BAB 3

METODOLOGI

3.1 Data

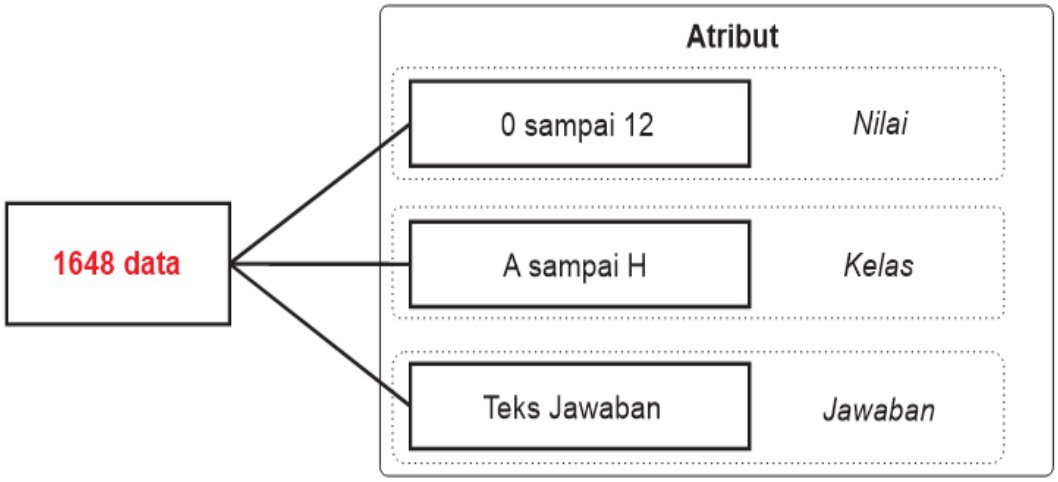
Data menggunakan dataset yang sudah di *preprocessing* pada penelitian (Thamrin, 2021). Data tersebut berisi jawaban nilai UTS MKDU bahasa indonesia di laksanakan di UMKT pada semester 2 tahun 2020. Terdapat tiga kolom variabel data yang terdiri dari data nilai, kelas, dan jawaban. Nilai dari teks jawaban memiliki range "0" sampai "12" dari jumlah kelas "A" sampai "H" dengan keseluruhan memperoleh data sebanyak 1648 baris. Data disimpan ke dalam file (data_penelitian.csv) dengan format CSV (*Comma Separated Values*). Jika di buka menggunakan aplikasi Microsoft Excel maka tampilannya sebagai berikut.

nilai,kelas,jawaban
8,A,4 faktor yg sebab sebab bahasa melayu angkat jadi bahasa indonesia bah
8,A,bahasa melayu angkat jadi bahasa satu indonesia 28 oktober 1928 peristi
8,A,empat faktor sebab bahasa melayu angkat jadi bahasa indonesia 1 bahasa
8,A,alas bahasa melayu pilih jadi bahasa indonesia 1 bahasa melayu rupa ling
6,A,bahasa indonesia tumbuh kembang bahasa melayu sejak dulu guna bahas
6,A,alas bahasa melayu jadi lingua franca bahasa antar antar wilayah indones
8,A,bahasa melayu jadi bahasa nasional indonesia karena 1 bahasa melayu ru
2,A,warna bahasa melayu rupa bahasa paling mudah simpel cepat erti
10,A,1 bahasa melayu rupa lingua franca indonesia bahasa hubung bahasa da
2,A,bahasa melayu pilih bahasa indonesia bahasa sudah jadi lingua franca bah
6,A,alas bahasa melayu jadi bahasa indonesia 1 bahasa melayu fungsi lingua f
2,A,bahasa melayu milik kosa kata yg mudah difahami guna bahasa daerah su
10,A,bahasa indonesia rupa varian bahasa melayu guna indonesia bunyi daru
10,A,historis buat bahasa melayu angkat jadi bahasa indonesia bukan kardan

Gambar 3. 1 Tampilan dataset menggunakan Aplikasi Excel

Data pada Gambar 3.1 adalah dataset yang sudah dilakukan *preprocessing*, tujuannya agar data lebih optimal dan mudah diproses dengan menghilangkan kata-kata imbuhan dan tanda-tanda simbol pada jawaban sehingga dalam proses ekstraksi fitur tingkat normalisasi pembobotan menjadi lebih baik. Data yang

digunakan juga memiliki struktur di dalam dataset penelitian, stuktur dari dataset yang digunakan dapat diilustrasikan pada Gambar 3.2 di bawah.

