

PENGLASIFIKASIAN PEKERJAAN DAN REKOMENDASI KURSUS *ONLINE* BERDASARKAN KEMAMPUAN PENGGUNA DENGAN MENGGUNAKAN SISTEM *MACHINE LEARNING*

Dionixius^{1*}

Matematika, FMIPA, Universitas Negeri Malang, dionixius.2003126@students.um.ac.id¹

*Email : dionixius.2003126@students.um.ac.id

Abstrak

Klasifikasi pekerjaan dan rekomendasi kursus online memiliki peran penting dalam menggabungkan kebutuhan pasar tenaga kerja dan pengembangan keterampilan individu. Pada era ini, pendekatan sederhana untuk klasifikasi pekerjaan dan rekomendasi kursus online dapat ditingkatkan melalui penerapan sistem machine learning. Dalam penelitian ini, kami menyajikan sistem yang menggunakan machine learning untuk mengelompokkan kemampuan berdasarkan pekerjaan pilihan dan memberikan rekomendasi kursus online yang sesuai berdasarkan keterampilan pengguna. Kami mengumpulkan deskripsi pekerjaan dari berbagai sumber dan melatih model machine learning untuk mengklasifikasikan pekerjaan ke dalam kategori yang relevan. Selanjutnya, kami mengintegrasikan sistem rekomendasi kursus online yang menganalisis keterampilan pengguna berdasarkan data sebelumnya. Dalam percobaan kami, sistem ini mampu memberikan klasifikasi pekerjaan dan rekomendasi kursus online yang akurat berdasarkan kemampuan pengguna. Dengan sistem ini, pengguna dapat dengan mudah menemukan kursus online terkait pekerjaan yang menarik bagi mereka dan meningkatkan keterampilan mereka sesuai dengan kebutuhan pasar. Kami percaya bahwa penggunaan machine learning dalam klasifikasi pekerjaan dan rekomendasi kursus online menawarkan potensi besar untuk mendorong pengembangan profesional dan beradaptasi dengan kebutuhan industri. Kepentingan praktis dari penelitian ini terletak pada pengembangan sistem yang akan membantu individu menemukan peluang karir yang cocok dan memilih kursus online yang dapat secara efektif meningkatkan keterampilan mereka di era digital yang terus berkembang.

Kata kunci: *Machine Learning, Kemampuan, Pekerjaan, Kursus Online, Pengelompokkan*

PENDAHULUAN

Pada era *digital* yang berkembang, penempatan kerja dan rekomendasi kursus *online* telah menjadi aspek yang semakin penting dalam mengintegrasikan kebutuhan pasar tenaga kerja dan keterampilan individu. Kemajuan teknologi telah membuka pintu bagi metode yang lebih canggih dan efisien dalam mengkategorikan pekerjaan dan memberikan rekomendasi kursus yang akurat secara *online*. Pendekatan yang menjanjikan adalah penggunaan sistem pembelajaran mesin dan pembelajaran mendalam, yang memungkinkan analisis data yang lebih akurat dan kompleks.

Natarajan, B., K. (2014) berpendapat bahwa pembelajaran mesin atau *Machine Learning* (ML) adalah cabang dari kecerdasan buatan atau *Artificial Intelligence* (AI) yang berfokus pada pengembangan sistem komputer yang dapat belajar secara otomatis tanpa harus diprogram secara manual oleh manusia [1]. Terdapat tiga jenis sistem dalam pembelajaran mesin, yaitu *supervised learning* atau pembelajaran terawasi adalah sistem yang akan melatih model yang sudah dilabeli untuk membuat prediksi baru, tujuan dari jenis ini adalah klasifikasi dan regresi, *unsupervised learning* atau pembelajaran tidak terawasi adalah sistem yang akan melatih model yang tidak memiliki label untuk membuat suatu prediksi baru, tujuan dari jenis ini adalah clustering dan reduksi dimensi, selanjutnya ada *reinforcement learning* (pembelajaran penguatan) yang mana sistem ini dibuat melalui interaksi dengan lingkungannya lalu mendapat umpan balik (*reward*) yang berisi tentang informasi tentang kualitas keputusannya, tujuan dari sistem ini adalah memaksimalkan akumulasi *reward* dalam jangka waktu tertentu.

