PENERAPAN METODE *NAÏVE BAYES* UNTUK MEREKOMENDASIKAN PEKERJAAN YANG SESUAI TERHADAP *FRESH GRADUATE*

Ikrima Ningrumsari Mulyanan, Muhammad Yusril Helmi Setyawan, Woro Isti Rahayu
D4 Teknik Informatika, Universitas Logistik dan Bisnis International
Ikrimaningrum02@gmail.com

ABSTRAK

Di era teknologi yang kini berkembang, banyak lulusan baru yang menghadapi tantangan dalam mencari pekerjaan yang cocok dengan spesialisasi mereka. Sebagai solusi, sistem rekomendasi pekerjaan khusus untuk lulusan teknik informatika telah dirancang dengan memanfaatkan metode Naïve Bayes. Sistem ini dirancang untuk memberikan saran pekerjaan berdasarkan faktor seperti keahlian, riwayat pendidikan, dan pengalaman magang. Dari pengujian yang dilakukan, sistem ini menunjukkan tingkat keberhasilan prediksi sebesar 78%. Meski terdapat hambatan pada data pelatihan, sistem ini mampu mengklasifikasikan profil ke dalam subsektor IT dengan ketepatan sebesar 83%.

Kata kunci: Sistem rekomendasi, *Naïve bayes*, Klasifikasi

1. PENDAHULUAN

Data adalah kumpulan sebuah informasi atau deskripsi tentang sesuatu yang diperoleh dengan mengamati atau mencari sumber tertentu. Data yang diperoleh dapat berupa asumsi atau fakta karena belum diolah lebih lanjut. Setelah diolah dengan penelitian atau eksperimen, data dapat berbentuk lebih kompleks seperti database, informasi atau bahkan solusi untuk menyelesaikan masalah tertentu. Data juga diperlukan untuk diolah dan disajikan dalam dunia bisnis [1].

Dalam era sekarang ini muncul sebuah bidang profesi baru yaitu data science. Data science merupakan gabungan dari inferensi data, pengembangan algoritma, dan juga teknologi untuk memecahkan masalah analitik yang kompleks. Dalam data science juga terdapat analisis prediktif suatu data untuk difilter dan ditemukan data yang benar agar menghasilkan suatu data yang akurat sesuai dengan data yang sebenarnya. Data science atau ilmu data adalah seperangkat prinsip fundamental yang mendukung dan memandu ekstraksi berprinsip informasi dan pengetahuan dari data [3].

Pada sistem rekomendasi ini, penulis menggunakan content-based filtering atau content-based recommendation. Content Based Recommendation memanfaatkan informasi beberapa item/data untuk digunakan sebagai referensi yang terkait dengan informasi yang digunakan sebelumnya. Tujuan dari content based recommendation agar dapat memprediksi persamaan dari sejumlah informasi yang didapat dari data sebelumnya [4].

Saat ini, banyak lulusan baru atau Fresh Graduate perguruan tinggi yang menghadapi tantangan dalam mencari pekerjaan yang sesuai dengan keahlian dan ketertarikan mereka. Meskipun terdapat sejumlah besar lowongan, tidak seluruhnya cocok dengan ekspektasi dan kompetensi para Fresh Graduate [53]. Oleh karena itu, diperlukan sebuah mekanisme yang bisa menyarankan pekerjaan yang tepat berdasarkan informasi relevan. Fresh Graduate yang dimaksudkan pada pembuatan sistem rekomendasi ini adalah pada lulusan yang berhubungan dengan teknik informatika.

Naïve Bayes adalah teknik klasifikasi berdasarkan Teorema Bayes dengan asumsi independensi di antara para prediktor. Dalam istilah sederhana, Naïve Bayes classifier mengasumsikan bahwa kehadiran fitur tertentu di kelas tidak terkait dengan kehadiran fitur lainnya. Model Naïve Bayes mudah dibangun dan sangat berguna untuk set data yang sangat besar [5].

Metode Naïve Bayes dikenali sebagai pendekatan Machine Learning yang efisien untuk klasifikasi [5]. Dengan menggunakan pendekatan ini, harapannya lulusan baru dapat menerima rekomendasi pekerjaan yang sesuai dengan keahlian, pengalaman, dan minat mereka.

2. TINJAUAN PUSTAKA

2.1. Klasifikasi

Naive Bayes adalah sebuah metode statistik yang digunakan untuk klasifikasi dan prediksi di bidang ilmu komputer dan statistik. Metode ini didasarkan pada teorema Bayes, yang memungkinkan kita untuk menghitung probabilitas suatu peristiwa berdasarkan probabilitas peristiwa lain yang berkaitan dengan peristiwa tersebut. Naive Bayes memiliki aplikasi luas dalam pengolahan bahasa alami, pengenalan pola, klasifikasi dokumen, deteksi spam, dan lainnya [2].

