BAB II

METODE PENELITIAN

2.1 Objek Penelitian

"Sirekap 2024" adalah aplikasi untuk mendokumentasikan formulir hasil penghitungan suara di TPS dan mengirimkannya ke jenjang selanjutnya[1]. Sirekap pada pemilu 2024 sebagai alat bantu berbasis teknologi yang membantu KPPS dalam menyederhanakan proses penginputan hasil perhitungan suara. Sirekap yang dipakai sebagai sarana publikasi hasil pemilihan dan alat bantu dalam pelaksanaan rekapitulasi suara Pilkada Serentak 2020 telah dipersiapkan oleh KPU RI dalam setahun terakhir (Gauru, Martini dan Alfirdaus, 2022). Proses pengambilan data pada tanggal 6 februari 2024 tepatnya pukul 22.00 WITA, terdapat sekitar delapan ribu data ulasan aplikasi "Sirekap 2024" pada google playstore, ulasan tersebut terdiri dari peringkat satu hingga lima.

2.2 Alat dan Bahan

Dalam penelitian ini Alat yang digunakan meliputi perangkat *hardware* dan *software* yang digunakan, diantaranya :

1. Perangkat keras(*Hardware*)
   - Laptop merk Lenovo
     Dengan spesifikasi sebagai berikut :
   - RAM 8192MB RAM
   - Processor Intel Core i3
   - Sistem operasi windows 10 Pro 64-bit

2. Perangkat Lunak (*Software*)

   Pada perangkat lunak yang digunakan yaitu, Visual Studio Code versi 1.73.0 dan *library* python. *Library* python yang digunakan :

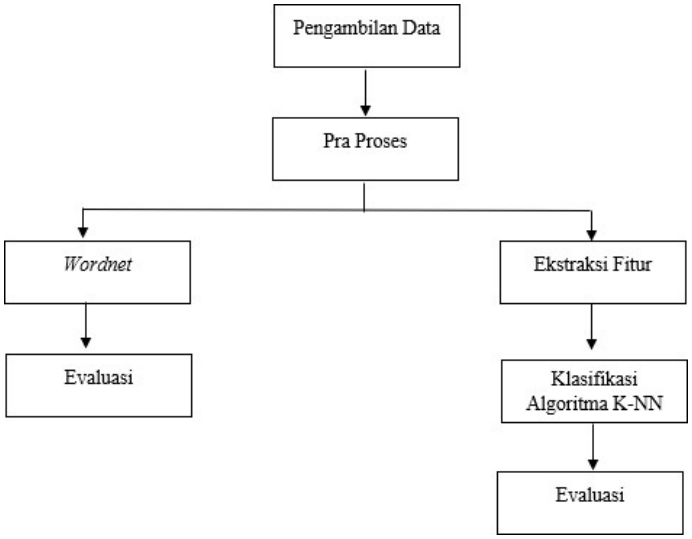
Tabel 2. 1 *Library* Python

***Library* Python**	**Versi**	**Keterangan**
Pandas	1.4.4	Digunakan untuk analisis data
Matplotlib	1.21.5	Digunakan untuk membuat visualisasi data
Numpy	1.21.5	Digunakan untuk mengolah dan memanipulasi data dalam bentuk *array*
Sckitlearn	1.0.2	Pustaka yang dapat digunakan secara gratis untuk pembelajaran mesin dan pemodelan data dalam bahasa pemrograman python.
NLTK(*Natural Language Toolkit*)	3.7	Pustaka yang dirancang untuk digunakan dalam pemrosesan bahasa alami menggunakan python, menyediakan berbagai kumpulan alat untuk analisis teks dan berbagai dataset untuk pengujian.
Textblob	0.18.0	Digunakan untuk mengetahui sebuah *text* bersifat positif, negatif, atau netral dalam analisis sentimen
Google-play-scraper	1.2.6	Digunakan untuk mengakses google play store dan mengekstrak informasi yang diperlukan.
sastrawi	1.0.1	Digunakan untuk memproses kata-kata ke bentuk dasarnya.