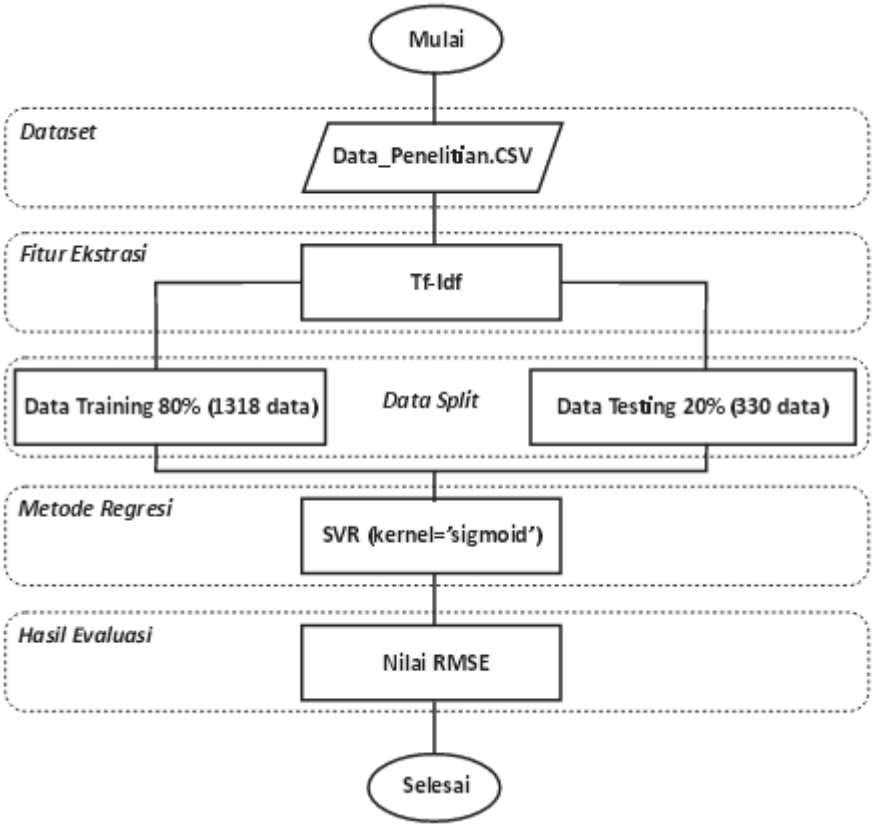


Gambar 3.2 Tampilan struktur dataset

Gambar di atas adalah struktur data dari Gambar 3.1 , peneliti akan lebih mudah dalam membedakan data dari setiap atribut masing-masing di dalam dataset.

3.2 Tahapan Penelitian

Dalam tahapan penelitian, metode yang digunakan akan menentukan nilai evaluasi RMSE dari Dataset (data_penelitian.csv). Untuk mempermudah dalam mengkonsep alur penelitian, maka dibuat sebuah model struktur *flowchart*. Tahapan di desain menggunakan *flowchart* bertujuan memudahkan dalam melakukan konsep proses penelitian berdasarkan alur pemrosesan dataset ke dalam pemrograman python yang dapat dilihat pada Gambar 3.3 di bawah.



Gambar 3.3 Tahapan pemrosesan data penelitian

Struktur tahapan di atas menjelaskan proses untuk mendapatkan nilai evaluasi RMSE untuk mendapatkan hasil perbandingan parameter SVR *kernel sigmoid* dengan *kernel* RBF. Penjelasan lebih lanjut tahapan-tahapan di atas sebagai berikut.

3.2.1 Memasukan Dataset

Dataset yang digunakan pada penelitian ini akan dimasukan dan dibuka menggunakan aplikasi Anaconda kemudian menggunakan *tools* Jupyter Notebook. Jupyter Notebook merupakan aplikasi bahasa pemrograman python (v 3.6) yang *include* di dalam aplikasi Anaconda.