Pada dasarnya, Naive Bayes menghitung probabilitas bahwa suatu objek termasuk dalam suatu kategori berdasarkan informasi yang diberikan oleh fitur-fitur yang terkait dengan objek tersebut. Metode ini cocok untuk masalah-masalah di mana data yang digunakan memiliki banyak fitur atau atribut, dan di mana perlu untuk melakukan klasifikasi dengan cepat [3].

2.2. Rekomendasi

Rekomendasi bertujuan untuk membantu orang atau organisasi membuat keputusan yang lebih baik dan informasi berdasarkan pada data atau pengetahuan

yang ada [5]. Rekomendasi dapat berupa saran, panduan, atau nasihat yang ditujukan untuk membantu pengambilan keputusan atau tindakan tertentu [6].

2.3. Pekerjaan

Pekerjaan adalah aktivitas atau tugas yang dilakukan oleh individu dalam pertukaran atas upah atau kompensasi tertentu. Aktivitas ini dapat berupa usaha fisik, mental, atau kreatif yang dilakukan untuk mencari nafkah, memberikan kontribusi pada masyarakat, atau memenuhi kebutuhan pribadi [2].

2.4. *Fresh Graduates*

Fresh graduates merupakan suatu istilah yang digunakan untuk merujuk kepada individu-individu yang baru saja menyelesaikan pendidikan formal mereka, seperti gelar sarjana atau gelar pendidikan tinggi lainnya, dan belum memiliki pengalaman kerja profesional yang signifikan setelah lulus [7].

3. METODOLOGI PENELITIAN

Metodologi yang digunakan dalam proses pelaksanaan penelitian ini. Penulis mengambil referensi dengan mengikuti tahapan model *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Dalam penelitian ini penulis membuat beberapa tahapan, sebagai berikut :

Berikut penjabaran tahapan alur metodologi penelitian :

1. *Business Understanding* atau pemahaman Bisnis, merujuk pada pemahaman mengenai tujuan dan sasaran bisnis, menilai situasi yang ada, serta mengkonversi tujuan bisnis tersebut menjadi sasaran dalam data mining.
2. *Data Understanding*, dalan tahap ini meliputi kegiatan pengumpulan data, analisis data, serta evaluasi atas mutu data yang akan digunakan selama pelaksanaan internship. Data yang menjadi rujukan dalam internship ini berasal dari berbagai platform media. Oleh karena itu, banyak literatur yang menjadi acuan.Sumber data yang digunakan dalam internship ini merupakan data-data yang terdapat dari berbagai media. Maka sumber literatur banyak didapatkan dari literatur, artikel atau jurnal, tulisan ilmiah, dan situs web pendukung.
3. *Data Preparation*, adalah suatu proses/langkah yang dilakukan untuk membuat data mentah menjadi data yang berkualitas (input yang baik untuk data mining tools). Tahap ini merupakan proses menghilangkan data mentah dan membersihkannya menjadi data yang berkualitas. preprocessing adalah langkah persiapan data yang diperlukan untuk klasifikasi rekomendasi pekerjaan. Preprocessing yang terdiri dari *Drop NA*, *Tokenization*, *Remove Punctuation*, *Remove Stopwords*, dan *Stemming*.
4. Pemodelan, Di tahapan ini, berbagai jenis teknik pemodelan diseleksi dan diaplikasikan pada dataset yang telah disusun guna memenuhi kebutuhan bisnis tertentu.
5. Evaluasi, Di fase ini, model yang telah dikembangkan diuji untuk menilai keakuratannya. Langkah ini bertujuan untuk menentukan seberapa baik model yang terpilih mencapai target bisnis, dan menentukan apakah diperlukan pendekatan lain atau model tambahan untuk evaluasi lebih lanjut.

Untuk menyelesaikan metodologi peneliltian sesuai dengan langkah-langkah di atas, dibutuhkan beberapa metode seperti di bawah ini.

3.1. *Drop* NA

Proses pembersihan data yang pertama adalah drop NA, dimana data NA kosong atau salah satu kolom dalam dataset adalah *null* atau *null*. Pada tahap ini data yang kosong akan dibuang atau dihapus [6].

3.2. *Tokenazitaion*

Teknik ini memisahkan dokumen menjadi serangkaian kata, membentuk vektor dari kata-kata tersebut, yang disebut sebagai "*bag-of-words*". Proses *tokenization* adalah dengan membagi teks yang semula berupa kalimat, nantinya akan dipisahkan dan menjadi token [7].

3.3. Remove Functuation

Pada tahap ini, tanda baca yang diberikan dalam data akan dihapus selain tanda baca, tag, angka, karakter khusus, emoticon dan lain-lain yang biasanya ada di data twitter akan dihancurkan [8].

