



SISTEM PREDIKSI KURSUS ONLINE & JOB DI INTELLIGO ID MENGUNAKAN RANDOM FOREST

Diki Wahyu Nugraha¹

¹⁾ *Universitas Logistik dan Bisnis Internasional*

¹⁾ dikiwahyu@ulbi.ac.id

INFO ARTIKEL	ABSTRAK
<p>Diterima 07 Agustus 2024 Direvisi 23 Agustus 2024 Diterima 25 Agustus 2024 Tersedia online 12 September September 2024</p> <p>Kata Kunci: Sistem, Prediksi, Kursus, Job, Random Forest.</p> <p>Keywords: <i>System, Prediction, Course, Job, Random Forest.</i></p>	<p>Penelitian ini bertujuan untuk mengetahui pengaruh lingkungan kerja fisik dan E-learning merupakan salah satu contoh dari pemanfaatan teknologi pada jaman sekarang, e-learning dapat dikembangkan ke dalam berbagai hal yang tentunya bertujuan untuk mendukung proses pembelajaran. Salah satu contoh perkembangan e learning adalah sistem prediksi online, sistem prediksi online merupakan bentuk dari e-learning yang digunakan untuk menyajikan bahan pembelajaran secara online. Di intelligo Id menyediakan kursus online di bidang web development dan data science dan ada beberapa kategori dari kursus tersebut, penelitian ini bertujuan untuk membangun sistem kursus online sebagai wadah untuk user agar dapat memprediksi kategori kelas yang ada di intelligo Id serta membantu admin intelligo Id dalam memprediksi lulusan bootcamp yang ada di intelligo Id. Agar dapat memprediksi kategori tipe kelas dan memprediksi job yang ada pada Intellgo Id pada sistem ini menggunakan metode Random Forest, karena dengan menggunakan metode atau algoritma Random Forest mendapatkan nilai akurasi 80%, dengan model yang dibangun dapat di implementasikan kedalam framwork flask dan menghasilkan web yang dapat memprediksikan kursus online dan juga job.</p> <p><i>ABSTRACT</i></p> <p>E-learning is one example of the use of technology in today's era, e-learning can be developed into various things that certainly aim to support the learning process. One example of the development of e-learning is an online prediction system, an online prediction system is a form of e-learning that is used to present learning materials online. In intelligo Id provides online courses in the fields of web development and data science and there are several categories of these courses, this study aims to build an online course system as a place for users to be able to predict class categories in intelligo Id and help intelligo Id admins in predicting bootcamp graduates in intelligo Id. In order to predict class type categories and predict jobs in Intellgo Id, this system uses the Random Forest method, because by using the Random Forest method or algorithm it gets an accuracy value of 80%, with the model that is built it can be implemented into the flask framework and produce a web that can predict online courses and also jobs.</p>

Corresponding author: ¹⁾ dikiwahyu@ulbi.ac.id

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi pada zaman sekarang dapat mempermudah seseorang untuk bisa mendapatkan ilmu pengetahuan atau keterampilan dari mana saja dan kapan saja, salah satu contoh dari pemanfaatan teknologi informasi tersebut mulai di terapkannya *e-learning* [1]. *E-learning* merupakan salah satu contoh dari pemanfaatan teknologi pada zaman sekarang, *e-learning* dapat dikembangkan ke dalam berbagai hal yang tentunya bertujuan untuk mendukung proses pembelajaran [2]. Salah satu contoh perkembangan *e-learning* adalah sistem kursus online, sistem kursus online merupakan bentuk dari *e-learning* yang digunakan untuk menyajikan bahan pembelajaran secara online [3].

Intellgo Id merupakan wadah dalam *learning* dan *assistand* dalam proses belajar dan IT *service* khususnya di bidang *data science*. Pada kursus yang ada di intelligo tersebut terdapat kategori atau tipe kelas. Permasalahan yang di angkat dalam penelitian ini yaitu dapat memprediksikan user mengenai kategori atau tipe kelas yang ada di intelligo id dan membantu admin intelligo id dalam merekomendasikan job pada lulusan bootcamp yang ada di intelligo id, dan pada kesempatan Tugas Akhir ini, saya ingin membangun sistem seperti judul pada laporan di atas “SISTEM PREDIKSI KURSUS ONLINE & JOB DI INTELLIGO ID MENGGUNAKAN *RANDOM FOREST*” dimana sistem ini dibangun untuk user yang akan mengikuti kelas agar mendapatkan memprediksi kelas yang ada di intelligo id serta dapat memprediksi Job pada lulusan bootcamp di intelligo id, untuk dapat memprediksi kelas dan job yang sesuai dengan user, menggunakan inputan dari user sehingga output nya memprediksi kelas sesuai skill dan kemampuan dari user. Untuk sistem prediksi ini, metode yang digunakan adalah algoritma dari *Random Forest*, karena metode *Random Forest* tersebut mempunyai nilai atau persentase akurasi yang lebih tinggi [4].

1.2 Identifikasi Masalah

Berdasarkan Latar belakang, maka dapat mengindetifikasi masalah sebagai berikut :

1. Bagaimana membangun sistem kursus online dan Job di Intellgo ID?
2. Bagaimana membangun sistem yang dapat memprediksi kelas kursus online & job di

Intelligo ID?

3. Bagaimana menerapkan metode *Random Forest* pada sistem ini ?

1.3 Tujuan dan Manfaat

Berdasarkan Latar Belakang dan Masalah-masalah yang ada, maka tujuan dari penelitian ini adalah :

1. Dapat membangun sistem kursus online dan job di Intelligo ID.
2. Dapat membangun sistem yang dapat memprediksikan kelas kursus online dan job di Intelligo ID.
3. Dapat menerapkan metode *Random Forest* pada sistem ini.

1.4 Ruang Lingkup

Sistem ini sebagai wadah bagi user yang akan mengambil kelas kursus online di Intelligo ID, dari data yang di inputkan user sistem dapat merekomendasikan kategori kelas yang cocok sesuai skill dan kemampuan dari user tersebut.

2. LANDASAN TEORI

2.1 E-Learning

E-Learning, atau pembelajaran elektronik, merupakan suatu metode pembelajaran yang menggunakan teknologi informasi dan komunikasi untuk menyampaikan materi pelajaran secara digital [5]. Dengan adanya *E-Learning*, siswa dapat mengakses materi pembelajaran dari mana saja dan kapan saja melalui perangkat elektronik seperti komputer, tablet, atau ponsel cerdas [6]. Sebagai contoh, David Wiley, seorang pendidik dan peneliti terkenal, menyatakan, "*E-Learning offers the individual learner unprecedented opportunities for accessing relevant, up-to-date information and learning resources at their own pace and in their own space*" [7]. Pernyataan ini mencerminkan keunggulan *E-Learning* dalam memberikan peluang kepada pembelajar untuk mengakses informasi yang relevan dan sumber daya pembelajaran sesuai dengan ritme dan ruang pribadi mereka, oleh karena itu pembelajaran elektronik perlu dikembangkan dengan bijaksana untuk memastikan kualitas dan efektivitasnya.

2.2 Sistem Rekomendasi

Recommendation system merupakan program yang bertindak sebagai perantara membuat daftar informasi yang dibutuhkan dan membandingkan kecocokan antara item dengan pengguna [8]. Skema rekomendasi kini menjadi sangat penting di berbagai bidang, termasuk Rekomendasi Kursus Online sejumlah layanan berbasis web [9]. Contohnya sistem rekomendasi film yang bisa

memprediksi film apa yang akan disukai pengguna berdasarkan atribut yang ada di film yang sebelumnya disukai [10]. Sebagian *Movie recommendation* menggunakan model *Collaborative filtering* (CF) seperti genre atau pemeran. Sistem rekomendasi biasanya melibatkan pemanfaatan hubungan antara fitur dan konten yang diketahui [11].

