

BAB 4 PERANCANGAN ALGORITME

4.1 Deskripsi Masalah

Pangan merupakan kebutuhan pokok dalam kehidupan sehari-hari yang dibutuhkan manusia untuk mendapatkan energi. Sebagai kebutuhan dasar pangan mempunyai arti penting bagi manusia. Pangan berasal dari hayati dan air, baik diolah maupun tidak yang diperuntukkan manusia untuk dikonsumsi. Untuk memenuhi kebutuhan pangan, manusia mengolah bahan pangan mentah menjadi makanan siap saji dengan cara memasak. Memasak adalah kegiatan mengolah atau membuat berbagai macam makanan, lauk-pauk, dan sebagainya (KBBI, 2016). Dalam memasak suatu makanan tentunya seorang ibu rumah tangga membutuhkan resep masakan. Sebuah resep masakan dibutuhkan untuk mengolah bahan-bahan masakan agar menjadi suatu makanan yang dapat disajikan/dihadangkan.

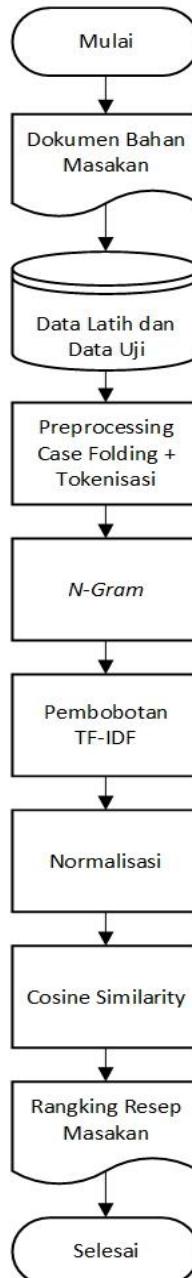
Namun tidak sedikit pula ibu rumah tangga yang masih kesulitan dalam mengolah bahan masakan. Hal tersebut menyebabkan banyaknya ibu rumah tangga yang lebih memilih membeli makanan yang sudah matang daripada memasak sendiri. Perilaku tersebut menyebabkan hilangnya sifat mandiri dari seorang ibu rumah tangga untuk membuat makanan sendiri. Solusi dari permasalahan tersebut adalah dengan menyediakan suatu wadah yang menampung macam-macam resep masakan. Informasi dari macam-macam resep masakan tersebut nantinya akan dibuat dalam suatu system yang dapat memberikan rekomendasi resep masakan berdasarkan bahan masakan yang ada. Solusi ini nantinya akan dijadikan sebuah penelitian yang menggunakan metode *N-gram* dan *cosine similarity* sebagai acuan untuk memperoleh rekomendasi resep masakan.

Metode *N-gram* adalah potongan *n*-karakter yang dihasilkan dari sebuah *string* kata. *N-gram* diperoleh dengan menambahkan *blank* di awal dan akhir suatu *string* kata. *N-gram* digunakan untuk melakukan pengurutan terhadap banyaknya kemunculan kata, dimana yang digunakan adalah hasil pemotongan *N-gram* yang telah terurut (Utomo,2015). Metode *N-gram* bisa diterapkan untuk mengolah ketersediaan jumlah bahan makanan yang tersedia.

Algoritma *cosine similarity* diterapkan pada sistem agar dapat mengetahui resep masakan yang paling mendekati keinginan pengguna. Sehingga pengguna bisa memilih resep yang diinginkan. Maka diperlukan suatu perhitungan kemiripan antar resep masakan menggunakan suatu fungsi *similarity*. Semakin besar hasil fungsi *similarity* maka kedua resep masakan yang dievaluasi semakin mirip. Hasil pengukuran ini memungkinkan perankingan resep masakan sesuai dengan kemiripannya terhadap *query* (Triana,2016). Kualitas hasil dari resep masakan yang didapatkan sangat tergantung pada *similarity* yang digunakan.

4.2 Deskripsi Umum Sistem

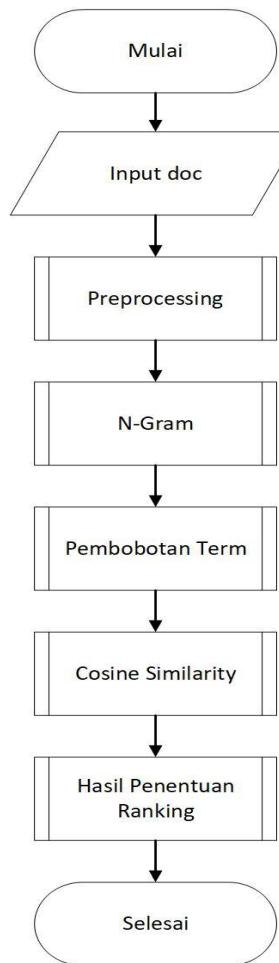
Rekomendasi resep masakan berdasarkan ketersediaan bahan masakan menggunakan metode *N-gram* dan *cosine similarity* merupakan suatu sistem yang berbasis *website*. Sistem akan memberikan rekomendasi resep masakan berdasarkan bahan masakan yang dimasukkan ke dalam kolom pencarian sistem. Proses untuk menentukan rekomendasi resep masakan dilakukan melalui beberapa proses yaitu *preprocessing*, pembobotan *term*, normalisasi serta hasil dari pembobotan *term*. Penentuan kelas atau kategori dari data uji dimasukkan akan mengikuti pada data latih yang sebelumnya telah diproses terlebih dahulu serta *N-gram*.



Gambar 0.1 Diagram Alir Metode

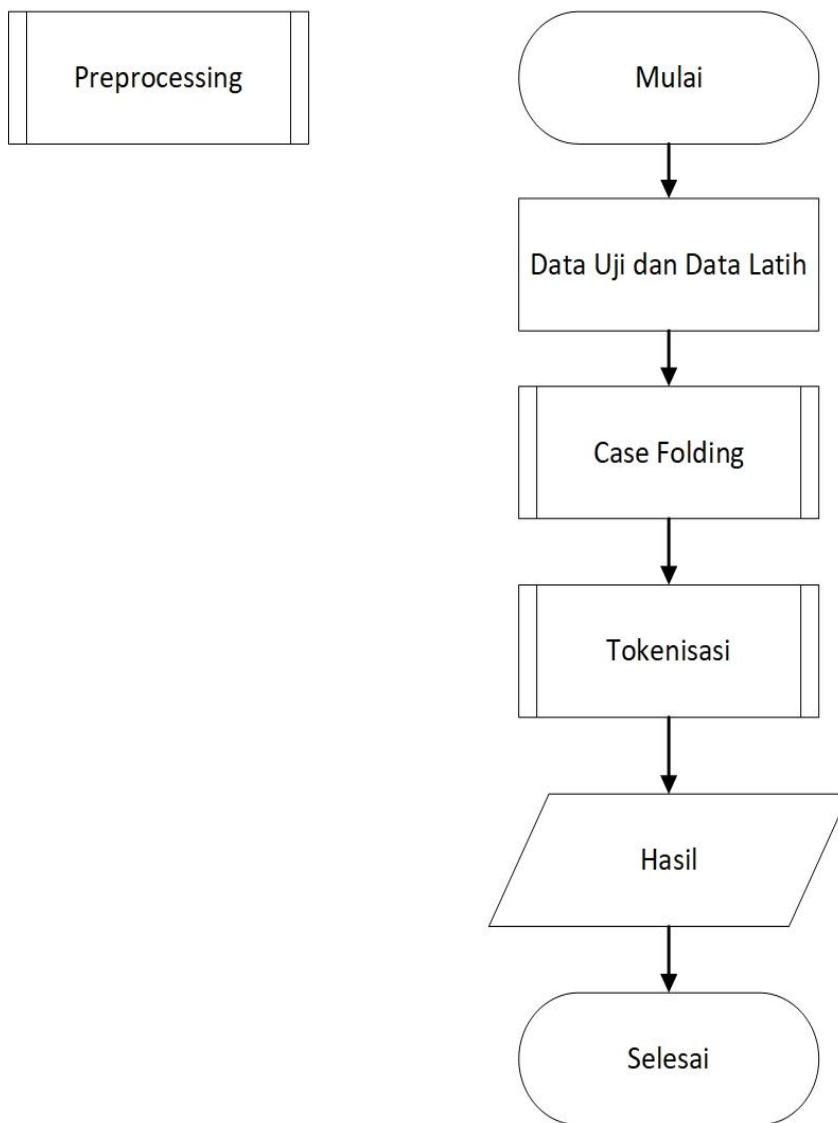
4.3 Diagram Alir Sistem

Pada Gambar 4.1 dijelaskan gambaran umum tentang sistem yang akan diberikan *input* dokumen berupa bahan masakan yang kemudian akan diproses. Berikutnya dilanjutkan dengan *preprocessing*, *N-gram*, Pembobotan *term*, perhitungan *cosine similarity*. Dimana *output* yang akan dihasilkan ialah berupa macam-macam rekomendasi resep masakan berdasarkan bahan masakan yang dimasukkan sebelumnya.



