eISSN : 2621-4970
pISSN : 2621-4962

Volume 6 | No 2
Juli | **2023**

Model Prediksi Keketatan Lolos SNMPTN Menggunakan Algoritma K-Nearest Neighbor

Ikriya Hanum[1], Yus Sholva[2], Helen Sastypratiwi[3], Fauzan Asrin[4]
Fakultas Teknik, Program Studi Informatika
Universitas Tanjungpura
ikriya.hanum@student.untan.ad.id

Abstrak

Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) adalah pola penerimaan melalui penelusuran kemampuan dan prestasi akademik sebagai sistem seleksi nasional. Siswa dapat membandingkan prestasi dengan peserta pada tahun-tahun sebelumnya yang telah lolos SNMPTN menggunakan data rapor yang digunakan pada SNMPTN dan data keketatan jurusan perguruan tinggi yang terdapat pada *website* LTMPT. Proses ini dapat dibantu dengan menggunakan algoritma *data mining*, yaitu *K-Nearest Neighbor* (*K-NN*) untuk prediksi. Tujuan pada penelitian ini adalah untuk memprediksi keketatan jurusan pada SNMPTN serta menganalisis performa dari algoritma yang digunakan dalam proses prediksi. Pada penelitian ini, algoritma *K-Nearest Neighbor* akan diujikan untuk melihat parameter terbaik pada algoritma dengan perhitungan nilai *root mean square error* (*RMSE*) pada parameter yang diujikan. Hasil evaluasi menggunakan *Leave One Out cross-validation* menunjukkan bahwa algoritma *K-Nearest Neighbor* yang memberikan hasil performa untuk prediksi paling baik dengan nilai k (jumlah data terdekat) =11 pada jurusan IPA dan nilai k (jumlah data terdekat) =13 pada jurusan IPS. Setelah didapatkan performa terbaik pada algoritma *K-Nearest Neighbor*, maka model algoritma tersebut digunakan untuk memprediksi keketatan jurusan pada SNMPTN melalui aplikasi yang dibangun pada penelitian.
Kata kunci: prediksi, k-nearest neighbor, SNMPTN, data rapor, keketatan jurusan

Abstract

SNMPTN is a national standardized university admission process that uses academic achievement and performance as requirements. Students can compare their achievements with those of participants in previous years who have passed using report cards and the acceptance rate information from the LTMPT website. This process can be helped with the predictive data mining algorithm K-Nearest Neighbor (K-NN). The purpose of this study is to predict the acceptance rate for SNMPTN college majors and analyze the performance of the algorithms used in the prediction process. In this study, the K-Nearest Neighbor algorithm will be tested to determine the best parameters in the algorithm by calculating the root mean square error (RMSE) value for the parameters tested. The results of the evaluation using the Leave One Out cross-validation show that the K-Nearest Neighbor algorithm gives the best performance results for the best predictions with a value of k (number of closest data) =11 in the Sciences department and a value of k (closest amount of data) =13 in the Social Sciences department. After obtaining the best performance model on the K-Nearest Neighbor algorithm, it is used to predict the acceptance rate for SNMPTN college majors through an application built on research.
Keywords: ***prediction, k-nearest neighbor, SNMPTN, report cards, acceptance rate***

PENDAHULUAN

Perguruan tinggi merupakan tahap akhir opsional pendidikan formal sebagai kelanjutan dari pendidikan menengah. Untuk melanjutkan pendidikan ke perguruan tinggi negeri (PTN), siswa dapat mendaftar melalui tiga jalur yaitu SNMPTN, SBMPTN dan Mandiri. Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) merupakan pola penerimaan melalui penelusuran kemampuan dan prestasi akademik sebagai sistem seleksi nasional.

Permasalahan yang sering dialami oleh siswa menjelang SNMPTN adalah siswa belum menemukan minat mereka sehingga siswa sering kali mengalami kesulitan dalam menentukan pilihan jurusan yang ingin diambil pada perguruan tinggi. Persaingan yang ketat juga dapat menyebabkan siswa menjadi pesimis akan pilihan yang diinginkannya. Permasalahan tersebut dapat membuat siswa menjadi ragu dan cenderung asal memilih jurusan yang tidak sesuai minat dan kemampuan mereka. Sedangkan sesuai dengan aturan yang berlaku, bagi siswa pendaftar yang diterima melalui jalur SNMPTN tidak dapat lagi diperkenankan untuk mendaftar pada SBMPTN.

Seleksi siswa dilakukan oleh masing-masing PTN. Secara umum tidak begitu diketahui faktor-faktor lain yang dapat menentukan kelulusan dalam SNMPTN. Namun, siswa dapat membandingkan prestasi dengan peserta pada tahun-tahun sebelumnya yang telah lolos SNMPTN menggunakan data rapor dari semester 1 hingga semester 5 semua mata pelajaran yang digunakan pada SNMPTN dan data keketatan jurusan perguruan tinggi yang terdapat pada *website* LTMPT. Proses ini dapat dibantu dengan menggunakan algoritma *data mining*, yaitu *K-Nearest Neighbor* (*K-NN*). *K-Nearest Neighbor* adalah metode pengklasifikasian dan prediksi yang menghitung kedekatan kasus antara kasus baru dengan kasus lama. Hasil yang diperoleh berdasarkan objek dengan kemiripan terdekat yang dihitung berdasarkan jarak terdekat. Oleh karena itu diperlukan suatu model prediksi dengan mengimplementasikan algoritma *K-Nearest Neighbor* (*K-NN*) untuk memprediksi keketatan persaingan lolos SNMPTN sehingga dapat digunakan untuk membantu proses bimbingan konseling siswa dalam memilih perguruan tinggi dan jurusan yang diperkirakan sesuai dengan prestasi mereka.

Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana mengimplementasikan hasil pemodelan prediksi dengan algoritma *K-Nearest Neighbor* (*K-NN*) untuk memprediksi keketatan persaingan lolos SNMPTN.

Tujuan Penelitian

Tujuan pada penelitian ini adalah dapat menerapkan model prediksi dengan algoritma *K-Nearest Neighbor* (*K-NN*) untuk memprediksi keketatan persaingan lolos SNMPTN.

Manfaat Penelitian

Manfaat pada penelitian ini adalah hasil dari prediksi dapat digunakan untuk membantu dalam proses bimbingan konseling siswa dalam memilih perguruan tinggi dan jurusan yang diperkirakan sesuai dengan prestasi mereka.

Tinjauan Pustaka

Terdapat beberapa penelitian sebelumnya yang berhubungan dengan penelitian ini seperti Wibowo & Fitrianah

[1] di mana pada penelitiannya melakukan prediksi kelulusan SNMPTN menggunakan algoritma *K-Nearest Neighbor* (*K-NN*) berdasarkan nilai rata-rata rapor semester 1 hingga semester 5. Penelitian menghasilkan nilai k=3 memiliki akurasi terbaik yaitu sebesar 80% pada jurusan IPA dan 89% pada jurusan IPS. Dewi & Nursikuwagus [2], melakukan penelitian serupa yaitu membangun sistem prediksi kelulusan siswa SMK pada SNMPTN berdasarkan nilai rata-rata siswa persemester dengan menggunakan metode *Fuzzy* Mamdani yang menunjukan tingkat akurasi (*accuracy*) sebesar 82%, *precision* 79,55%, dan *recall* 100%. Penelitian lainnya juga pernah dilakukan oleh Utomo dkk [3] yang melakukan prediksi penerimaan pada SNMPTN menggunakan algoritma *Decision Tree* C4.5 yang memanfaatkan WEKA CLI untuk proses prediksi. Data-data yang digunakan yaitu nilai mata pelajaran yang digunakan pada SNMPTN 2019 dari semester 1 sampai semester 5 beserta status lulus atau tidak siswa dalam mengikuti SNMPTN.

LANDASAN TEORI

1. SNMPTN

SNMPTN atau Seleksi Nasional Masuk Perguruan Tinggi merupakan jalur penerimaan mahasiswa baru program sarjana pada PTN yang berdasarkan nilai akademik saja atau nilai akademik dan prestasi lainnya (yang ditetapkan oleh PTN). Siswa yang dapat mendaftar pada SNMPTN merupakan siswa *eligible* yang masuk dalam pemeringkatan dengan jumlah sesuai dengan ketentuan kuota akreditasi sekolah.

Berdasarkan informasi resmi dari LTMPT [4], pemeringkatan siswa dilakukan oleh sekolah yang pada dasarnya memperhitungkan nilai mata pelajaran untuk jurusan IPA yaitu: Matematika, Bahasa Indonesia, Bahasa Inggris, Kimia, Fisika, dan Biologi. Sedangkan untuk jurusan IPS yaitu: Matematika, Bahasa Indonesia, Bahasa Inggris, Sosiologi, Ekonomi, dan Geografi. Sekolah dapat menambahkan kriteria lain berupa prestasi akademik dalam menentukan peringkat siswa apabila ada nilai yang sama.

2. *DataMining*

Data Mining merupakan proses penemuan pola dan pengetahuan yang didapatkan dari data berukuran besar. *Data Mining* bertujuan untuk memanfaatkan data dalam basis data dengan mengolahnya sehingga menghasilkan informasi baru yang berguna [5].

Metode pada pelatihan *data mining* dapat dibedakan menjadi *supervised learning* dan *unsupervised learning*. Sebagian besar metode *data mining* merupakan *supervised learning*; yaitu metode dimana algoritma diberikan data latih yang banyak dengan variabel target (label) yang telah ditentukan sebelumnya sebagai *training* sehingga kemudian algoritma dapat menentukan variabel target untuk dikaitkan pada variabel yang belum diketahui labelnya. Sementara itu, *unsupervised learning* merupakan metode yang mencari pola dan struktur di antara seluruh variabel tanpa diketahui variabel target yang diidentifikasikan sebelumnya. Dengan kata lain metode ini diterapkan tanpa adanya latihan dan tanpa ada pelatih yaitu label dari data [6].

Contoh klasik dari teknik *supervised learning* diwakili oleh proses klasifikasi (metode prediktif) yang dilakukan dengan menggunakan sebagian variabel untuk memprediksi satu atau lebih variabel lain; sedangkan contoh klasik dari teknik

unsupervised learning diwakili oleh proses klastering (metode deskriptif) yang dilakukan dengan identifikasi pola yang menggambarkan atau mewakilkan data agar dapat dengan mudah dipahami oleh pengguna [7].

3. Algoritma *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor* (*K-NN*) merupakan algoritma yang termasuk ke dalam *supervised learning* dan juga merupakan contoh dari *instance-based learning*. Pada *instance-based learning*, data yang dijadikan pembelajaran yaitu yang sudah diketahui kelasnya disimpan dan secara langsung dikaitkan/dibandingkan pada data yang belum diketahui kelasnya. Pekerjaan dilakukan saat data baru diberikan, tanpa membuat kesimpulan terlebih dahulu melalui data pembelajaran [8].

