Jurnal Mahajana Informasi. Vol. 7, No. 1, Juni 2022
e-ISSN : 2527-8290

TEKNIK KLASIFIKASI PREDIKSI KELULUSAN MAHASISWA SISTEM INFORMASI UNIVERSITAS SARI MUTIARA INDONESIA MENGGUNAKAN K- NEAREST NEIGHBORS

Intan P.J Tafonao[1)], Alexander F.K. Sibero[2*)]
[1,2]Program Studi Sistem Informasi Universitas Sari Mutiara Indonesia Medan Jl. Kapten Muslim No.79 Medan 20123 Medan Telp (061)-8476769
e-mail : alexsibero@gmail.com

Abstrak

Tidak stabilnya rasio tingkat kelulusan mahasiswa program studi Sistem Informasi pada Universitas Sari Mutiara Indonesia menciptakan kondisi yang membuat adanya suatu penumpukan data. Teknik data mining dapat digunakan untuk memprediksi kelulusan tepat waktu mahasiswa. Penelitian menggunakan metode k-Nearest Nieghbors yang merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data training yang jaraknya paling dekat dengan objek tersebut. Pada penelitian ini menggunakan data mahasiswa tahun angkatan 2014 sampai dengan 2017 dengan jumlah data sebanyak 104 orang. Hasil dari perhitungan algoritma kNN diimplememetasikan dengan jupyter notebook. Tingkat akurasi pengujian model kelulusan mahasiswa dengan menggunakan algoritma k-Nearest Neighbor (k NN) dipengaruhi oleh missing value indeks prestasi semester**.** missing value diganti dengan angka 0 maka hasil akurasi tertinggi adalah 95% dengan k=3. Jadi k terbaik adalah k=3 berdasarkan indeks prestasi sampai dengan semester 6.

Kata Kunci :K-Nearest Neighbors (kNN), Data Mining, Klasifikasi

1. PENDAHULUAN

Data akademik yang ada pada suatu perguruan tinggi akan semakin bertambah seiring dengan berlangsungnya proses kegiatan akademik. Tidak stabilnya rasio tingkat kelulusan mahasiswa program studi Sistem Informasi pada Universitas Sari Mutiara Indonesia menjadi tugas yang berat bagi program studi. Hal ini menciptakan kondisi yang membuat adanya suatu penumpukan data.

Perguruan tinggi dituntut untuk menyelenggarakan pendidikan yang berkualitas bagi mahasiswa sehingga menghasilkan sumber daya manusia yang memiliki daya saing. Dalam perguruan tinggi mahasiswa merupakan aset yang sangat penting bagi institusi pendidikan oleh karena itu perlu diperhatikan tingkat kelulusan mahasiswa tepat pada waktunya [1]. *Data mining* merupakan proses penggalian informasi dan berguna dari set data besar yang melibatkan konsep interdisipliner yang relatif baru yang melibatkan analisis data dan penemuan pengetahuan dari database dan menggunakan pendekatan multi-sisi yang mencakup analisis statistik, visualisasi data, penemuan pengetahuan, pengenalan pola dan manajemen basis data [2]. Algoritma yang digunakan untuk melakukan fungsi klasifikasi adalah algoritma *kNN*. Algoritma *k-Nearest Neighbors* adalah algoritma supervised learning dimana hasil dari instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori k-tetangga atau jarak terdekat [3].

Algoritma *K-Nearest Neighbor* (*kNN*) merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan (k) tetangga terdekatnya [4]. Penelitian sebelumnya oleh Mustakim, prediksi dilakukan terhadap mahasiswa program studi sistem informasi tahun ajaran 2014 sampai 2015 sebagai data *testing* dengan jumlah 50 data, serta berdasarkan dari data mahasiswa tahun ajaran 2012 sampai 2013 sebagai data *training* dengan jumlah sebanyak 165 data yang menghasilkan pengujian akurasi sebesar 82% [5].

