BAB IV. HASIL DAN PEMBAHASAN

4.1 Pendahuluan

Bab ini memberikan gambaran hasil eksperimen dan analisis data terkait perbandingan algoritma *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM) dalam prediksi pengangguran di Provinsi Lampung. Analisis yang mendalam akan dilakukan untuk mengevaluasi kinerja kedua algoritma dan memberikan pemahaman yang lebih baik tentang faktor-faktor yang mempengaruhi hasil prediksi.

4.2 Hasil

Setelah melakukan penelitian tentang perbandingan algoritma *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM) dalam prediksi pengangguran di Provinsi Lampung yang menggunakan bahasa pemrograman Python. Dalam tahapannya terdapat proses *Train Val Test* untuk menghindari *Data Leakage* yaitu kebocoran data Test yang masuk ke dalam proses *Training* dan *Validation*. Pembagian pada tahapan *Data Splitting* sebanyak 80 : 20, yaitu 80% data *Training* dan 20% data *Testing*. Pada prosesnya juga melakukan beberapa *Feature Engineering* yang menghasilkan beberapa data turunan, membuat binning pada kolom umur, melakukan *Cross Validation*, dan juga melakukan *Grid Search*, dan lain-lain. Membungkus semua proses ke dalam beberapa Pipeline. Target dari penelitian ini yaitu bekerja berkode 1 dan pengangguran berkode 2.

4.2.1 *Preprocessing*

Penelitian ini menggunakan Data Sakernas (Survei Angkatan Kerja Nasional) Provinsi Lampung yang memiliki 29.999 baris data dan 220 kolom.

	TAHUN	URUTAN	WEIGHT	KODE_PROV	KODE_KAB	PSU	SSU	STRATA	KLAS	JUMLAHUMUR	...	R47H3	R47H4	R47H5	R48_1	R48_2
0	20228	248991	191	18	1	10573	145360	182	2	4	...	NaN	NaN	NaN	2.0	4.0
1	20228	248992	119	18	1	10573	145360	182	2	4	...	NaN	NaN	NaN	2.0	4.0
2	20228	248993	298	18	1	10573	145360	182	2	4	...	NaN	NaN	NaN	2.0	4.0
3	20228	248995	202	18	1	10573	97985	182	2	7	...	NaN	NaN	NaN	2.0	4.0
4	20228	248996	203	18	1	10573	97985	182	2	7	...	NaN	NaN	NaN	2.0	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22994	20228	280558	90	18	72	5598	171768	181	1	4	...	NaN	NaN	NaN	1.0	4.0
22995	20228	280559	132	18	72	5598	171768	181	1	4	...	NaN	NaN	NaN	1.0	4.0
22996	20228	280560	150	18	72	5598	171768	181	1	4	...	NaN	NaN	NaN	1.0	3.0
22997	20228	280561	100	18	72	5598	240056	181	1	4	...	NaN	NaN	NaN	1.0	3.0
22998	20228	280562	133	18	72	5598	240056	181	1	4	...	NaN	NaN	NaN	1.0	3.0

22999 rows × 220 columns

Gambar 4. 1 Data Sakernas Provinsi Lampung

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22999 entries, 0 to 22998
Columns: 220 entries, TAHUN to JENISKEGIA
dtypes: float64(175), int64(45)
memory usage: 38.6 MB
```