Algoritma *K-Nearest Neighbor* melakukan prediksi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Pada algoritma *K-NN* diperlukan sebelumnya untuk menentukan nilai k, yaitu jumlah objek tetangga (*neighbor*) yang terdekat. Hasil akhir yaitu label kelas dengan jumlah terbanyak diantara k objek. *K-NN* juga dapat digunakan pada prediksi numerik, yang memberikan prediksi berbentuk nilai untuk kelas dengan label tidak diketahui. Dalam hal ini, model memberikan hasil berdasarkan rata-rata dari label nilai sejumlah k objek [9].

Untuk menentukan jarak antara data *training* dan data *testing* maka dilakukan perhitungan dengan rumus *Euclidean distance*. Rumus yang digunakan untuk mengukur nilai *Euclidean distance* dapat ditunjukkan pada persamaan berikut [10]:

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

dimana:

d = Jarak kedekatan antara x dan y

x = Data *Training*

y = Data *Testing*

i = *Record* (baris) ke-i data

n = Jumlah data training

4. *Cross Validation*

Saat mengevaluasi *hyperparameter* pada *estimator*, masih ada risiko *overfitting* pada dataset *testing* karena parameter dapat diatur hingga *estimator* bekerja secara optimal. Untuk mengatasi masalah ini, beberapa bagian dari dataset dapat dibagi menjadi "dataset validasi" untuk proses *training* yang dilakukan pada dataset *training*. Setelah evaluasi dilakukan pada dataset validasi, kemudian evaluasi terakhir dapat dilakukan pada dataset *test*.

Namun, dengan membagi data menjadi tiga bagian dapat mengurangi jumlah sampel yang dapat digunakan sebagai pembelajaran pada model dan hasil yang didapatkan dapat bergantung pada set data acak yang terpilih. Solusi dari permasalahan ini dapat dilakukan metode *cross-validation* (*CV*). Dataset *test* diperlukan untuk evaluasi terakhir, namun dataset validasi tidak lagi diperlukan pada *CV* [11].

5. *Root Mean Square Error*

Untuk melakukan evaluasi terhadap suatu peramalan atau prediksi dapat dilakukan dengan beberapa cara pengukuran yang salah satunya yaitu dengan menggunakan *Root Mean Square Error* (*RMSE*). *RMSE* adalah aturan penilaian kuadrat yang mengukur besarnya rata-rata kesalahan [12]. *RMSE* merupakan besarnya tingkat kesalahan hasil prediksi, dimana semakin kecil (mendekati 0) nilai *RMSE* maka hasil prediksi akan semakin akurat [13]. *RMSE* digunakan untuk

membandingkan nilai yang diprediksi oleh model hipotetis dengan nilai dari hasil pengamatan. Dengan kata lain, *RMSE* mengukur kualitas kesesuaian antara data aktual dan model prediksi [14].

Berikut persamaan *root means square error* (*RMSE*) [15]:

$$RMSE = \sqrt{\frac{\sum(prediksi - aktual)^2}{jumlah\ data}} \quad (2)$$

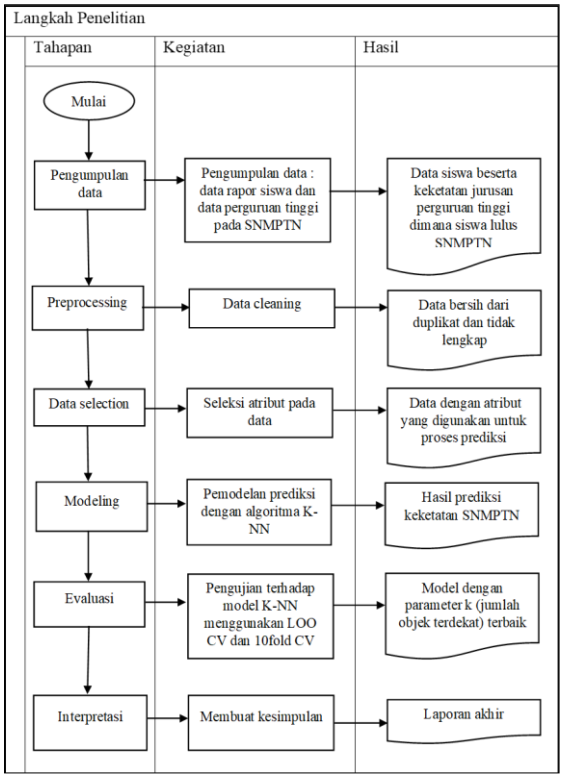
METODE PENELITIAN

Langkah awal penelitian yaitu dengan melakukan pengumpulan data. Data diperoleh dari nilai rapor alumni siswa SMA Negeri 1 Pontianak yang lolos SNMPTN pada tahun 2019 sampai dengan tahun 2021, dan data jurusan perguruan tinggi yang terdaftar pada SNMPTN melalui situs *website* resmi LTMPT. Data-data yang diperoleh kemudian digabungkan, disusun dan dimasukan pada *excel*. Maka didapatkanlah data-data siswa beserta jurusan PTN dimana siswa lolos SNMPTN. *File excel* tersebut kemudian diubah menjadi *file* dengan format *.csv*. Langkah selanjutnya adalah melakukan *preprocessing data* dimana data yang sudah dikumpulkan di periksa kembali untuk memastikan tidak terdapat data ganda atau tidak lengkap. Kemudian dilakukan proses *data selection* yaitu proses untuk memilah atribut-atribut pada data yang diperlukan pada proses prediksi.