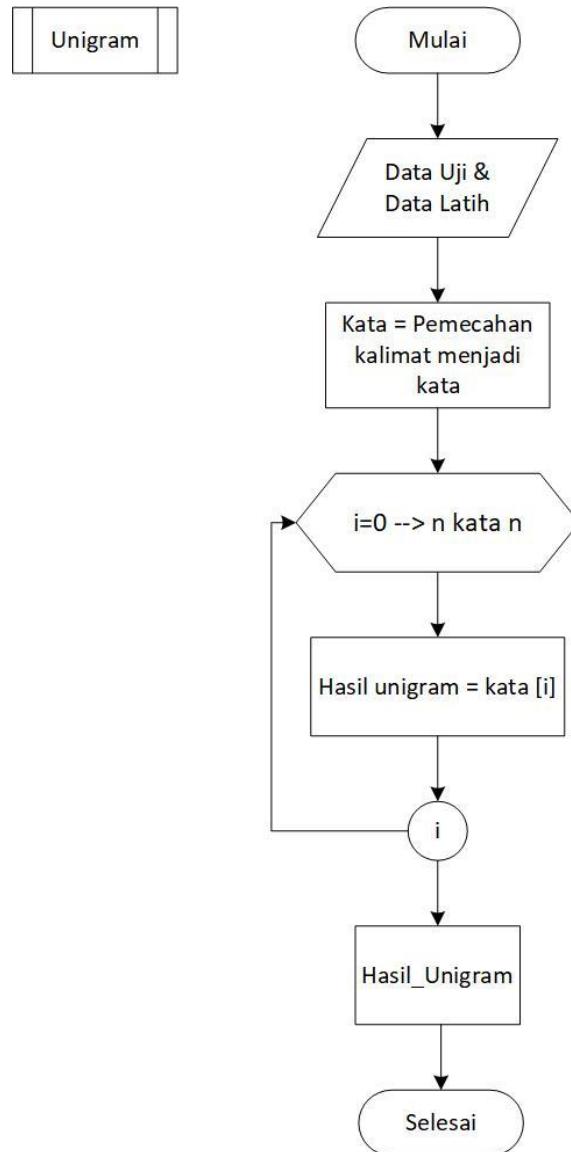
Gambar 0.2 Diagram Alir Sistem

Pada Gambar 4.3 ditunjukkan diagram alir dari proses *preprocessing*, melalui dari proses *case folding* dilakukan untuk mengubah semua huruf menjadi huruf kecil, kemudian tokenisasi untuk menghapus semua karakter kata dan memisah kata menjadi token-token. Hasil dari tahapan-tahapan yang ada pada *preprocessing* yang berupa daftar kata (*term*) yang kemudian akan diproses untuk melakukan pembobotan *term* hingga normalisasi.



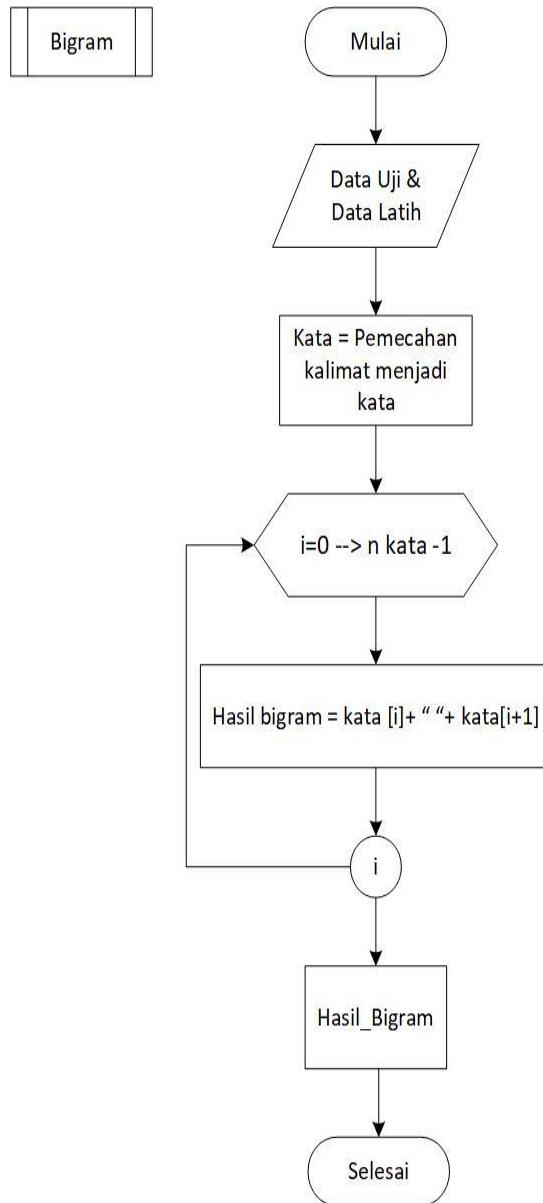
Gambar 0.3 Diagram Alir *Preprocessing*

Pada Gambar 4.4 ditunjukan diagram alir proses Tokenisasi *unigram*, maka selanjutnya akan dilakukan pemotongan kata sesuai dengan nilai *N-gram*. Algoritma ini akan membantu penilaian similaritas antar dokumen dengan nilai n dari karakternya.



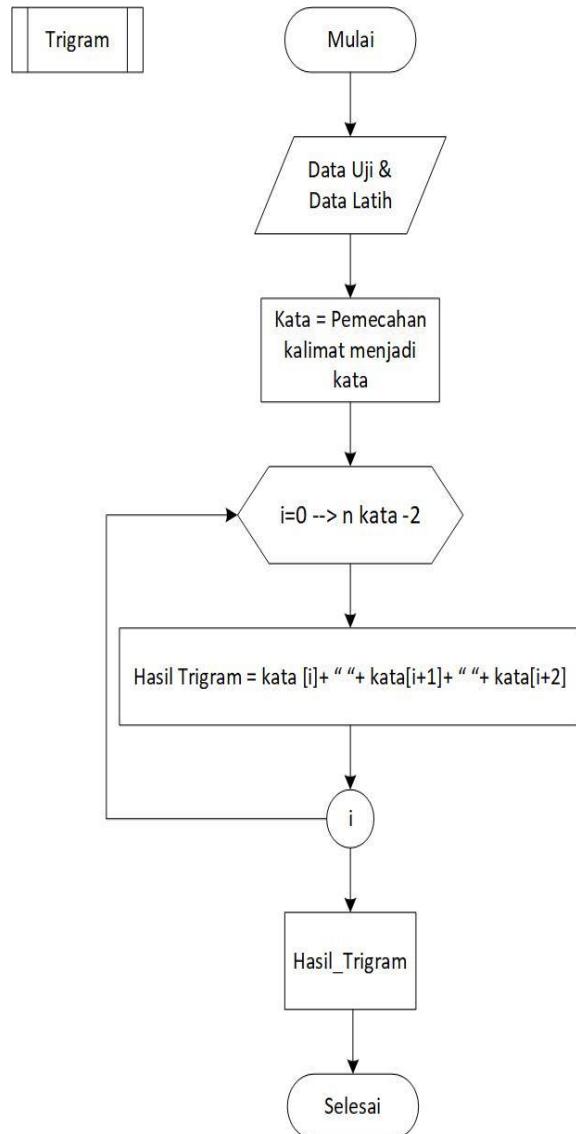
Gambar 0.4 Diagram Alir *Unigram*

Pada Gambar 4.5 ditunjukan diagram alir proses *Bigram*, maka selanjutnya akan dilakukan pemotongan kata sesuai dengan nilai *N-gram*. Algoritma ini akan membantu penilaian similaritas antar dokumen dengan nilai -1 dari karakternya.