2.3 Kursus

Kursus dalam konteks pendidikan adalah suatu rangkaian pembelajaran terstruktur yang dirancang untuk memberikan pemahaman mendalam tentang suatu topik atau keterampilan tertentu [12]. Kursus dapat diselenggarakan di berbagai tingkatan, mulai dari tingkat dasar hingga tingkat lanjutan, dan bisa mencakup berbagai disiplin ilmu, mulai dari ilmu pengetahuan alam hingga seni. Kursus ini sering kali diarahkan oleh instruktur yang kompeten di bidangnya dan dapat diselenggarakan dalam berbagai format, seperti kelas tatap muka, pembelajaran online, atau kombinasi keduanya [13]. Kursus memiliki tujuan untuk menyediakan pengalaman pembelajaran yang terstruktur dan terpandu, membantu peserta didik mencapai pemahaman yang lebih baik tentang subjek tertentu. Selain itu, kursus juga memberikan platform bagi peserta untuk berinteraksi dengan sesama peserta dan instruktur, memfasilitasi pertukaran ide dan pemahaman yang lebih mendalam. Kursus sering kali memanfaatkan berbagai metode pengajaran, seperti kuliah, diskusi kelompok, tugas, dan ujian, untuk mengukur pemahaman dan kemajuan peserta.

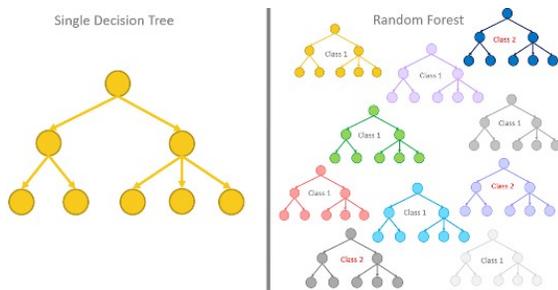
2.4 Decision Tree

Pohon Keputusan (Decision Tree) adalah salah satu metode dalam machine learning yang digunakan untuk tugas klasifikasi dan regresi [15]. Konsep dasar dari Decision Tree adalah membagi dataset menjadi subset yang semakin kecil berdasarkan serangkaian keputusan atau aturan [16]. Pada setiap simpul (node) dalam pohon, dilakukan pemilihan fitur yang paling baik untuk memisahkan data berdasarkan kriteria tertentu. Proses ini berlanjut hingga mencapai simpul daun (leaf node) yang menyatakan prediksi atau nilai yang dihasilkan oleh pohon untuk suatu *instance* data tertentu. Salah satu keunggulan utama dari Decision Tree adalah interpretabilitasnya yang tinggi. Model ini dapat divisualisasikan dengan mudah, sehingga memungkinkan analis dan pengambil keputusan untuk memahami logika pengambilan keputusan dengan jelas [22]. Selain itu, Decision Tree juga dapat menangani data kategorikal dan numerikal tanpa memerlukan transformasi khusus. Namun, satu tantangan utama adalah

kecenderungan untuk *overfitting*, yaitu ketika model terlalu kompleks dan sangat cocok dengan data pelatihan, tetapi tidak dapat menggeneralisasi dengan baik pada data baru.

2.5 Random Forest

Random Forest adalah sebuah metode dalam *machine learning* yang termasuk dalam kategori *ensemble learning*. *Ensemble learning* menggabungkan beberapa model prediksi untuk meningkatkan kinerja dan ketepatan prediksi dibandingkan dengan penggunaan satu model saja [23]. *Random Forest* bekerja dengan membangun sejumlah besar pohon keputusan selama proses pelatihan. Setiap pohon keputusan dibangun secara acak dengan menggunakan subset acak dari data pelatihan dan subset acak dari fitur-fitur yang tersedia. Dengan menggunakan keputusan mayoritas dari semua pohon keputusan yang terbentuk, *Random Forest* dapat memberikan prediksi yang stabil dan akurat [24]. Salah satu keunggulan utama dari *Random Forest* adalah kemampuannya untuk mengatasi *overfitting*, yaitu kecenderungan model untuk terlalu spesifik terhadap data pelatihan dan kurang mampu melakukan generalisasi pada data baru. Dengan cara menggabungkan banyak pohon keputusan yang dibangun secara acak, *Random Forest* dapat meningkatkan ketahanannya terhadap *overfitting*. Selain itu, metode ini juga dapat menangani dataset besar dengan fitur-fitur yang beragam dan dapat memberikan estimasi pentingnya masing-masing fitur dalam membuat prediksi [25].



Gambar 2.1 Decision Tree vs Random Forest

Dari gambar diatas bisa dilihat bawasanya ada sedikit perbedaan antara *Decision Tree* dan *Random Forest*, dimana *Decision Tree* hanya membuat satu pohon keputusan saja sedangkan *Random Forest* terdiri dari beberapa *Decision Tree* (Pohon keputusan).

Langkah-langkah *Random Forest*:

1. Pilih Dataset: Pilih dataset yang sesuai dengan masalah yang ingin di pecahkan. Dataset ini tersebut harus terstruktur dengan fitur-fitur yang ada dan label yang sesuai.
2. Bagi Dataset: Bagi dataset menjadi dua bagian, yaitu data latih (training data) dan data uji (testing data). Data latih akan digunakan untuk melatih model, sedangkan data uji akan digunakan untuk menguji performa model yang telah dilatih.
3. Pembuatan Pohon Keputusan : Random Forest terdiri dari banyak pohon keputusan atau terdiri dari beberapa Decision Tree. Setiap pohon keputusan dibuat menggunakan subset acak dari data latih. Pohon-pohon ini akan memiliki struktur yang mirip namun tidak sama.
4. Latih Pohon Keputusan: Setiap pohon keputusan dilatih menggunakan teknik seperti pembagian data berdasarkan fitur yang paling informatif pada setiap levelnya, dengan tujuan untuk meminimalkan kesalahan prediksi.
5. Agregasi Prediksi: Setelah semua pohon keputusan dilatih, prediksi dari setiap pohon akan diambil. Dalam kasus klasifikasi, akan dilakukan pemilihan mayoritas kelas.
6. Evaluasi Model: Gunakan data uji untuk mengevaluasi performa model Random Forest yang telah dibuat.

Jika dilihat dari penjelasan diatas Random Forest hampir sama dengan Decision Tree namun Random Forest notabeneanya mempunyai nilai akurasi yang lebih tinggi daripada Decision Tree, terkadang di suatu kondisi Decision Tree lebih dibutuhkan dibandingkan Random Forest walaupun Random Forest mempunyai nilai atau persentase akurasi yang lebih tinggi, adapun beberapa perbedaan atau kondisi yang memungkinkan dalam penggunaan dua model tersebut :

1. *Decision Tree* :

- a. Data Sederhana : *Decision Tree* mudah dipahami dan diinterpretasikan, cocok untuk data dengan struktur yang jelas dan hubungan antar fitur yang mudah dipahami.
- b. Kecepatan : *Decision Tree* relatif cepat untuk dilatih dan diprediksi, ideal untuk situasi di mana kecepatan dan efisiensi penting.
- c. Interpretasi : *Decision Tree* memberikan visualisasi yang jelas tentang proses pengambilan keputusan, membantu memahami faktor-faktor yang mempengaruhi prediksi.