Dataset akan di kemas di dalam variabel bernama 'df1', objek ini yang akan dipanggil untuk menampilkan isi dari dataset yang digunakan didalam aplikasi Anaconda. Untuk menampilkan variabel tersebut menggunakan kode yang terdapat pada **Lampiran 1**. Pada bagian ini *library* yang digunakan adalah modul *pandas.* Nilai dari masing-masing variabel akan otomatis masuk dengan

menggunakan fungsi *iloc*. *Iloc* yang di ambil pada *library pandas* berfungsi untuk melakukan *index* data bilangan bulat berdasarkan lokasi pemilihan posisi kolom.

proses ini juga dibuat variabel untuk menyimpan data atribut teks "jawaban" dan "nilai", dimana data teks jawaban dilambangkan sebagai variabel 'X' dan data nilai sebagai variabel 'y'. Untuk menampilkan variabel tersebut menggunakan kode **Lampiran 2.**

3.2.2 Ekstraksi Data Menggunakan TF-IDF

Pada bagian ekstraksi data, metode ekstraksi yang digunakan adalah TF-IDF, sehingga menampilkan hasil ekstraksi fitur dan elemen pada dataset. Kode dapat dilihat pada **lampiran 4**. Pada tahap ini menggunakan *modul sklearn.feature_extraction.text* untuk mengatasi masalah normalisasi data fitur hasil ekstraksi (Pedregosa et al., 2011). Data teks di *import* menggunakan fungsi *tfidfvectorizer* kemudian nilai dari setiap variabel 'X' yang mengandung *term*. Nilai variabel 'X' dan 'y' kemudian dikemas kembali kedalam variabel 'tfidf' menggunakan fungsi *tfidfvectorizer* untuk menentukan nilai frekuensinya.

Kosa kata dari nilai TF-IDF kemudian akan dipelajari menggunakan fungsi 'nama variabel.fit()' ke dalam variabel baru yaitu 'X_tfidf'. Selanjutnya 'X_tfidf' akan ditransformasikan ke dalam matrik representatif TF atau IDF. Hasilnya disimpan kedalam format *row*. Format *row* hanya menyimpan data selain 0, berarti nilai 0 yang memiliki nilai koma masih dapat dimasukan ke dalam data row (Maulana & Fitriyani, 2017).

3.2.3 Data *Split*

Kode untuk data *split* dapat dilihat pada **lampiran 5**. Pada tahapan ini *libary* yang digunakan adalah modul *sklearn.model_selection* dan *collections* (Pedregosa et al., 2011). untuk fungsi masing-masing *library* yang digunakan adalah modul *train_test_split* dan *Counter*. Untuk menggunakan fungsi pada bagian ini, dibuatlah variabel secara berturut yaitu 'X_train', 'X_test', 'y_train', 'y_test'. Variabel tersebut yang nantinya akan dihitung menggunakan fungsi *train_test_split*.

X_train difungsikan menampung data *source* yang akan dilatih. X_test difungsikan menampung data *target* yang akan dilatih. Sedangkan y_train difungsikan untuk data *source* sebagai testing dan y_test sebagai data *target* untuk testing.

Variabel X dan y adalah nama untuk mendefinisikan data *source* dan *target*. Parameter *test_size* definisi dari ukuran data testing. Dalam lampiran kode dijelaskan parameter "*test_size=0.2"* yang berarti data yang digunakan sebagai data testing adalah sebesar 20% dari dataset.

Pada metode pembagian data pada modul *train_test_split* masih membagi data secara *random* atau acak. jika kode program dijalankan ulang, maka nilai yang akan dimunculkan berbeda. Maka untuk mengatasinya digunakan parameter *random_state*. *random_state* di isi dengan nilai *integer* berapa saja, di mana fungsi angka *integer* yang dipilih digunakan untuk mengunci data acak pada angka *integer* tersebut.

Pada penelitian ini digunakan nilai *random_state* yaitu angka 42, di mana pada penelitian Verdikha (2021) juga menggunakan angka tersebut untuk sebagai tujuan perbandingan nilai evaluasi pada parameter *kernel* yang digunakan.