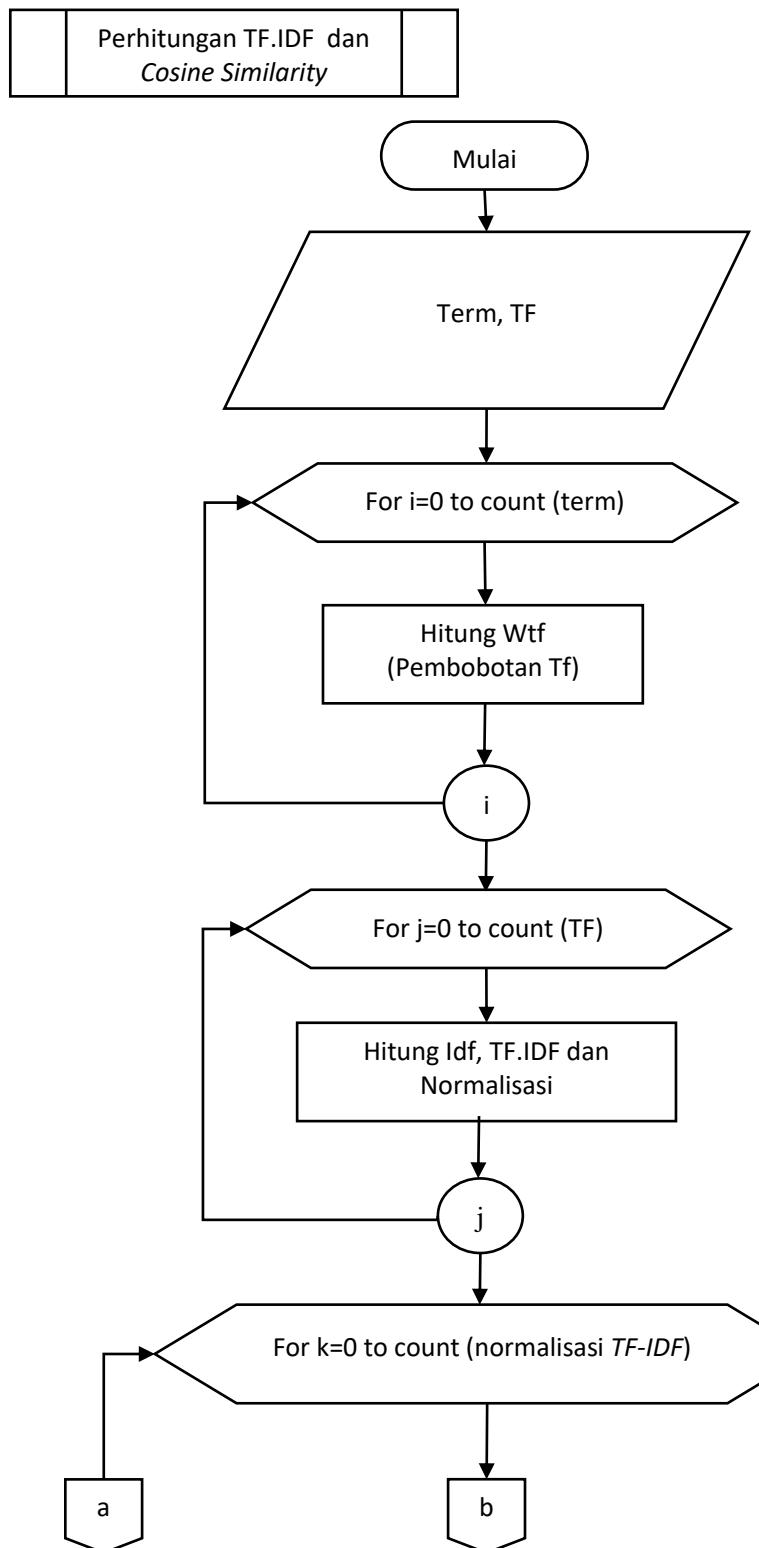
Gambar 0.5 Diagram Alir Bigram

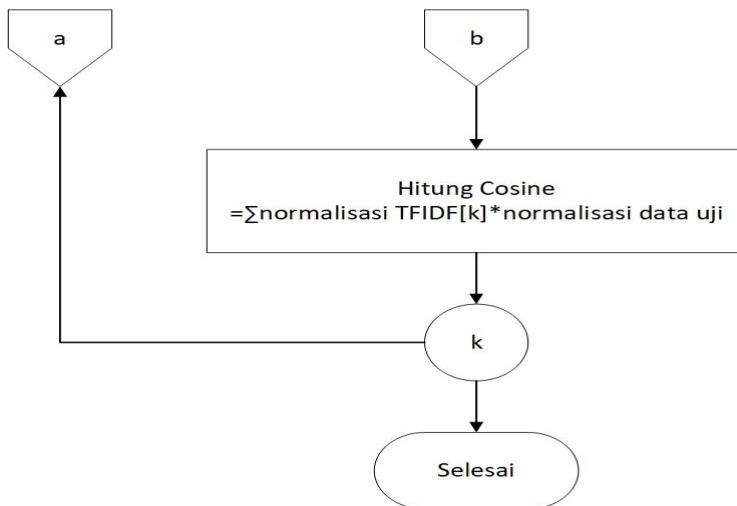
Pada Gambar 4.6 ditunjukkan diagram alir proses *Trigram*, maka selanjutnya akan dilakukan pemotongan kata sesuai dengan nilai *N-gram*. Algoritma ini akan membantu penilaian similaritas antar dokumen dengan nilai -2 dari karakternya.



Gambar 0.6 Diagram Alir Trigram

Pada Gambar 4.7 ditunjukkan diagram alir proses pembobotan *term* (kata), dimulai dengan memasukan seluruh *term* yang telah melalui *preprocessing* lalu dilakukan pembobotan dengan pembobotan *term*, yaitu dengan menghitung nilai *WTF* dari hasil nilai *TF*. Kemudian melakukan perhitungan nilai *TF-IDF*, nilai *IDF* didapatkan dengan melakukan perkalian pada dokumen latih dan hasil *WTF*. Lalu hasil dari perhitungan *TF-IDF* dinormalisasi. Setelah didapatkan hasil dari normalisasi kemudia dilanjutkan ke proses *cosine similarity* pada data uji terhadap data latih untuk mengetahui hasil kemiripan antara data uji dan data latih.





Gambar 0.7 Diagram Alir *TF-IDF* dan *Cosine similarity*

4.4 Manualisasi Perhitungan Data

Manualisasi perhitungan data ini akan menampilkan perhitungan dalam melakukan proses penentuan *ranking* dari dokumen data latih dan data uji.

4.4.1 *Preprocessing*

Pada tahap *preprocessing* dilakukan beberapa tahapan yaitu *case folding* dan tokenisasi. Hal tersebut dilakukan agar data terstruktur sehingga mempermudah dalam proses perhitungan. Tabel 4.1 ditunjukkan data latih yang digunakan, dan pada Tabel 4.2 ditunjukkan data uji yang digunakan.

Tabel 0.1 Data Latih

Dok	Resep Masakan	Kategori
1	Omelet Bayam Wortel Sosis: 4 butir telur , 3 ikat bayam , 1 buah wortel, 3 buah sosis , 1 buah bawang bombai, 1 sdm tepung terigu, 2 siung bawang putih, 2 siung bawang merah, Garam, Lada bubuk, Kaldu Jamur, Boncabe	Omelet
2	Sayur Bayam Sosis: 1 ikat bayam, 1 buah wortel , 1 buah sosis , 1 butir telur , 1 siung bawang putih, 5 siung bawang merah, 1 ruas kencur, Garam, Gula	Sayur bayam
3	Oseng Buncis Cambah Telur Sosis: 1 buah buncis, 250 gram cambah, 1 butir telur, 1 buah sosis, 3 siung bawang merah, 1 siung bawang putih,	Oseng Buncis

	1 buah tomat , 6 buah cabai kecil, 1 buah cabai besar , Garam, Gula, Kaldu Sapi	
4	Oseng Tahu Cambah: 1 buah tahu , 1 buah sosis , 250 gram Cambah , 6 buah cabai kecil, 5 siung bawang merah, 4 siung bawang putih,1 buah bawang bombai, 1 buah tomat, Perei , Saus Tiram ,Merica Bubuk	Oseng Tahu
5	Capcai Brokoli Jamur: 50 gram ayam, 1 ons brokoli, 8 buah jamur , 250 gram wortel , 1 siung bawang putih, 1 sdt saus tiram, 1 sdm tepung kanji, Garam ,Lada, Minyak Goreng	Capcai Brokoli

Tabel 0.2 Data Uji Query

Dok	Resep Masakan	Kategori
1	1 buah wortel,1 ikat bayam, 2 butir telur, 1 sdt kaldu jamur, 1 sdm minyak goreng, garam	

4.4.1.1 Case Folding

Tahapan *case folding* ialah proses yang dilakukan untuk mengubah semua huruf pada dokumen menjadi huruf kecil. Tabel 4.3 ditunjukkan hasil *case folding* yang dilakukan terhadap data latih yang akan digunakan dan pada Tabel 4.4 ditunjukkan hasil *case folding* yang dilakukan data uji yang akan digunakan.

Tabel 0.3 Hasil Case Folding data latih

Dok	Resep Masakan	Kategori
1	4 butir telur , 3 ikat bayam , 1 buah wortel, 3 buah sosis , 1 buah bawang bombai, 1 sdm tepung terigu, 2 siung bawang putih, 2 siung bawang merah, garam, lada bubuk, kaldu jamur, boncabe	Omelet
2	sayur bayam sosis: 1 ikat bayam, 1 buah wortel , 1 buah sosis , 1 butir telur , 1 siung bawang putih, 5 siung bawang merah, 1 ruas kencur, garam, gula	Sayur Bayam

3	oseng buncis cambah telur sosis: 1 buah buncis, 250 gram cambah, 1 butir telur, 1 buah sosis, 3 siung bawang merah, 1 siung bawah putih, 1 buah tomat , 6 buah cabai kecil, 1 buah cabai besar , garam, gula, kaldu sapi	Oseng Buncis
4	oseng tahu cambah: 1 buah tahu , 1 buah sosis , 250 gram Cambah , 6 buah cabai kecil, 5 siung bawang merah, 4 siung bawang putih,1 buah bawang bombai, 1 buah tomat, perei , saus tiram ,merica bubuk	Oseng tahu
5	capcaí brokoli jamur: 50 gram ayam, 1 ons brokoli, 8 buah jamur , 250 gram wortel , 1 siung bawang putih, 1 sdt saus tiram, 1 sdm tepung kanji, garam ,lada, minyak goreng	Capcaí Brokoli

Tabel 0.4 Hasil *Case Folding* data uji

Dok	Resep Masakan	Kategori
1	1 buah wortel,1 ikat bayam, 2 butir telur, 1 sdt kaldu jamur, 1 sdm minyak goreng, garam	

4.4.1.2 Tokenisasi

Tahapan tokenisasi dilakukan untuk memisahkan setiap kata yang dipisahkan oleh *whitespace* pada dokumen. Tabel 4.5 ditunjukkan hasil *tokenisasi* pada data latih menggunakan *unigram* telah melalui proses *case folding*.

Tabel 0.5 Hasil Tokenisasi data latih menggunakan *Unigram*

Dok	Resep Masakan	Kategori
1	4//butir//telur//3//ikat//bayam//1//buah//wortel//3//buah//osis //1//buah//bawang//bombai//1//sdm//tepung// terigu//2//suing//bawang//putih//2//suing// bawang//merah//garam//lada//bubuk//kaldu //jamur//boncabe	Omelet

2	1//ikat//bayam//1//buah //wortel//1//buah//osis//1//butir//telur//1//siung //bawang//putih//5//siung//bawang//merah//1 //ruas//kencur//garam,//gula	Sayur bayam
3	1//buah//buncis//250//gram//cambah//1//butir//telur//1 //buah//osis//3//suing//bawang//merah//1//siung //bawang//putih//1//buah//tomat//6//buah//cabai //kecil//1//buah//cabai//besar//garam//gula//kaldu //sapi	Oseng Buncis
4	1//buah//tahu//1//buah//osis//250//gram//cambah//6//buah //cabai//kecil//5//siung//bawang//merah//4//siung//bawang //putih//1//buah//bawang//bombai//1//buah//tomat//perei// saus//tiram//merica//bubuk	Oseng tahu
5	50//gram//ayam//1//ons//brokoli//8//buah// jamur//250//gram//wortel//1//siung//bawang//putih //1//sdt//saus//tiram//1//sdm//tepung//kanji//garam//lada// minyak goreng	Capcai Brokoli

Pada Tabel 4.6 ditunjukkan hasil *tokenisasi* pada data latih menggunakan *bigram* telah melalui proses *case folding*.