Berdasarkan beberapa penelitian terdahulu tidak semua atribut akan digunakan untuk penelitian ini melainkan hanya 21 atribut yang terpilih. Adapun atribut yang digunakan dalam penentuan klasifikasi pengangguran di Provinsi Lampung dapat dilihat pada tabel 4.1 :

Tabel 4. 1 Variabel yang Digunakan Sebagai Bahan Penelitian Berdasarkan Penelitian Terdahulu.

No	Variabel	Label
1	KLAS	Klasifikasi Perkotaan/Perdesaan
2	K4	Jenis Kelamin
3	K6	Umur
4	R4	Status Perkawinan
5	R6A	Pendidikan tertinggi yang ditamatkan
6	R6D	Pernah mengikuti pelatihan/kursus/training

7	R6E	Sertifikat dari pelatihan/kursus/training
8	R8A	Kesulitan melihat
9	R8B	Kesulitan mendengar
10	R8C	Kesulitan berjalan/naik tangga
11	R8D	Kesulitan memegang
12	R8E	Kesulitan berbicara
13	R8F	Kesulitan lain
14	R9A	Bekerja seminggu terakhir minimal 1 Jam
15	R9B	Melakukan kegiatan untuk memperoleh penghasilan / pendapatan uang dalam seminggu terakhir
16	R9C	Membantu kegiatan usaha atau pekerjaan keluarga/orang lain dalam seminggu terakhir
17	R10	Sementara tidak bekerja Seminggu terakhir dalam seminggu yang lalu, dan sebenarnya memiliki pekerjaan
18	R38A	Kapan memperoleh pekerjaan/memulai usaha setelah lulus dari pendidikan tertinggi yang ditamatkan
19	R38B	Pernah punya pekerjaan/usaha sebelumnya
20	R42A	Alasan utama berhenti bekerja di pekerjaan terakhir
21	R47B	Mendaftar program kartu prakerja

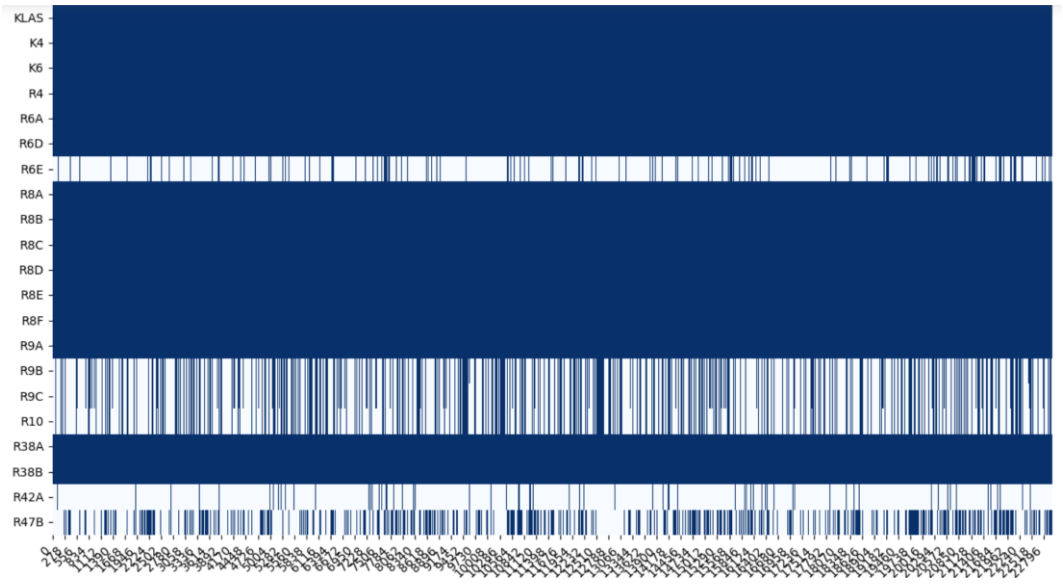
Terlihat masih ada beberapa kolom yang bernilai NaN. Sehingga perlu dilakukan perlakuan seperti imputasi nilai kedalam kolom tersebut pada setiap barisnya.

	KLAS	K4	K6	R4	R6A	R6D	R6E	R8A	R8B	R8C	...	R8E	R8F	R9A	R9B	R9C	R10	R38A	R38B	R42A	R47B
0	2	1	43	2	4	2	NaN	4	8	4	...	4	8	1	NaN	NaN	NaN	1	1	NaN	NaN
1	2	2	38	2	3	2	NaN	4	8	4	...	4	8	2	2.0	2.0	2.0	2	2	NaN	NaN
2	2	1	20	1	5	2	NaN	4	8	4	...	4	8	2	2.0	2.0	2.0	3	2	NaN	NaN
3	2	1	30	2	7	1	1.0	4	8	4	...	4	8	1	NaN	NaN	NaN	1	1	3.0	NaN
4	2	2	27	2	7	2	NaN	4	8	4	...	4	8	2	2.0	2.0	2.0	2	1	NaN	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22994	1	2	46	2	3	2	NaN	4	8	4	...	4	8	2	2.0	2.0	2.0	1	1	NaN	NaN
22995	1	1	26	1	4	2	NaN	4	8	4	...	4	8	1	NaN	NaN	NaN	2	2	NaN	2.0
22996	1	2	18	2	4	2	NaN	4	8	4	...	4	8	2	2.0	2.0	2.0	3	2	NaN	2.0
22997	1	1	32	2	3	2	NaN	4	8	4	...	4	8	1	NaN	NaN	NaN	1	1	NaN	NaN
22998	1	2	26	2	3	2	NaN	4	8	4	...	4	8	2	2.0	2.0	2.0	1	1	NaN	NaN

22999 rows × 21 columns

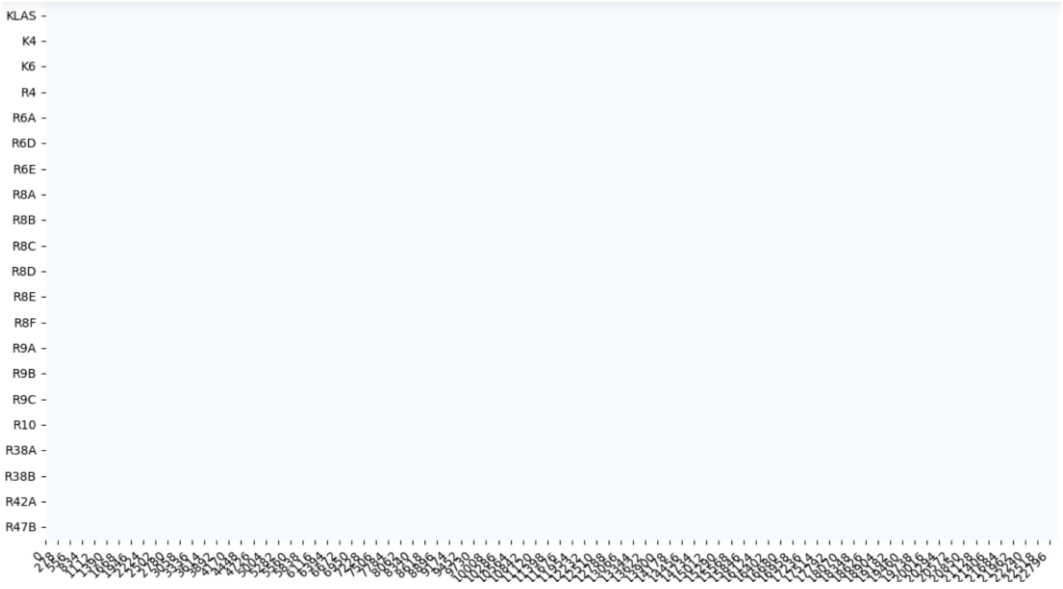
Gambar 4. 2 Data Sakernas masih ada bernilai NaN

Dari gambar berikut tampak pada R6E, R9B, R9C, R10, R42A, R47B, masih ada nilai-nilai yang kosong sehingga perlu di imputasi nilai.



