**BAB III**

**METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM**

* 1. **Metodologi Penelitian**

Metodologi penelitian pada penelitian ini meliputi pengambilan data, pelabelan manual data, *text preprocessing*, pembobotan (ekstraksi fitur), analisis sentimen (kernel linier, kernel polinomial, dan kernel sigmoid), dan pengujian.

|  |
| --- |
|  |

**Gambar 3.1 Metodologi penelitian**

* + 1. **Pengambilan Data**

Data pada penelitian ini bersifat sekunder atau data telah tersedia sebelumnya dan dikumpulkan oleh peneliti. Data bersumber dari media sosial instagram berupa komentar di akun resmi KPK, diambil secara manual dan media sosial twitter berupa *tweet* yang *memention* akun @KPK\_RI diambil oleh peneliti dengan cara *scrapping*.

* + 1. **Pelabelan Manual Data**

Pelabelan data secara manual dilakukan oleh 3 orang mahasiswa terpilih (1 mahasiswa UPN Veteran Yogyakarta dan 2 orang mahasiswa UNY) ditambah peneliti sendiri. Tujuan dilakukan pelabelan data manual oleh 3 orang + 1 orang, agar hasil *labelling* tidak subjektif jika dilakukan sendiri oleh peneliti. Cara menentukan hasil *labelling* data dengan dilakukan *voting* suara terbanyak. Ilustrasi *labelling* data manual sebagai berikut :

**Tabel 3.1**  Ilustrasi *labelling* data secara manual

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mahasiswa 1** | **Mahasiswa 2** | **Mahasiswa 3** | **Peneliti** | **HASIL** |
| **Komentar A** | Positif | Positif | Netral | - | Positif |
| **Komentar B** | Negatif | Netral | Negatif | - | Negatif |
| **Komentar C** | Positif | Negatif | Netral | Netral | Netral |

* + 1. **Text Preprocessing**

*Text preprocessing* adalah proses untuk mempersiapkan atau membersihkan data mentah yang sudah melalui pelabelan manual, untuk diekstraksi ke tahap berikutya. Proses ini meyeragamkan data mentah sehingga mudah di kelola oleh sistem. Adapun proses *preprocessing* yang digunakan antara lain *case folding*, *cleansing*, *tokenizing*, *stopword removal*, dan *stemming*. Berikut merupakan contoh komentar yang melalui tahap *preprocessing* :

**Komentar mentah** :

Semoga kinerja KPK tahun 2020 dalam memberantas korupsi semakin lebih baik dan efektif.

***Case Folding*** :

semoga kinerja kpk tahun 2020 dalam memberantas korupsi semakin lebih baik dan efektif.

***Cleansing*** :

semoga kinerja kpk tahun dalam memberantas korupsi semakin lebih baik dan efektif

***Tokenizing***:

“semoga” “kinerja” “kpk” “tahun” “dalam” “memberantas” “korupsi” “semakin” “lebih” “baik” “dan” “efektif”

***Stopword Removal***:

semoga kinerja kpk memberantas korupsi efektif

***Stemming***:

moga kinerja berantas korupsi efektif

**Komentar Hasil**:

moga kinerja berantas korupsi efektif

* + 1. **Pembobotan (Ekstraksi Fitur)**

Tahap pembobotan atau ekstraksi fitur akan menggunakan pembobotan TF-IDF. Pada TF-IDF perhitungan bobot *term* t dalam sebuah dokumen dilakukan dengan mengalikan nilai *Term Frequency* dengan *Inverse Document Frequency*. Berikut langkah-langkahnya :

1. Menghitung *Term Frequency*

*Term Frequency* (TF)adalah frekuensi kemunculan *term* dalam sebuah dokumen.**Semakin besar jumlah kemunculan term dalam sebuah dokumen, maka semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. Berikut contoh menghitung *term frequency*, dimana komentar akan melalui tahap *preprocessing* :**

Komentar 1 : Semangat kerja kpk, kami mendukungmu.

Hasil *preprocessing* : semangat kerja dukung

Komentar 2 : Kerja kpk buruk sekali

Hasil *preprocessing* : kerja buruk

Komentar 3 : Rindu dengan kpk yang dahulu

Hasil *preprocessing* : rindu

**Tabel 3.2** Menentukan *Term Frekuensi* (TF)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nomor | Term | Term Frekuensi | | |
| D1 | D2 | D3 |
| 1. | Semangat | 1 | 0 | 0 |
| 2. | Kerja | 1 | 1 | 0 |
| 3. | Dukung | 1 | 0 | 0 |
| 4. | Buruk | 0 | 1 | 0 |
| 5. | Rindu | 0 | 0 | 1 |

1. Menghitung *Inverse Document Frequency*

*Inverse Document Frequency* **(IDF)**merupakan hubungan ketersediaan sebuah *term* pada seluruh dokumen. **Semakin sedikit jumlah dokumen yang mengandung term yang dimaksud, maka nilai** *Inverse Document Frequency* **(IDF) semakin besar. Berikut contoh menghitung** *Inverse Document Frequency* **(IDF)  :**

**Menghitung nilai** *Inverse Document Frequency* **(IDF) dari term “kerja”, dimana *N* adalah jumlah seluruh dokumen atau komentar dalam sistem, dan *n* adalah jumlah dokumen atau komentar yang mengandung term dalam hal ini term “kerja”  
IDF =**

D1 = = = 0.176091

Dilanjut dengan mengalikan nilai Term Frequency (TF) dengan nilai *Inverse Document Frequency* **(IDF) dengan persamaan 2.2**, sehingga menghasilkan nilai TF-IDF sebesar :

D1 = = = 0.176091

**Tabel 3.3** Menentukan nilai *Term Frequency*- *Inverse Document Frequency* (TF-IDF)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nomor | Term | TF-IDF | | |
| D1 | D2 | D3 |
| 1. | Semangat | 0.477121 | 0 | 0 |
| 2. | Kerja | 0.176091 | 0.176091 | 0 |
| 3. | Dukung | 0.176091 | 0 | 0 |
| 4. | Buruk | 0 | 0.176091 | 0 |
| 5. | Rindu | 0 | 0 | 0.176091 |

* + 1. **Analisis Sentimen**

Analisis sentimen menerapkan Support Vector Machine (SVM) kernel linier Support Vector Machine (SVM) kernel polinomial, Support Vector Machine (SVM) kernel sigmoid berikut penjelasannya :

* + - 1. **Analisis Penerapan Support Vector Machine Kernel Linier**

Penerapan *Support Vector Machine* (SVM) kernel linier dibagi menjadi 2 tahap, tahap 1 yaitu tahap penerapan *Support Vector Machine* (SVM) pada data latih (*training*) yang sudah melewati proses *preprocessing* dan pembobotan. Proses pada data latih (*training*) digunakan untuk menghasilkan model analisis sentimen untuk acuan mengkategorikan sentimen pada data uji. Tahap 2 yaitu tahap penerapan pada data uji yang melewati proses *preprocessing*. Berikut langkah penerapan *Support Vector Machine* (SVM) kernel linier pada analisis sentimen :

1. Tahap pelatihan
2. Tahap pelatihan menggunakan metode *one against all*. Tujuan dari tahap ini yaitu merubah data latih (*training*) yang sudah melewati proses *preprocessing* dan pembobotan ke dalam format *support vector* serta memberi label pada tiap dokumen.

**Tabel 3.4** **Format data *support vector***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nomor | Term | TF-IDF | | |
| D1 (x1) | D2 (x2) | D3 (x3) |
| 1. | buruk | 0 | 0.477121 | 0 |
| 2. | dukung | 0.477121 | 0 | 0 |
| 3. | kerja | 0.176091 | 0.176091 | 0 |
| 4. | rindu | 0 | 0 | 0.477121 |
| 5. | semangat | 0.477121 | 0 | 0 |

1. Tahap pelatihan dibagi menjadi 3 kelas yaitu kelas positif, kelas negatif, dan kelas netral. **Tahap pelatihan pertama yang akan dilakukan yaitu pada kelas positif**. Kernelisasi menggunakan kernel linier. Matriks kernel linier dihitung dengan dimensi n x n, dimana n adalah banyaknya komentar pada data latih (*training*).

x1 = [0, 0.47 0.17, 0, 0.47]

x2 = [0.47, 0, 0.17, 0, 0]

x3 = [0, 0, 0, 0.47, 0]

Memberikan label “1” pada dokumen berlabel positif, dan memberikan label “-1” pada dokumen berlabel negatif dan netral.