3.4. *Remove Stopwords*

Ini adalah teknik yang menghilangkan kata-kata kosong yang sering digunakan dan tidak berguna untuk klasifikasi teks. Ini mengurangi ukuran tubuh pengumpulan data, analisis data, serta evaluasi atas mutu data yang akan digunakan selama pelaksanaan internship. Data yang menjadi rujukan dalam internship ini berasal dari berbagai platform media. Oleh karena itu, banyak literatur yang menjadi acuan.

3.5. *Stemming*

Bekerja dengan menghapus akhiran kata, menurut aturan tata bahasa, dan menurunkan kata dari sebuah kata [10].

3.6. *Naive Bayes*

Naive Bayes terkenal lebih unggul dibandingkan dengan metode klasifikasi lain yang lebih kompleks. Selanjutnya, kita akan melihat rumusan dari metode Naive Bayes. Teorema Bayes memberikan metode untuk menghitung probabilitas posterior P(X|H) berdasarkan P(X), P(H), dan P(H|X). Persamaan berikut akan menunjukkan perhitungannya.

$$P(H|X) = \frac{P(X|H)}{P(X)} . P(H) \quad (1)$$

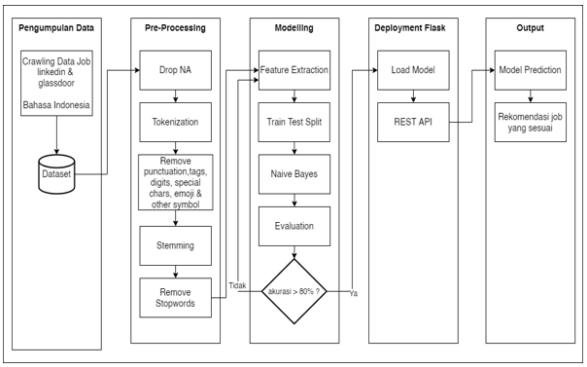
Keterangan :

X	=	Data dengan class yang belum diketahui
H	=	Hipotesis data merupakan suatu class spesifik
P(H|X)	=	Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)
P(H)	=	Probabilitas hipotesis H (prior probabilitas)
P(X|H)	=	Probabilitas X berdasarkan kondisi pada hipotesis H
P(X)	=	Probabilitas X

Naive Bayes menggunakan metode probabilitas posterior untuk memprediksi probabilitas kelas yang berbeda berdasarkan berbagai atribut [11]–[13]. Algoritma ini sebagian besar digunakan dalam klasifikasi teks dan dengan masalah memiliki beberapa kelas [14].

4. HASIL DAN DISKUSI

Di bagian ini akan menjawab semua hal yang ada pada bab sebelumnya tentang metodologi penelitian. Dari mulai hasil proses *scraping*, pembersihan data, pembuatan model klasifikasi rekomendasi pekerjaan menggunakan *naive bayes* sampai dengan evaluasi. Adapun untuk pembuatan model penulis menggunakan *flowchart* pada gambar.



Gambar 1. *Flowchart* Pembuatan Model

4.1. Pengumpulan Data

Data diambil dari beberapa *platform linkedin & glassdoor* menggunakan bahasa pemrograman Python, dengan library selenium. Data yang diambil untuk dijadikan suatu dataset terkait 10 bidang pekerjaan yang akan menjadi label setiap klasifikasi pekerjaan yang ada. Data berbentuk csv dan menggunakan 3 kolom yaitu deskripsi, *skill*, dan label. Berikut contoh hasil pengumpulan data pada tabel.

Tabel 1. Pengumpulan Data

Deskripsi	Skill	Label
Staff Administrasi merupakan pekerjaan yang banyak dicari oleh berbagai perusahaan, baik skala besar maupun kecil. Peran Admin memang penting karena mengatur segala urusan administrasi perusahaan. Meski demikian, sebagian orang menilai bahwa pekerjaan staff Administrasi itu mudah	Teliti, Keuangan, Dokumen, dan Admin	Staff Administrasi
Seorang ilmuwan data adalah individu yang mengendalikan teknologi pengolahan informasi. Mereka memiliki tanggung jawab yang meluas mulai dari analisis hingga memperoleh wawasan atau saran untuk pertumbuhan bisnis. Ilmuwan data memiliki tugas untuk menganalisis beragam jenis data dalam skala besar (big data) yang terkumpul di suatu perusahaan.	Python, R, SQL	Data Scientist
Programmer merupakan salah satu pekerjaan yang ditekuni oleh orang-orang yang membuat program berdasarkan lembar spesifikasi software engineering. Ketika membuat suatu program, ada diagram alur yang perlu dianalisis dan kemudian diubah menjadi bahasa komputer melalui kode-kode khusus agar bisa menjadi instruksi bagi komputer untuk melakukan fungsi tertentu. Beberapa contoh bahasa pemrograman meliputi CSS, SQL, HTML, JavaScript, PHP, XML, dan Python.	HTML, CSS, Javascript, SQL, PHP, Django, Java, Golang, UI, UX	*Programmer* (*Software Developer*)

Keseluruhan label yang akan menjadi rekomendasi pada penelitian ini adalah.

