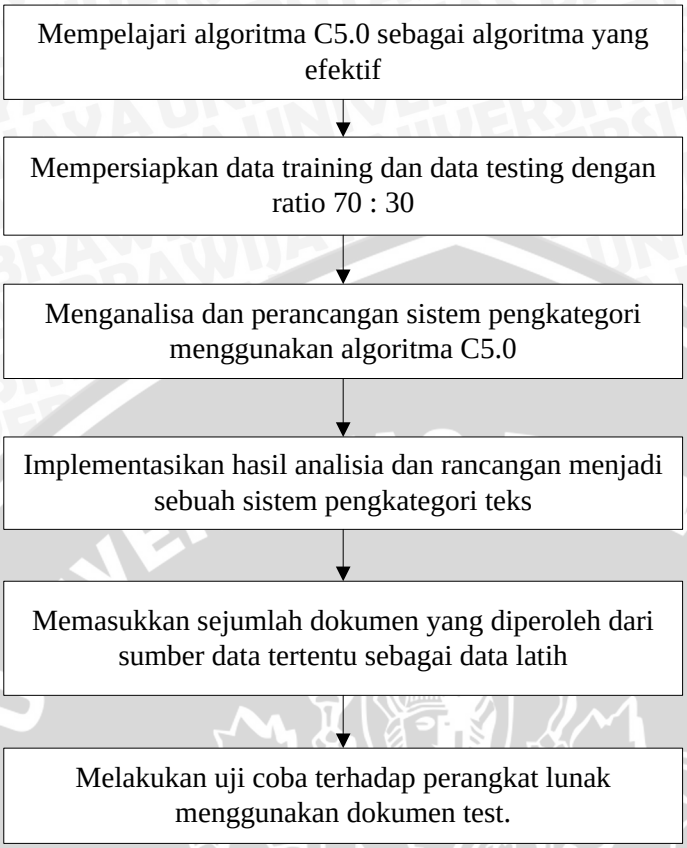
BAB III
METODOLOGI DAN PERANCANGAN

Dalam bab ini akan memberikan penjelasan mengenai metode dan langkah-langkah yang dilakukan dalam penelitian yaitu untuk membuat sistem klasifikasi emosi pada teks bahasa Indonesia menggunakan algoritma C5.0. Berikut langkah yang akan dilakukan :

1. Mempelajari literatur mengenai algoritma C5.0 sebagai suatu algoritma yang efektif untuk menyelesaikan masalah pengkategorian emosi pada teks.
2. Mempersiapkan data yang merupakan data training dan data testing, dengan perbandingan antara data training dan data testing dengan ratio 70 : 30.
3. Menganalisis dan melakukan perancangan sistem pengkategorian teks menggunakan algoritma C5.0.
4. Mengimplementasikan hasil analisis dan rancangan yang dilakukan sebelumnya menjadi sebuah sistem pengkategori teks otomatis.
5. Melakukan proses pelatihan terhadap sistem dengan memasukkan sejumlah dokumen yang diperoleh dari sumber data tertentu sebagai data latih.
6. Melakukan uji coba terhadap perangkat lunak menggunakan dokumen uji. Hasil yang diperoleh adalah dokumen yang telah terkategori.

Mengevaluasi tingkat keberhasilan sistem yaitu dengan membandingkan hasil pengkategorian yang dilakukan oleh sistem dengan hasil pengkategorian dari sumber.



Gambar 3.1: Alur penelitian

3.1 Analisis Data

Penelitian ini mengambil objek penelitian dari www.news.viva.co.id yang merupakan salah satu situs berita berbahasa Indonesia yang banyak dicari para pencari berita.

Di dalam situs vivanews ini terdapat beberapa kategori berita, seperti politik,showbiz,bisnis,nasional dan lain-lain. Dan dari situs ini dikumpulkan berita yang memuat emosi dan sesuai dengan tema penelitian. Dalam penelitian ini hasil dari berita yang dikumpulkan dikategorikan lagi menjadi 5 kategori, kategori yang digunakan adalah kategori emosi yang telah ditentukan yaitu marah, senang, sedih, dan takut. Hal ini bertujuan untuk memperoleh data latih (*training set*) yang tepat dan untuk mempermudah pengujian kebenaran dan keakuratan pada data *testing*.

3.1.1 Pengambilan Data

Pengambilan data training dilakukan dengan cara mengambil sebuah halaman berita yang ada pada web dari situs www.news.viva.co.id yang semula bentuknya sebuah halaman *web* akan diambil hanya teks kedalam format *txt* dikarenakan yang dipergunakan hanya teks dari halaman web tersebut. Pencarian pada tiap kategori yaitu marah, sedih, senang dan takut dilakukan dengan menginputkan kata kedalam kolom *search* yang tersedia pada halaman *web* sesuai dengan kategori berita yang akan diambil sebagai data training.

Kemudian setelah mendapatkan *result* dari *search* akan muncul beberapa *link* berita, pilih salah satu *link* tersebut maka akan muncul sebuah halaman berita. Dari *body* berita pada halaman *web* tersebut yang akan diambil hanya teks didalam isi berita tersebut, *copy* atau ditempel ke dalam *notepad* dan disimpan sesuai dengan nama kategori berita yang diambil dalam format "*.txt*". Karena sesuai pada batasan masalah yang telah dijelaskan pada subbab 1.4 pada point ke-2 yaitu sebuah dokumen teks dengan format *txt*. Dari pengambilan data set yang telah dijelaskan, maka data training yang akan digunakan seluruh kategorinya adalah sebuah dokumen teks dalam bentuk file "*notepad*" dengan format "*txt"*.

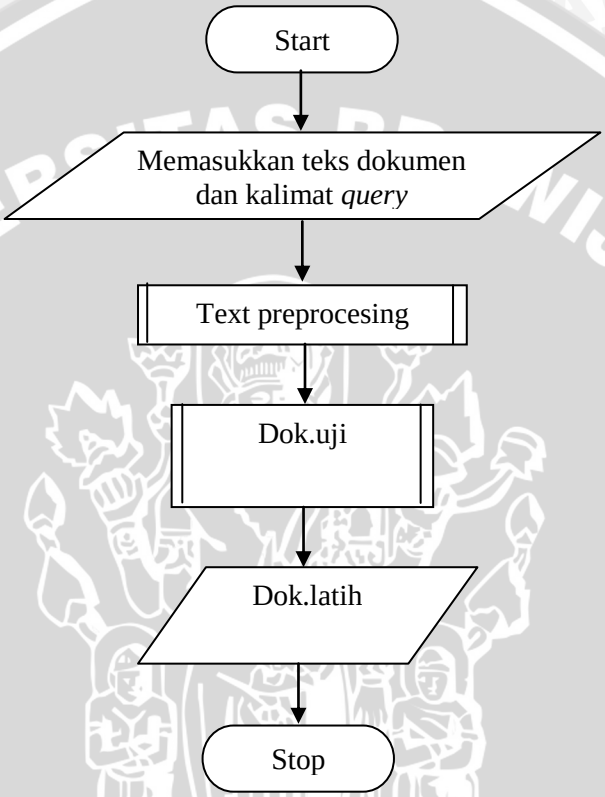
3.2 Deskripsi Umum Sistem

Pada penelitian ini dibagi dalam 3 tahap, yaitu *pre-processing*, pembelajaran, dan pengklasifikasian. Sistem akan dapat bekerja jika pengguna memasukkan data latih yang berupa dokumen teks. fungsi dari data latih itu sendiri adalah untuk membentuk classifier yang diperlukan dalam pengkategorian dokumen baru. Tahapan-tahapan yang dilakukan dalam pengisian data latih adalah sebagai berikut :

