i>Cosine Similarity</i> beserta pengujian kemiripan.
BAB V	:	KESIMPULAN DAN SARAN
		Pada bab ini penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Curriculum Vitae (CV) merupakan sebuah dokumen yang sering digunakan untuk melamar pekerjaan di sebuah perusahaan, yang mana isi dari CV meliputi biodata diri, kualitas diri yang biasanya terdapat kemampuan yang relevan dengan posisi pekerjaan yang ingin dilamar dan juga menjadi dokumen untuk mempertimbangkan apakah layak untuk penempatan posisi yang dilamar. Tahap seleksi CV masuk pada salah satu metode yang umum digunakan oleh perusahaan dalam penerimaan karyawan baru yaitu metode seleksi non ilmiah. Metode seleksi non ilmiah merupakan seleksi yang didasarkan pada pengalaman pelamar yang berpedoman pada relevansi spesifikasi pekerjaan yang dilamar (Hatmatri, 2020). Oleh karena itu, dengan membuat sistem berbasis *website* yang mampu memberikan rekomendasi pekerjaan adalah salah satu pilihan yang baik.

Pada sistem ini nantinya akan ditanamkan metode *Natural Language Processing* (NLP). NLP merupakan salah satu ilmu multidisiplin yang berfokus pada interaksi manusia dan komputer melalui bahasa alami manusia (Martinez, 2010). Dari hasil penelitian (Rosyadi dkk., 2020) tentang “Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (*Natural Language Processing*)” mendapatkan kesimpulan bahwa dengan menggunakan metode *Natural Language Processing* sistem dapat memahami bahasa yang digunakan manusia dalam membuat sebuah CV, karena nantinya sistem akan melakukan interaksi melalui CV kemudian akan dikomunikasikan dengan data lowongan pekerjaan sehingga sistem dapat mencari lowongan pekerjaan yang relevan dengan CV yang dikirimkan.

Kemudian dari hasil penelitian (Riyani dkk., 2019) tentang “Penerapan *Cosine Similarity* dan Pembobotan *TF-IDF* (*Term Frequency Inverse Document Frequency*) untuk Mendeteksi Kemiripan Dokumen” mendapatkan kesimpulan bahwa dengan metode *Cosine Similarity* telah berhasil mendeteksi kemiripan suatu dokumen. Dari hasil penelitian tersebut penulis mendapat gambaran bahwa dengan

menggunakan metode *Cosine Similarity* dapat mencocokkan dua buah objek data yang mana dari penulis akan mencocokkan dua data dokumen yaitu CV dengan data lowongan pekerjaan yang nantinya hasil kecocokan antara dua objek tersebut akan menghasilkan sebuah rekomendasi pekerjaan.

Metode *Cosine Similarity* sendiri menjadi salah satu metode yang baik digunakan untuk mencari kemiripan antar teks yang dibuktikan dari hasil penelitian (Ferio dkk., 2019) tentang “*Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan Metode User Based Collaborative Filtering Berbasis Metode Adjusted Cosine Similarity*” yang mana dalam penelitiannya mendapatkan hasil bahwa metode *Cosine Similarity* sangat baik dipergunakan pada data yang tidak terdapat nilai yang bersifat subjektif seperti *similarity* antara teks berdasarkan kemiripan kata dalam teks. Dari hal tersebut penulis mendapat keyakinan bahwasanya metode *Cosine Similarity* merupakan salah satu metode yang cocok digunakan pada penelitian kali ini.

Dari ketiga hasil penelitian diatas penulis dapat mengambil kesimpulan bahwa metode *Natural Language Processing* dapat dikombinasikan dengan metode *Cosine Similarity* dengan ini penulis mencoba membuat sistem untuk mengambil data dari CV yang nantinya diolah untuk bisa memberikan rekomendasi lowongan pekerjaan yang relevan dengan CV sesuai dengan keahlian yang dimiliki. Dalam sistem yang akan dibuat metode *Natural Language Processing* memiliki peran untuk mengolah data mentah yang didapatkan dari dokumen CV dan data mentah dari proses *Web Scraping* dengan melakukan beberapa pengolahan data kalimat seperti *case folding*, *stemming*, *filtering*, *tokenizing*, dan beberapa pengolahan lain yang mungkin dibutuhkan agar hasil pengujian memiliki akurasi yang tinggi. Untuk metode *Cosine Similarity* sendiri memiliki peran untuk membandingkan dua buah objek data untuk mencari tingkat kemiripan dari dua buah data yang disediakan, yang memiliki kemiripan tertinggi akan dijadikan rekomendasi lowongan pekerjaan.

2.2 Dasar Teori

2.2.1 *Curriculum Vitae* (CV)

Dalam dunia kerja ada beberapa persyaratan dokumen yang harus dipenuhi untuk bisa mendaftarkan diri atau bergabung ke perusahaan. Dokumen ini biasanya dapat membuat pihak perusahaan tertarik pada lamaran yang diajukan. Salah satu dokumen yang umum menjadi persyaratan bagi pelamar adalah *Curriculum Vitae* (CV).

Curriculum Vitae (CV) merupakan dokumen yang memuat data diri, riwayat pendidikan, pengalaman kerja, yang mencakup beberapa informasi lain yang relevan dengan pekerjaan yang akan diambil. Salah satu tahap seleksi untuk karyawan baru mengacu pada relasi CV, apakah CV pelamar mencakup semua hal yang dibutuhkan oleh perusahaan atau tidak. CV itu seperti pintu utama untuk mendapatkan pekerjaan yang dilamar. Dari CV, HRD (*Human Resources Development*) dapat menilai secara sekilas kualitas dari si pelamar. Dari hasil penelitian (Dewi dkk., 2021) tentang “*Pelaksanaan Rekrutmen, Seleksi dan Penempatan Kerja Karyawan (Studi pada Karyawan PT Gemala Kempa Daya)*” menjelaskan bahwa pelamar banyak gugur pada tahap seleksi berkas yang mana berkas ini berisi salah satunya yaitu CV. Oleh karena itu, dalam penelitian kali ini CV merupakan salah satu sumber data yang dipakai penulis dalam menentukan rekomendasi lowongan pekerjaan dengan mempertimbangkan data CV dengan data lowongan pekerjaan bisa memberikan rekomendasi pekerjaan yang cocok bagi pelamar dengan begitu diharap dapat meningkatkan rasio diterimanya pelamar pada sebuah perusahaan.

2.2.2 Glint

Dalam penelitian ini juga menggunakan sumber data yang semuanya didapat melalui salah satu proses *text mining* yaitu *web scraping* pada situs glint. Glint sendiri merupakan situs portal pencarian lowongan pekerjaan yang menjadi salah satu pusat perusahaan meng-*apply* atau mempromosikan lowongan pekerjaan yang ada pada perusahaan mereka. Alasan penulis mengambil situs portal pencarian lowongan pekerjaan dari glint karena glint sendiri adalah salah satu portal pencarian

lowongan pekerjaan yang mudah untuk diakses, tidak perlu melalui proses *login* untuk bisa mengakses tampilan lowongan pekerjaan berbeda dengan salah satu situs lowongan pekerjaan seperti *glassdoor* yang mana dalam situs ini untuk bisa mengakses tampilan lowongan pekerjaan diharuskan melalui proses *login*. Ditambah menurut penulis *website* *glint* memiliki *user experiences* yang baik bagi *user* pemula untuk mengakses halaman lowongan pekerjaan pada *website* ini serta arsitektur dari *website* *glint* tidak terlalu kompleks sehingga lebih mudah dilakukan proses *web scraping*. Ditambah dalam *website* *glints* menyediakan data lowongan pekerjaan yang dibutuhkan dalam proses rekomendasi seperti nama perusahaan, kategori lowongan pekerjaan, *skill* yang dibutuhkan, dan deskripsi dari pekerjaan.

2.2.3 Text Mining

Text Mining merupakan sebuah pengembangan dari ilmu pengetahuan yang bertujuan untuk mendapatkan informasi asli (teks asli) (Ivory, 2004). Dapat diartikan bahwa kegiatan untuk mengambil atau mengekstrak suatu hal seperti dokumen atau teks untuk mendapatkan informasi yang nantinya informasi tersebut digunakan untuk tujuan tertentu. Dari bahasanya yaitu *text mining* dapat diartikan dengan menambang teks yang berarti *text mining* bisa juga diartikan dengan penambangan data teks yang bertujuan untuk menemukan data pengetahuan tekstural *database*, yang mengacu pada proses ekstraksi pola data yang diambil dari dokumen teks (Tan, 1999). Dalam dunia analisis *big data*, *text mining* merupakan alat yang ampuh dalam memanfaatkan data tekstural yang tidak terstruktur dengan melakukan analisis dan mengekstrak pengetahuan-pengetahuan baru yang nantinya diidentifikasi pola yang signifikan dan korelasi yang signifikan dalam data (Hassani dkk., 2020).

Dalam melakukan *text mining* terdapat beberapa tahapan yang harus dilakukan untuk memaksimalkan hasil kemiripan dokumen, berikut merupakan beberapa tahapan yang dibutuhkan :

1. *Case Folding* yaitu tahapan untuk merubah semua huruf yang ada pada data menjadi huruf kecil, dalam hal ini semua huruf, dari huruf “a” sampai huruf “z” (Nugraha & Budiyanto, 2022)..

