

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

Bagas Afza Joko Ariyanto¹, Sam Farisa Chaerul Haviana², Dedy Kurniadi³

¹Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang

^{2,3}Dosen Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang

Correspondence Author: bagasariyanto156@std.unissula.ac.id, sam@unissula.ac.id,
ddy.kurniadi@unissula.ac.id

Abstract

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 similarity 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 program tidak memiliki error saat dijalankan. Terdapat dua 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. Pengujian yang kedua dengan membandingkan data CV yang memiliki bahasa campuran yaitu bahasa Indonesia dan bahasa Inggris dengan data lowongan pekerjaan yang memiliki satu bahasa yaitu bahasa Inggris dengan hasil nilai similarity antara keduanya rendah.

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

1. PENDAHULUAN

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 [1] 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 [2] 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.

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 [3]. 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 [4]. Dari hasil penelitian [5] 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 [6] 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 [7] 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.

1.1 *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 [1] 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.

1.2 *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

1.3 *Text Mining*

Text Mining merupakan sebuah pengembangan dari ilmu pengetahuan yang bertujuan untuk mendapatkan informasi asli (teks asli) [8]. 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 [9]. 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 [10].

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” [11].
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 [12].

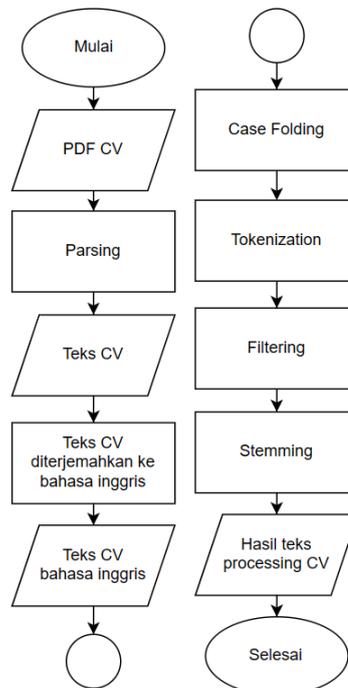
4. *Stemming* yaitu tahapan penggalan data teks untuk memperkecil jumlah indeks dari satu data hingga sebuah kata memiliki *suffix* maupun *prefix* akan kembali ke bentuk dasarnya. Dapat diartikan bahwa proses *Stemming* adalah proses untuk mengubah kata dari hasil *stopword* menjadi bentuk kata dasar (root word).

1.4 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 di persiapkan dapat dibuang 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 [13]. 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 [14].

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 [15].

Pada gambar 1 merupakan langkah – langkah dalam *text preprocessing* yang direpresentasikan dalam sebuah *flowchart*.



Gambar 1. *Flowchart* Alur *Text Preprocessing*

Pada gambar 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”, “dilihtkan” akan ditransformasi menjadi kata “lihat”.

1.5 Web Scraping

Web Scraping merupakan proses pengambilan data atau dokumen semiterstruktur 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 [16].

1.6 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 [17].

1.7 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 [6]. Cara metode *Cosine Similarity* mengukur kemiripan dokumen atau teks melalui perhitungan nilai sudut *cosinus* dari perkalian dua vektor yang dibandingkan [18]. *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.

1.8 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 [19]. *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. METODE PENELITIAN