Tabel 0.6 Hasil Tokenisasi data latih menggunakan Bigram

Dok	Resep Masakan	Kategori
1	4 butir//butir telur// telur 3// 3 ikat//ikat bayam//bayam 1//1 buah//buah wortel//wortel 3//3 buah//buah sosis// sosis 1//1 buah//buah bawang// bawang bombai// bombai 1// 1 sdm//sdm tepung// tepung terigu// terigu 2// 2 siung// siung bawang// bawang putih// putih 2// bawang merah// merah garam// garam lada// lada bubuk// bubuk kaldu// kaldu jamur// jamur boncabe	Omelet
2	1 ikat// ikat bayam//bayam 1//1 buah//buah wortel//wortel 1//buah sosis//osis 1//1 butir//butir telur//telur 1//1 siung//siung bawang//bawang putih//putih 5//5 siung//bawang merah//merah 1//1 ruas//ruas kencur//kencur garam//garam gula	Sayur Bayam
3	cambah telur //telur sosis//osis 1//1 buah//buah buncis // buncis 250 //250 gram//gram cambah//cambah 1 //1 butir //butir telur//telur 1//buah sosis//osis 3 //3 siung //siung bawang//bawang merah//merah 1//1 siung	Oseng Buncis

	//bawang putih//putih 1//buah tomat //tomat 6//6 buah//buah cabai//cabai kecil//kecil 1//cabai besar //besar garam//garam gula//gula kaldu//kaldu sapi	
4	1 buah// tahu 1// buah sosis// sosis 250// 250 gram// gram cambah//cambah 6// 6 buah// buah cabai// cabai kecil// kecil 5// 5 siung// siung bawang// bawang merah// merah 4// 4 siung// bawang putih// putih 1// buah bawang// bawang bombai//bombai 1// buah tomat// tomat perei// perei saus// saus tiram// tiram merica// merica bubuk	Oseng Tahu
5	50 gram// gram ayam// ayam 1// 1 ons// ons brokoli// brokoli 8// 8 buah// buah jamur// jamur 250// 250 gram// gram wortel// wortel 1// 1 siung// siung bawang// bawang putih// putih 1// 1 sdt// sdt saus//saus tiram// tiram 1// 1 sdm// sdm tepung// tepung kanji// kanji garam// garam lada// lada minyak// minyak goreng.	Capcai Brokoli

Berikutnya pada Tabel 4.7 ditunjukkan hasil *tokenisasi* pada data latih menggunakan *trigram* telah melalui proses *case folding*.

Tabel 0.7 Hasil Tokenisasi data latih menggunakan Trigram

Dok	Resep Masakan	Kategori
1	4 butir telur// butir telur 3// telur 3 ikat// 3 ikat bayam// ikat bayam 1// bayam 1 buah// 1 buah wortel//buah wortel 3// wortel 3 buah// 3 buah sosis// buah sosis 1// sosis 1 buah// 1 buah bawang// buah bawang bombai// bawang bombai 1// bombai 1 sdm// 1 sdm tepung// sdm tepung terigu// tepung terigu 2// terigu 2 siung// 2 siung bawang// siung bawang merah// bawang merah garam// merah garam lada// garam lada bubuk// lada bubuk kaldu// bubuk kaldu jamur// kaldu jamur boncabe	Omelet
2	1 ikat bayam// ikat bayam 1// bayam 1 buah// 1 buah wortel//buah wortel 1// wortel 1 buah// 1 buah sosis// buah sosis 1// sosis 1 butir//1 butir telur// butir telur 1// telur 1 siung// 1 siung bawang// siung bawang putih// bawang putih 5// putih 5 siung// 5 siung bawang// siung bawang merah// bawang merah 1// merah 1 ruas// 1 ruas	Sayur Bayam

	kencur// ruas kencur garam// kencur garam gula	
3	1 buah tahu// buah tahu 1// tahu 1 buah// 1 buah sosis// buah sosis 250// sosis 250 gram// 250 gram cambah// gram cambah 6// cambah 6 buah// 6 buah cabai// buah cabai kecil// cabai kecil 5// kecil 5 siung// 5 siung bawang// siung bawang merah// bawang merah 4// merah 4 siung// 4 siung bawang// siung bawang putih// bawang putih 1// putih 1 buah// 1 buah tomat// buah tomat perei// tomat perei saus// perei saus tiram// saus tiram merica// tiram merica bubuk	Oseng Buncis
4	1 buah tahu// buah tahu 1// tahu 1 buah// 1 buah sosis// buah sosis 250// sosis 250 gram// 250 gram cambah// gram cambah 6// cambah 6 buah// 6 buah cabai// buah cabai kecil// cabai kecil 5// kecil 5 siung// 5 siung bawang// siung bawang merah// bawang merah 4// merah 4 siung// 4 siung bawang// siung bawang putih// bawang putih 1// putih 1 buah// 1 buah tomat// buah tomat perei// tomat perei saus// perei saus tiram// saus tiram merica// tiram merica bubuk	Oseng tahu
5	50 gram ayam// gram ayam 1// ayam 1 ons// 1 ons brokoli// ons brokoli 8// brokoli 8 buah// 8 buah jamur// buah jamur 250// jamur 250 gram// 250 gram wortel// gram wortel 1// wortel 1 siung// 1 siung bawang// siung bawang putih// bawang putih 1// putih 1 sdt// 1 sdt saus// sdt saus tiram// saus tiram 1// tiram 1 sdm// 1 sdm tepung// sdm tepung kanji// tepung kanji garam// kanji garam lada// garam lada minyak// lada minyak goreng	Capcai Brokoli

Pada Tabel 4.8 ditunjukkan hasil tokenisasi pada data uji menggunakan *unigram* telah melalui proses *case folding*.

Tabel 0.8 Hasil Tokenisasi data uji menggunakan *Unigram*

Dok	Resep Masakan	Kategori
1	1//buah//wortel//1//ikat//bayam//2//butir//telur//1//sdt//kaldu//jamur//1//sdm//minyak//goreng//garam	

Berikutnya pada Tabel 4.9 merupakan hasil tokenisasi pada data uji yang telah melalui proses *case folding* menggunakan *Bigram* seperti berikut ini.

Tabel 0.9 Hasil Tokenisasi data uji menggunakan *Bigram*

Dok	Resep Masakan	Kategori
1	1 buah// buah wortel// wortel 1// 1 ikat// ikat bayam// bayam 2// 2 butir// butir telur// telur 1// 1 sdt// sdt kaldu// kaldu jamur// jamur 1// 1 sdm// sdm minyak// minyak goreng// goreng garam	

Pada Tabel 4.10 ditunjukkan hasil *tokenisasi* pada data uji menggunakan *trigram* telah melalui proses *case folding*.

Tabel 0.10 Hasil Tokenisasi data uji menggunakan *Trigram*

Dok	Resep Masakan	Kategori
1	1 buah wortel// buah wortel 1// wortel 1 ikat// 1 ikat bayam// ikat bayam 1// bayam 2 butir// 2 butir telur// butir telur 1// telur 1 sdt// 1 sdt kaldu//sdt kaldu jamur// kaldu jamur 1//jamur 1 sdm// 1 sdm minyak//sdm minyak goreng// minyak goreng garam	

4.2.1 Pembobotan

Pada tahap ini, perhitungan dengan menggunakan *TF-IDF* dimana *IDF* dapat menghitung suatu kemunculan kata pada dokumen. Persamaan yang digunakan ditunjukan pada persamaan 2.1 hingga 2.3.

4.2.1.1 Menghitung *TF* dan *IDF*

Perhitungan *TF* dan *IDF* dimulai dengan melakukan perhitungan pada banyaknya kemunculan *term* (kata) pada dokumen dengan menggunakan persamaan 2.1.

Setelah itu melakukan perhitungan untuk mengetahuin banyaknya dokumen *d* yang memiliki *t* (*df*), kemudian melakukan perhitungan *IDF* dengan menggunakan persamaan 2.2. Pada Tabel 4.11 menunjukan hasil dari perhitungan *TF* dan *IDF*.

Contoh Perhitungan *Unigram* pada *term* 4:

$$w_{tf_{t,d}} = 1 + \log_{10} 1 = 1$$

Setelah mendapatkan nilai $W_{TF}(t,d)$, dilanjutkan dengan menghitung berapa banyak dokumen d yang terdapat term t (df). Data uji tidak ikut dalam menghitung df . Kemudian dilanjutkan dengan mencari nilai IDF .

$$idf_t = \frac{2}{1} = 0.47712$$

Pada Tabel 4.11 ditunjukkan hasil perhitungan TF , W_{TF} , df , dan IDF .

