

LAMPIRAN



Lampiran 1. Hasil Wawancara

Narasumber : Bapak Anom (Kepala Bagian Perencanaan dan Kerjasama)

Wawancara

1. Apa permasalahan/kendala yang terjadi di Biro Akademik, Kemahasiswaan, Perencanaan, dan Kerja Sama (AKPK) Undiksha?

Sebelumnya perlu diketahui bahwa biro AKPK terdiri dari 3 bagian yaitu Bagian Akademik dan Statistik, Kemahasiswaan dan Humas, serta Perencanaan dan Kerjasama. Khusus pada bagian Perencanaan dan Kerjasama saat ini tidak mengalami permasalahan dalam menentukan jumlah mahasiswa baru yang diluluskan, namun mereka merasa perlu adanya prediksi yang lebih akurat terhadap jumlah mahasiswa baru yang mendaftar kembali.

2. Dari permasalahan program yang ada, program manakah yang paling membutuhkan suatu sistem untuk menyelesaikan permasalahan tersebut?

Dalam permasalahan yang ada, dari pihak biro AKPK saat ini membutuhkan suatu sistem yang terintegrasi untuk menyelesaikan/membantu dalam proses penentuan jumlah mahasiswa baru yang diluluskan sehingga dapat lebih cepat dan efisien.

3. Kendala apa saja yang membuat permasalahan tersebut muncul?

Kendala yang dihadapi adalah proses pengerjaannya masih dilakukan secara manual berupa pengolahan file excel dalam pengerjaannya sehingga membutuhkan waktu yang cukup lama.

4. Bagaimana alur penentuan jumlah mahasiswa baru yang diluluskan?

Tim penerimaan mahasiswa baru bersurat ke fakultas untuk meminta data daya tampung => Dekan berkoordinasi dengan wakil dekan, kepala jurusan, dan ketua prodi untuk menentukan daya tampung di masing-masing program studi => Dekan memberikan data daya tampung dari masing-masing prodi ke tim penerimaan => Tim penerimaan mahasiswa baru memasukan data daya tampung ke sistem penerimaan.

5. Sistem seperti apakah yang dibutuhkan untuk menyelesaikan permasalahan tersebut?

Sistem yang mampu memberikan prediksi jumlah mahasiswa baru yang mendaftar kembali secara lebih cepat dan akurat yang berbasis web sehingga memudahkan dalam menentukan jumlah mahasiswa baru yang diluluskan.

6. Dalam permasalahan tersebut apa sajakah kriteria/atributnya?

Ya, ada beberapa kriteria yang digunakan dalam penentuan jumlah mahasiswa baru yang diluluskan yaitu:


1. Nama Sekolah
2. Nilai Rapor Setiap Semester
3. Pilihan 1
4. Pilihan 2
5. Jurusan yang diterima

7. Apakah Undiksha (Biro AKPK) setuju sebagai tempat penelitian dan bersedia memberikan data secara lengkap kepada peneliti?

Ya, bersedia

Kepala Biro AKPK

Singaraja,
Peneliti


Drs. I Wayan Ariasa
NIP. 195904041982031005


Komang Aditya Pratama
NIM. 1515051053

Lampiran 2. Blackbox Testing

**PENGUJIAN BLACK BOX PADA SISTEM CERDAS UNTUK PREDIKSI
DAFTAR KEMBALI PENERIMAAN MAHASISWA BARU DI
UNDIKSHA DENGAN METODE NAIVE BAYES**

Nama Responden : 1 Gsr Pt Anon Arimbawa

Petunjuk :

Berikan tanda centang (✓) pada salah satu kotak yang telah disediakan.
Valid = Hasil yang diharapkan sesuai dengan keluaran yang dihasilkan oleh sistem
Tidak Valid = Hasil yang diharapkan tidak sesuai dengan keluaran yang dihasilkan oleh sistem

No	Nama	Skenario	Hasil yang diharapkan	Hasil	
				Valid	Tidak Valid
1	Mengecek validasi login	Mengisi email atau password salah	Menampilkan pesan "Data Login Yang Anda Masukkan Salah"	✓	
		Mengisi email dan password benar	Masuk ke halaman beranda	✓	
2	Mengelola data test	Memilih salah satu jalur pada sidebar menu Data Test	Menampilkan halaman data test	✓	
		Memilih tombol	Menampilkan form	✓	

	Tambah	tambah data test		
	Tidak mengisi form data test, kemudian memilih tombol Submit	Menampilkan pesan "Please fill out this field"	✓	
	Mengisi form data test, kemudian memilih tombol Submit	Menampilkan pesan "Data Test Berhasil Ditambahkan!"	✓	
	Memilih tombol Edit, kemudian melakukan perubahan pada data test, dan memilih tombol Submit	Muncul pesan "Data Test Berhasil Diperbarui!"	✓	
	Memilih tombol Hapus pada salah satu data test	Menampilkan konfirmasi "Apakah anda yakin akan menghapus?". Jika memilih OK maka akan muncul pesan "Data Test Berhasil Dihapus!"	✓	
	Memilih tombol Import Data	Mendownload template data test	✓	

		kemudian memilih Download Template			
		Memilih tombol Import Data, kemudian memilih tombol Choose File, memilih file excel, dan memilih tombol Import	Muncul pesan "Data Test Berhasil Diimpor!"	✓	
		Memilih tombol Export Data	Mendownload data test	✓	
		Memilih tombol Predict Data	Menampilkan hasil prediksi pada kolom Daftar Kembali masing-masing data test	✓	
3	Mengelola data train	Memilih salah satu jalur pada sidebar menu Data Train	Menampilkan halaman data train	✓	
		Memilih tombol Tambah	Menampilkan form tambah data train	✓	
		Tidak mengisi form data train, kemudian memilih tombol	Menampilkan pesan "Please fill out this field"	✓	

		Submit			
		Mengisi form data train, kemudian memilih tombol Submit	Menampilkan pesan "Data Train Berhasil Ditambahkan!"	✓	
		Memilih tombol Edit, kemudian melakukan perubahan pada data train, dan memilih tombol Submit	Muncul pesan "Data Train Berhasil Diperbarui!"	✓	
		Memilih tombol Hapus pada salah satu data train	Menampilkan konfirmasi "Apakah anda yakin akan menghapus?". Jika memilih OK maka akan muncul pesan "Data Train Berhasil Dihapus!"	✓	
		Memilih tombol Import Data, kemudian memilih Download Template	Mendownload template data train	✓	
		Memilih tombol Import Data	Muncul pesan "Data Train Berhasil"	✓	

		kemudian memilih tombol Choose File, memilih file excel, dan memilih tombol Import	Diimport!"		
		Memilih tombol Export Data	Mendownload data train	✓	
		Memilih tombol Performance Data	Menampilkan hasil perhitungan performance data train	✓	
4	Mengelola data rekapitulasi	Memilih salah satu jalur pada sidebar menu Rekapitulasi	Menampilkan halaman data rekapitulasi berdasarkan tingkatan Prodi		
		Memilih salah satu prodi	Menampilkan mahasiswa pada data test beserta daftar kembalnya	✓	
		Memilih Fakultas pada dropdown tingkatan	Menampilkan halaman data rekapitulasi berdasarkan	✓	

			tingkatan Fakultas		
		Memilih Keseluruhan pada dropdown tingkatan	Menampilkan halaman data rekapitulasi berdasarkan tingkatan Keseluruhan	✓	
		Memilih tombol Export Data	Mendownload data rekapitulasi berdasarkan tingkatan yang dipilih	✓	
5	Mengelola chart rekapitulasi	Memilih salah satu jalur pada sidebar menu Chart Rekapitulasi	Menampilkan halaman chart rekapitulasi	✓	
		Memilih navigation tombol, kemudian memilih Download	Mendownload chart rekapitulasi	✓	
6	Mengelola data prodi	Memilih sidebar menu Data Prodi	Menampilkan halaman data prodi	✓	
		Memilih tombol Tambah	Menampilkan form tambah data prodi	✓	
		Tidak mengisi form	Menampilkan pesan		

		data prodi, kemudian memilih tombol Submit	"Please fill out this field"	✓	
		Mengisi form data prodi, kemudian memilih tombol Submit	Menampilkan pesan "Data Prodi Berhasil Ditambahkan!"	✓	
		Memilih tombol Edit, kemudian melakukan perubahan pada data prodi, dan memilih tombol Submit	Muncul pesan "Data Prodi Berhasil Diperbarui!"	✓	
		Memilih tombol Hapus pada salah satu data prodi	Menampilkan konfirmasi "Apakah anda yakin akan menghapus?". Jika memilih OK maka akan muncul pesan "Data Test Berhasil Dihapus!"	✓	
		Memilih tombol Import Prodi, kemudian memilih Download Template	Mendownload template data prodi	✓	

		Memilih tombol Import Prodi, kemudian memilih tombol Choose File, memilih file excel, dan memilih tombol Import	Muncul pesan "Data Prodi Berhasil Diimport!"	✓	
		Memilih tombol Export Prodi	Mendownload data prodi	✓	
7	Mengelola data sekolah	Memilih sidebar menu Data Sekolah	Menampilkan halaman data sekolah	✓	
		Memilih tombol Tambah	Menampilkan form tambah data sekolah	✓	
		Tidak mengisi form data sekolah, kemudian memilih tombol Submit	Menampilkan pesan "Please fill out this field"	✓	
		Mengisi form data sekolah, kemudian memilih tombol Submit	Menampilkan pesan "Data Sekolah Berhasil Ditambahkan!"	✓	
		Memilih tombol Edit, kemudian melakukan	Muncul pesan "Data Sekolah Berhasil	✓	

		perubahan pada data sekolah, dan memilih tombol Submit	Diperbarui!"	✓	
		Memilih tombol Hapus pada salah satu data sekolah	Menampilkan konfirmasi "Apakah anda yakin akan menghapus?". Jika memilih OK maka akan muncul pesan "Data Test Berhasil Dihapus!"	✓	
		Memilih tombol Import Sekolah, kemudian memilih Download Template	Mendownload template data sekolah	✓	
		Memilih tombol Import Sekolah, kemudian memilih tombol Choose File, memilih file excel, dan memilih tombol Import	Muncul pesan "Data Sekolah Berhasil Diimport!"	✓	
		Memilih tombol Export Sekolah	Mendownload data sekolah	✓	