Gambar 4. 3 *Missing Value*

Berdasarkan redaksi pertanyaan dari kuesioner Sakernas maka untuk rincian R6E, R9B, R9C, R10, R47B diisi dengan nilai 2, sedangkan untuk R42A diisi dengan nilai 10. Setelah dilakukan imputasi pada ke 6 rincian diatas maka sudah tidak ada lagi missing value dari data tersebut yang dibuktikan dengan gambar di bawah ini sehingga data bisa langsung dilakukan ke tahap selanjutnya.



Gambar 4. 4 *Missing Value* telah terisi

Setelah semua *Missing Value* sudah di imputasi maka tahapan selanjutnya adalah melakukan *Fiture Engineering* membuat data turunan yaitu dengan membuat fungsi disabilitas, dimana fungsi disabilitas ini menggabungkan beberapa variabel seperti kesulitan melihat (R8A), kesulitan mendengar (R8B), kesulitan berjalan/naik tangga (R8C), kesulitan memegang (R8D), kesulitan berbicara (R8E), dan kesulitan lain (R8F) menjadi satu variabel disabilitas yang diberi label 1 adalah penyandang disabilitas dan 2 bukan penyandang disabilitas. Agar mempermudah proses training nantinya kolom R8 sebagai kolom disabilitas diletakkan setelah kolom R6E. Seperti tampak pada gambar berikut ini.

	KLAS	K4	K6	R4	R6A	R6D	R6E	R8	R8A	R8B	...	R8E	R8F	R9A	R9B	R9C	R10	R38A	R38B	R42A	R47B
0	2	1	43	2	4	2	2.0	2	4	8	...	4	8	1	2.0	2.0	2.0	1	1	10.0	2.0
1	2	2	38	2	3	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	2	2	10.0	2.0
2	2	1	20	1	5	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	3	2	10.0	2.0
3	2	1	30	2	7	1	1.0	2	4	8	...	4	8	1	2.0	2.0	2.0	1	1	3.0	2.0
4	2	2	27	2	7	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	2	1	10.0	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22994	1	2	46	2	3	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	1	1	10.0	2.0
22995	1	1	26	1	4	2	2.0	2	4	8	...	4	8	1	2.0	2.0	2.0	2	2	10.0	2.0
22996	1	2	18	2	4	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	3	2	10.0	2.0
22997	1	1	32	2	3	2	2.0	2	4	8	...	4	8	1	2.0	2.0	2.0	1	1	10.0	2.0
22998	1	2	26	2	3	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	1	1	10.0	2.0

22999 rows × 22 columns

Gambar 4. 5 Pemindahan letak kolom R8

Selain fungsi disabilitas *Fiture Engineering* lain yang dilakukan adalah binning kelompok umur. Berdasarkan klasifikasi usia menurut Kementerian Kesehatan sebagai berikut: 1) Masa Balita: 0–5 Tahun; 2) Masa Kanak-Kanak: 5–11 Tahun; 3) Masa Remaja Awal: 12–16 Tahun; 4) Masa Remaja Akhir: 17–25 Tahun; 5) Masa Dewasa Awal: 26–35 Tahun; 6) Masa Dewasa Akhir: 36–45 Tahun; 7) Masa Lansia Awal: 46–55 Tahun; 8) Masa Lansia Akhir: 56–65 Tahun; dan 9) Masa Manula: > 65 Tahun[27]. Hasil dari binning umur langsung menggantikan kolom sebelumnya. Seperti pada gambar berikut ini.

	KLAS	K4	K6	R4	R6A	R6D	R6E	R8	R8A	R8B	...	R8E	R8F	R9A	R9B	R9C	R10	R38A	R38B	R42A	R47B
0	2	1	6	2	4	2	2.0	2	4	8	...	4	8	1	2.0	2.0	2.0	1	1	10.0	2.0
1	2	2	6	2	3	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	2	2	10.0	2.0
2	2	1	4	1	5	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	3	2	10.0	2.0
3	2	1	5	2	7	1	1.0	2	4	8	...	4	8	1	2.0	2.0	2.0	1	1	3.0	2.0
4	2	2	5	2	7	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	2	1	10.0	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22994	1	2	7	2	3	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	1	1	10.0	2.0
22995	1	1	5	1	4	2	2.0	2	4	8	...	4	8	1	2.0	2.0	2.0	2	2	10.0	2.0
22996	1	2	4	2	4	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	3	2	10.0	2.0
22997	1	1	5	2	3	2	2.0	2	4	8	...	4	8	1	2.0	2.0	2.0	1	1	10.0	2.0
22998	1	2	5	2	3	2	2.0	2	4	8	...	4	8	2	2.0	2.0	2.0	1	1	10.0	2.0

22999 rows × 22 columns

Gambar 4. 6 *Binning* Umur

Fiture Engineering selanjutnya yaitu menggabungkan kolom Bekerja seminggu terakhir minimal 1 jam (R9A), Melakukan kegiatan untuk memperoleh penghasilan/pendapatan uang dalam seminggu terakhir (R9B), Membantu kegiatan usaha atau pekerjaan keluarga / orang lain dalam seminggu terakhir (R9C), dan Sementara tidak bekerja seminggu terakhir dalam seminggu yang lalu, dan sebenarnya memiliki pekerjaan (R10). Keempat kolom digabungkan menjadi sebuah kolom yaitu kolom Target yang menentukan bekerja atau pengangguran. Kode 1 untuk bukan pengangguran dan kode 2 untuk pengangguran.