3.2.4 Regresi SVR *kernel sigmoid* dan *rbf*

Pada metode SVR, terdapat beberapa parameter *kernel* yang dapat digunakan untuk mengatasi bilangan *non-linear*. diantaranya *kernel linear, poly, rbf,* dan *sigmoid.* Pada penelitian ini menggunakan metode dengan *kernel sigmoid* dan *rbf*. Isi parameter pada metode SVR dapat dilihat pada Tabel 3.1 di bawah.

Tabel 3.1 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*)

Skema parameter yang digunakan sama dengan yang dipakai penelitian Verdikha (2021), Pada bagian penggunaan *kernel* secara *default* adalah RBF atau yang dipakai pada penelitian terkait, sedangkan pada penelitian ini melakukan uji coba menggunakan *kernel* berbeda yaitu *sigmoid*. Dalam metode regresi terdapat jenis-jenis parameter selain parameter *kernel*, diantaranya *degree, gamma, coef0, C, epsilon, shrinking, cache_size, verbose, dan max_iter* dengan penjelasan masing-masing sebagai berikut.

3.2.4.1 *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 diataranya rbf, linear, poly, sigmoid.*

3.2.4.2 *Degree*

Sangat cocok dengan *kernel polynomial*, namun diabaikan oleh semua kernel. bilangan yang digunakan adalah *integer*, dengan nilai *default* = 3.

3.2.4.3 *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'.

3.2.4.4 *Coef0*

Fungsi ini signifikan terhadap kernel *poly* dan *sigmoid*. parameter ini independen atau berdiri sendiri. menggunakan nilai *default* yaitu 0,0.

3.2.4.5 *Tol*

Toleransi dalam pemberhentian kriteria. menggunakan nilai *default* yaitu 0,001.

3.2.4.6 parameter *C*

Parameter *C* adalah regulasi, fungsinya berbanding terbalik dengan *C*. harus benar-benar positif. hukumnya *penalty* kuadrat l2. menggunakan nilai *default* 1,0.

3.2.4.7 *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.

3.2.4.8 *Shrinking*

Ada dua jenis nilai, TRUE atau FALSE, untuk nilai yang digunakan adalah TRUE.

3.2.4.9 *Cache_Size*

Menentukan ukuran *cache kernel* dalam MB (Megabyte). defaultnya 200 MB

3.2.4.10 *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).

3.2.4.11 *Max_Iter*

Nilai default bilangan dari parameter ini adalah '-1'. Batas keras pada iterasi dalam pemecah, atau -1 tanpa batas.

Kode yang digunakan dalam penerapan metode SVR dengan parameter penjelasan di atas dapat dilihat pada **lampiran 6**. Penulis menggunakan modul *sklearn.svm.SVR* untuk mengatasi masalah regresi (Pedregosa et al., 2011)

Variabel baru dibuat dengan nama 'clf2'. Variabel ini dibuat sebagai rumus untuk menghitung data X_train dan y_train. kemudian data training tersebut di pelajari kembali oleh sistem. kemudian dibuat variabel 'y_pred' hasil dari regresi dari variabel X_test. Variabel y_pred disini adalah nilai prediksi. Nilai sebenarnya terdapat pada variabel y_test. Data dari keuda variabel ini yang kemudian digunakan untuk evaluasi Nilai RMSE.

3.2.5 Menentukan Hasil Evaluasi RMSE

Dalam menentukan nilai evaluasi RMSE, Modul yang digunakan adalah *library* sickit-learn yaitu *classification.metrics* import *mean_square_error* pada kasus prediksi nilai kesalahan (Pedregosa et al., 2011). modul ini berfungsi untuk menentukan nilai evaluasi RMSE terhadap metode yang digunakan. Kode yang digunakan untuk menentukan evaluasi dapat dilihat pada **Lampiran 7**.

Di dalam kode terdapat variabel y_pred sebagai nilai prediksi dan y_test sebagai nilai sebenarnya. Terdapat parameter yang berisikan nilai "TRUE" dan "FALSE", jika parameter dengan nilai TRUE, maka nilai prediksi kesalahan yang dihasilkan adalah Mean Square Error (MSE) dan sebaliknya jika nilai parameternya adalah FALSE, maka nilai yang prediksi kesalahan yang dihasilkan adalah RMSE (Verdikha et al., 2021).