[1] https://play.google.com/store/apps/details?id=id.go.kpu.sirekap2024

2.3 Prosedur Penelitian

Penelitian ini menganalisis ulasan pengguna pada aplikasi "Sirekap 2024" menggunakan algoritma *K-Nearest Neighbors* dengan tahapan penelitian yang dapat dilihat pada gambar 2.1



Gambar 2. 1 Diagram Alur Penelitian

Pada gambar alur penelitian diatas, tahapan pertama yang dilakukan adalah pengambilan data yang diambil dari google playstore. Setelah data dikumpulkan kemudian dilakukan tahapan pra proses yang terdiri dari beberapa tahapan. Langkah selanjutnya terbagi menjadi 2 tahapan besar yaitu tahapan *WordNet* dan tahapan klasifikasi K-NN. Kedua tahapan tersebut kemudian dievaluasi menggunakan nilai *F1-Score*.

1. Pengambilan Data

Pada proses pengambilan data, penelitian ini mengambil data sekunder yang diperoleh dari ulasan pengguna "Sirekap 2024". Berikut parameter yang digunakan dalam pengumpulan data :

a. *Add_id*
   Pada proses pengambilan data menggunakan *add_id* dengan nilai parameter "id.go.kpu.sirekap2024" sebagai identifikasi unik atau *ID* yang dimasukkan ke setiap elemen dalam kumpulan data. digunakan untuk tujuan identifikasi, pengelompokkan, atau referensi unik.
b. *Lang*
   Dalam penggunaan *lang* dengan nilai parameternya yaitu "*id*" yang merupakan argumen yang digunakan untuk menentukan bahasa ulasan yang ingin diambil dari Google Playstore.
c. *Country*
   Sedangkan fungsi parameter *country* dengan nilai parameternya yaitu "*id*". Dalam proses pengambilan data, fungsi ini untuk menentukan negara tempat aplikasi tersedia.

d. Sort

Pada parameter *sort* dengan nilai "*Sort.MOST_RELEVAN*" digunakan untuk menentukan metode pengurutan ulasan.

e. Count

Fungsi *count* disini untuk menentukan jumlah ulasan yang ingin diambil. sesuai dengan jumlah yang diinginkan, pada penelitian ini menentukan dengan jumlah 100.000.000.

2. Pra Proses

Pada penelitian ini berikut merupakan tahapan dari pra proses yang digunakan[2] :

a. Add id

Tahap ini menggunakan *library pandas*, '*Add id'* digunakan pada pra proses untuk menambahkan sebuah identifikasi unik atau *ID* ke setiap item dalam kumpulan data. Hal ini dilakukan untuk keperluan identifikasi, pengelompokan, atau referensi yang khas. (*source code* terdapat dalam lampiran 13)

b. Lowercase

Penggunaan *lowercase* merujuk pada konversi atau proses untuk mengubah semua huruf dalam teks menjadi huruf kecil. Ini adalah kebalikan dari *uppercase*, di mana semua huruf dalam teks diubah menjadi huruf besar. Dalam pengolahan teks, yaitu untuk mengkonversi semua huruf menjadi huruf kecil bertujuan untuk memproses data dengan benar dan konsisten. Penelitian ini menggunakan proses *lowercase* untuk mengubah semua huruf menjadi huruf kecil. (*source code* terdapat dalam lampiran 14)

c. Remove Unecessary Character

Proses ini akan menghapus karakter yang tidak diperlukan, dimana proses ini menghilangkan semua karakter dari sebuah teks atau *string* yang bukan huruf atau angka. Proses ini berfungsi untuk membersihkan teks dari simbol, tanda baca, atau karakter spesial lainnya, yang tidak diperlukan dalam analisis atau pemrosesan data. (*source code* terdapat dalam lampiran 15).

