

Penerapan Natural Language Processing Pada Website Pencarian Lowongan Pekerjaan

1st Muhammad Arlen Yusuf
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

yusufarlen@student.telkomuniversity.ac.id

2nd Kris Sujatmoko
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

krisujatmoko@telkomuniversity.ac.id

3rd Iwan Iwut Tritoasmoro
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

iwaniwut@telkomuniversity.ac.id

Abstrak— Mencari lowongan pekerjaan secara digital telah menjadi hal umum pada zaman sekarang. Banyak perusahaan menyediakan lowongan pekerjaan melalui website mereka, seperti Karir.com dan LinkedIn. Jumlah lowongan pekerjaan yang diposting mencapai ribuan, bahkan puluhan ribu, menimbulkan kesulitan bagi pelamar untuk menemukan pekerjaan yang sesuai dengan kualifikasinya. Oleh karena itu, diperlukan automasi untuk membantu pelamar menemukan pekerjaan yang cocok dengan skill mereka. Salah satu solusi automasi yang dapat digunakan adalah perancangan website yang mengintegrasikan Robotic Process Automation (RPA). Selain itu, untuk mencocokkan skill pelamar dengan persyaratan pekerjaan, dapat menggunakan Natural Language Processing (NLP) yang berfungsi untuk menyaring kata-kata dalam persyaratan pekerjaan dan mengubahnya menjadi dataset. Selain itu, NLP juga membandingkan kata-kata dalam skill pelamar dengan dataset persyaratan pekerjaan. Selain itu, NLP juga membandingkan kata-kata dalam skill pelamar dengan dataset persyaratan pekerjaan.

Kata kunci— RPA, AI, NLP. Website, data scraping, Google Form, UiPath

I. PENDAHULUAN

Pada zaman sekarang, banyak kegiatan yang dilakukan secara digital, salah satunya adalah mencari lowongan pekerjaan. Sekarang, sudah banyak perusahaan yang menyediakan lowongan pekerjaan melalui website. Selain itu, adapun website yang menyediakan informasi mengenai lowongan pekerjaan. Contohnya adalah Jobstreet dan Kalibr. Tercatat pada Juli 2022, Jobstreet dapat memposting lebih dari 30.000 lowongan dan kalibr memposting lebih dari 5.000 lowongan. Oleh karena itu, dibutuhkan automasi untuk membantu pelamar menemukan pekerjaan yang cocok dengan skill yang dimiliki pelamar.

Salah satu teknologi automasi adalah Robotic Process Automation (RPA). RPA bekerja dengan menirukan aktivitas manusia menggunakan software. RPA akan sangat membantu pekerjaan manusia, terlebih pekerjaan yang berulang atau monoton. RPA disini digunakan untuk mencari pekerjaan, menerapkan filter, dan melakukan data scrapping untuk kualifikasi pekerjaan. RPA juga dapat mengirimkan email kecocokan CV pelamar dengan lowongan pekerjaan yang ada.

AI (Artificial Intelligence) berperan untuk mencari kecocokan CV pelamar dengan kualifikasi lowongan pekerjaan. Disini AI yang digunakan adalah Natural Language Processing (NLP). NLP berfungsi untuk memfilter kata yang ada pada persyaratan lowongan pekerjaan lalu mengubahnya menjadi dataset. Selain itu NLP juga berfungsi untuk membandingkan kata dari CV pelamar dengan hasil dataset persyaratan pekerjaan.

II. KAJIAN TEORI

A. Website

Website adalah Sebuah software yang berfungsi untuk menampilkan dokumen - dokumen pada suatu *website* yang membuat pengguna dapat mengakses internet melalui *software* yang terkoneksi dengan internet [1]. *Website* terdiri dari beberapa bagian seperti *front-end* dan *back-end*, *front-end website* adalah bagian dari bentuk pemrograman untuk mengelola tampilan *website* yang akan dilihat oleh pengguna. Sementara itu, *Back-end website* adalah bagian pemrograman untuk mengelola server dari *website*.

B. BeautifulSoup

BeautifulSoup adalah sebuah *library python* yang memungkinkan kita untuk menganalisis dan mengambil data dari dokumen HTML dan XML secara otomatis. Dalam penggunaan *library* ini, kita dapat dengan mudah melakukan *web scraping* untuk menemukan dan mengekstrak informasi yang diinginkan dari struktur dokumen *web*. *BeautifulSoup* menyediakan alat yang efisien untuk mencari elemen berdasarkan nama, atribut, dan struktur, sehingga mempermudah proses pengambilan teks, tautan, gambar, tabel, dan elemen lainnya dari halaman *web*.