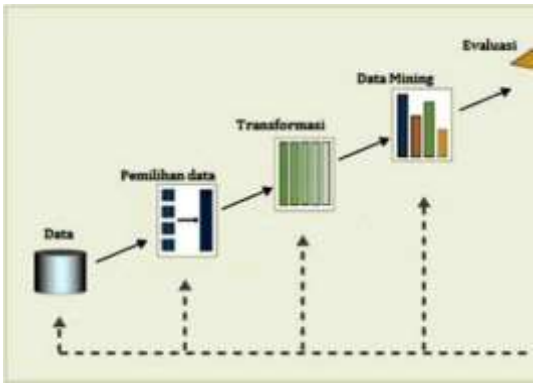
Oleh karena itu dengan adanya sebuah prediksi kelulusan mahasiswa dengan teknik klasifikasi menggunakan *kNN* diharapkan dapat digunakan oleh program studi untuk mencari solusi dan memberi perhatian khusus terhadap mahasiswa yang di prediksi lulus tidak tepat waktu, sehingga mahasiswa tersebut diharapkan dapat memperbaiki indeks prestasinya agar dapat lulus pada waktu yang seharusnya.

2. TINJAUAN PUSTAKA

2.1 *Data Mining*

Data mining adalah proses yang menggunakan teknik statik, matematika, kecerdasan buatan dan machine learning untuk mengekstraksi dan mengindentifikasi informasi yang bermanfaat serta mendapatkan pengetahuan dari berbagai basis data yang berukuran besar [6].

Knowledge Discovery In Database (KDD) dapat memproses seluruh data non-trivial untuk mengetahui pola dalam data, dimana pola yang ditemukan memiliki sifat yang sah dan mudah untuk dipahami :



Gambar 1. Proses KDD

Adapun tahapan *Knowledge Discovery In Database* (KDD) adalah :

1. Data
   Membuat himpunan data target, penetapan himpunan data dan memfokuskan pada subset variabel atau sampel data, dimana penelitiaan akan dilakukan.
2. Pemilihan Data
   Langkah pertama pemprosesan data dan pembersihan data adalah tindakan dasar seperti penghapusan *noise*.
3. Transformasi
   Pada tahap ini merupakan tahapan proses kreatif dan sangat tergantung pada pola informasi yang akan dicari dalam basis data.
4. Data Mining
   Dalam pemilihan algoritma data mining untuk melakukan pencarian proses data mining yaitu antara lain teknik, metode atau algoritma dalam data mining sangat bervariasi.
5. Evaluasi
   Tahap ini merupakan tahapan pemerikasaan, apakah pola yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

2.2 *k-Nearest Neighbors* (*kNN*)

Algoritma *k-Nearest Neighbors* (*kNN*) salah satu teknik klarisifikasi data yang kuat, dengan cara mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama berdasarkan pencocokan bobot [7].

Metode *kNN* dibagi menjadi dua fase, yaitu pembelajaran (*training*) dan klasifikasi atau pengujian (*testing*) [7]. Secara umum untuk mendefinisikan jarak antara dua *objek* x dan y, digunakan rumus jarak *Euclidien* dengan persamaan berikut:

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (\quad - \quad)^2}$$

Keterangan :

Xtraining	: data *training* ke-i,
Ytesting	: data *testing*,
i	: *record* (baris) ke-i dari tabel,
n	: jumlah data *training*.

2.3 Klasifikasi

Klasifikasi merupakan proses penemuan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui [8].

2.4 *Scikit-Learn (sklearn)*

Scikit-learn atau *sklearn* adalah modul untuk bahasa pemograman Python yang dibangun diatas *NumPy*, *SciPy*, dan *matplotlib*, dimana fungsinya dapat membantu melakukan pengolahan data ataupun melakukan pelatihan data untuk kebutuhan *machine learning* [9]. Ada banyak fitur yang dapat digunakan dengan *sklearn* ini, seperti *Classification, Regression, Clustering, Dimensionality reduction, Model selection,* dan *Preprocessing data.*

3. METODE PENELITIAN

3.1 Analisa Masalah

Tidak stabilnya tingkat kelulusan mahasiswa Sistem Informasi pada Universitas Sari Mutiara Indonesia mengakibatkan adanya penumpukan data dan menjadi tugas berat bagi program studi. Oleh karena itu, untuk meningkatkan kualitas pada perguruan tinggi Universitas Sari Mutiara Indonesia pada Fakultas Sain, Teknogi dan Informasi Program Studi Sistem Informasi, maka haruslah dilihat pola kelulusan mahasiswa pada setiap tahunnya. Data yang digunakan adalah data IP mahasiswa dari semester satu sampai dengan semester lima menggunakan salah satu fungsi data mining yaitu fungsi klasifikasi dengan menggunakan algoritma *k-Nearest Neighbors.*