2. *Random Forest* :

- a. Akurasi : *Random Forest* umumnya lebih akurat daripada *Decision Tree*, terutama untuk data kompleks dengan banyak fitur dan hubungan *non-linear*.
- b. *Overfitting* : *Random Forest* membantu mengurangi *overfitting* dengan menggabungkan banyak pohon keputusan, menghasilkan model yang lebih stabil dan *generalizable*.
- c. Data Kompleks : *Random Forest* dapat menangani data dengan *noise* dan *missing values* dengan lebih baik daripada *Decision Tree*.

Karena pada sistem ini membutuhkan akurasi, maka model algoritma yang dipakai yaitu *Random Forest*.

2.6 Machine Learning

Machine learning (ML) adalah cabang dari kecerdasan buatan yang fokus pada pengembangan algoritma dan model statistik yang memungkinkan sistem komputer untuk belajar dari data dan meningkatkan kinerjanya seiring waktu tanpa perlu pemrograman eksplisit. Pendekatan ini memungkinkan mesin untuk mengidentifikasi pola dan tren dalam data, membuat prediksi, dan mengambil keputusan tanpa keterlibatan manusia yang terlalu mendalam. Ada beberapa tipe utama machine learning, termasuk supervised learning, unsupervised learning, dan reinforcement learning, masing-masing dirancang untuk menangani berbagai jenis tugas dan scenario [27]. *Supervised learning* melibatkan pelatihan model pada data yang sudah dilabeli, di mana model belajar membuat prediksi atau klasifikasi dengan membandingkan outputnya dengan label yang benar. Sebaliknya, unsupervised learning beroperasi tanpa label, dan model mencoba mengidentifikasi pola atau struktur dalam data tanpa panduan eksternal. Sementara itu, reinforcement learning melibatkan pembelajaran melalui pengalaman, di mana agen atau model belajar membuat keputusan berdasarkan umpan balik yang diberikan sebagai hasil dari tindakan yang diambilnya. *Machine learning* telah merasuki berbagai aspek kehidupan sehari-hari, termasuk pengenalan wajah, penerjemahan bahasa otomatis, dan rekomendasi produk. Keberhasilan *machine learning* sangat bergantung pada kualitas data yang digunakan untuk melatih model, dan pengembang terus mengupayakan metode dan algoritma baru untuk meningkatkan akurasi dan efisiensi sistem ini. Meskipun memberikan berbagai manfaat, perlu juga mempertimbangkan aspek etika dan privasi dalam penggunaan machine learning untuk

memastikan bahwa perkembangan teknologi ini berjalan seiring dengan nilai-nilai masyarakat dan norma etika.

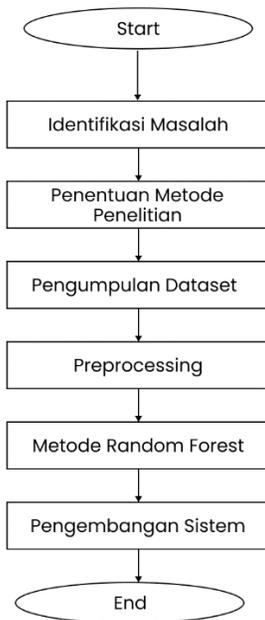
2.7 Web Development

Web development, atau pengembangan web, adalah proses pembuatan dan pemeliharaan situs web atau aplikasi web [28]. Ini melibatkan serangkaian tugas yang mencakup perancangan, pengembangan, dan penyesuaian elemen-elemen seperti tata letak, fungsionalitas, dan interaktivitas situs. Ahli *web development*, yang dikenal sebagai pengembang web, menggunakan berbagai bahasa pemrograman dan kerangka kerja untuk menciptakan pengalaman pengguna yang responsif dan efektif. Pengembangan web dibagi menjadi dua kategori utama yaitu *frontend* dan *backend*. *Frontend* mencakup semua aspek yang terlihat oleh pengguna, termasuk antarmuka pengguna, tata letak, dan elemen desain lainnya. Para pengembang *frontend* menggunakan bahasa pemrograman seperti *HTML*, *CSS*, dan *JavaScript* untuk menciptakan pengalaman pengguna yang menarik dan interaktif. Sementara itu, *backend* melibatkan pengelolaan server, database, dan logika aplikasi yang tidak terlihat oleh pengguna. Pengembang *backend* menggunakan bahasa seperti *Python*, *Ruby*, dan *Java*, serta kerangka kerja seperti *Django*, *Ruby on Rails*, dan *Spring* untuk memastikan fungsionalitas yang tepat dan efisien di balik layer [29]. Proses pengembangan web mencakup langkah-langkah seperti perencanaan, desain, pengkodean, pengujian, dan pemeliharaan. Perubahan dinamis dalam teknologi web memaksa para pengembang untuk terus memperbarui keterampilan mereka dan mengikuti tren terbaru.

METODE PENELITIAN

3.1 Diagram Alur Metodologi Penelitian

Tahapan dari metode penelitian untuk menciptakan Sistem Rekomendasi Kursus online dan Lulusan Bootcamp di Intelligo.id menggunakan metode *Random Forest Regression*.



Gambar 3. 1 Alur Diagram Metode Penelitian

1. Identifikasi Masalah

Tahap pertama dalam penelitian merupakan identifikasi masalah yaitu penulis menganalisis permasalahan yang biasa terjadi ketika akan mengambil tipe kelas yang ada di intelligo id. Meninjau dari permasalahan yang biasa terjadi maka akan lebih baiknya ada sebuah sistem rekomendasi kursus online di intelligo id, agar dapat membantu calon peserta dalam mengambil tipe kelas.

2. Penentuan Metode Penelitian

Tahap kedua yaitu penentuan metode penelitian pada sistem rekomendasi yang akan dibangun, pada penelitian ini penulis menggunakan metode *Random Forest Regression*.

3. Pengumpulan Dataset

Pada tahap ke tiga ini yaitu pengumpulan dataset serta melakukan Analisa data dan evaluasi kualitas data yang akan digunakan dalam pengerjaan atau pembuatan penelitian ini. Pemahaman mengenai data penelitian ini berdasarkan kebutuhan metode-metode yang mendukung dalam penelitian sistem prediksi kursus online dan juga prediksi lulusan bootcamp di intelligo id, maka peneliti menyiapkan data kelas yang ada di intelligo dan penjabaran dari data kelas yang ada dari LinkedIn termasuk deskripsi, tools, dan skill. Sehingga banyak data yang terkumpul yaitu 8 kelas.

4. Preprocessing

Tahap *Preprocessing* adalah tahap mengubah bentuk data yang sebelumnya yang belum terstruktur menjadi data terstruktur menyesuaikan kebutuhan dalam sistem prediksi kursus online dan prediksi lulusan bootcamp di intelligo id yang akan dibangun, seperti menggabungkan beberapa kolom menjadi satu kolom, memisahkan nilai pada kolom yang digunakan, dan membersihkan data mulai dari kode unik dan nilai *NULL* pada kolom yang digunakan, serta menambahkan lebel pada suatu kolom, dari data yang terkumpul bisa dikatakan mempunyai 4 fitur, yaitu nama kelas, deskripsi, skill, dan tools.

Adapun penjelasan dari fitur-fitur tersebut adalah, nama kelas merupakan jenis kelas yang ada di intelligo id, deskripsi penjelasan dari setiap kelas yang ada, skill merupakan kemampuan yang dikuasai disetiap kelas yang ada, dan yang terakhir tools merupakan peralatan yang mempuni pada setiap kelas yang ada. Namun setelah melalui tahap *preprocessing* skill dan tools digabungkan menjadi 1 fitur di deskripsi dan banyak data sebanyak 212 data.