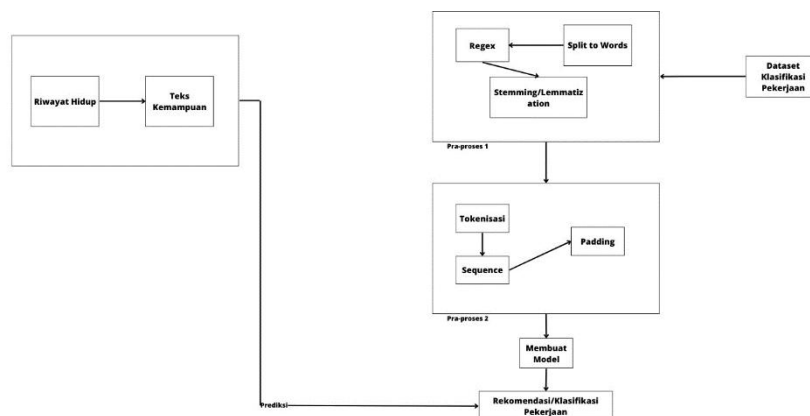
Selain pembelajaran mesin, ada pula sistem yang disebut dengan pembelajaran mendalam atau *Deep Learning* (DL). Pembelajaran mendalam adalah salah satu cabang dari sistem kecerdasan buatan yang menekankan pengembangan serta penerapan model jaringan saraf tiruan seperti manusia. Pembelajaran mendalam memakai sistem lapisan (*layer*) yang saling terhubung untuk menciptakan luaran yang bagus. Tujuan dari penggunaan lapisan tersebut adalah meminimalisasi kesalahan prediksi nantinya. Alpaydin, E. (2010) berpendapat bahwa dalam membuat sistem pembelajaran mendalam, dibutuhkan jumlah data yang sangat besar agar prediksi dari sistem tersebut semakin akurat [2]. Penggunaan sistem ini dapat dipakai dalam bidang pengenalan objek, pemrosesan bahasa alami, pemodelan bahasa, dan masih banyak lagi. Dalam

sistem pembelajaran mesin maupun pembelajaran mendalam, terdapat *library* yang populer dikalangan teknisi pembelajaran mesin, yaitu *library* TensorFlow dan Keras. Alasan kedua *library* tersebut sangatlah populer karena tingkat fleksibilitas yang tinggi, sistem yang sangat mutakhir untuk mendukung skala besar, mempunyai API yang mudah dan fleksibel untuk dipakai disegala sisi dan mudah untuk dilakukan *deployment* atau pengaplikasian disegala jenis gawai.

Dalam penelitian kali ini, saya melatih model menggunakan pembelajaran mesin dengan bantuan pembelajaran mendalam, dengan *dataset* dari situs Kaggle [6], dan melatihnya dengan model pemrosesan bahasa alami. Pedregosa, F., *et al.* (2011) berpendapat bahwa untuk sistem pembelajaran mesin, saya memakai *library* utama yaitu TensorFlow dan Keras [4]. Disisi lain berkas riwayat hidup akan disaring hingga menyisakan bagian kemampuan (*skill*), yang selanjutnya akan diprediksi oleh sistem pembelajaran mesin dengan mengeluarkan luaran berupa jenis pekerjaan dan rekomendasi kursus *online* berdasarkan jenis pekerjaan.

METODE

Menurut Bisong, E. (2019), membuat model menggunakan pembelajaran mesin, mengunggah data dan mengolahnya dapat memakai menggunakan Google Collab [3]. Data yang saya dapatkan dari situs Kaggle berjumlah 962 baris, selanjutnya data tersebut direduksi hingga tersisa 292 baris yang terdiri dari enam jenis pekerjaan, lalu terdapat dua kolom yaitu kolom kategori dan kolom *resume*. Data tersebut saya bagi menjadi dua bagian yaitu *training* sebesar 80% dan *testing* sebesar 20%. Sedangkan untuk penyaringan riwayat hidup, saya membuat kode tanpa bantuan sistem pembelajaran mesin, serta seluruh kode memakai bahasa pemrograman Python.