	KLAS	K4	K6	R4	R6A	R6D	R6E	R8	R8A	R8B	...	R8F	R9A	R9B	R9C	R10	R38A	R38B	R42A	R47B	TARGET
0	2	1	6	2	4	2	2.0	2	4	8	...	8	1	2.0	2.0	2.0	1	1	10.0	2.0	1
1	2	2	6	2	3	2	2.0	2	4	8	...	8	2	2.0	2.0	2.0	2	2	10.0	2.0	2
2	2	1	4	1	5	2	2.0	2	4	8	...	8	2	2.0	2.0	2.0	3	2	10.0	2.0	2
3	2	1	5	2	7	1	1.0	2	4	8	...	8	1	2.0	2.0	2.0	1	1	3.0	2.0	1
4	2	2	5	2	7	2	2.0	2	4	8	...	8	2	2.0	2.0	2.0	2	1	10.0	2.0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22994	1	2	7	2	3	2	2.0	2	4	8	...	8	2	2.0	2.0	2.0	1	1	10.0	2.0	2
22995	1	1	5	1	4	2	2.0	2	4	8	...	8	1	2.0	2.0	2.0	2	2	10.0	2.0	1
22996	1	2	4	2	4	2	2.0	2	4	8	...	8	2	2.0	2.0	2.0	3	2	10.0	2.0	2
22997	1	1	5	2	3	2	2.0	2	4	8	...	8	1	2.0	2.0	2.0	1	1	10.0	2.0	1
22998	1	2	5	2	3	2	2.0	2	4	8	...	8	2	2.0	2.0	2.0	1	1	10.0	2.0	2

22999 rows × 23 columns

Gambar 4. 7 Pembentukan Kolom Target

Setelah *Fiture Engineering* selesai dilakukan kolom pembentuk data turunan dihapus untuk mempermudah proses selanjutnya. Jumlah total keseluruhan kolom setelah Preprocessing adalah 13 kolom yaitu Klasifikasi Perkotaan/Perdesaan (KLAS), Jenis Kelamin (K4), Umur (K6), Status Perkawinan (R4), Pendidikan tertinggi yang ditamatkan (R6A), Pernah mengikuti pelatihan/kursus/training (R6D), Sertifikat dari pelatihan/kursus/training (R6E), Disabilitas (R8), Kapan memperoleh pekerjaan/memulai usaha setelah lulus dari

Pendidikan tertinggi yang ditamatkan (R38A), Pernah punya pekerjaan/usaha sebelumnya (R38B), Alasan utama berhenti bekerja di pekerjaan terakhir (R42A), Mendaftar program kartu prakerja (R47B).

	KLAS	K4	K6	R4	R6A	R6D	R6E	R8	R38A	R38B	R42A	R47B	TARGET
0	2	1	6	2	4	2	2.0	2	1	1	10.0	2.0	1
1	2	2	6	2	3	2	2.0	2	2	2	10.0	2.0	2
2	2	1	4	1	5	2	2.0	2	3	2	10.0	2.0	2
3	2	1	5	2	7	1	1.0	2	1	1	3.0	2.0	1
4	2	2	5	2	7	2	2.0	2	2	1	10.0	2.0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
22994	1	2	7	2	3	2	2.0	2	1	1	10.0	2.0	2
22995	1	1	5	1	4	2	2.0	2	2	2	10.0	2.0	1
22996	1	2	4	2	4	2	2.0	2	3	2	10.0	2.0	2
22997	1	1	5	2	3	2	2.0	2	1	1	10.0	2.0	1
22998	1	2	5	2	3	2	2.0	2	1	1	10.0	2.0	2

22999 rows × 13 columns

Gambar 4. 8 Menghilangkan kolom pembentuk data turunan

Tahapan *preprocessing* menghasilkan 12 variabel yang nantinya akan menjadi fitur sebelum dilakukan training model. Berikut adalah daftar variabel tersebut :

Tabel 4. 2 Ringkasan variabel yang digunakan dalam penelitian

No	Variabel	Label	Nilai
1	KLAS	Klasifikasi Perkotaan/Perdesaan	1 = Perkotaan 2 = Perdesaan
2	K4	Jenis Kelamin	1 = Laki laki 2 = Perempuan
3	K6	Umur	1 = Masa Balita: 0–5 Tahun 2 = Masa Kanak-Kanak: 6–11 Tahun 3 = Masa Remaja Awal: 12–16 Tahun 4 = Masa Remaja Akhir: 17–25 Tahun 5 = Masa Dewasa Awal: 26–35 Tahun

			6 = Masa Dewasa Akhir: 36–45 Tahun 7 = Masa Lansia Awal: 46–55 Tahun 8 = Masa Lansia Akhir: 56–65 Tahun 9 = Masa Manula: > 65 Tahun
