

IMPLEMENTATION OF LOCAL MEAN DISTANCE WEIGHTING K-NEAREST NEIGHBOR IN DETERMINING VOCATIONAL HIGH SCHOOL MAJORS IN PEKANBARU

PENERAPAN PEMBOBOTAN LOCAL MEAN DISTANCE K-NEAREST NEIGHBOR DALAM PENENTUAN JURUSAN SMK DI PEKANBARU

Khairul Umam Syaliman^{1*}, Dwi Gunawan²

^{1,2}Teknik Informatika, Politeknik Caltex Riau, Pekanbaru, Indonesia

^{1*}khairul.q14@gmail.com

ABSTRACT

Vocational High School is a formal education at the secondary education level. SMK X Pekanbaru is one of the private vocational schools in Pekanbaru that provides IT-based education. There are 5 (five) majors in SMK X Pekanbaru, namely Computer and Network Engineering (TKJ), Software Engineering (RPL), Accounting and Financial Institution (AKL), Office Automation and Governance (OTKP), and Online Business and Marketing (BDP). During the registration period, prospective SMK students will enter their score data and will also choose the majors they want to take. In the absence of a system that can determine majors for prospective SMK students, obstacles will arise including errors in determining majors and requiring a long time to process prospective student data. Based on the above problems, a system will be built that can speed up and simplify the determination of new student majors by using Supervised Learning algorithms in Machine Learning, namely Local Mean Distance Weight k-Nearest Neighbor (LMDWkNN). Based on the results of the confusion matrix testing carried out, the accuracy results were 88.89%, the precision, recall and F1 score were 89%, which states that the model is good enough to determine majors.

Keywords : *Accuracy, Local Mean Distance Weight k-Nearest Neighbor (LMDWkNN), Major Determination System, Vocational High School*

ABSTRAK

Sekolah Menengah Kejuruan merupakan salah satu pendidikan formal pada jenjang pendidikan menengah. SMK X Pekanbaru merupakan salah satu SMK swasta di Pekanbaru yang menyelenggarakan pendidikan berbasis IT. Terdapat 5 (lima) jurusan di SMK X Pekanbaru yaitu Teknik Komputer dan Jaringan (TKJ), Rekayasa Perangkat Lunak (RPL), Akuntansi dan Lembaga Keuangan (AKL), Otomasi dan Tata Kelola Perkantoran (OTKP), dan Bisnis dan Pemasaran Daring (BDP). Pada masa registrasi calon siswa SMK akan menginput data nilai dan juga akan memilih jurusan yang ingin diambil. Dengan belum adanya sistem yang dapat menentukan jurusan bagi calon siswa SMK maka akan timbul kendala diantaranya kesalahan dalam menentukan jurusan dan membutuhkan waktu yang lama dalam mengolah data calon siswa. Berdasarkan permasalahan diatas maka akan dibangun suatu sistem yang dapat mempercepat dan mempermudah dalam penentuan jurusan siswa baru dengan menggunakan algoritma Supervised Learning pada Machine Learning yaitu Local Mean Distance Weight k-Nearest Neighbor (LMDWkNN). Berdasarkan hasil pengujian matriks konfusi yang dilakukan, diperoleh hasil akurasi sebesar 88,89%, presisi, recall dan skor F1 sebesar 89% yang menyatakan bahwa model cukup baik untuk menentukan jurusan.

Kata Kunci : Akurasi, Local Mean Distance Weight k-Nearest Neighbor (LMDWkNN), Sistem Penentuan Jurusan, SMK

1. Pendahuluan

SMK X Pekanbaru merupakan salah satu SMK Swasta di Pekanbaru yang menyelenggarakan Pendidikan Berbasis IT. Selama masa pendaftaran, calon siswa SMK tidak hanya memberikan data nilai tingkat SMP, mereka juga akan memilih jurusan yang ingin mereka ambil. Penjurusan siswa merupakan suatu proses penempatan siswa ke dalam jurusan atau peminatan tertentu, sehingga siswa dapat menyerap semua mata pelajaran dengan optimal

dan sesuai dari kemampuan yang dimiliki siswa tersebut (Howay & Rianto, 2021; Mafakhir & Solichin, 2020; Rawal Dewa & Jasmir, 2023).

SMK X memiliki beberapa tahapan dalam melakukan pendaftaran calon siswa. Pertama, registrasi dengan cara mengirimkan SMS ke kontak PPDB SMK X untuk mendapatkan akun yang digunakan pada *website* PPDB SMK X. Setelah masuk, calon siswa dapat mengisi *form* pendaftaran yang berisi nilai dan jurusan pilihan calon siswa. Selanjutnya akan tampil bukti pendaftaran yang harus diserahkan ke SMK X. Calon siswa selanjutnya diharapkan untuk menunggu hasil pendaftaran pada *website* PPDB dan setelah dinyatakan lulus, calon siswa dapat melakukan daftar ulang ke SMK X Pekanbaru.