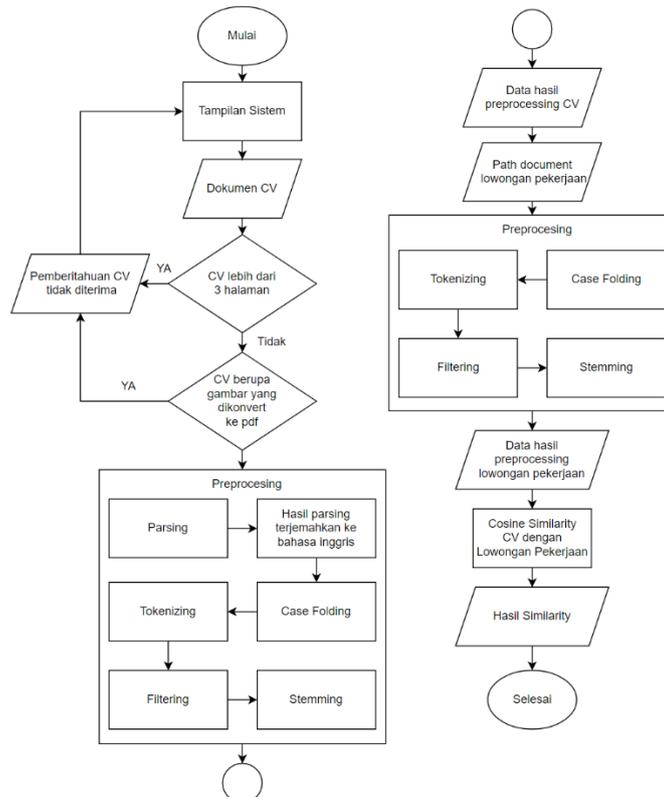
1. Data Collecting

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, engineering, 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.

2. Perancangan Arsitektur Sistem

Dalam merancang arsitektur suatu sistem, diperlukan *flowchart* yang menunjukkan langkah-langkah bagaimana sistem berjalan. Pada gambar 2 merupakan *flowchart* rancangan sistem yang akan dibangun :



Gambar 2. Flowchart Rancangan Arsitektur Sistem

Gambar 2 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 pengubahan 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. 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. HASIL DAN ANALISA

Sistem ini berupa pengembangan 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 *preprocessing data* seperti *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 *preprocessing data* 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

CV dengan format Applicant Tracking System (ATS) 3 halaman

Kalash Jindal

3rd year B.Tech. Computer Science and Engineering
Pioneer Group Institute of Engineering and Technology

EDUCATION

8.70/100 CGPA (Up to 2nd year)
Pioneer Group Institute of Engineering and Technology,
Raibareilly — B.Tech Computer Science and Engineering
2017 - till date

84.4/100 Percentage
Agarwal Public Inter College,
Sikapur — Intermediate
2016

90.4/100 Percentage
Agarwal Public Inter College,
Sikapur — High School
2014

TRAINING

Machine Learning Master Course — Coding Blocks
<https://www.codingblocks.com/courses/ml-course/> 2018-2019

Divide and Conquer, Sorting and Searching, and
Recursion and Algorithms — Coursera
<https://www.coursera.org/learn/divide-and-conquer/sort-searching-and-recursion-algorithms>
2017-2018

Introduction to Data Science in Python — Coursera
<https://www.coursera.org/learn/introduction-to-data-science-in-python>
2018

Programming with Python — Internshala Trainings
Certificate Number: A090515-9324-A030-000A-10CA30AC2ACAF

206, Thimarganj, Sikapur
Raibareilly, India
+91 9264634631
kalashjindal99@gmail.com
www.linkedin.com/company/kalashjindal/

SKILLS

Python, R, MATLAB, Data
Handling, Data Visualization,
Linear Algebra, Neural
Networks, Transfer Learning,
Feature Extraction, Deep
Learning, Sci Kit Learn, Numpy,
OpenCV, SQL, Java, C++, C#,
Using Pyqt module, Git, GitHub,
C++, etc.

HOBBIES

Table Tennis, Coding, Reading
books, Web surfing, etc.

PROJECTS

Cartpole Playing Game using Reinforcement Learning
A Cartpole (penbuild API by OpenAI Gym) is placed on the one-dimensional track having a pole which can move either left or right. The goal is to make the left or right momentum in such a way that the pole does not fall down from a certain angle. So, I have used the Gym library for importing the Cartpole model and Q-Learning principle to design the algorithm for consistently making momentum of the pole to achieve the highest accuracy.

Sentimental Analysis using MLP and LSTM on IMDb Dataset
In this project, I have worked on the IMDb dataset for checking the sentiment of the reviews given by the reviewers either the review is positive or negative which is used for giving the movie rating out of 5. I have also implemented the concept of early stopping, to overcome the overfitting problem.