4	R4	Status Perkawinan	1 = Belum kawin 2 = Kawin 3 = Cerai Hidup 4 = Cerai mati
5	R6A	Pendidikan tertinggi yang ditamatkan	1 = Tidak/belum tamat SD 2 = SD/MI/SDLB/Paket A 3 = SMP/MTs/SMPLB/Paket B 4 = SMA/MA/SMLB/Paket C 5 = SMK 6 = MAK 7 = Diploma I/II/III 8 = Diploma IV 9 = S1 10 = S2 11 = S2 Terapan 12 = S3
6	R6D	Pernah mengikuti pelatihan/kursus/training	1 = Ya 2 = Tidak
7	R6E	Sertifikat dari pelatihan/kursus/training	1 = Ya 2 = Tidak
8	R8	Disabilitas	1 = Ya 2 = Tidak
9	R38A	Kapan memperoleh pekerjaan/memulai usaha setelah lulus dari pendidikan tertinggi yang ditamatkan	1 = Bekerja setelah lulus pendidikan tertinggi 2 = Sudah bekerja sebelum lulus Pendidikan tertinggi 3 = Belum pernah bekerja / memulai usaha sejak lulus pendidikan tertinggi

10	R38B	Pernah punya pekerjaan/usaha sebelumnya	1 = Ya 2 = Tidak
11	R42A	Alasan utama berhenti bekerja di pekerjaan terakhir	1 = PHK 2 = Usaha terhenti / Bangkrut 3 = Pendapatan kurang memuaskan 4= Tidak cocok dengan lingkungan kerja 5= Habis masa kerja / kontrak 6= Mengurus rumah tangga 7= Takut terinfeksi Corona / COVID-19 8= Social/physical distancing, karantina mandiri. Perlakuan Pembatasan Kegiatan Masyarakat (PPKM) 9= Selain alasan diatas 10= Lainnya
12	R47B	Mendaftar program kartu prakerja	1 = Ya 2 = Tidak

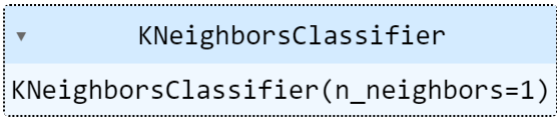
4.2.2 *Training Model*

Pelatihan model dalam konteks machine learning adalah proses di mana model pembelajaran mesin "mempelajari" pola atau hubungan dari data latihan. Data latihan adalah sekumpulan contoh input dan output yang digunakan untuk mengajar model. Tujuan pelatihan model adalah untuk membuat model yang dapat membuat prediksi yang akurat atau mengambil keputusan yang tepat tergantung pada jenis tugas pembelajaran mesin yang sedang dijalankan.

4.2.2.1. Proses Metode Klasifikasi *K-Nearest Neighbors* (KNN)

Tahapan awal proses *Training Model* pada metode klasifikasi KNN yaitu menentukan jumlah tetangga terdekatnya (n_neighbors). Jumlah tetangga terdekat

awal untuk penelitian ini adalah 1, menggunakan *KNeighborsClassifier*. Seperti pada gambar



Gambar 4. 9 KNeighborsClassifier dengan 1 tetangga terdekat

Dari ujicoba acak diatas didapat score 0.81.

4.2.2.2 *Tuning* Parameter Menentukan Nilai Terbaik

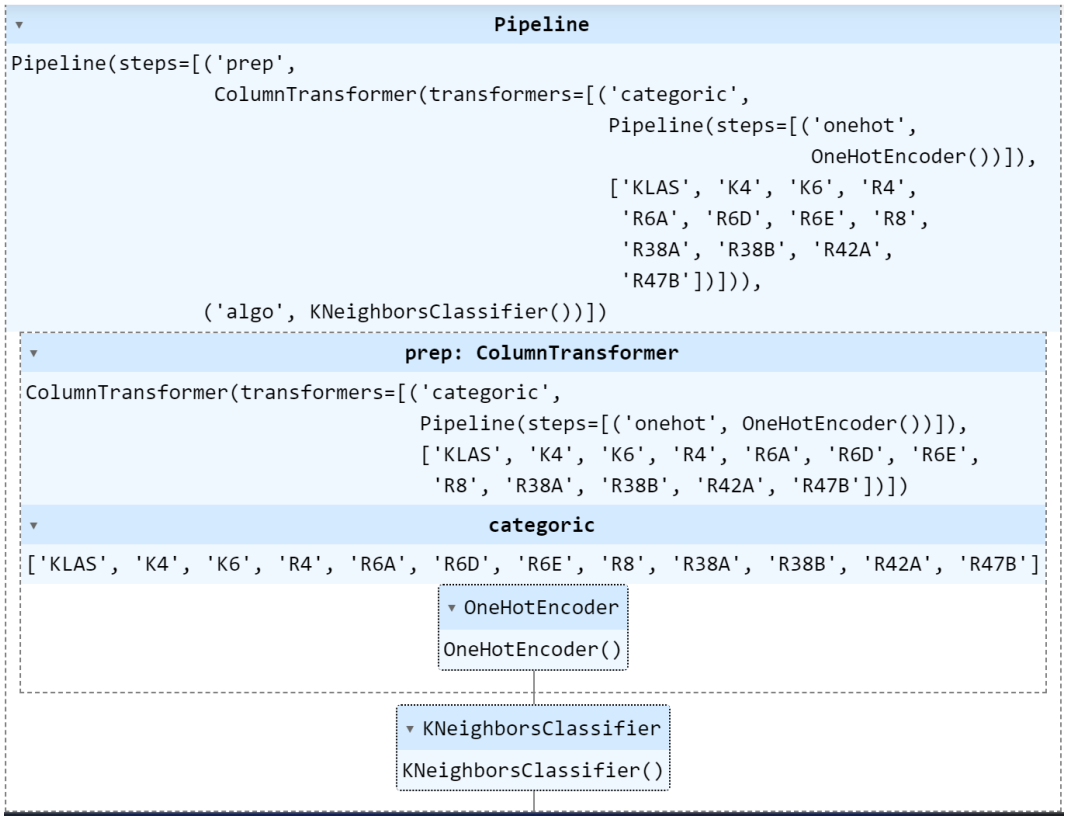
Untuk meningkatkan akurasi model, tahapan selanjutnya yaitu *Dataset Splitting*. Dataset dipisah menjadi data training dan data testing. Pada penelitian kali ini pembagiannya adalah sebanyak 80% training dan 20% testing. Jumlah data training setelah dataset splitting adalah 18.399 dan jumlah data testing setelah dataset splitting adalah 4600.