Menurut Wakil Kepala Sekolah bidang Kesiswaan dan Kepala tim PPDB SMK X, “Awal masuk sekolah atau sebelum melakukan ujian pertama yaitu UAS, siswa yang merasa kurang cocok dengan jurusan dapat meminta pindah jurusan”. Bahkan dengan jurusan yang dipilih sendiri, tidak sedikit yang merasa kurang cocok dan ingin pindah jurusan. Tidak hanya pada awal sekolah, setelah memasuki kelas 2 dan 3, terdapat siswa yang berhenti sekolah karena tidak sesuai dengan jurusan yang diambil.

Sesuai dengan hasil wawancara tersebut, dapat disimpulkan bahwa proses penjurusan masih menggunakan cara manual yaitu dengan penulisan data di kertas dan/atau secara digital menggunakan *Microsoft Excel*. Dengan jumlah pendaftar yang berjumlah kurang lebih 230 calon siswa per tahun, timbul kendala seperti kesalahan dalam memasukkan data serta membutuhkan waktu yang panjang untuk mengolah data tersebut dan memberikan hasil penjurusan yang sesuai bagi setiap calon siswa yang mendaftar (Ding & Luk, 2021; Rusmardiana et al., 2018; Yudi Hidayat et al., 2018).

Sebelumnya, terdapat penelitian yang terkait dengan pemilihan jurusan, contohnya adalah penelitian oleh (Deby, 2019) yang menyatakan bahwa dengan adanya permodelan klasifikasi penjurusan siswa menggunakan algoritma C4.5, kinerja guru dan siswa pada SMA 18 Palembang meningkat dan memberikan kemudahan dalam menentukan penjurusan siswa (Priyono et al., 2022). Selanjutnya terdapat penelitian oleh (Fibo et al., 2021) yang menyatakan bahwa menentukan penjurusan dengan menggunakan model klasifikasi *decision tree* C4.5 dapat mempercepat dan lebih akurat dalam pengambilan keputusan penjurusan siswa SMA N 2 Padang, Hal senada juga diungkapkan oleh (Atmaja, 2021) bahwa sistem penjurusan dapat membantu proses penentuan menjadi lebih baik .

Berdasarkan permasalahan tersebut, maka diperlukan sistem yang dapat meningkatkan kinerja guru dan calon siswa untuk mendapatkan kemudahan dalam penentuan penjurusan siswa. Pada penelitian ini akan dibuat sistem *Machine Learning* dengan menggunakan metode *Local Mean Based k-Nearest Neighbor* dan *Distance Weighting k-Nearest Neighbor* (LMDWkNN) yang merupakan pengembangan dari metode *k-Nearest Neighbor* dan sistem akan di implementasikan pada *Website*. Sistem ini diharapkan akan dapat membantu kegiatan penjurusan SMK agar dapat mempercepat dan meningkatkan akurasi dalam penjurusan SMK.

2. Literature Review

Penelitian Terdahulu

Metode penelitian *Local Mean Based* dan *Distance Weighting k-Nearest Neighbor* yang merupakan metode pengembangan dari algoritma *k-Nearest Neighbor* sudah pernah digunakan pada penelitian yang dilakukan oleh (Syaliman et al., 2018) dengan judul “Peningkatan Akurasi K-Nearest Neighbor Menggunakan Local Mean Based dan Distance Weight K-Nearest Neighbor”. Penelitian ini bertujuan untuk meningkatkan akurasi dari metode *k-Nearest Neighbor* (kNN) dengan cara menggantikan sistem *vote majority* pada *k-Nearest Neighbor* konvensional menggunakan metode *Distance Weight*. Hasil yang didapatkan dari penelitian tersebut adalah peningkatan akurasi sebesar 1.66% - 3.71% dengan rata-rata peningkatan akurasi sebesar 2.452% bila dibandingkan dengan menggunakan KNN konvensional.