8	Mengelola data UKT	Memilih sidebar menu Data UKT	Menampilkan halaman data UKT	✓	
		Memilih tombol Tambah	Menampilkan form tambah data UKT	✓	
		Tidak mengisi form data UKT, kemudian memilih tombol Submit	Menampilkan pesan "Please fill out this field"	✓	
		Mengisi form data UKT, kemudian memilih tombol Submit	Menampilkan pesan "Data UKT Berhasil Ditambahkan!"	✓	
		Memilih tombol Edit, kemudian melakukan perubahan pada data UKT, dan memilih tombol Submit	Muncul pesan "Data UKT Berhasil Diperbarui!"	✓	
		Memilih tombol Hapus pada salah satu data UKT	Menampilkan konfirmasi "Apakah anda yakin akan menghapus?". Jika memilih OK maka akan muncul pesan "Data Test Berhasil"	✓	

			Dihapus!"		
		Memilih tombol Import UKT, kemudian memilih Download Template	Mendownload template data UKT	✓	
		Memilih tombol Import UKT, kemudian memilih tombol Choose File, memilih file excel, dan memilih tombol Import	Muncul pesan "Data UKT Berhasil Diimport!"	✓	
		Memilih tombol Export UKT	Mendownload data UKT	✓	
9	Mengelola data profile	Memilih username pada sidebar	Menampilkan halaman data profile	✓	
		Memilih tombol Edit Profile	Menampilkan form edit profile	✓	
		Tidak mengisi form data profile, kemudian memilih tombol Submit	Menampilkan pesan "Please fill out this field"	✓	
		Mengisi form data profile, kemudian	Menampilkan pesan "Data Profile	✓	

		memilih tombol Submit	Berhasil Diperbarui!"		
		Memilih tombol Change Password, kemudian mengisi password, dan memilih tombol Submit	Muncul pesan "Password Berhasil Diperbarui!"	✓	



Lampiran 3. Whitebox Testing

1. Pengujian menampilkan hasil prediksi daftar kembali data test dengan metode Naive Bayes menggunakan bahasa pemrograman python.

Pre Condition	Nilai daftar_kembali pada data test belum diketahui
Post Condition/ Hasil yang diharapkan	<p>Sistem melakukan proses prediksi untuk mendapatkan nilai daftar_kembali data test pada setiap prodi berdasarkan nilai atribut-atribut (nama_sekolah, ukt, pilihan_1, pilihan_2, pilihan_3, prodi) pada data test yang dibandingkan dengan nilai atribut-atribut pada data train berserta nilai daftar_kembali data train, Untuk menghitung nilai daftar_kembali data test diawali dengan menghitung probabilitas/kemungkinan YA dan TIDAK daftar_kembali dari atribut-atribut data train yang diperlukan:</p> $p_i = \frac{m_i}{n}$

Data Test SBMPTN

Import Data

Export Data

Predict Data

Prodi

Semua Prodi

Tambah

Show

10

entries

Search:

No	Nama	Nama Sekolah	UKT	Tahun	Pilihan 1	Pilihan 2	Pilihan 3	Prodi	Daftar Kembali	Aksi
1	Putu Z	SMKN 3 SINGARAJA	UKT Kelompok 3	2019	Pendidikan Teknik Informatika	Bimbingan dan Konseling		Pilihan 1		<a>Edit <a>Hapus
2	Kadek Z	SMKN 1 SINGARAJA	UKT Kelompok 4	2019	Pendidikan Bahasa Inggris	Pendidikan Teknik Informatika	Pendidikan Ekonomi	Pilihan 2		<a>Edit <a>Hapus
3	Komang Z	SMK PGRI 1 Denpasar	UKT Kelompok 3	2019	Akuntansi (S1)		Pendidikan Teknik Informatika	Pilihan 3		<a>Edit <a>Hapus
4	Ketut Z	SMAN 4 SINGARAJA	UKT Kelompok 7	2019	Manajemen	Pendidikan Teknik Informatika	Ilmu Hukum	Pilihan 2		<a>Edit <a>Hapus

Showing 1 to 4 of 4 entries

Previous
1
Next

Berikut merupakan proses untuk mendapatkan nilai probabilitas daftar kembali Putu Z:

$$P(YA) = 20/30 = 0.67$$

$$P(\text{nama_sekolah} = \text{SMKN 3 Singaraja} \mid YA) = 4/20 = 0.2$$

$$P(\text{ukt} = \text{UKT Kelompok 3} \mid YA) = 2/20 = 0.1$$

$$P(\text{pilihan_1} = \text{Pendidikan Teknik Informatika} \mid YA) = 9/20 = 0.45$$

$$P(\text{pilihan_2} = \text{Bimbingan dan Konseling} \mid YA) = 1/20 = 0.05$$

$$P(\text{pilihan_3} = - \mid YA) = 9/20 = 0.45$$

$$P(\text{prodi} = \text{Pilihan 1} \mid YA) = 9/20 = 0.45$$

$$P(TIDAK) = 10/30 = 0.33$$

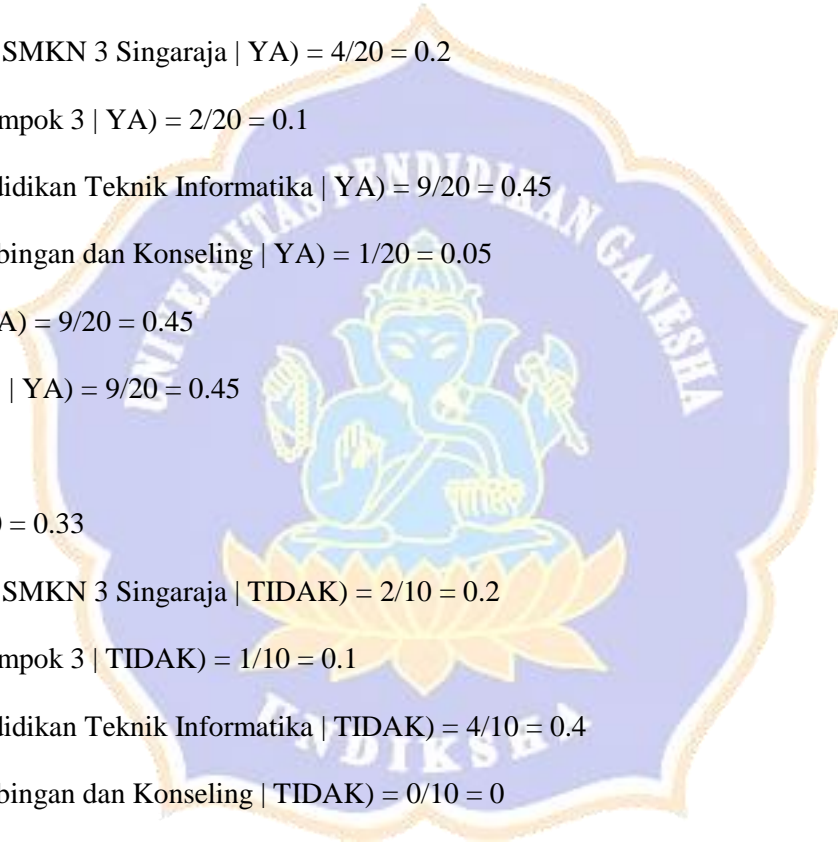
$$P(\text{nama_sekolah} = \text{SMKN 3 Singaraja} \mid TIDAK) = 2/10 = 0.2$$

$$P(\text{ukt} = \text{UKT Kelompok 3} \mid TIDAK) = 1/10 = 0.1$$

$$P(\text{pilihan_1} = \text{Pendidikan Teknik Informatika} \mid TIDAK) = 4/10 = 0.4$$

$$P(\text{pilihan_2} = \text{Bimbingan dan Konseling} \mid TIDAK) = 0/10 = 0$$

$$P(\text{pilihan_3} = - \mid TIDAK) = 2/10 = 0.2$$



$$P(\text{prodi} = \text{Pilihan 1} \mid \text{TIDAK}) = 4/10 = 0.4$$

Jika ada probabilitas yang bernilai nol maka dilakukan perhitungan tersebut dengan menerapkan Laplace Correction pada semua data dalam satu atribut yang sama:

$$\rho_i = \frac{m_i + 1}{n + k}$$

Berikut merupakan proses untuk mendapatkan nilai probabilitas nol:

$$P(\text{pilihan}_2 = \text{Bimbingan dan Konseling} \mid \text{YA}) = \frac{1+1}{20+10} = 0.07$$

$$P(\text{pilihan}_2 = \text{Bimbingan dan Konseling} \mid \text{TIDAK}) = \frac{0+1}{10+6} = 0.06$$

Kemudian perbandingan probabilitas YA dan TIDAK dari daftar_kembali dapat dicari dengan menggunakan persamaan dari teorema Bayes (Bustami, 2010):

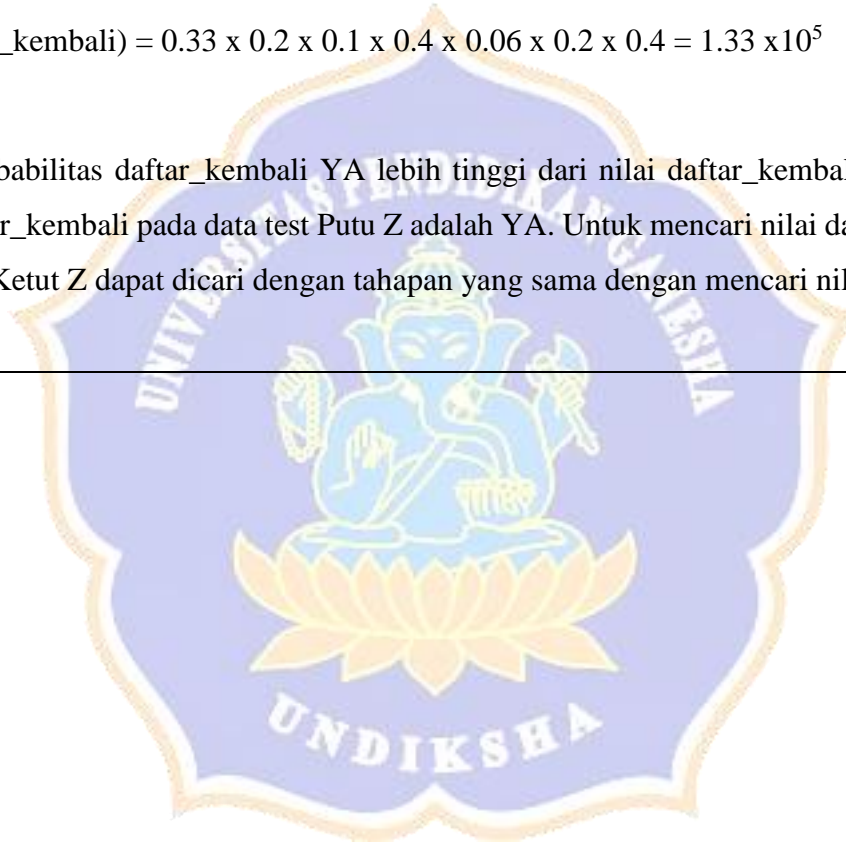
$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Berikut merupakan proses untuk mendapatkan nilai daftar_kembali:

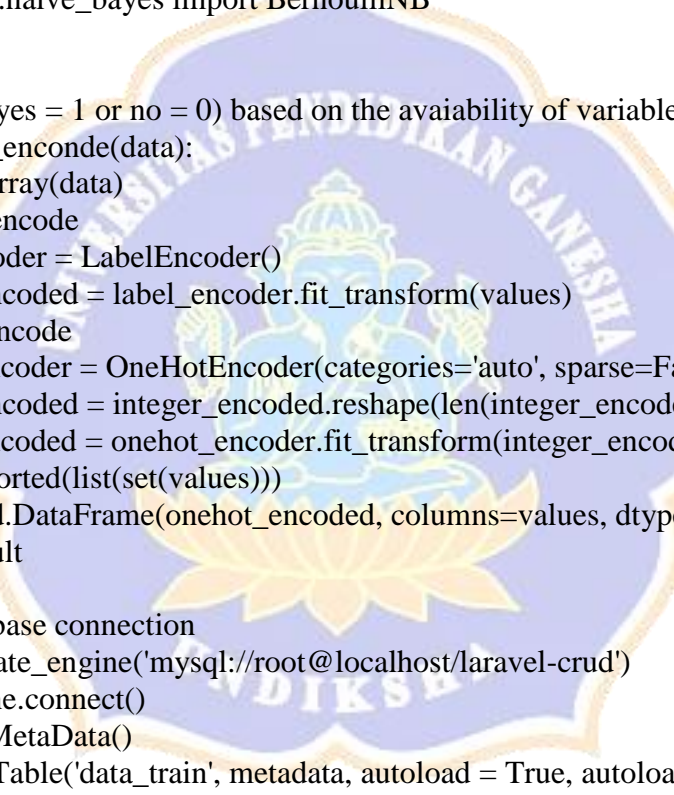
$$P(YA|daftar_kembali) = 0.67 \times 0.2 \times 0.1 \times 0.45 \times 0.07 \times 0.45 \times 0.45 = 8.1 \times 10^5$$

$$P(TIDAK|daftar_kembali) = 0.33 \times 0.2 \times 0.1 \times 0.4 \times 0.06 \times 0.2 \times 0.4 = 1.33 \times 10^5$$

Karena nilai probabilitas daftar_kembali YA lebih tinggi dari nilai daftar_kembali TIDAK, maka didapat kesimpulan bahwa nilai daftar_kembali pada data test Putu Z adalah YA. Untuk mencari nilai daftar_kembali pada data test Kadek Z, Komang Z, dan Ketut Z dapat dicari dengan tahapan yang sama dengan mencari nilai daftar_kembali pada data test Putu Z.



Hasil	<div><div>Data Test SBMPTN</div><div><div>✓ Data Test Berhasil Diprediksi!</div><div>Loading Time: 4.54 detik</div></div><div><div><div>Import Data</div><div>Export Data</div><div>Predict Data</div></div><div>Prodi</div><div>Semua Prodi</div><div>Tambah</div></div><div><div>Show</div><div>10</div><div>entries</div><div>Search:</div></div><table><tr><th>No</th><th>Nama</th><th>Nama Sekolah</th><th>UKT</th><th>Tahun</th><th>Pilihan 1</th><th>Pilihan 2</th><th>Pilihan 3</th><th>Prodi</th><th>Daftar Kembali</th><th>Aksi</th></tr><tr><td>1</td><td>Putu Z</td><td>SMKN 3 SINGARAJA</td><td>UKT Kelompok 3</td><td>2019</td><td>Pendidikan Teknik Informatika</td><td>Bimbingan dan Konseling</td><td></td><td>Pilihan 1</td><td>YA</td><td><div>Edit</div><div>Hapus</div></td></tr><tr><td>2</td><td>Kadek Z</td><td>SMKN 1 SINGARAJA</td><td>UKT Kelompok 4</td><td>2019</td><td>Pendidikan Bahasa Inggris</td><td>Pendidikan Teknik Informatika</td><td>Pendidikan Ekonomi</td><td>Pilihan 2</td><td>YA</td><td><div>Edit</div><div>Hapus</div></td></tr><tr><td>3</td><td>Komang Z</td><td>SMK PGRI 1 Denpasar</td><td>UKT Kelompok 3</td><td>2019</td><td>Akuntansi (S1)</td><td></td><td>Pendidikan Teknik Informatika</td><td>Pilihan 3</td><td>TIDAK</td><td><div>Edit</div><div>Hapus</div></td></tr><tr><td>4</td><td>Ketut Z</td><td>SMAN 4 SINGARAJA</td><td>UKT Kelompok 7</td><td>2019</td><td>Manajemen</td><td>Pendidikan Teknik Informatika</td><td>Ilmu Hukum</td><td>Pilihan 2</td><td>TIDAK</td><td><div>Edit</div><div>Hapus</div></td></tr></table></div>	No	Nama	Nama Sekolah	UKT	Tahun	Pilihan 1	Pilihan 2	Pilihan 3	Prodi	Daftar Kembali	Aksi	1	Putu Z	SMKN 3 SINGARAJA	UKT Kelompok 3	2019	Pendidikan Teknik Informatika	Bimbingan dan Konseling		Pilihan 1	YA	<div>Edit</div> <div>Hapus</div>	2	Kadek Z	SMKN 1 SINGARAJA	UKT Kelompok 4	2019	Pendidikan Bahasa Inggris	Pendidikan Teknik Informatika	Pendidikan Ekonomi	Pilihan 2	YA	<div>Edit</div> <div>Hapus</div>	3	Komang Z	SMK PGRI 1 Denpasar	UKT Kelompok 3	2019	Akuntansi (S1)		Pendidikan Teknik Informatika	Pilihan 3	TIDAK	<div>Edit</div> <div>Hapus</div>	4	Ketut Z	SMAN 4 SINGARAJA	UKT Kelompok 7	2019	Manajemen	Pendidikan Teknik Informatika	Ilmu Hukum	Pilihan 2	TIDAK	<div>Edit</div> <div>Hapus</div>
No	Nama	Nama Sekolah	UKT	Tahun	Pilihan 1	Pilihan 2	Pilihan 3	Prodi	Daftar Kembali	Aksi																																														
1	Putu Z	SMKN 3 SINGARAJA	UKT Kelompok 3	2019	Pendidikan Teknik Informatika	Bimbingan dan Konseling		Pilihan 1	YA	<div>Edit</div> <div>Hapus</div>																																														
2	Kadek Z	SMKN 1 SINGARAJA	UKT Kelompok 4	2019	Pendidikan Bahasa Inggris	Pendidikan Teknik Informatika	Pendidikan Ekonomi	Pilihan 2	YA	<div>Edit</div> <div>Hapus</div>																																														
3	Komang Z	SMK PGRI 1 Denpasar	UKT Kelompok 3	2019	Akuntansi (S1)		Pendidikan Teknik Informatika	Pilihan 3	TIDAK	<div>Edit</div> <div>Hapus</div>																																														
4	Ketut Z	SMAN 4 SINGARAJA	UKT Kelompok 7	2019	Manajemen	Pendidikan Teknik Informatika	Ilmu Hukum	Pilihan 2	TIDAK	<div>Edit</div> <div>Hapus</div>																																														
Status	Valid																																																							
Source Code	<div><div>1. from sqlalchemy import create_engine, MetaData, Table</div><div>2. import pandas as pd</div></div>																																																							



```

3. import numpy as np
4. from numpy import array
5. from sklearn.preprocessing import LabelEncoder
6. from sklearn.preprocessing import OneHotEncoder
7. from sklearn.naive_bayes import BernoulliNB
8. import sys
9.
10. # get value (yes = 1 or no = 0) based on the availability of variables in data
11. def one_hot_encode(data):
12.     values = array(data)
13.     # integer encode
14.     label_encoder = LabelEncoder()
15.     integer_encoded = label_encoder.fit_transform(values)
16.     # binary encode
17.     onehot_encoder = OneHotEncoder(categories='auto', sparse=False)
18.     integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
19.     onehot_encoded = onehot_encoder.fit_transform(integer_encoded)
20.     values = sorted(list(set(values)))
21.     result = pd.DataFrame(onehot_encoded, columns=values, dtype=int)
22.     return result
23.
24. # make database connection
25. engine = create_engine('mysql://root@localhost/laravel-crud')
26. conn = engine.connect()
27. metadata = MetaData()
28. tabletrain = Table('data_train', metadata, autoload = True, autoload_with = engine)
29. tabletest = Table('data_test', metadata, autoload = True, autoload_with = engine)
30.