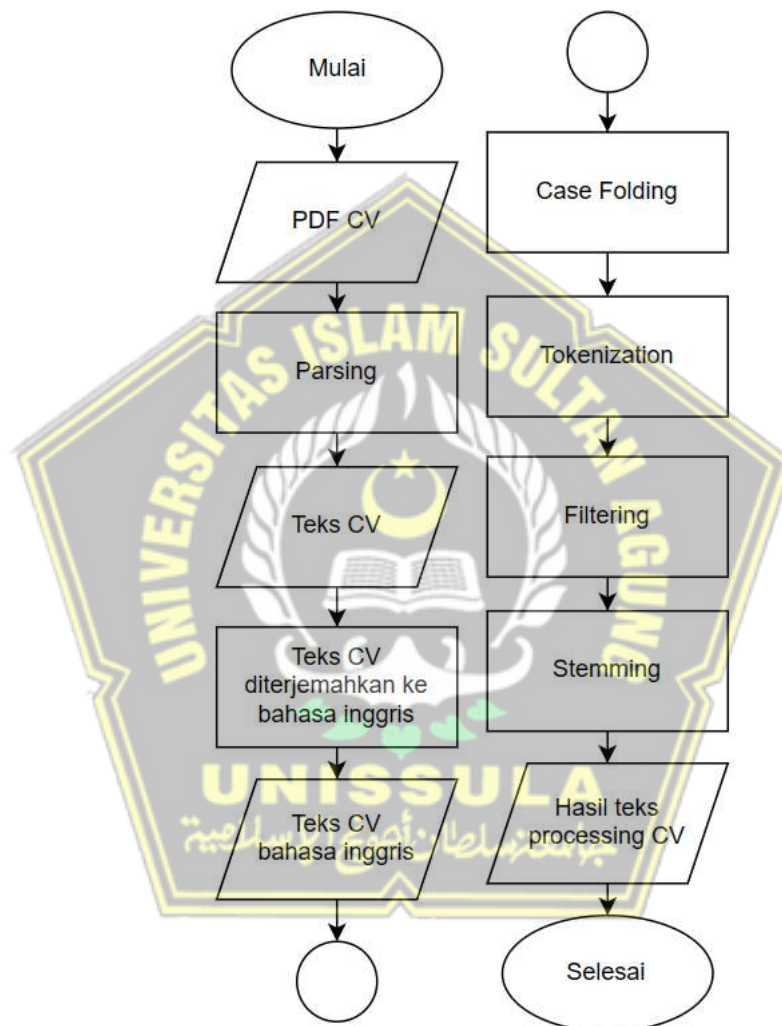
2. *Tokenizing* yaitu tahapan penggalan data teks untuk memecah kalimat menjadi kata tunggal atau frasa (*parsing*). *Tokenizing* bisa diartikan juga proses pemecahan kalimat – kalimat menjadi kata atau biasa disebut dengan token.
3. *Filtering* yaitu tahapan penggalan data teks untuk mengambil kata penting dari hasil proses token. Bisa dengan menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist* atau *stopword* adalah kata – kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-word*. Dalam *Filtering* sendiri daftar *stopword* yang menggunakan bahasa Indonesia contohnya “yang”, “dan”, “di”, “dari”, dan seterusnya. Untuk *wordlist* adalah kata – kata yang dianggap sebagai kata kunci (Sari dkk., 2018).
4. *Stemming* yaitu tahapan penggalan data teks untuk memperkecil jumlah indeks dari satu data hingga sebuah kata memiliki *suffix* maupun *prefix* akan kembali ke bentuk dasarnya. Dapat diartikan bahwa proses *Stemming* adalah proses untuk mengubah kata dari hasil *stopword* menjadi bentuk kata dasar (*root word*).

2.2.4 Text Preprocessing

Proses *text mining* diperlukan karena struktur dokumen teks yang tidak teratur membuat proses pengambilan data memakan waktu lama. Langkah-langkah ini pada dasarnya dimaksudkan untuk membuat teks yang dipersiapkan dapat diubah agar lebih terstruktur dan teratur. Dalam persiapan dokumen teks untuk proses *text preprocessing* merupakan sebuah tahapan proses yang sangat penting dalam menentukan kualitas tahap selanjutnya (Rahayu & RMS, 2018). Tujuan dilakukannya *text preprocessing* adalah untuk mempersiapkan dokumen agar menjadi data yang terstruktur sehingga dapat dimanfaatkan secara maksimal. *Text preprocessing* merupakan sebuah tahapan awal dari proses dokumen teks guna mempersiapkan teks menjadi data yang siap untuk diolah guna proses yang lebih lanjut (Feldman & Sanger, 2007).

Text preprocessing sendiri merupakan tahapan awal dalam mengolah dokumen sebelum teks bisa digunakan ke tahap selanjutnya, dimana tahapan proses seleksi data yang kemudian dilakukan pada setiap data terkait (Nugroho, 2011).

Pada gambar 2.1 merupakan langkah – langkah dalam *text preprocessing* yang direpresentasikan dalam sebuah *flowchart*.



Gambar 2. 1 Flowchart Alur Text Preprocessing

Pada gambar 2. 1 merupakan *flowchart* alur *text preprocessing* dimana tahapannya adalah :

1. Ekstraksi data pdf (*Parsing* PDF) : *Parsing* umumnya dikenal sebagai *syntactic analysis* adalah proses menganalisis teks yang terdiri dari urutan token. Dokumen sendiri dapat terdiri dari beberapa bagian, seperti bahasa dan format tertentu. Dan fungsi *parsing* menguraikan sekumpulan dokumen menjadi

masing-masing komponen. Pada tahap ini, *parsing* mengambil data dari dokumen PDF dan mengubahnya menjadi format teks.

2. *Translate to English* : Dari hasil *parsing* atau ekstraksi data PDF akan dilakukan kesetaraan bahasa menjadi bahasa Inggris yang merupakan bahasa Internasional. Alasan penulis melakukan *translate* ke bahasa Inggris karena CV mayoritas CV menggunakan bahasa Inggris serta data lowongan pekerjaan mayoritas menggunakan bahasa Inggris dalam *field* data yang penting seperti *skill* yang dibutuhkan, kategori pekerjaan, nama pekerjaan yang ditawarkan, serta deskripsi pekerjaan mayoritas menggunakan bahasa Inggris.
3. *Case Folding* : *Case Folding* sendiri merupakan salah satu langkah dalam urutan *text preprocessing* tujuan dari *Case Folding* sendiri adalah untuk mengubah semua huruf dalam hasil *parsing* dokumen PDF menjadi huruf kecil dimana hanya huruf 'a' sampai 'z' yang akan dirubah ke huruf kecil. Dalam *Case Folding* juga dilakukan proses penghapusan angka, penghapusan angka ini dilakukan jika angka tersebut tidak berhubungan dengan objek yang dianalisis. Dalam hal ini misalnya penomoran nomor rumah, nomor telepon, *handphone*, dll. Dan juga dilakukan penghapusan tanda baca. Penghapusan tanda baca dilakukan dengan menghilangkan karakter seperti [!'"#\$%&'()*+,-./:;<=>?@[\\]^_`{|}~]. Langkah terakhir dalam kasus *Case Folding* adalah menghilangkan *whitespace*.
4. *Tokenizing* : *Tokenizing* digunakan untuk memecah kalimat menjadi kata tunggal atau frasa (*parsing*). *Tokenizing* bisa diartikan juga proses pemecahan kalimat – kalimat menjadi kata atau biasa disebut dengan token. Sebagai contoh ketika ada kalimat “Saya pergi ke pasar” dari kalimat tersebut akan dipecah menjadi token atau kata sehingga akan menjadi “Saya”, “Pergi”, “Ke”, “Pasar”, biasanya acuan dari pemecahan token adalah spasi.
5. *Filtering* : *Filtering* digunakan untuk mengambil kata penting dari hasil proses token. Bisa dengan menggunakan algoritma *stoplist* (membuang kata yang kurang penting). *Stoplist* atau *stopword* adalah kata – kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-word*. Dalam *Filtering* sendiri daftar *stopword* yang menggunakan bahasa Inggris antara lain 'i', 'me', 'my',

'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're" dan seterusnya. Dalam *Filtering* dapat menambah *stopword* sesuai yang dibutuhkan.

6. *Stemming* : *Stemming* digunakan untuk memperkecil jumlah indeks dari satu data hingga sebuah kata memiliki *suffix* maupun *prefix* akan kembali ke bentuk dasarnya. Dapat diartikan bahwa proses *Stemming* adalah proses untuk mengubah kata dari hasil *stopword* menjadi bentuk kata dasar (root word). Sebagai contoh *Stemming* antara lain kata “melihat”, “memperlihatkan”, “dilihat”, “dilihatkan” akan ditransformasi menjadi kata “lihat”.

2.2.5 Web Scraping

Web Scraping merupakan proses pengambilan data atau dokumen semi-terstruktur dari internet, umumnya berupa halaman-halaman *Web* dalam bahasa markup seperti HTML (*HyperText Markup Language*) atau XHTML (*Extensible HyperText Markup Language*), dan menganalisis dokumen tersebut untuk diambil data tertentu dari halaman target yang nantinya akan digunakan bagi kepentingan lain (Turland, 2010).

2.2.6 Cosine Similarity

Cosine Similarity digunakan untuk menghitung nilai kemiripan antar kalimat dan menjadi salah satu teknik untuk mengukur kemiripan antar teks. Metode *Cosine Similarity* sendiri memiliki kelebihan tidak terpengaruh pada panjang pendeknya suatu dokumen dan memiliki tingkat akurasi yang tinggi (Riyani dkk., 2019). Cara metode *Cosine Similarity* mengukur kemiripan dokumen atau teks melalui perhitungan nilai sudut *cosinus* dari perkalian dua vektor yang dibandingkan (Lahitani, 2022). *Cosine Similarity* menggunakan perhitungan angka dari 0 sampai 1. Apabila hasil pengujian kemiripan mendekati angka 0 kemungkinan besar objek yang dibandingkan tidak mirip namun jika hasil pengujian mendekati angka 1 kemungkinan besar objek yang dibandingkan memiliki kemiripan.

