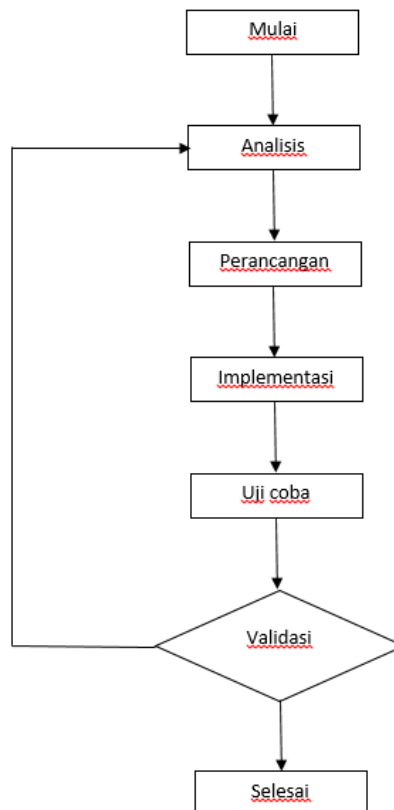


BAB III METODOLOGI PENELITIAN

3.1 Kerangka Alur Penelitian

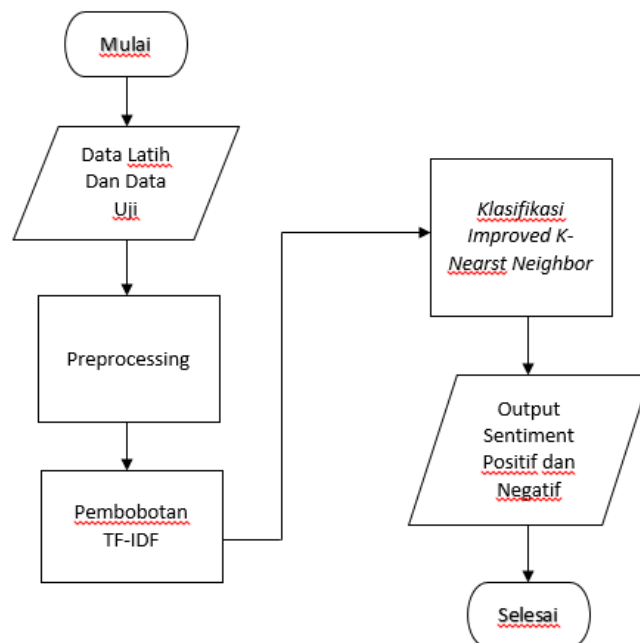
System Development Life Cycle merupakan proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem yang umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana, analisis, perancangan, implementasi, uji coba, validasi, dan evaluasi. Pengembangan perangkat lunak model SDLC dapat dilihat pada gambar (3.1) di bawah ini:



Gambar 3.1. Pengembangan Perangkat Lunak Model SDLC

3.2 Tahap Analisis

Pada tahap analisis ini akan dijelaskan diawali dengan mengambil data opini pada media sosial Twitter kemudian di olah dengan melakukan preprocessing pada data sampai informasi apa yang dihasilkan oleh metode klasifikasi Improved K-Nearest Neighbor. Hasil dari preprocessing dilakukan perhitungan bobot pada setiap kata menggunakan tf-idf, selanjutnya dilakukan proses klasifikasi dengan algoritma improved k-nearest neighbor pada tahap ini dilakukan untuk mengklasifikasikan menjadi 2 kategori yaitu opini positif dan negatif. Tahap analisis terdapat pada gambar (3.2) di bawah ini:

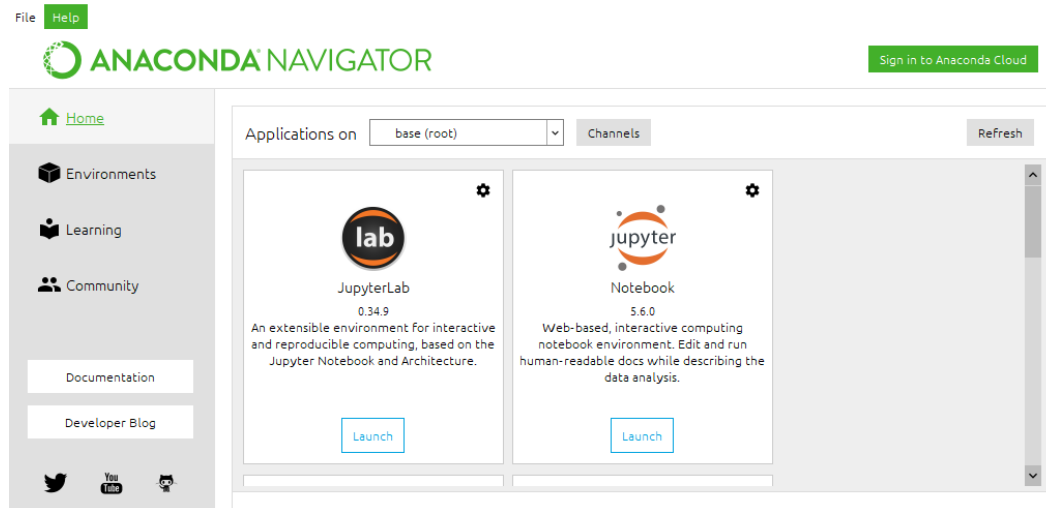


Gambar 3.2. Tahap Analisis

3.3 Pengumpulan Data

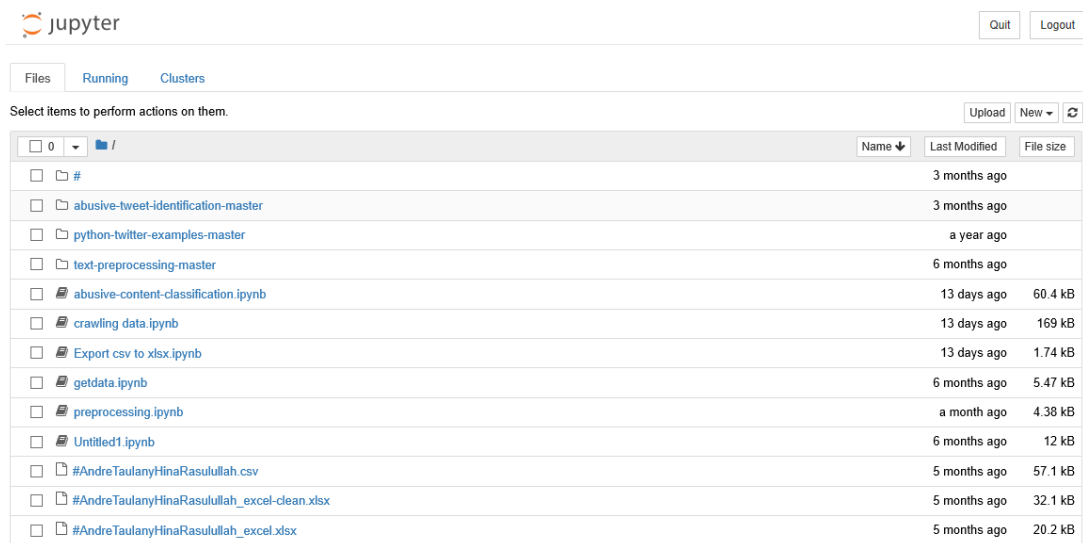
Proses pengumpulan data atau crawling data Twitter ini menggunakan *API key Twitter* dan proses pengumpulan data Twitter menggunakan bahasa pemrograman Python. *API key Twitter* adalah *Application Programming Interface* (API) di mana layanan mencakup serangkaian perintah, fungsi, komponen, dan protokol yang membantu pemrogram membuat sistem perangkat lunak. *API key Twitter*