**Tabel 3.5** Format data *support vector* pada kelas positif

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nomor | Term | TF-IDF | | |
| D1 (x1) | D2 (x2) | D3 (x3) |
| 1. | buruk | 0 | 0.47 | 0 |
| 2. | dukung | 0.47 | 0 | 0 |
| 3. | kerja | 0.17 | 0.17 | 0 |
| 4. | rindu | 0 | 0 | 0.47 |
| 5. | semangat | 0.47 | 0 | 0 |
| **Label** | | **1** | **-1** | **-1** |

Dilakukan perhitungan xi = 1

= [0, 0.47 0.17, 0, 0.47]T [0, 0.47 0.17, 0, 0.47]

= 0.4707

= [0.47, 0, 0.17, 0, 0]T [0, 0.47 0.17, 0, 0.47]

= 0.0289

= [0, 0, 0, 0.47, 0]T [0, 0.47 0.17, 0, 0.47]

= 0

Dilakukan perhitungan xi = 2

= [0, 0.47 0.17, 0, 0.47]T [0.47, 0, 0.17, 0, 0]

= 0.0289

= [0.47, 0, 0.17, 0, 0]T [0.47, 0, 0.17, 0, 0]

= 0.2498

= [0, 0, 0, 0.47, 0]T [0.47, 0, 0.17, 0, 0]

= 0

Dilakukan perhitungan xi = 3

= [0, 0.47 0.17, 0, 0.47]T [0, 0, 0, 0.47, 0]

= 0

= [0.47, 0, 0.17, 0, 0]T [0, 0, 0, 0.47, 0]

= 0

= [0, 0, 0, 0.47, 0]T [0, 0, 0, 0.47, 0]

= 0.2209

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai matrik dengan persamaan

= 1 x 1 x 0.4707 + 0.12 = 0.4807

= 1 x (-1) x 0.0289 + 0.12 = -0.0189

= 1 x (-1) x 0 + 0.12 = 0.01

= (-1) x 1 x 0.0289 + 0.12 = - 0.0189

= (-1) x (-1) x 0.2498 + 0.12 = 0.2598

= (-1) x (-1) x 0 + 0.12 = 0.01

= (-1) x 1 x 0 + 0.12 = 0.01

= (-1) x (-1) x 0 + 0.12 = 0.01

= (-1) x (-1) x 0.2209 + 0.12 = 0.2309

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai *error* dengan persamaan

= (0.4807 + (-0.0189) + 0.01) x 0.26

= 0.4718 x 0.26

= 0.1226

= ((-0.0189) + 0.2598 + 0.01) x 0.26

= 0.2509 x 0.26

= 0.6523

= (0.01+ 0.01+ 0.2309) x 0.26

= 0.2509 x 0.26

= 0.6523

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Mengitung delta alpa dengan persamaan

= min {max [0.5 (1-0.1226), -0.26], 1-0.26}

= min {max [0.4386, -0.26], 0.74 }

= min { [0.4386], 0.76 }

= 0.4386

= min {max [0.5 (1-0.0652), -0.26], 1-0.26 }

= min {max [0.1738 -0.26], 0.74 }

= min { [0.4673], 0.74 }

= 0.4673

= min {max [0.5 (1-0.0652), -0.26], 1-0.26 }

= min {max [0.1738 -0.26], 0.74 }

= min { [0.4673], 0.74 }

= 0.4673

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai dengan persamaan

= 0.26 + 0.4386 = 0,6986

= 0.26 + 0.4673 = 0,7273

= 0.26 + 0.4673 = 0,7273

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Mencari nilai bias

Nilai positif paling tinggi, terletak pada data ke 1

= (1 x 0.6986 x 0.4707) + ((-1) x 0.7273 x 0.0289) + ((-1) x 0.7273 x 0)

= 0.3288 + (-0.021) + 0

= 0.3078

Nilai negatif paling tinggi, terletak pada data ke 2

= (1 x 0.6986 x 0.0289) + ((-1) x 0.7273 x 0.2498) + ((-1) x 0.7273 x 0)

= 0.0201 + (-0.1816) + 0

= -0.1615

Maka nilai bias adalah sebagai berikut :

= -0.0731

Setelah kelas label positif selesai, selanjutnya melakukan **proses pada kelas label negatif**, hingga didapatkan nilai sebagai berikut :

Nilai positif paling tinggi, terletak pada data ke 2

= ((-1) x 0.6986 x 0.0289) + (1 x 0.7273 x 0.2498) + ((-1) x 0.7273 x 0)

= (-0.0201) + 0.1816+ 0

= 0.1615

Nilai negatif paling tinggi, terletak pada data ke 1

= ((-1) x 0.6986 x 0.4707) + (1 x 0.7273 x 0.0289) + ((-1) x 0.7273 x 0)

= -0.3288 + 0.0210 + 0

= -0.0307

Maka nilai bias adalah sebagai berikut :

= 0.0731

Setelah kelas label positif dan negatif selesai, selanjutnya melakukan **proses pada kelas label netral** hingga didapatkan nilai sebagai berikut :

Nilai positif paling tinggi, terletak pada data ke 3

= ((-1) x 0.6911 x 0) + ((-1) x 0.7198 x 0) + (1 x 0.7286 x 0.2209)

= 0 + 0 + 0.1609

= 0.1609

Nilai negatif paling tinggi, terletak pada data ke 1

= ((-1) x 0.6911 x 0,4707) + ((-1) x 0.7198 x 0,0289) + (1 x 0.7286 x 0)

= (-0.3253) + (-0.0208) + 0

= -0.3461

Maka nilai bias adalah sebagai berikut :

= 0.0925

1. Tahap pengujian

Setelah kelas label positif, negatif, dan netral selesai didapatkan nilai dan nilai *bias* selanjutnya melakukan proses pengecekan sentimen pada data uji, prosesnya sebagai berikut :

Misal melakukan pengujian dengan data uji “Semangat kerjanya KPK, rakyat Indonesia mendukungmu”. Komentar data uji akan dilakukan proses *preprocessing,* sehingga didapatakan term sebagai berikut “semangat”, “kerja”, “rakyat”, “indonesia”, “dukung” data *term* data uji akan di cocokkan dengan data *term* data latih, sehingga menghasilkan nilai matriks sebagai berikut :

**Tabel 3.6** Format data *term* data uji

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | D1 (x1) (“dukung”) | D1 (x1)  (“kerja”) | D1 (x1)  (“semangat”) | D2 (x2)  (“kerja”) | D3 (x3) |
|  | 0.47 | 0.17 | 0.47 | 0.17 | 0 |

Selanjutnya melakukan proses klasifikasi dengan fungsi keputusan , sehingga didapatkan hasil sebagai berikut :

Kelas positif

= (1 x 0.6986 x 0.47) + (1 x 0.6986 x 0.17) + (1 x 0.6986 x 0.47) + ((-1) x 0,7273 0.17) + ((-1) x 0,7273x 0) + (-0.0731)

= 0.3283 + 0.1187 + 0.3283 + (–0.1236) + (0) + (-0.0731)

= 0.5786

Kelas negatif

= ((-1) x 0,6986 x 0.47) + ((-1) x 0,6986 x 0.17) + ((-1) x 0,6986 x 0.47) + (1 x 0,7273 x 0.17) + ((-1) x 0,7273 x 0) + 0.0731

= (-0.3283 + (-0.1187) + (-0.3283) + 0.1236 + 0 + 0.0731

= -0.5787

Kelas netral

= ((-1) x 0.6911 x 0.47) + ((-1) x 0.6911 x 0.17) + ((-1) x 0,6911x 0.47) + ((-1) x 0.7198 x 0.17) + (1 x 0,7286 x 0) + 0.0925

= (-0.3248) + (-0.1174) + (-0.3248) + (-0.1223) + 0 + 0.0925

= -0.7969

Dari hasil kelas label diatas menunjukkan bahwa kelas sentimen positif mempunyai nilai paling tinggi dari pada kelas senitimen lain, sehingga komentar data uji “Semangat kerjanya KPK, rakyat Indonesia mendukungmu” tergolong ke kelas sentimen positif.