Berikut merupakan rumus yang digunakan oleh *Cosine Similarity* :

$$\text{Cos } a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (1)$$

Keterangan :

A = Vektor A, yang akan dibandingkan kemiripan

B = Vektor B, yang akan dibandingkan kemiripan

$A.B$ = Dot product antara vektor A dan vektor B

$|A|$ = Panjang vektor A

$|B|$ = Panjang vektor B

$|A||B|$ = Cross product antara $|A|$ dan $|B|$

2.2.7 *Natural Language Processing*

Natural Language Processing (NLP) adalah salah satu cabang ilmu dari AI (*Artificial Intelligence*) yang berfokus pada pengolahan bahasa alami. Yang mana bahasa alami adalah bahasa yang umum digunakan manusia untuk berkomunikasi. Agar komputer bisa memahami bahasa manusia, komputer harus mengolah terlebih dahulu bahasa yang diterima agar bisa dipahami dengan baik oleh komputer (Sembilu dkk., 2018). *Natural Language Processing* sendiri memiliki beberapa hal yang dibahas didalamnya salah satunya yaitu *parsing*. *Parsing* adalah analisis gramatikal dari sebuah kalimat dengan melakukan pemecahan kalimat menjadi beberapa *part-of-speech*. Analisis ini berguna untuk tugas pemrosesan *downstream* yang lebih kompleks.

2.2.8 *Flask*

Flask merupakan sebuah *microframework* yang dikembangkan oleh Armin Ronacher. Tujuan inti dari *Flask* adalah untuk menyederhanakan *framework* sedemikian rupa sehingga menjadi sangat ringan dan cepat. Dengan menggunakan tagline "*web development, one drop at a time*". *Flask* memungkinkan pengguna untuk membuat situs *web* dengan cepat dengan menggunakan *library* yang sederhana (Putra & Putera, 2019).

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Dalam penelitian ini, metode atau algoritma yang digunakan adalah *Cosine Similarity* dan *Natural Language Processing*, dimana dengan mengkombinasikan kedua metode ini dengan metode *Natural Language Processing* sebagai bagian dari *text preprocessing* dan metode *Cosine Similarity* sebagai penghitungan *similarity* antara data CV dan data lowongan pekerjaan dapat menghasilkan sistem rekomendasi lowongan pekerjaan. Adapun tahapan yang harus dilakukan dalam penelitian ini, antara lain:

3.1.1 Study Literature

Peneliti akan meninjau beberapa e-book, artikel, jurnal, hasil penelitian terdahulu seperti tesis dan skripsi, serta berbagai situs *website* untuk bisa mempelajari teori dan berlatih mengenai *web scarping*, *NLP parsing*, metode *Cosine Similarity*, penerapan *framework flask*, dan karakteristik CV

3.1.2 Data Collecting

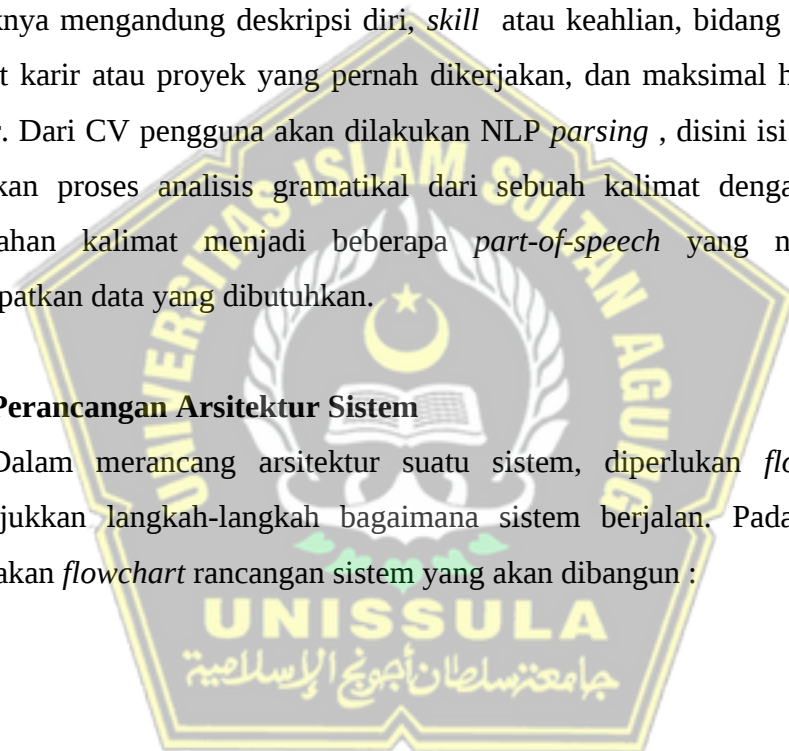
Pertama akan dilakukan *data collecting* atau pengumpulan data. Data yang dibutuhkan yaitu data lowongan pekerjaan dan data dari CV pengguna. Untuk mendapatkan data lowongan pekerjaan akan dilakukan proses *Web Scraping* pada sebuah situs penyedia lowongan pekerjaan yaitu glints (<https://glints.com/id>). Data lowongan pekerjaan akan diambil berdasarkan kategori pekerjaan. Dari glints sendiri terdapat 14 kategori lowongan pekerjaan yaitu, *software*, *data science*, *design*, *operations*, *marketing*, *product management*, *bussiness*, *enggineering*, *management*, *finance*, *Human Resources Development (HRD)*, *media* dan *komunikasi*, *konsultan*, dan kategori lainnya. Dan dari setiap kategori akan diambil kurang lebih 30 data lowongan pekerjaan dengan rincian nama pekerjaan, nama perusahaan, kategori pekerjaan, *skill* atau keahlian yang dibutuhkan, deskripsi dari pekerjaan, dan alamat perusahaan. Sehingga jumlah data yang digunakan kurang

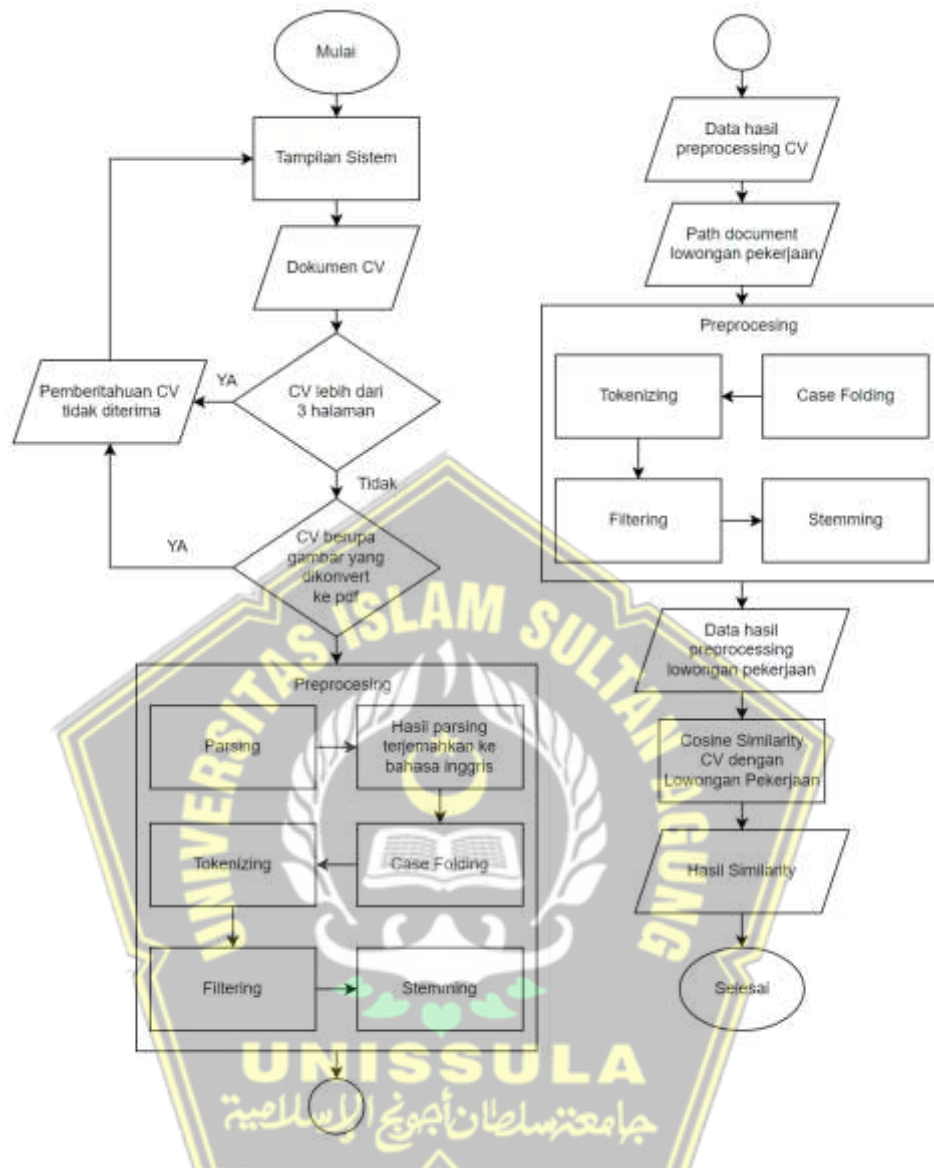
lebih sekitar 420 data lowongan pekerjaan. Sebelum data lowongan pekerjaan digunakan akan dilakukan penggabungan isi *field* yang nantinya menjadi teks yang akan digunakan dalam penelitian ini. Untuk data *field* yang digabung adalah data nama pekerjaan, *skill* atau keahlian yang dibutuhkan, deskripsi pekerjaan, dan alamat perusahaan.