Data yang telah siap kemudian dapat digunakan untuk proses *data mining* yaitu prediksi. Pemodelan prediksi dilakukan dengan menggunakan algoritma *K-NN* dan menghasilkan prediksi keketatan jurusan pada SNMPTN. Untuk mengetahui model terbaik yang dibangun, maka dapat dilakukan proses selanjutnya yaitu evaluasi. Proses evaluasi algoritma yang digunakan yaitu dengan melakukan *10 Fold* dan *Leave One Out Cross Validation*. Langkah terakhir adalah menarik kesimpulan berdasarkan hasil yang didapat pada penelitian.

Alur/Langkah Penelitian

Terdapat enam langkah penelitian ini yang dapat dilihat pada Gambar. 1.



Gambar. 1 Langkah penelitian

ANALISIS DAN PERANCANGAN

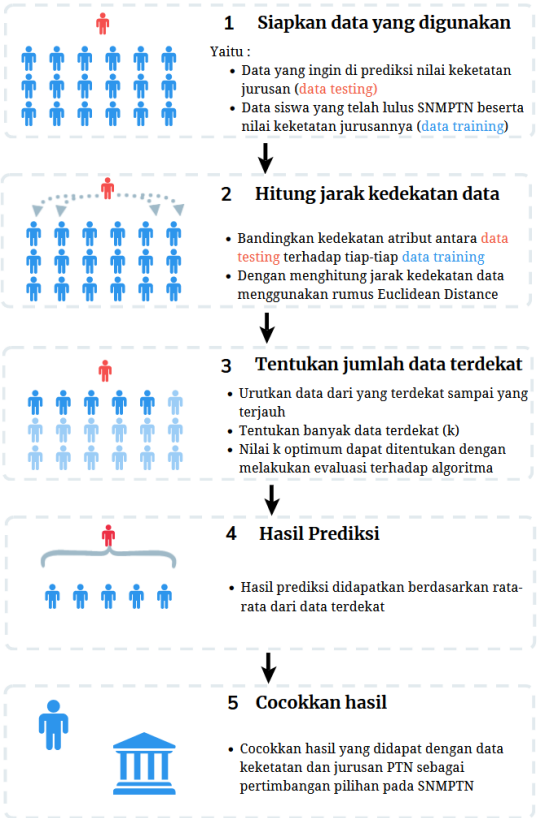
1. Pemodelan

SNMPTN adalah seleksi masuk perguruan tinggi yang menggunakan nilai rapor dan prestasi lainnya yang ditentukan oleh masing-masing PTN. Strategi yang dapat diterapkan oleh siswa dalam memilih jurusan pada SNMPTN yaitu dengan membandingkan tingkat keketatan dengan prestasi yang dimilikinya. Hal tersebut dapat dilakukan dengan membandingkan prestasi dengan peserta pada tahun-tahun sebelumnya yang telah lolos SNMPTN, dengan menggunakan data rapor dan data keketatan jurusan perguruan tinggi yang terdapat pada

website LTMPT. Proses ini dapat dibantu dengan menggunakan algoritma data mining *K-Nearest Neighbor* (*K-NN*).

Algoritma *K-NN* membandingkan kemiripan tiap-tiap data pada data baru yang ingin diprediksi. Perbandingan dilakukan pada tiap-tiap atribut yang dimiliki oleh data. Data yang digunakan adalah data yang berasal dari satu sekolah untuk meminimalkan pengaruh atribut sekolah pada seleksi SNMPTN. Atribut yang digunakan untuk melihat kemiripan data yaitu nilai-nilai yang digunakan untuk perangkingan siswa *eligible*.

Algoritma *K-Nearest Neighbor* melakukan prediksi berdasarkan hasil dari data-data terdekat. Hasil prediksi keketatan SNMPTN kemudian dapat dicocokkan pada data keketatan jurusan PTN, sebagai pertimbangan pilihan jurusan yang dapat diambil pada SNMPTN. Gambaran mengenai penerapan algoritma *K-NN* ditampilkan pada Gambar. 2.



Gambar. 2 Penerapan *K-NN*

Pada proses perhitungan *K-NN*, yang dapat dilakukan setelah menyiapkan data yang digunakan adalah data dibedakan menjadi data *training* berupa a yang sudah diketahui labelnya dan data *testing* berupa b yang akan diprediksi labelnya. Tiap-tiap atribut pada data *testing* akan dibandingkan kedekatanya terhadap data *training*. Untuk melihat kedekatan data dilakukan dengan perhitungan jarak (d) menggunakan rumus *Euclidean distance* seperti ditampilkan sebagai berikut:

$$d(a,b) = \sqrt{\frac{(a_1-b_1)^2+(a_2-b_2)^2+\cdots+(a_{38}-b_{38})^2+(a_{39}-b_{39})^2+}{(a_{40}-b_{40})^2+(a_{41}-b_{41})^2+(a_{42}-b_{42})^2+(a_{43}-b_{43})^2+(a_{44}-b_{44})^2}}$$

$$d(1,4) = \sqrt{\frac{(2021-2022)^2+(32-33)^2+\cdots+(89.6-80)^2+(87.4-80)^2+}{(88.6-80)^2+(89.6-80)^2+(90.8-80)^2+(88.4-80)^2+(89.07-80)^2}}$$

$$d(1,4) = \sqrt{(-1)^2+(-1)^2+\cdots+(9.6)^2+(7.4)^2+(8.6)^2+(9.6)^2+(10.8)^2+(8.4)^2+(9.07)^2}$$

$$d(1,4) = \sqrt{1+1+\cdots+92.16+54.76+73.96+92.16+116.64+70.56+82.2649}$$

$$d(1,4) = \sqrt{3567.7087}$$

$$d(1,4) = 59.7303$$

Tiap-tiap atribut pada data *testing* dibandingkan dengan data *training* dengan menghitung akar jumlah dari selisih kuadrat tiap atribut, hingga didapatkanlah jarak kedekatannya. Proses ini dilakukan sampai jarak antara data *testing* terhadap seluruh data *training* didapatkan. Data-data tersebut kemudian diurutkan dari jarak yang terdekat seperti yang ditampilkan pada Tabel 1 berikut:

Tabel. 1 Jarak

Id	Jarak (d)	Label (Keketatan)
6	58.1913	5.83
15	58.4457	6.39
12	59.4105	7.12
4	59.7304	7.89
14	59.8501	5.69
27	59.9802	7.96
26	61.3876	11.24
23	61.8088	7.34
…	…	…

Proses selanjutnya adalah menentukan nilai k, yaitu jumlah data *training* yang terdekat dengan data *testing* untuk perhitungan prediksi. Nilai k optimum didapatkan dengan melakukan evaluasi terhadap algoritma. Nilai k digunakan untuk mendapatkan hasil prediksi, yang dilakukan dengan menghitung rata-rata label data terdekat sejumlah k objek. Apabila nilai k yang digunakan adalah 5, maka perhitungan prediksi didapatkan sebagai berikut:

$$Prediksi = \frac{5.83 + 6.39 + 7.12 + 7.89 + 5.69}{5}$$

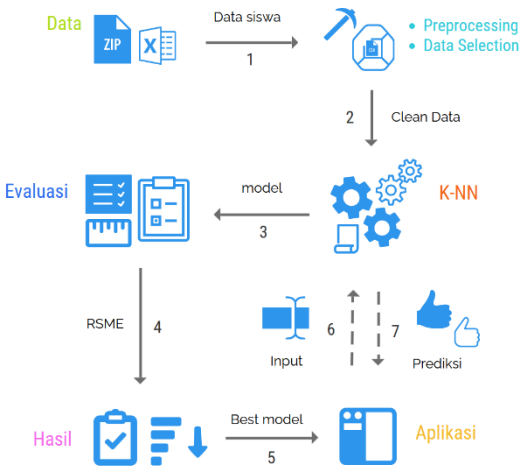
$$\text{Prediksi} = \frac{32.92}{5}$$

$$\text{Prediksi} = 6.58$$

Dengan demikian, hasil prediksi keketatan jurusan yang didapatkan adalah 6.58. Hasil yang didapatkan tersebut dapat dicocokkan pada data keketatan jurusan PTN sebagai pertimbangan jurusan yang dapat dipilih pada SNMPTN.

2. *Workflow* Aplikasi

Workflow aplikasi yang dibuat dalam penelitian ini ditunjukan pada Gambar. 3. Pertama, data berupa data siswa yang lolos pada SNMPTN dilakukan *preprocessing* dengan *cleaning* untuk menghilangkan data ganda dan tidak lengkap, serta dilakukan proses data *selection* untuk menyeleksi atribut data hingga akhirnya diperoleh data yang siap digunakan untuk proses *data mining*. Selanjutnya dilakukan *modeling* terhadap dataset tersebut. Pada model dilakukan evaluasi dengan *cross-validation*. Pengukuran yang digunakan pada evaluasi adalah nilai *root mean square error* (*RMSE*). *RMSE* sebagai acuan untuk menemukan model dengan performa terbaik. Model dengan performa terbaik digunakan sebagai model prediksi pada aplikasi yang dibangun. Aplikasi tersebut dirancang agar pengguna dapat melakukan prediksi dengan memasukkan input baru, dan kemudian akan memberikan hasil prediksi sesuai data *training* yang diberikan.



Gambar. 3 *Workflow* aplikasi

IMPLEMENTASI DAN PEMBAHASAN

1. Pengumpulan Data

Data yang dikumpulkan untuk digunakan dalam penelitian ini merupakan data yang diperoleh dari nilai rapor alumni siswa SMA Negeri 1 Pontianak yang lolos SNMPTN pada tahun 2019 sampai dengan tahun 2021, dan data jurusan perguruan tinggi yang terdaftar pada SNMPTN melalui situs *website* resmi LTMPT. Data-data yang dikumpulkan tersebut diantaranya yaitu: nama siswa, tahun kelulusan, rangking siswa *eligible*, jurusan, nilai semua mata pelajaran yang digunakan untuk perangkingan dari semester 1 sampai dengan semester 5, rata-rata nilai pada tiap semester, rata-rata nilai pada tiap mata pelajaran, rata-rata keseluruhan, serta jurusan dan perguruan tinggi dimana siswa diterima pada SNMPTN.

Data yang terkumpul yaitu sebanyak 171 data siswa yang dapat dibedakan berdasarkan jurusan masing-masing yaitu jurusan IPA sebanyak 107 data dan jurusan IPS sebanyak 64 data. Jumlah data

untuk setiap jurusan dapat dilihat pada Tabel 2.