* + - 1. **Analisis Penerapan Support Vector Machine Kernel Polinomial**

Penerapan *Support Vector Machine* (SVM) kernel polinomial mempuyai langkah yang sama dengan *Support Vector Machine* (SVM) kernel linier, perbedaan mendasar ada pada penggunaan rumus kernelnya. Berikut langkah penerapan *Support Vector Machine* (SVM) kernel polinomial pada analisis sentimen :

1. Tahap pelatihan
2. Tahap pelatihan menggunakan metode *one against all*. Tujuan dari tahap ini yaitu merubah data latih (*training*) yang sudah melewati proses *preprocessing* dan pembobotan ke dalam format *support vector* serta memberi label pada tiap dokumen.
3. Tahap pelatihan dibagi menjadi 3 kelas yaitu kelas positif, kelas negatif, dan kelas netral. **Tahap pelatihan pertama yang akan dilakukan yaitu pada kelas positif**. Kernelisasi menggunakan kernel polinomial. Matriks kernel polinomial dihitung dengan dimensi n x n, dimana n adalah banyaknya komentar pada data latih (*training*). Dengan p = 3, =1, C=1, dan coef0 = 0

x1 = [0, 0.47 0.17, 0, 0.47]

x2 = [0.47, 0, 0.17, 0, 0]

x3 = [0, 0, 0, 0.47, 0]

Memberikan label “1” pada dokumen berlabel positif, dan memberikan label “-1” pada dokumen berlabel negatif dan netral, berdasarkan pada tabel 3.5.

Dilakukan perhitungan xi = 1

= (1 [0, 0.47 0.17, 0, 0.47]T [0, 0.47 0.17, 0, 0.47] + 0)3

= (1 x 0.4707 + 0 )3

= 0.1042

= (1 [0.47, 0, 0.17, 0, 0]T [0, 0.47 0.17, 0, 0.47] + 0) 3

= (1 x 0.0289 + 0)3

= 0.000024

= (1 [0, 0, 0, 0.47, 0]T [0, 0.47 0.17, 0, 0.47] + 0) 3

= (1 x 0 + 0)3

= 0

Dilakukan perhitungan xi = 2

= (1 [0, 0.47 0.17, 0, 0.47]T [0.47, 0, 0.17, 0, 0] + 0) 3

= (1 x 0.0289 + 0)3

= 0.000024

= (1 [0.47, 0, 0.17, 0, 0]T [0.47, 0, 0.17, 0, 0] + 0) 3

= (1 x 0.2498 + 0)3

= 0.0155

= (1 [0, 0, 0, 0.47, 0]T [0.47, 0, 0.17, 0, 0] + 0) 3

= (1 x 0 + 0)3

= 0

Dilakukan perhitungan xi = 3

= (1 [0, 0.47 0.17, 0, 0.47]T [0, 0, 0, 0.47, 0] + 0) 3

= (1 x 0 + 0)3

= 0

= (1 [0.47, 0, 0.17, 0, 0]T [0, 0, 0, 0.47, 0] + 0) 3

= (1 x 0 + 0)3

= 0

= (1 [0, 0, 0, 0.47, 0]T [0, 0, 0, 0.47, 0] + 0) 3

= (1 x 0.2209 + 0)3

= 0.0107

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung matrik dengan persamaan

= 1 x 1 x 0.1042 + 0.12 = 0.1142

= 1 x (-1) x 0.000024 + 0.12 = 0.0099

= 1 x (-1) x 0 + 0.12 = 0.01

= (-1) x 1 x 0.000024 + 0.12 = 0.0099

= (-1) x (-1) x 0.0155 + 0.12 = 0.0255

= (-1) x (-1) x 0 + 0.12 = 0.01

= (-1) x 1 x 0 + 0.12 = 0.01

= (-1) x (-1) x 0 + 0.12 = 0.01

= (-1) x (-1) x 0.0107 + 0.12 = 0.0207

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai error dengan persamaan

= (0.1142 + 0.0099 + 0.01) x 0.86

= 0.1341 x 0.86

= 0.1154

= (0.0099 + 0.0255 + 0.01) x 0.86

= 0.0454 x 0.86

= 0.0391

= (0.01+ 0.01+ 0.0207) x 0.86

= 0.0407 x 0.26

= 0.0350

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Mengitung delta alpa dengan persamaan

= min {max [1 (1-0.1154), -0.86], 1-0.86}

= min {max [0.8846, -0.86], 0.14 }

= min { [0.8846], 0.14 }

= 0.14

= min {max [1 (1-0.0391), -0.86], 1-0.86 }

= min {max [0.9609 -0.86], 0.14 }

= min { [0.9609], 0.14 }

= 0.14

= min {max [1 (1-0.0350), -0.86], 1-0.86 }

= min {max [0.965 -0.86], 0.14 }

= min { [0.965], 0.74 }

= 0.14

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai dengan persamaan

= 0.86 + 0.14 = 1

= 0.86 + 0.14 = 1

= 0.86 + 0.14 = 1

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Mencari nilai bias

Nilai positif paling tinggi, terletak pada data ke 1

= (1 x 1 x 0.1042) + ((-1) x 1 x ) + ((-1) x 1 x 0)

= 0.1042 + (-0.021) + 0

= 0.1042

Nilai negatif paling tinggi, terletak pada data ke 2

= (1 x 1 x 0.000024) + ((-1) x 1 x 0.0155) + ((-1) x 1 x 0)

= 0.000024+ (-0.0155) + 0

= -0.0155

Maka nilai bias adalah sebagai berikut :

= -0.04435

Setelah kelas label positif selesai, selanjutnya melakukan **proses pada kelas label negatif**, hingga didapatkan nilai sebagai berikut :

Nilai positif paling tinggi, terletak pada data ke 2

= ((-1) x 1 x 0.0.000024) + (1 x 1 x 0.0155) + ((-1) x 1 x 0)

= (-0.000024) + 0.0155+ 0

= 0.0155

Nilai negatif paling tinggi, terletak pada data ke 1

= ((-1) x 1 x 0.1042) + (1 x 1 x 0.000024) + ((-1) x 1 x 0)

= -0.1042 + 0.000024 + 0

= -0.1042

Maka nilai bias adalah sebagai berikut :

= -0.4435

Setelah kelas label positif dan negatif selesai, selanjutnya melakukan **proses pada kelas label netral** hingga didapatkan nilai sebagai berikut :

Nilai positif paling tinggi, terletak pada data ke 3

= ((-1) x 1 x 0) + ((-1) x 1 x 0) + (1 x 1 x 0.0107)

= 0 + 0 + 0.0107

= 0.0107

Nilai negatif paling tinggi, terletak pada data ke 1

= ((-1) x 1 x 0.1042) + ((-1) x 1 x 0.000024) + (1 x 1 x 0)

= (-0.1042) + (-0.000024) + 0

= -0.1043

Maka nilai bias adalah sebagai berikut :

= 0.0467

1. Tahap pengujian

Setelah kelas label positif, negatif, dan netral selesai didapatkan nilai dan nilai *bias* selanjutnya melakukan proses pengecekan sentimen pada data uji, prosesnya sebagai berikut :

Misal melakukan pengujian dengan data uji “Kerja KPK buruk sekali, gak guna”. Komentar data uji akan dilakukan proses *preprocessing,* sehingga didapatakan term sebagai berikut “kerja”, “buruk” data *term* data uji akan di cocokkan dengan data *term* data latih, sehingga menghasilkan nilai matriks sebagai berikut :

**Tabel 3.7** Format data *term* data uji

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | D1 (x1) (“kerja”) | D2 (x2)  (“buruk”) | D2 (x2)  (“kerja”) | D3 (x3) |
|  | 0.17 | 0.47 | 0.17 | 0 |

Selanjutnya melakukan proses klasifikasi dengan fungsi keputusan , sehingga didapatkan hasil sebagai berikut :

Kelas positif

= (1 x 1 x 0.17) + ((-1) x 1 x 0.47) + ((-1) x 1 x 0.17) + ((-1) x 1 + 0) + (-0.0443)

= 0.17 + (-0.47) + (-0.17) + (0) + (-0.0443)

= -0.5143

Kelas negatif

= ((-1) x 1 x 0.17) + (1 x 1 x 0.47) + (1 x 1 x 0.17) + ((-1) x 1 x 0) + (-0.0443)

= (-0.17) + 0.47 + 0.17 + 0 + (-0.0443)

= 0.4256

Kelas netral

= ((-1) x 1 x 0.17) + ((-1) x 1 x 0.47) + ((-1) x 1 x 0.17) + (1 x 1 x 0) + 0.0467

= (-0.17) + (-0.47) + (-0.17) + 0 + 0.0467

= -0.7632

Dari hasil kelas label diatas menunjukkan bahwa kelas sentimen negatif mempunyai nilai paling tinggi dari pada kelas senitimen lain, sehingga komentar data uji “Kerja KPK buruk sekali, gak guna” tergolong ke kelas sentimen negatif.