Untuk mendapatkan data yang kedua yaitu data dari CV. Pengguna bisa melakukan *upload file* CV dengan ekstensi PDF dimana CV tidak berupa gambar yang di-*convert* menjadi PDF, teks didalamnya dapat dilakukan proses *copy paste*, setidaknya mengandung deskripsi diri, *skill* atau keahlian, bidang yang digeluti, riwayat karir atau proyek yang pernah dikerjakan, dan maksimal halaman CV 3 lembar. Dari CV pengguna akan dilakukan NLP *parsing*, disini isi dari CV akan dilakukan proses analisis gramatikal dari sebuah kalimat dengan melakukan pemecahan kalimat menjadi beberapa *part-of-speech* yang nantinya akan mendapatkan data yang dibutuhkan.

3.1.3 Perancangan Arsitektur Sistem

Dalam merancang arsitektur suatu sistem, diperlukan *flowchart* yang menunjukkan langkah-langkah bagaimana sistem berjalan. Pada gambar 3.1 merupakan *flowchart* rancangan sistem yang akan dibangun :





Gambar 3. 1 Flowchart Rancangan Arsitektur Sistem

Gambar 3.1 merupakan *flowchart* rancangan sistem yang akan dibangun dimana tahapannya adalah:

1. Ekstraksi data pdf (*Parsing* PDF) : Pada tahap ini, *file* PDF CV yang diterima akan diekstraksi menjadi teks. Contoh penulis melakukan *upload file* CV.pdf dengan isi didalamnya kalimat “1. Memiliki pengalaman dibidang web development!”, ketika dilakukan *parsing* nilai return tidak lagi berupa *file* CV.pdf namun sudah dalam bentuk teks yang berisi kalimat yang ada didalam *file* CV.pdf yaitu “1. Memiliki pengalaman dibidang web development!”.

2. *Translate to English* : Tahap yang kedua adalah melakukan penerjemah dari teks yang sudah diparsing kedalam bahasa Inggris. Contoh dari hasil tahap pertama berhasil didapatkan data teks dari PDF yang berisikan “1. Memiliki pengalaman dibidang web development!” jika melalui tahap ini teks tersebut akan berubah menjadi “1. Have experience in the field of web development!”.
3. *Case Folding* : Tahap yang ketiga adalah *Case Folding* dimana pada tahap ini akan dilakukan proses membuat teks menjadi huruf kecil, pengapusan *whitespace*, penghapusan angka, dan penghapusan tanda baca. Sebagai contoh dari hasil tahap ke dua menghasilkan teks berisi “1. Have experience in the field of web development!”, setelah teks melewati tahap ini maka sekarang teks berubah menjadi “have experience in the field of web development”.
4. *Tokenizing* : Tahap yang keempat adalah *Tokenizing* dimana pada tahap ini akan dilakukan pemecahan kalimat menjadi per kata atau token. Sebagai contoh dari hasil tahap tiga sekarang teks berisi “have experience in the field of web development”, setelah teks melewati tahap ini maka sekarang teks berubah menjadi perkata sehingga isi teks berubah menjadi “have”, “experience”, “in”, “the”, “field”, “of”, “web”, “development”.
5. *Filtering* : Tahap yang kelima adalah *Filtering* dimana pada tahap ini akan dilakukan pengambilan kata penting dengan menghilangkan stopword. Sebagai contoh pada tahap keempat teks berisi “have”, “experience”, “in”, “the”, “field”, “of”, “web”, “development”, setelah teks melewati tahap ini maka sekarang teks berubah menjadi “experience”, “field”, “web”, “development”.
6. *Stemming* : Tahap yang terakhir adalah *Stemming* dimana pada tahap ini akan dilakukan perubahan kata dari hasil tahap lima menjadi bentuk kata dasar (*root word*). Sebagai contoh pada tahap lima teks berisi “experience”, “field”, “web”, “development”, setelah teks melewati tahap ini maka sekarang teks berubah menjadi “experi”, “field”, “web”, “develop”.

Setelah hasil *parsing* data CV melalui *text preprocessing* tahapan diatas akan dilakukan juga pada data lowongan pekerjaan. Jika kedua data tersebut sudah melalui *text preprocessing* kedua data tersebut akan melalui *cosine similarity* untuk mendapatkan nilai kemiripan antara keduanya.

3.1.4 Gambaran Sistem

Pada penelitian kali ini penulis akan mengembangkan sistem berbasis *web application* untuk memberikan rekomendasi lowongan pekerjaan yang relevan atau memiliki kemiripan dengan CV yang dikirim oleh pengguna. Untuk prosesnya, pengguna bisa mengirimkan *file* CV dengan format pdf yang nantinya oleh sistem *file* tersebut akan dilakukan *parsing* data yang ada pada CV kemudian hasil data tersebut akan melalui proses *text preprocessing* seperti menerjemahkan data menjadi bahasa Inggris, *case folding*, *tokenizing*, *filtering*, dan *stemming* agar data CV yang didapat memiliki kualitas yang baik. Untuk data lowongan pekerjaan sendiri didapatkan melalui proses *web scraping* pada salah satu *website* penyedia lowongan pekerjaan yaitu glints (<https://glints.com/id>). Data lowongan pekerjaan yang diambil dari *website* glints juga melalui proses *text preprocessing* agar data yang didapatkan memiliki kualitas yang baik. Dari kedua data tersebut nantinya akan dilakukan uji kemiripan menggunakan metode *Cosine Similarity*. Data lowongan pekerjaan yang memiliki tingkat kemiripan yang tinggi dengan data CV akan ditampilkan kepada pengguna sebagai lowongan pekerjaan yang direkomendasikan oleh sistem.

3.1.5 Identifikasi Perangkat Lunak

Pada tahap analisis kebutuhan, peneliti menganalisis apa saja perangkat lunak yang dibutuhkan untuk mengembangkan aplikasi ini sehingga proses *input* hingga mengeluarkan hasil dari rekomendasi yang dapat berjalan sesuai dengan yang diinginkan. Adapun perangkat lunak yang digunakan untuk mengembangkan sistem ini adalah:

1. Python 3.11.0

Bahasa pemrograman yang digunakan dalam penelitian ini adalah Bahasa Python dengan versi 3.11.0 digunakan dikarenakan Bahasa ini memiliki *library* yang lengkap dan *open source*.

2. Library String

Library string python merupakan *library* yang digunakan sebagai pengolahan kalimat atau teks, dalam penelitian ini *library string* python digunakan untuk salah

satu proses *text mining* yaitu *Case Folding* dengan membuat teks menjadi huruf kecil dengan function `lower()` dan juga menghapus tanda baca seperti `!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~.`

3. Library NLTK (*Natural Language Tool Kit*)

Library NLTK (Natural Language Tool Kit) merupakan *library* yang digunakan untuk pengolahan data bahasa manusia. *Library NLTK* sendiri adalah *library* yang digunakan untuk proses NLP (*Natural Language Processing*). *Library NLTK* digunakan dalam penelitian ini untuk melakukan proses *text mining* yaitu *tokenizing*, *filtering*, dan *stemming*.

4. Library Googletrans Versi 3.1.0

Library Googletrans merupakan *library* yang digunakan untuk menerjemahkan teks sesuai bahasa yang diinginkan layaknya Google Translate. Dalam penelitian ini *library Googletrans* digunakan untuk menerjemahkan data *parsing* dari CV dan data gabungan dari lowongan pekerjaan menjadi bahasa Inggris.

5. Library Pdfminer Versi 20181108

Library Pdfminer python merupakan *library* yang digunakan untuk mengekstrak atau mem-*parsing* dokumen PDF ke dalam bentuk teks. Dalam penelitian ini *library Pdfminer* digunakan untuk mem-*parsing* dokumen CV dari pengguna.

6. Library Pandas

Library Pandas python merupakan *library* yang digunakan untuk memproses data, mulai pembersihan data, manipulasi data, hingga melakukan analisis data. Dalam penelitian ini *library Pandas* digunakan untuk membuat laporan data pada saat testing dalam *google colab* dan juga digunakan dalam pengolahan data menjadi data json yang nantinya akan ditampilkan dalam tampilan *website*.

7. Library Scikit-Learn

Library Scikit-Learn python merupakan *library* yang difokuskan dalam *machine learning*. Dalam penelitian ini *library Scikit-Learn* digunakan untuk melakukan *CountVectorizer* yaitu menghitung *frekuensi* kata dalam dokumen atau teks. Dengan *CountVectorizer* dapat mengubah teks menjadi sebuah representasi vector ini digunakan untuk melakukan *CountVectorizer* antara teks data CV dengan teks data lowongan pekerjaan. Dengan *library Scikit-Learn* juga digunakan untuk melakukan penghitungan *Cosine similarity* dari hasil *CountVectorizer*. Pada tabel 3. 1 merupakan contoh dari *CountVectorizer* :

Data = ["Saya lebih rajin sedangkan dia lebih pintar"]

Tabel 3. 1 Contoh hasil *CountVectorizer*

Data	Lebih	Saya	Rajin	Sedangkan	Dia	Pintar
Frekuensi	2	1	1	1	1	1

8. Library Re (RegEx atau Ekspresi Reguler)

Library Re python merupakan *library* yang digunakan untuk membentuk pola pencarian urutan karakter dalam artian digunakan untuk memeriksa apakah *string* berisi pola pencarian yang ditentukan. Dalam penelitian ini *re* digunakan untuk salah satu proses *text mining* yaitu *Case Folding* dengan menghapus angka dari 0 sampai 9.