5. Metode *Random Forest Regression*

Random Forest merupakan pengklasifikasi yang bersifat *esemble*, dan *random forest* akan menciptakan hutan atau *forest* menggunakan sejumlah pohon keputusan atau *decision tree* jumlah suara terbanyak (voting) dari seluruh pohon keputusan akan digunakan untuk menentukan hasil prediksi dari prediksi kelas dan job.

Dalam pembuatan model *Random Forest Classifier* ini menggunakan parameter *default*, dimana parameter default yang dimaksud adalah $n_estimators$ dan $criterion='gini'$. Parameter $n_estimators$ ini adalah seberapa banyak pohon keputusan yang akan dibuat pada model ini, namun karna yang digunakan default maka otomatis pohon keputusan yang dibuat pada *Random Forest Classifier* 100 Tree ($n_estimators=100$). Sedangkan $criterion='gini'$ digunakan untuk mengukur impurity atau seberapa seragam kategori target dalam sebuah himpunan data, semakin seragam atau sama kategori target, semakin rendah nilai *gini impurity* dan semakin murni atau bersih himpunan data tersebut.

$$\text{Gini}(p) = 1 - \sum_{i=1}^K p_i^2$$

K : Kategori Target

P_i : Sample Kategori Target

i : Total *Sample*

Kedalaman *Tree* atau pohon keputusan pada model ini sesuai parameter default yang disebutkan di atas yaitu 100 *Tree* (pohon keputusan), dari 100 pohon keputusan tersebut diambil label terbanyak yang menjadi nilai prediksinya.

6. Pengembangan Sistem

Pengembangan sistem merupakan tahap dimana sistem yang dibuat akan dirancang menuju pada tujuan dari pembuatan artikel ilmiah ini sehingga benar benar menjadikan sebuah sistem rekomendasi. Pada tahap pengembangan sistem terdapat analisis sistem dan desain sistem.

1. Analisis Sistem

Sistem rekomendasi kursus online dan lulusan bootcamp intelligo id ini dapat melakukan beberapa hal seperti :

(a) Identifikasi Masalah

Berdasarkan latar belakang serta pembahasan di atas, maka permasalahan yang dapat di uraikan yaitu bagaimana menerapkan metode *Random Forest*

dengan pendekatan *Random Forest Regression* pada sistem rekomendasi kursus online dan rekomendasi lulusan bootcamp di intelligo id.

(b) Analisis Kebutuhan Fungsional

1. *User* atau calon peserta dapat mencari rekomendasi kelas yang ada di intelligo id, dengan form di menu rekomendasi kelas yang ada di menu *website* intelligo id.
2. *User 2* atau instansi yang berelasi di intelligo id dapat melihat rekomendasi peserta yang lulus dari bootcamp intelligo berdasarkan skill yang menonjol dari peserta tersebut.

Sistem dapat memberikan atau menampilkan hasil rekomendasi dari dua rekomendasi di atas dengan hasil persentase prediksi tertinggi.

HASIL DAN IMPLEMENTASI

4.1 Implementasi

Lingkungan Implementasi

Lingkungan implementasi pada penelitian ini dilakukan oleh penulis pada rencana pembuatan sistem rekomendasi di salah satu *startup* untuk memenuhi kebutuhan pada rekomendasi kursus online dan lulusan bootcamp yang ada di Intelligo Id. Adapun hasil penerapan sistem rekomendasi pekerjaan yang telah dibuat pada penelitian ini dijelaskan pada bagian hasil implementasi beserta pengujiannya.

4.2 Hasil

Bab ini menjawab hal-hal yang telah disampaikan pada bab sebelumnya tentang metodologi penelitian dari tahap *preprocessing, cleaning, model*, dan hasil prediksi.

1. Preprocessing

	nama_kelas	deskripsi	skill	tools
0	Data Scientist	Data Scientist adalah profesional yang memilik...	Pengalaman dalam analisis eksploratif data.\r\n...	Python/\r\nJupyter Notebooks/\r\nPandas/\r\nScikit-le...
1	Data Engineer	Data Engineer bertanggung jawab untuk merancan...	Pengalaman dalam pengelolaan database (SQL, N...	Apache Hadoop/\r\nApache Spark/\r\nSQL/NoSQL dat...
2	Data Analyst	Data Analyst berfokus pada interpretasi data u...	Keterampilan analisis statistik.\r\nPenguasaan a...	Microsoft Excel/\r\nTableau/\r\nPower BI/\r\nSQL...
3	AI Engineer	Artificial Intelligence, AI Engineer merancang...	Pemahaman mendalam tentang machine learning da...	TensorFlow/\r\nPyTorch/\r\nPython/\r\nJupyter Not...
4	Natural Language Processing	Kelas ini membahas aplikasi dan teknik-teknik...	Pemahaman konsep-konsep dasar NLP.\r\nKeteramp...	NLTK (Natural Language Toolkit)/\r\nSpaCy/\r\nWo...
5	Computer Vision	Kelas ini membahas konsep-konsep pengolahan ci...	Pemahaman tentang dasar-dasar pengolahan citra...	OpenCV/\r\nTensorFlow for Computer Vision/\r\nCN...
6	Python Programming	Kelas ini membahas dasar-dasar pemrograman Pyt...	Keterampilan dasar dalam pemrograman Python.\r...	Python/\r\nFlask/\r\nDjango/\r\nHTML/CSS/\r\nGit
7	Business Intelligence	Kelas ini fokus pada penggunaan alat-alat bisn...	Penguasaan alat-alat BI seperti Tableau atau P...	Tableau/\r\nPower BI/\r\nSQL/\r\nMicrosoft Excel...
8	Machine Learning	Kelas ini menyajikan informasi rinci tentang ...	Supervised Learning/\r\nUnsupervised Learning/\r...	Scikit-learn/\r\nJupyter Notebooks/\r\nPython/\r...

Gambar 4. 1 Read CSV

Data awal atau data mentah seperti gambar di atas, terdiri sari 4 kolom dan 9 baris Dimana kolom tersebut terdiri dari nama_kelas, deskripsi, skill, dan tools. Data tersebut nantinya digunakan untuk perbandingan dengan inputan *user* menggunakan model yang pilih.

Karena yang akan digunakan sebagai perbandingan hanya satu kolom yaitu kolom deskripsi, maka kolom skill dan tools digabungin ke kolom deskripsi menjadi satu kolom.

```
# Menggabungkan kolom deskripsi, skill, dan tools menjadi kolom baru full_deskripsi
df['deskripsi'] = df['deskripsi'] + '\n' + df['skill'] + '\n' + df['tools']

# Memisahkan nilai pada full_deskripsi berdasarkan '\n', '\r', atau '.'
df['deskripsi'] = df['deskripsi'].apply(lambda x: '\n'.join(filter(None, [desc.strip() for desc in re.split(r'[\n\r.]', x)])))
```

Gambar 4. 2Menggabungkan kolom

	nama_kelas	deskripsi
0	Data Scientist	Data Scientist adalah profesional yang memilik...
1	Data Engineer	Data Engineer bertanggung jawab untuk merancan...
2	Data Analyst	Data Analyst berfokus pada interpretasi data u...
3	AI Engineer	Artificial Intelligence, AI Engineer merancang...
4	Natural Language Processing	Kelas ini membahas aplikasi dan teknik-teknik ...
5	Computer Vision	Kelas ini membahas konsep-konsep pengolahan ci...
6	Python Programming	Kelas ini membahas dasar-dasar pemrograman Pyt...
7	Business Intelligence	Kelas ini fokus pada penggunaan alat-alat bisn...
8	Machine Learning	Kelas ini menyajikan informasi rinci tentang ...