1. Pengguna memasukkan dokumen berita yang akan dijadikan dokumen latih.
2. Tahap *preprocessing* diawali dengan *tokenization*, meliputi *case folding*, kemudian menghilangkan semua angka dan tanda baca, dan *parsing* tiap-tiap kata yang menyusun dokumen. *Filtering*, penghilangan *stopwords* yang terdapat pada dokumen, tapi ada beberapa kata yang tidak dihilangkan karena termasuk dalam kategori emosi. *Stemming* kata, pemotongan imbuhan dan mengembalikan kata ke bentuk dasarnya.

3. Penentuan atribut yang digunakan sebagai inputan dalam klasifikasi menggunakan algoritma *C5.0.*
4. Penghitungan *information gain* dari masing-masing atribut yang telah ditentukan.
5. Pembuatan pohon keputusan dan disimpan untuk keperluan pengujian.

Berikut adalah gambar aliran data pada system ditunjukkan pada gambar 3.2



Gambar 3.2: Flowchart Deskripsi umum sistem

3.2.1 Batasan sistem

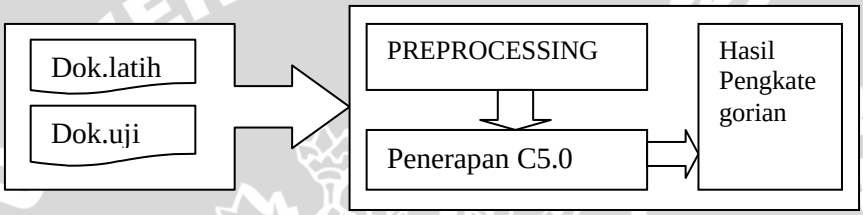
Sistem yang dibuat memiliki keterbatasan sebagai berikut :

1. Sistem hanya menangani dokumen berita berbahasa Indonesia saja.
2. Sumber berita hanya diperoleh dari satu situs berita berbahasa Indonesia yaitu www.news.viva.co.id
3. Sistem hanya dapat menangani dokumen dalam format *.txt*
4. Proses *stemming* hanya melakukan pemotongan terhadap awalan dan atau akhiran (*confix stripping*) dan tidak memperhatikan sisipan. *Stemming* juga tidak memperhatikan kata-kata yang memiliki imbuhan spesifik atau

imbuhan asing seperti *-wan, -wati, -isasi*. Kata-kata tersebut dianggap sebagai kata tunggal.

3.3 Rancangan Proses

Bagian ini menjelaskan secara rinci mengenai proses yang dilakukan sistem pengkategori secara berurutan dan sistematis. Pada awalnya sistem melakukan tahapan *preprocessing* pada tiap-tiap dokumen. Selanjutnya pembentukan *classifier* dilakukan menggunakan algoritma *C5.0*. Perancangan pada sistem bisa digambarkan dalam skema perancangan system berikut ini



Gambar 3.3 Skema Rancangan Proses

3.3.1 Perancangan *Preprocessing*

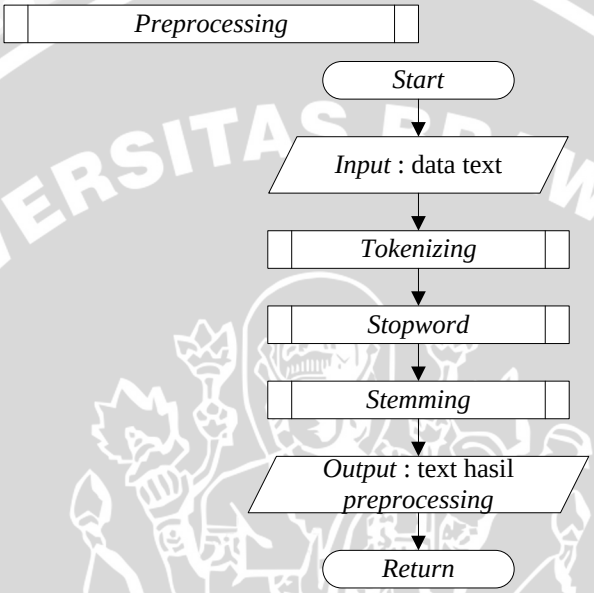
Pada sub bab 3.3.1 menjelaskan perancangan tahap *preprocessing*, pada tahapan ini dilakukan proses pemecahan dokumen string panjang menjadi string token berdasar beberapa variabel pemisahnya (*tokenizing*), penghapusan kata sering muncul namun kurang bermakna (*stopword*) dan pencarian kata dasar (*stemming*). Proses *preprocessing* digambarkan pada *flowchart* 3.4.

Tahap *prerocessing* terbagi dalam tiga sub proses, berikut alur pemprosesan teks pada gambar 3.4 *flowchart* perancangan *preprocessing*.

1. Data teks "*txt"* hasil dari pengambilan *body* berita akan menjadi *input* dalam proses *preprocessing*.
2. Data teks pertama kali akan dilakukan *tokenizing*, pemecahan partikel, penghapusan tanda baca dan merubah alphabet menjadi huruf kecil. Hasil dari proses *tokenizing* berupa string dalam bentuk token.
3. Hasil *tokenizing*, selanjutnya melalui proses *stopword* penghapusan kata-kata yang kurang memiliki makna. Hasil dari proses ini berupa kumpulan string yang tidak ada pada daftar *stopword*.

4. Hasil dari *stopword*akan dilakukan pencarian kata dasar dari sejumlah kata berimbuhan. Hasil dari proses *stemming* berupa kumpulan string dalam bentuk kata dasar.
5. Hasil dari *preprocessing* disimpan pada string *list* .

Detail proses dari setiap proses pada tahap *preprocessing*akan di jelaskan pada sub bab berikutnya.



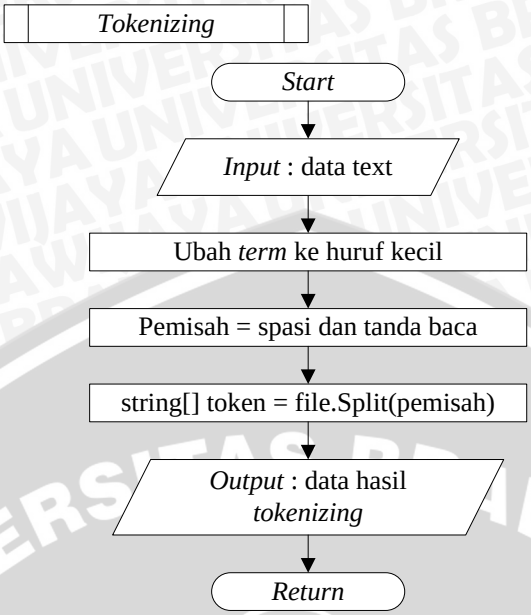
Gambar 3.4 *Flowchart* perancangan *preprocessing*

3.3.1.1 Perancangan Proses *Tokenizing*

Tahap *tokenizing* terdapat 2 sub proses, *case folding* dan *parsing*.