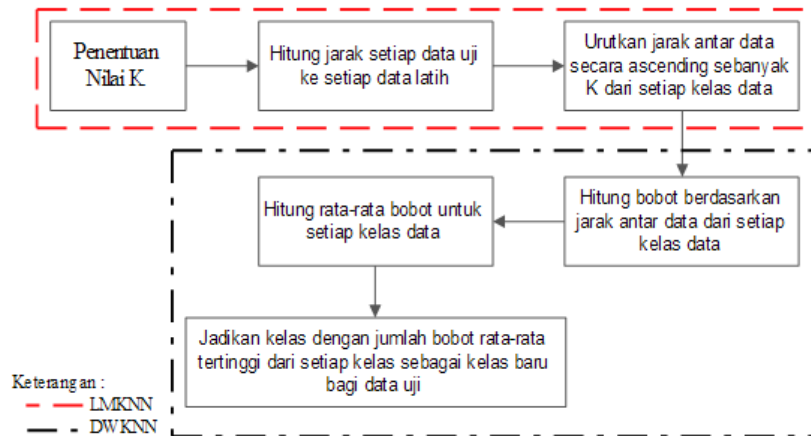
Penelitian terkait penentuan jurusan juga sudah pernah dilakukan oleh (Deby Afriansyah, 2019) dengan judul “Klasifikasi Menentukan Penjurusan Siswa Pada SMA 18 Palembang” yang menggunakan metode *Naïve Bayes* dan 5 variabel yaitu nilai matematika semester 1, nilai IPA semester 1, nilai matematika semester 2, nilai IPA semester 2, dan nilai tes. Hasil yang didapatkan berbasis *website* dan tingkat akurasi pada penelitian ini adalah 93.55%.

Selanjutnya adalah penelitian (Putra & Wibowo, 2020) dengan judul “Prediksi Keputusan Minat Penjurusan Siswa SMA Yadika 5 Menggunakan Algoritma *Naïve Bayes*” yang menggunakan variabel nilai Matematika dan IPA semester 1 dan 2. Dengan hasil yang dikeluarkan adalah hasil prediksi jurusan pada SMA. Tingkat akurasi yang didapatkan pada penelitian ini adalah 93.75%. Pada penelitian ini sudah terdapat sebuah sistem yang dapat melakukan prediksi terhadap jurusan siswa.

Selanjutnya adalah penelitian yang dilakukan (Fibo et al., 2021) dengan judul “Menentukan Penjurusan Siswa Dengan Menggunakan Metode *Decision Tree* Algoritma C4.5 (Studi Kasus : SMA Negeri 2 Padang)” yang menggunakan metode *Decision Tree* Algoritma C4.5 dan 5 variabel yaitu nilai B. Indonesia, nilai Matematika, nilai B. Inggris, nilai IPA, Rekomendasi Guru BK SMP. Hasil yang didapatkan berbasis model dengan menggunakan *tools RapidMiner* dan tingkat akurasi pada penelitian ini adalah 68,42%.

Local Mean Distance Weighted *k*-Nearest Neighbor

Metode *Local Mean Distance Weighted k-Nearest Neighbor* (LMDWkNN) yang diusulkan oleh (Syaliman et al., 2018) merupakan penggabungan dari metode LMKNN (Mitani & Hamamoto, 2006) dan DWkNN (Gou et al., 2011). Metode ini menggantikan sistem *vote majority* pada kNN dengan metode *Distance Weight* untuk mendapatkan bobot untuk setiap kelas data yang dihasilkan dari perhitungan kemiripan data dengan pendekatan *Local Mean*. Tahapan dari metode ini dapat dilihat pada gambar 1:



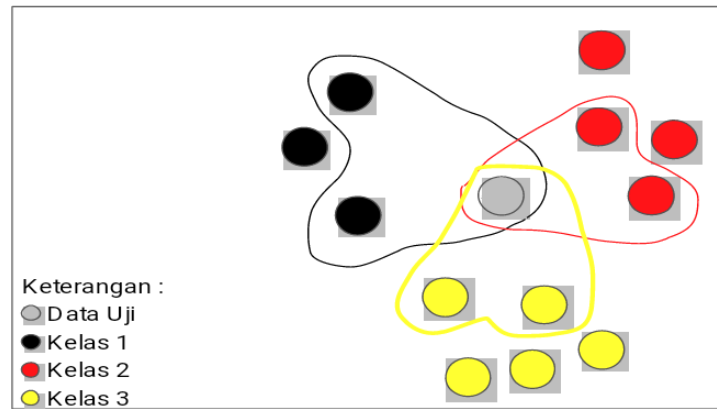
Gambar 1. Tahapan LMDWkNN

Berdasarkan gambar 1, dapat dilihat bahwa tahapan dari LMDWkNN terdiri dari beberapa langkah (Yuliska & Syaliman, 2020):

- i) Penentuan Nilai *k*, hal pertama yang dilakukan adalah penentuan jumlah tetangga terdekat berdasarkan nilai *k* untuk setiap kelas data nantinya.
- ii) Perhitungan jarak dari data uji ke seluruh data sampel menggunakan model jarak *Euclidean* (persamaan 1).

$$(x_1, x_2) = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2} \tag{1}$$

- iii) Lakukan pengurutan data secara *ascending* sebanyak *k* untuk setiap kelas data, perhatikan gambar 2 sebagai ilustrasi.