Gambar 2. Label Rekomendasi

4.2. Data *Preparation*

Setelah data dikumpulkan, melakukan langkah persiapan data yang mencakup beberapa langkah. Langkah awal adalah adalah membersihkan data dengan melakukan drop NA. Kemudian, melakukan proses *tokenization* yaitu dengan membagi dokumen menjadi kata/istilah (*bag-of-word*). Contoh data yang melewati *tokenization* pada tabel.

Tabel 1. Proses *Tokenization*

Data Sebelum Proses *Tokenization*	**Data Setelah Proses *Tokenization***
Data Scientist -adalah suatu profesi analisis data, yang akan memproses data menjadi sebuah kesimpulan.	[Data, Scientist, -, adalah, suatu, profesi, analisis, data, ,, yang, akan, memproses, data, menjadi, sebuah, kesimpulan]

Langkah berikutnya adalah "Penghapusan Tanda Baca". Di tahap ini, tanda baca dihilangkan, *tag*, angka, spesial karakter, *emoticon*, dan simbol lainnya. Contoh data yang melewati tahap ini pada tabel.

Tabel 2. Proses *Remove Punctuation*

Data Sebelum Proses *Remove Punctuation*	**Data Setelah Proses *Remove Punctuation***
Data Scientist -adalah suatu profesi analisis data, yang akan memproses data menjadi sebuah kesimpulan.	Data Scientist adalah suatu profesi analisis data yang akan memproses data menjadi sebuah kesimpulan.

Setelah itu, dilakukan *stemming* untuk membuang akhiran kata menurut beberapa tata bahasa aturan dan mendapatkan kara dasar dari kata ttersebut. Berikut adalah contoh data yg melewati tahap ini pada tabel.

Tabel 3. Proses *Stemming*

Data Sebelum Proses *Stemminng*	**Data Setelah Proses *Stemminng***
Data Scientist adalah suatu profesi analisis data yang akan memproses data menjadi sebuah kesimpulan	Data Scientist adalah suatu profesi analisis data yang akan proses data jadi buah simpul.

Tahap terakhir adalah proses *remove stopwords* untuk menghilangkan kata-kata yang sering digunakan yang tidak berarti dan tidak berguna seperti pada tabel.

Tabel 4. Proses *Remove Stopwords*

Data Sebelum Proses *Remove Stopwords*	**Data Setelah Proses *Remove Stopwords***
Data Scientist adalah suatu profesi analisis data yang akan proses data jadi buah simpul	Data Scientist profesi analisis data proses data buah simpul

4.3. *Modelling*

Pada proses modelling, dilakukan proses feature selection dengan menggunakan teknik TF-IDF. Penerapan TF-IDF dilakukan dengan memanfaatkan n = 1 (unigram). Berikut contoh penerapan teknik TF-IDF di bawah ini.

Keterangan :

Dokumen 1 = Python itu mudah

Dokumen 2 = PHP itu lebih mudah

Tabel 5. Kemunculan Term Pada Dokumen

Term	**D1**	**D2**	**Skor TF**
Python	1	0	1/5=0.2
Itu	1	1	2/5=0.4
Mudah	1	1	2/5=0.4
PHP	0	1	1/5=0.2
Lebih	0	1	1/5=0.2

Tabel tersebut menunjukkan perhitungan Frekuensi Istilah (TF) dari kata-kata di dua dokumen, D1 dan D2. TF dihitung dengan membagi frekuensi kata di suatu dokumen dengan total kata di dokumen tersebut. Contohnya, kata "python" muncul 1 kali di D1 dengan skor TF 0,2, sementara "itu" dan "mudah" muncul 2 kali dengan skor TF 0,4. Kata "PHP" dan "lebih" masing-masing memiliki skor TF 0,2. Tabel ini mengukur seberapa sering setiap kata muncul relatif terhadap total kata di kedua dokumen.

Tabel 6. Skor DF

Term	**D1**	**D2**	**Skor DF**
Python	1	0	1
Itu	1	1	2
Mudah	1	1	2
PHP	0	1	1
Lebih	0	1	1

Tabel tersebut menampilkan hitungan Document Frequency (DF) untuk kata-kata di dokumen D1 dan D2. DF mengukur berapa banyak dokumen yang memuat kata tertentu. Sebagai contoh, "python" ditemukan hanya di D1 (DF=1), sedangkan "itu" dan "mudah" ditemukan di kedua dokumen (DF=2). Sementara "PHP" dan "lebih" masing-masing ditemukan di satu dokumen (DF=1). Tabel ini menggambarkan seberapa sering sebuah kata muncul di antara kedua dokumen yang diteliti.