Perancangan proses *tokenizing*ini digambarkan pada gambar 3.5 *flowchart* proses *tokenizing*. Tahap *tokenizing* terbagi dalam dua tahap, *case folding* dan *parsing*. Berikut penjelasan untuk proses *tokenizing* pada gambar 3.5.

1. Data dari*html*akan di simpan kedalam bentuk *txt* untuk proses *tokenizing*
2. Data teks akan dirubah kebentuk huruf kecil.
3. Variabel pemisah mengunakan semua tanda baca, sepasi dan karakter bukan alfabet.
4. Data teks bertipe string akan dipisah sesuai dengan *variable* pemisah.
5. Hasil data pemisahan berupa data teks yang terpisah.



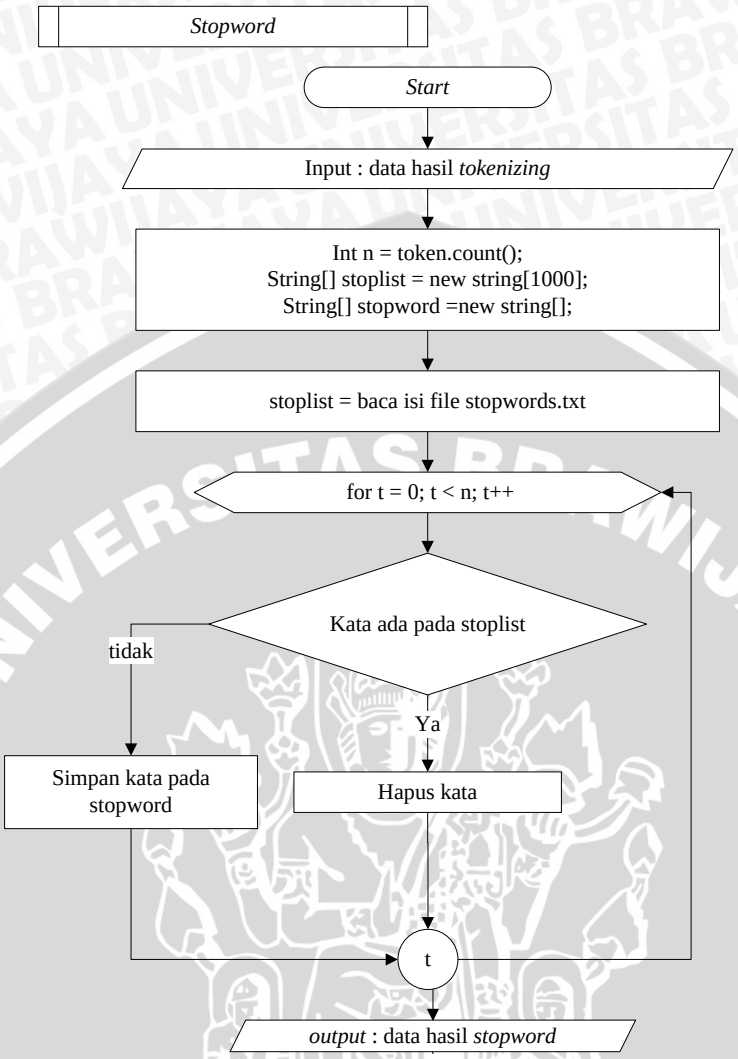
Gambar 3.5 Flowchart proses tokenizing

3.3.1.2 Perancangan Proses *Stopword*

Proses *stopword* adalah proses penghapusan kata-kata yang sering muncul tapi tidak memberikan informasi. Hasil proses *stopword* adalah kata-kata yang merepresntasikan informasi dari suatu dokumen berita. Data *stopword* diambil dari sebuah web kamus kata dasar bahasa Indonesia :

http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/

Sistem akan membaca dokumen teks berisi data *stopword*, kemudian disimpan pada *array string* stoplist. Dilakukan pengecekan setiap kata pada dokumen berita, jika kata terdapat pada *stopword* dilakukan penghapusan. Gambar perancangan proses *stopword* terdapat pada gambar 3.6 *flowchart* proses *stopword*.



Gambar 3.6 *Flowchart* proses *stopword*

Tahap *stopword,* dilakukan pengecekkan pada *list stopword* yang sudah disiapkan. Berikut deskripsi dari proses *stopword* pada gambar 3.6 *flowchart* proses *stopword*.

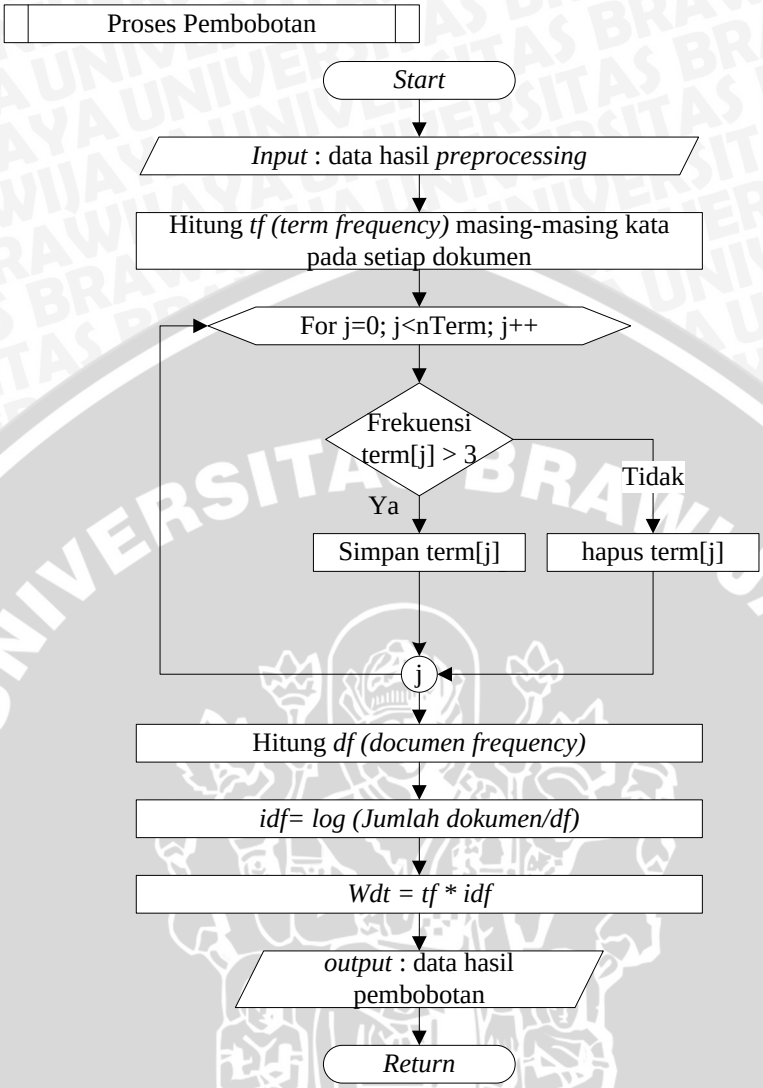
1. Data berupa kumpulan kata hasil *tokenizing* dimasuk dalam sistem.
2. Baca file stopwords.txt simpan pada string[] stoplist.
3. Satu persatu kata melalui proses pengecekan, jika kata ada dalam *list stopword* dilakukan penghapusan kata.
4. Kata tidak ada dalam *stopword,* dilakukan penyimpanan kata pada stopword.
5. Hasil dari sistem adalah kumpulan kata yang tidak ada pada stoplist.