Gambar 4. 3 Hasil menggabungkan beberapa kolom

Setelah di gabungkan ada perubahan bentuk lagi Dimana kolom nama_kelas beberapa data dijadikan satu otomatis data deskripsi juga menjadi satu berdasarkan baris apa yang digabungkan di kolom nama_kelas.

```
df['nama_kelas'] = df['nama_kelas'].replace('Computer Vision', 'AI Engineer')
df['nama_kelas'] = df['nama_kelas'].replace('Natural Language Processing', 'AI Engineer')
df['nama_kelas'] = df['nama_kelas'].replace('Business Intelligence', 'Data Analyst')
df['nama_kelas'] = df['nama_kelas'].replace('Python Programming', 'Data Scientist')
df['nama_kelas'] = df['nama_kelas'].replace('Machine Learning', 'Data Scientist')
```

Gambar 4. 4 Replace nama kelas

```
# Memisahkan nilai pada kolom deskripsi berdasarkan '\n', ',', dan '.'
df['deskripsi'] = df['deskripsi'].apply(lambda x: [desc.strip() for desc in re.split(r'[\n,\.]', x) if desc.strip()])

# Membuat DataFrame baru dengan menggunakan explode
df = df.explode('deskripsi').reset_index(drop=True)
```

Gambar 4. 5 memisahkan nilai kolom

Setelah dapat menggabungkan kolom tentunya dilanjut dengan memisahkan isi deskripsi, karna di python tidak dapat membaca data yang Panjang seperti deskripsi sebelumnya. Maka kolom deskripsi dipisah berdasarkan \n, ',', dan ".". jadi setiap ada koma dan juga titik di setiap deskripsi akan dipisahkan menjadi baris baru, karna hal ini data terjadi perubahan lagi.

	nama_kelas	deskripsi
0	Data Scientist	Data Scientist adalah profesional yang memilik...
1	Data Scientist	menginterpretasi
2	Data Scientist	dan menghasilkan wawasan dari data
3	Data Scientist	Mereka menggunakan berbagai teknik analisis un...
4	Data Scientist	Seorang Data Scientist adalah profesional mult...
...
207	Data Scientist	Scikit-learn
208	Data Scientist	Jupyter Notebooks
209	Data Scientist	Python
210	Data Scientist	Pandas
211	Data Scientist	Matplotlib/Seaborn

212 rows x 2 columns

Gambar 4. 6 Memisahkan data deskripsi

Dari gambar di atas terlihat data mengalami perubahan di banyaknya baris, karena hal sebelumnya yang dipisahkan berdasarkan \n, ',', dan '.'.

2. Cleaning

Tahap ini merupakan tahap pembersihan data, khususnya data pada kolom deskripsi. Karena pada dasarnya python tidak bisa membaca data yang unik seperti Huruf kapital dan juga kode-kode atau karakter unik pada teks seperti '@', '*', '?' dan banyak lainnya. Maka data deskripsi dibersihkan agar python dapat membaca dan memproses data tersebut.

```

def clean_text(text):
    text = text.lower() #lowercase atau case folding
    text = re.sub('@[^\s]+', '', text) #remove username
    text = re.sub('\[.*?\]', '', text) # remove square brackets
    text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', '', text) # remove URLs
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text) # remove punctuation
    text = re.sub('\w*\d\w*', '', text)
    text = re.sub('[^a-z\d]', '', text)
    text = re.sub('\n', '', text)
    return text

clean1 = lambda x: clean_text(x)

df['deskripsi_clean'] = pd.DataFrame(df['deskripsi'].apply(clean1))

# converting type of columns to 'category'

df['label'] = df['nama_kelas'].astype('category')
df['label'] = df['label'].cat.codes
df

```

Gambar 4. 7 Cleaning data

Dari kode diatas juga memberikan label pada kolom nama kelas, dan berikut hasil data yang telah *cleaning* dan pemberian lebel di setiap kelas.

	nama_kelas	deskripsi	deskripsi_clean	label
0	Data Scientist	Data Scientist adalah profesional yang memilik...	data scientist adalah profesional yang memilik...	3
1	Data Scientist	menginterpretasi	menginterpretasi	3
2	Data Scientist	dan menghasilkan wawasan dari data	dan menghasilkan wawasan dari data	3
3	Data Scientist	Mereka menggunakan berbagai teknik analisis un...	mereka menggunakan berbagai teknik analisis un...	3
4	Data Scientist	Seorang Data Scientist adalah profesional mult...	seorang data scientist adalah profesional mult...	3
...
207	Data Scientist	Scikit-learn	scikitlearn	3
208	Data Scientist	Jupyter Notebooks	jupyter notebooks	3
209	Data Scientist	Python	python	3
210	Data Scientist	Pandas	pandas	3
211	Data Scientist	Matplotlib/Seaborn	matplotlibseaborn	3

212 rows x 4 columns

Gambar 4. 8 Hasil Cleaning

Terlihat jelas perbedaan data deskripsi sebelum di bersihkan dengan sesudah dibersihkan.

3. Modeling

Pada tahap pemodelan disini menggunakan dua model yaitu *Decision Tree* dan juga *Random Forest*, dari dua model tersebut memiliki nilai hasil prediksi yang berbeda namun nantinya diambil satu model yang mempunyai nilai akurasi persentase yang paling tinggi.

```
df['label'].value_counts()

0    89
3    88
1    53
2    39
4    33
Name: label, dtype: int64

from imblearn.over_sampling import RandomOverSampler, SMOTEN, SMOTE, ADASYN
from imblearn.combine import SMOTEENN

vec = TfidfVectorizer().fit(df['deskripsi_clean'])
vec_transform = vec.transform(df['deskripsi_clean'])

X = vec_transform.toarray()
y = df['label']

over = RandomOverSampler(random_state=42)
X_over, y_over = over.fit_resample(X, y)

X_train, X_test, y_train, y_test = train_test_split(X_over, y_over, test_size=0.20, random_state=42)
```

Gambar 4. 9 Modeling

1. Model Random Forest

Random Forest

```
model_rf = RandomForestClassifier()

model_rf.fit(X_train, y_train)
# model_rf.fit(X_over, y_over)

prediksi = model_rf.predict(X_test)

print('Akurasinya adalah: ', model_rf.score(X_test, y_test))
print(metrics.classification_report(y_test, prediksi))
```

	precision	recall	f1-score	support
0	0.75	0.90	0.82	10
1	1.00	1.00	1.00	11
2	1.00	0.95	0.98	21
3	0.90	0.86	0.88	22
4	1.00	1.00	1.00	25
accuracy			0.94	89
macro avg	0.93	0.94	0.94	89
weighted avg	0.95	0.94	0.95	89

Gambar 4. 10 Model random forest

Dari kode di atas terlihat bawasanya pada model yang menggunakan random forest mempunyai nilai akurasi sekitar 94% sehingga yang akan digunakan yaitu model

random forest karena persentase nilai akurasi lebih tinggi daripada model menggunakan decision tree.

2. Save Model

```
import pickle

filename = 'model_rf.pkl'
pickle.dump(model_rf, open(filename, 'wb'))

import pickle

filename = 'vec.pkl'
pickle.dump(vec, open(filename, 'wb'))
```

Gambar 4. 11 Save model

Setelah mengetahui model yang akan digunakan maka model tersebut disimpan menggunakan library pickle, karena model dengan akurasi tertinggi yaitu model random forest jadi yang disimpan model tersebut dengan nama 'model_rf.pkl'.