sendiri memiliki *consumer keys*, *consumer secret*, *access key*, dan *access secret*. *Consumer keys*, *access keys*, dan *access secret* ini digunakan untuk mengakses data Twitter. Selain memerlukan *API key Twitter* untuk mengambil data Twitter, tetapi juga menggunakan alat analisis sentimen pendukung seperti *anaconda* sebagai distribusi Python. Tampilan dari *anaconda* pada Gambar 3.3.



Gambar 3.3. Tampilan Anaconda

Dengan menggunakan *anaconda* sebagai tools karena *anaconda* sudah memiliki Jupyter Notebook. Jupyter Notebook adalah evolusi dari Ipython atau Interactive Python. Notebook Jupyter adalah editor dalam bentuk aplikasi web yang berjalan di localhost komputer, dan Jupyter Notebook dapat melakukan beberapa hal, seperti menulis kode Python, *equations*, dan visualisasi, dan juga dapat digunakan sebagai *markdown editor*. Tampilan dari Jupyter Notebook pada Gambar 3.4.



Gambar 3.4. Tampilan Jupyter Notebook

Dengan adanya *Jupyter Notebook*, sekarang dapat melakukan proses pendeklarasian *library tweepy*, *library csv*, *library pandas*. *Library tweepy* merupakan suatu API yang disediakan oleh pihak Twitter guna mengakses data-data yang ada di Twitter. *Library csv* adalah library yang menyediakan layanan baca dan menulis suatu data bertipe file csv. *Library pandas* merupakan library pada *python* yang digunakan untuk pengolahan data. Setelah itu masukkan *consumer key*, *consumer secret*, *access key*, dan *access secret* yang telah didapatkan.

Setelah proses *crawling* data selesai dan berhasil disimpan dalam bentuk file dokumen csv, kemudian masuk ke dalam proses pelabelan data tweet. Dalam proses pelabelan data ini, seharusnya untuk menentukan nilai tersebut dilakukan oleh ahli bahasa atau yang terkait dibidangnya dalam menentukan sentimen tweet yang terdapat dari hasil data *crawling*. Namun demikian disini pada saat proses pelabelan data hanya dilakukan dengan tiga orang mahasiswa yang bukan pakar dari bahasa ataupun psikologi dibidangnya. Konsekuensi dari penelitian ini dalam melakukan proses pelabelan data adalah terdapatnya beberapa tweet yang tidak sama ataupun sesuai dengan sentimen analisis sebenarnya. Karena proses pelabelan data Twitter ini dilakukan secara manual, dan juga dalam proses pelabelan ini secara manual membutuhkan waktu yang tidak cepat, hal tersebut

menjadi kelemahan pada saat jumlah data yang akan di proses pelabelan sangat banyak.

Data yang digunakan dalam penelitian ini terbagi menjadi dua jenis, yaitu data latih dan data uji. Data latih yang digunakan diambil dari kumpulan komentar semua postingan ataupun hashtag yang ada di akun Pedulilindungi yang berjumlah 80%. Data inilah yang akan digunakan untuk membentuk analisis sentimen. 20% dari keseluruhan data, nantinya akan digunakan sebagai data uji. Detail dari kumpulan dataset yang didapat dari hasil crawling dapat dilihat pada Tabel 3.1.

Tabel 3.1. Data Latih dan Data Uji

Media Sosial	Data Latih	Data Uji
Twitter	800	200

File dari data Pedulilindungi dapat di lihat pada link dibawah ini.

<https://drive.google.com/drive/folders/1pawn-a-xtkYwUN29FEMAUcNSe6KEbpN4?usp=sharing>

3.4 Pre-processing

Pre-processing dilakukan untuk menghindari data yang kurang sempurna, gangguan pada data, dan data-data yang tidak konsisten. Adapun langkah-langkah Preprocessing dalam penelitian ini yaitu cleansing, case folding, tokenizing, stopword removal, stemming.

3.4.1 *Cleansing*

Proses membersihkan dokumen dari kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah karakter HTML, kata kunci, ikon emoji, hashtag (#), username (@username), url (http://situs.com), dan email (nama@situs.com). Hasil cleansing dapat dilihat pada Tabel (3.2) di bawah ini:

Tabel 3.2. Hasil Cleansing

Kode	Input Komentar Asli	Output <i>Cleansing</i>
D1	Kenapa siiii lg sibuk tp malah Suusah login!! @pedulilindungi	Kenapa siiii lg sibuk tp malah Suusah login
D2	Susah di buka aplikasi Nya @pedulilindungi	Susah di buka aplikasi Nya
D3	Aplikasinya cukup baik dan mudah digunakan.. trskan #pedulilindungi	Aplikasinya cukup baik dan mudah digunakan trskan
D4	Sangat cepet keluar Sertifikatnya.. Teruskan !!	Sangat cepet keluar Sertifikatnya Teruskan
D5	Bagaimana ini banyak yg nanya kok ga pada dibales.. bukannya di bantu di permudah !	Bagaimana ini banyak yg nanya kok ga pada dibales bukannya di bantu di permudah
D6	Teruslah memberi Update pada aplikasi agar semakin lebih baik lagi	Teruslah memberi Update pada aplikasi agar semakin lebih baik lagi

3.4.2 Case folding

Case folding yaitu pengubahan bentuk huruf menjadi huruf kecil. Kalimat yang digunakan adalah kalimat hasil *cleansing*. Hasil dari *cleansing* akan diperiksa apakah karakter menggunakan huruf kapital, maka huruf tersebut akan diubah menjadi huruf kecil. Hasil *case folding* dapat dilihat pada Tabel (3.3) di bawah ini:

Tabel 3.3. Hasil Case Folding

Kode	Input Hasil <i>Cleansing</i>	Output <i>Case Folding</i>
D1	Kenapa siiii lg sibuk tp malah Suusah login	kenapa siiii lg sibuk tp malah suusah login
D2	Susah di buka aplikasi Nya	susah di buka aplikasi nya
D3	Aplikasinya cukup baik dan mudah digunakan trskan	aplikasinya cukup baik dan mudah digunakan trskan
D4	Sangat cepet keluar Sertifikatnya Teruskan	sangat cepet keluar sertifikatnya teruskan
D5	Bagaimana ini banyak yg nanya kok ga pada dibales bukannya di bantu di permudah	bagaimana ini banyak yg nanya kok ga pada dibales bukannya di bantu di permudah
D6	Teruslah memberi Update pada aplikasi agar semakin lebih baik lagi	teruslah memberi update pada aplikasi agar semakin lebih baik lagi