3.3.2 Perancangan Proses *tf-idf*

Selanjutnya proses setelah melalui pencarian kata dasar, dilakukan proses pembobotan kata. Pembobotan kata adalah proses pencarian bobot kata dari sebuah dokumen satu ke dokumen lainnya. Dalam proses ini digunakan metode *term frequency - inverse document frequency (tf-idf).*

Proses pembobotan kata dapat dilihat dari gambar 3.7 *flowchart* proses pembobotan.Pada gambar 3.7 dijelaskan proses pembobotan menggunakan metode tf-idf, berikut penjelasan proses pembobotan.

1. User masukkan kumpulan kata yang ada didalam suatu dokumen berita.
2. Perhitungan banyaknya kata yang muncul dalam satu dokumen.
3. Proses *feature selection*, kata yang akan diproses selanjutnya adalah kata yang memiliki frekuensi kemunculan >=3 pada satu dokumen.
4. Dilakukan perhitungan jumlah kata yang muncul pada suatu dokumen, proses tersebut adalah perhitungan *documen frequency.*
5. Setelah hasil df (*documen frequency) diketahui*, proses berlanjut pada perhitungan *inversdocumen frequency* dengan menggunakan persamaan 2.2.
6. Diketahui hasil idf dari setiap dokumen, proses selanjutnya menggunakan persamaan 2.1, dengan mengalikan setiap kemunculan kata dengan hasil idf dari tiap dokumen.
7. Output hasil dari pembobotan berupa kata dan bobot kata.

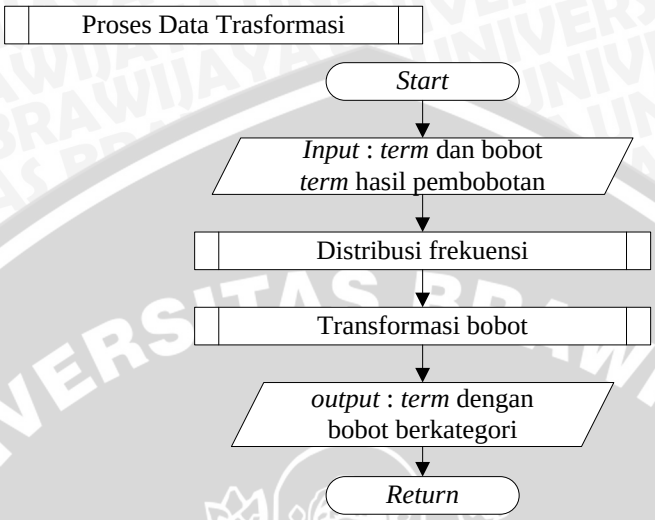


Gambar 3.7 *Flowchart* proses pembobotan

3.3.3 Perancangan Data Transformasi

Pada sub bab perancangan transformasi data, dilakukan perubahan dari data numerik menjadi data kategori. Dilakukan transformasi data dengan cara seperti ini karena C5.0 tidak dapat memproses data dengan tipe numerik, maka data pembobotan yang bertipe numerik akan diletakan dalam kelompok-kelompok dengan range data tertentu. Terdapat dua proses dalam perancangan transformasi data, distribusi frekuensi yang merupakan metode pembuatan range dari data hasil pembobotan dan pengkategorian bobot merupakan proses pemberian kategori pada bobot sesuai hasil range kategori yang dihasilkan dari distribusi frekuensi.

Untuk perancangan proses data transformasi digambarkan pada gambar 3.8 *flowchart* proses data transformasi.



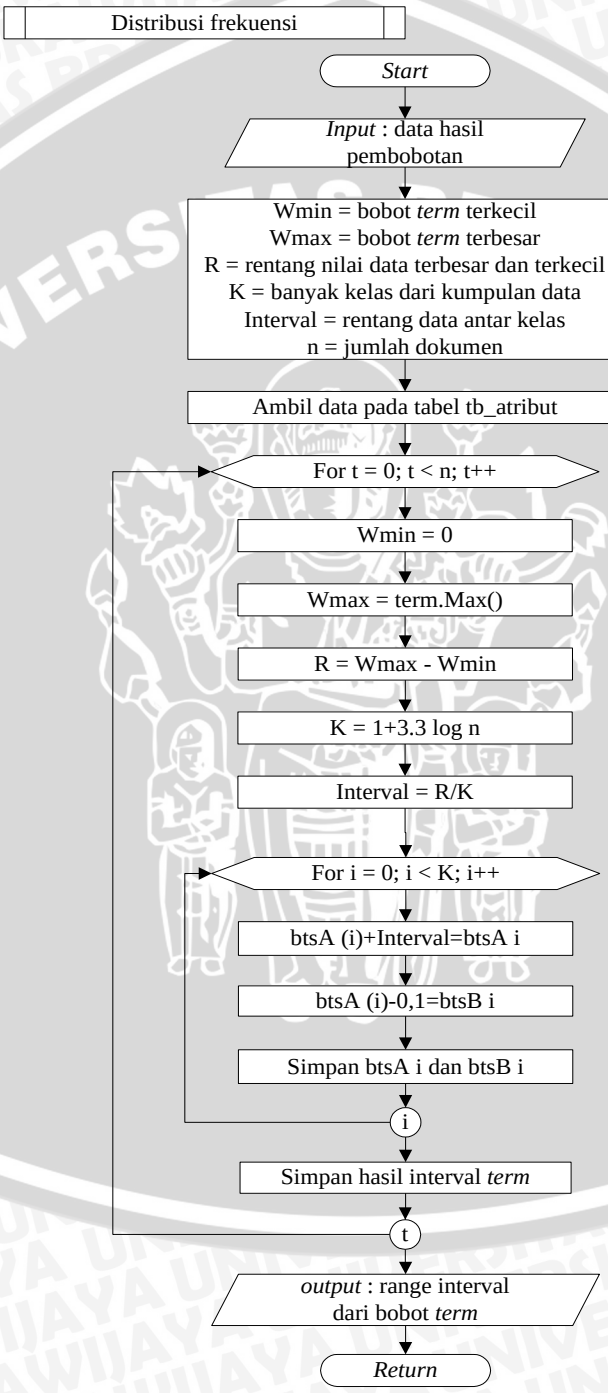
Gambar 3.8 *Flowchart* proses data transformasi

Untuk sub proses pertama pada proses data transformasi adalah distribusi frekuensi. Metode distribusi frekuensi digunakan untuk menentukan pembuatan kelompok data berdasarkan persebaran kumpulan data numerik hasil proses pembobotan.