Tabel. 2 Jumlah Dataset

Jurusan	Jumlah Data
IPA	107
IPS	64
Total	171

2. *Preprocessing*

Tahapan *preprocessing* untuk mempersiapkan data yang dilakukan yaitu *data cleaning*. Proses *data cleaning* dilakukan untuk mengecek data yang hilang dan membersihkan data yang terkumpul agar terhindar dari adanya data duplikat. Proses ini dapat dilihat pada Gambar. 4. Hasil dari *preprocessing* yaitu data bersih dapat dilihat pada Gambar. 5.

```
df = df.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
df = df.drop_duplicates()
```

Gambar. 4 *Preprocessing*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 171 entries, 0 to 170
Data columns (total 49 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   NAMA      171 non-null    object
 1   JURUSAN   171 non-null    object
 2   TAHUN     171 non-null    int64
 3   RANK      171 non-null    int64
 4   1_BIND    171 non-null    int64
 5   1_MAT     171 non-null    int64
 6   1_BING    171 non-null    int64
 7   1_MP1     171 non-null    int64
 8   1_MP2     171 non-null    int64
 9   1_MP3     171 non-null    int64
 10  1_AVG     171 non-null    float64
 11  2_BIND    171 non-null    int64
 12  2_MAT     171 non-null    int64
 13  2_BING    171 non-null    int64
 14  2_MP1     171 non-null    int64
 15  2_MP2     171 non-null    int64
 16  2_MP3     171 non-null    int64
 17  2_AVG     171 non-null    float64
```

Gambar. 5 Hasil *preprocessing*

3. *Data Selection*

Data Selection untuk memilah atribut-atribut pada data yang diperlukan untuk proses *data mining* pada penelitian sebagaimana diperlihatkan pada Gambar. 6.

```
df = df.drop(columns=['NAMA','JURUSAN','Prodi','PTN'])
```

Gambar. 6 Data *selection*

Data kemudian di inisialisasi untuk dibedakan antara data dengan *role* label yaitu keketatan jurusan dengan atribut lainnya. Seperti tampak pada Gambar. 7.

```
X = df.iloc[:, :-1].values #non-Label
Y = df.iloc[:, 44].values #Label
```

Gambar. 7 Inisialisasi

Dengan demikian *role* pada data yang digunakan untuk proses data *mining* dapat ditunjukan pada Tabel 3.

Tabel. 3 Data

Role	Data	
	IPA	IPS
Atribut	Tahun	Tahun
	Rangking	Rangking
	Nilai Bahasa Indonesia semester 1	Nilai Bahasa Indonesia semester 1
	Nilai Matematika semester 1	Nilai Matematika semester 1
	Nilai Bahasa Inggris semester 1	Nilai Bahasa Inggris semester 1
	Nilai Biologi semester 1	Nilai Geografi semester 1
	Nilai Fisika semester 1	Nilai Sosiologi semester 1
	Nilai Kimia semester 1	Nilai Ekonomi semester 1
	Nilai rata-rata semester 1	Nilai rata-rata semester 1
	Nilai Bahasa Indonesia semester 2	Nilai Bahasa Indonesia semester 2
	Nilai Matematika semester 2	Nilai Matematika semester 2
	Nilai Bahasa Inggris semester 2	Nilai Bahasa Inggris semester 2
	Nilai Biologi semester 2	Nilai Geografi semester 2
	Nilai Fisika semester 2	Nilai Sosiologi semester 2
	Nilai Kimia semester 2	Nilai Ekonomi semester 2
	Nilai rata-rata semester 2	Nilai rata-rata semester 2

	Nilai Bahasa Indonesia semester 3	Nilai Bahasa Indonesia semester 3
	Nilai Matematika semester 3	Nilai Matematika semester 3
	Nilai Bahasa Inggris semester 3	Nilai Bahasa Inggris semester 3
	Nilai Biologi semester 3	Nilai Geografi semester 3
	Nilai Fisika semester 3	Nilai Sosiologi semester 3
	Nilai Kimia semester 3	Nilai Ekonomi semester 3
	Nilai rata-rata semester 3	Nilai rata-rata semester 3
	Nilai Bahasa Indonesia semester 4	Nilai Bahasa Indonesia semester 4
	Nilai Matematika semester 4	Nilai Matematika semester 4
	Nilai Bahasa Inggris semester 4	Nilai Bahasa Inggris semester 4
	Nilai Biologi semester 4	Nilai Geografi semester 4
	Nilai Fisika semester 4	Nilai Sosiologi semester 4
	Nilai Kimia semester 4	Nilai Ekonomi semester 4
	Nilai rata-rata semester 4	Nilai rata-rata semester 4
	Nilai Bahasa Indonesia semester 5	Nilai Bahasa Indonesia semester 5
	Nilai Matematika semester 5	Nilai Matematika semester 5
	Nilai Bahasa Inggris semester 5	Nilai Bahasa Inggris semester 5
	Nilai Biologi semester 5	Nilai Geografi semester 5
	Nilai Fisika semester 5	Nilai Sosiologi semester 5
	Nilai Kimia semester 5	Nilai Ekonomi semester 5
	Nilai rata-rata semester 5	Nilai rata-rata semester 5
	Nilai rata-rata Bahasa Indonesia	Nilai rata-rata Bahasa Indonesia
	Nilai rata-rata Matematika	Nilai rata-rata Matematika
	Nilai rata-rata Bahasa Inggris	Nilai rata-rata Bahasa Inggris
	Nilai rata-rata Biologi	Nilai rata-rata Geografi
	Nilai rata-rata Fisika	Nilai rata-rata Sosiologi
	Nilai rata-rata Kimia	Nilai rata-rata Ekonomi
	Nilai rata-rata total	Nilai rata-rata total
Label	Keketatan	Keketatan

4. *Modeling*

Data untuk prediksi dibedakan sesuai dengan jurusan yaitu IPA dan IPS. Dengan demikian prediksi dengan jurusan IPA menggunakan data jurusan IPA, sedangkan prediksi dengan jurusan IPS menggunakan data jurusan IPS. Dengan menggunakan *library* sklearn, *KNeighborsRegressor* digunakan sebagai model untuk prediksi sebagaimana diperlihatkan pada Gambar. 8. Secara *default* parameter yang digunakan yaitu nilai k=5 dan rumus yang digunakan untuk perhitungan jarak yaitu menggunakan *Euclidean distance* dengan nilai p=2.