```
((18399, 12), (4600, 12), (18399,), (4600,))
```

Setelah *dataset splitting*, dataset dimasukkan ke dalam pipeline untuk mempermudah proses machine learning. Setelah itu melakukan proses Column Transformer yang dimasukkan ke dalam variabel preprocessor dimana isinya adalah kolom:

['KLAS','K4','K6','R4','R6A','R6D','R6E','R8','R38A','R38B','R42A','R47B']

Tahapan selanjutnya adalah memasukkan proses sebelumnya ke dalam sebuah pipeline, dimana pipeline tersebut berisi tahapan sebelumnya yaitu preprocessor dan algoritma yang akan dipakai yaitu *KNeighborsClassifier.*



Gambar 4. 10 Dataset Splitting dan Column Transformer KNN

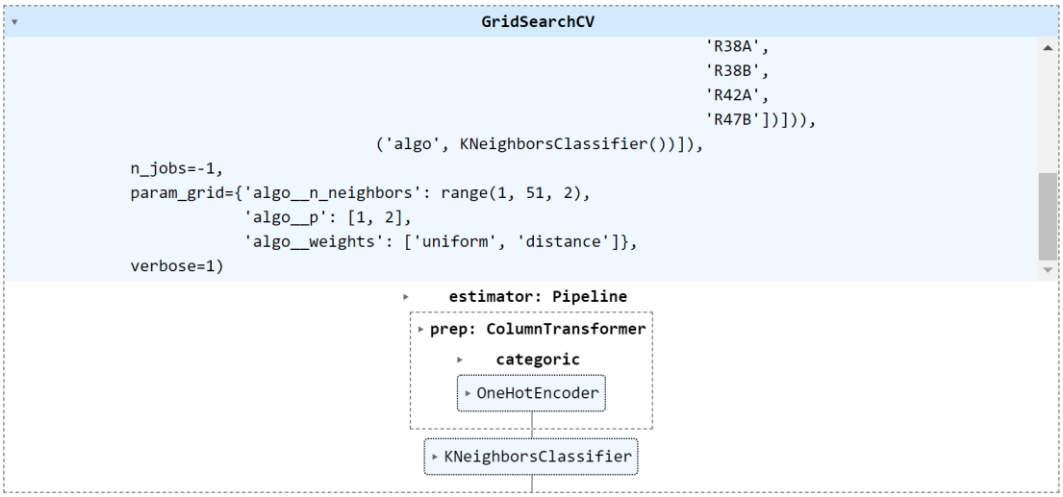
Dari ujicoba ini didapat hasil score sebesar 0.80.

Langkah selanjutnya untuk meningkatkan hasil score dari model adalah menggunakan *Grid Search.* Grid search adalah suatu teknik yang digunakan dalam peningkatan model machine learning untuk mencari kombinasi hyperparameter terbaik dari suatu model. Hyperparameter adalah parameter-parameter yang tidak dipelajari oleh model selama pelatihan, dan mereka perlu diatur sebelum proses pelatihan dimulai. Hyperparameter yang digunakan pada penelitian ini yaitu :

1. n_neighbors: Jumlah tetangga terdekat yang akan dipertimbangkan. (jumlah tetangga merupakan angka ganjil dari 1 sampai 49)
2. weights: Menentukan cara memberikan bobot pada tetangga (uniform atau distance-based).

3. p: Parameter untuk jenis jarak (1 untuk Manhattan distance, 2 untuk Euclidean distance).

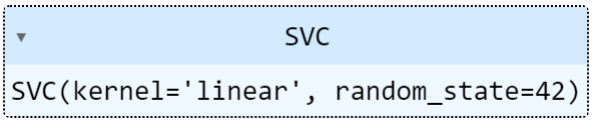
Selain parameter diatas, pada penelitian ini juga mengatur jumlah *cross validation* sebanyak 3, 4, 5, 6 fold.