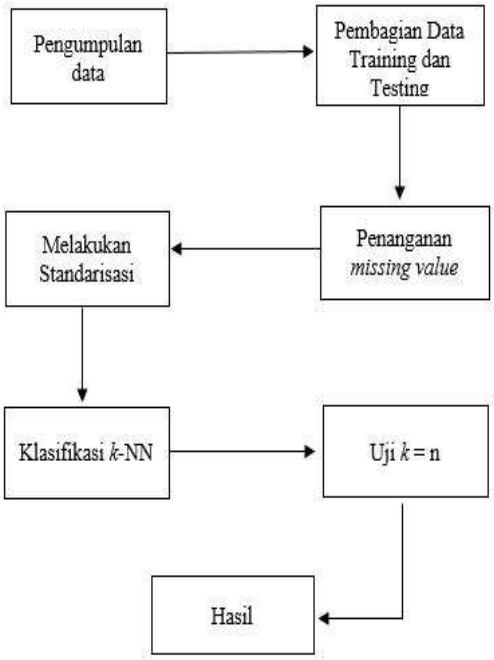
Sumber data yang digunakan berasal dari Fakultas Sain Teknologi dan Informasi

Program Studi Sistem Informasi Tahun Ajar mulai dari 2014 sampai dengan 2017. Atribut data yang digunakan untuk melakukan proses prediksi kelulusan mahasiswa yaitu : Nama, Nomor Induk Mahasiswa, Indeks Prestasi semester 1 (satu) sampai 6 (enam) dan status keterangan lulus. Pada penelitian ini nilai k yang digunakan adalah 5 (lima).

Berdasarkan masalah yang telah diuraikan di atas maka pemecahan masalahnya adalah dengan melakukan pengklasifikasian yang menggunakan Data Mining dengan Algoritma *k-Nearest Neighbors* untuk melihat setiap mahasiswa akan lulus tepat waktu atau terlambat.

3.2 Alur Penelitian

Alur penelitian ini merupakan gambaran dasar mengenai proses penelitian dengan metode *k-Nearest Neighbors* yang digunakan dalam penelitian, dapat dilihat pada gambar di bawah ini :



Gambar 2. Alur Penelitian

Penjelasan alur diatas adalah sebagai berikut:

1. Pengumpulan Data
   Data yang dikumpulkan merupakan data mahasiswa sistem informasi tahun ajaran 2014 sampai 2017, dan data yang diperoleh sebanyak 104 data.
2. Pembagian Data *Training* dan *Testing*
   Data *training* memiliki proporsi lebih besar dibandingkan data *testing*. Pada penelitian ini dataset dibagi menjadi 80% data *training*, dan 20% data *testing*. Memisahkan data menjadi *training* dan *testing* dimaksudkan agar model yang diperoleh nantinya memiliki kemampuan generalisasi yang baik dalam melakukan klasifikasi data.
3. Melakukan Penanganan *Missing Value Missing value* biasanya muncul sebagai *Not a Number* (*Nan*), ?, atau tidak ada nilai nya sama sekali alias *blank cell*. Untuk kasus ini mengganti nilai *missing value* adalah pilihan yang lebih baik karena tidak ada data yang terbuang. Nilai standar yang biasanya digunakan untuk menggantikan *missing value* adalah dengan nilai rata-rata dari seluruh nilai dalam variabel atau kolom tersebut.
4. Melakukan Standarisai Numerik
   Untuk menyamakan skala data agar terdistribusi normal dan meningkatkan performa model *kNN*.
5. Klasifikasi *k-Nearest Neighbors*
   Proses *training* menggunakan metode *kNN* dijalankan dan menggunakan *k*=5.
6. Uji nilai *k*=n
   Menguji setiap data tes dengan *k*=2,3,4,5,...,24. Dan mendapatkan nilai akurasi dari setiap *k*=n.
7. Hasil
   Prediksi kelulusan dengan nilai akurasi yang didapatkan

Untuk pengumpulan data yang digunakan sebagai data *training* dan data *testing* dalam penelitian ini di dapat dengan mengumpulkan data dari admin sistem informasi Universitas Sari Mutiara Indonesia. Data yang didapat merupakan data mahasiswa sistem informasi tahun ajaran 2014 sampai 2017. Data yang di peroleh sebanyak 104 data dengan atribut Nomor Induk Mahasiswa, Nama Mahasiswa, Indeks Prestasi Semester 1 sampai Indeks Prestasi semester 6 dan Keterangan lulus (Tepat waktu/ Terlambat).

