BAB 3

METODE PENELITIAN

3.1 Data

Data yang digunakan dalam penelitian AES diambil dari hasil jawaban esai pelajaran Bahasa Indonesia pada semester 2 tahun 2020 di Universitas Muhammadiyah Kalimantan Timur yang diteliti oleh Verdhika (2021), dalam penelitiannya ini menghasilkan 1648 baris dan memiliki 3 kolom yang berisikan nilai, kelas, dan jawaban esai. Nilai dari teks jawaban memiliki range “0” sampai “12” dari jumlah kelas “A” sampai “H” dengan keseluruhan memperoleh data sebanyak 1648 baris.

```
nilai kelas jawaban
8,A, 4 faktor yg sebab sebab bahasa melayu angkat j...
8,A, bahasa melayu angkat jadi bahasa satu indonesi...
8,A, empat faktor sebab bahasa melayu angkat jadi b...
8,A, alas bahasa melayu pilih jadi bahasa indonesia...
6,A, bahasa indonesia tumbuh kembang bahasa melayu ...
      ...         ...         ...
6,H, bahasa rupa salah satu unsur identitas suatu b...
4,H, cakup jumlah bahasa saling mirip tutur wilayah...
4,H, empat faktor bahasa melayu angkat jadi bahasa ...
4,H, memang banyak guna bagi besar masyarakat indon...
4,H, bahasa melayu ini akhir jadi bahasa indonesia ...
```

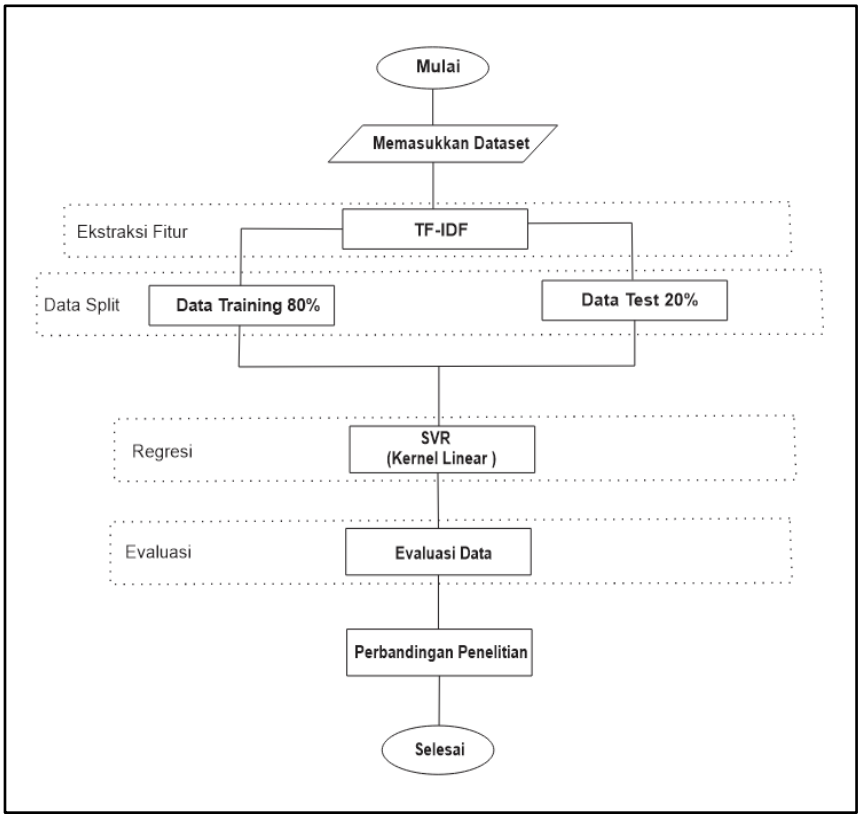
Gambar 3.1 Memasukan Data

Data diatas sebelumnya sudah melalui beberapa tahapan-tahapan pra-proses. Tujuannya agar data lebih mudah diproses dengan menghilangkan beberapa kata imbuhan dan tanda simbol pada jawaban, sehingga pada saat memproses data ke dalam ekstraksi fitur, pembobotannya dapat lebih baik.

3.2 Tahapan Penelitian

Tahap penelitian ini akan mencari nilai RMSE menggunakan metode SVR dengan *kernel liniear* dengan data jawaban siswa sebanyak 1648 dan membandingkan hasil tersebut dengan penelitian sebelumnya yang dilakukan oleh Verdikha (2021), yang mana dalam penelitian Verdikha menggunakan SVR dengan *kernel* RBF.

Adapun *flowchart* yang dibuat untuk mempermudah berjalannya konsep penelitian ini bisa dilihat pada gambar 3.2 dibawah.