Gambar 4. 11 GridSearch dan Cross Validation KNN

4.2.2.3 Proses Metode *Support Vector Machine* (SVM)

Tahapan awal proses *Training Model* pada metode klasifikasi SVM yaitu menentukan kernelnya dan menentukan nilai C nya. Nilai C yaitu parameter penalti yang mengontrol *trade-off* antara menciptakan margin yang sebesar mungkin dan mengizinkan beberapa titik pelatihan melanggar margin. Nilai C yang lebih tinggi memberikan penalti yang lebih besar terhadap pelanggaran margin. Dan besaran C untuk penelitian ini adalah 1. Kernel untuk penelitian ini menggunakan kernel linear. Kernel linear mengimplikasikan bahwa model SVM akan mencoba memisahkan kelas dengan sebuah hyperplane linear. Seperti pada gambar



Gambar 4. 12 Kernel Linear dan Nilai C = 1

Dari ujicoba diatas didapat score 0,79.

4.2.2.4 *Tuning* Parameter Menentukan nilai terbaik

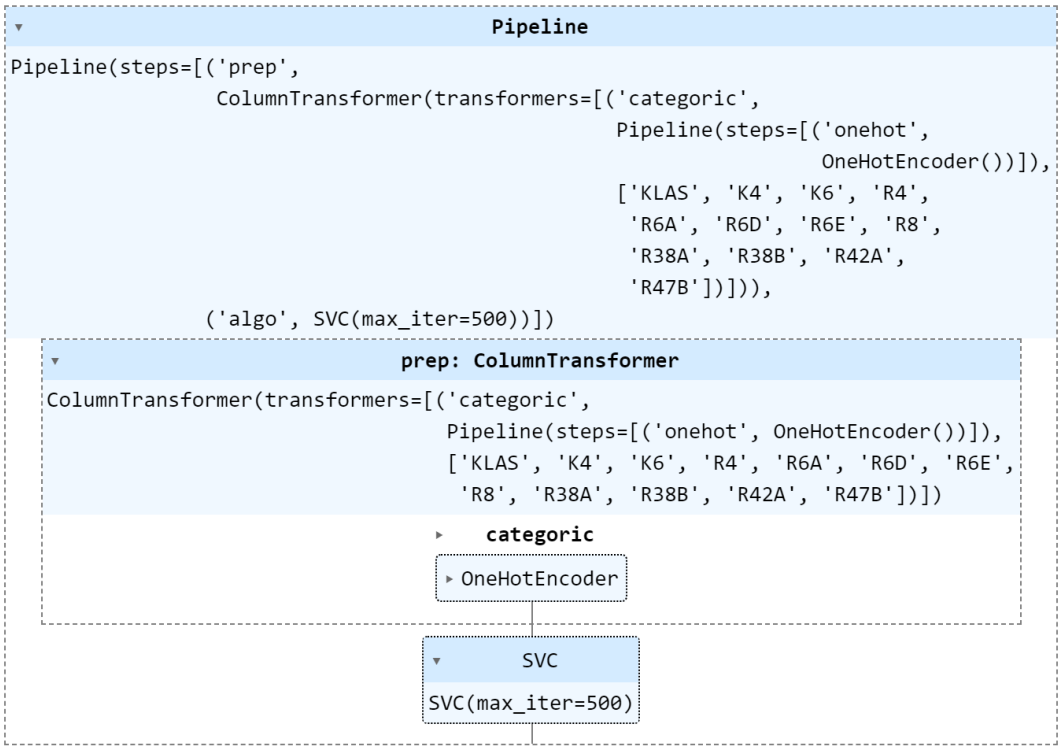
Untuk meningkatkan akurasi model, tahapan selanjutnya yaitu *Dataset Splitting*. Dataset dipisah menjadi data training dan data testing. Pada penelitian kali ini pembagiannya adalah sebanyak 80% training dan 20% testing. Jumlah data training setelah dataset splitting adalah 18.399 dan jumlah data testing setelah dataset splitting adalah 4.600.

```
((18399, 12), (4600, 12), (18399,), (4600,))
```

Setelah *dataset splitting*, dataset dimasukkan ke dalam pipeline untuk mempermudah proses machine learning. Setelah itu melakukan proses Column Transformer yang dimasukkan ke dalam variabel *preprocessor* dimana isinya adalah kolom:

['KLAS','K4','K6','R4','R6A','R6D','R6E','R8','R38A','R38B','R42A','R47B']

Tahapan selanjutnya adalah memasukkan proses sebelumnya ke dalam sebuah pipeline, dimana pipeline tersebut berisi tahapan sebelumnya yaitu *preprocessor* dan algoritma yang akan dipakai yaitu *SupportVectorClassifier*.



Gambar 4. 13 Dataset Splitting dan Column Transformer SVM

Dari ujicoba ini didapat hasil score sebesar 0,54.