Tabel 7. Skor IDF

Term	**Skor DF**	**Skor IDF (1/DF)**
Python	1	1
Itu	2	0.5
Mudah	2	0.5
PHP	1	1
Lebih	1	1

Tabel tersebut menunjukkan Document Frequency (DF) dan Inverse Document Frequency (IDF) dari beberapa kata dalam metode pembobotan TF-IDF. Kolom "Skor DF" menampilkan frekuensi kemunculan kata dalam kumpulan dokumen.

Sementara kolom "Skor IDF (1/DF)" menggambarkan keunikan kata tersebut; kata yang jarang muncul memiliki skor IDF tinggi. Sebagai contoh, "python" dan "PHP" memiliki DF=1 dan IDF=1, sedangkan "itu" dan "mudah" memiliki DF=2 dan IDF=0,5. Tabel ini memberikan pandangan tentang relevansi dan keunikan kata-kata dalam sebuah kumpulan dokumen.

Tabel 8. Nilai TF-IDF

Term	Skor TF	Skor DF	Skor TF.IDF
Python	0.2	1	0.2
Itu	0.4	0.5	0.16
Mudah	0.4	0.5	0.16
PHP	0.2	1	0.2
Lebih	0.2	1	0.2

Tabel tersebut menunjukkan perhitungan pembobotan TF-IDF untuk beberapa kata. Kolom "Term" mencantumkan kata-kata yang dievaluasi. "Skor TF" mengukur frekuensi kemunculan kata dalam dokumen tertentu, sementara "Skor IDF" menilai seberapa jarang kata tersebut muncul di seluruh kumpulan dokumen, memberikan bobot lebih pada kata-kata yang lebih unik.

"Skor TF.IDF" adalah hasil kali dari TF dan IDF, menunjukkan signifikansi kata dalam konteks dokumen dan keseluruhan kumpulan dokumen. Sebagai contoh, walaupun "itu" dan "mudah" sering muncul dalam satu dokumen, skor TF.IDF-nya lebih rendah karena frekuensinya yang tinggi di seluruh dokumen. Kemudian dengan proses di atas, menghasilkan output pada gambar.

```
array([0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.14870912, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.26993568, 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.20102865, 0.        ,
       0.24328494, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
```

Gambar 3. TF-IDF pada *Python*

Setelah melewati *feature selection*, dilakukan pemodelan dengan proses *splitting* data dengan beberapa Eksperimen. Eksperimen pertama data testing 10%, Eksperimen kedua 20%, dan Eksperimen ketiga 30%. Proses pemodelan menggunakan model *Multinomial Naive Bayes Classifier*. Berikut di bawah ini skenarionya pada tabel.

Tabel 9. Skenario Eksperimen

No	Eksperimen	Test Size	Train Size
1	Eksperimen 1	10%	90%
2	Eksperimen 2	20%	80%
3	Eksperimen 3	30%	70%

Tabel tersebut menampilkan tiga uji coba menggunakan metode Naïve Bayes, khususnya varian "MultinomialNB". Tabel ini mencakup:

a) Kolom "No" untuk urutan uji coba.
b) "Percobaan" yang menunjukkan identifikasi setiap uji coba.
c) "Jenis Naïve Bayes" yang menunjukkan bahwa semua uji coba menggunakan varian "MultinomialNB".
d) "Test Size" dan "Train Size" yang menampilkan persentase data untuk pengujian dan pelatihan.

Dari tabel ini, kita bisa menyimpulkan bahwa setiap uji coba memiliki proporsi pembagian data yang berbeda, yang bisa mempengaruhi hasil model. Misalnya, "Percobaan 1" menggunakan 90% data untuk pelatihan dan 10% untuk pengujian. Hasil dari pengujian terhadap 3 pecobaan tersebut terdapat pada tabel dan gambar sampai gambar .

Tabel 10. Hasil Pengujian

No	Eksperimen	Test Size	Train Size	Akurasi
1	Eksperimen 1	10%	90%	76.74%
2	Eksperimen 2	20%	80%	85.88%
3	Eksperimen 3	30%	70%	82.81%

Tiga eksperimen dilakukan dengan pembagian data berbeda untuk pelatihan dan pengujian. "Eksperimen 1" mencapai akurasi 76.74%, "Eksperimen 2" 85.88%, dan "Eksperimen 3" 82.81%. Meskipun dengan proporsi pelatihan berbeda, "Eksperimen 2" yang menggunakan 80% data pelatihan dan 20% pengujian menghasilkan akurasi tertinggi.