Tabel 0.11 Hasil perhitungan TF dan IDF Unigram

No	Term	Tf						df	idf
		d1	d2	d3	d4	d5	uji		
1	4	1	0	0	1	0	0	2	0.47712 1
2	Butir	1	1	1	0	0	1	4	0.17609 1
3	Telur	1	1	1	0	0	1	4	0.17609 1
4	3	1.3010	0	1	0	0	0	2	0.47712 1
5	Ikat	1	1	0	0	0	1	3	0.30103
6	bayam	1	1	0	0	0	1	3	0.30103
7	1	1.4771 2	1.7781 5	1.6989 7	1.602 1	1.602 1	1	6	0
8	buah	1.4771 2	1.3010 3	1.6989 7	1.699		1	6	0
9	wortel	1	1	0	0	1	1	4	0.17609 1
10	sosis	1	1	1	1	0	0	4	0.17609 1
11	bombai	1	0	0	1	0	0	2	0.47712 1
12	sdm	1	0	0	0	1	1	3	0.30103
13	tepung	1	0	0	0	1	0	2	0.47712 1
14	terigu	1	0	0	0	0	0	1	0.77815 1
15	2	1.3010 3	0	0	0	0	1	2	0.47712 1
16	siung	1.3010 3	1.3010 3	1.3010 3	1.301	1	0	5	0.07918 1
17	bawang	1.4771 2	1.3010 3	1.3010 3	1.301	1	0	5	0.07918 1
18	putih	1	1	1	1	1	0	5	0.07918 1
19	merah	1	1	1	1	0	0	4	0.17609 1
20	garam	1	1	1	0	1	1	5	0.07918 1

21	lada	1	0	0	0	1	0	2	0.47712 1
22	bubuk	1	0	0	1	0	0	2	0.47712 1
23	kaldu	1	0	1	0	0	1	3	0.30103
24	jamur	1	0	0	0	1	1	3	0.30103
25	Boncab e	1	0	0	0	0	0	1	0.77815 1
26	5	0	1	0	1	0	0	2	0.47712 1
27	ruas	0	1	0	0	0	0	1	0.77815 1
28	kencur	0	1	0	0	0	0	1	0.77815 1
29	gula	0	1	1	0	0	0	2	0.47712 1
30	buncis	0	0	1	0	0	0	1	0.77815 1

Contoh Perhitungan *Bigram* pada term omelet bayam:

$$w_{tf_{t,d}} = 1 + \log_{10} 1 = 1$$

Setelah mendapatkan nilai W_{TF} (t,d), dilanjutkan dengan menghitung berapa banyak dokumen d yang terdapat term t (df). Data uji tidak ikut dalam menghitung df . Kemudian dilanjutkan dengan mencari nilai *IDF*. Contoh perhitungan untuk term 4 butir:

$$idf_t = \frac{6}{1} = 0.77815$$

Pada Tabel 4.12 memperlihatkan hasil perhitungan *TF*, W_{TF} , *df*, dan *IDF*.

Tabel 0.12 Hasil perhitungan *TF* dan *IDF Bigram*

No	Term	Tf						df	Idf
		d1	d2	d3	d4	d5	uji		
1	4 butir	1	0	0	0	0	0	1	0.77815
2	butir telur	1	1	1	0	0	1	4	0.17609
3	telur 3	1	0	0	0	0	0	1	0.77815
4	3 ikat	1	0	0	0	0	0	1	0.77815
5	ikat bayam	1	1	0	0	0	1	3	0.30103
6	bayam 1	1	1	0	0	0	0	2	0.47712
7	1 buah	2	0	1.6021	1.6021	0	1	5	0.07918
8	buah wortel	1	1	0	0	0	1	3	0.30103
9	wortel 3	1	0	0	0	0	0	1	0.77815

10	3 buah	1	0	0	0	0	0	1	0.77815
11	buah sisis	1	1	1	1	0	0	4	0.17609
12	sisis 1	1	1	0	0	0	0	2	0.47712
13	buah bawang	1	0	0	1	0	0	2	0.47712
14	bawang bombai	1	0	0	1	0	0	2	0.47712
15	bombai 1	1	0	0	1	0	0	2	0.47712
16	1 sdm	1	0	0	0	1	1	3	0.30103
17	sdm tepung	1	0	0	0	1	0	2	0.47712
18	tepung terigu	1	0	0	0	0	0	1	0.77815
19	terigu 2	1	0	0	0	0	0	1	0.77815
20	2 siung	1.301	0	0	0	0	0	1	0.77815
21	siung bawang	1.301	1.301	1.301	1.301	1	0	5	0.07918
22	bawang putih	1	1	1	1	1	0	5	0.07918
23	putih 2	1	0	0	0	0	0	1	0.77815
24	bawang merah	1	1	1	1	0	0	4	0.17609
25	merah garam	1	0	0	0	0	0	1	0.77815
26	garam lada	1	0	0	0	0	0	1	0.77815
27	lada bubuk	1	0	0	0	0	0	1	0.77815
28	bubuk kaldu	1	0	0	0	0	0	1	0.77815
29	kaldu jamur	1	0	0	0	0	1	2	0.47712
30	jamur boncabe	1	0	0	0	0	0	1	0.77815

Contoh Perhitungan *Trigram* pada term 4 butir telur:

$$w_{tf_{t,d}} = 1 + \log_{10} 1 = 1$$

Setelah mendapatkan nilai $W_{TF}(t,d)$, dilanjutkan dengan menghitung berapa banyak dokumen d yang terdapat term t (df). Data uji tidak ikut dalam menghitung df . Kemudian dilanjutkan dengan mencari nilai *IDF*. Contoh perhitungan untuk term 4 butir telur:

$$idf_t = \frac{6}{1} = 0.7782$$

Tabel 4.13 memperlihatkan hasil perhitungan *TF*, W_{TF} , df , dan *IDF*.

Tabel 0.13 Hasil perhitungan *TF* dan *IDF Trigram*

No	Term	Tf						df	Idf
		d1	d2	d3	d4	d5	uji		
1	4 butir telur	1	0	0	0	0	0	1	0.778151
2	butir telur 3	1	0	0	0	0	0	1	0.778151
3	telur 3 ikat	1	0	0	0	0	0	1	0.778151
4	3 ikat bayam	1	0	0	0	0	0	1	0.778151
5	ikat bayam 1	1	1	0	0	0	0	2	0.477121
6	bayam 1 buah	1	1	0	0	0	0	2	0.477121
7	1 buah wortel	1	1	0	0	0	1	3	0.30103
8	buah wortel 3	1	0	0	0	0	0	1	0.778151
9	wortel 3 buah	1	0	0	0	0	0	1	0.778151
10	3 buah sosis	1	0	0	0	0	0	1	0.778151
11	buah sosis 1	1	0	0	0	0	0	2	0.477121
12	sosis 1 buah	1	0	0	0	0	0	2	0.477121
13	1 buah bawang	1	0	0	0	0	0	2	0.477121
14	buah bawang bombai	1	0	0	0	0	0	2	0.477121
15	bawang bombai 1	1	0	0	0	0	0	2	0.477121
16	bombai 1 sdm	1	0	0	0	0	0	1	0.778151
17	1 sdm tepung	1	0	0	0	0	0	2	0.477121
18	sdm tepung terigu	1	0	0	0	0	0	1	0.778151
19	tepung terigu 2	1	0	0	0	0	0	1	0.778151
20	terigu 2 siung	1	0	0	0	0	0	1	0.778151
21	2 siung bawang	1	0	0	0	0	0	1	0.778151

22	siung bawang putih	1	0	0	0	0	0	5	0.079181
23	bawang putih 2	1	0	0	0	0	0	1	0.778151
24	putih 2 siung	1.301	0	0	0	0	0	1	0.778151
25	siung bawang merah	1	1	1	1	0	0	4	0.176091
26	bawang merah garam	1	0	0	0	0	0	1	0.778151
27	merah garam lada	1	0	0	0	0	0	1	0.778151
28	garam lada bubuk	1	0	0	0	0	0	1	0.778151
29	lada bubuk kaldu	1	0	0	0	0	0	1	0.778151
30	bubuk kaldu jamur	1	0	0	0	0	0	1	0.778151

4.2.1.2 Menghitung *TF-IDF Weighting*

Proses perhitungan *TF-IDF Weighting* dilakukan dengan mengkalikan nilai *TF* dan *IDF*, proses perhitungan *TF-IDF Weighting* dapat menggunakan persamaan 2.3.

Contoh perhitungan *Unigram* pada term 4 dalam d1:

Selanjutnya menghitung nilai *TF-IDF* dengan mengalikan hasil dari *TF* dan *IDF*.

$$w_{t,d} = 1 * 47712 = 0.47712$$

Pada Tabel 4.14 memperlihatkan hasil perhitungan *TF-IDF Weight*.

Tabel 0.14 Hasil *TF-IDF Weighting Unigram*

No	Term	Wtd						Uji
		d1	d2	d3	d4	d5		
1	4	0.47712	0	0	0.4771	0	0	0
2	Butir	0.17609	0.17609	0.17609	0	0	0	0.176
3	Telur	0.17609	0.17609	0.17609	0	0	0	0.176
4	3	0.62075	0	0.47712	0	0	0	0
5	Ikat	0.30103	0.30103	0	0	0	0	0.301
6	Bayam	0.30103	0.30103	0	0	0	0	0.301
7	1	0	0	0	0	0	0	0
8	Buah	0	0	0	0	0	0	0

9	Wortel	0.17609	0.17609	0	0	0.1761	0.176
10	Sosis	0.17609	0.17609	0.17609	0.1761	0	0
11	Bombai	0.47712	0	0	0.4771	0	0
12	Sdm	0.30103	0	0	0	0.301	0.301
13	Tepung	0.47712	0	0	0	0.4771	0
14	Terigu	0.77815	0	0	0	0	0
15	2	0.62075	0	0	0	0	0.477
16	Siung	0.10302	0.10302	0.10302	0.103	0.0792	0
17	bawang	0.11696	0.10302	0.10302	0.103	0.0792	0
18	Putih	0.07918	0.07918	0.07918	0.0792	0.0792	0
19	Merah	0.17609	0.17609	0.17609	0.1761	0	0
20	Garam	0.07918	0.07918	0.07918	0	0.0792	0.079
21	Iada	0.47712	0	0	0	0.4771	0
22	Bubuk	0.47712	0	0	0.4771	0	0
23	Kaldu	0.30103	0	0.30103	0	0	0.301
24	Jamur	0.30103	0	0	0	0.301	0.301
25	boncabe	0.77815	0	0	0	0	0
26	5	0	0.47712	0	0.4771	0	0
27	Ruas	0	0.77815	0	0	0	0
28	Kencur	0	0.77815	0	0	0	0
29	Gula	0	0.47712	0.47712	0	0	0
30	Buncis	0	0	0.77815	0	0	0

Contoh perhitungan *Bigram* pada term 4 butir dalam d1:

Selanjutnya menghitung nilai *TF-IDF* dengan mengalikan hasil dari *TF* dan *IDF*.

$$w_{t,d} = 1 * 0.778 = 0.7782$$

Pada Tabel 4.15 memperlihatkan hasil perhitungan *TF-IDF Weight*.