Gambar 1. Diagram Metodologi

1. Dataset Klasifikasi Pekerjaan dan Pra-proses data

Data yang dipakai dalam penelitian ini diambil dari situs Kaggle, dengan judul *Resume Dataset*. Data tersebut terdiri dari 962 baris lalu direduksi menjadi 292 baris dan memiliki 2 jenis kolom.

Tabel 1. Jenis Data

Atribut	Tipe Data
Kategori	Teks
<i>Resume</i>	Teks

Bagian data yang akan diolah yaitu bagian kategori dan resume. Bagian tersebut akan direduksi dan dibersihkan sehingga menjadi teks yang sederhana, dengan tujuan meningkatkan nilai akurasi prediksi nantinya. Pembersihan teks yang akan dilakukan terdiri dari:

- a. *Split to Words*

Membagi teks menjadi kata (*split to words*) adalah teknik yang akan memecah isi teks menjadi kata-per-kata, dengan tujuan agar komputer lebih mudah membaca teks yang akan ditokenisasi.

b. *Regex*

Regex atau reguler ekspresi adalah teknik untuk memanipulasi tipe data *string*. Teks adalah contoh dari tipe data *string*. Tujuan dari teknik ini untuk mereduksi kata-kata yang tidak penting didalam teks. Didalam teknik ini diberikan kode untuk mengubah kata menjadi huruf kecil (*lowercasing*), menghapus tautan, menghapus spasi berlebih, serta menghapus tanda baca dan non-ASCII karakter.

c. Huruf Kecil (*Lowercase*)

Nasser, Ibrahim M. (2020) berpendapat bahwa setelah data dibentuk kedalam bentuk bingkai data (*dataframe*), isi dari data tersebut harus diubah menjadi huruf kecil dengan fungsi `str.lower()`. [7]

d. *Stemming/Lemmatization*

Stemming/lemmatisasi adalah teknik yang akan mengubah seluruh kata dalam suatu teks menjadi kata dasar, contohnya adalah kata 'bermain', yang mana kata tersebut berasal dari bentuk 'main'.

e. Tokenisasi

Tokenisasi adalah teknik yang akan memecah suatu kalimat dalam teks menjadi kata-kata kecil, tujuan dari teknik ini yaitu untuk mempermudah proses analisa teks. Bird, S., *et al.* berpendapat bahwa tokenisasi dapat memakai NLTK (*Natural Language Toolkit*) [5]. Hasil dari teknik ini akan dilanjutkan pada *padding*.

f. *Sequencing*

Sequencing atau pengurutan mengacu pada urutan kata dalam teks. Data teks dipecah menjadi urutan kata atau token yang membentuk *sequence input*. Ini memungkinkan model untuk memahami konteks sebelumnya saat memprediksi kata berikutnya dalam kalimat.

g. *Padding*

Padding atau lapisan adalah teknik yang akan menambahkan elemen atau nilai tambahan pada teks yang sudah ditokenisasi. Tujuan dari teknik ini adalah menyamakan panjang maksimal dari suatu teks yang akan dipelajari nantinya oleh mesin. Terdapat dua jenis *padding*, yaitu *post* dan *pre*. Perbedaan kedua jenis tersebut terletak diposisi, jika *post* maka *padding* akan dilakukan diakhir kalimat sesuai dengan panjang yang sudah ditentukan, jika *pre* maka *padding* akan dilakukan sebelum kalimat sesuai dengan panjang yang sudah ditentukan.

2. Membuat Model

```

1 import tensorflow as tf
2
3 class MyCallback(tf.keras.callbacks.Callback):
4     def on_epoch_end(self, epoch, logs={}):
5         if(logs.get('accuracy') > 0.95 and logs.get('val_accuracy') > 0.95):
6             print("\nAkurasi model telah mencapai >90%")
7             self.model.stop_training = True
8     callbacks = MyCallback()
9
10 model = tf.keras.Sequential([
11     tf.keras.layers.Embedding(input_dim=1400, output_dim=32),
12     tf.keras.layers.LSTM(64),
13     tf.keras.layers.Dense(32, activation='relu'),
14     tf.keras.layers.Dense(64, activation='relu'),
15     tf.keras.layers.Dense(6, activation='softmax')
16 ])
17
18 model.compile(
19     optimizer=tf.keras.optimizers.Adam(learning_rate=1e-3),
20     loss='categorical_crossentropy',
21     metrics=['accuracy']
22 )