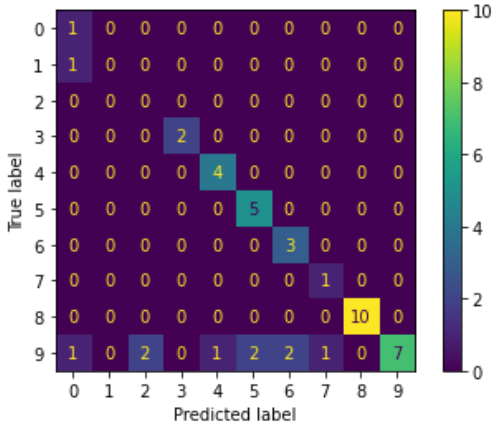
```
Akurasinya adalah:  0.7674418604651163
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         1
           1       0.00      0.00      0.00         1
           2       0.00      0.00      0.00         0
           3       1.00      1.00      1.00         2
           4       0.80      1.00      0.89         4
           5       0.71      1.00      0.83         5
           6       0.60      1.00      0.75         3
           7       0.50      1.00      0.67         1
           8       1.00      1.00      1.00        10
           9       1.00      0.44      0.61        16

    accuracy                           0.77        43
   macro avg       0.59      0.74      0.62        43
weighted avg       0.87      0.77      0.76        43
```

Gambar 4. *Report Classification* Eksperimen 1

Gambar berikut merupakan tampilan pada python pengujian atau experimen pertama. Pada Experimen ini menggunakan 10% dari data keseluruhan sebagai data pengujian, lalu mengindikasikan fraksi dari keseluruhan data yang dialokasikan untuk pelatihan.



Gambar 5. *Confusion Matrix* Eksperimen 1

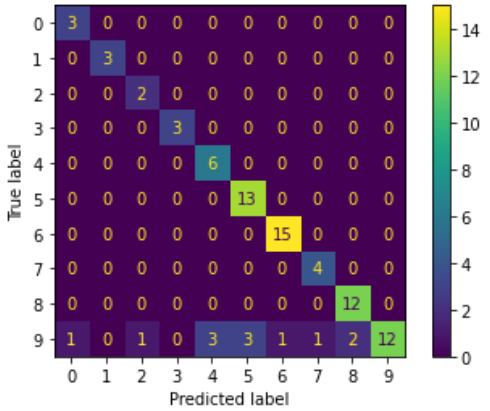
Pada experimen 1 ini 90% dari keseluruhan data dijadikan sebagai data pelatihan akurasi percobaan 1 mencapai akurasi sebesar 76,74%.

Akurasinya adalah: 0.8588235294117647

	precision	recall	f1-score	support
0	0.75	1.00	0.86	3
1	1.00	1.00	1.00	3
2	0.67	1.00	0.80	2
3	1.00	1.00	1.00	3
4	0.67	1.00	0.80	6
5	0.81	1.00	0.90	13
6	0.94	1.00	0.97	15
7	0.80	1.00	0.89	4
8	0.86	1.00	0.92	12
9	1.00	0.50	0.67	24
accuracy			0.86	85
macro avg	0.85	0.95	0.88	85
weighted avg	0.89	0.86	0.84	85

Gambar 6. *Report Classification* Eksperimen 2

Gambar berikut merupakan tampilan pada python pengujian atau experimen kedua. Pada Experimen 2 menggunakan 20% dari data keseluruhan sebagai data pengujian, lalu mengindikasikan fraksi dari keseluruhan data yang dialokasikan untuk pelatihan.



Gambar 7. *Confusion Matrix* Eksperimen 2

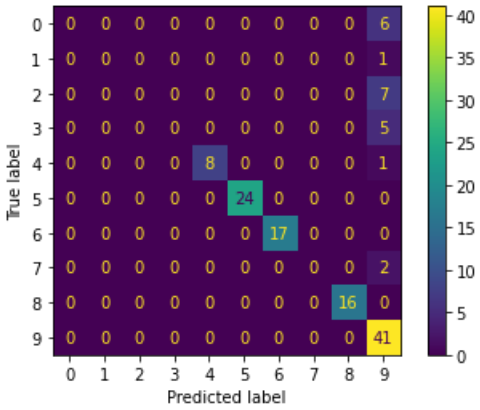
Pada experimen 2 ini 80% dari keseluruhan data dijadikan sebagai data pelatihan akurasi percobaan 2 mencapai akurasi sebesar 85,88%. Kita dapat menarik kesimpulan bahwa pada percobaan 2, meski hanya menggunakan 80% dari data untuk pelatihan, berhasil mencapai akurasi tertinggi, mengindikasikan efisiensi tertentu dalam model tersebut dibandingkan dengan eksperimen lainnya.