Tabel 0.15 Hasil *TF-IDF Weighting Bigram*

No	Term	Wtd					
		d1	d2	d3	d4	d5	Uji
1	4 butir	0.7782	0	0	0	0	0
2	butir telur	0.1761	0.176	0.1761	0	0	0.176
3	telur 3	0.7782	0	0	0	0	0
4	3 ikat	0.7782	0	0	0	0	0
5	ikat bayam	0.301	0.301	0	0	0	0.301
6	bayam 1	0.4771	0.477	0	0	0	0
7	1 buah	0.1028	0.117	0.1269	0.1269	0	0.079
8	buah wortel	0.301	0.301	0	0	0	0.301
9	wortel 3	0.7782	0	0	0	0	0
10	3 buah	0.7782	0	0	0	0	0
11	buah sosis	0.1761	0.176	0.1761	0.1761	0	0
12	sosis 1	0.4771	0.477	0	0	0	0

13	buah bawang	0.4771	0	0	0.4771	0	0
14	bawang bombai	0.4771	0	0	0.4771	0	0
15	bombai 1	0.4771	0	0	0.4771	0	0
16	1 sdm	0.301	0	0	0	0.301	0.301
17	sdm tepung	0.4771	0	0	0	0.4771	0
18	tepung terigu	0.7782	0	0	0	0	0
19	terigu 2	0.7782	0	0	0	0	0
20	2 siung	1.0124	0	0	0	0	0
21	siung bawang	0.103	0.103	0.103	0.103	0.0792	0
22	bawang putih	0.0792	0.079	0.0792	0.0792	0.0792	0
23	putih 2	0.7782	0	0	0	0	0
24	bawang merah	0.1761	0.176	0.1761	0.1761	0	0
25	merah garam	0.7782	0	0	0	0	0
26	garam lada	0.7782	0	0	0	0	0
27	lada bubuk	0.7782	0	0	0	0	0
28	bubuk kaldu	0.7782	0	0	0	0	0
29	kaldu jamur	0.4771	0	0	0	0	0.477
30	jamur boncabe	0.7782	0	0	0	0	0

Contoh perhitungan *Trigram* pada term 4 butir telur dalam d1:

Selanjutnya menghitung nilai *TF-IDF* dengan mengalikan hasil dari *TF* dan *IDF*.

$$w_{t,d} = 1 * 0.778 = 0.778$$

Pada Tabel 4.16 memperlihatkan hasil perhitungan *TF-IDF Weight*.

Tabel 0.16 Hasil *TF-IDF Weighting Trigram*

No	Term	Wtd					
		d1	d2	d3	d4	d5	Uji
1	4 butir telur	0.778	0	0	0	0	0
2	butir telur 3	0.778	0	0	0	0	0
3	telur 3 ikat	0.778	0	0	0	0	0
4	3 ikat bayam	0.778	0	0	0	0	0
5	ikat bayam 1	0.477	0.477	0	0	0	0
6	bayam 1 buah	0.477	0.477	0	0	0	0
7	1 buah wortel	0.301	0.301	0	0	0	0
8	buah wortel 3	0.778	0.477	0	0	0	0
9	wortel 3 buah	0.778	0.477	0	0	0	0
10	3 buah sosis	0.778	0.301	0	0	0	0.3
11	buah sosis 1	0.778	0	0	0	0	0
12	sosis 1 buah	0.778	0	0	0	0	0
13	1 buah bawang	0.778	0	0	0	0	0

14	buah bawang bombai	0.477	0	0	0	0	0
15	bawang bombai 1	0.477	0	0	0	0	0
16	bombai 1 sdm	0.477	0	0	0	0	0
17	1 sdm tepung	0.477	0	0	0	0	0
18	sdm tepung terigu	0.477	0	0	0	0	0
19	tepung terigu 2	0.778	0	0	0	0	0
20	terigu 2 siung	0.477	0	0	0	0	0
21	2 siung bawang	0.778	0	0	0	0	0
22	siung bawang putih	0.778	0	0	0	0	0
23	bawang putih 2	0.778	0	0	0	0	0
24	putih 2 siung	0.778	0	0	0	0	0
25	siung bawang merah	0.079	0	0	0	0	0
26	bawang merah garam	0.778	0	0	0	0	0
27	merah garam lada	1.012	0	0	0	0	0
28	garam lada bubuk	0.176	0.176	0.176	0.176	0	0
29	lada bubuk kaldu	0.778	0	0	0	0	0
30	bubuk kaldu jamur	0.778	0	0	0	0	0

4.2.1.3 Normalisasi

Hasil perhitungan *TF-IDF weighting* $w_{t,d}$ diatas akan dinormalisasikan dengan menggunakan persamaan 2.4 berikut contoh perhitungan *Unigram* pada term 4 dalam d1:

$$w_{t,d} = \frac{0.47712}{\sqrt{2.4268}} = 0,1966$$

Setelah proses normalisasi dilakukan untuk mempermudah proses perhitungan *cosine similarity* (derajat kemiripan). Pada Tabel 4.17 memperlihatkan hasil perhitungan normalisasi *TF-IDF weighting unigram*.

Tabel 0.17 Hasil Normalisasi *TF-IDF Weighting Unigram*

No	Term	Normalisasi wtd					
		d1	d2	d3	d4	d5	Uji
1	4	0.1966	0	0	0.2118	0	0
2	Butir	0.0726	0.10671	0.0841	0	0	0.0875

3	Telur	0.0726	0.10671	0.0841	0	0	0.0875
4	3	0.2558	0	0.2278	0	0	0
5	Ikat	0.124	0.18241	0	0	0	0.1496
6	Bayam	0.124	0.18241	0	0	0	0.1496
7	1	0	0	0	0	0	0
8	Buah	0	0	0	0	0	0
9	Wortel	0.0726	0.10671	0	0	0.066	0.0875
10	Sosis	0.0726	0.10671	0.0841	0.07817	0	0
11	bombai	0.1966	0	0	0.2118	0	0
12	Sdm	0.124	0	0	0	0.1128	0.1496
13	tepung	0.1966	0	0	0	0.1787	0
14	Terigu	0.3206	0	0	0	0	0
15	2	0.2558	0	0	0	0	0.2372
16	Siung	0.0424	0.06243	0.0492	0.04573	0.0297	0
17	bawang	0.0482	0.06243	0.0492	0.04573	0.0297	0
18	Putih	0.0326	0.04798	0.0378	0.03515	0.0297	0
19	Merah	0.0726	0.10671	0.0841	0.07817	0	0
20	Garam	0.0326	0.04798	0.0378	0	0.0297	0.0394
21	lada	0.1966	0	0	0	0.1787	0
22	Bubuk	0.1966	0	0	0.2118	0	0
23	Kaldu	0.124	0	0.1437	0	0	0.1496
24	Jamur	0.124	0	0	0	0.1128	0.1496
25	boncabe	0.3206	0	0	0	0	0
26	5	0	0.28912	0	0.2118	0	0
27	Ruas	0	0.47154	0	0	0	0
28	kencur	0	0.47154	0	0	0	0
29	Gula	0	0.28912	0.2278	0	0	0
30	Buncis	0	0	0.3716	0	0	0

Hasil perhitungan *TF-IDF weighting* $w_{t,d}$ diatas akan dinormalisasikan dengan menggunakan persamaan 2.4 berikut contoh perhitungan *Bigram* pada term 4 butir dalam d1:

$$w_{t,d} = \frac{0.7782}{\sqrt{3.3316}} = 0.2336$$

Setelah proses normalisasi dilakukan untuk mempermudah proses perhitungan *cosine similarity* (derajat kemiripan). Pada Tabel 4.18 memperlihatkan hasil perhitungan normalisasi *TF-IDF weighting bigram*.

Tabel 0.18 Hasil Normalisasi *TF-IDF Weighting Bigram*

No	Term	Normalisasi wtd					
		d1	d2	d3	d4	d5	Uji
1	4 butir	0.2336	0	0	0	0	0
	butir						
2	telur	0.0529	0.0708	0.0545	0	0	0.0733
3	telur 3	0.2336	0	0	0	0	0
4	3 ikat	0.2336	0	0	0	0	0
	ikat						
5	bayam	0.0904	0.1211	0	0	0	0.1253
6	bayam 1	0.1432	0.1919	0	0	0	0
7	1 buah	0.0309	0.0469	0.0393	0.040825	0	0.033
	buah						
8	wortel	0.0904	0.1211	0	0	0	0.1253
9	wortel 3	0.2336	0	0	0	0	0
10	3 buah	0.2336	0	0	0	0	0
	buah						
11	sisis	0.0529	0.0708	0.0545	0.056672	0	0
12	sisis 1	0.1432	0.1919	0	0	0	0
	buah						
13	bawang	0.1432	0	0	0.153553	0	0
	bawang						
14	bombai	0.1432	0	0	0.153553	0	0
15	bombai 1	0.1432	0	0	0.153553	0	0
16	1 sdm	0.0904	0	0	0	0.089944	0.1253
	sdm						
17	tepung	0.1432	0	0	0	0.142557	0
	tepung						
18	terigu	0.2336	0	0	0	0	0
19	terigu 2	0.2336	0	0	0	0	0
20	2 siung	0.3039	0	0	0	0	0
	siung						
21	bawang	0.0309	0.0414	0.0319	0.033154	0.023658	0
	bawang						
22	putih	0.0238	0.0318	0.0245	0.025483	0.023658	0
23	putih 2	0.2336	0	0	0	0	0
	bawang						
24	merah	0.0529	0.0708	0.0545	0.056672	0	0
	merah						
25	garam	0.2336	0	0	0	0	0
26	garam	0.2336	0	0	0	0	0
	lada						
27	bubuk	0.2336	0	0	0	0	0

28	bubuk kaldu	0.2336	0	0	0	0	0
29	kaldu jamur	0.1432	0	0	0	0	0.1986
30	jamur boncabe	0.2336	0	0	0	0	0

Hasil perhitungan *TF-IDF weighting* $w_{t,d}$ diatas akan dinormalisasikan dengan menggunakan persamaan 2.4 berikut contoh perhitungan *trigram* pada term 4 butir telur dalam d1:

$$w_{t,d} = \frac{0,778}{\sqrt{4,1277}} = 0,1885$$

Setelah proses normalisasi dilakukan untuk mempermudah proses perhitungan *cosine similarity* (derajat kemiripan). Pada Tabel 4.19 memperlihatkan hasil perhitungan normalisasi *TF-IDF weighting trigram*.