4. Hasil Prediksi Kelas

```
import joblib

model = joblib.load('model_rf.pkl')
vec = joblib.load('vec.pkl')

print("Rekomendasi Kelas Intelligo ID")

nama = input('Siapa nama kamu? ')
usia = input('Berapa usia kamu? ')
pendidikan_terakhir = input('Apa pendidikan terakhir kamu? ')
pekerjaan = input('Apa pekerjaan di bidang kerja kamu atau jurusan jika kamu mahasiswa? ')
tujuan_kelas = input('Apa tujuan kamu mengambil kelas di Intelligo ID? (Peningkatan karir, Pengembangan skill, Hobi)')
preferensi_belajar = input('Bagaimana kamu lebih suka belajar? (Online, Offline, Kombinasi)')
sumber_info = input('Dari mana kamu mengetahui tentang kelas ini? (Rekomendasi teman, Media sosial, Mesin pencari)')

# Pertanyaan tambahan mengenai preferensi belajar online
if preferensi_belajar.lower() == 'online':
    waktu_belajar = input('Kapan waktu yang paling cocok untukmu belajar online? (Pagi, Siang, Sore, Malam)')

# Pertanyaan tambahan mengenai bidang keahlian atau topik tertentu
topik_spesifik = input('Apakah ada topik atau bidang keahlian tertentu yang ingin kamu pelajari lebih dalam?')

# Pertanyaan tambahan mengenai durasi atau jenis kelas
durasi_kelas = input('Apakah kamu mencari kelas yang singkat atau yang lebih panjang? (Short Course, Full Course)')
jenis_kelas = input('Preferensi terhadap jenis kelas (Kelas interaktif, Video on-demand, Live streaming)')

# Pertanyaan tambahan mengenai anggaran atau biaya
anggaran = input('Berapa anggaran yang kamu siapkan untuk mengikuti kelas ini?')
```

Gambar 4. 12 Simulasi testing

Setelah menyimpan model yang digunakan, berlanjut ke tahap simulai testing diamana di gambar kode di atas model yang di simpan di import yang nantinya akan digunakan, pada gambar diatas juga megambarkan beberpa form yang nantinya di isi oleh user.

```

deskripsi_tes = (
    "Nama: " + nama +
    "\nUsia: " + usia +
    "\nPendidikan Terakhir: " + pendidikan_terakhir +
    "\nPekerjaan: " + pekerjaan +
    "\nTujuan Kelas: " + tujuan_kelas +
    "\nPreferensi Belajar: " + preferensi_belajar
)

if preferensi_belajar.lower() == 'online':
    deskripsi_tes += "\nWaktu Belajar Online: " + waktu_belajar

deskripsi_tes += (
    "\nTopik Spesifik: " + topik_spesifik +
    "\nDurasi Kelas: " + durasi_kelas +
    "\nJenis Kelas: " + jenis_kelas +
    "\nAnggaran: " + anggaran
)

# Menampilkan hasil
print("Deskripsi Tes:")
print(deskripsi_tes)

```

Gambar 4. 13 Deskripsi tes

```

Deskripsi Tes:
Nama: dimas
Usia: 22
Pendidikan Terakhir: Sarjana
Pekerjaan: Manajemen
Tujuan Kelas: peningkatan karir
Preferensi Belajar: online
Waktu Belajar Online: malam
Topik Spesifik: data analyst
Durasi Kelas: full
Jenis Kelas: kelas interaktif
Anggaran: 500rb

input_text = vec.transform([deskripsi_tes]).toarray()
label_rekomendasi = model.predict(input_text)
print(label_rekomendasi)

[3]

```

Gambar 4. 14 contoh inputan user

Gambar diatas menunjukkan contoh dari inputan user nantinya, dan dari data diatas mendapatkan nilai label '3'.

```
# Berdasarkan nilai label, berikan rekomendasi kelas
if label_rekomendasi == 0:
    rekomendasi_kelas = "AI Engineer"
elif label_rekomendasi == 1:
    rekomendasi_kelas = "Data Analyst"
elif label_rekomendasi == 2:
    rekomendasi_kelas = "Data Engineer"
elif label_rekomendasi == 3:
    rekomendasi_kelas = "Data Scientist"
elif label_rekomendasi == 4:
    rekomendasi_kelas = "Python Programming"
else:
    rekomendasi_kelas = "Tidak ada rekomendasi kelas yang sesuai dengan label yang diberikan"

# Tampilkan rekomendasi kelas
print("Rekomendasi kelas:")
print(f"Berdasarkan preferensi Anda, kami merekomendasikan Anda untuk mengikuti kelas: {rekomendasi_kelas}")
```

Rekomendasi kelas:
Berdasarkan preferensi Anda, kami merekomendasikan Anda untuk mengikuti kelas: Data Scientist

Gambar 4. 15 hasil prediksi

Karena nilai yang didapat dari inputan user yaitu label 3 maka dapat di rekomendasikan bawasanya user tadi cocok untuk mengambil kelas pada label 3 (Data Science).

```
probas = model.predict_proba(input_text)
top_n_labels_idx = np.argsort(-probas)
top_n_probs = np.round(-np.sort(-probas),3)
top_n_labels = [model.classes_[i] for i in top_n_labels_idx]

results = list(zip(top_n_labels, top_n_probs))

pd.DataFrame(results)
```

	0	1
0	[3, 1, 0, 2, 4]	[0.38, 0.28, 0.18, 0.12, 0.04]

Gambar 4. 16 nilai skor

Kode di atas digunakan untuk mengetahui skor dari label yang di dapat, bisa di lihat di bagian hasil label 3 mempunyai skor paling tinggi dari label-label lainnya. Maka dari itu di atas di katakana bahwa user direkomendasikan ke kelas label 3.

```

hasil_sort = pd.DataFrame(results)
hasil_sort.rename(columns={0 : 'recommendation', 1 : 'probability'}, inplace=True)

hasil_sort = hasil_sort.explode(['recommendation', 'probability']).reset_index(drop=True)

hasil_sort

```

	recommendation	probability
0	3	0.38
1	1	0.28
2	0	0.18
3	2	0.12
4	4	0.04

Gambar 4. 17 mengubah bentuk data skor

```

# Membuat dictionary mapping antara label dan nama_kelas
label_to_kelas = {
    0: 'AI Engineer',
    1: 'Data Analyst',
    2: 'Data Engineer',
    3: 'Data Scientist',
    4: 'Python Programming'
}

# Menggantikan label dengan nama_kelas
hasil_sort['recommendation'] = hasil_sort['recommendation'].replace(label_to_kelas)

hasil_sort

```

	recommendation	probability
0	Data Scientist	0.38
1	Data Analyst	0.28
2	AI Engineer	0.18
3	Data Engineer	0.12
4	Python Programming	0.04

Gambar 4. 18 Merubah label

Gambar kode diatas menunjukkan bahwa label dirubah menjadi nama kelas.