Dari gambar 3.9 *flowchart* proses distribusi frekuensi, berikut tahap pembuatan interval kelas dengan metode distribusi frekuensi.

1. Sejumlah kata (*term*) dalam suatu dokumen di inputkan kedalam sistem.
2. Dilakukan perulangan sebanyak jumlah kata.
3. Pada nilai Wmin ditetapkan 0, karena kata yang tidak ada dalam dokumen memiliki nilai 0 sehingga tidak perlu dilakukan lagi pencarian nilai minimum.
4. Mencari nilai maksimum dari bobot kata.
5. Menghitung jumlah kelas yang dapat terbentuk dari kumpulan bobot kata.
6. menghitung interval setiap kelas.
7. Untuk menentukan batas atas kelas (btsA) dan batas bawah kelas (btsB), btsA dimulai dari 0 dijumlahkan dengan interval kelas dan btsB didapat dari btsA di kurangi nilai terkecil dari dokumen pada penelitian ini menggunakan 0,1 sebagai nilai terkecil karna data numerik dari bobot berbentuk desimal.

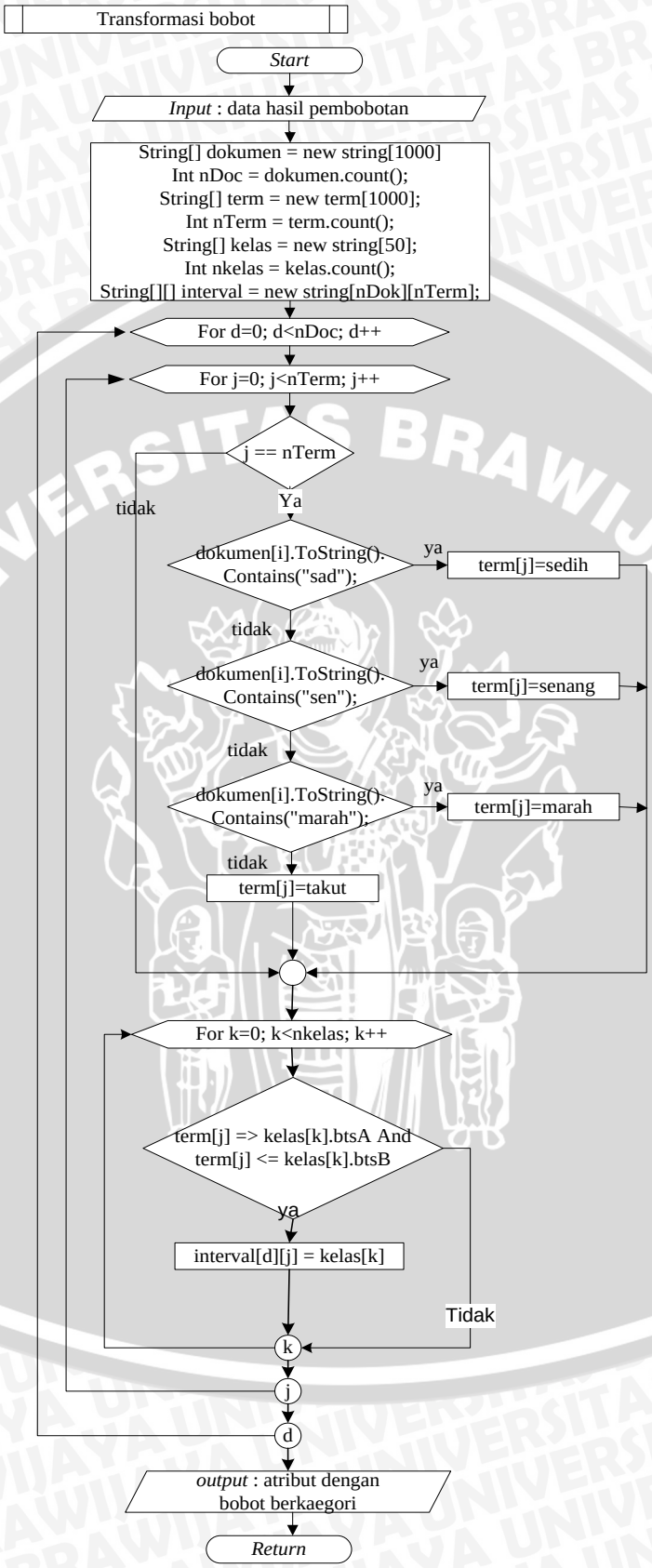
8. Simpan hasil btsA dan btsB untuk tiap perulangan sampai dengan sejumlah kelas yang terbentuk.
9. Simpan hasil range interval *term.*
10. Output yang dihasilkan berupa kelompok range interval dari bobot term.



Gambar 3.9 *Flowchart* proses distribusi frekuensi

Setelah dilakukan pembuatan interval setiap kelas dengan metode distribusi frekuensi, selanjutnya dilakukan proses transformasi data. Pada proses ini data akan dimasukan ke-dalam interval sesuai dengan hasil distribusi frekuensi. Berikut proses transformasi data yang ditunjukan pada gambar 3.10 *flowchart* proses transformasi bobot.

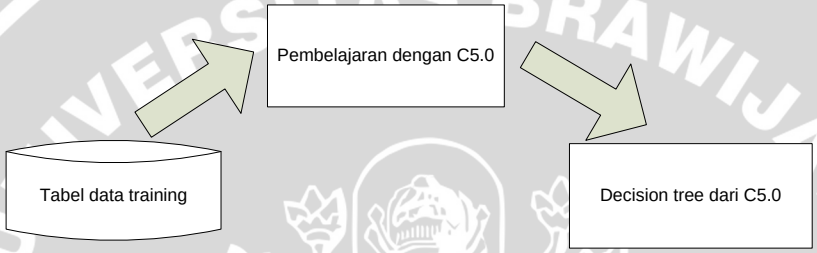
1. Data inputan adalah hasil proses pembobotan.
2. Bentuk *array string* untuk menyimpan dokumen. Bentuk *integer* nDoc untuk menyimpan banyak dokumen. Bentuk *array* term tipe *string* untuk menyimpan kata. Bentuk *integer* nTerm untuk menyimpan banyak kata. Bentuk *array* kelas bertipe *string* untuk menyimpan data kelas. Bentuk *integer* nkelas untuk menyimpan banyak kelas.
3. Lakukan perulangan sejumlah dokumen.
4. Lakukan perulangan sejumlah kata pada dokumen indek ke-d.
5. Kondisi j sama dengan nTerm, dilakukan pengkategorian kelas tujuan.
6. Dokumen ke-d string pertama diawali "sed". Jika kondisi terpenuhi, atribut tujuan term ke-j adalah sedih. Jika kondisi tidak terpenuhi lanjut ke proses selanjutnya.
7. Dokumen ke-d string pertama diawali "sen". Jika kondisi terpenuhi, atribut tujuan term ke-j adalah senang. Jika kondisi tidak terpenuhi lanjut ke proses selanjutnya.
8. Dokumen ke-d string pertama diawali "mar". Jika kondisi terpenuhi, atribut tujuan term ke-j adalah marah.Jika kondisi tidak terpenuhi, atribut tujuan term ke-j adalah takut.
9. Indek ke-j sama dengan nTerm. tidak terpenuhi, dilakukan pengkategorian bobot term sesuai hasil distribusi frekuensi.
10. Lakukan perulangan sebanyak nKelas yang terbentuk.
11. term[j] => kelas[k].btsA dan term[j] <= kelas[k].btsB. Kondisi terpenuhi, interval dokumen ke-d term ke-j memiliki kelas interval ke-k. Kondisi tidak terpenuhi, kembali ke perulangan poin 10.
12. Hasil dari proses transformasi bobot adalah atribut dengan bobot berkategori.
13. Proses berlajut ke sub proses selanjutnya.