Emoji Prediction using LSTM
This project is based on scraping the right emoji from the set of emoji according to the sentiment of the piece of text. While modelling the neural network, I have used LSTM for it with activation function softmax and GloVe (Global Vector For Word Representation), which is an unsupervised learning algorithm for obtaining the vector representation of words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcased interesting linear substructures of the word vector space. Adam is used for optimizing the model, the loss is categorical cross-entropy.

Text Generation using Markov Chain
In this project, I have worked on a probabilistic model for text/natural language generation. Using the Markov Chain algorithm for generating the text upto a certain length.

Image Classification using SVM
In this project, I have worked on support vector machine for extracting the features from the training dataset containing images of dog, cat, horse and humans of around 100. Applying data augmentation for increasing the size of dataset and using SVM for classifying the testing set into 4 classes.

Odd one out
This project works to find out odd one out, i.e., the least similar element out from any inputted list. Word2Vec model using GloVe(pre-trained model) is used for word embedding/vector representation of the words.

Bollywood Word Analogies
In this project, I have worked with Word2Vec model for finding the analogies of word, i.e., used to complete the sentence "A is to B and C is to ...". The goal is to find out the correct word, I have measured the cosine similarity of the word vectors of (B-C) and (A-D). Here, A is the list of word vectors present in the vocab of word2vec model.

Titanic Survivor Prediction using Decision Trees
Titanic Survivor is the very first question on kaggle. In this problem, I have used decision trees for predicting whether a person will survive or not on the sinking date.

Diabetic Detection
This project is modeled using the KNN and the dataset used is of people of any country having high diabetes patients and the goal is to find out that the patient is diabetic or not on testing date, I have used binary classifier for classifying the possibility.

Fantasy Cricket Game
This game is a genre of dream 11. I have work on this project while completing my python training. In this project, I have used python, SQLite for database management and qt designer for UI designing.

Air Quality Prediction
Dataset used in this project is of the different places with having 5 features. So, I have used Linear regression with multiple features for finding out the air quality of testing date and at last, I achieve an accuracy of 98%.

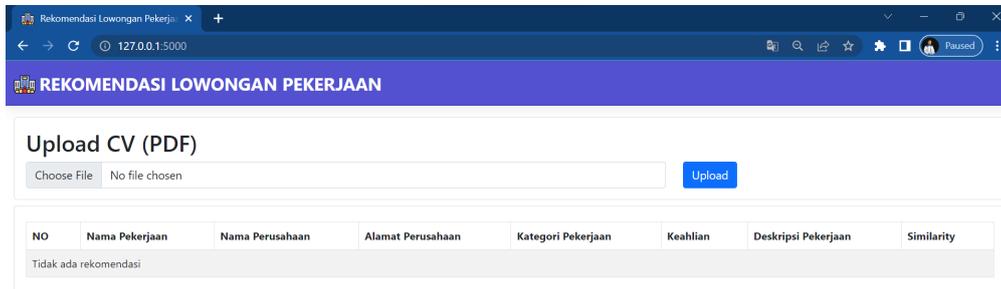
Hard Work Pays Off
This project is based on the data of the students and how they performed in the evaluation exam. And I have implemented Linear Regression for finding out the minimum time required for each student to do coding daily. R2 score is used for calculating the accuracy and I got 97% accuracy.

Separating Chemicals
In this project, I have worked on logistic regression to create a model to solve this challenge. Dataset is labeled with a and b with each having 3 features and need is to label the testing date. I attain 97% accuracy.

3.3 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 3. Tampilan Sistem

Pada gambar 3 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.