```

```

31. # select data from a table in database
32. jalur = sys.argv[1]
33. prodi = sys.argv[2]
34. if (prodi == 'semua'):
35.     recordtrain = conn.execute("SELECT
        id_train,nama,npsn,id_ukt,jalur,tahun,kode_pilihan_1,kode_pilihan_2,kode_pilihan_3,prodi_pilihan,daftar_
        kembali,(CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3
        THEN kode_pilihan_3 END) AS prodi FROM data_train WHERE jalur LIKE
        '%%{ }%%'.format(jalur)).fetchall()
36.     recordtest = conn.execute("SELECT
        id_test,nama,npsn,id_ukt,jalur,tahun,kode_pilihan_1,kode_pilihan_2,kode_pilihan_3,prodi_pilihan,daftar_k
        embali,(CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3
        THEN kode_pilihan_3 END) AS prodi FROM data_test WHERE jalur LIKE
        '%%{ }%%'.format(jalur)).fetchall()
37. else:
38.     recordtrain = conn.execute("SELECT
        id_train,nama,npsn,id_ukt,jalur,tahun,kode_pilihan_1,kode_pilihan_2,kode_pilihan_3,prodi_pilihan,daftar_
        kembali,(CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3
        THEN kode_pilihan_3 END) AS prodi FROM data_train WHERE jalur LIKE '%%{ }%%' AND (CASE
        prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3 THEN
        kode_pilihan_3 END) = { }".format(jalur,prodi)).fetchall()
39.     recordtest = conn.execute("SELECT
        id_test,nama,npsn,id_ukt,jalur,tahun,kode_pilihan_1,kode_pilihan_2,kode_pilihan_3,prodi_pilihan,daftar_k
        embali,(CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3
        THEN kode_pilihan_3 END) AS prodi FROM data_test WHERE jalur LIKE '%%{ }%%' AND (CASE
        prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3 THEN
        kode_pilihan_3 END) = { }".format(jalur,prodi)).fetchall()
40.

```

```

41. # import data
42. dfcolumns =
    ['id','nama','npsn','id_ukt','jalur','tahun','kode_pilihan_1','kode_pilihan_2','kode_pilihan_3','prodi_pilihan','dafa',
    'tar_kembali','prodi']
43. dfa = pd.DataFrame(recordtrain, columns=dfcolumns)
44. dfb = pd.DataFrame(recordtest, columns=dfcolumns)
45.
46. # data preparation
47. dfa = dfa.replace(np.nan, '-', regex=True)
48. dfb = dfb.replace(np.nan, '-', regex=True)
49. dfa = dfa.replace(" ", '-', regex=True)
50. dfb = dfb.replace(" ", '-', regex=True)
51.
52. # convert categorical variable
53. dfa0 = one_hot_encode(dfa['npsn'])
54. dfa1 = one_hot_encode(dfa['id_ukt']).add_suffix(' (ukt)')
55. dfa2 = one_hot_encode(dfa['kode_pilihan_1']).add_suffix(' (plh1)')
56. dfa3 = one_hot_encode(dfa['kode_pilihan_2']).add_suffix(' (plh2)')
57. dfa4 = one_hot_encode(dfa['kode_pilihan_3']).add_suffix(' (plh3)')
58. dfa5 = one_hot_encode(dfa['prodi_pilihan']).add_suffix(' (prodi)')
59.
60. dfb0 = one_hot_encode(dfb['npsn'])
61. dfb1 = one_hot_encode(dfb['id_ukt']).add_suffix(' (ukt)')
62. dfb2 = one_hot_encode(dfb['kode_pilihan_1']).add_suffix(' (plh1)')
63. dfb3 = one_hot_encode(dfb['kode_pilihan_2']).add_suffix(' (plh2)')
64. dfb4 = one_hot_encode(dfb['kode_pilihan_3']).add_suffix(' (plh3)')
65. dfb5 = one_hot_encode(dfb['prodi_pilihan']).add_suffix(' (prodi)')
66.

```



```

67. dfa0 = dfa0.reindex(columns=dfb0.columns)
68. dfa1 = dfa1.reindex(columns=dfb1.columns)
69. dfa2 = dfa2.reindex(columns=dfb2.columns)
70. dfa3 = dfa3.reindex(columns=dfb3.columns)
71. dfa4 = dfa4.reindex(columns=dfb4.columns)
72. dfa5 = dfa5.reindex(columns=dfb5.columns)
73.
74. dfa0 = dfa0.replace(np.nan, 0, regex=True)
75. dfa1 = dfa1.replace(np.nan, 0, regex=True)
76. dfa2 = dfa2.replace(np.nan, 0, regex=True)
77. dfa3 = dfa3.replace(np.nan, 0, regex=True)
78. dfa4 = dfa4.replace(np.nan, 0, regex=True)
79. dfa5 = dfa5.replace(np.nan, 0, regex=True)
80.
81. # combine variables
82. dfx = pd.concat([dfa[['id']],dfa0,dfa1,dfa2,dfa3,dfa4,dfa5,dfa[['daftar_kembali']],dfa[['prodi']]],axis=1,
    sort=False)
83. dfy = pd.concat([dfb[['id']],dfb0,dfb1,dfb2,dfb3,dfb4,dfb5,dfb[['daftar_kembali']],dfb[['prodi']]],axis=1,
    sort=False)
84. var = pd.concat([pd.Series(dfa0.columns),pd.Series(dfa1.columns),
85.                 pd.Series(dfa2.columns),pd.Series(dfa3.columns),
86.                 pd.Series(dfa4.columns),pd.Series(dfa5.columns)],
87.                 ignore_index=True)
88.
89. # instantiate classifier, groupby, and dictionary
90. bnb = BernoulliNB()
91. gy = dfy.groupby('prodi')
92. gx = dfx.groupby('prodi')

```



```

93. dx = dict(list(gx))
94.
95. # group data by prodi
96. for (no_x, dfx_prodi), (no_y, dfy_prodi) in zip(gx,gy):
97.     # matching kode_prodi of dfx_prodi with dfy_prodi
98.     if no_y in dx:
99.         dfx_prodi = dx[no_y]
100.
101.     # drop dfx_prodi's columns to match dfy_prodi's columns
102.     dfy_prodi.drop([col for col, val in dfy_prodi.sum().iteritems() if val == 0], axis=1, inplace=True)
103.     dfx_prodi = dfx_prodi.reindex(columns=dfy_prodi.columns)
104.     var =
        pd.concat([pd.Series(dfx_prodi.columns.drop(['id','daftar_kembali','prodi'])),ignore_index=True)
105.
106.     # implement naive bayes fit data
107.     bnb.fit(dfx_prodi[var], dfx_prodi['daftar_kembali'])
108.
109.     # implement naive bayes predict data
110.     y_pred = bnb.predict(dfy_prodi[var])
111.
112.     # combine daftar_kembali with no_pendaftaran
113.     y_pred0 = pd.DataFrame(data=y_pred, columns=['daftar_kembali'])
114.     y_pred0.index = dfy_prodi['id']
115.     values = pd.concat([y_pred0],axis=1, sort=False)
116.
117.     # update the database with new values
118.     for key,row in values.iterrows():
119.         if (prodi == 'semua'):

```


	<pre> 120. update = conn.execute("UPDATE data_test SET daftar_kembali = '{}' WHERE id_test = {}").format(row['daftar_kembali'],key)) 121. else: 122. update = conn.execute("UPDATE data_test SET daftar_kembali = '{}' WHERE id_test = {} AND (CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3 THEN kode_pilihan_3 END) = {}".format(row['daftar_kembali'],key,prodi)) 123. 124. print(update) </pre>
--	--

2. Pengujian menampilkan hasil perhitungan performance data train dengan metode Naive Bayes, K-Fold Cross Validation, dan Confusion Matrix menggunakan bahasa pemrograman python.

Pre Condition	Data train yang akan dicari nilai akurasi, presisi, recall, specificity, dan F1 score																														
Post Condition/ Hasil yang diharapkan	<p>Sistem melakukan proses perhitungan performance data train (nilai akurasi, presisi, recall, specificity, F1 score) berdasarkan nilai atribut-atribut (nama_sekolah, ukt, pilihan_1, pilihan_2, pilihan_3, prodi) pada data train. Untuk menghitung performance data train diawali dengan membagi data train dengan K-Fold Cross Validation (k=10) sehingga dilakukan pengulangan sebanyak 10 folds dengan mendapatkan 9/10 data sebagai data train baru dan 1/10 data sebagai data test yang acak pada setiap pengulangan.</p> <p>Berikut adalah tabel Confusion Matrix dari hasil K-Fold Cross Validation:</p> <table><tr><th>Fold</th><th>TP</th><th>FN</th><th>FP</th><th>TN</th><th>Akurasi</th><th>Presisi</th><th>Recall</th><th>Specificity</th><th>F1 Score</th></tr><tr><td>1</td><td>5</td><td>2</td><td>0</td><td>3</td><td>0.80</td><td>1.00</td><td>0.72</td><td>1.00</td><td>0.84</td></tr><tr><td>2</td><td>5</td><td>2</td><td>2</td><td>1</td><td>0.60</td><td>0.72</td><td>0.72</td><td>0.34</td><td>0.72</td></tr></table>	Fold	TP	FN	FP	TN	Akurasi	Presisi	Recall	Specificity	F1 Score	1	5	2	0	3	0.80	1.00	0.72	1.00	0.84	2	5	2	2	1	0.60	0.72	0.72	0.34	0.72
Fold	TP	FN	FP	TN	Akurasi	Presisi	Recall	Specificity	F1 Score																						
1	5	2	0	3	0.80	1.00	0.72	1.00	0.84																						
2	5	2	2	1	0.60	0.72	0.72	0.34	0.72																						

3	5	2	1	2	0.70	0.84	0.72	0.67	0.78
4	6	0	3	1	0.70	0.67	1.00	0.25	0.81
5	5	2	2	1	0.60	0.72	0.72	0.34	0.72
6	7	0	2	1	0.80	0.78	1.00	0.34	0.88
7	7	2	1	0	0.70	0.88	0.78	0.00	0.83
8	7	1	0	2	0.90	1.00	0.88	1.00	0.94
9	6	0	3	1	0.70	0.67	1.00	0.25	0.81
10	6	0	2	2	0.80	0.75	1.00	0.50	0.86

Pengukuran performance dari tabel di atas adalah:

$$\text{Akurasi} = \frac{(0.8 + 0.6 + 0.7 + 0.7 + 0.6 + 0.8 + 0.7 + 0.9 + 0.7 + 0.8)}{10} = 0.73 = 73\%$$

$$\text{Presisi} = \frac{(1 + 0.72 + 0.84 + 0.67 + 0.72 + 0.78 + 0.88 + 1 + 0.67 + 0.75)}{10} = 0.8 = 80\%$$


$$\text{Recall} = \frac{(0.72 + 0.72 + 0.72 + 1 + 0.72 + 1 + 0.78 + 0.88 + 1 + 1)}{10} = 0.85 = 85\%$$

$$\text{Specificity} = \frac{(1 + 0.34 + 0.67 + 0.25 + 0.34 + 0.34 + 0 + 1 + 0.25 + 0.5)}{10} = 0.47 = 47\%$$

$$\text{F1 Score} = \frac{(0.84 + 0.72 + 0.78 + 0.81 + 0.72 + 0.88 + 0.83 + 0.94 + 0.81 + 0.86)}{10} = 0.82 = 82\%$$

Maka data mahasiswa baru yang diprediksi menggunakan metode Naive Bayes memiliki akurasi sebesar 73%, presisi sebesar 80%, recall sebesar 85%, specificity sebesar 47%, dan F1 score sebesar 82%.