d. Spellchecker

Penggunaan *spellchecker* dirancang untuk mendeteksi dan memperbaiki kesalahan pengejaan dalam teks yang dimasukkan pengguna dan mengubah kata yang tidak baku atau kata slang menjadi kata yang sesuai sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Sebelum masuk proses *spellchecker* terlebih dahulu dibuat kamus_tidak_baku yang berlandaskan dari Kamus Besar Indonesia (KBI) (Hasdiana, 2018). Pada prosesnya *file* csv diinput untuk proses pembuatan kamus tidak baku dan langkah pertama dalam membuat kamus tidak_baku yaitu menginput data csv hasil scraping menggunakan *library pandas*, kemudian membersihkan data dari karakter non-alfabet dan memecah teks menjadi kata-kata, lalu menghapus kata-kata duplikat, lalu membersihkan kata yang mengandung karakter numerik dan menghapus kata yang terdiri dari satu hingga tiga huruf. Langkah selanjutnya mencari kata-kata yang tidak terrdapat di list kamus pada kolom data pandas.Tujuan utama *spellchecker* untuk membantu memperbaiki kesalahan pengejaan yang mungkin terjadi karena ketik cepat, kesalahan aturan pengejaan, atau kesalahan ketik dan kata slank. *Spellchecker* akan memeriksa setiap kata dalam teks untuk melihat apakah ada yang tidak sesuai dengan kamus kata yang sudah disediakan. (*source code* terdapat dalam lampiran 17).

[2] https://github.com/gioprana89/scraping-google-play

e. *Stemming*

tahapan ini digunakan untuk menghilangkan atau memotong akhiran pada kata-kata dalam teks dengan tujuan untuk menghasilkan bentuk kata dasar atau kata akar. Hal ini dilakukan untuk mengurangi variasi dalam kata-kata yang muncul dalam teks, sehingga kata-kata yang sebenarnya memiliki makna yang sama dapat dipresentasikan secara konsisten sebagai satu entitas (*source code* terdapat dalam lampiran 18)

3. WordNet

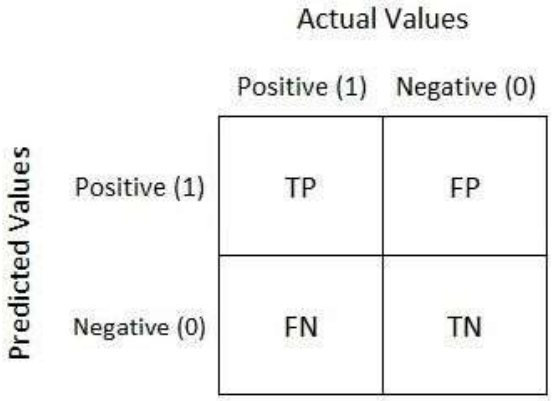
WordNet merupakan pusat data yang memuat istilah-istilah dalam bahasa Inggris beserta keterkaitannya. Istilah ini mencakup kata benda, kata kerja, kata sifat, atau kata keterangan, serta hubungan-hubungan antara istilah tersebut (Siahaan *et al.*, 2023). Dalam metode *WordNet* yang pertama dilakukan dalam penelitian ini yaitu data diterjemah dengan API Google Translate yang kemudian dilakukan perhitungan menggunakan metode *WordNet* dengan *Library Textblob*. Tabel 2.2 terdapat kolom indeks yang merupakan nilai polaritas sentimen dari -1 (sangat negatif) hingga 1 (sangat positif). Dalam tabel ini, polaritas yang lebih besar dari 0,5 dikategorikan sebagai sangat positif dengan peringkat 5. Polaritas antara 0,2 dan 0,5 diberi peringkat 4, menunjukkan sentimen positif. Polaritas netral, yang berkisar antara -0,1 dan 0,1, diberi peringkat 3. Sentimen negatif dengan polaritas antara -0,5 dan -0,2 diberi peringkat 2, sementara polaritas kurang dari -0,5, yang menunjukkan sentimen sangat negatif, diberi peringkat 1. Sehingga pada tahap ini akan menghasilkan label ranking yang terdiri dari ranking 1-5 dari ulasan "Sirekap 2024" (*Source code* terdapat dalam lampiran 22).