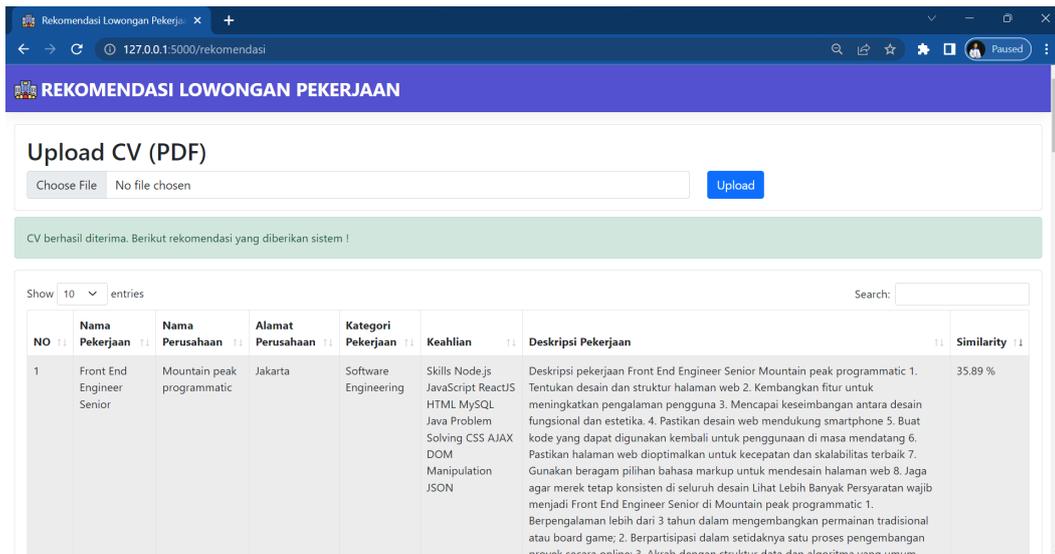
2. Halaman saat format CV yang dimasukkan salah



Gambar 4. Tampilan saat format CV salah

Pada gambar 4 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



Gambar 5. Halaman rekomendasi

Pada gambar 5 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

3.4 Hasil Pengujian Sistem

Ada beberapa pengujian yang lakukan 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 2 merupakan hasil dari pengujian performa dari sistem rekomendasi lowongan pekerjaan.

Table 2. Pengujian Sistem

<i>Input</i>	<i>Output</i>	<i>Waktu Yang Dibutuhkan (Menit)</i>
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 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 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 3 merupakan hasil pengujian dari *Black Box*.

Table 3. Pengujian *Black Box*

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

Table 4. Pengujian *Translate to English*

Skenario Pengujian	Kasus Pengujian	Hasil Pengujian	Kesimpulan
Parsing data CV berbahasa Inggris	<i>Upload PDF CV</i> 1 halaman berbahasa Inggris	Sesuai	Tidak ada error
Parsing data CV berbahasa Indonesia	<i>Upload PDF CV</i> 1 halaman berbahasa Indonesia	Sesuai	Tidak ada error
Parsing data CV dengan kombinasi bahasa didalamnya	<i>Upload PDF CV</i> 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	<i>Upload CV PDF</i> 3 halaman berbahasa Inggris	Sesuai	Tidak ada error
Parsing data CV dengan lebih dari 1 halaman berbahasa Indonesia	<i>Upload PDF CV</i> 3 halaman berbahasa Indonesia	Sesuai	Tidak ada error

3.5 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.

4. 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.