Tabel 0.19 Hasil Normalisasi *TF-IDF Weighting Trigram*

No	Term	Normalisasi wtd					
		d1	d2	d3	d4	d5	uji
1	4 butir telur	0.1885	0	0	0	0	0
2	butir telur 3	0.1885	0	0	0	0	0
3	telur 3 ikat	0.1885	0	0	0	0	0
4	3 ikat bayam	0.1885	0	0	0	0	0
5	ikat bayam 1	0.1156	0.1471	0	0	0	0
6	bayam 1 buah	0.1156	0.1471	0	0	0	0
7	1 buah wortel	0.0729	0.0928	0	0	0	0
8	buah wortel 3	0.1885	0.1471	0	0	0	0
9	wortel 3 buah	0.1885	0.1471	0	0	0	0
10	3 buah sosis	0.1885	0.0928	0	0	0	0.1007
11	buah sosis 1	0.1885	0	0	0	0	0
12	sosis 1 buah	0.1885	0	0	0	0	0
13	1 buah bawang	0.1885	0	0	0	0	0
14	buah bawang bombai	0.1156	0	0	0	0	0
15	bawang bombai 1	0.1156	0	0	0	0	0
16	bombai 1 sdm	0.1156	0	0	0	0	0
17	1 sdm tepung	0.1156	0	0	0	0	0
18	sdm tepung terigu	0.1156	0	0	0	0	0
19	tepung terigu 2	0.1885	0	0	0	0	0

20	terigu 2 siung	0.1156	0	0	0	0	0
21	2 siung bawang	0.1885	0	0	0	0	0
22	siung bawang putih	0.1885	0	0	0	0	0
23	bawang putih 2	0.1885	0	0	0	0	0
24	putih 2 siung	0.1885	0	0	0	0	0
25	siung bawang merah	0.0192	0	0	0	0	0
26	bawang merah garam	0.1885	0	0	0	0	0
27	merah garam lada	0.2453	0	0	0	0	0
28	garam lada bubuk	0.0427	0.0543	0.04267	0.0447	0	0
29	lada bubuk kaldu	0.1885	0	0	0	0	0
30	bubuk kaldu jamur	0.1885	0	0	0	0	0

4.2.2 Cosine similarity

Setelah perhitungan normalisasi selesai, kemudian dilanjutkan dengan menghitung nilai *Cosine similarity*. Perhitungan *cosine similiraty* dihitung dengan mengkalikan data latih dengan data uji yang ingin diketahui kategorinya. Persamaan yang akan digunakan untuk menghitung *cosine similarity* dapat dilihat pada persamaan 2.6. Berikut contoh perhitungan *Cosine similarity Unigram* untuk d1 :

$$CosCim(d_j, q) = (0.1966 * 0) = 0$$

Pada Tabel 4.20 ditunjukkan hasil perhitungan *cosine similarity* data latih terhadap data uji.

Tabel 0.20 Hasil cosine similarity Unigram

CosSim				
d1 x uji	d2 x uji	d3 x uji	d4 x uji	d5 x uji
0	0	0	0	0
0.0063512	0.00934	0.00736	0	0
0.0063512	0.00934	0.00736	0	0
0	0	0	0	0
0.0185609	0.027295	0	0	0
0.0185609	0.027295	0	0	0
0	0	0	0	0
0	0	0	0	0

0.0063512	0.00934	0	0	0.005774
0	0	0	0	0
0	0	0	0	0
0.0185609	0	0	0	0.016874
0	0	0	0	0
0	0	0	0	0
0.0606632	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0.0012842	0.001888	0.001488	0	0.001167
0	0	0	0	0
0	0	0	0	0
0.0185609	0	0.021508	0	0
0.0185609	0	0	0	0.016874
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0.4403738	0.266522	0.090479	0	0.271321

Setelah didapatkan hasil *cosine similarity* maka akan ditentukan nilai yang terbesar. Dan akan menampilkan nilai tertinggi dengan nilai *similarity* sebesar 0,44037 pada dokumen D1. Ditunjukkan pada Tabel 4.21

Tabel 0.21 Urutan kemiripan data uji

Dok	Nilai <i>Cosine similarity</i>	Kategori
D1	0.44037	Omelet

Berikut contoh perhitungan *cosine similarity bigram* untuk d1. Setelah perhitungan normalisasi selesai, kemudian dilanjutkan dengan menghitung nilai *cosine similarity*.

$$CosCim(d_1, q) = (0.2336 * 0) = 0$$

Pada Tabel 4.22 ditunjukkan hasil perhitungan *cosine similarity* data latih terhadap data uji.

Tabel 0.22 Hasil cosine similarity Bigram

0.028445	0	0	0	0
0	0	0	0	0
0.02945	0.285077	0.028641	0	0

Setelah didapatkan hasil *cosine similarity* maka akan ditentukan nilai yang terbesar. Dan akan menampilkan nilai tertinggi dengan nilai *similarity* sebesar 0.2850 pada dokumen D2. Ditunjukkan pada Tabel 4.23.

Tabel 0.23 Urutan kemiripan data uji Bigram

Dok	Nilai <i>Cosine similarity</i>	Kategori
D2	0,2850	Sayur Bayam

Berikut contoh perhitungan *cosine similarity bigram* untuk d1. Setelah perhitungan normalisasi selesai, kemudian dilanjutkan dengan menghitung nilai *cosine similarity*.

$$CosCim(d_j, q) = 0.1885 * 0 = 0$$

Pada Tabel 4.24 ditunjukkan hasil perhitungan *cosine similarity* data latih terhadap data uji.

Tabel 0.24 Hasil *cosine similarity Trigram*

CosSim				
d1 x uji	d2 x uji	d3 x uji	d4 x uji	d5 x uji
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0.009349	0	0	0
0	0	0	0	0
0	0	0	0	0
0.018988		0	0	0

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0.02944	0.285077	0.02864	0	0

Setelah didapatkan hasil *cosine similarity* maka akan ditentukan nilai yang terbesar. Dan akan menampilkan nilai tertinggi dengan nilai *similarity* sebesar 0.2850 pada dokumen D2. Ditunjukkan pada Tabel 4.25.

Tabel 0.25 Urutan kemiripan data uji *Trigram*

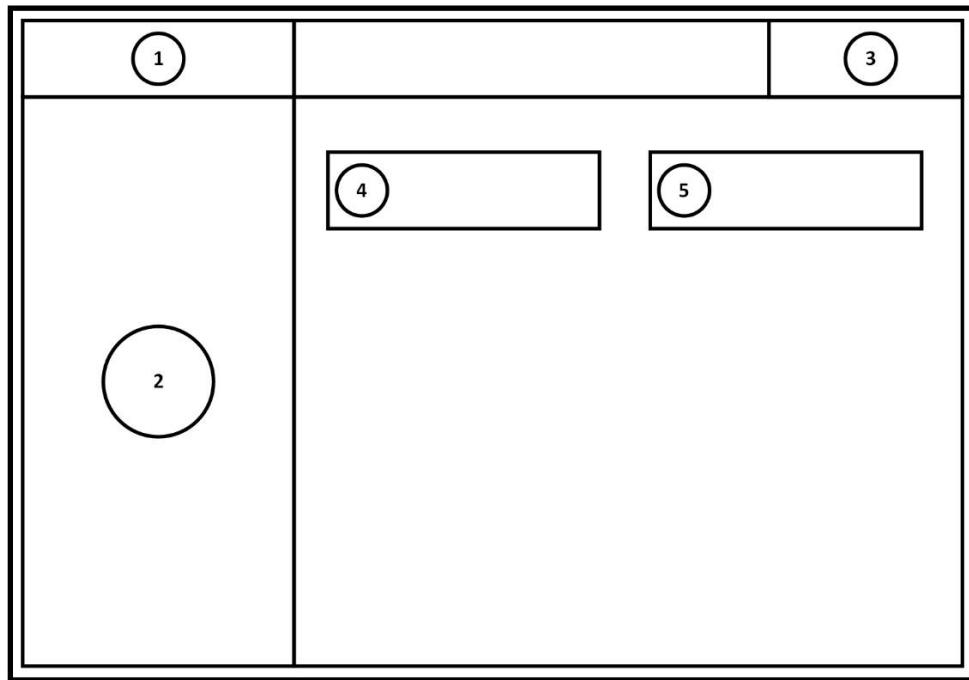
Dok	Nilai <i>Cosine similarity</i>	Kategori
D2	02850	Sayur Bayam

4.3 Perancangan Antarmuka (*Pengguna Interface*)

Perancangan antarmuka atau UI (*Pengguna Interface*) dibutuhkan sebagai penghubung antara pengguna dengan sistem. Adapun perancangan antarmuka yang akan dibangun dalam sistem adalah sebagai berikut:

4.3.1 Halaman Awal

Halaman awal ialah halaman utama yang pertama kali tampil. Dalam halaman ini terdapat beberapa komponen yang terdiri dari judul sistem, nama pembuat sistem, menu sistem, kolom data uji dan pilihan perhitungan. Tampilan dari antarmuka halaman awal ditunjukkan pada Gambar 4.8 di bawah ini.