3.4.3 Tokenizing

Tokenizing adalah proses memecah teks menjadi kata tunggal dan penghapusan tanda baca serta angka, sesuai dengan kamus data yang telah ditentukan. Pada penelitian ini fitur yang digunakan dalam memecah teks adalah unigram yaitu token yang terdiri hanya satu kata. Memotong setiap kata dalam kalimat berdasarkan pemisah kata seperti titik (.), koma (,), dan spasi. Bagian yang hanya memiliki satu karakter non alfabet dan angka akan dibuang. Sehingga menghasilkan kata-kata penyusun kalimat. Kalimat yang digunakan adalah kalimat dari hasil *case folding*. Hasil *tokenizing* dapat dilihat pada Tabel (3.4) di bawah ini:

Tabel 3.4. Tokenizing

Kode	Hasil <i>Case Folding</i>	Output <i>Tokenizing</i>
D1	kenapa siii lg sibuk tp malah susah login	kenapa, siii, lg, sibuk, tp, malah, susah, login
D2	susah di buka aplikasi nya	susah, di, buka, aplikasi, nya,
D3	aplikasinya cukup baik dan mudah digunakan trskan	aplikasinya, cukup, baik, dan, mudah, digunakan, trskan
D4	sangat cepet keluar sertifikatnya teruskan	sangat, cepet, keluar, sertifikatnya teruskan
D5	bagaimana ini banyak yg nanya kok ga pada dibales bukannya di bantu di permudah	bagaimana, ini, banyak, yg, nanya, kok, ga, pada, dibales, bukannya, di, bantu, permudah
D6	teruslah memberi update pada aplikasi agar semakin lebih baik lagi	teruslah, memberi, update, pada, aplikasi, agar, semakin, lebih, baik, lagi

3.4.4 *Stopword removal*

Stopwords removal adalah proses menghilangkan kata tidak penting dalam text. Hal ini dilakukan untuk memperbesar akurasi dari pembobotan term. Untuk proses ini, diperlukan suatu kamus kata-kata yang bisa dihilangkan. Dalam Bahasa Indonesia, misalnya kata: dan, atau, mungkin, ini, itu, dll adalah kata-kata yang dapat dihilangkan. Pada penelitian ini menggunakan kamus *stopword* Tala dimana datanya berjumlah 748 data. Kamus *stopword* Tala telah banyak dimanfaatkan oleh peneliti untuk membuang kata-kata yang tidak penting dalam bahasa Indonesia. Contoh *stopword* pada kamus Tala dapat dilihat pada Tabel (3.5) di bawah ini:

Tabel 3.5. Kamus Stopword Tala

<i>Stopword Tala</i>			
Di	Adalah	Beberapa	Khususnya
Kapan	Adapun	Sangat	Ini
Pula	Aku	Besar	Kok
Dari	Bagaimana	Entah	Tapi
Karena	Kenapa	Ketika	Dan

Dalam penelitian ini, tidak semua kata yang ada didalam kamus Tala pada Tabel (3.5) di atas digunakan sebagai *stopword*. Hal ini dikarenakan kata tersebut berpengaruh terhadap makna atau nilai sentimen khususnya untuk sentimen yang bernilai negatif. Hasil *stopword removal* terdapat pada Tabel (3.6) dibawah ini:

Tabel 3.6. Hasil Stopword Removal

Kode	Input Hasil <i>Tokenizing</i>	Output <i>Stopword Removal</i>
D1	kenapa, siii, lg, sibuk, tp, malah, susah, login	siii, lg, sibuk, tp, susah, login
D2	susah, di, buka, aplikasi, nya,	susah, buka, aplikasi, nya,
D3	aplikasinya, cukup, baik, dan, mudah, digunakan, trskan	aplikasinya, cukup, baik, mudah, gunakan, trskan
D4	sangat, cepet, keluar, sertifikatnya teruskan	cepat, keluar, sertifikatnya, teruskan
D5	bagaimana, ini, banyak, yg, nanya, kok, ga, pada, dibales, bukannya, di, bantu, permudah	banyak, nanya, ga, bales, bukannya, bantu, permudah
D6	teruslah, memberi, update, pada, aplikasi, agar, semakin, lebih, baik, lagi	teruslah, memberi, update, aplikasi, semakin, baik

3.4.5 Stemming

Stemming adalah pengubahan kata ke bentuk kata dasar atau penghapusan imbuhan. *Stemming* disini menggunakan kamus daftar kata berimbuhan yang mempunyai kata dasarnya dengan membandingkan kata-kata yang ada di dalam komentar dengan daftar kamus stem. Data kata dasar didapatkan dari Kamus Bahasa Indonesia online dimana datanya berjumlah 29932 data. Algoritma *stemming* yang digunakan adalah algoritma *stemming* Nazief & Andriani. Hasil *stemming* dapat dilihat pada Tabel (3.7) di bawah ini:

Tabel 3.7. Hasil Stemming

Kode	Input Hasil <i>Stopword Removal</i>	Output <i>Stemming</i>
D1	siii, lg, sibuk, tp, susah, login	sibuk, tp, ga, susah, login
D2	susah, buka, aplikasi, nya	susah, buka, aplikasi
D3	aplikasinya, cukup, baik, mudah, gunakan, trskan	aplikasi, baik, mudah, gunakan, trs
D4	cepat, keluar, sertifikatnya, teruskan	cepat, keluar, sertifikat, terus
D5	banyak, nanya, ga, bales, bukannya, bantu, permudah	banyak, nanya, ga, bales, bukan, bantu, mudah
D6	teruslah, memberi, update, aplikasi, semakin, baik	terus, memberi, update, aplikasi, semakin, baik

3.5 Pembobotan TF-IDF

Metode TF-IDF merupakan metode untuk menghitung bobot dari kata yang digunakan pada information retrieval. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat. Metode ini akan menghitung nilai Term Frequency (TF) dan Inverse Document Frequency (IDF) pada setiap token (kata) disetiap dokumen dalam kokrpus. Term frequency (TF) adalah jumlah kemunculan kata pada suatu dokumen. Semakin banyak suatu kata muncul pada dokumen, maka

semakin besar kata berpengaruh pada dokumen tersebut. Sebaliknya, semakin sedikit suatu kata muncul pada dokumen, maka semakin kecil kata tersebut berpengaruh pada dokumen tersebut. Inverse document frequency (IDF) adalah pembobotan kata yang didasarkan pada banyaknya dokumen yang mengandung kata tertentu. Pada penelitian ini digunakan metode TF-IDF persamaan (2.1) dan (2.2) sebagai proses pembobotan yaitu akan dilakukan pembobotan pada setiap kata berdasarkan tingkat kepentingan didalam sekumpulan dokumen masukan. Pembobotan ini bertujuan untuk memberikan nilai pada suatu kata yang dimana nilai dari kata tersebut akan dijadikan sebagai input pada proses klasifikasi. Terdapat dua contoh data latih dan data uji pada Tabel (3.8) dan Tabel (3.9).