Tabel 2. 2 Kriteria Indeks

Indeks	Peringkat
>0,5	5
0,2 – 0,5	4
(-0,1)-0,1	3
(-0,5)-(-0,2)	2
<(-0,5)	1

4. Evaluasi *WordNet*

Pada evaluasi tahapan *WordNet, F1-Score* diperoleh dari nilai yang terdapat di *confusion matrix* yang merupakan pengukuran yang sering digunakan dalam masalah klasifikasi, dimana output dapat terdiri dari dua kelas atau lebih yang memiliki 4 nilai yaitu *true positive* (TP), *false positive* (FP), *true negative* (TN) dan *false negative* (FN) (Istighfarizky *et al.*, 2022). (*Source code* terdapat pada lampiran 24).



Gambar 2. 2 Dasar Confusion Matrix

Pada gambar diatas merupakan dasar *confusion matrix* 2x2, pada penelitian ini menggunakan lebih dari 2 kelas, yang berarti masuk kedalam *confusion matrix multi class.*

	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
Rating 1	TP	FP	FP	FP	FP
Rating 2	FN	TN	TN	TN	TN
Rating 3	FN	TN	TN	TN	TN
Rating 4	FN	TN	TN	TN	TN
Rating 5	FN	TN	TN	TN	TN

Gambar 2. 3 *Confusion matrix 5x5*

Dari gambar 2.3, menunjukkan *Confusion matrix* klasifikasi *multi class* pada sistem evaluasi rating. *Matrix* ini digunakan untuk mengevaluasi kinerja metode klasifikasi yang memprediksi lima tingkat rating (rating 1 hingga rating 5).*Confusion matrix* adalah tabel yang menggambarkan jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji yang diklasifikasikan dengan salah (Normawati dan Prayogi, 2021).

Tabel 2. 3 Penjelasan nilai konfusi matriks

Nilai	Keterangan
True Positive (TP)	Data *Positive* yang diprediksi benar
True Negative (TN)	Data *Negative* yang diprediksi benar
False Positive (FP)	Data *Negative* namun diprediksi sebagai data positif
False Negative (FN)	Data *positve* namun diprediksi sebagai data negatif

Dari keempat nilai tersebut akan menjadi dasar dalam perhitungan yaitu:

a. *Precision*

Precision merupakan proporsi dari *True Positive* (TP) terhadap total prediksi positif. Sehingga *precision* bertujuan untuk mengurangi jumlah *False Positive* (FP). Berikut rumus *precision*:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

b. *Recall*

Recall merupakan proporsi dari *True Positive* (TP) dengan hasil data yang positif . sehingga *recall* bertujuan untuk mengurangi jumlah *False Negative* (FN). Berikut rumus *recall*:

$$\text{Recall} = \frac{TP}{TP+FP} \quad (2)$$

b. *F1-Score*

F1-score merupakan matrik evaluasi yang menggabungkan *precision* dan *recall* menjadi satu nilai. Berikut rumus F1-Score:

$$\text{F1-Score} = \frac{2 \times precision \times recall}{precision+recall} = \frac{2TP}{2TP+FP+FN} \quad (3)$$

c. *Average Macro*

Pada *avarage macro* merujuk untuk menghitung rata-rata dari *presisi,recall dan F1-Score*, berikut cara mencari rata-rata *macro*:

$$MAF = \frac{\sum_{K=1}^{K} F1Score_k}{K} \quad (4)$$

Keterangan:

MAF = *Macro Average F1-Score*

K = Jumlah kelas pada klasifikasi *multiclass*

5. Tahapan Klasifikasi KNN

a. Ekstraksi Fitur TF-IDF

Tahap selanjutnya yaitu ekstraksi fitur yang digunakan pada penelitian ini adalah *Term Frequency –Inverse Document Frequency* (TF-IDF) .TF-IDF merupakan teknik pembobotan kata yang menggabungkan perhitungan nilai *Term Frequency* (TF) dan jumlah kemunculan kata pada seluruh koleksi dokumen (Karo Karo *et al.*, 2023). *Term Frequency* mengukur frekuensi kemunculan sebuah kata dalam dokumen tertentu, dimana semakin sering kata tersebut muncul, semakin besar nilai TF (Amly, Yusra dan Fikry, 2023). Sementara itu, *Inverse Document Frequency* (IDF) mengukur jumlah dokumen yang mengandung kata tersebut dalam seluruh dataset, sehingga semakin jarang kata tersebut muncul, semakin besar nilai IDF-nya (Syahrandi, Latipah and Verdikha, 2023). Hasil dari pembobotan kata adalah perkalian antara nilai TF dan IDF, dimana bobotnya akan lebih kecil jika kata tersebut muncul lebih sering, dan sebaliknya, akan lebih besar jika kata tersebut muncul lebih jarang (Umar, Riadi dan Purwono, 2020). Berikut rumus persamaan TF-IDF: (*Source code* terdapat pada lampiran 27).

$$tfidf(t,d) = tf(t,d) \times idf(t) \quad (5)$$

Keterangan :

tfidf(t,d) = bobot *term*

tf(t,d) = *term* frekuensi kata t pada dokumen d

idf(t) = *Invers* dokumen frekuensi kata t

kemudian utnuk mencari nilai IDF menggunakan persamaan berikut :

$$idf(t) = \log\left(\frac{N+1}{Nt+1}\right) + 1 \quad (6)$$

Keterangan :
t = *term*
N = total keseluruhan dokumen
Nt = total dokumen dengan *term*

Pada ekstraksi fitur TF-IDF pada penelitian ini berikut parameter yang digunakan dari *library* scikit-learn :

a. ngram_range : pada parameter ngram_range digunakan untuk menetapkan rentang nilai n yang digunakan dalam proses ekstraksi fitur. Pengguna dapat menentukan nilai minimum dan maksimum dari n-gram yang diinginkan. Penelitian ini menggunakan nilai (1,1).
b. norm : pada parameter norm proses normalisasi dilakukan dengan cara menghitung jumlah kuadrat dari setiap elemen pada vektor, mengambil akar kuadrat, dan dari hasil penjumlahan tersebut, dan kemudian membagi setiap nilai elemen pada vektor dengan nilai akar kuadrat tersebut. Penelitian ini menggunakan nilai l2.

b. Normalisasi Tf-IDF

Setelah mendapatkan nilai TF-IDF kemudian proses normalisasi. Normalisasi data merupakan metode yang digunakan untuk mengubah skala nilai data menjadi rentang 0 sampai 1. Proses ini penting sebelum melakukan data *mining* agar tidak terjadi dominasi oleh parameter tertentu. Dalam penelitian ini, digunakan metode L2-*Norm* dengan rumus berikut:

$$v_{norm} = \frac{\vec{v}}{||v||} = \frac{\vec{v}}{\sqrt{v1^2+v^2+v^2 \ldots\ldots v^n}} \quad (7)$$

Keterangan:
$\vec{v}$ = Nilai vektor yang dinormalisasikan
$\vec{||v||}p$ = $\vec{v}$ pada dokumen dengan nilai p = 2