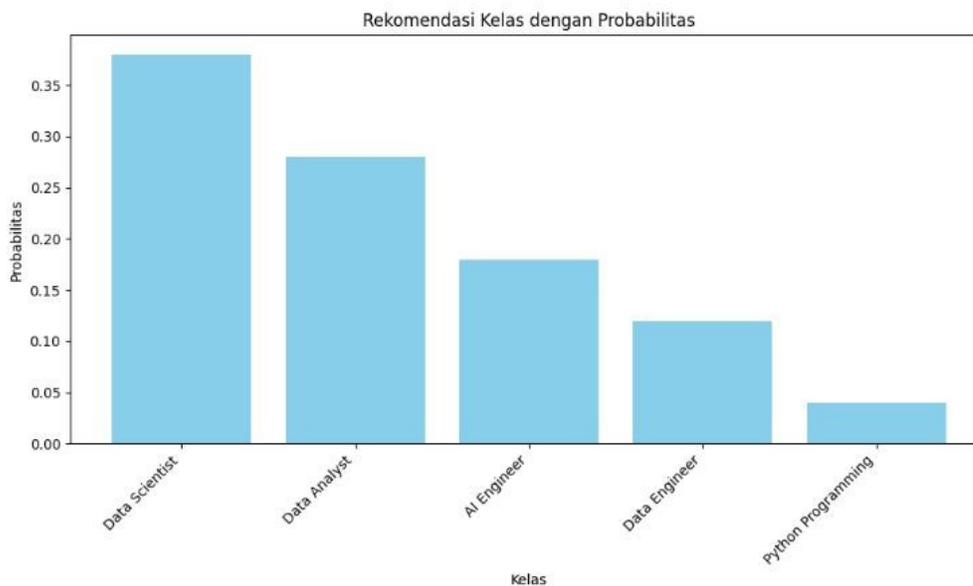
```
import matplotlib.pyplot as plt

# Membuat bar chart
plt.figure(figsize=(10, 6))
plt.bar(hasil_sort['recommendation'], hasil_sort['probability'], color='skyblue')
plt.xlabel('Kelas')
plt.ylabel('Probabilitas')
plt.title('Rekomendasi Kelas dengan Probabilitas')
plt.xticks(rotation=45, ha='right') # Rotasi label agar lebih mudah dibaca
plt.tight_layout()

# Menampilkan plot
plt.show()
```

Gambar 4. 19 membuat plot

Kode diatas digunakan untuk membuat plot dari skor yang didapat sehingga dengan gambar plot user dapat mudah membaca hasil rekomendasi.



Gambar 4. 20 Hasil plot

5. Hasil Prediksi Job

Dari data dan model yang digunakan dalam fitur rekomendasi job ini sama dengan rekomendasi kelas yang di atas namun ada sedikit perbedaan pada *form* inputan yang digunakan pada fitur ini, berikut hasil dari akurasi model yang digunakan pada fitur rekomendasi job lulusan *bootcamp* ini :

```
[ ] from sklearn import metrics

prediksi = model_dt.predict(X_test)

print('Akurasinya adalah: ', model_dt.score(X_test, y_test))

print(metrics.classification_report(y_test, prediksi))
```

Akurasinya adalah: 0.8571428571428571				
	precision	recall	f1-score	support
0	0.75	0.94	0.83	16
1	0.87	1.00	0.93	20
2	0.95	0.90	0.93	21
3	0.86	0.46	0.60	13
accuracy			0.86	70
macro avg	0.86	0.83	0.82	70
weighted avg	0.86	0.86	0.85	70

Gambar 4. 21 Akurasi Rekomendasi Job

Seperti yang ada di gambar di atas dalam rekomendasi job menggunakan model *Decission Tree* memperoleh angka akurasi 85%.

```
[ ] model_rf = RandomForestClassifier()

model_rf.fit(X_train, y_train)
# model_rf.fit(X_over, y_over)
```

▾ RandomForestClassifier
 RandomForestClassifier()

```
[ ] prediksi = model_rf.predict(X_test)

print('Akurasinya adalah: ', model_rf.score(X_test, y_test))

print(metrics.classification_report(y_test, prediksi))
```

Akurasinya adalah: 0.8714285714285714				
	precision	recall	f1-score	support
0	0.72	0.81	0.76	16
1	0.91	1.00	0.95	20
2	1.00	0.90	0.95	21
3	0.82	0.69	0.75	13
accuracy			0.87	70
macro avg	0.86	0.85	0.85	70
weighted avg	0.88	0.87	0.87	70

Gambar 4. 22 Akurasi Rekomendasi Job Random Forest

Pada gambar diatas menunjukkan bahwa menggunakan *Random Forest* mempunyai nilai akurasi yang lebih tinggi daripada menggunakan *Decission Tree*, pada model random *Random Forest* memperoleh nilai akurasi 87%.

```
[ ] nama = "Jhon Doe"
    summarize = """
    John Doe adalah lulusan bootcamp Data Science dengan latar belakang pendidikan dalam Ilmu Komputer. Selama tiga bulan praktik di proyek analisis data, John berhasil menyelesaikan proy
    John memiliki pengalaman dalam menganalisis dataset keuangan, yang sangat relevan dengan tugas seorang Data Analyst yang umumnya terlibat dalam analisis data untuk memberikan wawasan
    Keterampilan John dalam bahasa pemrograman seperti Python dan SQL juga menjadi keunggulan tambahan. Sebagai seorang Data Analyst, keterampilan ini sangat penting untuk melakukan anali
    Tingkat kesulitan proyek mini yang tinggi yang berhasil dihadapi oleh John menunjukkan bahwa ia dapat mengatasi tantangan dan menyelesaikan tugas yang kompleks dengan baik, sebuah ket
    """
    print(summarize)

    input_text = vec.transform([summarize]).toarray()
    label_rekomendasi = model.predict(input_text)
    print(label_rekomendasi)

Rekomendasi Job untuk lulusan Bootcamp Data Science di Intelligo ID
-----
John Doe adalah lulusan bootcamp Data Science dengan latar belakang pendidikan dalam Ilmu Komputer. Selama tiga bulan praktik di proyek analisis data, John berhasil menyelesaikan proy
John memiliki pengalaman dalam menganalisis dataset keuangan, yang sangat relevan dengan tugas seorang Data Analyst yang umumnya terlibat dalam analisis data untuk memberikan wawasan
Keterampilan John dalam bahasa pemrograman seperti Python dan SQL juga menjadi keunggulan tambahan. Sebagai seorang Data Analyst, keterampilan ini sangat penting untuk melakukan anali
Tingkat kesulitan proyek mini yang tinggi yang berhasil dihadapi oleh John menunjukkan bahwa ia dapat mengatasi tantangan dan menyelesaikan tugas yang kompleks dengan baik, sebuah ket
[1]
```

Gambar 4. 23 Sample input data rekomendasi job

Gambar diatas merupakan sample inputan dari rekomendasi job yang dilakukan oleh admin.

```
[ ] # Berdasarkan nilai label, berikan rekomendasi job
    if label_rekomendasi == 0:
        rekomendasi_job = "AI Engineer"
    elif label_rekomendasi == 1:
        rekomendasi_job = "Data Analyst"
    elif label_rekomendasi == 2:
        rekomendasi_job = "Data Engineer"
    elif label_rekomendasi == 3:
        rekomendasi_job = "Data Scientist"
    else:
        rekomendasi_job = "Tidak ada rekomendasi Job yang sesuai dengan Hasil kemampuan yang diberikan"

    # Tampilkan rekomendasi kelas
    print("Rekomendasi Job lulusan Bootcamp untuk 'Jhon Doe' adalah: ")
    print(f"Berdasarkan hasil pembelajaran anda, kami merekomendasikan Job sebagai : {rekomendasi_job}")

Rekomendasi Job lulusan Bootcamp untuk 'Jhon Doe' adalah:
Berdasarkan hasil pembelajaran anda, kami merekomendasikan Job sebagai : Data Analyst
```

Gambar 4. 24 Sample hasil rekomendasi job

Gambar diatas merupakan sample hasil dari rekomendasi job, hampir sama dengan rekomendasi kelas, seperti gambar diatas hasil dari data sample yang diinputkan memperoleh nilai atau label 1, jadi model merekomendasikan job ke label 1 yaitu *Data Analyst*.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil analisis pada laporan tugas akhir dapat disimpulkan bahwa:

1. Sistem rekomendasi menghasilkan rekomendasi kursus online dan lulusan bootcamp yang ada di intelligo id.

Dalam pengujian yang dilakukan pada *user* 1 direkomendasikan dengan persentase kecocokan sebesar sekitar 80% yang diperoleh dari hasil algoritma *Random Forest Regression*.