Tabel 3.8. Contoh Data Latih

Kode	Sebelum <i>Pre – Processing</i>	Setelah <i>Pre – Processing</i>	Kelas
D1	Kenapa siii lg sibuk tp malah Suusah login!! @pedulilindungi	sibuk, tapi, susah, login	Negatif
D2	Susah di buka aplikasi Nya @pedulilindungi	susah, buka, aplikasi	Negatif
D3	Aplikasinya cukup baik dan mudah digunakan.. trskan #pedulilindungi	aplikasi, baik, mudah, gunakan, terus	Positif
D4	Sangat cepet keluar Sertifikatnya.. Teruskan !!	cepat, keluar, sertifikat, terus	Positif
D5	Bagaimana ini banyak yg nanya kok ga pada dibales.. bukannya di bantu di permudah !	banyak, tanya, ga, balas, bukan, bantu, mudah	Negatif

Tabel 3.9. Contoh Data Uji

Kode	Sebelum <i>Pre – Processing</i>	Setelah <i>Pre – Processing</i>	Kelas
D6	Teruslah memberi Update pada aplikasi agar semakin lebih baik lagi	terus, memberi, update, aplikasi, semakin, baik	?

Berdasarkan Tabel (3.10) dan Tabel (3.11) di bawah, D1 sampai D6 merupakan data yang akan diuji bobot dokumennya. D1 sampai dengan D5 adalah data yang sudah diketahui kategori/kelasnya, sedangkan D6 adalah data yang masih belum diketahui kategori sentimennya dan yang akan diuji. Untuk menentukan ke kelas manakah D6 akan dimasukkan. Pertama hitung bobot setiap kata dengan metode TF-IDF.

Tahap 1: Menghitung jumlah frekuensi tiap kata pada tiap dokumen (TF-DF)

Tabel 3.10. Perhitungan TF dan IDF

No	Token	TF						DF
		D6	D1	D2	D3	D4	D5	
1	sibuk	0	1	0	0	0	0	1
2	tapi	0	1	0	0	0	0	1
3	susah	0	1	1	0	0	0	2
4	login	0	1	0	0	0	0	1
5	buka	0	0	1	0	0	0	1
6	aplikasi	1	0	1	1	0	0	3
7	baik	1	0	0	1	0	0	2
8	mudah	0	0	0	1	0	1	2
9	gunakan	0	0	0	1	0	0	1
10	terus	1	0	0	1	1	0	3
11	cepat	0	0	0	0	1	0	1
12	keluar	0	0	0	0	1	0	1
13	banyak	0	0	0	0	0	1	1
14	tanya	0	0	0	0	0	1	1
15	ga	0	0	0	0	0	1	1
16	balas	0	0	0	0	0	1	1
17	bukan	0	0	0	0	0	1	1
18	bantu	0	0	0	0	0	1	1
19	memberi	1	0	0	0	0	0	1
20	semakin	1	0	0	0	0	0	1

Tahap 2: Menghitung IDF menggunakan persamaan (2.1)

Tabel 3.11. Perhitungan IDF

No	Token	IDF
		LOG (D/df)
1	sibuk	0,778
2	tapi	0,778
3	susah	0,477
4	login	0,778
5	buka	0,778
6	aplikasi	0,301
7	baik	0,477
8	mudah	0,477
9	gunakan	0,778
10	terus	0,301
11	cepat	0,778
12	keluar	0,778
13	banyak	0,778
14	tanya	0,778
15	ga	0,778
16	balas	0,778
17	bukan	0,778
18	bantu	0,778
19	memberi	0,778
20	semakin	0,778

Tahap 3: Menghitung TF-IDF menggunakan persamaan (2.2)

Tabel 3.12. Perhitungan TF-IDF

Wdt = TF.IDF					
D6	D1	D2	D3	D4	D5
0	0,778	0	0	0	0
0	0,778	0	0	0	0
0	0,477	0,477	0	0	0
0	0,778	0	0	0	0
0	0	0,778	0	0	0
0,301	0	0,301	0,301	0	0
0,477	0	0	0,477	0	0
0	0	0	0,477	0	0,477
0	0	0	0,778	0	0
0,301	0	0	0,301	0,301	0
0	0	0	0	0,778	0
0	0	0	0	0,778	0
0	0	0	0	0	0,778
0	0	0	0	0	0,778
0	0	0	0	0	0,778
0	0	0	0	0	0,778
0	0	0	0	0	0,778
0,778	0	0	0	0	0
0,778	0	0	0	0	0

3.6 Klasifikasi Improved K-nearest neighbor

Penentuan k-values yang tepat diperlukan agar didapatkan akurasi yang tinggi dalam proses kategorisasi dokumen uji. Algoritma *Improved k-Nearest Neighbor* melakukan modifikasi dalam penentuan k-values. Dimana penetapan k-values tetap dilakukan, hanya saja tiap-tiap kategori memiliki k-values yang berbeda. Perbedaan k-values yang dimiliki pada setiap kategori disesuaikan dengan besar-kecilnya jumlah dokumen latih yang dimiliki kategori tersebut. Sehingga ketika k-values semakin tinggi, hasil kategori tidak terpengaruh pada kategori yang memiliki jumlah dokumen latih yang lebih besar. Untuk menghitung similaritas

antara dua dokumen menggunakan metode *Cosine Similarity* (CosSim). Dipandang sebagai pengukuran (*similarity measure*) antara vector dokumen (D) dengan vektor *query* (Q). Semakin sama suatu vektor dokumen dengan vector *query* maka dokumen dapat dipandang semakin sesuai dengan *query*. Kemiripan antar dokumen dapat menggunakan persamaan (2.6). Hasil dari perhitungan kemiripan diurutkan kemudian akan disimpan untuk proses selanjutnya.

Tahap 1 : Hasil perkalian dari setiap dokumen dijumlahkan pada Tabel (3.13), kemudian hitung panjang setiap dokumen dengan cara kuadratkan bobot setiap kata pada Tabel (3.14) dalam setiap dokumen, jumlahkan nilai kuadrat dan terakhir akarkan.