```
model = KNeighborsRegressor(n_neighbors=k, p=2)
```

Gambar. 8 *Modeling*

5. Evaluasi

Proses evaluasi algoritma pada model yang dibuat yaitu dengan melakukan *10Fold Cross Validation* seperti Gambar. 9 dan *Leave One Out Cross Validation* seperti pada Gambar. 10. *CV* dibangun dalam bahasa pemrograman Python dengan memanfaatkan *library* seperti *Scikit-Learn* dan lainnya. Hasil dari evaluasi dengan *CV* dapat dilihat pada Tabel 4 dan Tabel 5.

```
score= cross_val_score(model, X, Y, cv=10, scoring='neg_root_mean_squared_error')
print(score)
print(score.mean())
```

Gambar. 9 10*fold CV*

```
score= cross_val_score(model, X, y, cv=loo, scoring='neg_root_mean_squared_error')
print(score)
print(score.mean())
```

Gambar. 10 *LOO CV*

Tabel. 4 Evaluasi 10*fold CV*

Nilai k	**Nilai *RMSE K-NN* 10*fold CV***	
	IPA	**IPS**
k=1	13.9321	6.3284
k=2	11.1756	5.1129
k=3	10.1331	5.2331
k=4	10.1606	5.0038
k=5	9.9893	5.1380
k=6	9.6639	5.1608
k=7	9.4923	5.0300
k=8	9.5704	4.8543
k=9	9.2767	4.7774
k=10	9.2602	4.7320
k=11	9.3260	4.8348
k=12	9.2937	4.8654

k=13	9.2506	4.8221
k=14	9.3124	4.8165
k=15	9.2786	4.8543
k=16	9.3514	4.8354
k=17	9.3466	4.8247
k=18	9.3647	4.8576
k=19	9.3190	4.8890
k=20	9.3664	4.9359

Tabel. 5 Evaluasi *LOO CV*

Nilai k	**Nilai *RMSE K-NN LOO CV***	
	IPA	**IPS**
k=1	7.8345	4.3875
k=2	6.6747	3.8502
k=3	6.4041	3.9248
k=4	6.3044	3.8043
k=5	6.0881	3.7264
k=6	5.9827	3.6574
k=7	5.8509	3.8070
k=8	5.9082	3.5754
k=9	5.8189	3.7042
k=10	5.8197	3.6804
k=11	5.7296	3.5625
k=12	5.9241	3.5398
k=13	6.0079	3.5046
k=14	6.0061	3.5071
k=15	6.0858	3.5172
k=16	5.9682	3.6214
k=17	6.0056	3.6327
k=18	5.9626	3.6507
k=19	6.0044	3.6426
k=20	6.0747	3.6250

Cara lain untuk menentukan parameter terbaik adalah menggunakan *grid search* seperti pada Gambar. 11. Hasil dari *grid search* dengan menggunakan 10-*fold CV* pada data siswa jurusan IPS ditampilkan pada Gambar. 12.

```
n_neighbors = list(range(1,21))
#Ubah ke dictionary
hyperparameters = dict(n_neighbors=n_neighbors)
#KNN
knn_2 = KNeighborsRegressor()
#GridSearch
clf = GridSearchCV(knn_2, hyperparameters, cv=10, scoring='neg_root_mean_squared_error',
#Fit model
best_model = clf.fit(X,y)
#Print nilai Hyperparameters
print('Best n_neighbors:', best_model.best_estimator_.get_params()['n_neighbors'])
```

Gambar. 11 *Grid search*