Gambar 3.10 *Flowchart* proses transformasi bobot

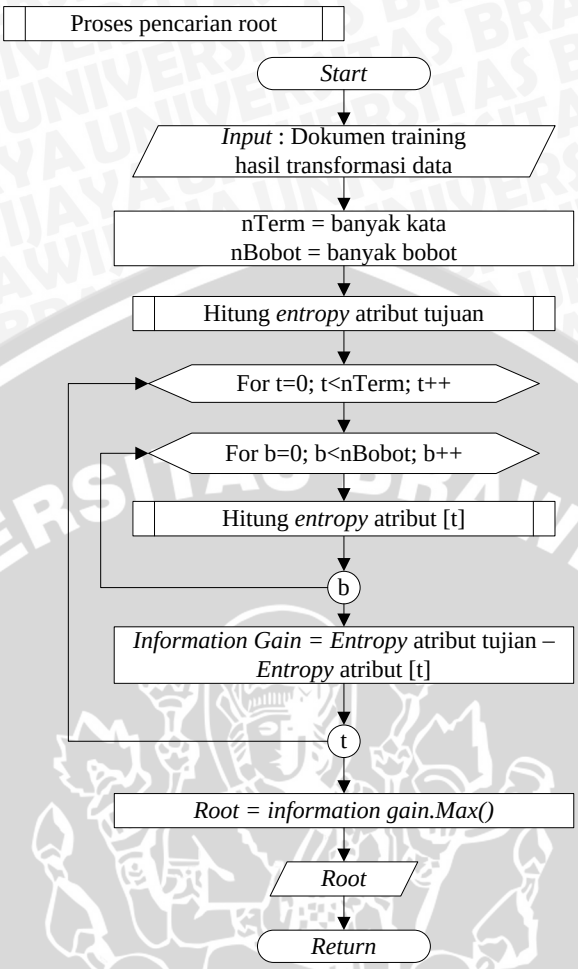
3.3.4 Perancangan Proses Latih

Pada tahap pembelajaran dokumen berita yang disiapkan untuk pemebelajaran akan melalui proses pembelajaran dengan C5.0. Skema proses pembelajaran dijelaskan pada gambar 3.11, dari data dokumen pembelajaran akan melalu proses pengklasifikasian secara manual untuk dikenali oleh sistem. Hasil dari pembelajaran sistem dengan C5.0 berupa *decision tree*. Berikut gambar penjelasan proses pembelajaran dijelaskan pada Gambar 3.12 *Flowchart* proses latih C5.0.



Gambar 3.11 *Flowchart* proses latih C5.0

Pada penelitian ini pembelajaran C5.0 dibagi dalam dua sub proses, sub proses pertama adalah proses pencarian *root* utama dari *decision tree* dan sub proses kedua adalah proses pebentukan *node decision tree*. Perancangan proses latih digambarkan pada gambar 3.11 *flowchart* .

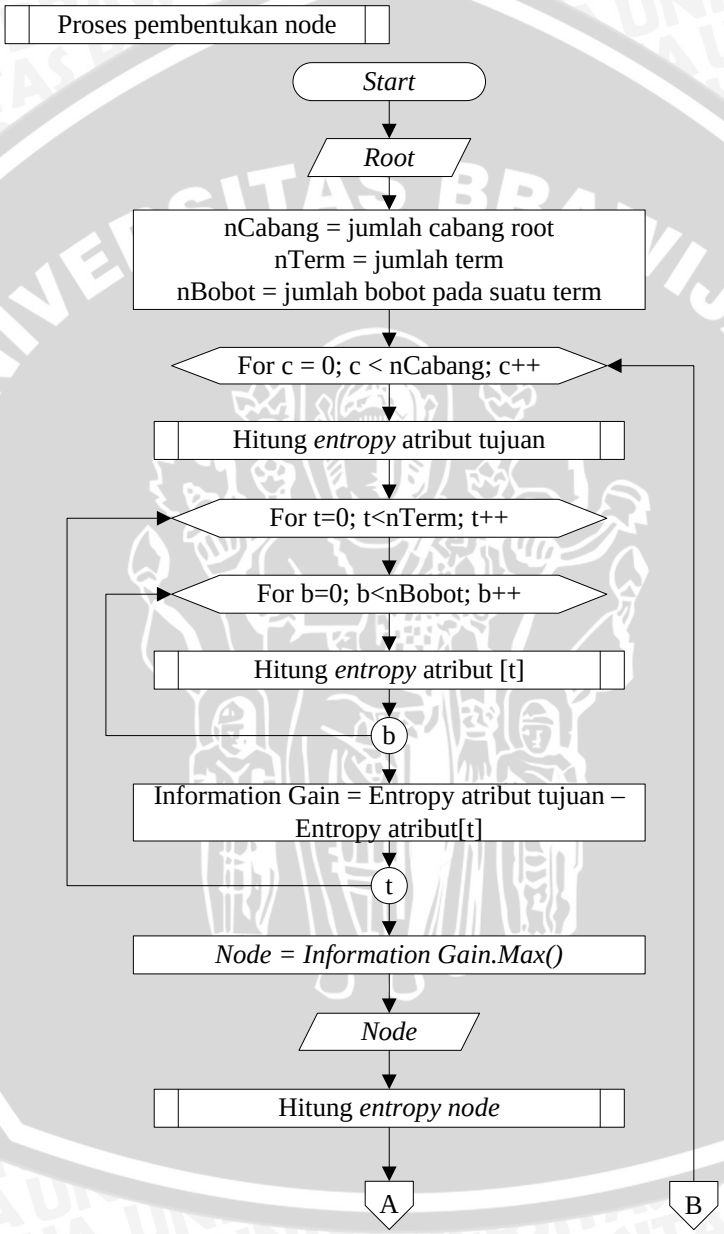


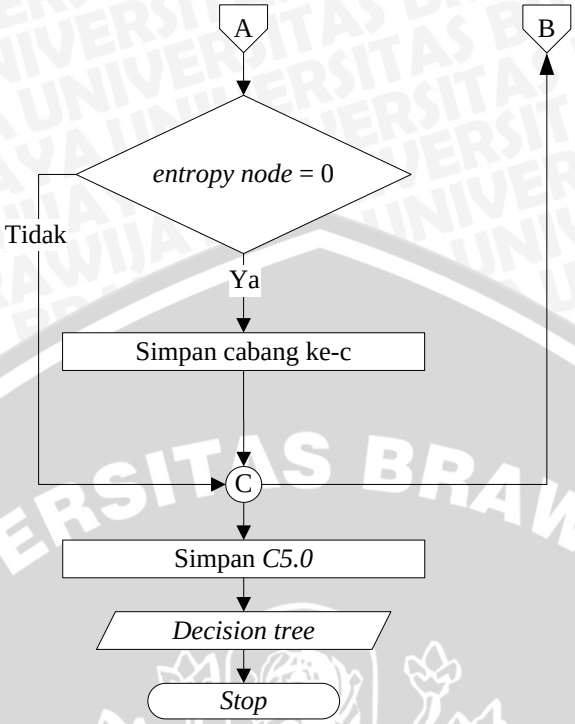
Gambar 3.12 *Flowchart* proses pencarian *root*