Tabel 3.13. Perhitungan Perkalian Skalar

No	Token	WD6*WDi				
		D6*D1	D6*D2	D6*D3	D6*D4	D6*D5
1	sibuk	0	0	0	0	0
2	tapi	0	0	0	0	0
3	susah	0	0	0	0	0
4	login	0	0	0	0	0
5	buka	0	0	0	0	0
6	aplikasi	0	0,0906	0,0906	0	0
7	baik	0	0	0,2276	0	0
8	mudah	0	0	0	0	0
9	gunakan	0	0	0	0	0
10	terus	0	0	0,0906	0,0906	0
11	cepat	0	0	0	0	0
12	keluar	0	0	0	0	0
13	banyak	0	0	0	0	0
14	tanya	0	0	0	0	0
15	ga	0	0	0	0	0
16	balas	0	0	0	0	0
17	bukan	0	0	0	0	0
18	bantu	0	0	0	0	0
19	memberi	0	0	0	0	0
20	semakin	0	0	0	0	0
Jumlah		0	0,0906	0,4089	0,0906	0

Tabel 3.14. Perhitungan Panjang Vektor

No	Token	Panjang Vektor					
		D6	D1	D2	D3	D4	D5
1	sibuk	0	0,606	0	0	0	0
2	tapi	0	0,606	0	0	0	0
3	susah	0	0,2276	0,2276	0	0	0
4	login	0	0,606	0	0	0	0
5	buka	0	0	0,606	0	0	0
6	aplikasi	0,0906	0	0,0906	0,0906	0	0
7	baik	0,2276	0	0	0,2276	0	0
8	mudah	0	0	0	0,2276	0	0,2276
9	gunakan	0	0	0	0,606	0	0
10	terus	0,0906	0	0	0,0906	0,0906	0
11	cepat	0	0	0	0	0,606	0
12	keluar	0	0	0	0	0,606	0
13	banyak	0	0	0	0	0	0,606
14	tanya	0	0	0	0	0	0,606
15	ga	0	0	0	0	0	0,606
16	balas	0	0	0	0	0	0,606
17	bukan	0	0	0	0	0	0,606
18	bantu	0	0	0	0	0	0,606
19	memberi	0,606	0	0	0	0	0
20	semakin	0,606	0	0	0	0	0
Jumlah		1,620	2,044	0,924	1,242	1,302	3,861
SQRT		1,27	1,43	0,96	1,11	1,14	1,96

Cosine Similarity

Kemudian hitung *similarity* Dokumen 6 dengan Dokumen 1,2,3,4,5

$$\text{Cos (D6, D1)} = 0 / (1,27 * 1,43) = 0$$

$$\text{Cos (D6, D2)} = 0,0906 / (1,27 * 0,96) = 0,074$$

$$\text{Cos (D6, D3)} = 0,4089 / (1,27 * 1,11) = 0,288$$

$$\text{Cos (D6, D4)} = 0,0906 / (1,27 * 1,14) = 0,062$$

$$\text{Cos (D6, D5)} = 0 / (1,27 * 1,96) = 0$$

Hasil perhitungan terdapat pada Tabel (3.15) dibawah:

Tabel 3.15. Hasil Perhitungan Cosine Similarity

D1	D2	D3	D4	D5
0	0,074	0,288	0,062	0

Tahap 2: hasil pengurutan perhitungan similarity pada Tabel (3.16) dari yang terbesar hingga yang terkecil:

Tabel 3.16. Hasil Pengurutan Perhitungan Similarity

D3	D2	D4	D1	D5
1	2	3	4	5

3.6.1 Penerapan Klasifikasi *Improved K-nearest neighbor*

Selanjutnya pada algoritma *Improved K-nearest neighbor* hitung nilai n (k-values baru) pada masing-masing kategori dengan menggunakan persamaan (2.7). Hasil perhitungan n dapat dilihat pada Tabel (3.17) dan (3.18) di bawah.

Tabel 3.17. Jumlah Data Latih

Data Latih		
Positif	Negatif	Jumlah
2	3	5

$$n = 3 \cdot 2/3 = 2 \text{ Positif}$$

$$n = 3 \cdot 3/3 = 3 \text{ Negatif}$$

Tabel 3.18. K-Baru

Nilai K	n (k-Baru)	
	Positif	Negatif
3	2	3

Kemudian hitung perbandingan kemiripan dokumen uji dengan dokumen latih pada tiap kategori menggunakan persamaan (2.5).

Jumlahkan nilai similaritas sebanyak top n tetangga yang termasuk dalam suatu kategori.

$$\sum \text{CosSim positif} = D3 + D4 = 0,228 + 0,062 = 0,29$$

$$\sum \text{CosSim negatif} = D1 + D2 + D5 = 0 + 0,074 + 0 = 0,074$$

Selanjutnya hitung penjumlahan nilai similaritas sebanyak top n tetangga pada Data Latih.

$$\begin{aligned} \sum \text{CosSim data latih} &= D3 + D2 + D4 \\ &= 0,288 + 0,074 + 0,062 \\ &= 0,424 \end{aligned}$$

Selanjutnya menghitung nilai maksimum perbandingan antara kemiripan D6 dengan dokumen latih sebanyak top n tetangga pada suatu kategori kemiripan D6 pada data latih.

$$P_{(x,cm)} \text{ positif} = 0,29 / 0,424 = 0,684$$

$$P_{(x,cm)} \text{ negatif} = 0,074 / 0,424 = 0,175$$

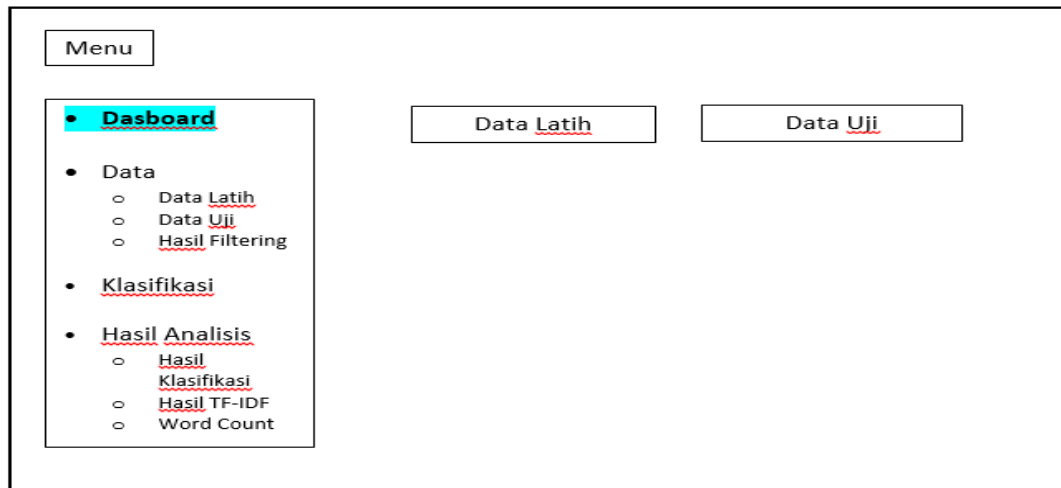
Nilai maksimum merupakan kategori dari D6, sehingga D6 diklasifikasikan sebagai kategori positif.

3.7 Tahap Perancangan

Tahap perancangan merupakan tahapan lanjut dari tahap analisis dimana dalam tahap ini disajikan desain dari aplikasi seperti perancangan antarmuka, dan perancangan data base yang akan diterapkan kedalam sistem analisis yang akan dibuat.

3.7.1 Rancangan Halaman Dashboard

Rancangan halaman dashboard berisikan informasi mengenai sistem. Tampilan halaman dashboard dapat dilihat pada Gambar 3.5.