* + - 1. **Analisis Penerapan Support Vector Machine Kernel Sigmoid**

Penerapan *Support Vector Machine* (SVM) kernel sigmoid mempuyai langkah yang sama dengan *Support Vector Machine* (SVM) kernel linier dan kernel polinomial, perbedaan mendasar ada pada penggunaan rumus kernelnya. Berikut langkah penerapan *Support Vector Machine* (SVM) kernel sigmoid pada analisis sentimen :

1. Tahap pelatihan
2. Tahap pelatihan menggunakan metode *one against all*. Tujuan dari tahap ini yaitu merubah data latih (*training*) yang sudah melewati proses *preprocessing* dan pembobotan ke dalam format *support vector* serta memberi label pada tiap dokumen.
3. Tahap pelatihan dibagi menjadi 3 kelas yaitu kelas positif, kelas negatif, dan kelas netral. **Tahap pelatihan pertama yang akan dilakukan yaitu pada kelas positif**. Kernelisasi menggunakan kernel sigmoid. Matriks kernel sigmoid dihitung dengan dimensi n x n, dimana n adalah banyaknya komentar pada data latih (*training*). Dengan =1, C=1, dan coef0 = 0

x1 = [0, 0.47 0.17, 0, 0.47]

x2 = [0.47, 0, 0.17, 0, 0]

x3 = [0, 0, 0, 0.47, 0]

Memberikan label “1” pada dokumen berlabel positif, dan memberikan label “-1” pada dokumen berlabel negatif dan netral, berdasarkan pada tabel 3.5.

Dilakukan perhitungan xi = 1

= tanh (1 [0, 0.47 0.17, 0, 0.47]T [0, 0.47 0.17, 0, 0.47] + 0)

= tanh (1 x 0.4707 + 0 )

= 0.4487

= tanh (1 [0.47, 0, 0.17, 0, 0]T [0, 0.47 0.17, 0, 0.47] + 0)

= tanh (1 x 0.0289 + 0)

= 0.2889

= tanh (1 [0, 0, 0, 0.47, 0]T [0, 0.47 0.17, 0, 0.47] + 0)

= tanh (1 x 0 + 0)

= 0

Dilakukan perhitungan xi = 2

= tanh (1 [0, 0.47 0.17, 0, 0.47]T [0.47, 0, 0.17, 0, 0] + 0)

= tanh (1 x 0.0289 + 0)

= 0.0288

= tanh (1 [0.47, 0, 0.17, 0, 0]T [0.47, 0, 0.17, 0, 0] + 0)

= tanh (1 x 0.2498 + 0)

= 0.2447

= tanh (1 [0, 0, 0, 0.47, 0]T [0.47, 0, 0.17, 0, 0] + 0)

= tanh (1 x 0 + 0)

= 0

Dilakukan perhitungan xi = 3

= tanh (1 [0, 0.47 0.17, 0, 0.47]T [0, 0, 0, 0.47, 0] + 0)

= tanh (1 x 0 + 0)

= 0

= tanh (1 [0.47, 0, 0.17, 0, 0]T [0, 0, 0, 0.47, 0] + 0)

= tanh (1 x 0 + 0)

= 0

= (1 [0, 0, 0, 0.47, 0]T [0, 0, 0, 0.47, 0] + 0)

= (1 x 0.2209 + 0)

= 0.2173

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung matrik dengan persamaan

= 1 x 1 x 0.4387 + 0.12 = 0.4487

= 1 x (-1) x 0.0288 + 0.12 = -0.0188

= 1 x (-1) x 0 + 0.12 = 0.01

= (-1) x 1 x 0.0288 + 0.12 = -0.0188

= (-1) x (-1) x 0.2447 + 0.12 = 0.2547

= (-1) x (-1) x 0 + 0.12 = 0.01

= (-1) x 1 x 0 + 0.12 = -0.01

= (-1) x (-1) x 0 + 0.12 = 0.01

= (-1) x (-1) x 0.2173 + 0.12 = 0.2273

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai error dengan persamaan

= (0.4487 + (-0.0188) + 0.01) x 0.56

= 0.4399 x 0.56

= 0.2463

= ((-0.0188) + 0.2547 + 0.01) x 0.56

= 0.2459 x 0.56

= 0.1376

= (0.01 + 0.01 + 0.2273) x 0.56

= 0.2473 x 0.56

= 0.1385

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Mengitung delta alpa dengan persamaan

= min {max [1 (1-0.2463), -0.56], 1-0.56}

= min {max [0.7537, -0.56], 0.44 }

= min { [0.7537], 0.44 }

= 0.44

= min {max [1 (1-0.1376), -0.56], 1-0.56 }

= min {max [0.8624], -0.56], 0.44 }

= min { [0.8624], 0.44 }

= 0.44

= min {max [1 (1-0.1385), -0.56], 1-0.56 }

= min {max [0.8615 -0.56], 0.44 }

= min { [0.8615], 0.44 }

= 0.44

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Menghitung nilai dengan persamaan

= 0.56 + 0.44 = 1

= 0.56 + 0.44 = 1

= 0.56 + 0.44 = 1

Sehingga secara keseluruhan akan membentuk nilai matrik sebagai berikut:

1. Mencari nilai bias

Nilai positif paling tinggi, terletak pada data ke 1

= (1 x 1 x 0.4387) + ((-1) x 1 x 0.0288) + ((-1) x 1 x 0)

= 0.4387 + (-0.0288) + 0

= 0.4098

Nilai negatif paling tinggi, terletak pada data ke 3

= (1 x 1 x 0) + ((-1) x 1 x 0) + ((-1) x 1 x 0.2173)

= 0 + 0 + (-0.2173)

= -0.2173

Maka nilai bias adalah sebagai berikut :

= -0.3075

Setelah kelas label positif selesai, selanjutnya melakukan **proses pada kelas label negatif**, hingga didapatkan nilai sebagai berikut :

Nilai positif paling tinggi, terletak pada data ke 2

= ((-1) x 0.9368 x 0.0288) + (1 x 0.9911 x 0.2447) + ((-1) x 0.9907 x 0)

= (-0.0269) + 0.2425 + 0

= 0.2155

Nilai negatif paling tinggi, terletak pada data ke 1

= ((-1) x 0.9368 x 0.4387) + (1 x 0.9911 x 0.0288) + ((-1) x 0.9907 x 0)

= (-0.4109) + 0.0285 + 0

= -0.3824

Maka nilai bias adalah sebagai berikut :

= -0.0834

Setelah kelas label positif dan negatif selesai, selanjutnya melakukan **proses pada kelas label netral** hingga didapatkan nilai sebagai berikut :

Nilai positif paling tinggi, terletak pada data ke 3

= ((-1) x 0.9206 x 0) + ((-1) x 0.9749 x 0) + (1 x 0.9907 x 0.2173)

= 0 + 0 + 0.2152

= 0.2153

Nilai negatif paling tinggi, terletak pada data ke 1

= ((-1) x 0.9206 x 0.4387) + ((-1) x 0.9749 x 0.0288) + (1 x 0.9907 x 0)

= (-0.4038) + (-0.0280) + 0

= -0.4321

Maka nilai bias adalah sebagai berikut :

= 0.1083

1. Tahap pengujian

Setelah kelas label positif, negatif, dan netral selesai didapatkan nilai dan nilai *bias* selanjutnya melakukan proses pengecekan sentimen pada data uji, prosesnya sebagai berikut :

Misal melakukan pengujian dengan data uji “Aku rindu sama kinerja kpk taun 2018.”. Komentar data uji akan dilakukan proses *preprocessing,* sehingga didapatakan term sebagai berikut “rindu”, “kinerja”, “taun”, data *term* data uji akan di cocokkan dengan data *term* data latih, sehingga menghasilkan nilai matriks sebagai berikut :

**Tabel 3.8** Format data *term* data uji

|  |  |  |  |
| --- | --- | --- | --- |
|  | D1 (x1) | D2 (x2) | D3 (x3)  (“rindu”) |
|  | 0 | 0 | 0.47 |

Selanjutnya melakukan proses klasifikasi dengan fungsi keputusan , sehingga didapatkan hasil sebagai berikut :

Kelas positif

= (1 x 1 x 0) + ((-1) x 1 x 0) + ((-1) x 1 x 0.47) + (-0.3075)

= 0 + 0 + (-0.47) + (-0.3075)

= -0.7775

Kelas negatif

= ((-1) x 0.9368 x 0) + (1 x 0.9911 x 0) + ((-1) x 0,9907 x 0.47) + (-0.0834)

= 0 + 0 + 0.4656 + (-0.0834)

= 0.3821

Kelas netral

= ((-1) x 0.9206 x 0) + ((-1) x 0.9749 x 0) + ( 1 x 0.9907 x 0.47) + 0.1083

= 0 + 0 + 0.4656 + 0.1083

= 0.5740

Dari hasil kelas label diatas menunjukkan bahwa kelas sentimen netral mempunyai nilai paling tinggi dari pada kelas senitimen lain, sehingga komentar data uji “Aku rindu sama kinerja kpk taun 2018” tergolong ke kelas sentimen netral.