2. Model tersebut divisualisasikan menggunakan flask.

Penggunaan flask sebagai *framework* dapat melakukan visualisasi terhadap hasil rekomendasi teratas, hasil rekomendasi secara keseluruhan, dan juga hasil pengujian terhadap model yang digunakan.

Saran

Berdasarkan hasil analisis pada laporan Internship II disarankan untuk :

1. Dapat mengembangkan sistem rekomendasi ini dalam Akurasi persentase yang lebih tinggi.

Daftar Pustaka

1. Naufal Ryandhika., Gilland Kasyrafurrahman., Bagas Ramadhan (2021). Aplikasi Sistem Informasi Kursus Mengemudi Berbasis WEB (Studi Kasus : Kursus Setir Mobil Santa). ISSN 2339-1971
2. Radisya Eka Putri, Tine Agustin Wulandar. (2021). Perancangan Aplikasi Web E-Learning Berbasis LMS Menggunakan Moodle Di PT Global Infotech Solution, Volume 4 Nomor 2.
3. Ananda Hadi Elyas, PENGGUNAAN MODEL PEMBELAJARAN E-LEARNING DALAM MENINGKATKAN KUALITAS PEMBELAJARAN, ISSN : 1829 – 7463
4. Dewi Pusparani Sinambela , Husni Naparin, Muhammad Zulfadhila, Nurul Hidayah 2023. Implementasi Algoritma Decision Tree dan Random Forest dalam Prediksi Perdarahan Pascasalin, e-ISSN: 2714-9730
5. Sodiq Anshori 2019. Pemanfaatan Teknologi Informasi Dan Komunikasi Sebagai Media Pembelajaran. ISSN 2579-9878
6. Amriana, Ayu Puspitasari, Hawadis, 2020. E-LEARNING QUIPPER SCHOOL SEBAGAI PLATFORM ONLINE DALAM OPTIMALISASI PEMBELAJARAN SEJARAH, P-ISSN: 2355-2743; E-ISSN: 2549-3612

7. Nawaz, Using e-learning as a tool for 'education for all' in developing states, ISSN 2141-6559
8. Yefta Croshtian., Kelvin (2022). Rancang Bangun Aplikasi Kursus Online Berbasis Web Dengan Sistem Rekomendasi Metode Content-Based-Filtering. 2477-2062
9. Dionixiu, 2023. PENGKLASIFIKASIAN PEKERJAAN DAN REKOMENDASI KURSUS ONLINE BERDASARKAN KEMAMPUAN PENGGUNA DENGAN MENGGUNAKAN SISTEM MACHINE LEARNING.
10. Alifia Rahma Fitrianti, Asih Rohmani, Widjanarto, 2020. Sistem Rekomendasi Film Berbasis Website dengan Metode Prototype Menggunakan Algoritma K-Nearest Neighbors (KNN). Journal of Information System Vol. 5
11. Arwin Halim , Hernawati Gohzali, Dita Maria Panjaitan, Ilham Maulana. Sistem Rekomendasi Film menggunakan Bisecting K-Means dan Collaborative Filtering. ISBN: 978-602-60280-1-3
12. I Gede Suputra Widharma. PERANCANGAN SIMULASI SISTEM PENDAFTARAN KURSUS BERBASIS WEB DENGAN METODE SDLC. JURNAL MATRIX, VOL. 7, NO. 2
13. Felby Yugus Rinduanita. RANCANG BANGUN SISTEM INFORMASI KURSUS MENGEMUDI BERBASIS WEB PADA LEMBAGA KURSUS MENGEMUDI BERLIAN. Jurnal Manajemen Informatika. Volume 7
14. Dyah Rhetno Wardhani, Adhityo Kuncoro, 2019. Perancangan Sistem Pembelajaran E-learning Pada Kursus Online dengan Metode Web Based Learning Menggunakan Dokeos di SMK Nasional Depok. p-ISSN 1412-9434/e-ISSN 2549-7227
15. Fandi Yulian Pamuji, Viry Puspaning Ramadhan, 2021. Komparasi Algoritma Random Forest Dan Decision Tree Untuk Memprediksi Keberhasilan Immunotherapy. E-ISSN: 2580-8044
16. Octarian Prasetya Moerdyanto, I Kadek Dwi Nuryana, 2023. Prediksi Kelulusan Tepat Waktu Menggunakan Pendekatan Pohon Keputusan Algoritma Decision Tree. ISSN : 2686-2220
17. Agung Wibowo, Abdul Rohman. Prediksi Predikat Kelulusan Mahasiswa Menggunakan Naive Bayes dan Decision Tree pada Universitas XYZ. e-ISSN 2745-7265 p-ISSN 2088-5555
18. Silmi Ath Thahirah Al Azhima, Dwicky Darmawan, Nurul Fahmi Arief Hakim, 2022. HYBRID MACHINE LEARNING MODEL UNTUK MEMPREDIKSI PENYAKIT JANTUNG DENGAN METODE LOGISTIC REGRESSION DAN RANDOM FOREST. ISSN : 2477-0043
19. Jumjumi Asbullah, Samsudin, 2023. Prediksi Harga Cryptocurrency Binance Berdasarkan Informasi Blockchain dengan Menggunakan Algoritma Random Forest. ISSN 2548-8368
20. Fajar Mu'Alim, Rahmi Hidayati, 2022. Implementasi Metode Random Forest Untuk Penjurusan Siswa Di Madrasah Aliyah Negeri Sintang. Jurnal JUPITER, Vol. 14
21. Erlin, Yenny Desnelita, Nurliana Nasution, Laili Suryati, Fransiskus Zoromi, 2020. Impact of SMOTE on Random Forest Classifier Performance based on Imbalanced Data.

ISSN: 2476-9843

22. Nuruddin Wiranda, Harja Santana Purba, R Ati Sukmawati, 2020. Survei Penggunaan Tensorflow pada Machine Learning untuk Identifikasi Ikan Kawasan Lahan Basah. ISSN : 2460-7681
23. Rosita Diana, Heri Warni, Tata Sutabri, 2019. PENGGUNAAN TEKNOLOGI MACHINE LEARNING UNTUK PELAYANAN MONITORING KEGIATAN BELAJAR MENGAJAR PADA SMK BINA SRIWIJAYA PALEMBANG. ISSN : 2338-1477 – EISSN : 2541-6375
24. Amirah, Fitrah Karimah, 2023. Preliminary Study for Cyber Intrusion Detection Using Machine Learning Approach.
25. Yudin Wahyudin, Dhian Nur Rahayu, 2020. ANALISIS METODE PENGEMBANGAN SISTEM INFORMASI BERBASIS WEBSITE: A LITERATUR REVIEW. E-ISSN : 2621-1106
26. Maulana Nurul Hakim, Rinda Cahyana, 2015. PENGEMBANGAN SISTEM INFORMASI GEOGRAFIS UNTUK MEMUDAHKAN Pencarian Informasi Fasilitas Sosial dan Lokasinya. ISSN : 2302-7339
27. Steven, Wahyu Rifaldi, Ucu Nugraha, 2023. Strategi Pengamanan Front-end dalam Pengembangan Website. ISSN: 3031-3449
28. Prasetyo Bella Ramadhanu, Adhie Thyo Priandika, 2021. RANCANG BANGUN WEB SERVICE API APLIKASI SENTRALISASI PRODUK UMKM PADA UPTD PLUT KUMKM PROVINSI LAMPUNG. E-ISSN: 2746-3699
29. Brian Pratama Putra, Yerymia Alfa Susetyo, 2020. IMPLEMENTASI API MASTER STORE MENGGUNAKAN FLASK, REST DAN ORM DI PT XYZ. ISSN:2302-8149