Akurasinya adalah: 0.828125

	precision	recall	f1-score	support
0	0.00	0.00	0.00	6
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	7
3	0.00	0.00	0.00	5
4	1.00	0.89	0.94	9
5	1.00	1.00	1.00	24
6	1.00	1.00	1.00	17
7	0.00	0.00	0.00	2
8	1.00	1.00	1.00	16
9	0.65	1.00	0.79	41
accuracy			0.83	128
macro avg	0.47	0.49	0.47	128
weighted avg	0.72	0.83	0.76	128

Gambar 8. *Report Classification* Eksperimen 3

Gambar berikut merupakan tampilan pada python pengujian atau experimen ketiga. Experimen 3 menggunakan 30% dari data keseluruhan sebagai data pengujian, lalu mengindikasikan fraksi dari keseluruhan data yang dialokasikan untuk pelatihan.



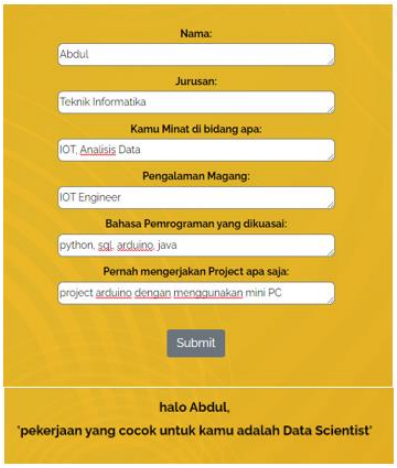
Gambar 9. *Confusion Matrix* Eksperimen 3

Pada eksperimen 3 ini 70% dari keseluruhan data dijadikan sebagai data pelatihan akurasi percobaan 3 mencapai akurasi sebesar 82,81%. Setelah mendapatkan akurasi dari pengujian model dengan 3 percobaan seperti diatas, maka akurasi paling tinggi diperoleh dengan menggunakan model MultinomialNB + Test Size 20% + Train Size 80% dengan total akurasi 85.88% serta jika dibulatkan menjadi 86%.

Gambar 10. Tampilan *Home*

Setelah mendapatkan akurasi dari pengujian model dengan 3 Eksperimen sebelumnya, maka akurasi paling tinggi diperoleh dengan menggunakan model yang di bulatkan menjadi 86%. Maka model yang paling tinggi akan di *deploy* pada sistem yang akan dibangun menggunakan *framework Flask*.

Perhatikan gambar diatas! Gambar tersebut merupakan hasil implementasi *framework* dan ada pada halaman *home*.

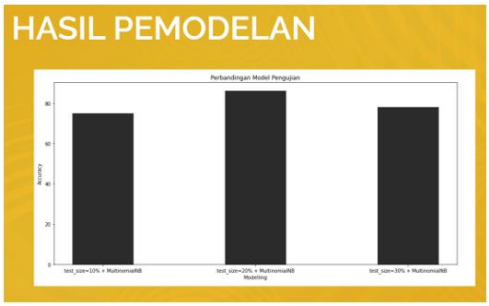


Gambar 11. Tampilan *Form* Input

User diminta mengisikan nama, jurusan, pengalaman magang, bahasa pemrograman yang dikuasai serta project apakah yang telah dikerjakan. Lalu setelah itu klik pada *button submit* dan data tersebut akan disimpan dan dibaca oleh sistem.

Gambar 12. Tampilan Hasil Prediksi

Gambar berikut merupakan *output* dari halaman home yang sebelumnya sudah diisi oleh user. Sistem akan membaca *input* yang diisikan oleh user dan akan menampilkan hasil prediksi



Gambar 13. Tampilan Hasil Pemodelan

Gambar berikut merupakan tampilan dari hasil permodelan yang telah dibuat. Model berikut mencakup model MultinomialNB + Test Size + Train Size.

5. KESIMPULAN DAN SARAN

Berdasarkan evaluasi dari penelitian ini, penulis menarik sejumlah kesimpulan sebagai berikut: Metode *recommendation system* dapat menjadi solusi dalam merekomendasikan pekerjaan untuk *fresh graduate*.

Penelitian ini bertujuan untuk merancang dan membangun sistem rekomendasi pekerjaan dengan metode Naïve Bayes untuk lulusan baru, yang melibatkan tahapan dari pemahaman data hingga evaluasi kinerja. Efektivitas metode ini diukur berdasarkan relevansi rekomendasi pekerjaan bagi Fresh Graduate. Hasil menunjukkan potensi dan tantangan dari metode ini, dengan sistem mencapai prediksi sebesar 78% dan kemampuan khusus dalam menempatkan profil ke subsektor IT dengan akurasi 83%.