C. Selenium

Selenium adalah sebuah *library python* yang digunakan dalam pengembangan *web* untuk mengotomatisasi interaksi dalam *web browser*. Dalam konteks *web scraping*, *selenium* digunakan jika teknik *parsing* tidak memungkinkan dan tidak praktis digunakan karena memiliki interaksi yang rumit yang memerlukan otomatisasi *browser* untuk menjalankannya

D. Natural Language Processing

Natural Language Processing (NLP) adalah penerapan ilmu komputer, khususnya linguistik komputasional (computational linguistics), untuk mengkaji interaksi antara komputer dengan bahasa (alami) manusia [2]. Tujuan penerapan NLP dalam website yaitu dengan mengimplementasikan *feature importance* untuk menganalisis prediksi hasil akhir dalam teks yang dipengaruhi oleh matriks menggunakan Tf-idf. Metode ini menilai pentingnya kata berdasarkan frekuensinya.

III. METODE

Dalam rangka perancangan website ini, penulis menggunakan beberapa metode diantaranya:

A. Studi Pustaka

Penulis melakukan pendekatan dengan tinjauan pustaka yaitu mempelajari jurnal-jurnal yang berisi tentang teori-teori sebagai bahan referensi yang berhubungan dengan permasalahan yang ada.

B. Eksperimen

Penulis menggunakan aplikasi VSCode dengan melakukan perancangan website pencarian lowongan pekerjaan. Penulis juga menggunakan aplikasi *JupiterLab* dan *Jupyter notebook* untuk membuat *code script Natural Language Processing* dan *script code web scraping*. Adapun analisis yang dilakukan dalam metode pengumpulan data sebagai berikut:

1. Kebutuhan:

Bagian ini untuk mendefinisikan informasi yang dibutuhkan, fungsi, kinerja, dan antarmuka, parameternya sebagai berikut :

- a. *Dataset skill job*
- b. *Scraping job result*

2. Desain:

Pada bagian ini memuat *script code* dan gambaran sistem jaringan yang dibuat. **GAMBAR 3. 1**

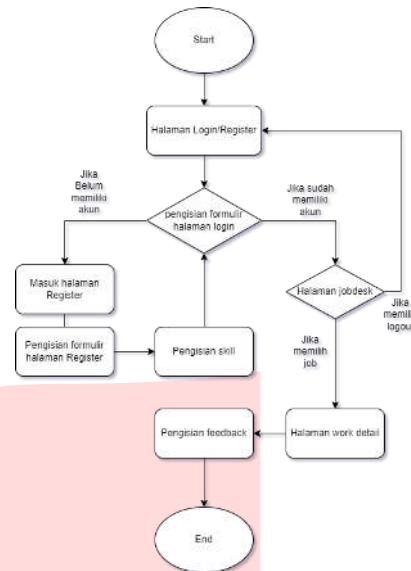
Desain Sistem Keseluruhan Website merupakan desain sistem keseluruhan website. **GAMBAR 3. 2**

Script Code Website merupakan *Script code website*. **GAMBAR 3. 3**

Script Code BeautifulSoup merupakan *Script code beautifulsoup*. **GAMBAR 3. 4**

Script Code Selenium merupakan *Script code Selenium*. **GAMBAR 3. 5**

Script Code Natural Language Processing merupakan *Script code Natural Language Processing*.



GAMBAR 3. 1
Desain Sistem Keseluruhan Website

```

<body>
  <nav class="navbar navbar-light bg-navbar fixed-top mb-2">
    <div>
      <a class="navbar-brand text-white" href="{ url('searchwork') }">
        <strong><i>Job FINDER</i></strong></a>
    </div>
    <a href="{ url('logout') }">
      <button type="button" class="btn btn-outline-light" title="sign out">
        <i>sign out</i>
      </button>
    </a>
  </nav>
  <!-- login form -->
  <div id="login" class="mt-5">
    <i>chi class="text-center text-white pt-5">Login Admin</i>
    <div class="container">
      <div id="login-row" class="row justify-content-center align-items-center">
        <div id="login-column" class="col-md-12 mt-4">
          <form class="needs-validation" action="/searchwork" novalidate>
            <div class="input-group">
              <input type="text" class="form-control" placeholder="Search vacancy Here" required pattern="[a-zA-Z0-9 ]{1,30}" style="background-image: none" name="search" value="{ request('search') }">
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
  </body>
  