Hasil	<p>Data Train SBMPTN</p> <div> <p>✓ Data Train "Pendidikan Teknik Informatika" Berhasil Dieksekusi!</p> <p>Total: 10, Miss: 4, Akurasi: 60.00%, Presisi: 71.43%, Recall: 71.43%, Specificity: 33.33%, F1 Score: 71.43%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 2, Akurasi: 80.00%, Presisi: 100.00%, Recall: 66.67%, Specificity: 100.00%, F1 Score: 80.00%, Plh 1: 100.00%, Plh 2: 66.67%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 2, Akurasi: 80.00%, Presisi: 85.71%, Recall: 85.71%, Specificity: 66.67%, F1 Score: 85.71%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 2, Akurasi: 80.00%, Presisi: 71.43%, Recall: 100.00%, Specificity: 60.00%, F1 Score: 83.33%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 40.00%</p> <p>Total: 10, Miss: 4, Akurasi: 60.00%, Presisi: 80.00%, Recall: 57.14%, Specificity: 66.67%, F1 Score: 66.67%, Plh 1: 100.00%, Plh 2: 66.67%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 3, Akurasi: 70.00%, Presisi: 77.78%, Recall: 87.50%, Specificity: 0.00%, F1 Score: 82.35%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 5, Akurasi: 50.00%, Presisi: 71.43%, Recall: 62.50%, Specificity: 0.00%, F1 Score: 66.67%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 2, Akurasi: 80.00%, Presisi: 85.71%, Recall: 85.71%, Specificity: 66.67%, F1 Score: 85.71%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 2, Akurasi: 80.00%, Presisi: 71.43%, Recall: 100.00%, Specificity: 60.00%, F1 Score: 83.33%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Total: 10, Miss: 4, Akurasi: 60.00%, Presisi: 66.67%, Recall: 66.67%, Specificity: 50.00%, F1 Score: 66.67%, Plh 1: 100.00%, Plh 2: 100.00%, Plh 3: 0.00%</p> <p>Average:</p> <p>Total: 10.0, Miss: 3.0, Akurasi: 70.00%, Presisi: 78.16%, Recall: 78.33%, Specificity: 50.33%, F1 Score: 77.19%, Plh 1: 100.00%, Plh 2: 93.33%, Plh 3: 4.00%</p> </div>
Status	Valid
Source Code	<pre> 1. from sqlalchemy import create_engine, MetaData, Table 2. import pandas as pd 3. import numpy as np 4. from numpy import array 5. from sklearn.preprocessing import LabelEncoder 6. from sklearn.preprocessing import OneHotEncoder 7. from sklearn.model_selection import KFold 8. from sklearn.naive_bayes import BernoulliNB 9. import sys 10. 11. # get value (yes = 1 or no = 0) based on the availability of variables in data </pre>



```

12. def one_hot_encode(data):
13.     values = array(data)
14.     # integer encode
15.     label_encoder = LabelEncoder()
16.     integer_encoded = label_encoder.fit_transform(values)
17.     # binary encode
18.     onehot_encoder = OneHotEncoder(categories='auto', sparse=False)
19.     integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
20.     onehot_encoded = onehot_encoder.fit_transform(integer_encoded)
21.     values = sorted(list(set(values)))
22.     result = pd.DataFrame(onehot_encoded, columns=values, dtype=int)
23.     return result
24.
25. # make database connection
26. engine = create_engine('mysql://root@localhost/laravel-crud')
27. conn = engine.connect()
28. metadata = MetaData()
29. tabletrain = Table('data_train', metadata, autoload = True, autoload_with = engine)
30. tableprodi = Table('data_prodi', metadata, autoload = True, autoload_with = engine)
31.
32. # select data from a table in database
33. jalur = sys.argv[1]
34. prodi = sys.argv[2]
35. if (prodi == 'semua'):
36.     recordtrain = conn.execute("SELECT
        id_train,nama,npsn,id_ukt,jalur,tahun,kode_pilihan_1,kode_pilihan_2,kode_pilihan_3,prodi_pilihan,daftar_
        kembali,(CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3

```

```

THEN kode_pilihan_3 END) AS prodi FROM data_train WHERE jalur LIKE
'%%{ }%%'.format(jalur)).fetchall()
37. else:
38.     recordtrain = conn.execute("SELECT
id_train,nama,npsn,id_ukt,jalur,tahun,kode_pilihan_1,kode_pilihan_2,kode_pilihan_3,prodi_pilihan,daftar_
kembali,(CASE prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3
THEN kode_pilihan_3 END) AS prodi FROM data_train WHERE jalur LIKE '%%{ }%%' AND (CASE
prodi_pilihan WHEN 1 THEN kode_pilihan_1 WHEN 2 THEN kode_pilihan_2 WHEN 3 THEN
kode_pilihan_3 END) = {}".format(jalur,prodi)).fetchall()
39.
40. # import data
41. dfcolumns =
['id','nama','npsn','id_ukt','jalur','tahun','kode_pilihan_1','kode_pilihan_2','kode_pilihan_3','prodi_pilihan','daf
tar_kembali','prodi']
42. dfa = pd.DataFrame(recordtrain, columns=dfcolumns)
43.
44. # data preparation
45. dfa = dfa.replace(np.nan, '-', regex=True)
46. dfa = dfa.replace("", '-', regex=True)
47.
48. # convert categorical variable
49. dfa0 = one_hot_encode(dfa['npsn'])
50. dfa1 = one_hot_encode(dfa['id_ukt']).add_suffix(' (ukt)')
51. dfa2 = one_hot_encode(dfa['kode_pilihan_1']).add_suffix(' (plh1)')
52. dfa3 = one_hot_encode(dfa['kode_pilihan_2']).add_suffix(' (plh2)')
53. dfa4 = one_hot_encode(dfa['kode_pilihan_3']).add_suffix(' (plh3)')
54. dfa5 = one_hot_encode(dfa['prodi_pilihan']).add_suffix(' (prodi)')
55.

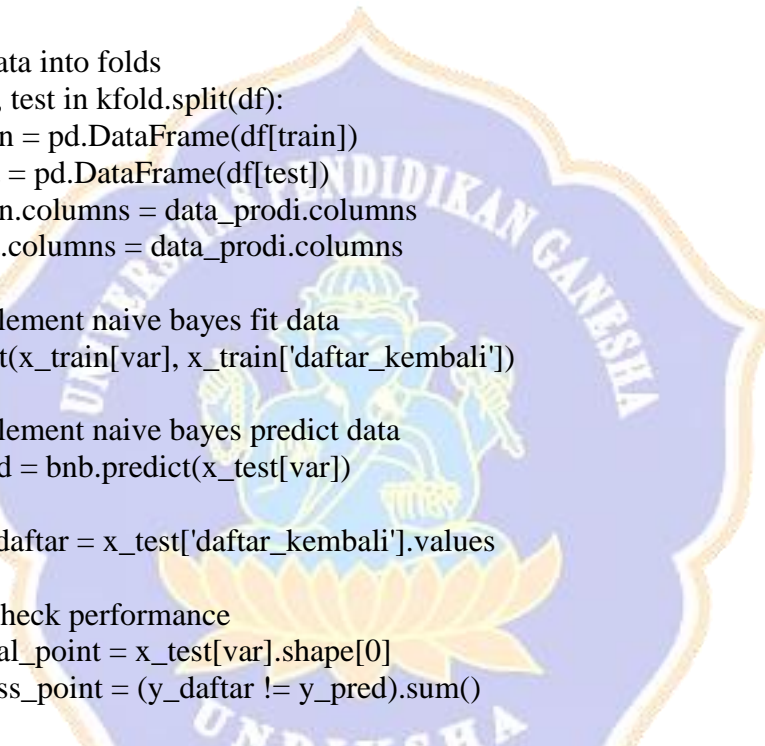
```



```

56. # combine variables
57. dfx = pd.concat([dfa[['id']],dfa0,dfa1,dfa2,dfa3,dfa4,dfa5,dfa[['daftar_kembali']],dfa[['prodi']]],axis=1,
    sort=False)
58. var = pd.concat([pd.Series(dfa0.columns),pd.Series(dfa1.columns),
59.                 pd.Series(dfa2.columns),pd.Series(dfa3.columns),
60.                 pd.Series(dfa4.columns),pd.Series(dfa5.columns)],
61.                 ignore_index=True)
62.
63. i = 0
64. avgperfs = np.zeros(shape=(10,2))
65.
66. # instantiate classifier, groupby, and dictionary
67. bnb = BernoulliNB()
68. gx_train = dfx.groupby('prodi')
69. dx_train = dict(list(gx_train))
70.
71. # group data by prodi
72. for no_prodi, data_prodi in gx_train:
73.     # drop dfx_prodi's unnecessary columns
74.     data_prodi.drop([col for col, val in data_prodi.sum().iteritems() if val == 0], axis=1, inplace=True)
75.     var = pd.concat([pd.Series(data_prodi.columns.drop(['id','daftar_kembali','prodi']))],ignore_index=True)
76.
77.     df = data_prodi.values
78.     perfs = np.zeros(shape=(10,2))
79.     k = len(data_prodi)
80.
81.     if (k >= 10):
82.         k = 10

```

```

83.     kfold = KFold(k, False, None)
84.     else:
85.         continue
86.
87.     # split data into folds
88.     for train, test in kfold.split(df):
89.         x_train = pd.DataFrame(df[train])
90.         x_test = pd.DataFrame(df[test])
91.         x_train.columns = data_prodi.columns
92.         x_test.columns = data_prodi.columns
93.
94.         # implement naive bayes fit data
95.         bnb.fit(x_train[var], x_train['daftar_kembali'])
96.
97.         # implement naive bayes predict data
98.         y_pred = bnb.predict(x_test[var])
99.
100.        y_daftar = x_test['daftar_kembali'].values
101.
102.        # check performance
103.        total_point = x_test[var].shape[0]
104.        miss_point = (y_daftar != y_pred).sum()
105.
106.        TP = ((y_daftar == "YA") & (y_pred == "YA")).sum()
107.        FN = ((y_daftar == "YA") & (y_pred == "TIDAK")).sum()
108.        FP = ((y_daftar == "TIDAK") & (y_pred == "YA")).sum()
109.        TN = ((y_daftar == "TIDAK") & (y_pred == "TIDAK")).sum()
110.