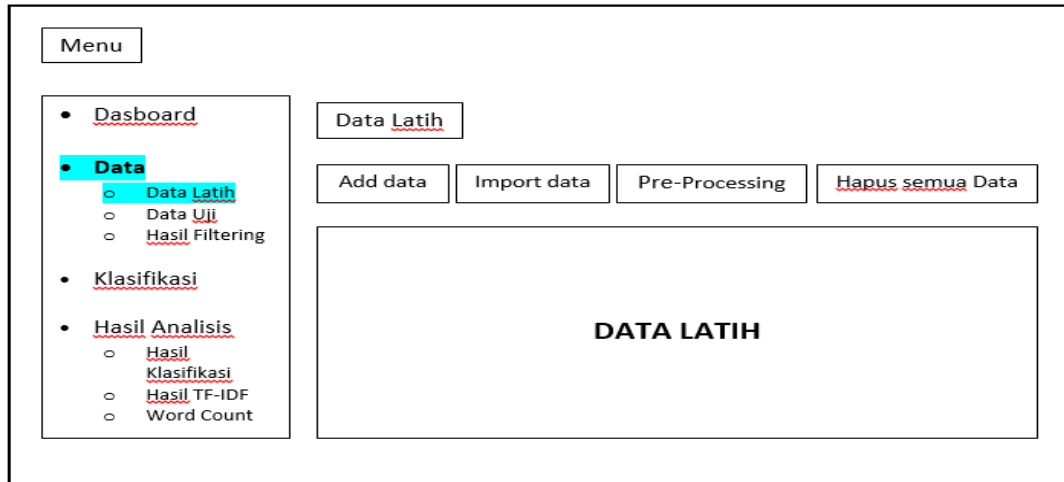
Gambar 3.5. Rancangan Halaman Menu Dashboard

Keterangan:

1. Menu dashboard merupakan halaman awal ketika sistem dijalankan.
2. Di dalam menu dashboard terdapat beberapa tampilan salah satunya ada Data Latih dan Data Uji yang berfungsi untuk memberitahukan berapa data latih dan data Uji yang sudah di import.

3.7.2 Rancangan Halaman Menu Data

Halaman menu data ini berfungsi untuk menampilkan beberapa menu yaitu: Data Latih, Data Uji, dan Hasil Filtering yang dapat dilihat pada Gambar 3.6.



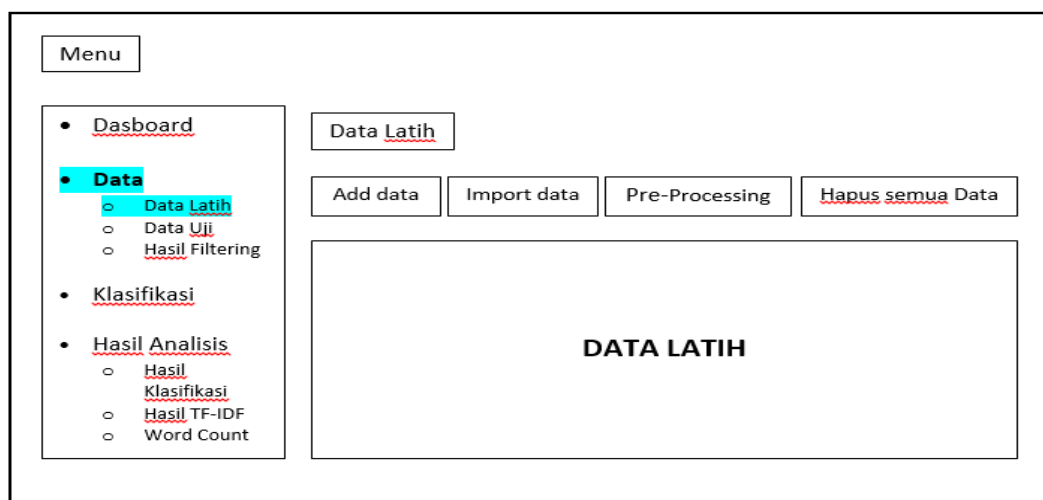
Gambar 3.6. Rancangan Halaman Menu Data

Keterangan:

1. Menu data berfungsi untuk menampilkan beberapa menu yaitu data latih, data uji, dan hasil filtering.

3.7.3 Rancangan Halaman Data Latih

Halaman ini berfungsi untuk menampilkan data latih yang sudah di crawling dari *tweet* mention Pedulilindungi. Halaman data latih dapat dilihat pada Gambar 3.7.



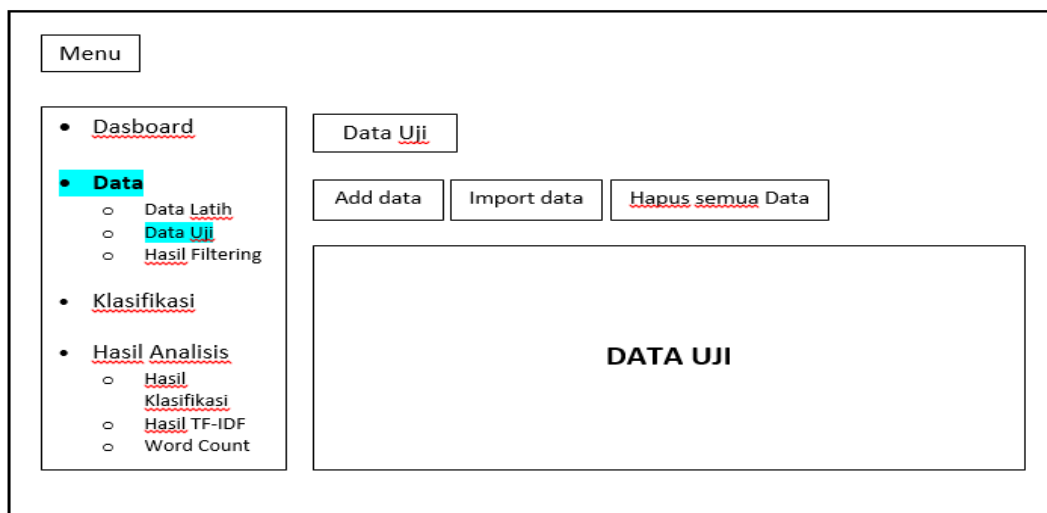
Gambar 3.7. Rancangan Halaman Data Latih

Keterangan:

1. Menu data latih berfungsi untuk menampilkan data latih yang sebelumnya sudah di *crawling*. Pada menu ini terdapat 4 *button* penting yaitu add data, import data csv, pre-processing dan menghitung tf-idf, dan hapus semua data.
2. *Button* “add data” berfungsi untuk membuat data secara manual.
3. *Button* “import data csv” berfungsi untuk mengimport file data csv.
4. *Button* “pre-processing dan menghitung tf-idf” berfungsi untuk menjalankan proses pre-processing sekaligus menghitung tf-idf.
5. *Button* “hapus semua data” berfungsi untuk menghapus semua data dengan cepat.

3.7.4 Rancangan Halaman Data Uji

Halaman ini berfungsi untuk menampilkan data uji yang sudah di *crawling* dari tweet mention Pedulilindungi. Halaman data latih dapat dilihat pada Gambar 3.8.