3.4 Sample Data

Berikut adalah sample data mahasiswa yang digunakan dalam penelitian ini :

Tabel 1. Sample data mahasiswa

NIM	Nama	Indeks Prestasi					
		sem1	sem2	sem3	sem4	sem5	sem6
140818001	Armina Pusry Mawa	3.58	3.7	3.84	3.7	3.34	3.66
140818002	Devy Haryanti Hasib	0.38	0	0	0	0	0
140818003	Merlina Sitompul	2.88	3.14	3.03	3.43	3.25	3.24
140818009	Muhammad Dibigarm	2.94	0	0	0	0	0
...	.....	...	...	...	...	...	...
...	.....	...	...	...	...	...	...
150818009	Satria Simbolon	3.17	2.82	3.13	3.68	3.4	3.69
150818010	Sriwahyuni Sigalinggi	3.52	3.34	3.26	3.6	3.55	3.68
160416005	Caritas Tri Dewi Opi	3.25	3.45	3.64	3.65	3.4	3.6
160416006	Devi Jayantri Sipahut	3.05	3.24	3	3.13	3.32	3.63
170416003	Ahmad Fiqih Siregar	3.8	0	3.68	0	0	0
170416004	Ade Kartika Telaumb	3.1	3.18	3.43	3.1	2.82	3.53
170416005	Ade Ria Murdani	3.6	3.39	3.68	3.3	3.55	3.61

4. HASIL DAN PEMBAHASAN

4.1 Perhitungan Secara Manual Algoritma *kNN* pada Dataset

Setelah data *training* dan data *testing* sudah didapatkan langkah selanjutnya melakukan langkah-langkah dalam Algoritma *kNN*.

1. Menentukan Paramater *k*
   Disini paramater yang dipakai adalah *k*=5
2. Menghitung kuadrat jarak terkecil berdasarkan perhitungan jarak *euclidean distance* masing-masing obyek terhadap data sample yang diberikan

Contoh perhitungan data *testing* mahasiswa ke 1 terhadap ke 83 data *training*:

$$\sqrt{(2.94-3.58)^2+(2.50-3.70)^2+(2.21-3.84)^2+(1.30-3.70)^2+(1.11-3.34)^2+(0.00-3.66)^2}$$

= 5.3511

$$\sqrt{(2.78-3.58)^2+(2.95-3.70)^2+(2.47-3.84)^2+(1.68-3.70)^2+(0.93-3.34)^2+(0.26-3.66)^2}$$

= 4.9525

$$\sqrt{(2.88-3.58)^2+(0.00-3.70)^2+(0.00-3.84)^2+(0.00-3.70)^2+(0.00-3.34)^2+(0.00-3.66)^2}$$

= 8.1955

… …..

$$\sqrt{(3.45-3.58)^2+(3.37-3.70)^2+(3.38-3.84)^2+(2.90-3.70)^2+(2.40-3.34)^2+(3.47-3.66)^2}$$

= 1.3773

$$\sqrt{(3.05-3.58)^2+(3.05-3.70)^2+(3.28-3.84)^2+(2.25-3.70)^2+(2.82-3.34)^2+(3.29-3.66)^2}$$

= 1.8779

Berikut hasil dari *euclidien distance* masing-masing objek terhadap data *testing* yang diberikan :

Table 4. *Euclidien distance* data *testing*

NIM	Data Uji 1	Data Uji 2	Data Uji 3	…	Data Uji 21
140818006	5.351168	4.5398	4.54969	…	4.353964
140818007	4.952565	4.93136	4.20535	…	4.012667
140818008	8.195535	2.5	7.20177	…	7.038722
150818001	1.260159	7.73077	0.59169	…	0.926715
150818002	8.165684	3.16	7.23195	…	7.044345
150818003	0.680588	8.54506	1.11369	…	1.227192
…	…	…	…	…	…
160416003	0.639687	8.27091	0.85317	…	0.798874
160416009	0.536563	8.81691	1.29892	…	1.376481
160416010	8.198152	2.47	7.20183	…	7.039936
170416064	1.253116	7.58321	0.89476	…	0.501099
170416065	1.377353	7.64311	1.24764	…	0.665207
170416066	1.877978	7.14344	1.29588	…	0.627057

3. Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak euclidian terkecil. Setelah objek-objek tersebut di urutkan maka menentukan peringkat dari data yang mempunyai jarak terkecil ke terbesar.
4. Mengumpulkan kategori Y (klasifikasi nearest neighbor) Setelah mendapatkan peringkat pada jarak, maka pada langkah selanjutnya mengumpulkan kategori Y.