Gambar 2. Tetangga Terdekat Setiap Kelas

- iv) Hitungan bobot berdasarkan berdasarkan jarak antar data dari setiap kelas data menggunakan persamaan 2.

$$w_i = \frac{1}{d(x_q, x_i)} \quad (2)$$

- v) Selanjutnya akan hitung rata-rata bobot dari setiap kelas data dengan persamaan 3.

$$\text{sum_}w_c = \sum_{i=1}^{k_{NN}} w_i, (c = c_i^{NN}) \quad (3)$$

- vi) Pilih kelas dengan bobot tertinggi sebagai kelas untuk data yang baru.

Pengujian Model

Pengujian model yang diimplementasikan salah satu cara dapat dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan sebuah tabel matriks yang mempresentasikan hasil evaluasi model klasifikasi (Putra & Wibowo, 2020). Terdapat beberapa istilah dalam *confusion matrix*, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN).

True Positive (TP) adalah jumlah kelas aktual dan prediksinya merupakan kelas positif. Sedangkan *False Positive* (FP) merupakan jumlah kelas actual negatif yang kelas prediksinya merupakan kelas positif. Selanjutnya *True Negative* (TN) dimana kelas aktual dan kelas prediksinya merupakan kelas negatif. Terakhir *False Negative* (FN) yang merupakan jumlah kelas aktualnya kelas positif sedangkan kelas prediksinya negatif.

Nilai-nilai tersebut digunakan untuk menghitung kinerja model yang sudah diimplementasikan. Berikut beberapa indikator kinerja yang dapat digunakan:

- i) **Accuracy** merupakan efektivitas keseluruhan dari hasil pengklasifikasian.

$$ACC = \frac{TP+TN}{TP+TN+FN+FP} \quad (4)$$

- ii) **True Positive Rate (Recall)** merupakan efektivitas dari klasifikasi dalam melakukan identifikasi label positif.

$$TPR = \frac{TP}{TP+FN} \quad (5)$$

- iii) **Positive Predictive Value (Precision)** merupakan persentase label data dengan label positif dari klasifikasi.

$$PPV = \frac{TP}{TP+FP} \quad (6)$$

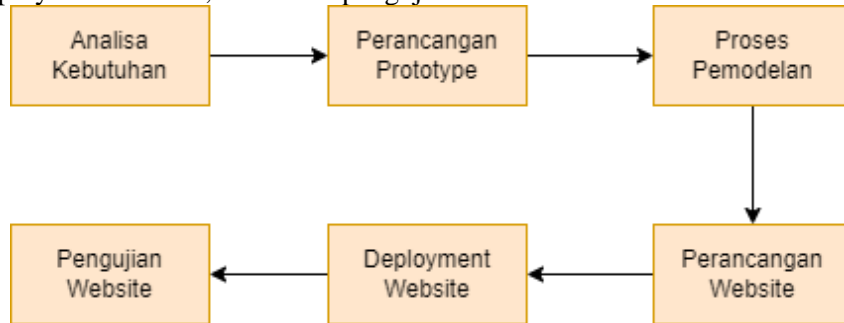
- iv) **F1-Score** merupakan perbandingan rata-rata *precision* dan *recall* yang telah dibobotkan.

$$F1 = 2 \times \frac{PPV \times TPR}{PPV+TPR} \quad (7)$$

3. Metode Penelitian

Alur Penelitian

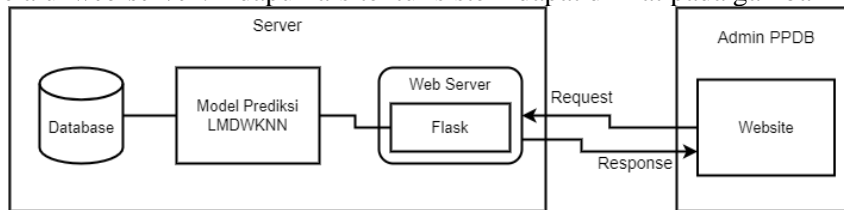
Alur penelitian dari perancangan aplikasi ini dapat dilihat pada gambar 3. Penelitian dimulai dari analisa kebutuhan, perancangan prototype, perancangan model, perancangan website, deployment website, kemudian pengujian website.



Gambar 3. Alur Penelitian

Arsitektur Sistem

Arsitektur sistem yang akan digunakan pada penelitian ini berjalan dengan memasukkan *dataset* oleh pengguna ke dalam sistem. Tepat setelah *dataset* disimpan ke *database*, akan berjalan klasifikasi penjurusan dengan algoritma LMDWKNN. Setelah klasifikasi berhasil, secara otomatis *dataset* dan hasil klasifikasi yang sudah dilakukan akan disimpan ke dalam *database* melalui *web server*. Adapun arsitektur sistem dapat dilihat pada gambar 4.