```

```

111.     perfs[0][0] = perfs[0][0] + total_point
112.     total_txt = "{}".format(total_point)
113.     perfs[1][0] = perfs[1][0] + miss_point
114.     miss_txt = "{}".format(miss_point)
115.
116.     if (total_point != 0):
117.         akurasi = (TP + TN) / total_point
118.         akurasi_txt = "{:0.2f}%".format(akurasi * 100)
119.         perfs[2][0] = perfs[2][0] + (akurasi * 100)
120.         perfs[2][1] += 1
121.     else:
122.         akurasi_txt = "NaN"
123.
124.     if ((TP + FP) != 0):
125.         presisi = TP / (TP + FP)
126.         presisi_txt = "{:0.2f}%".format(presisi * 100)
127.         perfs[3][0] = perfs[3][0] + (presisi * 100)
128.         perfs[3][1] += 1
129.     else:
130.         presisi_txt = "NaN"
131.
132.     if ((TP + FN) != 0):
133.         recall = TP / (TP + FN)
134.         recall_txt = "{:0.2f}%".format(recall * 100)
135.         perfs[4][0] = perfs[4][0] + (recall * 100)
136.         perfs[4][1] += 1
137.     else:
138.         recall_txt = "NaN"

```

```

139.
140.     if ((TN + FP) != 0):
141.         specificity = TN / (TN + FP)
142.         specificity_txt = "{:0.2f}%".format(specificity * 100)
143.         perfs[5][0] = perfs[5][0] + (specificity * 100)
144.         perfs[5][1] += 1
145.     else:
146.         specificity_txt = "NaN"
147.
148.     if ((presisi_txt != "NaN") & (recall_txt != "NaN")):
149.         if ((presisi + recall) != 0):
150.             f1_score = 2 * ((presisi * recall) / (presisi + recall))
151.             f1_score_txt = "{:0.2f}%".format(f1_score * 100)
152.             perfs[6][0] = perfs[6][0] + (f1_score * 100)
153.             perfs[6][1] += 1
154.         else:
155.             f1_score_txt = "NaN"
156.     else:
157.         f1_score_txt = "NaN"
158.
159.     if hasattr(x_test, '1 (prodi)'):
160.         if ((x_test['1 (prodi)'].values == 1).sum() > 0):
161.             prob_plh_1 = (((x_test['1 (prodi)'].values == 1) & (y_pred == "YA")).sum() / (x_test['1
(prodi)'].values == 1).sum())
162.             plh_1_txt = "{:0.2f}%".format(prob_plh_1 * 100)
163.             perfs[7][0] = perfs[7][0] + (prob_plh_1 * 100)
164.             perfs[7][1] += 1
165.     else:

```

```

166.         plh_1_txt = "NaN"
167.     else:
168.         plh_1_txt = "NaN"
169.
170.     if hasattr(x_test, '2 (prodi)'):
171.         if ((x_test['2 (prodi)'].values == 1).sum() > 0):
172.             prob_plh_2 = (((x_test['2 (prodi)'].values == 1) & (y_pred == "YA")).sum() / (x_test['2
(prodi)'].values == 1).sum())
173.             plh_2_txt = "{:0.2f}%".format(prob_plh_2 * 100)
174.             perfs[8][0] = perfs[8][0] + (prob_plh_2 * 100)
175.             perfs[8][1] += 1
176.         else:
177.             plh_2_txt = "NaN"
178.     else:
179.         plh_2_txt = "NaN"
180.
181.     if hasattr(x_test, '3 (prodi)'):
182.         if ((x_test['3 (prodi)'].values == 1).sum() > 0):
183.             prob_plh_3 = (((x_test['3 (prodi)'].values == 1) & (y_pred == "YA")).sum() / (x_test['3
(prodi)'].values == 1).sum())
184.             plh_3_txt = "{:0.2f}%".format(prob_plh_3 * 100)
185.             perfs[9][0] = perfs[9][0] + (prob_plh_3 * 100)
186.             perfs[9][1] += 1
187.         else:
188.             plh_3_txt = "NaN"
189.     else:
190.         plh_3_txt = "NaN"
191.

```

```

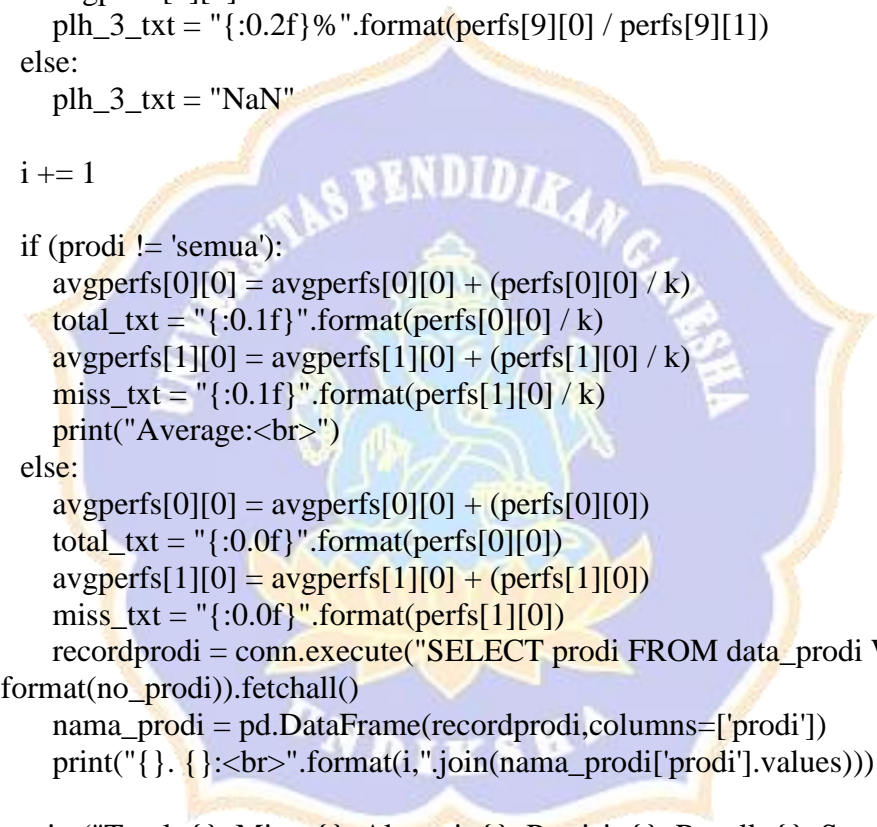
192.         if (prodi != 'semua'):
193.             print("Total: {}, Miss: {}, Akurasi: {}, Presisi: {}, Recall: {}, Specificity: {}, F1 Score: {}, Plh
              1: {}, Plh 2: {}, Plh 3:
              {}<br>".format(total_txt,miss_txt,akurasi_txt,presisi_txt,recall_txt,specificity_txt,f1_score_txt,plh_1_txt,pl
              h_2_txt,plh_3_txt))
194.
195.         # average of all fold probabilities
196.         if (perfs[2][1] > 0):
197.             avgperfs[2][0] = avgperfs[2][0] + (perfs[2][0] / perfs[2][1])
198.             avgperfs[2][1] += 1
199.             akurasi_txt = "{:0.2f}%".format(perfs[2][0] / perfs[2][1])
200.         else:
201.             akurasi_txt = "NaN"
202.
203.         if (perfs[3][1] > 0):
204.             avgperfs[3][0] = avgperfs[3][0] + (perfs[3][0] / perfs[3][1])
205.             avgperfs[3][1] += 1
206.             presisi_txt = "{:0.2f}%".format(perfs[3][0] / perfs[3][1])
207.         else:
208.             presisi_txt = "NaN"
209.
210.         if (perfs[4][1] > 0):
211.             avgperfs[4][0] = avgperfs[4][0] + (perfs[4][0] / perfs[4][1])
212.             avgperfs[4][1] += 1
213.             recall_txt = "{:0.2f}%".format(perfs[4][0] / perfs[4][1])
214.         else:
215.             recall_txt = "NaN"
216.

```

```

217. if (perfs[5][1] > 0):
218.     avgperfs[5][0] = avgperfs[5][0] + (perfs[5][0] / perfs[5][1])
219.     avgperfs[5][1] += 1
220.     specificity_txt = "{:0.2f}%".format(perfs[5][0] / perfs[5][1])
221. else:
222.     specificity_txt = "NaN"
223.
224. if (perfs[6][1] > 0):
225.     avgperfs[6][0] = avgperfs[6][0] + (perfs[6][0] / perfs[6][1])
226.     avgperfs[6][1] += 1
227.     f1_score_txt = "{:0.2f}%".format(perfs[6][0] / perfs[6][1])
228. else:
229.     f1_score_txt = "NaN"
230.
231. if (perfs[7][1] > 0):
232.     avgperfs[7][0] = avgperfs[7][0] + (perfs[7][0] / perfs[7][1])
233.     avgperfs[7][1] += 1
234.     plh_1_txt = "{:0.2f}%".format(perfs[7][0] / perfs[7][1])
235. else:
236.     plh_1_txt = "NaN"
237.
238. if (perfs[8][1] > 0):
239.     avgperfs[8][0] = avgperfs[8][0] + (perfs[8][0] / perfs[8][1])
240.     avgperfs[8][1] += 1
241.     plh_2_txt = "{:0.2f}%".format(perfs[8][0] / perfs[8][1])
242. else:
243.     plh_2_txt = "NaN"
244.