```

Gambar 2. Model Pembelajaran Mesin

Setelah melakukan pra-proses data, tahapan selanjutnya adalah membuat model yang tepat untuk data yang akan dilatih. Terdapat banyak jenis model yang bisa dikombinasikan satu sama lain dalam sistem pembelajaran mesin dan pembelajaran mendalam, contohnya LSTM (*Long-Short Term Memory*), *embedding*, naïve-bayes dan sebagainya. Dalam penelitian kali ini, saya memakai model *embedding* dan LSTM dipadukan dengan tiga lapisan *dense*. Penggunaan lapisan *embedding* memiliki keunggulan dalam hal menangkap hubungan semantik antar kata, dan penggunaan LSTM memiliki keunggulan dalam hal menangkap informasi berurutan pada data. Untuk lapisan *dense*, saya memakai 32, 64 dan 6 unit. Ketiga nilai tersebut adalah jumlah neuron tiap lapisannya, dan pada lapisan akhir 6 unit menandakan luaran dari model tersebut terdapat 6 jenis, yang memiliki arti akan terbentuknya 6 kategori pekerjaan berbeda. Setelah sistem sudah selesai, sistem akan disimpan kedalam berkas berekstensi h5, untuk memudahkan *deployment* dan mempercepat proses prediksi.

3. Penyaringan Berkas Riwayat Hidup

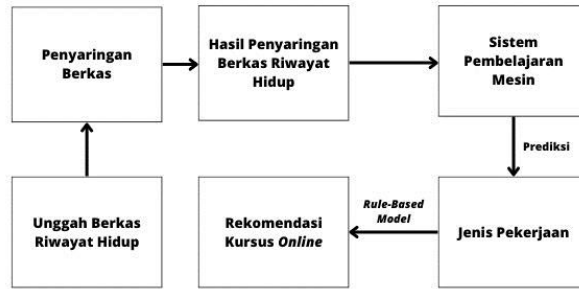
Untuk membuat kode yang akan menyaring bagian-bagian pada berkas riwayat hidup dan hanya menyisakan bagian kemampuan saja, saya tidak memakai sistem pembelajaran mesin. Pada tahap ini hal yang dibutuhkan yaitu mirip dengan pra-proses data seperti yang sudah dibahas sebelumnya. Tahapan dalam penyaringan meliputi reguler ekspresi (*regex*), lemmatisasi dan tokenisasi. Kode untuk penyaringan berkas riwayat hidup akan dibuat kedalam API (*Application Programming Interface*). API (*Application Programming Interface*) adalah kumpulan aturan atau alat yang digunakan oleh program komputer untuk berkomunikasi dengan program lain. API memungkinkan berbagai aplikasi atau komponen perangkat lunak untuk saling berinteraksi, berbagi data, dan melakukan tindakan tertentu. Penggunaan API yaitu untuk memudahkan *deployment* sistem pembelajaran mesin. Penyebaran atau *deployment* adalah proses penerapan aplikasi atau sistem dalam lingkungan produksi yang tersedia untuk pengguna akhir. Ini mencakup serangkaian langkah untuk menginstal, mengonfigurasi, dan menjalankan aplikasi atau sistem secara efisien dalam lingkungan produksi. Media atau *framework* yang dipakai dalam penelitian ini untuk *deployment* yaitu memakai Flask. Flask mudah dipahami dan fleksibel untuk dipakai dalam *deployment* sistem pembelajaran mesin. Selanjutnya, kode penyaringan yang sudah dimasukkan kedalam API tersebut akan menggunakan model pembelajaran mesin sebelumnya untuk memprediksi jenis pekerjaan.