```
Best n_neighbors: 10
```

Gambar. 12 Hasil *grid search*

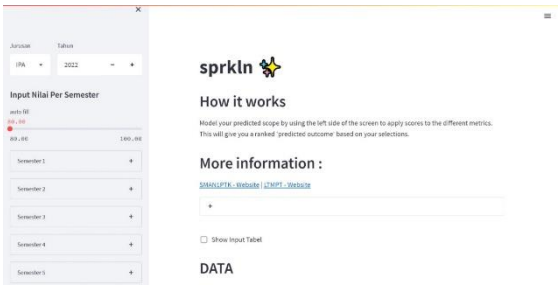
Pemilihan nilai k ditentukan dengan tingkat *error* yang rendah dalam melakukan prediksi. Model dengan nilai k terbaik dari proses pengujian merupakan hasil rata-rata dari *cross-validation* yang memberikan nilai paling mendekati 0. Nilai k tersebut merupakan jumlah objek terdekat dengan nilai input yang akan digunakan untuk proses prediksi dengan menggunakan algoritma *K-Nearest Neighbor*. Dengan demikian, berdasarkan hasil pengujian terdapat beberapa poin yang diperoleh antara lain:

1) Hasil evaluasi 10-*fold cross-validation model* menggunakan algoritma *K-Nearest Neighbor*: nilai k terbaik pada data siswa jurusan IPA yang memberikan nilai rata-rata *RMSE* terendah sebesar 9.2506 adalah k=13 dan nilai k terbaik pada data siswa jurusan IPS yang memberikan nilai rata-rata *RMSE* terendah sebesar 4.7320 adalah k=10.

2) Hasil evaluasi *Leave One Out cross-validation model* menggunakan algoritma *K-Nearest Neighbor*: nilai k terbaik pada data siswa jurusan IPA yang memberikan nilai rata-rata *RMSE* terendah sebesar 5.7296 adalah k=11 dan nilai k terbaik pada data siswa jurusan IPS yang memberikan nilai rata-rata *RMSE* terendah sebesar 3.5046 adalah k=13.

6. Aplikasi

Hasil dari pemodelan data dibuktikan dengan tangkapan layar (*screenshot*) dari aplikasi yang telah dibuat. Berikut adalah halaman utama yang pertama kali muncul saat membuka aplikasi sebagaimana ditampilkan pada Gambar. 13.



Gambar. 13 Halaman utama aplikasi

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat diberikan kesimpulan sebagai berikut:

1. Pemodelan prediksi menghasilkan suatu aplikasi yang dapat melakukan prediksi keketatan persaingan jurusan pada SNMPTN. Aplikasi dapat melakukan prediksi dengan algoritma *K-NN* yang dilatih menggunakan data yang diunggah sesuai dengan format yang ditentukan. Nilai k optimum dari algoritma *K-NN* merupakan hasil dari *grid search* menggunakan evaluasi *Leave One Out cross-validation* terhadap data yang dilatih. Pada aplikasi, hasil prediksi ditampilkan berdasarkan tabel data sejumlah k optimum atau dapat dipilih, yang diurutkan sesuai jarak kedekatan terdekat dengan nilai input.
2. Dari pengujian nilai k (jumlah objek terdekat) menggunakan *Leave One Out cross-validation* pada algoritma *K-Nearest Neighbor* didapatkan hasil evaluasi nilai k optimum terhadap data siswa pada jurusan IPA adalah k=11.
3. Dari pengujian nilai k (jumlah objek terdekat) menggunakan *Leave One Out cross-validation* pada algoritma *K-Nearest Neighbor* didapatkan hasil evaluasi nilai k optimum terhadap data siswa pada jurusan IPS adalah k=13.

Saran

Hasil dari penelitian menunjukkan bahwa algoritma *K-Nearest Neighbor* dapat digunakan sebagai model untuk prediksi nilai keketatan persaingan jurusan PTN pada SNMPTN. Akan tetapi algoritma *K-NN* sensitif pada data extrim/*outlier*, sehingga perlu dilakukan perbandingan dengan mencoba menggunakan algoritma prediksi lainnya, agar dapat dilakukan analisa hasil dari performa algoritma yang lebih baik lagi untuk prediksi nilai keketatan SNMPTN, dan juga terhadap aplikasi dapat dilakukan pengembangan agar nantinya dapat mendeteksi data *outlier*, serta proses pemilihan atribut untuk prediksi agar dapat dipilih secara dinamis maupun di evaluasi untuk proses prediksi.

DAFTAR PUSTAKA

[1] Wibowo, A. T., & Fitrianah, D. (2018). *A K-Nearest Algorithm Based Application To Predict Snmptn Acceptance For High School Students In Indonesia*. International Research Journal Of Computer Science (IRJCS) Issue 01, 5, 9–19. https://doi.org/10.26562/IRJCS.2018.JACS10083

[2] Dewi, Z. C., & Nursikuwagus, A. (2018). *Analisis Prediksi Kelulusan Siswa SMK pada SNMPTN Menggunakan Metode Fuzzy Mamdani (Studi Kasus : SMK Negeri 4 Bandung) Analysis of SMK Students Passing Prediction on SNMPTN Using Fuzzy Mamdani Method (Case Study: SMK Negeri 4 Bandung)*.

[3] Utomo, D. K., Supianto, A. A., & Purnomo, W. (2019). *Sistem Prediksi Penerimaan SNMPTN menggunakan Algoritme Decision Tree C4.5* (Vol. 3, Issue 9). http://j-ptiik.ub.ac.id

[4] LTMPT. (2021). *Informasi Sistem Seleksi Masuk Perguruan Tinggi Negeri*. http://ltmpt.ac.id

[5] Pradnyana, G., & Agustini, K. (2022). *Konsep Dasar Data Mining*. http://pustaka.ut.ac.id

[6] Larose, D. T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley, New York, 203-231.

[7] Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques*. Springer Science & Business Media.
[8] Witten, I.H., Frank, E., dan Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques* (3rd ed.). Elsevier.
[9] Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann Publishers Inc.
[10] Suntoro, J. (2019). *DATA MINING Algoritme dan Implementasi Menggunakan Bahasa Pemrograman PHP*.
[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel V. and Thirion, B., Grisel, O., Blondel, M., Prettenhofer P. and Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
[12] Sutoyo, E., & Almaarif, A. (2020). *Educational Data Mining for Predicting Student Graduation Using the Naïve Bayes Classifier Algorithm*. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 4(1), 95 - 101. https://doi.org/10.29207/resti.v4i1.1502
[13] Sulaiman, A., & Juarna, A. (2021). *Peramalan Tingkat Pengangguran Di Indonesia Menggunakan Metode Time Series Dengan Model Arima Dan Holt-Winters*. Jurnal Ilmiah Informatika Komputer Volume 26 No.1 April 2021. https://doi.org/10.35760/ik.2021.v26i1.3512
[14] Prasetyo, Vincentius Riandaru and Lazuardi, Hamzah and Mulyono, Aldo Adhi and Lauw, Christian (2021). *Penerapan Aplikasi RapidMiner Untuk Prediksi Nilai Tukar Rupiah Terhadap US Dollar Dengan Metode Regresi Linier*. Jurnal Nasional Teknologi dan Sistem Informasi (TEKNOSI), 7 (1). pp. 8-17. ISSN 2476-8812.
[15] Budiman, H. (2016). *Analisis Dan Perbandingan Akurasi Model Prediksi Rentet Waktu Support Vector Machines Dengan Support Vector Machines Particle Swarm Optimization Untuk Arus Lalu Lintas Jangka Pendek*. Systemic: Information System and Informatics Journal, 2(1), 19–24. https://doi.org/10.29080/systemic.v2i1