9. Visual Code Studio

Visual Studio Code dipilih sebagai *text editor* pada pengembangan aplikasi dalam penelitian ini, *Visual Studio Code* dipilih dikarenakan mendukung banyak bahasa pemrograman dan *framework*, multi platform, performa yang sangat cepat, mempunyai banyak *extensions* yang dapat mempermudah proses pengembangan *website*.

10. Framework Flask

Framework Flask dipilih sebagai *framework* dalam pengembangan *website* dikarenakan dengan menggunakan *flask* penulis bisa membuat *website* yang menggunakan bahasa python sehingga akan lebih mudah dan praktis karena *core* dari sistem yang akan dikembangkan sendiri berbahasa python.

3.1.6 Perancangan *User Interface*

Berikut ini merupakan rancangan/ *design system* yang digunakan pada penelitian ini :

1. Halaman utama

Pada gambar 3.2 merupakan perancangan antar muka untuk halaman utama dimana halaman utama adalah halaman awal yang akan dilihat oleh pengguna.

NO	Nama Pekerjaan	Nama Perusahaan	Kategori Pekerjaan	Keahlian	Deskripsi Pekerjaan	Similarity

Gambar 3. 2 Rancangan antar muka halaman utama

Gambar 3. 2 merupakan gambaran mengenai rancangan antar muka untuk halaman utama, dimana halaman ini adalah halaman pertama yang akan dilihat oleh pengguna. Pada halaman ini pengguna akan melihat *header* bertuliskan “Rekomendasi Lowongan Pekerjaan” dengan *icon* perusahaan disamping kiri. Pengguna juga melihat *form upload* untuk *upload* PDF CV nantinya. Kemudian, akan ada satu buah tabel yang berisikan *field* nama pekerjaan, nama perusahaan, kategori pekerjaan, keahlian, deskripsi pekerjaan, dan nilai *similarity* dalam tabel ini akan menampung data rekomendasi lowongan pekerjaan yang diberikan oleh sistem.

2. Halaman rekomendasi

Pada gambar 3.3 merupakan rancangan halaman antar muka untuk halaman rekomendasi.

N	Nama Pekerja	Nama Perusahaan	Alamat Perusahaan	Kategori Pekerjaan	Keahlian	Deskripsi Pekerjaan	Similari
1	Web Developer	Confidential Company Powered by Glints	Jakarta	Software Engineering	Skills API Web Developer S	Deskripsi Pekerjaan Web Developer - Handling web API development Pekerjaan Web Developer - D3/S1 all major - 2 years of minimum experience using Node.JS. Get used to handling web API development. Proficient in back end development. Lihat Lebih Banyak	42.16 %

Gambar 3. 3 Rancangan antar muka halaman rekomendasi

Setelah pengguna meng-*upload* PDF CV di halaman utama pengguna akan diarahkan ke halaman rekomendasi dimana seperti yang diperlihatkan gambar 3.3 halaman ini dibuat untuk menerima hasil dari rekomendasi lowongan pekerjaan yang disarankan oleh sistem dan pemberian notifikasi bahwa CV berhasil diterima.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Perhitungan

Di dalam perhitungan untuk memperoleh hasil kemiripan pada suatu CV dengan data lowongan pekerjaan menggunakan metode *cosine similiarity*. Dimana nilai yang dihasilkan digunakan sebagai bahan acuan rekomendasi lowongan pekerjaan.

4.2 Contoh Dokumen CV

Tabel 4. 1 merupakan sebagian objek yang diambil untuk dijadikan contoh data penelitian, terdapat 6 buah contoh dokumen yang diambil berikut rincian contoh CV yang digunakan sebagai pembanding.

Tabel 4. 1 Rincian Contoh Dokumen CV

Format CV	Isi
CV dengan format <i>Applicant Tracking System</i> (ATS) 1 halaman	

Format CV	Isi
CV dengan format Applicant Tracking System (ATS) 2 halaman	Nur Muhammad Syaifuddin 0815 814 4365-6209 nurmuhammad01@gmail.com https://www.linkedin.com/in/nurmuhammad01/ https://github.com/NurMuhammad01 I graduated from IS Informatics Engineering, Sultan Ageng Tjema University, Semarang. My main skills currently are developing Android app using the Kotlin programming language, cross mobile platforms with Flutter, and building Android APIs using Springboot EDUCATION Sultan Ageng Tjema University Bachelor of Informatics Engineering Semarang, ID October 2019-February 2023 EXPERIENCE QCamp Android Developer Expert Level Facilitator 2022 Remote October 2022-Present - Helping participants if they experience technical or non-technical problems in the "Learn Android Intermediate Application Development" class so that they graduate on time. QCamp Android Developer Intermediate Level Facilitator 2022 Remote October 2022-December 2022 - Helping participants if they experience technical or non-technical problems in the "Learning Android Application Fundamentals" class so that they graduate on time. QTS PKDA Android Developer Intermediate Batch 3 Facilitator 2022 Remote Currently a Coaching August 2022-November 2022 - Helping participants if they experience technical or non-technical problems in the "Learn Android Intermediate Application Development" class so that they graduate on time. QTS PKDA Android Developer Batch 2 Facilitator 2022 Remote Currently a Coaching April 2022-August 2022 - Helping participants if they experience technical or non-technical problems in the "Learning Android Application Fundamentals" class so that they graduate on time. Trainer at Android Workshop Informatics Engineering, Sultan Ageng Tjema University Semarang, ID June 2022-July 2022 Teaching Materials: - Android Introduction - View, ViewGroup, Activity, Fragment - LiveData, RecyclerView - Capstone Project Discussion Based Coaching May 2022-July 2022 - Helping student participants on the Android developer learning path through "Discussion Based" on the coded platform with comments. Best Graduate on Android Developer Learning Path in QCamp 2021 Remote May 2021-May 2021 Study and graduate with the best rating in class: - Getting started with Kotlin Programming - Learn to Make Android Applications for Beginners - Learn Android Application Fundamentals - Learn Android Layouts - Become an Android Developer Expert SKILLS - Android Native (Java/Kotlin): Model-View-Controller, Dependency Injection, Clean Architecture, Reactive Programming, Concurrency Programming - Flutter: Model-View-Controller, Dependency Injection, Clean Architecture, Reactive Programming, Asynchronous Programming - Backend API with Spring Boot, MVC, JPA, PostgreSQL, Validation TOOLS - Android Studio - IntelliJ IDEA - GitLab - Postman LICENCES AND CERTIFICATES Please check my licenses and certificates in the portfolio link in the header section.

Format CV	Isi
CV dengan yang mengandung gambar didalamnya sebanyak 3 buah	

Keterangan : Untuk isi dari CV dapat dilihat pada lampiran

4.3 Implementasi Text Preprocessing

1. Parsing

Sebelum dokumen dapat diolah harus dilakukan *text preprocessing* terlebih dahulu. Pada gambar 4.1 menunjukan *source code* untuk melakukan *parsing* dokumen sehingga mendapatkan data berupa teks.

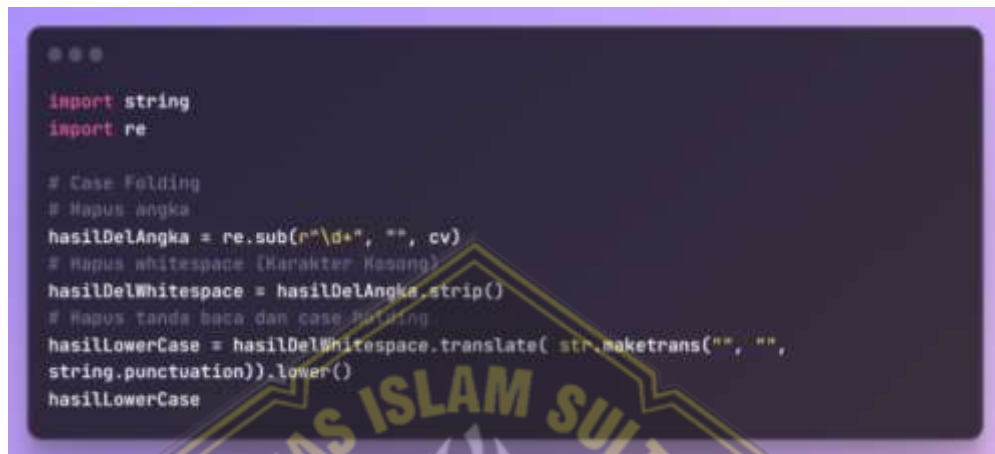


Gambar 4. 1 Parsing

Pada gambar 4.1 merupakan tampilan *source code* untuk *parsing* . Dimana pertama dilakukan *import* beberapa *function* yang dibutuhkan dari *library Pdfminer*. Selanjutnya membuat inisiasi objek dari *PDFResourceManager* yang digunakan untuk menyimpan *font* dan gambar. Selanjutnya melakukan inisiasi objek *StringIO* yang digunakan untuk menyimpan teks yang diekstraksi dari PDF. Kemudian menginisiasi objek dari *TextConverter* yang digunakan untuk mengonversi PDF menjadi teks. Untuk *PDFPageInterpreter* digunakan menginterpretasikan setiap halaman PDF dan mengekstrak teksnya, setiap teks yang diekstrak akan dimasukan pada *StringIO*. Setelah semua halaman habis perulangan akan berhenti dan mengembalikan hasil ekstrak teks sebagai *string*.