Pada tahap pertama dijelaskan proses penentuan *root* dari perbandingan *entropy class* dengan *entropy* atribut. Penjelasan proses pencarian *root* dijelaskan pada gambar 3.12 *flowchart* proses pencarian *root*. Berikut penjelasan proses penentuan *root* dari *decision tree.*

1. Data masukan pada proses ini berasal dari hasil transformasi data.
2. Bentuk *integer* nTerm untuk menyimpan banyak kata. Bentuk *integer* nBobot untuk menyimpan banyak bobot.
3. Hitung *entropy* kelas tujuan.
4. Lakukan perulangan sebanyak nTerm.
5. Lakukan perulangan sebanyak nBobot.
6. Hitung *entropy* atribut pada term ke-t bobot ke-b.
7. Hitung informasi *gain* pada setiap term ke-t.

8. *Root* adalah hasil perhitungan setiap term informasi *gain* yang memiliki nilai tertinggi.
9. Hasil dari proses pencarian *root* adalah satu kata sebagai *root*.
10. Proses berlanjut ke pencarian node.





Gambar 3.13 *Flowchart* proses pembentukan *node*

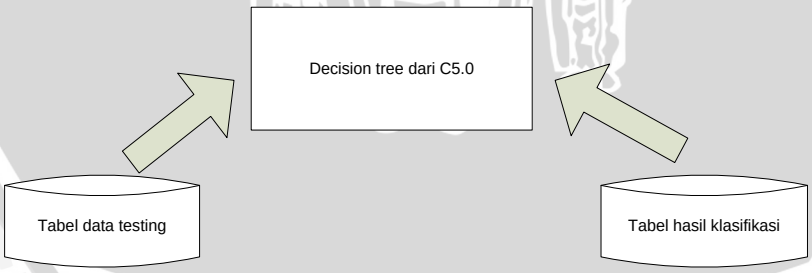
Pada tahap selanjutnya adalah tahap pembentukan *node*, dari hasil penentuan *root* pada proses sebelumnya selanjutnya cabang setiap root akan dilakukan perhitungan *entropy* dan perbandingan *information gain* sampai pada cabang terdapat satu *class* yang sama atau hasil *entropy* dari node bernilai nol. Berikut penjelasn proses pembentukan node yang digambarkan pada gmabar 3.13 *flowchart* proses pembentukan *node* .

1. Dari hasil *root* yang dihasilkan pada proses sebelumnya, kemudian dilakukan perhitungan *entropy class* setiap cabang root.
2. Dilakukan pengecekan pada cabang *root*, dan dilakukan perulangan pada setiap cabang *root*.
3. Pada tiap cabang akan dilakukn perhitungan masing – masing *entropyclass* pada tiap cabang dari *root*.
4. Lakukan perulangan sejumlah term atribut yang ada pada cabang.
5. Perulangan kedua dimana setiap *term* memiliki bobot kategori, dilakukan perulangan sampai sejumlah bobot kategori yang ada pada *term* atribut.
6. Dilakukan perhitungan *entropy* pada atribut.

7. Kembali ke perulangan kedua pada poin 5, sampai bobot pada *term* atribut tersebut habis.
8. Dilakukan perhitungan *information gain* dengn membandingkan *entropy class* dengan *entropy* atribut.
9. Setelah didapat *information gain*, diambil nilai terbesar dari *information gain* tersebut untuk digunakan sebagai *node*.
10. Dari hasil *node* dilakukan pengecekan *entropy class* yang berada pada cabang tersebut.
11. Jika *entropy class* bernilai nol, maka cabang tersebut merupakan akhir dari tree.
12. Menyimpan hasil perhitungan cabang ke-c tersebut.
13. *Entropy* tidak bernilai nol maka kembali ke cabang selanjutnya pada point ke dua.
14. Hasil dari proses ini berupa pohon keputusan dari proses pembelajaran.

3.3.5 Perancangan Proses *Testing*

Berikut merupakan skema dari proses klasifikasi ditunjukan pada gambar 3.15. Dokumen testing yang telah disiapkan akan dilakukn pengklasifikasian sesuai hasil dari pembelajaran sistem pada proses pembelajaran. Hasil dari proses klasifikasi berupa dokumen yang sudah memiliki kategori.