Table 6. Hasil Klasifikasi k-NN

NIM	Keterangan	Euclidean Distance data	Peringkat Jarak Terdekat	Klasifikasi
170416002	Tepat Waktu	0.29816103	2	Ya
170416013	Tepat Waktu	0.367014986	6	Tidak
170416022	Tepat Waktu	0.301330383	3	Ya
170416024	Tepat Waktu	0.267020598	1	Ya
170416025	Tepat Waktu	0.421188794	10	Tidak
170416026	Tepat Waktu	0.339558537	5	Ya
170416030	Tepat Waktu	0.399374511	8	Tidak
170416044	Tepat Waktu	0.326036808	4	Ya
170416050	Tepat Waktu	0.380525952	7	Tidak
170416061	Tepat Waktu	0.41521079	9	Tidak

5. Dengan menggunakan kategori mayoritas,maka dapat hasil klasifikasi Setelah mengumpulkan kategori Y dengan *k* =5 maka langkah selanjutnya adalah mengumpulkan kategori mayoritas. Hasil dari kategori mayoritas dapat di lihat pada tabel di bawah ini :

Tabel 7. Hasil Kategori Mayoritas

NIM	Keterangan	Euclidean Distance data testing	Peringkat Jarak	Klasifikasi
170416002	Tepat Waktu	0.29816103	2	Ya
170416022	Tepat Waktu	0.301330383	3	Ya
170416024	Tepat Waktu	0.267020598	1	Ya
170416026	Tepat Waktu	0.339558537	5	Ya
170416044	Tepat Waktu	0.326036808	4	Ya

Dari hasil klasifikasi diatas dapat dilihat pada mayoritas klasifikasi yang muncul adalah "Ya", sehingga data *testing* yang ingin diklasifikasikan termasuk kategori Tepat Waktu. Hasil data *testing* dapat dilihat pada tabel dibawah ini :

Tabel 8. Hasil data *testing*

NIM	Index Prestasi						Keterangan
	sem1	sem2	sem3	sem4	sem5	sem6	
140818001	3.58	3.7	3.84	3.7	3.34	3.66	Tepat Waktu
140818002	0.38	0	0	0	0	0	Terlambat
140818003	2.88	3.14	3.03	3.43	3.25	3.24	Tepat Waktu
150818006	3.63	0	0	0	0	0	Terlambat
150818007	3.91	3.89	3.61	3.7	3.76	3.91	Tepat Waktu
160416004	3.25	3.45	3.31	3.5	3.32	3.66	Tepat Waktu
160416005	3.25	3.45	3.64	3.65	3.4	3.6	Tepat Waktu
160416008	3.05	2.97	3.36	2.95	3.05	0	Terlambat
170416051	3.5	3.18	3.15	1.95	0	0	Terlambat
170416055	3.55	3.63	3.75	3.55	3.09	3.5	Tepat Waktu
170416056	3.15	3.18	3.35	2.83	2.95	3.38	Tepat Waktu

4.2 Perhitungan Algoritma *kNN* menggunakan *Jupyter Notebook*

Proses klasifikasi *kNN* pada *jupyter notebook* adalah sebagai berikut :

1. Install terlebih dahulu *packages* dibawah ini dan kemudian *import package* yang akan digunakan yaitu :
   - *pandas as pd* digunakan untuk proses analisis data seperti manipulasi data, persiapan data, dan pembersihan data.
   - *numpy as np* digunakan untuk hal-hal yang berbau angka, tepatnya untuk perhitungan aljabar seperti operasi vektor dan matriks.
   - *seaborn as sns* digunakan untuk menghasilkan visualisasi yang lebih menarik .
   - *matplotlib.pyplot as plt* digunakan untuk visualisasi dasar seperti membuat plot grafik.

2. *Import dataset* menggunakan *library pandas* lalu read_csv (nama file csv) dan tampilkan.
3. Data dibagi menjadi dua jenis: fitur (X) dan label (y).
   Lokasi X [:, 2:8] artinya: pilih semua baris pada *dataset*, ada 9 kolom pada *dataset*, kita memilih enam kolom: indeks 2, 3, 4, 5, 6 dan 7.
   Lokasi y [:, -1] artinya: pilih semua baris pada *dataset*, dan kolom terakhir

4. Menangani nilai kosong *Not a Number* (*NaN*). Seperti terlihat pada *dataset* ada beberapa nilai indeks prestasi yang tidak lengkap (*missing value*). Untuk mengatasinya maka digunakan fungsi *SimpleImputer* yang diimport dari *Scikit-Learn* agar nilai yang kosong digantikan dengan mean/rata-rata kolom itu.