4. Rekomendasi kursus *online*

Data rekomendasi kursus *online* diambil menggunakan metode *web scrapping*. Data diambil dari berbagai situs penyedia kursus *online* (MOOC). Metode yang dipakai dalam penerapan rekomendasi kursus *online* adalah dengan *rule-based model*. Metode yang mengandalkan logika keputusan yang sudah ditentukan sebelumnya, salah satu contoh dari metode ini adalah penggunaan *if-elif*. Daftar rekomendasi akan ditampilkan setelah sistem pembelajaran mesin berhasil memprediksi hasil penyaringan berkas riwayat hidup.

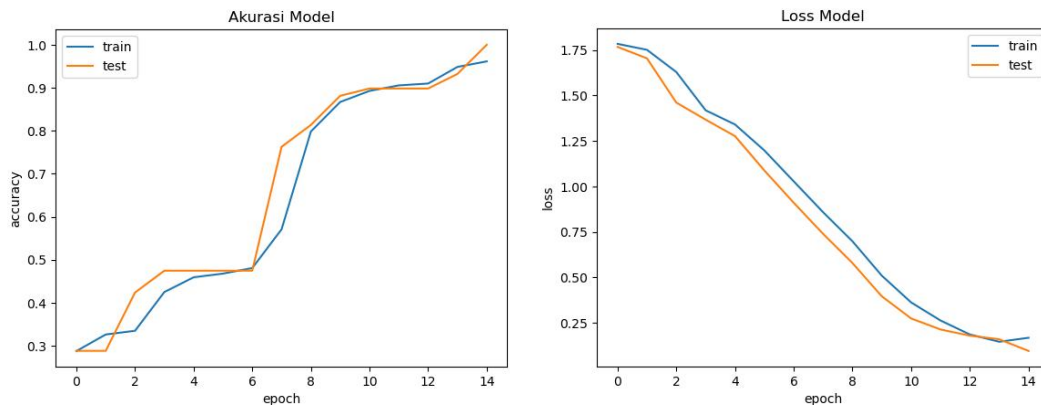
HASIL DAN PEMBAHASAN

Setelah merancang sistem pembelajaran mesin dan menyaring bagian pada berkas riwayat hidup, sistem akan melakukan prediksi terhadap bagian kemampuan dari berkas riwayat hidup dan mengklasifikasikan bagian tersebut masuk kedalam jenis pekerjaan apa berdasarkan model pembelajaran mesin yang sudah dibuat.



Gambar 3. Alur Proses

Dalam membuat model pada sistem pembelajaran mesin, haruslah memilih parameter yang tepat karena penggunaan parameter yang kurang tepat akan mengakibatkan model kurang optimal. Untuk melihat model apakah sudah optimal atau tidak, dapat dilihat dari hasil grafik setelah model melakukan *epoch*. *Epoch* adalah masa pelatihan dari model yang sedang dibuat, semakin panjang dan semakin meningkat masa pelatihan, maka hasil model akan semakin optimal. Berikut adalah hasil *epoch* pada model menggunakan pemrosesan bahasa alami. Semakin tingginya akurasi suatu model, maka model tersebut semakin optimal, begitupun untuk nilai error (*loss*), semakin rendah nilai error maka akan semakin optimal model tersebut.



Gambar 4. Grafik Hasil Pelatihan

Setelah mendapat model yang optimal, maka model harus disimpan menjadi berkas berekstensi h5. Berkas h5 adalah berkas yang hanya bisa dibaca menggunakan *library* `tf.keras.model`, yang mana *library* tersebut akan membaca dan menjalankan model yang sudah dibuat pada sistem pembelajaran mesin.

```

1 model.save('klasifikasipekerjaan.h5')
2 loaded_model = tf.keras.models.load_model('./klasifikasipekerjaan.h5')
    
```