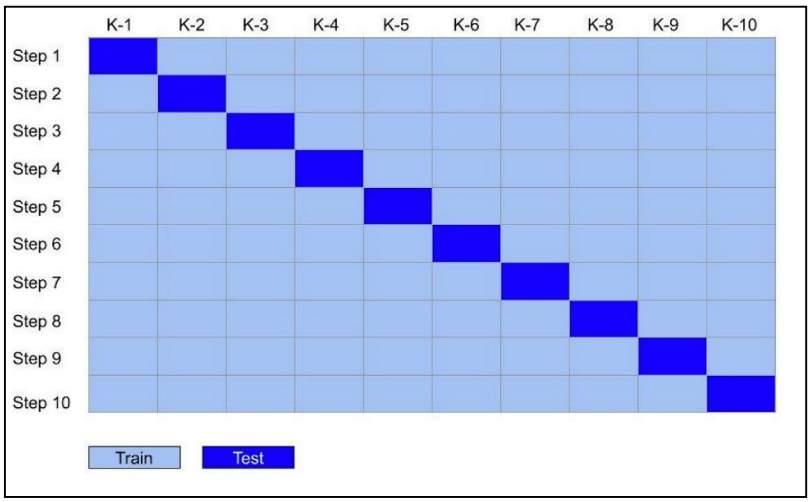
c. Klasifikasi Algoritma *K-Nearest Neighbors*

K-Nearest Neighbor (KNN) adalah algoritma yang mampu mengklasifikasikan objek berdasarkan data pelatihan yang terkait dengan objek tersebut (Barus, 2022). Prinsip kerja *K-Nearest Neighbor* (K-NN) adalah menemukan jarak terdekat antara data yang akan dievaluasi dengan k tetangga terdekat dalam data *training.* Dalam penggunaannya, algoritma KNN memiliki beberapa keunggulan, termasuk kesederhanaan dan kemudahan pemahaman, sifat non-parametrik, kemudahan dalam penyesuaian model, serta ketahanannya terhadap *noise*. (Saifurridho, Martanto & Hayati, 2024). Pada penelitian ini setelah tahap ekstraksi fitur TF-IDF, dilakukan analisis terhadap teknik klasifikasi yang telah diperoleh dengan memanfaatkan data *training*. Berikut klasifikasi algoritma *K-Nearest Neighbor* (KNN) yang menggunakan *library* sklearn dengan parameter dibawah ini (*Source code* terdapat pada lampiran 30):

a. n_neighbor : merupakan jumlah tetangga yang akan digunakan untuk menentukan label kelas suatu sampel. Pada penelitian ini nilai k yang digunakan yaitu k = 10.
b. P : parameter yang digunakan untuk menentukan jenis jarak yang digunakan. Pada penelitian ini menggunakan jarak *Euclidean*.

6. Evaluasi KNN

Pada tahap ini evaluasi *f1-score* hampir sama dengan evaluasi *WordNet*, Akan tetapi tahapan evaluasi ini *f1-score* menggunakan *K-fold Cross Validation*. Dalam penggunaan rumus, evaluasi *F1-Score* tahap ini terdapat pada rumus (1) - (4) . *K-fold cross validation* merupakan salah satu dari teknik yang difungsikan untuk memilah data menjadi data *training* serta data *testing* (Ridwansyah, 2022). Nilai yang akan diperoleh dengan *K-Fold Cross Validation* dengan menerapkan beberapa nilai K yang dilakukan sebanyak 10-*fold validation* (Fikriani, Asror, dan Murti 2019). 10-*fold cross validation* digunakan dalam membagi *dataset* ke data *training* dan data *testing*. Evaluasi tersebut bertujuan untuk menilai seberapa akurat sebuah model yang telah dibuat. nilai K= 10, Angka 10 digunakan sebagai batas akhir karena metode 10-*fold cross validation* merupakan metode yang paling umum digunakan dan memiliki estimasi performa yang akurat (Refaeilzadeh, et al., 2020)Berikut penggunaan *K-Fold Cross Validation* dengan nilai K= 10 pada gambar 2.4.



Gambar 2. 4 *Cross Validation k = 10*

Berikut parameter yang digunakan dalam proses *cross validation* (*Source code* terdapat pada lampiran 28).

a. y_true : Parameter ini sekumpulan nilai aktual atau target dari variabel dependen yang ada dalam dataset, dan dipakai dalam evaluasi.
b. y_pred : parameter ini digunakan untuk membandingkan hasil prediksi model dengan nilai target sebenarnya(y_true).
c. *Average* : parameter ini digunakan untuk mencari rata-rata dalam perhitungan *f1-Score.*