5. Mengkodekan kolom label y menggunakan fungsi *LabelEncoder* yang diimport dari *Scikit-Learn* yang nilai datanya bertipe string diubah menjadi numerik (0 dan 1) agar dapat digunakan dengan baik. 0 = Tepat Waktu dan 1 = Terlambat.

```
In [8]: from sklearn.preprocessing import LabelEncoder
        labelencoder_y = LabelEncoder()
        y = labelencoder_y.fit_transform(y)

In [9]: y

Out[9]: array([0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
               1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
               0, 0, 0, 0, 0, 0, 0, 1, 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,
               1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0])
```

6. Membagi dataset ke dalam data *training* dan data *testing*. Biasanya data *training* memiliki proporsi lebih besar dibandingkan data *testing*. Pada penelitian ini *dataset* sejumlah 104 data dibagi menjadi 80% untuk data *training*, dan 20% untuk data *testing* sehingga parameter *test*_size=0.2.

7. Mengaktifkan *package StandarScaler* dari *Sklearn* untuk proses standarisasi data agar terdistribusi normal dan meningkatkan kinerja model *kNN* ini.

```
In [15]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()

         X_train = sc.fit_transform(X_train)
         X_test = sc.transform(X_test)
```

8. Mengaktifkan package untuk klasifikasi *kNN* dengan mengimport *package k-Nearest Neighbors* dari *Sklearn*. Berikutnya mengaktifkan fungsi klasifikasi untuk *kNN* (knn) dan melakukan proses *training* menggunakan nilai *k*=5. Kemudian memasukkan data *training* pada fungsi klasifikasi untuk *kNN*.

```
In [18]: from sklearn.neighbors import KNeighborsClassifier
In [19]: knn = KNeighborsClassifier(n_neighbors=5, metric='euclidean')
In [20]: knn.fit(X_train, y_train)
Out[20]: KNeighborsClassifier(metric='euclidean')
```

9. Selanjutnya menentukkan prediksi atau peramalannya. Pada hasil prediksi klasifikasi kelas 0 berjumlah 18 (Tepat Waktu) sedangkan kelas 1 berjumlah 3 (Terlambat).

10. Import *package confusion_matrix* untuk melihat keakuratan data hasil prediksi dengan data aktualnya. Kemudian menampilkan matriks hasil prediksinya.

```
In [23]: from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test, y_pred)
         print(cm)

         [[15  0]
          [ 3  3]]
```

Berdasarkan matriks hasil prediksi diatas, didapatkan hasil memprediksikan secara benar yaitu sebanyak 15 dengan selisih kesalahan 0 dan yang sesuai prediksi salah sebanyak 3 dengan selisih kesalahan yaitu sebanyak 3.