Gambar 3.8. Rancangan Halaman Data Uji

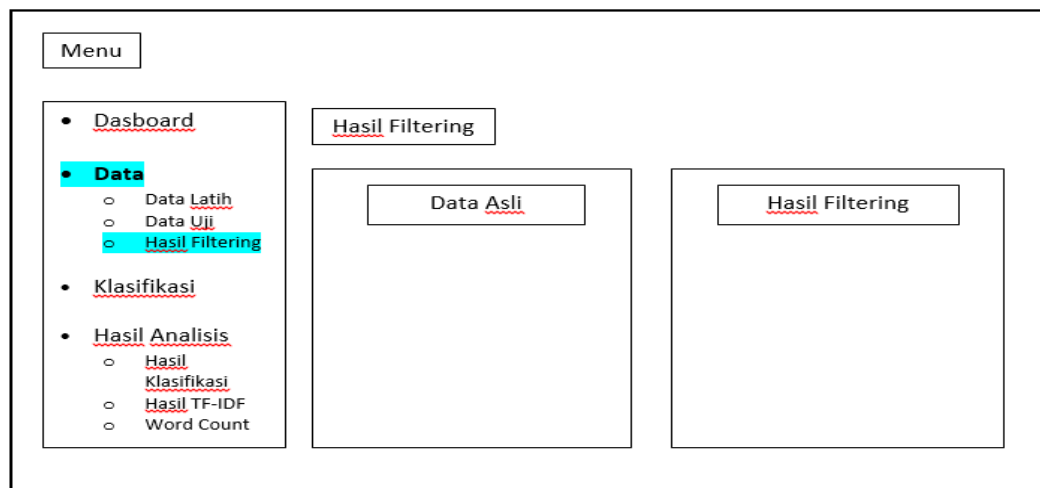
Keterangan:

1. Menu data uji berfungsi untuk menampilkan data uji yang sebelumnya sudah di *crawling*. Pada menu ini terdapat 3 *button* penting yaitu add data, import data csv, dan hapus semua data.

2. *Button* “add data” berfungsi untuk membuat data secara manual.
3. *Button* “import data csv” berfungsi untuk mengimport file data csv.
4. *Button* “hapus semua data” berfungsi untuk menghapus semua data dengan cepat.

3.7.5 Rancangan Halaman Hasil Filtering

Halaman ini berfungsi untuk menampilkan hasil dari pre-processing dan perhitungan tf-idf. Rancangan halaman hasil filtering dapat dilihat pada Gambar 3.9.



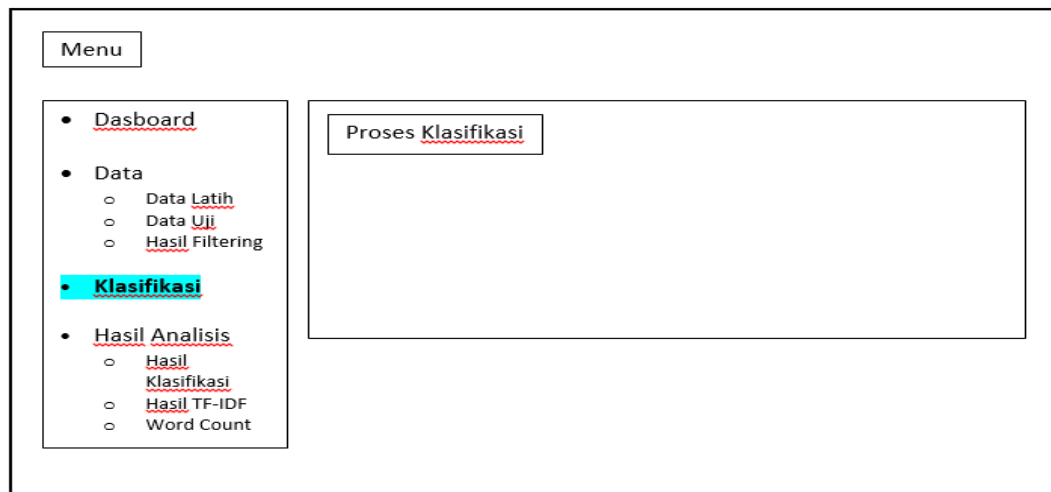
Gambar 3.9. Rancangan Halaman Hasil Filtering

Keterangan:

1. Menu hasil filtering berfungsi untuk menampilkan hasil dari pre-processing dan perhitungan tf-idf.
2. Kemudian terdapat tabel data asli dan hasil filtering.

3.7.6 Rancangan Halaman Klasifikasi

Rancangan halaman ini berfungsi untuk menampilkan *button* proses klasifikasi yang digunakan untuk memproses klasifikasi data dan dapat dilihat pada Gambar 3.10.



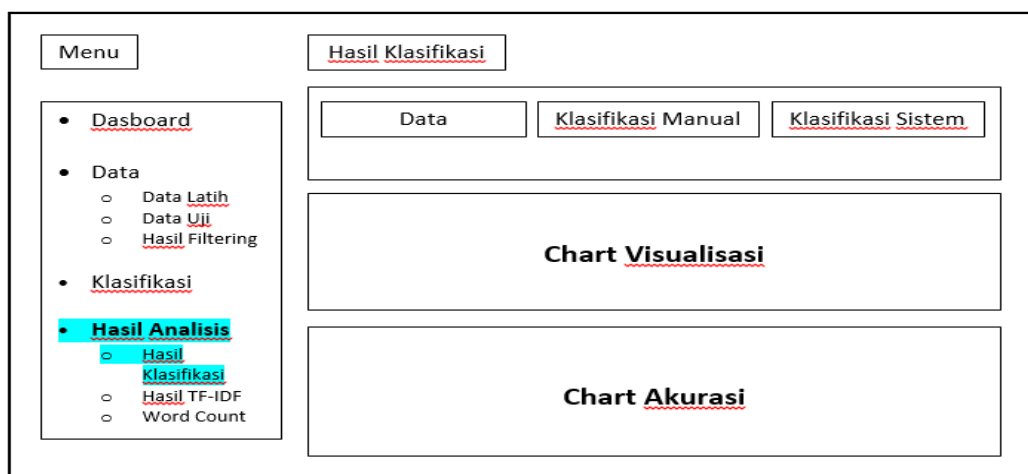
Gambar 3.10. Rancangan Halaman Klasifikasi

Keterangan:

1. Menu Klasifikasi berfungsi untuk menjalankan proses klasifikasi.

3.7.7 Rancangan Halaman Hasil Analisis

Rancangan halaman hasil analisis ini berfungsi untuk menampilkan beberapa tampilan yaitu: hasil klasifikasi, hasil tf-idf, dan word count yang dapat dilihat pada Gambar 3.11.