DAFTAR PUSTAKA

- [1] R. Dewi, B. Givan, dan H. Wiinarno, "Pelaksanaan Rekrutmen, Seleksi dan Penempatan Kerja Karyawan (Studi pada Karyawan PT Gemala Kempa Daya)," *J. Adm. Bisnis*, vol. 1, no. 1, hal. 49–55, 2021, [Daring]. Tersedia pada: www.igpgroup.co.id/lamaran.
- [2] Suhandi, H. Wijayanto, dan S. Olde, "Dinamika Permasalahan Ketenagakerjaan Dan Pengangguran Di Indonesia," *J. Bina Bangsa Ekon.*, vol. 13, no. 1, hal. 85–94, 2020, doi: 10.46306/jbbe.v13i1.33.
- [3] N. L. P. C. Savitri, R. A. Rahman, R. Venyutzky, dan N. A. Rakhmawati, "Analisis Klasifikasi Sentimen Terhadap Sekolah Daring pada Twitter Menggunakan Supervised Machine Learning," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, hal. 47–58, 2021, doi: 10.28932/jutisi.v7i1.3216.
- [4] A. R. Martinez, "Natural language processing," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 3, hal. 352–357, 2010, doi: 10.1002/wics.76.
- [5] H. E. Rosyadi, F. Amrullah, R. D. Marcus, dan R. R. Affandi, "Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (Natural Language Processing)," *BRILLANT J. Ris. dan Konseptual*, vol. 5, no. 1, hal. 619–626, 2020.
- [6] A. Riyani, M. Z. Naf'an, dan A. Burhanuddin, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," *J. Linguist. Komputasional*, vol. 2, no. 1, hal. 23–27, 2019, [Daring]. Tersedia pada: <https://doi.org/10.26418/jlk.v2i1.17>
- [7] G. Ferio, R. Intan, dan S. Rostianingsih, "Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan Metode User Based Collaborative Filtering Berbasis Algoritma Adjusted Cosine Similarity," *J. Infra*, vol. 7, no. 1, hal. 1–7, 2019.
- [8] M. Y. Ivory, "Website usability engineering," in *The Practical Handbook of Internet Computing*, CRC Press, 2004, hal. 44-1-44-17. doi: 10.1201/9780203507223.
- [9] A.-H. Tan, "Text mining: The state of the art and the challenges," in *Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases*, 1999, vol. 8, hal. 65–70.
- [10] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, dan M. R. Yeganegi, "Text mining in big data analytics," *Big Data Cogn. Comput.*, vol. 4, no. 1, hal. 1, 2020.
- [11] A. Nugraha dan U. Budiyanto, "Adaptive E-Learning System Berbasis Vark Learning Style dengan Klasifikasi Materi Pembelajaran Menggunakan K-NN (K-Nearest Neighbor)," *Technomedia J.*, vol. 7, no. 2, hal. 248–261, 2022, doi: 10.33050/tmj.v7i2.1900.
- [12] N. H. A. Sari, M. A. F. Fauzi, dan P. P. Adikara, "Klasifikasi Dokumen Sambat Online Menggunakan Metode K-Nearest Klasifikasi Dokumen Sambat Online Menggunakan Metode K-Nearest Neighbor dan Features Selection Berbasis Categorical Proportional Difference," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. August, hal. 2449–2454, 2018.
- [13] S. Rahayu dan A. S. RMS, "Penerapan Metode Naive Bayes Dalam Pemilihan Kualitas Jenis Rumput Taman CV. Rumput Kita Landscape," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 9, no. 2, hal. 162–171, 2018.
- [14] R. Feldman dan J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.
- [15] E. Nugroho, *Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp*. 2011. doi: 10.1080/01402390.2011.569130.
- [16] M. Turland, *Phparchitect's guide to web scraping with PHP*. Canada: Marco Tabini, 2010.
- [17] M. G. L. Putra dan M. I. A. Putera, "Analisis Perbandingan Metode Soap Dan Rest Yang Digunakan Pada Framework Flask Untuk Membangun Web Service," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 14, no. 2, hal. 1–7, 2019, doi: 10.33005/scan.v14i2.1480.
- [18] A. R. Lahitani, "Automated Essay Scoring menggunakan Cosine Similarity pada Penilaian Esai Multi Soal," *J. Kaji. Ilm.*, vol. 22, no. 2, hal. 107–118, 2022, doi: 10.31599/jki.v22i2.1121.
- [19] N. Sembilu, F. Samopa, dan M. Er, "Perbandingan Algoritma Kemiripan Teks Untuk Perbaikan Dan Saran Penulisan Frasa Dalam Bahasa Alami," *Sisfo*, vol. 08, no. 01, 2018, doi: 10.24089/j.sisfo.2018.09.002.