* 1. **Metode Pengembangan Sistem**

Metode yang digunakan dalam pengembangan sistem penelitian ini adalah *Guidelines for Rappid APPLication Engineering* (GRAPPLE). Menurut Schmuller (2004), GRAPPLE merupakan metodologi yang fleksibel dan memberikan panduan yang jelas dalam proses pengembangan sistem, serta ditujukan untuk sistem bersifat orientasi objek. Metode GRAPPLE terdiri dari 5 tahapan yaitu *requirement gathering*, *analysis*, *design*, *development* dan *deployment*. Pada penelitian ini akan menggunakan tahapan *requirement gathering*, *analysis*, dan *design*.

* + 1. **Requirement Gathering**

*Requirement gathering* membahas mengenai analisis masalah dan kebutuhan sistem (*Requirements*) yang dimaksudkan untuk memperjelas pemahaman terhadap suatu masalah.

* + - 1. **Analisis Masalah**

Analisis masalah menggambarkan permasalahan pada penelitian ini, mengenai analisis sentimen pada komentar twitter dan instagram tentang Komisi Pemberantasan Korupsi (KPK). Tujuan dilakukannya penelitian ini adalah untuk mengetahui komentar masyarakat apakah bersifat positif, negatif, atau netral. Dikembangkan sebuah sistem cerdas yang dapat melakukan analisis sentimen secara sendirinya atau otomatis merupakan sebuah solusi dari permasalahan ini. Permasalahan pada penelitian ini yaitu bagaimana mengklasifikasikan komentar di twitter dan instagram mengenai Komisi Pemberantasan Korupsi (KPK) apakah bersifat positif, negatif, atau netral, serta penelitian ini dapat dimanfaatkan untuk menilai kinerja Komisi Pemberantasan Korupsi (KPK) dengan membandingkan kernel linier dan kernel polinomial pada metode *Support Vector Machine* (SVM) sehingga juga dapat diketahui kernel manakah yang lebih akurat dalam analisis sentimen.

* + - 1. **Analisis Kebutuhan**

Analisis kebutuhan bertujuan untuk mendeskripsikan penentuan [kebutuhan](https://id.wikipedia.org/wiki/Kebutuhan" \o "Kebutuhan) atau kondisi apa yang harus dipenuhi dalam suatu sistem. Dalam menyelesaikan beberapa permasalahan, perlu adanya beberapa analisis kebutuhan antara lain :

1. **Kebutuhan Data Input**

Kebutuhan data *input* merupakan sebuah data masukan yang disimpan dalam *database* yang berguna dalam membangun sistem ini, antara lain :

1. Data admin

Data admin yang diperlukan berupa *id\_user*, *username* dan *password* dari admin untuk mengakses sistem.

1. Data kamus *stopword*

Data kamus *stopword* berasal dari <http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/>. Penghilangan *stop word* bertujuan mengurangi ukuran indeks, waktu pemrosesan, dan mengurangi level *noise* sehingga fokus pada kata-kata penting. Kata-kata yang dimaksud seperti “yang”, “di”, “dan”, “ke”, “dari”, “jika”, “saat”, “hanya”, “atau”.

1. Data kamus *stemming*

Data kamus *stemming* berasal dari <http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/>. *Stemming* adalah proses mengubah kata dalam sebuah dokumen dengan menghilangkan imbuhan kata dan mengubahnya menjadi kata dasar atau akar kata (*stem*) (Wulandari et al., 2019).

1. Data latih

Data latih yang dimasukan ke sistem berjumlah 90% dari jumlah dataset berasal dari *scrapping* komentar instagram dan *scrapping* komentar twitter mengenai Komisi Pemberantasan Korupsi (KPK) dari tahun 2018 – 2020 akan tersimpan di *database*. Data latih berfungsi untuk melatih algoritma yang digunakan.

1. Data uji

Data uji yang digunakan berjumlah 10% dari jumlah dataset berasal dari *scrapping* komentar instagram dan *scrapping* komentar twitter mengenai Komisi Pemberantasan Korupsi (KPK) dari tahun 2018 – 2020 akan tersimpan di *database*. Data uji berfungsi untuk menguji dari kekuatan algoritma yang digunakan.

1. **Kebutuhan Proses Sistem**

Kebutuhan proses sistem merupakan keadaan ketika sistem sedang dijalankan, maka terdapat beberapa proses yang berjalan, antara lain :

1. Proses data latih

Proses ini akan memproses data latih yang telah di masukkan ke sistem untuk di olah ke proses *preprocessing*, pembobotan, serta proses *training*.

1. Proses *preprocessing*

Proses *preprocessing* berfungsi untuk mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah untuk diproses oleh sistem (Syakuro, 2017). Pada sistem yang dibangun ini menggunakan proses *preprocessing* antara lain *case folding*, *cleansing*, *tokenizing*, *stopword removal*, dan *stemming*.

1. Proses pembobotan

Proses pembobotan berfungsi untuk menghitung kemunculan suatu kata pada tiap dokumen (*term*) serta pemberian bobot pada setiap *term*. Proses pembobotan yang digunakan menggunakan pembobotan *Term Frequency* (TF) dengan *Inverse Document Frequency* (IDF) atau TF-IDF.

1. Proses *training*

Proses *traning* akan melatih data latih yang telah di proses preprocessing dan pembobotan dengan kernel linier dan kernel polinomial pada metode *Support Vector Machine* (SVM) untuk menjadi acuan data pada proses data uji.

1. Proses data uji

Proses ini akan menguji data yang belum diketahui klasifikasinya.

1. **Kebutuhan Data Output**

Kebutuhan data *output* merupakan data keluaran hasil dari proses sistem yang dijalankan, antara lain :

1. Perbandingan hasil klasifikasi sentimen komentar yang telah melalui proses *preprocessing* serta pembobotan pada kernel linier dan kernel polinomial dengan metode *Support Vector Machine* (SVM)
2. Hasil pengujian akurasi, *recall*, presisi dari sistem yang telah dibangun
3. Hasil sentimen mengenai Komisi Pemerantasan Korupsi (KPK) per media sosial
4. **Kebutuhan Perangkat Lunak**

Kebutuhan perangkat lunak (*software*) yang diperlukan dalam membangun sistem analisis sentimen ini antara lain :

1. Sistem operasi berupa Windows 10
2. *Text editor* berupa Visual Studio Code
3. *Web server* berupa XAMPP
4. *Data Base Managemet System* (DBMS) berupa MySQL
5. *Browser* berupa Google Chrome
6. **Kebutuhan Perangkat Keras**

Kebutuhan perangkat keras (*hardware*) yang diperlukan dalam membangun sistem analisis sentimen ini antara lain :

1. *Processor* dangan *Internal Graphic Processing Unit* (IGPU)AMD Ryzen 5 2500U Radeon Vega Mobile Gfx 2.00 GHz
2. *Random Access Memory* (RAM) 8 GB DDR4
3. *Storage State Drive* (SSD) 350 GB
   * 1. **Analysis dan Design**

*Analysis* dan *design* merupakan tahap dimana akan dilakukan analisis sistem, perancangan sistem, dan perancangan *Unified Modeling Language* (UML). Menurut Schmuller (2004) tahap *analysis* dan *design* harus bolak-balik sampai tahap *design* selesai, serta menggabungkan *analysis* dan *design* menjadi satu tahap.