Gambar 5. Penyimpanan Model

Setelah sistem pembelajaran mesin sudah siap, selanjutnya jalankan berkas penyaringan riwayat hidup yang sudah dibuat kedalam API. Pada penelitian kali ini saya memakai layanan Google Drive sebagai media mengunggah dan menyimpan berkas riwayat hidup. Setelah berkas riwayat hidup diunggah, berkas harus diambil dengan memakai tautan API Google Drive. Langkah awal dalam melakukan prediksi dengan menjalankan API yang sudah memakai *framework* Flask dengan metode “POST”.

```

1 @app.route(
2     "/predict", methods=["POST"]
3 ) # Ini untuk bisa nampilin nama pekerjaan
4 def predict():
5     data = request.form
6     url = data.get("cv_url")
7     submitted_data = []
8     submitted_data.append(url)
9
10    print("Checkpoint 1: URL SUDAH MASUK")
11
12    api_url = eval(repr(submitted_data[0]))
13
14    response = requests.get(api_url)
15    with open("./file.pdf", "wb") as file:
16        file.write(response.content)
    
```

Gambar 6. Metode Flask

Kode API ini akan menjalankan kode untuk penyaringan berkas riwayat hidup. Gambar dibawah ini adalah kode untuk mengambil serta menjalankan sistem pembelajaran mesin yang sudah dirancang sebelumnya tanpa harus menjalankannya dengan manual. Pada akhir kode adalah alamat *server* lokal yang akan dipakai untuk memrediksi berkas riwayat hidup.

```

1 if __name__ == "__main__":
2     nltk_data_path = "nltk_data"
3     nltk.data.path.append(nltk_data_path)
4     print("CHECKPOINT: NLTK SUDAH BERHASIL")
5     model_path = "D:\LAPORAN\klasifikasipekerjaan.h5"
6     model = tf.keras.models.load_model(model_path)
7     app.run(host="0.0.0.0", port="5000")
    
```

Gambar 7. Inisiasi Model Pembelajaran Mesin

Setelah berhasil menjalankan kode api penyaringan berkas riwayat hidup, jalankan kode yang akan memunculkan hasil prediksi pekerjaan dan rekomendasi kursus *online*.

```

1 import requests
2 url = "http://127.0.0.1:5000/predict"
3 data = {
4     'cv_url' : "tautan API"
5 }
6 r = requests.post(url, data)
7
    
```

Gambar 8. Kode untuk Melihat Prediksi

Pada bagian ‘tautan API’ dapat diubah menjadi tautan API Google Drive. Setelah itu jalan kode, maka akan muncul hasil prediksi dan bagian berkas riwayat hidup yang sudah melewati proses penyaringan.

```

r.content
✓ 0.0s
b'({"hasil_prediksi": "Data Scientist", "text_output": "skill public speaking teamwork problem solving machine learning python sql analysis data understandi
    
```

Gambar 9. Hasil Prediksi dan Penyaringan

Selanjutnya, gunakan metode *rule-based model* dalam penentuan rekomendasi kursus online berdasarkan jenis pekerjaan yang sudah muncul. Gambar dibawah ini adalah luaran dari kode penentuan rekomendasi kursus *online*.

```

1 def recommend_course(job_title):
2     if job_title == "Data Scientist":
3         print("Kami Merekomendasikan Kursus Online Untuk Data Scientist:")
4         print("- Python Tutorial for Beginners: https://www.youtube.com/watch?v=8124kv-632k&ab_channel=freeCodeCamp.org")
5         print("- Machine Learning Playlist: https://youtube.com/playlist?list=PLblh5JKOoLUICaGLRoHQDuF_7q2GfuJF")
6     elif job_title == "Business Analyst":
7         print("Kami Merekomendasikan Kursus Online Untuk Business Analyst:")
8         print("- Business Analytics Specialization: https://www.coursera.org/specializations/business-analytics")
9         print("- Google Data Analytics Professional Certificate: https://www.coursera.org/professional-certificates/google-data-analytics")
10    elif job_title == "DevOps Engineer":
11        print("Kami Merekomendasikan Kursus Online Untuk DevOps Engineer:")
12        print("- DevOps Engineering Course for Beginners: https://www.youtube.com/watch?v=j5Zsa_e0XeY&ab_channel=freeCodeCamp.org")
13        print("- The Complete DevOps Engineer Course 2.0 - Java & Kubernetes: https://www.udemy.com/course/the-complete-devops-engineer-course-20-java-kubern")
14    elif job_title == "Java Developer":
15        print("Kami Merekomendasikan Kursus Online Untuk Java Developer:")
16        print("- Java Programming for Complete Beginners: https://www.udemy.com/course/java-programming-tutorial-for-beginners/")
17        print("- Java Programming and Software Engineering Fundamentals Specialization: https://www.coursera.org/specializations/java-programming")
18    elif job_title == "Operations Manager":
19        print("Kami Merekomendasikan Kursus Online Untuk Operations Manager:")
20        print("- Operations Management A-Z: Business Processes and Systems: https://www.udemy.com/course/operations-management-a-z/")
21        print("- Introduction to Operations Management: https://www.coursera.org/learn/wharton-operations")
22    elif job_title == "Web Designer":
23        print("Kami Merekomendasikan Kursus Online Untuk Web Designer:")
24        print("- Web Design for Everybody: Basics of Web Development & Coding Specialization: https://www.coursera.org/specializations/web-design")
25        print("- HTML, CSS, and Javascript for Web Developers: https://www.coursera.org/learn/html-css-javascript-for-web-developers")
26    else:
27        print("Job title not found.")
28
29
30 print("Kami Merekomendasikan Anda Menjadi Seorang:", job_title)
31 recommend_course(job_title)
32