Gambar 0.8 Halaman Awal

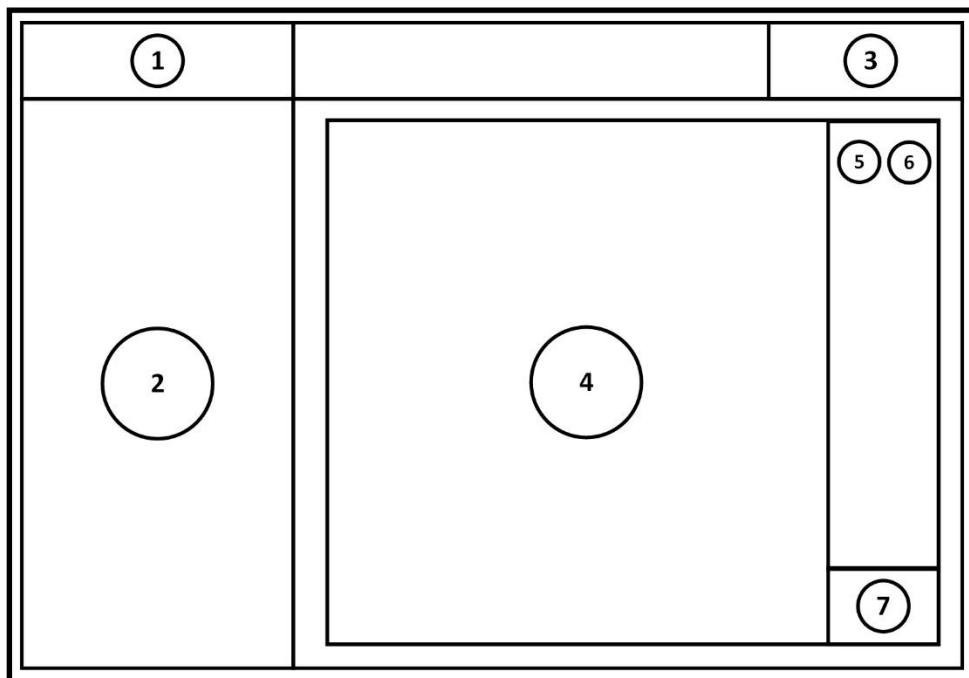
Keterangan:

1. Judul sistem
2. Menu sistem
3. Nama pembuat sistem
4. Kolom data uji
5. Kolom pilihan perhitungan terhadap data uji

4.3.2 Antarmuka Data Latih

Halaman antarmuka data latih berfungsi untuk memasukkan data latih baru. Pada halaman ini juga disediakan fitur untuk mengubah data latih dan

menghapus data latih sesuai keinginan. Antarmuka halaman data latih ditunjukkan pada Gambar 4.9 di bawah ini.



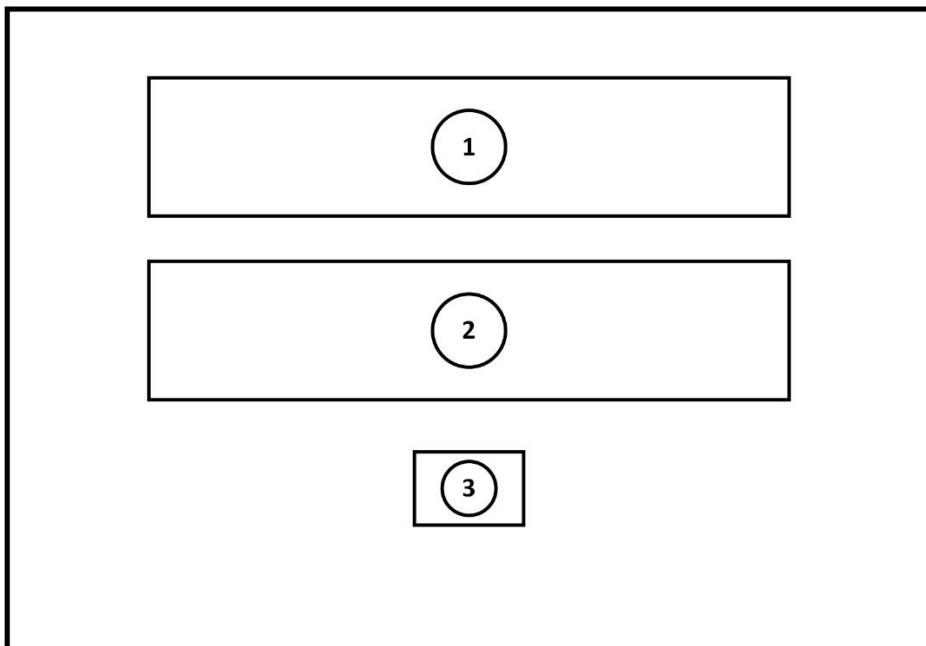
Gambar 0.9 Halaman Data Latih

Keterangan:

1. Judul sistem
2. Menu sistem
3. Nama pembuat sistem
4. Tabel data latih
5. Tombol edit data latih
6. Tombol hapus data latih
7. Tombol tambah data latih

4.3.3 Halaman Tambah Data Latih

Halaman tambah data latih berfungsi untuk menambahkan data latih baru untuk dimasukkan ke dalam database sistem. Pada halaman ini terdapat dua kolom yang terdiri dari kolom judul data latih dan kolom isi data latih. Pada halaman ini juga terdapat tombol simpan untuk menyimpan data latih. Antarmuka halaman tambah data latih ditunjukkan pada Gambar 4.10 di bawah ini.



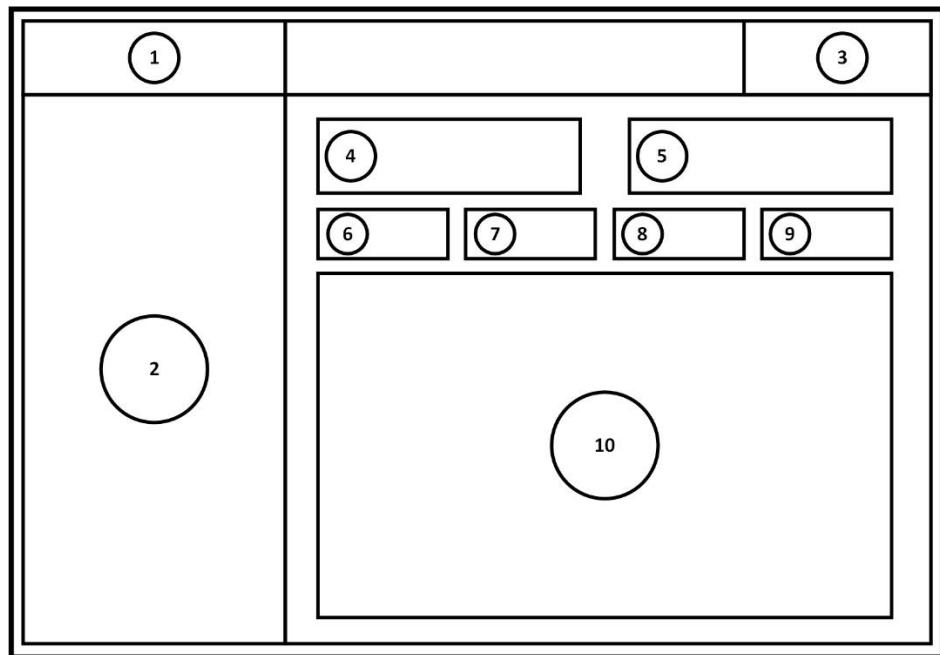
Gambar 0.10 Halaman Tambah Data Latih

Keterangan:

1. Kolom judul data latih
2. Kolom isi data latih
3. Tombol simpan

4.3.4 Halaman Hasil Pengujian

Halaman pengujian terdiri dari kolom data uji, kolom pilihan perhitungan, serta tombol perhitungan. Kolom data uji digunakan untuk memasukkan data uji yang akan dihitung. Kolom pilihan perhitungan digunakan untuk memilih perhitungan yang akan diterapkan pada data uji. Pilihan yang terdapat pada kolom perhitungan yaitu *unigram*, *bigram*, dan *trigram*. Pada halaman ini juga disediakan tombol untuk melakukan perhitungan yang terdiri dari tombol *TF-IDF*, normalisasi, *cosine similarity*, dan *ranking*. Antarmuka halaman pengujian ditunjukkan pada Gambar 4.11 di bawah ini.



Gambar 0.11 Hasil Pengujian

Keterangan:

1. Judul sistem
2. Judul sistem
3. Nama pembuat sistem
4. Kolom data uji
5. Kolom pilihan perhitungan
6. Tombol *TF-IDF*
7. Tombol Normalisasi
8. Tombol *Cosine similarity*
9. Tombol *Ranking*
10. Tabel hasil pengujian

4.4 Perancangan Pengujian dan Analisis

Perancangan pengujian dan analisis dilakukan untuk mengetahui adanya kesalahan atau tidak ketika melakukan implementasi dengan menggunakan metode *N-Gram* dan *Cosine similarity*. Pengujian akan dilakukan dengan *recall* dan *precision* dari masing masing proses dari metode *N-gram*. Perancangan dari table skenario ditunjukkan pada Tabel 4.34 berikut ini.

Tabel 0.26 Struktur tabel pengujian *N-gram*

Query	Threshold							
	30%	R/TR	50%	R/TR	70%	R/TR	90%	T/R

Tabel 0.27 Struktur tabel pengujian data uji

Query	Unigram			Bigram			Trigram		
	Threshold	Recall	Precision	Threshold	Recall	Precision	Threshold	Recall	Precision

Pengujian ini dilakukan untuk menemukan nilai *threshold* untuk hasil dari pengukuran kemiripan antar *query* untuk menentukan rekomendasi resep masakan.

Tabel 0.28 Perancangan Tabel Pengujian

Query	N-Gram	Threshold			
		30%	50%	70%	90%