* + - 1. **Perancangan Sistem**

Perancangan sistem terdiri dari arsitektur sistem, *use case diagram*, *data flow diagram*, dan *flowchart diagram*. Berikut penjelasannya :

1. **Arsitektur Sistem**

Perancangan arsitektur sistem bertujuan menggambarkan pemetaan rencana kebutuhan dalam suatu sistem. Terdapat 2 aktor dalam perancangan sistem ini, yaitu admin dan pengunjung. Aktor admin mempunyai akses untuk memasuki sistem dengan beberapa hak akses, antara lain mengelola kamus *preprocessing*, mengelola data latih, mengelola pembobotan, mengelola model *Support Vector Machine* (SVM), mengeloladata uji serta mendapatkan hasil analisis sentiment berserta akurasi kernel linier dan kernel polinomial metode *Support Vector Machine* (SVM). Aktor pengunjung merupakan aktor dengan akses terbatas, yaitu hanya dapat melihat hasil akurasi kernel linier dan kernel polinomial metode *Support Vector Machine* (SVM) serta hasil sentimen per media sosial instagram dan twitter. Teknik pengambilan data menggunakan teknik *scrapping* pada twitter dan pengambilan data manual pada instagram, lalu dijadikan menjadi 1 file dan disimpan di *database*

|  |
| --- |
|  |

**Gambar 3.2 Arsitektur sistem**

1. **Use Case Diagram**

*Use case diagram* menjelaskan gambaran interaksi atau aktivitas apa saja yang dapat dilakukan oleh aktor dengan sistem, Terdapat 2 aktor dalam use case diagram yang dibuat, yaitu admin dan pengguna. Berikut ini penjelasan lebih lanjut

1. Tampilan awal disajikan tampilan beranda, kedua aktor dapat melihat tampilan beranda, untuk aktor admin dapat melakukan *login* untuk aktivitas lebih lanjut, dengan *username* dana *password* yang sudah ditentukan sebelumnya.
2. Aktor admin berhasil *login*, terdapat 7 entitas yang dapat dilakukan oleh admin, meliputi mengelola (*input*, *edit*, *delete*) data latih, mengelola (*input*, *edit*, *delete*) kamus data (*stopword* dan *stemming*), mengelola pembobotan, melakukan *training* data latih, mengelola data uji (*input*, *edit*, *delete*), melakukan analisis sentimen untuk melihat klasifikasi komentar, dan melihat hasil akurasi, presisi serta *recall*.
3. Aktor pengunjung dapat melihat akurasi, presisi, dan *recall* kernel linier, kernel polinomial, dan kernel sigmoid pada metode *Support Vector Machine* (SVM) serta melihat grafik sentimen mengenai Komisi Pemberantasan Korupsi (KPK) pada halaman web beranda tanpa perlu *login*.

|  |
| --- |
|  |

**Gambar 3.3 Use case diagram sistem**

1. **Data Flow Diagram (DFD)**

Data Flow Diagram (DFD) menggambarkan aliran data pada suatu sistem, Berikut Data Flow Diagram (DFD) dari sistem ini :

1. Data Flow Diagram (DFD) Level 0

*Data Flow Diagram* (DFD) level 0 mengambarkan secara *basic* atau sederhana aliran data pada sistem dari *input* sampai *ourput* Terdapat 2 entitas pada DFD dibawah ini yaitu admin dan pengunjung.Berikut adalah gambarannya :

|  |
| --- |
|  |

**Gambar 3.4 *Data Flow Diagram* (DFD) level 0**

1. Data Flow Diagram (DFD) Level 1

*Data Flow Diagram* (DFD) level 1 menggambarkan aliran data pada sistem yang lebih kompleks dari DFD level 0. Pada DFD level 1 ini akan terbentuk sebuah *store data* dan aliran data. Terdapat 3 proses pada DFD level 1 dibawah ini yaitu *login*, mengelola data, dan hasil analisis sentimen. Terdapat 7 *store* data yang akan disimpan di MySQL meliputi *store* data user, *strore* data stowwords, *store* data kamuskds, *store* data datalatihs, *store* data pembobotans, *store* data datujis, dan *store* data akurasi. Terdapat 2 *store* data yang akan disimpan sebagai model *manager* *training* data latih yaitu *store* data model. Berikut adalah gambaran dari DFD level 1 :

|  |
| --- |
|  |

**Gambar 3.5** ***Data Flow Diagram* (DFD) level 1**

1. Data Flow Diagram (DFD) Level 2 (Admin)

*Data Flow Diagram* (DFD) Level 2 (admin) menggambarkan rincian atau menguraikan proses-proses apa saja dari *Data Flow Diagram* (DFD) level 1. Terdapat 8 proses yang digambarkan, yaitu proses mengolah data *stopword*, mengolah data *stemming*, mengolah data latih, mengolah data pembobotan, *training* model, mengolah data uji, hasil analisis sentimen, serta menghitung akurasi, *presisi*, dan *recall*

|  |
| --- |
|  |

**Gambar 3.6**  ***Data Flow Diagram* (DFD) level 2 (Admin)**

1. **Flowchart Diagram**

*Flowchart* menggambarkan alur dari suatu proses atau algoritma, berikut adalah *flowchart* secara keseluruhan :

1. Flowchart Sistem

*Flowchart* sistem menggambarkan proses atau alur dimulai dari memasukkan komentar data latih, kemudian akan lanjut ke proses *text preprocessing*, lalu diteruskan ke proses pembobotan atau ekstraksi fitur TF-IDF, terakhir akan membentuk model SVM kernel liner dan kernel polinomial. Untuk flowchart sistem kedua yaitu mengklasifikasikan komentar data uji, melewati tahap *text preprocessing*, pembobotan atau ekstraksi fitur TF-IDF, lalu akan *meload* model SVM kernel linier, kernel polinomial, dan kernel sigmoid lalu akan dilakukan analisis sentimen komentar data uji dengan model yang *diload* sebelumnya, terakhir akan ditampilkan hasil analisis sentimen pada SVM tiap kernel.

|  |
| --- |
|  |

**Gambar 3.7 *Flowchart* sistem**

1. Flowchart Case Folding

*Flowchart case folding* menggambarkansalah satu proses *preprocessing* yang berfungsi untuk mengubah huruf *uppercase* menjadi *lowercase*. Dimulai dengan memasukkan data latih atau data uji ke sistem, lalu sistem akan mengubah data latih urut berdasarkan id, jika sudah selesai akan lanjut ke tahap berikutnya yaitu tahap *cleansing*.

|  |
| --- |
|  |

**Gambar 3.8** ***Flowchart case folding***

1. Flowchart Cleansing

*Cleansing* merupakan salah satu proses *preprocessing* setelah proses *case folding*, yang berfungsi menghapus URL, menghapus angka, menghapus tanda baca, menghapus kata ganda atau *double*, dan menghapus karakter atau huruf yang ganda atau *double*. Setelah proses cleansing selesai akan lanjut tahap berikutnya, yaitu tahap *tokeninizing*.

|  |
| --- |
|  |

**Gambar 3.9** ***Flowchart cleansing***

1. Flowchart Tokenizing

*Flowchart tokenizing* menggambarkan salah satu proses *preprocessing* dengan mengubah kalimat dari bentuk *string* menjadi bentuk *array*, untuk keperluan proses *preprocessing* selanjutnya yaitu stopword removal.

|  |
| --- |
|  |

**Gambar 3.10** ***Flowchart tokenizing***

1. Flowchart Stopword Removal

*Flowchart stopword removal* menggambarkan salah satu proses *preprocessing* dengan menghapus kata yang tidak tidak berhubungan (*irrelevant*) dengan subyek utama seperti “yang”, “di”, “dan”, “ke”, “dari”, “jika”, “saat”, “hanya”, “atau” dan masih banyak lagi sesuai kamus yang sudah didaftarkan. Sistem akan mengecek apakah dalam suatu komentar terdapat kata yang sama dengan kata di kamus stopword removal, jika sama maka kata dalam komentar tersebut akan dihapus lalu komentar akan menjadi hasil komentar *stopword removal*. Jika tidak ada kata yang sesuai maka komentar dari hasil *tokenizing* akan menjadi hasil komentar *stopword removal*.

|  |
| --- |
|  |

**Gambar 3.11** ***Flowchart stopword removal***

1. Flowchart Stemming

*Flowchart stemming* menggambarkan salah satu proses *preprocessing* dengan mengubah sebuah kata imbuhan menjadi sebuah kata dasar (*stem*). Kata dasar pada proses stemming disesuaikan dengan kamus yang sudah didaftarkan.