Melakukan optimalisasi data dengan menggunakan dataset yang lebih besar untuk meningkatkan performa model. Menambahkan lebih banyak atribut atau fitur yang mungkin mempengaruhi rekomendasi pekerjaan. Mencoba metode klasifikasi lain selain Naïve Bayes, seperti Pohon Keputusan atau Jaringan Syaraf Tiruan. Menyatukan sistem ini dengan platform pencarian kerja untuk membantu lulusan baru menemukan pekerjaan yang sesuai. Rekomendasi ini diharapkan dapat mendukung penelitian dan inovasi berikutnya di bidang rekomendasi pekerjaan dengan pendekatan pembelajaran mesin.

DAFTAR PUSTAKA

[1] H. Abdurrohman, R. Dini, and A. P. Muharram, "Evaluasi Performa metode Deep Learning untuk Klasifikasi Citra Lesi Kulit The HAM10000," *Seminar Nasional Instrumentasi, Kontrol dan Otomasi (SNIKO)*, pp. 10–11, 2018.

[2] A. Apriani, H. Zakiyudin, and K. Marzuki, "Penerapan Algoritma Cosine Similarity dan

Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta," *Jurnal Bumigora Information Technology (BITe)*, vol. 3, no. 1, pp. 19–27, Jul. 2021, doi: 10.30812/bite.v3i1.1110.

[3] R. J. Brunner and E. J. Kim, "Teaching data science," in *Procedia Computer Science*, Elsevier B.V., 2016, pp. 1947–1956. doi: 10.1016/j.procs.2016.05.513.

[4] V. S. Dave, B. Zhang, M. Al Hasan, K. Aljadda, and M. Korayem, "A Combined Representation Learning Approach for Better Job and Skill Recommendation," *Proceedings of The 27th ACM International Conference on Information and Knowledge Management*, pp. 1997–2005, Oct. 2018.

[5] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Applied Sciences (Switzerland)*, vol. 10, no. 21, pp. 1–20, Nov. 2020, doi: 10.3390/app10217748.

[6] D. Ramdan, F. Valentino, N. Ikhsan, and A. Sani, "Penerapan Data Mining Terhadap Prediksi Mahasiswa Drop Out Pada Kampus STMIK Widuri Jakarta Dengan Metode Decision Tree C4.5," *JBPI - Jurnal Bidang Penelitian Informatika*, vol. 1, no. 2, pp. 141–150, 2023, [Online]. Available: https://ejournal.kreatifcemerlang.id/index.php/jbpi

[7] M. A. Rofiqi, Abd. C. Fauzan, A. P. Agustin, and A. A. Saputra, "Implementasi Term-Frequency Inverse Document Frequency (TF-IDF) Untuk Mencari Relevansi Dokumen Berdasarkan Query," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 1, no. 2, pp. 58–64, Dec. 2019, doi: 10.28926/ilkomnika.v1i2.18.

[8] I. Lailia Nur Rohmatun Nazila and D. Tri Utari, "Pemodelan Topik Keluhan Masyarakat Pasca Pandemi Menggunakan Metode Latent Dirichlet Allocation (LDA) (Studi kasus: Kabupaten Sleman Tahun 2022)," *Prosiding Pendidikan Matematika, Matematika, dan Statistika*, vol. 7, 2023.

[9] A. P. Wibawa, F. Miftahuddin, and Suyono, "K-Medoids Clustering Untuk Pembentukan Database Stopword Bahasa Jawa," *Ranah (Jurnal Kajian Bahasa)*, pp. 261–269, 2021, doi: 10.26499/rnh/v9i2.1490.

[10] A. Guterres, Gunawan, and J. Santoso, "Stemming Bahasa Tetun Menggunakan Pendekatan Rule Based," *Teknika*, vol. 8, no. 2, pp. 142–147, Oct. 2019, doi: 10.34148/teknika.v8i2.224.

[11] M. Ridho Handoko and Neneng, "Sistem Pakar Diagnosa Penyakit Selama Kehamilan Menggunakan Metode Naive Bayes Berbasis Web," *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol. 2, no. 1, pp. 50–58, 2021, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI

[12] A. Damuri, U. Riyanto, H. Rusdianto, and M. Aminudin, "Implementasi Data Mining dengan Algoritma Naïve Bayes Untuk Klasifikasi Kelayakan Penerima Bantuan Sembako," *JURIKOM (Jurnal Riset Komputer)*, vol. 8, no. 6, pp. 219–225, Dec. 2021, doi: 10.30865/jurikom.v8i6.3655.

[13] D. Alita, I. Sari, and A. Rahman Isnain, "Penerapan Naive Bayes Classifier Untuk Pendukung Keputusan Penerima Beasiswa," *JDMSI*, vol. 2, no. 1, pp. 17–23, 2021.

[14] M. Ibrahim, "Enriching Consumer Health Vocabulary Using Enhanced GloVe Word Embedding," *Corneel University*, 2020