Gambar 3.2 Tahapan Penelitian

Gambar 3.2 di atas menjelaskan tahapan-tahapan untuk mendapatkan nilai evaluasi RMSE agar dapat melakukan perbandingan nilai SVR *kernel liniear* dengan SVR *kernel* RBF, adapun bahasa pemrograman yang digunakan pada penelitian ini adalah *python* dengan versi 3.8.3 dan *tools* yang digunakan adalah *jupyter notebook* versi 6.4.5 dan modul *library* yang digunakan untuk melakukan *regresi* dan evaluasi pada penelitian (Barupal & Fiehn, 2019) adalah *sklearn* versi 1.1.1.

Adapun penjelasan pada tahapan penelitian yang dilakukan pada tabel diatas adalah sebagai berikut:

3.2.1 Memasukan Data

Melakukan penginputan *dataset* (data_penelitian.csv) kedalam aplikasi *Anaconda* menggunakan *Jupyter* dan memakai bahasa pemrograman *python* versi 3, adapun data-data yang di gunakan telah melewati tahap-tahapan normalisasi.

Untuk melakukan penginputan *dataset* disini menggunakan objek ‘df1’, objek inilah yang akan dipanggil untuk menampilkan isi dari *dataset* yang telah diinput ke dalam objek ‘df1’, adapun cara untuk menampilkan objek dari *dataset* menggunakan kode bisa di lihat pada **Lampiran 1**, untuk bagian ini *library* yang digunakan adalah *modul pandas*, nilai dari masing-masing variabel akan masuk secara otomatis dengan menggunakan fungsi *iloc*, *iloc* sendiri diambil dari *library pandas* yang berfungsi melakukan index data bilangan bulat berdasarkan lokasi pemilihan kolom.

Adapun variabel yang ada pada **Lampiran 2** yang mana variabel tersebut digunakan untuk menyimpan data atribut teks “jawaban” dan “nilai”, yang mana dijelaskan disini untuk data teks jawaban itu adalah ‘X’ dan data untuk nilai adalah ‘y’.

3.2.2 Ekstraksi Fitur

Pada tahapan Ekstraksi fitur TF-IDF menggunakan *sklearn*, peneliti melakukan pemberian nilai pada sebuah dokumen lalu mengubah data-data yang awalnya adalah *dataset* menjadi bentuk yang dapat dipahami oleh model pembelajaran menggunakkan ekstraksi fitur TF-IDF dengan modul *sklearn* dan kelas *TfidfVectorizer*. *source code* yang digunakan untuk Ekstraksi data TF-IDF dapat dilihat pada **Lampiran 4** dan berikut penjelasannya. Pada tahapan ini peneliti menggunakan modul *sklearn.feature_extraction.text* untuk dapat mengatasi normalisasi data fitur ekstraksi (Barupal & Fiehn, 2019). Kemudian data teks di *import* menggunakan fungsi *tfidfvectorizer* menggunakan variabel ‘X’ yang mengandung *term*. Nilai dari variabel ‘X’ dan ‘y’ disimpan kembali kedalam *tfidfvectorizer* agar dapat menentukan nilai frekuensinya.

3.2.3 *Data split*

Pada bagian *data split* disini tujuannya untuk mengubah sebuah data menjadi *data training* dan *data test*. *Data training* digunakan untuk menjalankan fungsi dari algoritma ML dan *dataset* digunakan untuk melihat keakuratannya. Pada penelitian ini *data split* dibagi menjadi 3 (tiga) tahapan pengujian, contohnya dapat dilihat pada tabel 3.1 dibawah ini.

Tabel 3.1 Pembagian *Data training* dan *Data test*

Data training	*Data test*
80	**20**
70	**30**
60	**40**

Tabel 3.1 diatas menjelaskan bahwa *data training* dan *data test* dibagi menjadi 3 (tiga) bagian, yang pertama 80:20, kedua 70:30 dan yang terakhir adalah 60:40 yang mana data-data tersebut selanjutnya akan di*regresi* menggunakan metode SVR dengan *kernel liniear*. Terdapat beberapa parameter dan variabel di dalam *data split*, contohnya:

1) ***X_train***, berguna untuk menampung data *source* yang akan dilatih.
2) ***X_test***, untuk menampung data target yang akan dilatih.
3) ***y_train***, menampung data *source* yang akan diuji.
4) ***y_test***, menampung data target yang akan diuji.
5) ***X_reshape*** dan *y*, nama variabel saat mendefinisikan data sumber dan data target.
6) ***Parameter test_size***, mendefinisikan ukuran *data testing* atau rasio.
7) ***Parameter random_state***, menginisialisasi generator nomor acak yang memutuskan pada *data split* menjadi *train* dan *test*.

Untuk penelitian ini, menggunakan nilai *random_state* dengan angka 42 yang mana nilai tersebut sama seperti nilai yang ada pada penelitian sebelumnya yang dilakukan oleh Verdikha (2021), yang mana tujuannya sendiri adalah untuk melakukan perbandingan nilai evaluasi pada parameter *kernel*.

Adapun tahapan implementasi pada *data split* yang dapat dilihat pada Lampiran 5, 6 dan 7 yang telah berbentuk *source code*.