11. Menentukkan probabilitas dari prediksi.

12. Kemudian mengukur ketepatan atau keakuratan hasil prediksi.

Precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif

Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif

F1-Score merupakan perbandingan rata-rata presisi dan recall.

```
In [25]: from sklearn.metrics import classification_report
         print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.83	1.00	0.91	15
1	1.00	0.50	0.67	6
accuracy			0.86	21
macro avg	0.92	0.75	0.79	21
weighted avg	0.88	0.86	0.84	21

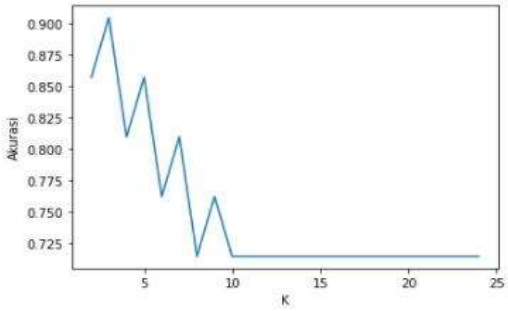
13. Akurasi hasil prediksi dengan keseluruhan data sebesar 85 %.

4.3 Hasil dan Pembahasan

Pada algoritma *k-Nearest Neighbors* dilakukan pengujian terhadap dataset dengan *missing value* IP (Indeks Prestasi) semester 1 sampai dengan 5, pengujian dataset dengan *missing value* IP (Indeks Prestasi) semester 1 sampai dengan 6 dan pengujian terhadap *dataset* dimana *missing value Not a Number* (*NaN*) diganti dengan rata-rata kolom itu. Pengujian klasifikasi ini dilakukan untuk menemukan *k* yang sesuai, dilihat dari akurasi yang dihasilkan. Semakin tinggi akurasi, maka nilai *k* tersebut yang akan dijadikan referensi untuk semua data set.

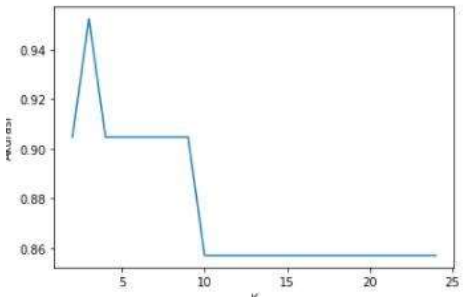
Setelah dilakukan pengujian terhadap *dataset* maka didapatkan perbandingan tingkat akurasi dengan *k* yang berbeda-beda. Berikut hasil akurasi k terbaik:

1. Pengujian terhadap dataset dimana *missing value* (*NaN*) diganti dengan rata-rata kolom.



Pada grafik diatas menunjukan bahwa *k*=2 memiliki nilai akurasi 86%, *k*=3 nilai akurasi 90%, *k*=4 nilai akurasi 81%, *k*=5 nilai akurasi 85%, *k*=6 nilai akurasi 76%, *k*=7 nilai akurasi 80%, *k*=8 nilai akurasi 71%, *k*=9 nilai akurasi 76%, dan turun lagi pada *k*=10 sampai *k*=24 dengan nilai akurasi 71%.

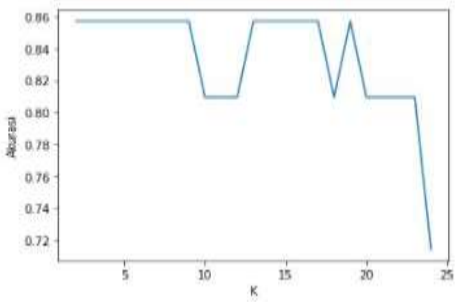
2. Pengujian *dataset* dengan *missing value* IP (Indeks Prestasi) semester 1 sampai dengan 6. Dimana pada *dataset* nilai indeks prestasi semester 1 sampai 6 diganti numerik dengan angka 0 pada nilai yang kosong.



Pada grafik diatas menunjukan bahwa *k*=2 memiliki nilai akurasi 90%, *k*=3 nilai akurasi

95%, *k*=4 sampai *k*=9 nilai akurasi 90%, pada *k*=10 sampai *k*=24 nilai akurasi 86%.

3. Pengujian *dataset* dengan *missing value* IP (Indeks Prenstasi) semester 1 sampai dengan 5. Dimana pada dataset nilai indeks prestasi semester 1 sampai 5 diganti numerik dengan angka 0 pada nilai yang kosong.



Pada grafik diatas menunjukan bahwa *k*=2 sampai *k*=9 memiliki nilai akurasi 86%, *k*=10 sampai *k*=12 nilai akurasi 81%, *k*=13 sampai *k*=17 nilai akurasi 86%, *k*=18 nilai akurasi 81%, *k*=19 nilai akurasi 86%, *k*=20 sampai *k*=23 nilai akurasi 81% dan *k*=24 nilai akurasi 71%.

Berdasarkan hasil dari akurasi setiap pengujian model, didapatkan k terbaik yaitu k 3 dengan nilai akurasi 95%, menggunakan dataset dimana nilai indeks prestasi semester 1 sampai 6 diganti numerik dengan angka 0 pada nilai yang kosong.

4.4. Hasil Prediksi Kelulusan

Dari hasil klasifikasi prediksi berdasarkan pengujian data *testing* didapatkan sebanyak 15 orang diprediksi Tepat Waktu dan 6 orang diprediksi Terlambat. Hasil prediksi dapat dilihat pada tabel dibawah:

Tabel 9 Hasil prediksi kelulusan

NIM	Index Prestasi						Keterangan
	sem1	sem2	sem3	sem4	sem5	sem6	
140818001	3.58	3.7	3.84	3.7	3.34	3.66	Tepat Waktu
140818002	0.38	0	0	0	0	0	Terlambat
140818003	2.88	3.14	3.03	3.43	3.25	3.24	Tepat Waktu