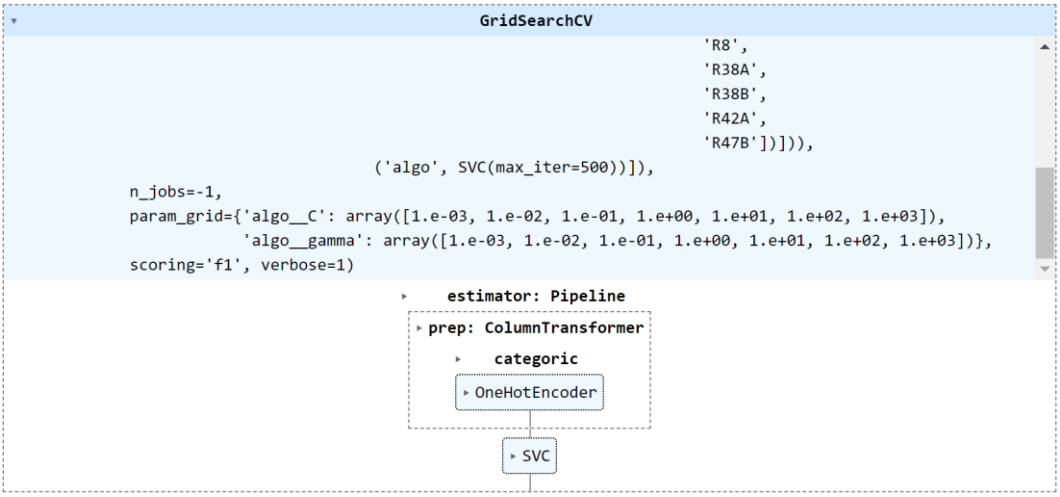
Langkah selanjutnya untuk meningkatkan hasil score dari model adalah menggunakan *Grid Search. Grid search* adalah teknik yang digunakan dalam peningkatan model *machine learning* untuk mencari kombinasi hyperparameter terbaik dari suatu model. Hyperparameter adalah parameter-parameter yang tidak dipelajari oleh model selama pelatihan, dan mereka perlu diatur sebelum proses pelatihan dimulai. Hyperparameter yang digunakan pada penelitian ini yaitu :

1. 'algo__C': Ini menunjukkan parameter C dari model SVM.

   Nilai-nilai yang dijelajahi adalah sejumlah nilai yang berada dalam rentang 10^{-3} sampai 10^{3}.

2. 'algo__gamma': Ini menunjukkan parameter gamma dari model SVM. Nilai-nilai yang dijelajahi adalah sejumlah nilai yang berada dalam rentang 10^{-3} sampai 10^{3}.

Selain parameter diatas, pada penelitian ini juga mengatur jumlah *cross validation* sebanyak 3, 4, 5, 6, 7 fold.



Gambar 4. 14 Grid Search dan Cross Validation SVM

4.2.4 Evaluasi Model

Berikut ini adalah hasil dari pengujian antara algoritma *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM):

Tabel 4. 3 Hasil Pengujian antara KNN dan SVM

No	Algoritma	Pengujian Pertama	Dataset Splitting	GridSearchCV
1	KNN	81%	80%	**87%**
2	SVM	79%	54%	84%

Dari hasil pengujian pada penelitian ini dapat dilihat bahwa akurasi menurun saat dilakukan dataset splitting, lalu meningkat disaat ditambah perlakuan *GridSearch* dengan mengkombinasikan pula *Cross Validation*.

Algoritma *K-Nearest Neighbors* selalu lebih besar hasil akurasi nya dibandingkan dengan algoritma *Support Vector Machine* baik itu pada pengujian pertama, pengujuian dengan dilakukan dataset splitting, maupun setelah ditambahkan perlakuan *Grid Search* yang dikombinasikan dengan *Cross Validation* yaitu diangka 87% dimana *Support Vector Machine* berada di angka 84%, 3 angka dibawah algoritma *K-Nearest Neighbors.*

Pada Penelitian ini juga mengkombinasikan jumlah Cross Validation. Pada Algoritma KNN mengkombinasikan Cross Validation sebanyak 3, 4, 5, 6, sedangkan pada Algoritma SVM mengkombinasikan Cross Validation sebanyak 3, 4, 5, 6, 7.

Tabel 4. 4 Hasil Pengujian Antara KNN dan SVM dengan Kombinasi Jumlah CV

CV	**SVM**	**KNN**
3	0.8118673766338674	0.**87**41325189877114
4	0.**84**19987511926093	0.**87**43998525925529
5	0.8175338074583898	0.**87**42844104781478
6	0.**84**44999974544306	0.**87**46539638603877
7	0.8144522119799884	

4.3 *Eksploratory Data Analysis* (EDA)

Pada bab ini, penelitian akan menjelaskan proses eksplorasi data (EDA) yang dilakukan untuk memahami karakteristik dan pola data yang digunakan dalam penelitian. EDA merupakan tahapan kritis dalam siklus analisis data, yang bertujuan untuk merinci struktur, distribusi, dan relasi antar variabel dalam dataset. Dengan merinci informasi ini, penelitian dapat memperoleh pemahaman yang lebih mendalam tentang data, memastikan kualitas data yang diolah, dan menemukan wawasan yang mungkin berguna dalam pembangunan model prediksi pengangguran. Analisis ini mencakup visualisasi data, statistik deskriptif, dan eksplorasi hubungan antar variabel. Pemahaman mendalam terhadap data dapat membantu memvalidasi asumsi, mengidentifikasi outlier, dan mengeksplorasi pola yang mungkin relevan dalam konteks penelitian ini.

Penelitian akan membahas distribusi variabel kunci yang terkait dengan pengangguran di Provinsi Lampung. Selanjutnya, penjelasan mengenai tren dan pola-pola khusus akan disajikan untuk membantu membimbing pemilihan variabel yang signifikan untuk dimasukkan dalam model prediksi. Melalui eksplorasi ini, diharapkan penelitian dapat menggambarkan dengan jelas konteks data yang digunakan dan memberikan dasar yang kokoh untuk interpretasi hasil yang diperoleh dari model *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM) yang akan dikembangkan pada bab-bab selanjutnya.

Langkah-langkah EDA yang dilakukan dalam penelitian ini akan diuraikan secara rinci, termasuk jenis visualisasi yang digunakan dan analisis statistik yang diterapkan. Proses ini menjadi landasan untuk pengambilan keputusan yang