3.2.4 *Regresi* dan Evaluasi

Pada tahapan *regresi* dan evaluasi merupakan pemodelan data dan meminimalkan *error* atau selisih antara nilai prediksi dengan nilai sebenarnya dan mengevaluasi performa algoritma yang digunakan, pada tahapan ini algoritma yang digunakan peneliti adalah SVR dan RMSE sebagai pengevaluasinya dan menggunakan *kernel liniear* sebagai tahapan *regresi* yang mana SVR dengan *kernel liniear* ini akan di bandingkan hasilnya dengan penelitian sebelumnya yang diteliti oleh (Verdikha et al., 2021) yang menggunakan SVR dengan *kernel* RBF (*Radial Basis Function*). untuk mengetahui tahapan akurasi penggunaan metode svr dengan parameter *kernel liniear* bisa di lihat pada **Lampiran 6**, untuk metode *regresi* sendiri terdapat beberapa parameter selain parameter *kernel*, contohnya seperti yang ada pada tabel 3.1 berikut.

Tabel 3.2 Parameter Pada Metode SVR

Metode	Parameter
SVR	class sklearn.svm.SVR(*, *kernel*='rbf', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=-1)

Metode yang digunakan penelitian ini sama seperti penelitian terdahulu yang dilakukan oleh (Verdikha et al., 2021), yang mana disini menggunakan SVR dengan *kernel* RBF dan penelitian ini menggunakan uji coba menggunakan metode SVR dengan *kernel* yang berbeda yaitu *kernel linear*. Dalam metode *regresi* terdapat jenis-jenis parameter selain parameter *kernel*, diantaranya adalah *Kernel*, *degree, gamma, coef0, C, epsilon, shrinking, cache_size, verbose,* dan *max_iter* dengan penjelasan masing-masing sebagai berikut:

1) ***Kernel***, berfungsi melakukan pemetaan data yang mengatasi bilangan *non-linear*. setiap jenis *kernel* menggunakan sistem yang berbeda. *Kernel rbf* merupakan *kernel default*. Terdapat berbagai jenis *kernel* diantaranya *rbf, linear, poly, sigmoid.*

2) ***Degree***, sangat cocok dengan *kernel polynomial*, namun diabaikan oleh semua *kernel*. bilangan yang digunakan adalah *integer*, dengan nilai *default* = 3.
3) ***Gamma***, nilai dalam parameter *gamma* ada dua yaitu *scale* dan *auto*. jika nilai *gamma* adalah *scale*, maka menggunakan '1/ (n_features * X.var ())' sebagai nilai *gamma*. Jika nilai *gamma* adalah *auto*, maka '1/n_features'.
4) ***Coef0***, fungsi ini signifikan terhadap *kernel poly* dan *sigmoid*. parameter ini independen atau berdiri sendiri. menggunakan nilai *default* yaitu 0,0.
5) ***Tol***, toleransi dalam pemberhentian kriteria. menggunakan nilai *default* yaitu 0,001.
6) ***Parameter C***, adalah regulasi yang berfungsi berbanding terbalik dengan *C*. harus benar-benar positif. hukumnya *penalty* kuadrat l2. menggunakan nilai *default* 1,0.
7) ***Epsilon***, parameter ini menentukan tabung *epsilon* di mana tidak ada *penalty* yang terkait dalam fungsi dengan poin prediksi dalam jarak *epsilon* dengan nilai aktual. nilai default adalah 0,1.
8) ***Cache_size***, menentukan ukuran *cache kernel* dalam MB (Megabyte). defaultnya 200 MB.
9) ***Verbose***, dengan dua jenis nilai yaitu TRUE atau FALSE. sedangkan untuk defaultnya adalah FALSE, FALSE adalah aktifan keluaran *verbose*. pengaturan ini perlu diperhatikan karena memanfaatkan pengaturan *runtime* atau yang sedang berjalan saat itu. jika diaktifkan, mungkin parameter ini tidak berfungsi dalam mengatasi masalah *multithread* atau penanganan masalah secara bersamaan (ganda).
10) ***Max_iter***, Nilai default bilangan dari parameter ini adalah '-1'. Batas keras pada iterasi dalam pemecah, atau -1 tanpa batas.

Dalam menentukan nilai evaluasi RMSE, Modul yang digunakan dalam penelitian ini adalah *library sickit-learn* yaitu *classification. Metrics import mean_square_error* pada kasus prediksi nilai kesalahan (Barupal & Fiehn, 2019).

Pada penelitian ini tahapan evaluasi dilakukan kepada ketiga hasil *data test* dari *data split* yang telah di*regresi*, yang pertama *data test* 20%, kedua *data test* 30% dan yang terakhir adalah *data test* 40%, kegunaan dari evaluasi RMSE sendiri adalah untuk mengetahui tingkat kesalahan dalam penggunaan metode SVR dengan *kernel liniear*.