```

GAMBAR 3. 2
Script Code Website

```

[18]: def open_page_job(job_url):
    job_page = requests.get(job_url)
    job_soup = BeautifulSoup(job_page.content, "html.parser")

    title = job_soup.find("h5", class_="title")
    location = job_soup.find("li", class_="job-location")
    company = job_soup.find("a", class_="link")
    salary = job_soup.find("span", class_="salary")
    total_experience = job_soup.find("li", class_="job-experience")

    job_desc = job_soup.find_all("section", class_="b-matte_content")
    min_qualification = job_desc[0]
    description = job_desc[1]

    #Check None
    if company is None:
        company = "-"
    else:
        company = company.text

    if total_experience is None:
        total_experience = "NULL"
    else:
        total_experience = total_experience.text

    if salary is None:
        salary = "-"
    else:
        salary = salary.text
    #Check None
  
```

GAMBAR 3. 3
Script Code BeautifulSoup

```

In [32]: job_id= []
job_title = []
company_name = []
location = []
date = []
job_link = []
for job in jobs:
    job_id = job.get_attribute('data-entity-urn')
    job_id.append(job_id)

    job_title = job.find_element(By.CSS_SELECTOR, 'h3').get_attribute('innerText')
    job_title.append(job_title)

    company_name = job.find_element(By.CSS_SELECTOR, 'h4').get_attribute('innerText')
    company_name.append(company_name)

    location = job.find_element(By.CLASS_NAME, 'job-search-card_location').get_attribute('innerText')
    location.append(location)

    date = job.find_element(By.CSS_SELECTOR, 'div:div:time').get_attribute('datetime')
    date.append(date)

    job_link = job.find_element(By.CSS_SELECTOR, 'a').get_attribute('href')
    job_link.append(job_link)

```

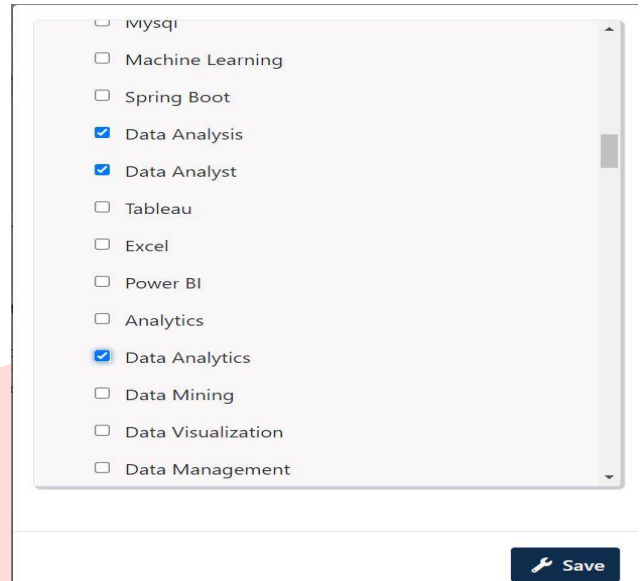
GAMBAR 3.4 Script Code Selenium

```

4 connection = mysql.connector.connect(host='127.0.0.1',
5                                     database='job_finder',
6                                     user='root',
7                                     password='')
8
9 def skill_weight():
10     sql_select_query = "select * from weight_result"
11     return pd.read_sql(sql_select_query, connection)
12
13 def list_job():
14     sql_select_query = "select * from search_works"
15     return pd.read_sql(sql_select_query, connection)
16
17 def job_point(skill):
18     skill_obj_str = skills_lower()
19     skill_obj = skill_obj_str.split(',')
20     df_skill = skill_weight()
21     list_job = list(df_skill['job'].drop_duplicates())
22
23     point_skill = []
24     for i in list_job:
25         point = 0
26         for j in skill_obj:
27             try:
28                 point += df_skill[(df_skill.job == i)&(df_skill.skill == j)][weight_perc].values[0]
29             except:
30                 pass
31         point_skill.append(point)
32
33     keys = list(point_skill.keys())
34     values = list(point_skill.values())