```

```

245.     if (perfs[9][1] > 0):
246.         avgperfs[9][0] = avgperfs[9][0] + (perfs[9][0] / perfs[9][1])
247.         avgperfs[9][1] += 1
248.         plh_3_txt = "{:0.2f} %".format(perfs[9][0] / perfs[9][1])
249.     else:
250.         plh_3_txt = "NaN"
251.
252.     i += 1
253.
254.     if (prodi != 'semua'):
255.         avgperfs[0][0] = avgperfs[0][0] + (perfs[0][0] / k)
256.         total_txt = "{:0.1f}".format(perfs[0][0] / k)
257.         avgperfs[1][0] = avgperfs[1][0] + (perfs[1][0] / k)
258.         miss_txt = "{:0.1f}".format(perfs[1][0] / k)
259.         print("Average:<br>")
260.     else:
261.         avgperfs[0][0] = avgperfs[0][0] + (perfs[0][0])
262.         total_txt = "{:0.0f}".format(perfs[0][0])
263.         avgperfs[1][0] = avgperfs[1][0] + (perfs[1][0])
264.         miss_txt = "{:0.0f}".format(perfs[1][0])
265.         recordprodi = conn.execute("SELECT prodi FROM data_prodi WHERE kode_prodi =
            {}".format(no_prodi)).fetchall()
266.         nama_prodi = pd.DataFrame(recordprodi, columns=['prodi'])
267.         print("{} . {}:<br>".format(i, ".join(nama_prodi['prodi'].values)))
268.
269.     print("Total: {}, Miss: {}, Akurasi: {}, Presisi: {}, Recall: {}, Specificity: {}, F1 Score: {}, Plh 1: {},
        Plh 2: {}, Plh 3:

```

```

    { }<br>".format(total_txt,miss_txt,akurasi_txt,presisi_txt,recall_txt,specificity_txt,f1_score_txt,plh_1_txt,pl
    h_2_txt,plh_3_txt))
270.
271. if (prodi == 'semua'):
272.     # average of all prodi probabilities
273.     avgperfs[0][0] = avgperfs[0][0] / i
274.     total_txt = "{:0.2f}".format(avgperfs[0][0])
275.
276.     avgperfs[1][0] = avgperfs[1][0] / i
277.     miss_txt = "{:0.2f}".format(avgperfs[1][0])
278.
279.     if (avgperfs[2][1] > 0):
280.         avgperfs[2][0] = avgperfs[2][0] / avgperfs[2][1]
281.         akurasi_txt = "{:0.2f}%".format(avgperfs[2][0])
282.     else:
283.         akurasi_txt = "NaN"
284.
285.     if (avgperfs[3][1] > 0):
286.         avgperfs[3][0] = avgperfs[3][0] / avgperfs[3][1]
287.         presisi_txt = "{:0.2f}%".format(avgperfs[3][0])
288.     else:
289.         presisi_txt = "NaN"
290.
291.     if (avgperfs[4][1] > 0):
292.         avgperfs[4][0] = avgperfs[4][0] / avgperfs[4][1]
293.         recall_txt = "{:0.2f}%".format(avgperfs[4][0])
294.     else:
295.         recall_txt = "NaN"

```

```

296.
297.     if (avgperfs[5][1] > 0):
298.         avgperfs[5][0] = avgperfs[5][0] / avgperfs[5][1]
299.         specificity_txt = "{:0.2f}%".format(avgperfs[5][0])
300.     else:
301.         specificity_txt = "NaN"
302.
303.     if (avgperfs[6][1] > 0):
304.         avgperfs[6][0] = avgperfs[6][0] / avgperfs[6][1]
305.         f1_score_txt = "{:0.2f}%".format(avgperfs[6][0])
306.     else:
307.         f1_score_txt = "NaN"
308.
309.     if (avgperfs[7][1] > 0):
310.         avgperfs[7][0] = avgperfs[7][0] / avgperfs[7][1]
311.         plh_1_txt = "{:0.2f}%".format(avgperfs[7][0])
312.     else:
313.         plh_1_txt = "NaN"
314.
315.     if (avgperfs[8][1] > 0):
316.         avgperfs[8][0] = avgperfs[8][0] / avgperfs[8][1]
317.         plh_2_txt = "{:0.2f}%".format(avgperfs[8][0])
318.     else:
319.         plh_2_txt = "NaN"
320.
321.     if (avgperfs[9][1] > 0):
322.         avgperfs[9][0] = avgperfs[9][0] / avgperfs[9][1]
323.         plh_3_txt = "{:0.2f}%".format(avgperfs[9][0])

```

	<pre> 324. else: 325. plh_3_txt = "NaN" 326. 327. print("Average:
Total: {}, Miss: {}, Akurasi: {}, Presisi: {}, Recall: {}, Specificity: {}, F1 Score: {}, Plh 1: {}, Plh 2: {}, Plh 3: {}".format(total_txt,miss_txt,akurasi_txt,presisi_txt,recall_txt,specificity_txt,f1_score_txt,plh_1_txt,plh_2_t xt,plh_3_txt)) </pre>
--	---



Lampiran 4. Hasil Pengujian Kesesuaian

Dari tahap perancangan peneliti membuat rancangan perhitungan manual mengenai “Pengembangan Sistem Cerdas untuk Prediksi Daftar Kembali Penerimaan Mahasiswa Baru di Undiksha dengan Metode Naive Bayes”, pada tahap *deployment* perhitungan tersebut diimplementasikan menjadi sistem. Adapun uji kesesuaian antara perhitungan manual pada sistem menggunakan metode Naive Bayes adalah sebagai berikut:

Implementasi data train yang sudah diketahui nilai prediksi daftar kembalinya (prodi Pendidikan Teknik Informatika):

Data Train SBMPTN

Import Data

Export Data

Performance Data

Prodi

Pendidikan Teknik Informatika

Tambah

Show50entries

Search:

No	Nama	Nama Sekolah	UKT	Tahun	Pilihan 1	Pilihan 2	Pilihan 3	Prodi	Daftar Kembali	Aksi
1	Putu A	SMKN 3 SINGARAJA	UKT Kelompok 4	2017	Akuntansi (S1)	Pendidikan Teknik Informatika	Pendidikan Guru Sekolah Dasar	Pilihan 2	YA	<div>EditHapus</div>
2	Kadek A	SMAN 4 SINGARAJA	UKT Kelompok 4	2017	Pendidikan Teknik Informatika	Ilmu Hukum	Manajemen	Pilihan 1	TIDAK	<div>EditHapus</div>
3	Komang A	SMA Kertha Wisata	Bidikmisi	2017	Pendidikan Teknik Informatika	Pendidikan Bahasa Inggris		Pilihan 1	YA	<div>EditHapus</div>
4	Ketut A	SMAN 1 SERIRIT	UKT Kelompok 4	2017	Pendidikan Teknik Informatika	Bimbingan dan Konseling		Pilihan 1	YA	<div>EditHapus</div>
5	Putu C	SMAN 1 SERIRIT	UKT Kelompok 4	2017			Pendidikan Teknik Informatika	Pilihan 3	YA	<div>EditHapus</div>
6	Kadek C	SMAN 1 PRAYA TIMUR	UKT Kelompok 4	2017		Pendidikan Teknik Informatika		Pilihan 2	TIDAK	<div>EditHapus</div>
7	Komang C	SMAS HANG TUAH 2 GEDANGAN	UKT Kelompok 7	2017		Pendidikan Teknik Informatika		Pilihan 2	YA	<div>EditHapus</div>
8	Ketut C	SMKN 1 NGASEM	UKT Kelompok 3	2017			Pendidikan Teknik Informatika	Pilihan 3	TIDAK	<div>EditHapus</div>
9	Putu J	SMAN 1 BLAHBATUH	UKT Kelompok 7	2017	Pendidikan Teknik Informatika	Pendidikan Bahasa Inggris		Pilihan 1	YA	<div>EditHapus</div>
10	Kadek J	SMAN 2 SEMARAPURA	UKT Kelompok 7	2017	Pendidikan Teknik Informatika	Ilmu Hukum		Pilihan 1	YA	<div>EditHapus</div>
11	Komang J	SMAN 3 KUPANG	UKT Kelompok 7	2017			Pendidikan Teknik Informatika	Pilihan 3	TIDAK	<div>EditHapus</div>
12	Ketut J	SMK PGRI 1 Denpasar	UKT Kelompok 7	2017			Pendidikan Teknik Informatika	Pilihan 3	TIDAK	<div>EditHapus</div>

13	Putu N	SMAN 2 DENPASAR	UKT Kelompok 7	2018	Pendidikan Teknik Informatika			Pilihan 1	YA	Edit Hapus
14	Kadek N	SMAN 2 DENPASAR	UKT Kelompok 7	2018	Pendidikan Teknik Informatika			Pilihan 1	YA	Edit Hapus
15	Komang N	SMKN 1 SINGARAJA	Bidikmisi	2018	Pendidikan Bahasa Inggris	Pendidikan Teknik Informatika	Pendidikan Ekonomi	Pilihan 2	YA	Edit Hapus
16	Ketut N	SMAN 4 SINGARAJA	UKT Kelompok 7	2018	Manajemen	Pendidikan Teknik Informatika		Pilihan 2	YA	Edit Hapus
17	Putu Q	SMKN 3 SINGARAJA	UKT Kelompok 6	2018	Akuntansi (S1)	Manajemen	Pendidikan Teknik Informatika	Pilihan 3	YA	Edit Hapus
18	Kadek Q	SMKN 3 SINGARAJA	UKT Kelompok 7	2018	Pendidikan Guru Sekolah Dasar	Akuntansi (S1)	Pendidikan Teknik Informatika	Pilihan 3	YA	Edit Hapus
19	Komang Q	SMKN 3 SINGARAJA	UKT Kelompok 7	2018	Pendidikan Teknik Informatika	Manajemen	Akuntansi (S1)	Pilihan 1	TIDAK	Edit Hapus
20	Ketut Q	SMAN 4 SINGARAJA	UKT Kelompok 3	2018	Manajemen	Akuntansi (S1)	Pendidikan Teknik Informatika	Pilihan 3	YA	Edit Hapus
21	Putu R	SMAN 1 KUBUTAMBAHAN	UKT Kelompok 1	2018	Pendidikan Jasmani Kesehatan dan Rekreasi	Pendidikan Teknik Informatika	Akuntansi (S1)	Pilihan 2	YA	Edit Hapus
22	Kadek R	SMA Lain Lain	UKT Kelompok 3	2018	Pendidikan Teknik Informatika	Pendidikan Bahasa dan Sastra Indonesia	Biologi (S1)	Pilihan 1	YA	Edit Hapus
23	Komang R	SMKN 3 SINGARAJA	UKT Kelompok 7	2018	Pendidikan Teknik Informatika	Teknologi Pendidikan	Pendidikan Jasmani Kesehatan dan Rekreasi	Pilihan 1	TIDAK	Edit Hapus
24	Ketut R	SMAN 4 SINGARAJA	UKT Kelompok 7	2018			Pendidikan Teknik Informatika	Pilihan 3	YA	Edit Hapus
25	Putu U	SMKN 3 SINGARAJA	UKT Kelompok 7	2018	Pendidikan Teknik Informatika	Pendidikan Jasmani Kesehatan dan Rekreasi	Teknologi Pendidikan	Pilihan 1	YA	Edit Hapus
26	Kadek U	SMAN 4 SINGARAJA	UKT Kelompok 7	2018	Manajemen	Pendidikan Teknik Informatika	Ilmu Hukum	Pilihan 2	TIDAK	Edit Hapus
27	Komang U	SMAN 3 SINGARAJA	UKT Kelompok 7	2018	Pendidikan Teknik Informatika	Sistem Informasi	Teknologi Pendidikan	Pilihan 1	YA	Edit Hapus
28	Ketut U	SMAN 1 TEJAKULA	UKT Kelompok 7	2018	Akuntansi (S1)	Pendidikan Teknik Informatika	Ilmu Hukum	Pilihan 2	TIDAK	Edit Hapus
29	Komang Y	SMAN 6 MATARAM	UKT Kelompok 6	2018	Pendidikan Teknik Informatika	Sistem Informasi		Pilihan 1	TIDAK	Edit Hapus
30	Ketut Y	SMAN 2 Banjar	UKT Kelompok 6	2018	Pendidikan Guru Sekolah Dasar	Pendidikan Teknik Informatika		Pilihan 2	YA	Edit Hapus