Gambar 4. Arsitektur Sistem

Perancangan dan Simulasi Model

Pada tahap perancangan model, akan dilakukan simulasi perhitungan berdasarkan pada sub bab 2 bagian *Local Mean Distance Weighted k-Nearest Neighbor* (LMDWkNN). Diasumsikan terdapat sebuah *dataset* dengan jumlah data sebanyak 12 record, yang mana data tersebut memiliki 8 atribut dan 5 kelas, sebanyak 80% dari data akan dijadikan sebagai data latih dan 20% dari data dijadikan sebagai data uji, dimana data uji akan ditentukan kelasnya. Adapun paparan dari rincian data dapat dilihat pada Tabel 3.3

Tabel 1. Rincian Data

No	X1	X2	X3	X4	X5	X6	X7	X8	Jurusan	Keterangan
1	79	82	92	91	81	73	86	95	TKJ	data latih 1
2	76	92	80	93	77	91	79	76	AKL	data latih 2
3	77	84	91	89	93	77	86	94	RPL	data latih 3
4	72	95	80	74	89	94	95	85	BDP	data latih 4
5	92	91	89	86	76	93	75	72	OTKP	data latih 5
6	75	95	83	92	73	90	83	76	AKL	data latih 6
7	95	93	84	71	75	91	71	79	OTKP	data latih 7
8	70	73	93	73	91	81	75	92	RPL	data latih 8
9	71	75	92	90	81	70	83	93	TKJ	data latih 9
10	76	93	81	81	81	96	93	87	BDP	data latih 10

11	79	94	88	87	79	80	88	91	TKJ	data uji 1
12	74	85	90	91	76	76	70	92	TKJ	data uji 2

Setelah data dibagi menjadi data latih dan data uji, dilakukan tahapan pertama dari klasifikasi dengan metode LMDWkNN yaitu:

- i) Penentuan Nilai k , dimisalkan k bernilai 2
- ii) Perhitungan jarak menggunakan model jarak *Euclidean* (persamaan 1).

$$D(Data Uji 1, Data Latih 1) = \sqrt{(79 - 79)^2 + (94 - 82)^2 + \dots + (91 - 95)^2}$$

$$D(Data Uji 1, Data Latih 1) = 15.78$$

adapun jarak dari setiap data latih dengan data uji 1 dapat dilihat pada tabel 2.

Tabel 2. Jarak Data Uji ke Setiap Data Latih

	Data Latih 1	Data Latih 2	Data Latih 3	Data Latih 4	Data Latih 5	Data Latih 6	Data Latih 7	Data Latih 8	Data Latih 9	Data Latih 10
Uji 1	15.78	23.32	18.30	25.77	29.80	21.28	33.15	32.53	24.15	19.90
Uji 2	18.36	26.83	23.77	40.74	33.17	27.95	36.57	27.71	18.57	35.04

- iii) Lakukan pengurutan data secara *ascending* sebanyak k untuk setiap kelas data. Pada tahapan ini, lakukan terlebih dahulu pengurutan terhadap seluruh data (tabel 3)

Tabel 3. Urutan Jarak Data Latih dan Data Uji

	Urutan Jarak Terdekat									
	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th
	Data Latih									
Uji 1	1	3	10	6	2	9	4	5	8	7
Uji 2	1	9	3	2	8	6	5	10	7	4

Selanjutnya lakukan proses pengurutan berdasarkan kelas data sebanyak k

Tabel 4. Urutan Jarak Terdekat Untuk Setiap Kelas Data

Data	Kelas	Urutan Data	
Uji 1	TKJ	Data Latih 1	Data Latih 9
	RPL	Data Latih 3	Data Latih 8
	AKL	Data Latih 6	Data Latih 2
	OTKP	Data Latih 5	Data Latih 7
	BDP	Data Latih 10	Data Latih 4
Uji 2	TKJ	Data Latih 1	Data Latih 9
	RPL	Data Latih 3	Data Latih 8
	AKL	Data Latih 2	Data Latih 6
	OTKP	Data Latih 5	Data Latih 7
	BDP	Data Latih 10	Data Latih 4

- iv) Hitungan bobot dari setiap kelas data menggunakan persamaan 2. Hasil dari perhitungan bobot jarak dapat dilihat pada tabel 5.

Tabel 5. Bobot Jarak k Tetangga Terdekat dari Setiap Kelas

Data	Kelas	Bobot Jarak	
Uji 1	TKJ	0.0634	0.0414
	RPL	0.0546	0.0307
	AKL	0.0470	0.0429
	OTKP	0.0336	0.0302

Data	Kelas	Bobot Jarak	
	BDP	0.0503	0.0388
Uji 2	TKJ	0.0545	0.0538
	RPL	0.0421	0.0307
	AKL	0.0373	0.0358
	OTKP	0.0302	0.0273
	BDP	0.0285	0.0245

- v) Hitung rata-rata bobot dari setiap kelas data dengan persamaan 3 (Tabel 6).