2. Case Folding

Setelah data teks didapatkan melalui proses *parsing* , selanjutnya data tersebut akan memasuki tahap *case folding*, pada gambar 4.2 merupakan *source code* dari *case folding*.



```
import string
import re

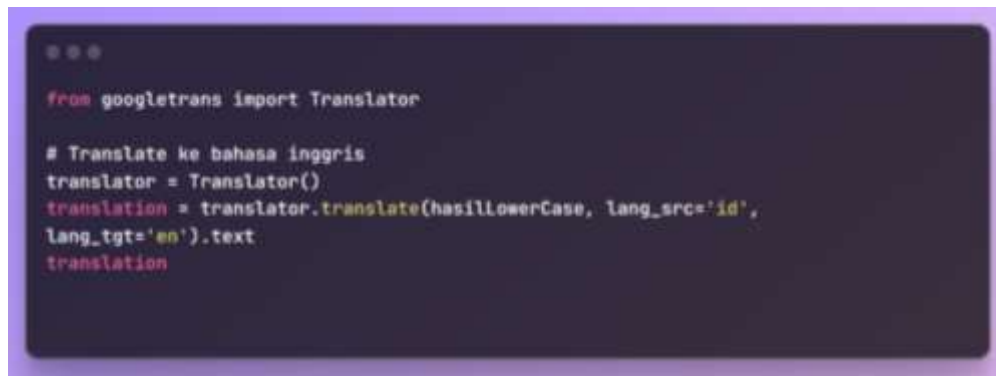
# Case Folding
# Hapus angka
hasilDelAngka = re.sub(r"\d+", "", cv)
# Hapus whitespace (Karakter Kosong)
hasilDelWhitespace = hasilDelAngka.strip()
# Hapus tanda baca dan case Folding
hasilLowerCase = hasilDelWhitespace.translate( str.maketrans("", "",
string.punctuation)).lower()
hasilLowerCase
```

Gambar 4. 2 Case Folding

Pada gambar 4.2 merupakan tampilan *source code* untuk *case folding*. Dimana pertama dilakukan *import* dua *library* yaitu *string* dan *re* (RegEx atau Ekspresi Reguler). Untuk proses *Case Folding* yang pertama adalah menghilangkan angka yang ada pada teks menggunakan *library re*. Dimana untuk *pattern* yang digunakan ada angka 0 sampai 9 dan +. Setelah menghapus angka selanjutnya yang akan dihapus adalah *whitespace* dengan perintah *.strip()*, untuk proses yang terakhir yaitu menghilangkan tanda baca dan membuat semua huruf menjadi huruf kecil disini penulis menggunakan *library string* untuk melakukan hal tersebut.

3. Translate to english

Setelah data teks melalui proses *Case Folding* otomatis data teks cukup bersih dari teks berupa angka, *whitespace*, tanda baca, dan data teks sudah berbentuk huruf kecil. Selanjutnya data tersebut akan memasuki tahap *translate to english*. Pada gambar 4.3 merupakan *source code* dari *translate to english*.



```

from googletrans import Translator

# Translate ke bahasa inggris
translator = Translator()
translation = translator.translate(hasilLowerCase, lang_src='id',
lang_tgt='en').text
translation

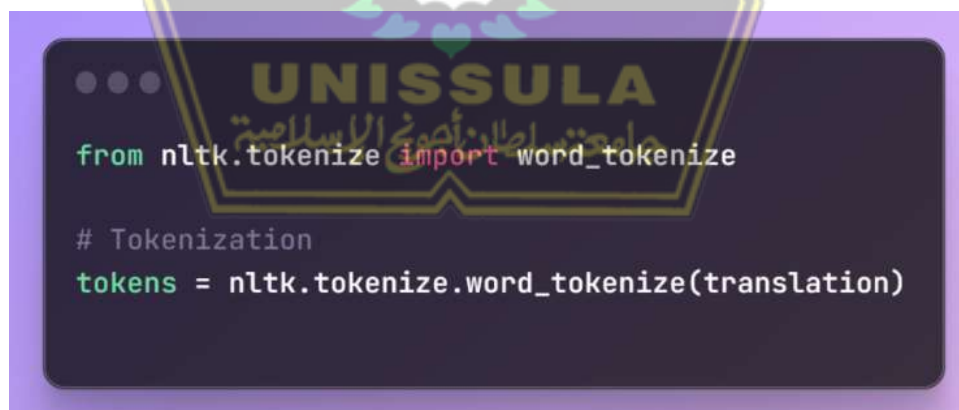
```

Gambar 4. 3 *Translate to english*

Pada gambar 4.3 merupakan tampilan *source code* untuk translate to english. Dimana pertama akan dilakukan *import function Translator* dari *library googletrans*. Selanjutnya melakukan inisiasi objek untuk *Translator()*. Disini penulis akan menerjemah teks dari hasil *case holding* dengan bahasa asal bahasa Indonesia *lang_src='id'* menjadi bahasa Inggris *lang_tgt='en'*.

4. Tokenization

Setelah data teks melalui proses *tranlaste to english* harusnya data teks sekarang sudah berupa kalimat berbahasa Inggris. Selanjutnya data teks tersebut akan memasuki tahap *tokenization*. Pada gambar 4.4 merupakan *source code* dari *tokenization*.



```

from nltk.tokenize import word_tokenize

# Tokenization
tokens = nltk.tokenize.word_tokenize(translation)

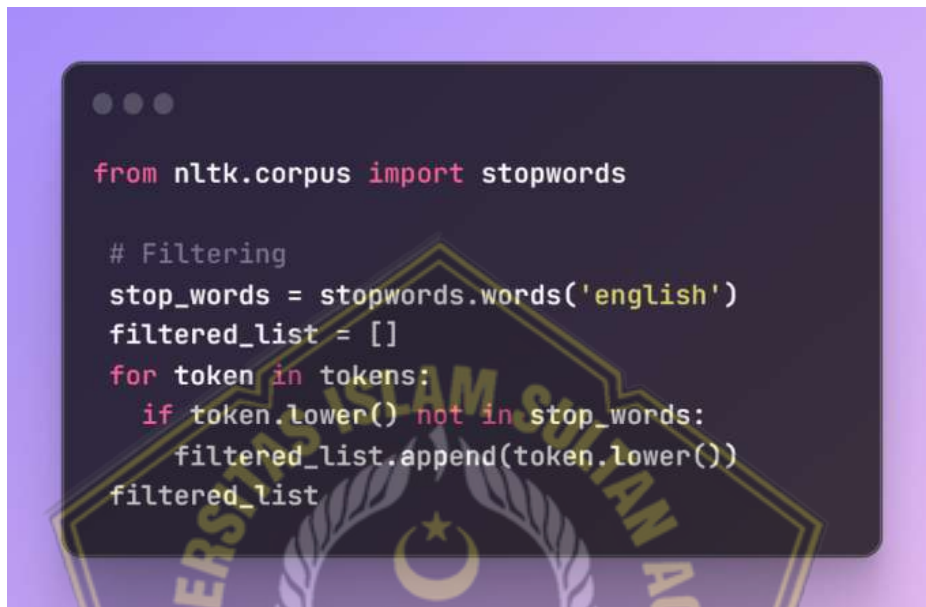
```

Gambar 4. 4 *Tokenization*

Pada gambar 4.4 merupakan tampilan *source code* untuk tokenization. Dimana pertama akan dilakukan *import function word_tokenize* dari *library nltk.tokenize*, disini penulis akan melakukan tokenisasi menjadi bentuk per kata. Dengan memasukan parameter berupa teks yang sudah melawati proses *Case Folding* dan *translate to english* ke dalam *function nltk.tokenize.word_tokenize()*

5. Filtering

Setelah data teks melalui proses *tokenization* harusnya data teks sekarang sudah pecah ke bentuk token atau kata. Selanjutnya data token tersebut akan memasuki tahap *filtering*. Pada gambar 4.5 merupakan *source code* dari *filtering*.



```
from nltk.corpus import stopwords

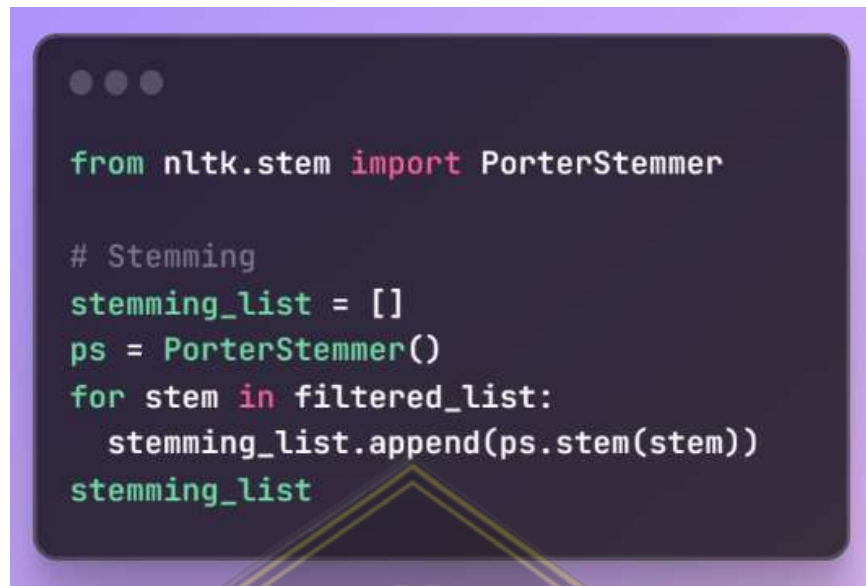
# Filtering
stop_words = stopwords.words('english')
filtered_list = []
for token in tokens:
    if token.lower() not in stop_words:
        filtered_list.append(token.lower())
filtered_list
```

Gambar 4. 5 Filtering

Pada gambar 4.5 merupakan tampilan *source code* untuk *filtering*. Dimana pertama akan dilakukan *import function stopwords* dari *library nltk.corpus* untuk mendapatkan data *stopwords* dalam bahasa Inggris. Selanjutnya data *stopwords* yang sudah didapatkan ditampung ke dalam sebuah variabel. Selanjutnya hasil dari token akan dibandingkan dengan list *stopword* apakah ada yang sama, jika token yang dibandingkan dengan list *stopword* sama maka token tersebut tidak akan dimasukan ke dalam *array*.