Showing 1 to 30 of 30 entries

Previous 1 Next

< >

Implementasi data test yang akan dicari nilai prediksi daftar kembalinya:

Data Test SBMPTN

Import Data Export Data Predict Data Prodi Pendidikan Teknik Informatika Tambah

Show 10 entries Search:

No	Nama	Nama Sekolah	UKT	Tahun	Pilihan 1	Pilihan 2	Pilihan 3	Prodi	Daftar Kembali	Aksi
1	Putu Z	SMKN 3 SINGARAJA	UKT Kelompok 3	2019	Pendidikan Teknik Informatika	Bimbingan dan Konseling		Pilihan 1		Edit Hapus
2	Kadek Z	SMKN 1 SINGARAJA	UKT Kelompok 4	2019	Pendidikan Bahasa Inggris	Pendidikan Teknik Informatika	Pendidikan Ekonomi	Pilihan 2		Edit Hapus
3	Komang Z	SMK PGRI 1 Denpasar	UKT Kelompok 3	2019	Akuntansi (S1)		Pendidikan Teknik Informatika	Pilihan 3		Edit Hapus
4	Ketut Z	SMAN 4 SINGARAJA	UKT Kelompok 7	2019	Manajemen	Pendidikan Teknik Informatika	Ilmu Hukum	Pilihan 2		Edit Hapus

Showing 1 to 4 of 4 entries Previous 1 Next

Dari inputan yang telah dilakukan pada perhitungan manual, digunakan persamaan berikut untuk mendapat probabilitas masing-masing atribut yang diperlukan:

$$\rho_i = \frac{m_i}{n}$$

Berikut merupakan proses untuk mendapatkan nilai probabilitas:

$$P(YA) = 20/30 = 0.67$$

$$P(\text{nama_sekolah} = \text{SMKN 3 Singaraja} \mid YA) = 4/20 = 0.2$$

$$P(\text{ukt} = \text{UKT Kelompok 3} \mid YA) = 2/20 = 0.1$$

$$P(\text{pilihan_1} = \text{Pendidikan Teknik Informatika} \mid YA) = 9/20 = 0.45$$

$$P(\text{pilihan_2} = \text{Bimbingan dan Konseling} \mid YA) = 1/20 = 0.05$$

$$P(\text{pilihan_3} = - \mid YA) = 9/20 = 0.45$$

$$P(\text{prodi} = \text{Pilihan 1} \mid YA) = 9/20 = 0.45$$

$$P(\text{TIDAK}) = 10/30 = 0.33$$

$$P(\text{nama_sekolah} = \text{SMKN 3 Singaraja} \mid \text{TIDAK}) = 2/10 = 0.2$$

$$P(\text{ukt} = \text{UKT Kelompok 3} \mid \text{TIDAK}) = 1/10 = 0.1$$

$$P(\text{pilihan_1} = \text{Pendidikan Teknik Informatika} \mid \text{TIDAK}) = 4/10 = 0.4$$

$$P(\text{pilihan}_2 = \text{Bimbingan dan Konseling} \mid \text{TIDAK}) = 0/10 = 0$$

$$P(\text{pilihan}_3 = - \mid \text{TIDAK}) = 2/10 = 0.2$$

$$P(\text{prodi} = \text{Pilihan 1} \mid \text{TIDAK}) = 4/10 = 0.4$$

Jika ada probabilitas yang bernilai nol maka dilakukan perhitungan tersebut dengan menerapkan Laplace Correction pada semua data dalam satu atribut yang sama:

$$\rho_i = \frac{m_i + 1}{n + k}$$

Berikut merupakan proses untuk mendapatkan nilai probabilitas nol:

$$P(\text{pilihan}_2 = \text{Bimbingan dan Konseling} \mid \text{YA}) = \frac{1+1}{20+10} = 0.07$$

$$P(\text{pilihan}_2 = \text{Bimbingan dan Konseling} \mid \text{TIDAK}) = \frac{0+1}{10+6} = 0.06$$

Kemudian perbandingan probabilitas YA dan TIDAK dari daftar_kembali dapat dicari dengan menggunakan persamaan dari teorema Bayes (Bustami, 2010):

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Berikut merupakan proses untuk mendapatkan nilai daftar_kembali:

$$P(\text{YA}|\text{daftar_kembali}) = 0.67 \times 0.2 \times 0.1 \times 0.45 \times 0.07 \times 0.45 \times 0.45 = 8.1 \times 10^5$$

$$P(\text{TIDAK}|\text{daftar_kembali}) = 0.33 \times 0.2 \times 0.1 \times 0.4 \times 0.06 \times 0.2 \times 0.4 = 1.33 \times 10^5$$

Karena nilai probabilitas daftar_kembali YA lebih tinggi dari nilai daftar_kembali TIDAK, maka didapat kesimpulan bahwa nilai daftar_kembali pada data test Putu Z adalah YA. Untuk Kadek Z, Komang Z, dan Ketut Z masing-masing mendapatkan YA, TIDAK, dan TIDAK pada nilai masing-masing daftar_kembali dengan menggunakan tahapan yang sama dengan mencari nilai daftar_kembali pada data test Putu Z.

Berdasarkan hasil perhitungan manual dan sistem tersebut terlihat bahwa tidak ada perbedaan hasil prediksi antara prediksi manual dan prediksi yang dilakukan pada sistem, sehingga dapat dikatakan bahwa hasil antara perhitungan manual dan sistem sudah sesuai.



Lampiran 5. Pengujian Respon Pengguna

**PENGUJIAN SISTEM USABILITY SCALE PADA SISTEM CERDAS
UNTUK PREDIKSI DAFTAR KEMBALI PENERIMAAN MAHASISWA
BARU DI UNDIKSHA DENGAN METODE NAIVE BAYES**

Nama Responden : *I Gsa Pt Anom Arinkupa*

Petunjuk :

Berikan tanda centang (✓) pada salah satu kotak yang telah disediakan.
1 = Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Kurang Setuju, 4 = Setuju, 5 = Sangat Setuju

No	Pertanyaan	Jawaban				
		1	2	3	4	5
1	Saya merasa sistem mudah untuk digunakan					✓
2	Saya merasa sistem memiliki tampilan yang bagus					✓
3	Saya merasa sistem sudah mencakup semua yang dibutuhkan				✓	
4	Saya merasa bahwa fungsi-fungsi yang terintegrasi dalam sistem sudah berjalan dengan baik					✓
5	Saya merasa konten yang terdapat pada sistem sudah konsisten				✓	
6	Saya merasa sistem tidak pernah mengalami gangguan dalam penggunaannya				✓	
7	Saya merasa sistem sudah berjalan dengan cepat				✓	
8	Saya merasa keakuratan prediksi dari sistem bisa dipercaya				✓	
9	Saya merasa keamanan data yang ada di sistem sudah terjamin					✓
10	Saya merasa saya akan sering menggunakan sistem ini				✓	

Singaraja, ³¹..... Januari 2020

Anwar
(1st Pl Anwar Gimbakan)



Lampiran 6. Hasil Pengujian Respon Pengguna

Adapun cara perhitungan kuesioner SUS adalah sebagai berikut.

1. Menentukan bobot penilaian

Sangat Tidak Setuju = 0

Tidak Setuju = 1

Kurang Setuju = 2

Setuju = 3

Sangat Setuju = 4

Penentuan bobot ditentukan berdasarkan skala Likert.

2. Kemudian setiap pernyataan dikalikan dengan 2,5 lalu dijumlahkan.

3. Adapun rating dan skala konversi skor rerata SUS adalah sebagai berikut:

0-25 = *worst imaginable*

26-39 = *poor*

40-52 = *ok*

53-74 = *good*

75-85 = *excellent*

86-100 = *best imaginable*

Adapun perhitungan pengujian respon pengguna yang telah dilakukan peneliti:

Bobot	Jumlah	Nilai
Sangat Tidak Setuju	0	0
Tidak Setuju	0	0
Kurang Setuju	0	0
Setuju	6	18
Sangat Setuju	4	16
Total		34

Hasil akhir = $34 \times 2,5 = 85$

Dari hasil perhitungan pengujian respon pengguna diperoleh nilai SUS sebesar 85%. Hal ini menunjukkan bahwa tingkat usability sistem tergolong dalam kategori *excellent* dan dapat diterima serta digunakan dengan mudah oleh pengguna.