Tabel 6. Bobot Rata-rata Setiap Kelas

Data	Kelas	Rata-rata bobot
Uji 1	TKJ	0.0524
	RPL	0.0427
	AKL	0.0449
	OTKP	0.0319
	BDP	0.0445
Uji 2	TKJ	0.0542
	RPL	0.0364
	AKL	0.0365
	OTKP	0.0287
	BDP	0.0265

- vi) Pilih kelas dengan bobot tertinggi sebagai kelas untuk data yang baru. Berdasarkan tabel 6, dapat dilihat bahwa bobot tertinggi untuk data uji 1 dan uji 2 adalah TKJ dengan nilai 0.0524 dan 0.0542 berturut-turut, sehingga TKJ dipilih sebagai kelas untuk data uji 1 dan uji 2.

Untuk model telah diimplementasikan, akan dilakukan evaluasi dengan menggunakan *Confusion Matrix*. Evaluasi yang dilakukan akan mengukur kinerja dari model yang diimplementasikan dengan melihat nilai *accuracy*, *precision*, *recall*, dan *F-1 score*. Jika menggunakan hasil klasifikasi sebelumnya, maka nilai dari *accuracy*, *precision*, *recall*, dan *F-1 score* adalah 100%.

4. Hasil dan Pembahasan

LMDWkNN diimplementasikan dengan menggunakan Bahasa pemrograman python sebagai backend, html dan javascript sebagai frontend, serta *library flask* di pada python digunakan untuk menghubungkan backend dan frontend (*router*). Adapun potongan *source code* dari LMDWkNN adalah sebagai berikut (*Source Code 1*).

```
class LMDWKNN:
    # tentukan nilai K
    def __init__(self, k):
    # masukkan data latih kelas nya
    def fit(self, X, y):
    # rumus jarak Euclidean Distance
    def distance(self, X1, X2):
        return np.sqrt(np.sum(np.square(X1 - X2)))
    # rumus bobot
    def weight(self, dist):
        return 1 / (dist + 1e-8)
    def predict(self, X_test):
        ...
        # hitung jarak antar data uji dan data latih
        for i in range(len(X_test)):
            ...
            # Urutkan jarak sebanyak k dari setiap kelas
```

```

...
# hitung bobot untuk setiap kelas data
...
# hitung rata rata bobot
...
# ambil kelas dengan rata rata bobot tertinggi
...
return final_output
...

```

Source Code 1. Backend LMDWkNN

Berdasarkan source code 1, dilakukan proses pencarian nilai k terbaik yang digunakan sebagai panduan untuk proses klasifikasi pada data yang baru. Adapun proses pencarian nilai k pada proses training ini adalah dengan mencoba nilai k dari 1 s/d 10. Adapun hasil eksekusi pada proses pelatihan ini dapat dilihat pada gambar 5.

```

test_K = [1,2,3,4,5,6,7,8,9,10]

for i in test_K:
    lmdwknn = LMDWKNN(i)
    lmdwknn.fit(X_train, y_train)
    result = lmdwknn.score(X_test, y_test)
    print('For K = {} Accuracy with LMDWKNN Classifier is {:.2%}'.format(i, result))

```

✓ 10.6s

For K = 1 Accuracy with LMDWKNN Classifier is 88.89%
 For K = 2 Accuracy with LMDWKNN Classifier is 86.11%
 For K = 3 Accuracy with LMDWKNN Classifier is 85.56%
 For K = 4 Accuracy with LMDWKNN Classifier is 83.33%
 For K = 5 Accuracy with LMDWKNN Classifier is 83.89%
 For K = 6 Accuracy with LMDWKNN Classifier is 81.67%
 For K = 7 Accuracy with LMDWKNN Classifier is 81.11%
 For K = 8 Accuracy with LMDWKNN Classifier is 79.44%
 For K = 9 Accuracy with LMDWKNN Classifier is 77.78%
 For K = 10 Accuracy with LMDWKNN Classifier is 75.00%

Gambar 5. Proses Training

Selanjutnya melakukan proses klasifikasi pada data yang baru dengan menggunakan nilai k yang memiliki nilai akurasi paling tinggi, yaitu saat k bernilai 1. Adapun hasil implementasi pada proses klasifikasi dapat dilihat pada gambar 6.

```

lmdwknn = LMDWKNN(1)
lmdwknn.fit(X_train, y_train)
result = lmdwknn.score(X_test, y_test)
print('Akurasi dengan LMDWKNN adalah: {:.2%}'.format(result))