```

Gambar 10. *Rule-Based Model* Penentuan Kursus *Online*

```

Kami Merekomendasikan Anda Menjadi Seorang: Data Scientist
Kami Merekomendasikan Kursus Online Untuk Data Scientist:
- Python Tutorial for Beginners: https://www.youtube.com/watch?v=8124kv-632k&ab_channel=freeCodeCamp.org
- Machine Learning Playlist: https://youtube.com/playlist?list=PLblh5JKOoLUICaGLRoHQDuF_7q2GfuJF

```

Gambar 11. Hasil Prediksi dan Penentuan Kursus *Online*

Terlihat bahwa pada gambar 9, bagian kemampuan (*skill*) yang sudah melewati penyaringan dan prediksi sistem pembelajaran mesin mendapatkan luaran yang sesuai dengan basis data (*database*). Dengan adanya rekomendasi kursus *online*, pengguna dapat memperdalam pengetahuan dan meningkatkan kemampuan kedepannya.

PENUTUP

Penggunaan sistem pembelajaran mesin ataupun pembelajaran mendalam maupun keduanya sangatlah membantu kehidupan sehari-hari. Dalam kasus ini, penggunaan pembelajaran mesin dan pembelajaran mendalam serta memakai model pemrosesan bahasa alami sangatlah membantu peneliti dalam mengklasifikasikan pekerjaan dan merekomendasikan kursus *online* yang cocok untuk pengguna. Sistem yang dibuat dalam penelitian ini sudah optimal, terlihat dari nilai akurasi dan nilai error (*loss*), serta luaran berupa jenis pekerjaan dan rekomendasi kursus *online* sesuai dengan harapan penulis. Pembuatan sistem dalam penelitian ini adalah sistem sederhana dari salah satu sistem ATS (*Applicant Tracking System*) yang sudah dipakai pada dunia pekerjaan. Keterbatasan dalam sistem dalam penelitian ini adalah basis data yang perlu ditambah, menambahkan proses pencocokan misalnya dengan cosinus (*cosine similiarity*), dan dapat diterapkan kedalam bentuk *website* maupun aplikasi *mobile*. Peneliti berharap kedepannya sistem ini dapat dikembangkan dan bisa disebarakan ke masyarakat umum, agar semakin memperluas jejaring keilmuan dan saran pekerjaan yang cocok untuk pengguna.

DAFTAR RUJUKAN

- [1] B. K. Natarajan, Machine learning: a theoretical approach. Elsevier, 2014.
- [2] E. Alpaydin, "Introduction to Machine Learning. [SI]." The MIT Press, 2010.
- [3] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Springer, 2019, pp. 59–64.
- [4] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
- [5] E. Klein, E. Loper, and S. Bird, "Natural language processing with python O'reilly media Inc," 2009.

- [6] G. Dutta, “Resume Dataset | Kaggle.” <https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset> (accessed Jun. 20, 2023).
- [7] I. M. Nasser and A. H. Alzaanin, “Machine Learning and Job Posting Classification: A Comparative Study,” vol. 4, pp. 6-14, Sep 2020.