140818004	2.69	0	0	0	0	0	Terlambat
140818005	3.27	3.23	3.24	3.55	3.16	3.82	Tepat Waktu
150818006	3.63	0	0	0	0	0	Terlambat
150818007	3.91	3.89	3.61	3.7	3.76	3.91	Tepat Waktu
150818008	3.35	2.26	2.82	0	0	0	Terlambat
150818009	3.17	2.82	3.13	3.68	3.4	3.69	Tepat Waktu
150818010	3.52	3.34	3.26	3.6	3.55	3.68	Tepat Waktu
160416004	3.25	3.45	3.31	3.5	3.32	3.66	Tepat Waktu
160416005	3.25	3.45	3.64	3.65	3.4	3.6	Tepat Waktu
160416006	3.05	3.24	3	3.13	3.32	3.63	Tepat Waktu
160416007	3.75	3.55	3.69	3.48	3.5	0	Tepat Waktu
160416008	3.05	2.97	3.36	2.95	3.05	0	Terlambat
170416051	3.5	3.18	3.15	1.95	0	0	Terlambat
170416052	2.95	3.13	3.28	3	2.82	3.24	Tepat Waktu
170416053	3.45	3.55	3.63	3.4	3.36	3.55	Tepat Waktu
170416054	3.05	3.29	3.35	2.8	1.64	3.21	Tepat Waktu
170416055	3.55	3.63	3.75	3.55	3.09	3.5	Tepat Waktu
170416056	3.15	3.18	3.35	2.83	2.95	3.38	Tepat Waktu

5. KESIMPULAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut:

1) Rata-rata mahasiswa diprediksi lulus tepat waktu jika Indeks Prestasi semester terpenuhi sampai 6 semester dan berdasarkan pengujian data *testing* didapatkan sebanyak 15 orang diprediksi Tepat Waktu dan 6 orang diprediksi Terlambat.
2) Tingkat akurasi pengujian model kelulusan mahasiswa dengan menggunakan algoritma *k-Nearest Neighbor* (*kNN*) dipengaruhi oleh missing value indeks prestasi semester.
3) *Missing value* dengan mean/rata-rata hasil akurasi tertinggi adalah 90% dengan *k*=3. Sedangkan jika *missing value* diganti dengan angka 0 maka hasil akurasi tertinggi adalah 95% dengan *k*=3. Jadi *k* terbaik adalah *k*=3 dengan nilai akurasi tertinggi untuk pengujian model ini.

5.2 Saran

Dari hasil pengujian yang telah dilakukan dan hasil kesimpulan yang didapatkan, terdapat beberapa saran untuk penelitian lebih lanjut, antara lain:

1) Dengan menerapkan kedalam suatu aplikasi yang dapat langsung digunakan untuk keperluan perguruan tinggi untuk memprediksi kelulusan mahasiswa.
2) Diharapkan untuk penelitian lainnya agar lebih menggunakan lebih banyak variabel

agar hasilnya menjadi semakin lebih baik dan dapat diuji coba dengan menggunakan metode lainnya.

DAFTAR PUSTAKA

[1] A. Rohman, "Model Algoritma K-Nearest Neighbor (K-Nn) Untuk Prediksi Kelulusan Mahasiswa," Neo Tek., vol. 1, no. 1, 2015, doi: 10.37760/neoteknika.v1i1.350.

[2] I. A. Nikmatun and I. Waspada, "Implementasi Data Mining untuk Klasifikasi Masa Studi Mahasiswa Menggunakan Algoritma K-Nearest Neighbor," J. SIMETRIS, vol. 10, no. 2, pp. 421–432, 2019.

[3] S. Agarwal, Data mining: Data mining concepts and techniques. 2014.

[4] F. Gorunescu, Data Mining: Concepts, Models and Techniques. Springer Science & Business Media, 2011.

[5] Mustakim and G. Oktaviani F, "Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa," vol. 13, no. 2, pp. 195–202, 2016.

[6] E. Turban, DECISION support systems and intelligent systems (sistem pendukung keputusan dan sistem cerdas). Yogyakarta: Andi, 2005.

[7] A. Y. Saputra and Y. Primadasa, "Penerapan Teknik Klasifikasi Untuk Prediksi Kelulusan Mahasiswa Menggunakan Algoritma K-Nearest Neighbor," Techno.Com, vol. 17, no. 4, pp. 395–403, 2018, doi: 10.33633/tc.v17i4.1864.

[8] J. H. & M. Kamber, Data Mining Concepts and Techniques Second Edition. United States of America: British Library Cataloguing-in-Publication Data, 2006.

[9] W. Andhika, "Belajar machine-learning, basic of scikit-learn," 2019. https://medium.com/@wahyuandhika/belajar-machine-learning-basic-of-scikit-learn-a1685db819a8.