```

✓ 1.0s

Akurasi dengan LMDWKNN adalah: 88.89%

Gambar 6. Proses Klasifikasi

Selanjutnya melakukan evaluasi model dengan menggunakan *confusion matrix*, adapun hasil implementasi dapat dilihat pada gambar 7.

```

from sklearn.metrics import confusion_matrix, classification_report

lmdwknn = LMDWKNN(1)
lmdwknn.fit(X_train, y_train)
y_pred = lmdwknn.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print('Classification report: \n', classification_report(y_test, y_pred))

```

✓ 1.4s

Classification report:				
	precision	recall	f1-score	support
AKT	0.80	0.94	0.86	34
BDP	0.95	0.97	0.96	36
OTKP	0.91	0.76	0.83	42
RPL	0.91	0.88	0.89	33
TKJ	0.89	0.91	0.90	35
accuracy			0.89	180
macro avg	0.89	0.89	0.89	180
weighted avg	0.89	0.89	0.89	180

Gambar 7. Evaluasi Model Klasifikasi

Setelah model dirasa memiliki performa yang cukup baik dalam melakukan proses klasifikasi tahapan terakhir adalah menghubungkan backend dan frontend, adapun *source code* dari router ini dapat dilihat pada gambar 8.

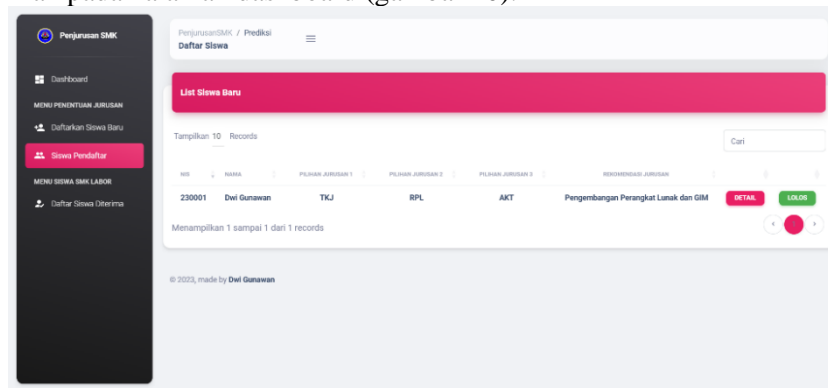
```

1 @views.route('/')
2 def home():
3     menu = 'Dashboard'
4     submenu = 'Dashboard'
5     conn = mysqlconnect()
6     cur = conn.cursor()
7
8     cur.execute("SELECT COUNT(nis) as jumlah FROM `siswa`")
9     jumlahdata = cur.fetchone()
10    jumlah = int(jumlahdata[0])

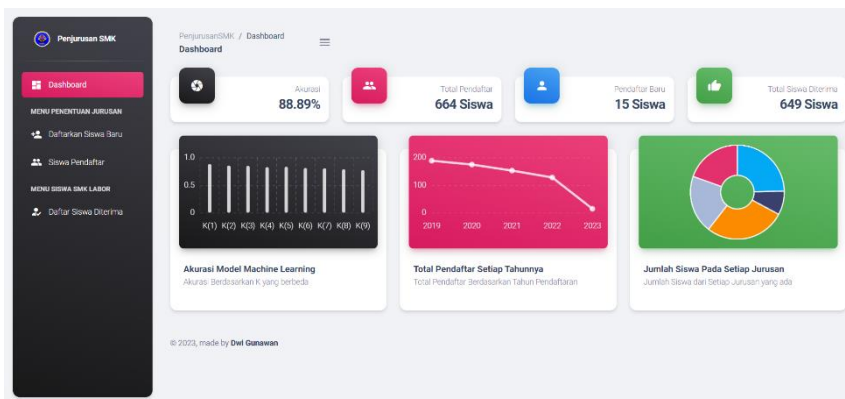
```

Gambar 8. Router Dashboard

Setelah dihubungkan hasil klasifikasi pada data yang baru akan langsung dilakukan secara otomatis setelah data baru tersebut dimasukkan nilai-nilainya saat proses pendaftaran siswa baru (gambar 9). Adapun rekapan informasi umum serta hasil dari akurasi dari proses klasifikasi akan ditampilkan pada halaman dashboard (gambar 10).



Gambar 9. Pendaftaran Siswa



Gambar 10. Dashboard

5. Kesimpulan

Berdasarkan hasil pengujian dengan menggunakan *confusion matrix* menunjukkan nilai akurasi sebesar 88.89%, serta nilai untuk presisi, *recall* dan *F1 score* sebesar 89%, hal ini menunjukkan bahwa metode LMDWkNN kinerja yang bagus dalam penentuan jurusan siswa SMK X di Pekanbaru.