6. Stemming

Setelah data teks melalui proses *Filtering* data teks sekarang sudah bersih dari data *stopword*. Selanjutnya data teks tersebut akan memasuki tahap *stemming*. Pada gambar 4.6 merupakan *source code* dari *stemming*.



```

from nltk.stem import PorterStemmer

# Stemming
stemming_list = []
ps = PorterStemmer()
for stem in filtered_list:
    stemming_list.append(ps.stem(stem))
stemming_list

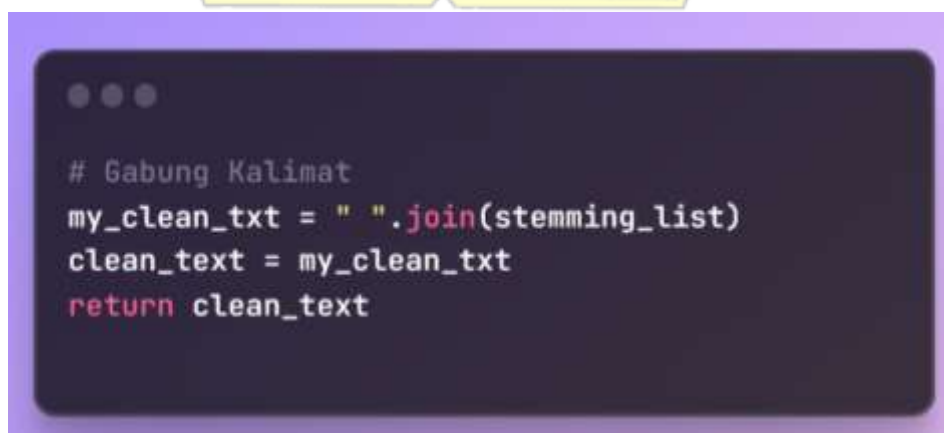
```

Gambar 4. 6 Stemming

Pada gambar 4.6 merupakan tampilan *source code* untuk *stemming*. Dimana pertama akan dilakukan *import function PorterStemmer* dari *library nltk.stem*. Dari hasil token yang lolos tahap *Filtering* token tersebut akan diambil kata dasarnya. Pertama akan dilakukan inisiasi objek untuk *PorterStemmer*, kemudian data token satu per satu akan melalui tahap *stemming*.

7. Penggabungan hasil *text preprocessing*

Setelah data teks melalui proses *text preprocessing* data yang masih berupa token akan dikembalikan ke bentuk kalimat untuk bisa digunakan untuk kebutuhan mencari *similarity*. Pada gambar 4.7 merupakan *source code* untuk menggabungkan token menjadi kalimat.



```

# Gabung Kalimat
my_clean_txt = " ".join(stemming_list)
clean_text = my_clean_txt
return clean_text

```

Gambar 4. 7 Penggabungan hasil *text preprocessing*

Pada gambar 4.7 merupakan tampilan *source code* dari menggabungkan hasil *text preprocessing* menjadi kembali ke bentuk kalimat dengan melakukan *join* dari *array stemming_list* dimana *array* ini berisi data teks hasil *Stemming* yang mana berarti berisi data teks yang sudah melalui proses *text preprocessing*.

4.4 Hasil Implementasi Sistem

Setelah mengumpulkan data contoh CV yang dibutuhkan sebagai bahan uji sistem, selanjutnya diimplementasikan di sistem. Berikut ini merupakan hasil tangkapan layar hasil implementasi sistem.

1. Halaman Utama



Gambar 4. 8 Halaman utama

Pada gambar 4.8 merupakan tampilan halaman utama yang mana tampilan halaman utama merupakan halaman yang akan ditampilkan pengguna saat pertama kali masuk. Halaman ini berisi judul tugas akhir, *form* untuk *upload file* PDF CV, dan tabel yang masih kosong, dimana tabel tersebut digunakan untuk menampilkan hasil rekomendasi lowongan pekerjaan yang di sarankan oleh sistem.

2. Halaman saat format CV yang dimasukan salah

The screenshot shows a web interface for job recommendations. At the top, there's a header 'REKOMENDASI LOWONGAN PEKERJAAN'. Below it, a section titled 'Upload CV (PDF)' contains a 'Choose File' button and a message 'No file chosen'. A blue 'Upload' button is also present. Below this, a pink banner displays the error message: 'CV gagal diterima. Silakan sesuaikan CV anda!'. At the bottom, there is a table with columns: NO, Nama Pekerjaan, Nama Perusahaan, Alamat Perusahaan, Kategori Pekerjaan, Keahlian, Deskripsi Pekerjaan, and Similarity. The table currently shows 'Tidak ada rekomendasi'.

Gambar 4. 9 Tampilan saat format CV yang dimasukan salah

Pada gambar 4.9 merupakan tampilan halaman disaat sistem tidak menerima *input* dokumen CV yang tidak sesuai dengan ketentuan seperti lebih dari 3 halaman dan format CV berupa gambar yang di-convert menjadi PDF. Dengan memunculkan pemberitahuan bahwa CV gagal diterima. Silakan sesuaikan CV anda!

3. Halaman Rekomendasi

The screenshot shows the same web interface as before, but now it displays a list of job recommendations. The 'Upload CV (PDF)' section still shows the error message. Below it, a green banner says 'CV berhasil diterima. Berikut rekomendasi yang sesuai untuk anda!'. The table below has columns: NO, Nama Pekerjaan, Nama Perusahaan, Alamat Perusahaan, Kategori Pekerjaan, Keahlian, Deskripsi Pekerjaan, and Similarity. It lists one recommendation for a 'Front End Engineer Senior' position at 'Mountain peak programming'.

NO	Nama Pekerjaan	Nama Perusahaan	Alamat Perusahaan	Kategori Pekerjaan	Keahlian	Deskripsi Pekerjaan	Similarity
1	Front End Engineer Senior	Mountain peak programming	Jakarta	Software Engineering	Skills: Node.js, JavaScript, ReactJS, HTML, MySQL, Java, Python, Solving, CSS, AJAX, OOH, Manipulation, JS/JSX	Deskripsi pekerjaan Front End Engineer Senior Mountain peak programming: 1. Bertugas desain dan struktur halaman web 2. Kembangkan fitur untuk meningkatkan pengalaman pengguna 3. Merancang keseluruhan arsitektur desain fungsional dan estetika 4. Pastikan desain web mendukung smartphone 5. Buat kode yang dapat digunakan kembali untuk pengurusan di masa mendatang 6. Pastikan layanan web dioptimalkan untuk kecepatan dan skalabilitas terbaik 7. Bereskan beragam pilihan bahasa markup untuk mendesain halaman web 8. Jaga agar merek tetap konsisten di seluruh desain 9. Lihat Lebih Banyak Persyaratan wajib menjadi Front End Engineer Senior di Mountain peak programming 1. Berpengalaman lebih dari 3 tahun dalam mengembangkan pemrosesan tradisional atau board game; 2. Berpartisipasi dalam setidaknya satu proses pengembangan model secara online; 3. Menulis desain struktur data dan algoritma untuk umum	29.88 %

Gambar 4. 10 Halaman Rekomendasi

Pada gambar 4.9 merupakan tampilan ketika *user* sudah menjalankan atau memasukan *form input* PDF CV. Pada halaman ini sistem akan menampilkan rekomendasi lowongan pekerjaan bagi pengguna yang ditampilkan dalam sebuah

tabel dimana berisi no, nama pekerjaan, nama perusahaan, alamat perusahaan, kategori pekerjaan, keahlian, deskripsi pekerjaan, dan *similarity* atau tingkat kesamaan.

4.5 Hasil Pengujian Sistem

Ada beberapa pengujian yang dilakukan untuk menguji sistem rekomendasi lowongan pekerjaan antara lain pengujian performa sistem dan pengujian *Black Box*. Berikut merupakan rincian dari hasil pengujian sistem rekomendasi lowongan pekerjaan:

1. Pengujian Performa Sistem

Pada tabel 4. 2 merupakan hasil dari pengujian performa dari sistem rekomendasi lowongan pekerjaan.

Tabel 4. 2 Pengujian Performa Sistem

Input	Output	Waktu Yang Dibutuhkan (Menit)
PDF CV dengan 238 kata	Berpindah ke halaman rekomendasi keluar hasil rekomendasi	02:47
PDF CV dengan 375 kata	Berpindah ke halaman rekomendasi keluar hasil rekomendasi	02:57
PDF CV dengan 857 kata	Berpindah ke halaman rekomendasi keluar hasil rekomendasi	03:20
PDF CV dengan 182 kata	Berpindah ke halaman rekomendasi keluar hasil rekomendasi	02:37
PDF CV dengan 215 kata	Berpindah ke halaman rekomendasi keluar hasil rekomendasi	02:42
PDF CV dengan 285 kata	Berpindah ke halaman rekomendasi keluar hasil rekomendasi	02:52

Dari hasil pengujian performa sistem pada tabel 4. 2 menunjukkan bahwa performa sistem bergantung kepada jumlah kata yang ada pada CV semakin kaya kata yang terdapat pada CV, proses pada *text preprocessing* akan semakin lama. Sistem ini bergantung kepada koneksi internet. Uji test pada tabel 4. 2 dilakukan pada kecepatan koneksi internet rata-rata 100 Kbps sampai 200 Kbps.