|  |
| --- |
|  |

**Gambar 3.12** ***Flowchart stemming***

1. Flowchart Pembobotan TF-IDF

Flowchart pembobotan *TF-IDF* menggambarkan pembobotan sebuah data setelah melalui proses *preprocessing* yang mana akan menghasilkan bobot atau nilai pada setiap *term* atau kata. Data komentar yang telah selesai melalui proses *preprocessing* akan di hitung *Term Frequency* nya(TF), jika terdapat *term* yang sama di 1 komentar maka jumlah TF akan di tambah 1. Lalu jika dalam kondisi tidak, akan dihitung jumlah komentar atau komentar (K), setelah menghitung nTerm, dan yang terakhir akan menghasilkan nilai bobot TF-IDF pada tiap term. Proses ini akan berulang pada tiap term tiap komentar atau dokumen.

|  |
| --- |
|  |

**Gambar 3.13** ***Flowchart* pembobotan *TF-IDF***

1. Flowchart Data Latih SVM Kernel Linier

*Flowchart* ini menggambarkan aliran dari input data latih yang selanjutnya masuk ke proses *preprocessing*, setelah itu akan dilakukan pembobotan TF-IDF yang akan menghasilkan bobot pada tiap *term*, lalu diubah menjadi bentuk vektor dan dilakukan *traning* data latih hingga terbentuk model data latih SVM kernel linier. Parameter yang digunakan pada kernel liner untuk menentukan bobot yaitu = 1.

|  |
| --- |
|  |

**Gambar 3.14** ***Flowchart* data latih SVM kernel linier**

1. Flowchart Data Latih SVM Kernel Polinomial

*Flowchart* ini menggambarkan aliran dari input data latih yang selanjutnya masuk ke proses *preprocessing*, setelah itu akan dilakukan pembobotan TF-IDF yang akan menghasilkan bobot pada tiap *term*, lalu diubah menjadi bentuk vektor dan dilakukan *traning* data latih hingga terbentuk model data latih SVM kernel polinomial. Parameter yang digunakan pada kernel liner untuk menentukan bobot antara lain *degree* = 3, = 0.5, dan = 0.5.

|  |
| --- |
|  |

**Gambar 3.15** ***Flowchart* data latih SVM kernel polinomial**

1. Flowchart Data Latih SVM Kernel Sigmoid

*Flowchart* ini menggambarkan aliran dari input data latih yang selanjutnya masuk ke proses *preprocessing*, setelah itu akan dilakukan pembobotan TF-IDF yang akan menghasilkan bobot pada tiap *term*, lalu diubah menjadi bentuk vektor dan dilakukan *traning* data latih hingga terbentuk model data latih SVM kernel sigmoid. Parameter yang digunakan pada kernel liner untuk menentukan bobot antara lain = 0.5, dan = 0.5, = 0.5, dan .

|  |
| --- |
|  |

**Gambar 3.16** ***Flowchart* data latih SVM kernel sigmoid**

1. Flowchart Data Uji SVM Kernel Linier

*Flowchart* ini menggambarkan aliran datalatih yang diinputkan akan melalui proses *preprocessing*, lalu sistem akan meload model SVM kernel linier data *training*, setelah kalimat di *preprocesing,* kata pada kalimatakan dicocokkan dengan *term* pada model yang di load, lalu akan dilakukan fungsi keputusan, lalu akan dipilih nilai f(x) paling tinggi dari 3 kelas (positif, negatif, atau netral), nilai f(x) tertinggi akan menjadi nilai sentimen dari suatu kalimat data uji.

|  |
| --- |
|  |

**Gambar 3.17** ***Flowchart* data uji SVM kernel linier**

1. Flowchart Data Uji SVM Kernel Polinomial

*Flowchart* ini menggambarkan aliran dataltih yang diinputkan akan melalui proses *preprocessing*, lalu sistem akan meload model SVM kernel polinomial data *training*, setelah kalimat di *preprocesing,* kata pada kalimatakan dicocokkan dengan *term* pada model yang di load, lalu akan dilakukan fungsi keputusan, lalu akan dipilih nilai f(x) paling tinggi dari 3 kelas (positif, negatif, atau netral), nilai f(x) tertinggi akan menjadi nilai sentimen dari suatu kalimat data uji.

|  |
| --- |
|  |

**Gambar 3.18** ***Flowchart* data uji SVM kernel polinomial**

1. Flowchart Data Uji SVM Kernel Sigmoid

*Flowchart* ini menggambarkan aliran datalatih yang diinputkan akan melalui proses *preprocessing*, lalu sistem akan meload model SVM kernel sigmoid data *training*, setelah kalimat di *preprocesing,* kata pada kalimatakan dicocokkan dengan *term* pada model yang di load, lalu akan dilakukan fungsi keputusan, lalu akan dipilih nilai f(x) paling tinggi dari 3 kelas (positif, negatif, atau netral), nilai f(x) tertinggi akan menjadi nilai sentimen dari suatu kalimat data uji.

|  |
| --- |
|  |

**Gambar 3.19** ***Flowchart* data uji SVM kernel sigmoid**

* + - 1. **Perancangan Basis Data**

Perancangan basis data sistem ini menggunakan *web server* xampp dengan *Data Base Management System* (DBMS) MySql. Terdapat 7 *table* yang digunakan pada *database* sistem ini, antara lain *table* “datalatihs”, *table* “dataujis”, *table* “kamuskds”, *table* “pembobotans”, *table* stopwords”, *table* “user”, dan *table* “akurasi”. Berikut detail dari 7 *table* yang digunakan :

* + - 1. **Entity Relationship Diagram (ERD)**

*Entitity Relationship Diagram* (ERD) menggambarkan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Terdapat 7 entitas dalam sistem yang dibuat, meliputi entitas user, datalatihs, pembobotanas, stopwords, kamuskds, dataujis, dan akurasi. Berikut adalah *Entitity Relationship Diagram* (ERD) dari sistem yang dibuat :

|  |
| --- |
|  |

**Gambar 3.20** ***Entity Relationship Diagram*** (ERD)

* + - 1. **Relasi Antar Tabel**

Relasi Antar Tabel (RAT) akan menggambarkan hubungan antar tabel yang berfungsi untuk mengatur operasi suatu basis data. Berikut Relasi Antar Tabel (RAT) dari sistem yang dibuat :

|  |
| --- |
|  |

**Gambar 3.21** **Relasi Antar Tabel (RAT)**

* + - 1. **Struktur Tabel Basis Data**
         1. Tabel user

Tabel *database* yang mempunyai kolom id\_user, username, dan password.

**Tabel 3.9 Tabel data *user***

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| id\_user | *Int* (11) | *Primary key* , *Auto Increment* | Id user |
| username | *Varchar* (256) |  | Username user |
| password | *Varchar* (256) |  | Password user |

* + - * 1. Tabel stopwords

Tabel *database* yang mempunyai kolom id\_stopword, dan kata\_stopword. Berfungsi menyimpan data kamus *stopword*, untuk proses *stopword removal*.

**Tabel 3.10 Tabel data *stopword***

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| id\_stopword | *Varchar* (11) | *Primary key* , *Auto Increment* | Id stopword |
| kata\_stopword | *Varchar* (256) |  | Kata stopword |

* + - * 1. Tabel kamuskds

Tabel *database* yang mempunyai kolom id\_katakd, dan katakd. Berfungsi menyimapan data kamus *stemming* kata dasar.

**Tabel 3.11 Tabel data kamus kata dasar**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| id\_katakd | *Int* (11) | *Primary key* , *Auto Increment* | Id kata dasar |
| katakd | *Varchar* (256) |  | Kata dasar |

* + - * 1. Tabel datalatihs

Tabel *database* yang mempunyai kolom *id\_datalatih*, *komentar, prepro, medsos*,dan *sentimen*. Berfungsi menyimpan data latih / data *training*.

**Tabel 3.12 Tabel data data latih**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| id\_datalatih | *Int* (11) | *Primary key* , *Auto Increment* | Id user |
| komentar | *Varchar* (256) |  | Komentar data latih |
| prepro | *Varchar* (256) |  | Preprocessing data latih |
| medsos | *Varchar* (256) |  | Sumber media sosial |
| sentimen | *Varchar* (256) |  | Label sentimen (positif, negatif, atau netral) |

1. Tabel pembobotans

Tabel *database* yang mempunyai kolom id\_pembobotan, term, docId, Count, dan bobot. Berfungsi menyimpan data latih / data *training* yang sudah dilakukan proses pembobotan.