```

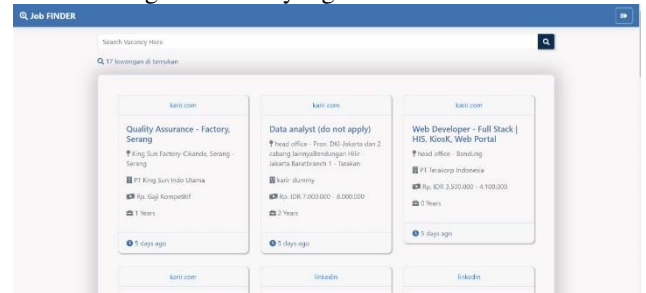
GAMBAR 3.5 Script Code Natural Language Processing



GAMBAR 4.1 Pengisian Skill User

B. Menampilkan Pekerjaan Yang Sesuai

Gambar 4.2 menunjukkan hasil proses oleh NLP. Dapat kita lihat bahwa pekerjaab yang ditampilkan akan sesuai dengan skill yang telah diisi oleh user.



GAMBAR 4.2 Hasil Pekerjaan

3. Implementasi

Bagian ini merupakan pengujian dan analisis, dan website. Dimana terdapat 4 bagian analisis yang akan dibahas yaitu website, BeautifulSoup, Selenium dan Natural Language Processing

4. Uji coba:

Pada bagian ini akan dilakukan pengujian menggunakan aplikasi VSCode, JupyterLab, dan Jupyter Notebook untuk mendapatkan hasil dari scraping dari beberapa website dengan library python yang dipakai. Visual Studio Code Visual Studio Code adalah kode editor sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS [3]. dan Jupyter Notebook merupakan aplikasi open source berbasis peramban untuk pengembangan dan presentasi interaktif proyek ilmu data. Serta Jupyter Lab yang merupakan sebuah "antarmuka web generasi berikutnya" untuk Jupyter notebook. JupyterLab merupakan sebuah aplikasi lanjutan dari jupyter notebook, dengan menambahkan fitur drag-and-drop, serta penyelusur berkas, penampil data, penyunting teks, dan konsol perintah. Berbeda dengan Jupyter notebook standar yang menetapkan setiap notebook memiliki kernel tersendiri.

IV. HASIL DAN PEMBAHASAN

A. Pengisian Skill User

Pada gambar dibawah menunjukkan skill yang dimiliki oleh user. Fungsi dari NLP sendiri adalah mencocokkan skill yang diisi user pada bagian registrasi dan pekerjaan yang tersedia pada database.

V. KESIMPULAN

Pengujian dilakukan untuk mengukur waktu eksekusi dan keakuratan data yang diperoleh, serta pengalaman pengguna dalam menggunakan sistem. Hasil pengujian menunjukkan bahwa subsistem registrasi dan login terdaftar dengan benar dan dimasukkan ke dalam database berdasarkan data yang dimasukkan oleh pengguna. Website scraping menggunakan BeautifulSoup membutuhkan waktu 35 detik untuk mengambil 30 pekerjaan dari situs web Karir.com, sedangkan website scraping Selenium membutuhkan waktu 5 menit untuk mengambil 30 pekerjaan. Selain itu, subsistem pembelajaran mesin dengan komponen pemrosesan natural language processing bekerja dengan lancar dan berhasil menilai pekerjaan berdasarkan keterampilan pengguna. Dengan hasil pengujian tersebut dapat disimpulkan bahwa proyek ini telah berhasil mengimplementasikan dan mengimplementasikan tiga subsistem utama dengan benar, sehingga pengguna dapat menggunakan sistem ini untuk mendapatkan pekerjaan sesuai dengan kebutuhan dan kemampuannya. Selain itu,

performa sistem dari segi waktu eksekusi dan akurasi data juga cukup baik.

REFERENSI

- [1] M. Florentina, Web Data Extraction With Robot Process Automation. Study On LinkedIn Web Scraping Using Uipath Studio., Annals of Constantin Brancusi'University of Targu-Jiu: Engineering Series 1, 2020.
- [2] S. Amber and P. James, Natural Language Annotation for Machine Learning, California: O'Reilly, 2012.
- [3] A. Agustini and W. J. Kurniawan, Sistem E-Learning Do'a dan Iqro'dalam Peningkatan Proses Pembelajaran pada TK Amal Ikhlas, Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi (JMApTeKsi), 2020, pp. 154-159.