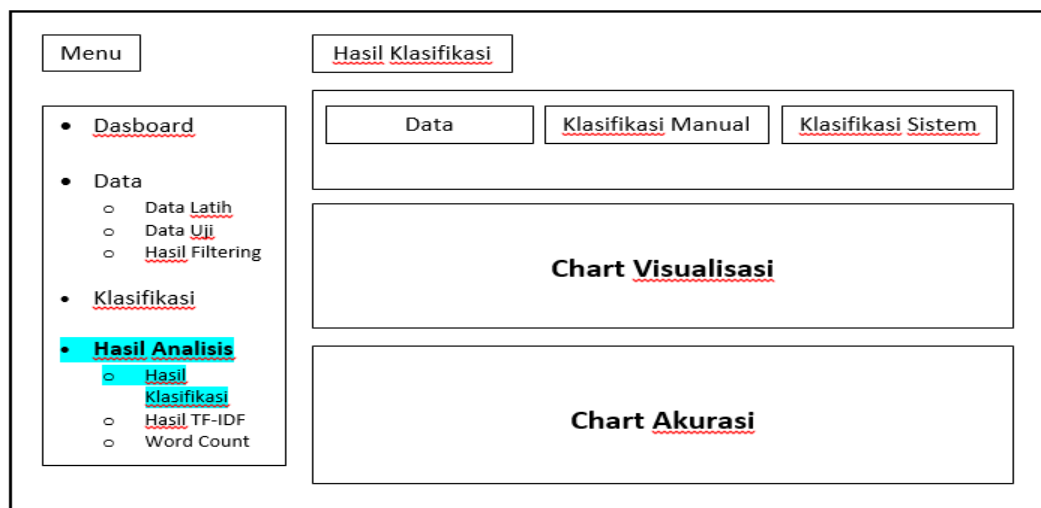
Gambar 3.11. Rancangan Halaman Hasil Analisis

Keterangan:

1. Menu hasil analisis berfungsi untuk menampilkan beberapa tampilan yaitu: hasil klasifikasi, hasil tf-idf, dan word count.
2. Hasil klasifikasi berfungsi untuk menampilkan tabel hasil klasifikasi.
3. Hasil tf-idf berfungsi untuk menampilkan tampilan hasil hasil tf idf.
4. Hasil word count berfungsi untuk menampilkan tampilan hasil dari kata yang paling sering muncul pada data latih dan data uji.

3.7.8 Rancangan Halaman Hasil Klasifikasi

Halaman ini berfungsi untuk menampilkan beberapa tampilan menu yaitu: tabel hasil klasifikasi, hasil visualisasi, dan hasil akurasi yang dapat dilihat pada Gambar 3.12.



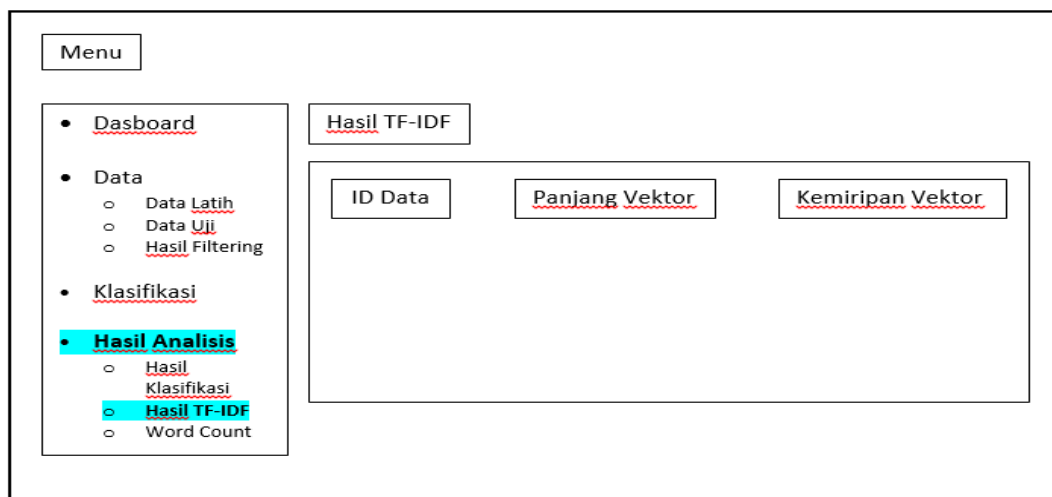
Gambar 3.12. Rancangan Halaman Hasil Klasifikasi

Keterangan:

1. Menu hasil klasifikasi berfungsi untuk menampilkan beberapa tampilan yaitu: hasil klasifikasi, hasil visualisasi, dan akurasi.
2. Hasil klasifikasi berfungsi untuk menampilkan tabel hasil klasifikasi.
3. Hasil visualisasi berfungsi untuk menampilkan tampilan hasil dari visualisasi.
4. Hasil akurasi berfungsi untuk menampilkan tampilan hasil dari akurasi.

3.7.9 Rancangan Halaman Hasil TF-IDF

Halaman ini berfungsi untuk menampilkan hasil dari TF-IDF yang dapat dilihat pada gambar 3.13.



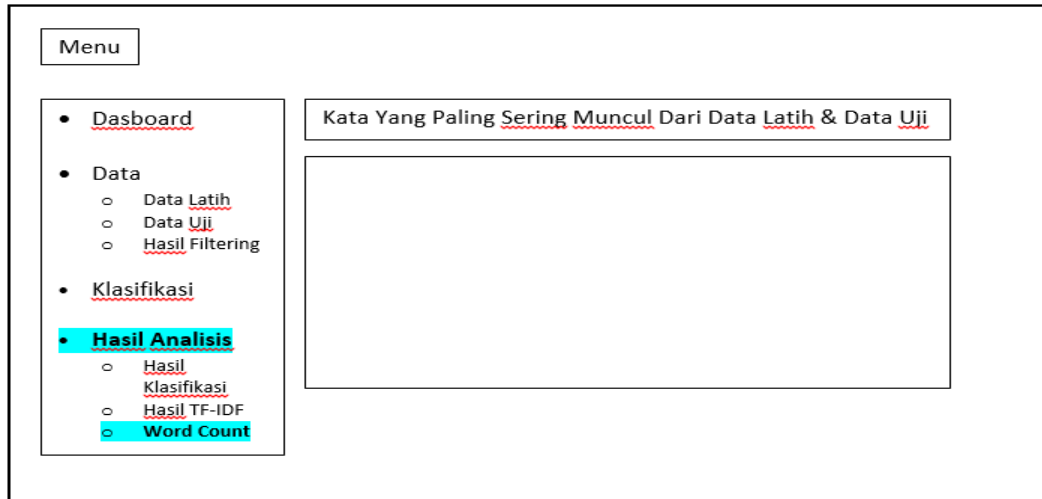
Gambar 3.13. Rancangan Halaman Hasil TF-IDF

Keterangan:

1. Menu hasil tf-idf berfungsi untuk menampilkan tabel hasil tf-idf.
2. Di dalam tabel sendiri ada beberapa keterangan yaitu: ID Data, Panjang Vektor, Kemiripan Vektor.

3.7.10 Rancangan Halaman Word Count

Pada halaman ini terdapat tampilan kata yang paling sering muncul dari data latih dan data uji yang dapat dilihat pada gambar 3.14.



Gambar 3.14. Rancangan Halaman Word Count

Keterangan:

1. Hasil word count berfungsi untuk menampilkan tampilan hasil dari kata yang paling sering muncul pada data latih dan data uji.