References

Atmaja, N. S. (2021). Attribution-NonCommercial 4.0 International. Some rights reserved Sistem Pendukung Keputusan Sistem Pendukung Keputusan Pemilihan Jurusan Menggunakan Metode PROMETHEE (Studi Kasus: SMK Negeri 6 Medan). *Jurnal Nasional Informatika Dan Teknologi Jaringan*, 5(2), 75–84.

- <https://doi.org/10.30743/infotekjar.v5i2.3575>
- Deby Afriansyah, M. (2019). KLASIFIKASI MENENTUKAN PENJURUSAN SISWA PADA SMA 18 PALEMBANG. *Bina Darma Conference on Computer Science*.
- Ding, C., & Luk, K.-M. (2021). Determination of Major Axis of Elliptically Polarized Wave Generated by Artificial Anisotropic Polarizers. *IEEE Transactions on Antennas and Propagation*, 69(10), 7059–7063. <https://doi.org/10.1109/TAP.2021.3076268>
- Fibo, M., Ikhbal, D., & Kurniadi, D. (2021). Menentukan Penjurusan Siswa Dengan Menggunakan Metode Decision Tree Algoritma C4.5 (Studi Kasus : SMA Negeri 2 Padang).
- Gou, J., Xiong, T., & Kuang, Y. (2011). A novel weighted voting for K-nearest neighbor rule. *Journal of Computers*, 6(5), 833–840. <https://doi.org/10.4304/jcp.6.5.833-840>
- Howay, S., & Rianto. (2021). Sistem Rekomendasi Jurusan Pada Sekolah Menengah Kejuruan (Smk) Dengan Algoritma K-Means. *Syntax Idea*, 3(10), 2215–2222. <https://doi.org/10.1016/j.solener.2019.02.027%0Ahttps://www.golder.com/insights/block-caving-a-viable-alternative/%0A???>
- Mafakhir, A. Z., & Solichin, A. (2020). Penerapan Metode Naïve Bayes Classifier Untuk Penjurusan Siswa Pada Madrasah Aliyah Al-Falah Jakarta. *Fountain of Informatics Journal*, 5(1), 21. <https://doi.org/10.21111/fij.v5i1.4007>
- Mitani, Y., & Hamamoto, Y. (2006). A local mean-based nonparametric classifier. *Pattern Recognition Letters*, 27(10), 1151–1159. <https://doi.org/10.1016/j.patrec.2005.12.016>
- Priyono, H., Sari, R., & Mardiana, T. (2022). Klasifikasi Pemilihan Jurusan Sekolah Menengah Kejuruan Menggunakan Gradient Boosting Classifier. *Jurnal Informatika*, 9(2), 131–139. <https://doi.org/10.31294/inf.v9i2.12654>
- Putra, D., & Wibowo, A. (2020). Prediksi Keputusan Minat Penjurusan Siswa SMA Yadika 5 Menggunakan Algoritma Naïve Bayes. *Prosiding Seminar Nasional Riset Dan Information Science (SENARIS)*, 2, 84–92.
- Rawal Dewa, S., & Jasmir, J. (2023). Sistem Pendukung Keputusan Pemilihan Jurusan Menggunakan Metode SAW Pada SMK Negeri 2 Sarolangun. *Jurnal Manajemen Sistem Informasi*, 8(1), 115–127. <https://doi.org/10.33998/jurnalmsi.2023.8.1.768>
- Rusmardiana, A., Akhirina, T. Y., Yulistyanti, D., & Pauziah, U. (2018). A Web-Based High School Major Decision Support System in Banten Using Tsukamoto's Fuzzy Method. *2018 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 233–238. <https://doi.org/10.1109/ISITIA.2018.8711337>
- Syaliman, K. U., Nababan, E. B., & Sitompul, O. S. (2018). Improving the accuracy of k-nearest neighbor using local mean based and distance weight. *Journal of Physics: Conference Series*, 978(1), 1–6. <https://doi.org/10.1088/1742-6596/978/1/012047>
- Yudi Hidayat, E., Sabiq Taufiqurrahman, A., Luthfiarta, A., Zeniarja, J., Agus Santoso, H., Nugraha, A., & Rismiyati. (2018). Implementation of Weighted Naive Bayes Algorithm for Major Determination in Indonesian High School. *2018 International Seminar on Application for Technology of Information and Communication*, 580–584. <https://doi.org/10.1109/ISEMANTIC.2018.8549761>
- Yuliska, Y., & Syaliman, K. U. (2020). Peningkatan Akurasi K-Nearest Neighbor Pada Data Index Standar Pencemaran Udara Kota Pekanbaru. *IT Journal Research and Development*, 5(1), 11–18. [https://doi.org/10.25299/itjrd.2020.vol5\(1\).4680](https://doi.org/10.25299/itjrd.2020.vol5(1).4680)