2. Pengujian *Black Box*

Pengujian *black box* adalah untuk *meng-input* dan menguji apakah fungsi-fungsi yang terdapat pada sistem yang sedang berjalan sudah sesuai dengan tujuan, dan tercermin dalam hasil keluaran. Adapun rencana yang akan dibuat yaitu *input*, hasil yang ingin di harapkan, *output* dan kesimpulan, yang akan diuji adalah *form upload CV*, dan hasil rekomendasi Pada tabel 4. 3 merupakan hasil pengujian dari *Black Box*.

Tabel 4. 3 Pengujian *Black Box*

<i>Input</i>	<i>Hasil yang diharapkan</i>	<i>Output</i>	<i>Kesimpulan</i>
<i>Upload PDF CV dengan bahasa campuran bahasa Inggris dan bahas Indonesia dengan data lowongan pekerjaan bahasa Inggris</i>	Menampilkan rekomendasi lowongan pekerjaan	Begitu berpindah ke halaman rekomendasi keluar hasil rekomendasi	Berhasil
<i>Upload PDF CV berbahasa Indonesia dengan data lowongan pekerjaan berbahasa Inggris</i>	Menampilkan rekomendasi lowongan pekerjaan	Begitu berpindah ke halaman rekomendasi keluar hasil rekomendasi	Berhasil
<i>Upload PDF CV 3 halaman berbahasa Inggris dengan data lowongan pekerjaan berbahasa Inggris</i>	Menampilkan rekomendasi lowongan pekerjaan	Begitu berpindah ke halaman rekomendasi keluar hasil rekomendasi	Berhasil
<i>Upload PDF CV dengan isi CV berupa gambar dan tidak bisa dilakukan proses <i>copy paste</i></i>	Menampilkan pemberitahuan untuk pengecek format CV yang dikirim	Begitu berpindah ke halaman cek keluar pemberitahuan untuk mengecek format CV	Berhasil
<i>Upload PDF CV dengan lebih dari 3 halaman</i>	Menampilkan pemberitahuan untuk pengecek format CV yang dikirim	Begitu berpindah ke halaman cek keluar pemberitahuan untuk mengecek format CV	Berhasil

3. Pengujian *Translate to English*

Pada pengujian ini akan dilakukan uji *parsing* dengan beberapa CV yang memiliki beberapa bahasa didalamnya dan menguji hasil *translate to english* apakah hasil *parsing* yang masuk kedalam proses *translate to english* sudah sesuai atau ada beberapa kata yang tidak ter-*translate* dengan baik. Pada tabel 4.4 merupakan hasil dari pengujian *translate to english*.

Tabel 4. 4 Pengujian *Translate to English*

Skenario Pengujian	Kasus Pengujian	Hasil Pengujian	Kesimpulan
Parsing data CV berbahasa Inggris	Upload PDF CV 1 halaman berbahasa Inggris	Sesuai	Tidak ada error
Parsing data CV berbahasa Indonesia	Upload PDF CV 1 halaman berbahasa Indonesia	Sesuai	Tidak ada error
Parsing data CV dengan kombinasi bahasa didalamnya	Upload PDF CV 1 halaman dengan bahasa campuran bahasa Inggris dan bahasa Indonesia	Terdapat beberapa kata yang tidak bisa diterjemahkan dengan baik kedalam satu bahasa	Tidak ada error
Parsing data CV dengan lebih dari 1 halaman berbahasa Inggris	Upload CV PDF 3 halaman berbahasa Inggris	Sesuai	Tidak ada error
Parsing data CV dengan lebih dari 1 halaman berbahasa Indonesia	Upload PDF CV 3 halaman berbahasa Indonesia	Sesuai	Tidak ada error

Dari hasil pengujian terdapat kasus uji yang mengalami kegagalan saat melakukan *parsing data* CV yang memasuki proses *translate to english*. Dimana *testing* yang pertama untuk menguji CV yang memiliki bahasa campuran untuk pengujian menggunakan CV yang memiliki kombinasi bahasa Inggris dan bahasa Indonesia, dengan hasil beberapa kata tidak bisa diterjemahkan ke satu bahasa yang sesuai.

4.6 Validasi Implementasi Algoritma

1. Konsistensi *Cosine*

Konsistensi metode *cosine similarity* terbukti konsisten , dapat dibuktikan dari pengujian yang dilakukan selama 10x dengan dokumen terkait, hal ini menghasilkan angka yang sama di setiap prosesnya.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian ini, dapat diambil kesimpulan tentang rekomendasi lowongan pekerjaan menggunakan metode *cosine similarity* adalah sebagai berikut metode *cosine similarity* bekerja dengan baik, hal ini terbukti dengan hasil yang akurat dan konsisten di setiap perhitungannya. Dari hasil tingkat kemiripan dokumen CV dengan data lowongan pekerjaan dapat dijadikan parameter kecocokan pelamar dengan lowongan yang tersedia. Semakin tinggi nilai kemiripan atau *similarity* semakin berpeluang untuk diterima dalam lowongan pekerjaan yang bersangkutan.

5.2 Saran

Saran yang dapat diterapkan untuk pengembangan sistem yang ini lebih lanjut dimasa mendatang yaitu : Pada sistem rekomendasi lowongan pekerjaan menggunakan metode *cosine similarity* adalah setelah hasil didapatkan sistem dapat memberikan saran perbaikan pada CV pelamar dan *upgrading skill* atau keahlian yang belum sesuai dengan apa yang dibutuhkan oleh rekomendasi lowongan pekerjaan tertinggi supaya rasio diterimanya pelamar lebih tinggi.

DAFTAR PUSTAKA

- Dewi, R., Givan, B., & Wiinarno, H. (2021). Pelaksanaan Rekrutmen, Seleksi dan Penempatan Kerja Karyawan (Studi pada Karyawan PT Gemala Kempa Daya). *Jurnal Administrasi Bisnis*, 1(1), 49–55. www.igpgroup.co.id/lamaran.
- Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.
- Ferio, G., Intan, R., & Rostianingsih, S. (2019). Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan Metode User Based Collaborative *Filtering* Berbasis Algoritma Adjusted Cosine Similarity. *Jurnal Infra*, 7(1), 1–7.
- Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, 4(1), 1.
- Ivory, M. Y. (2004). Website usability engineering. In *The Practical Handbook of Internet Computing* (hal. 44-1-44–17). CRC Press. <https://doi.org/10.1201/9780203507223>
- Lahitani, A. R. (2022). Automated Essay Scoring menggunakan Cosine Similarity pada Penilaian Esai Multi Soal. *Jurnal Kajian Ilmiah*, 22(2), 107–118. <https://doi.org/10.31599/jki.v22i2.1121>
- Martinez, A. R. (2010). Natural language processing. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3), 352–357. <https://doi.org/10.1002/wics.76>
- Nugraha, A., & Budiyanto, U. (2022). Adaptive E-Learning System Berbasis Vark Learning Style dengan Klasifikasi Materi Pembelajaran Menggunakan K-NN (K-Nearest Neighbor). *Technomedia Journal*, 7(2), 248–261. <https://doi.org/10.33050/tmj.v7i2.1900>
- Nugroho, E. (2011). Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp. In *Journal of Strategic Studies*. <https://doi.org/10.1080/01402390.2011.569130>
- Putra, M. G. L., & Putera, M. I. A. (2019). Analisis Perbandingan Metode Soap Dan Rest Yang Digunakan Pada Framework Flask Untuk Membangun Web Service. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, 14(2), 1–7.

<https://doi.org/10.33005/scan.v14i2.1480>

- Rahayu, S., & RMS, A. S. (2018). Penerapan Metode Naive Bayes Dalam Pemilihan Kualitas Jenis Rumput Taman CV. Rumput Kita Landscape. *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, 9(2), 162–171.
- Riyani, A., Naf' an, M. Z., & Burhanuddin, A. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. *Jurnal Linguistik Komputasional*, 2(1), 23–27. <https://doi.org/10.26418/jlk.v2i1.17>
- Rosyadi, H. E., Amrullah, F., Marcus, R. D., & Affandi, R. R. (2020). Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (Natural Language Processing). *BRILIANT: Jurnal Riset dan Konseptual*, 5(1), 619–626.
- Sari, N. H. A., Fauzi, M. A. F., & Adikara, P. P. (2018). Klasifikasi Dokumen Sambat Online Menggunakan Metode K-Nearest Klasifikasi Dokumen Sambat Online Menggunakan Metode K-Nearest Neighbor dan Features Selection Berbasis Categorical Proportional Difference. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(August), 2449–2454.
- Sembilu, N., Samopa, F., & Er, M. (2018). Perbandingan Algoritma Kemiripan Teks Untuk Perbaikan Dan Saran Penulisan Frasa Dalam Bahasa Alami. *Sisfo*, 08(01). <https://doi.org/10.24089/j.sisfo.2018.09.002>
- Suhandi, Wijayanto, H., & Olde, S. (2020). Dinamika Permasalahan Ketenagakerjaan Dan Pengangguran Di Indonesia. *Jurnal Bina Bangsa Ekonomika*, 13(1), 85–94. <https://doi.org/10.46306/jbbe.v13i1.33>
- Tan, A.-H. (1999). Text mining: The state of the art and the challenges. *Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases*, 8, 65–70.
- Turland, M. (2010). *Phpparchitect's guide to web scraping with PHP*. Marco Tabini.