**Tabel 3.13** **Tabel data pembobotan**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| id\_pembobotan | *Int* (11) | *Primary key* , *Auto Increment* | Id pembobotan |
| Term | *Varchar* (256) |  | Term dari data latih |
| docId | *Int* (11) |  | Nomor term ada pada dokumen ke berapa |
| Count | *Int* (11) |  | Jumlah masing – masing term pada dokumen |
| Bobot | *Float* |  | Hasil pembobotan |

1. Tabel dataujis

Tabel *database* yang mempunyai kolom id\_datauji, komentar\_mentah, komentar\_hasil, medsos, sentimen, linier, poli, dan sigmoid. Berfungsi menyimpan data uji untuk proses analisis sentimen.

**Tabel 3.14** **Tabel data data uji**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| id\_datauji | *Int* (11) | Primary key , *Auto Increment* | Id data uji |
| komentar\_mentah | *Varchar* (256) |  | Data uji awal |
| komentar\_hasil | *Varchar* (256) |  | Data uji setelah proses preprocessing |
| medsos | *Varchar* (256) |  | Sumber media sosial |
| sentimen | *Varchar* (256) |  | Label manual sentimen (positif, negatif, atau netral) |
| linier | *Varchar* (256) |  | Hasil analisis sentimen kernel linier (positif, negatif, atau netral) |
| poli | *Varchar* (256) |  | Hasil analisis sentimen kernel polinomial(positif, negatif, atau netral) |
| sigmoid | *Varchar* (256) |  | Hasil analisis sentimen kernel sigmoid (positif, negatif, atau netral) |

1. Tabel akurasi

Tabel *database* ini berisi hasil akurasi, presisi, dan recall dari metode *Support Vector Machine* (SVM) kernel linier, kernel polinomial, dan kernel sigmoid.

**Tabel 3.15 Tabel data akurasi**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama kolom | Tipe data | Constraint / batasan | Keterangan |
| akurasi1 | *Float* |  | Hasil akurasi kernel linier |
| presisi1 | *Float* |  | Hasil presisii kernel linier |
| recall1 | *Float* |  | Hasil recall kernel linier |
| akurasi2 | *Float* |  | Hasil akurasi kernel polinomial |
| presisi2 | *Float* |  | Hasil presisi kernel polinomial |
| recall2 | *Float* |  | Hasil recall kernel polinomial |
| akurasi3 | *Float* |  | Hasil akurasi kernel sigmoid |
| presisi3 | *Float* |  | Hasil presisi kernel sigmoid |
| recall3 | *Float* |  | Hasil recall kernel sigmoid |

* + - 1. **Perancangan Antarmuka Sistem**

Perancangan antarmuka merupakan mekanisme komunikasi berwujud *interface* / antarmuka antara *user* / pengguna dengan sebuah sistem atau biasa disebut dengan *User Interface* (UI). Berikut merupakan *wireframe* (kerangka untuk penataan *item - item* pada sebuah *webiste*) rancangan antarmuka dari sistem yang dibangun berupa :

1. Rancangan Halaman Pengunjung

Halaman pengunjung dapat melihat jumlah data latih dan data uji yang digunakan pada sistem, akurasi dari metode yang digunakan, serta sentimen mengenai Komisi Pemberantasan Korupsi (KPK) di twitter dan instagram.

|  |
| --- |
|  |

**Gambar 3.22** **Rancangan halaman pengunjung**

1. Rancangan Halaman Beranda (Admin)

Berisi halaman pertama yang akan tampil setelah admin melakukan login pada navbar di halaman pengunjung.

|  |
| --- |
|  |

**Gambar 3.23** **Rancangan halaman beranda (admin)**

1. Rancangan Halaman List Data Latih (Admin)

Halaman yang akan menampilkan *list* data latih yang akan digunakan di sistem. Terdapat tombol tambah data baru, hapus semua data, serta admin dapat *upload file* untuk menginput data latih

|  |
| --- |
|  |

**Gambar 3.24** **Rancangan halaman list data latih (admin)**

1. Rancangan Halaman Tambah Data Baru Data Latih (Admin)

Halaman akan menampilkan *form input* data latih. Kemudian admin dapat menyimpan dengan menekan tombol “Tambah”.

|  |
| --- |
|  |

**Gambar 3.25** **Rancangan halaman tambah data baru data latih (admin)**

1. Rancangan Halaman Edit Data Latih (Admin)

Halaman untuk menyunting atau *mengedit* data latih yang telah *diinputkan* sebelumnya.

|  |
| --- |
|  |

**Gambar 3.26** **Rancangan halaman edit data latih (admin)**

1. Rancangan Halaman List Data Stopword (Admin)

Halaman yang akan menampilkan *list* data *stopword* yang akan digunakan di sistem. Terdapat tombol tambah data baru, hapus semua data, serta admin dapat upload file untuk menginput data *stopword*.

|  |
| --- |
|  |

**Gambar 3.27** **Rancangan halaman list data stopword (admin)**

1. Rancangan Halaman Tambah Data Baru Data Stopword (Admin)

Halaman akan menampilkan *form input* data *stopword*. Kemudian admin dapat menyimpan dengan menekan tombol “Tambah”.

|  |
| --- |
|  |

**Gambar 3.28** **Rancangan halaman tambah data baru data stopword (admin)**

1. Rancangan Halaman Edit Data Stopword (Admin)

Halaman untuk menyunting atau *mengedit* data stopword yang telah *diinputkan* sebelumnya.

|  |
| --- |
|  |

**Gambar 3.29** **Rancangan halaman edit data stopword (admin)**

1. Rancangan Halaman List Data Kamus Kata Dasar (Admin)

Halaman yang akan menampilkan *list* data kamus kata dasar yang akan digunakan di sistem. Terdapat tombol tambah data baru, hapus semua data, serta admin dapat upload file untuk menginput data kamus kata dasar.

|  |
| --- |
|  |

**Gambar 3.30** **Rancangan halaman list data kamus kata dasar (admin)**

1. Rancangan Halaman Tambah Data Baru Kamus Kata Dasar (Admin)

Halaman akan menampilkan *form input* data kamus kata dasar. Kemudian admin dapat menyimpan dengan menekan tombol “Tambah”.

|  |
| --- |
|  |

**Gambar 3.31** **Rancangan halaman tambah data baru kamus kata dasar (admin)**

1. Rancangan Halaman Edit Kamus Kata Dasar (Admin)

Halaman untuk menyunting atau *mengedit* data kamus kata dasar yang telah *diinputkan* sebelumnya.

|  |
| --- |
|  |

**Gambar 3.32** **Rancangan halaman edit kamus kata dasar (admin)**

1. Rancangan Halaman Pembobotan (Admin)

Halaman untuk melakukan pembobotan terhadap data latih yang telah *diinputkan*. Admin dapat menekan tombol “Proses” untuk menjalankan proses pembobotan lalu hasil pembobotan akan tampil di *list*. Admin juga dapat menakan tombol “Hapus Semua Data Pembobotan” untuk menghapus pembobotan, dan admin dapat menekan tombol “Training Data Latih” setelah menekan tombol “Proses” untuk melatih data latih yang telah diberi bobot.

|  |
| --- |
|  |

**Gambar 3.33** **Rancangan halaman pembobotan (admin)**

1. Rancangan Halaman Hasil Analisis Sentimen (Admin)

Halaman yang akan menampilkan *list* data hasil analisis sentimen. Terdapat tombol “Tambah Data Baru”, “Hapus Semua Data”, serta admin dapat upload file untuk menginput data *stopword*. Dihalaman ini data uji yang *diinputkan* akan langsung di ketahui hasil analisis sentimennya.

|  |
| --- |
|  |

**Gambar 3.34** **Rancangan halaman list hasil analisis sentimen (admin)**

1. Rancangan Halaman Tambah Data Analisis Sentimen / Data uji

Halaman akan menampilkan *form input* data analisis sentimen / data uji. Kemudian admin dapat menyimpan dengan menekan tombol “Tambah”.

|  |
| --- |
|  |

**Gambar 3.35** **Rancangan halaman tambah data analisis sentimen (admin)**

1. Rancangan Halaman Edit Data Analisis Sentimen / Data Uji

Halaman untuk menyunting atau *mengedit* data analisis sentimen / data uji yang telah *diinputkan* sebelumnya.

|  |
| --- |
|  |

**Gambar 3.36** **Rancangan halaman edit data analisis sentimen (admin)**