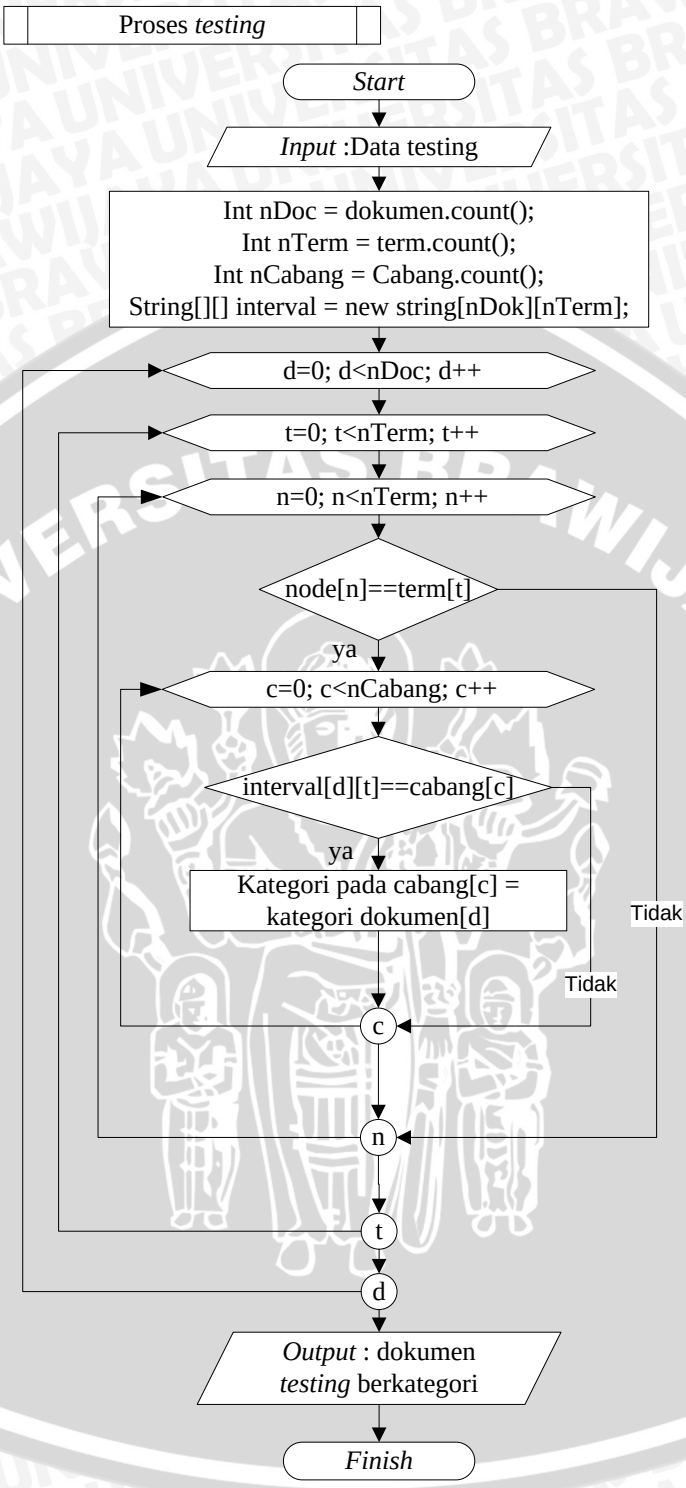


Gambar 3.14 Proses klasifikasi

Pada gamar *flowchart* 3.14 yang merupakan proses klasikasi penemuan kategori untuk dokumen, berikut penjelasan proses pengklasifikasian.

1. Data masukan pada proses testing adalah data tes hasil transformasi data.

2. Bnetuk integer nDoc untuk menyimpan banyak dokumen test. Bentuk integer nTerm unutk menyimpan banyak kata. Bentuk integer nCabang untuk menyimpan banyak cabang dicesion tree. Bnetuk array interval bertipe string untuk menyimpan interval bobot kata.
3. Lakukan perulangan sejumlah dokumen test.
4. Lakukan perulangan sejmlah kata pada dokumen ke-d.
5. lakukan perulangan sejumlah node.
6. NOde ke-n sama dengan term ke-t. Kondisi terpenuhi berlanjut ke pengecekan cabang.
7. Lakukan pengecekan cabang node ke-n.
8. interval dokumen ke-d term ke-t sama dengan cabang ke-c. Kondisi memenuhi, kategori cabang ke-c adalah kategori dokumen ke-d.
9. Kondisi poin 8 tidak terpenuhi kembali pada perulangan point 7.
10. Hasil dari proses ini adalah dokumen tes berkategori.



Gambar 3.15 *Flowchart* proses *testing*

3.4 Perhitungan Manual

Pada perhitungan manual ini membahas contoh perhitungan manual untuk data training, data testing, pembobotan tf-idf, proses generalisasi dengan distribusi frekuensi dan pengklasifikasian dengn algoritma C5.0. Perhitungan manual diawali setelah kata melalui tahap *preprocessing.*

3.4.1 Sumber Data

Data yang digunakan dalam perhitungan manual berjumlah 12 dokumen berita. 12 dokumen berita tersebut dibagi menjadi dua bagian, 8 dokumen berita sudah mempunyai kategori yang nantinya akan di gunakan sebagai contoh perhitungan dokumen *training*. 4 dokumen berita sisanya tidak memiliki kategori dan digunakan sebagai data *testing.* 8 dokumen *training* terbagi menjadi empat kategori emosi, 2 dokumen dalam kategori sedih, 2 dokumen dalam kategori senang, 2 dalam kategori marah, dan 2 sisanya dalam kategori takut. Dokumen dalam perhitungan manual dan perhitungan jumlah frekuensi kata ada pada lampiran.

3.4.2 Proses Pembobotan

Setelah melakukan perhitungan frekuensi setiap kata pada 12 dokumen, kemudian dilakukan *stopword* dan *stemming* secara manual. Data yang diambil dalam proses selanjutnya adalah kata yang memiliki frekuensi leih dari sama dengan 3. Daftar kata yang dihasilkan dari tahap processing tersebut terdapat dalam dua tabel berikut, tabel 3.1 hasil *preprocessing* dari data *training* dan tabel 3.2 hasil *preprocessing* untuk data *testing*. Kedua tabel yang berisi kata tersebut akan dilakukan proses pembobotan. Berikut hasil perhitungan *term frequency*, *document frequency* dan *invers document frequency*, perhitungan idf dengan menggunakan persamaan 2.1 dan dengan jumlah dokumen sebanyak 8 untuk data *training* dan 4 untuk data *testing.*

Berikut contoh perhitungan idf pada salah satu kata paksa :

$$idf = \log \frac{N}{df}$$

$$idf(paksa) = \log \frac{8}{2} = \log 4 = \mathbf{0.60206}$$

Tabel 3.1

Data training Term-frequency, doc frequency dan Invers doc frequency

term	D1	D2	D3	D4	D5	D6	D7	D8	df	Idf
kesal	2	0	0	0	0	0	0	0	1	0.90309
paksa	2	3	0	0	0	0	0	0	2	0.60206
takut	0	0	4	0	0	0	0	0	1	0.90309
ngeri	0	0	0	2	0	0	0	0	1	0.90309
suka	0	0	0	0	3	0	0	0	1	0.90309
senang	2	0	0	0	0	3	0	0	2	0.60206
korban	0	0		0	0	0	7	0	1	0.90309
bunuh	0	0	1	0	0	0	0	4	2	0.60206

Tabel 3.2

Data testing Term-frequency, doc frequency dan Invers doc frequency

Term	D1	D2	D3	D4	df	idf
Kesal	5	0	0	0	1	0.60206
Paksa	4	0	0	0	1	0.60206
Takut	2	0	0	0	1	0.60206
Suka	0	3	0	0	1	0.60206
senang	0	7	0	0	1	0.60206
korban	0	0	3	1	2	0.30103
bunuh	0	0	0	5	1	0.60206

Dengan telah diketahui hasil *idf* selanjutnya dilakukan pembobotan kata pada setiap dokumen menggunakan persamaan 2.2.

Contoh perhitungan berikut untuk kata **korban** pada data training..

$wdt = 7\ x\ 0.90309 = 6.32163$

Table 3.3 Data training hasil pembobotan

	Wdt							
Term	D1	D2	D3	D4	D5	D6	D7	D8
kesal	1.80618	0	0	0	0	0	0	0
paksa	1.20412	2.70927	0	0	0	0	0	0
takut	0	0	3.61236	0	0	0	0	0
ngeri	0	0	0	1.80618	0	0	0	0
Suka	0	0	0	0	2.70927	0	0	0
Senang	1.80618	0	0	0	0	2.70927	0	0
Korban	0	0	0	0	0	0	6.32163	0
bunuh	0	0	0.60206	0	0	0	0	2.40824

Tabel 3.4 Data testing hasil pembobotan

Term	Wdt			
	D1	D2	D3	D4
Kesal	3.0103	0		0
Paksa	2.40824	0	0	1.80618
Takut	1.20412	0	0	0
Nferi	0	0	0	0
suka		1.80618		
Senang		4.21442		
Korban			1.80618	
Bunuh			0.30103	1.50515

3.4.3 Proses Data *Transformation*

Dalam penerapan kedalam sisitem distribusi frekuensi dilakukan pada setiap atribut karna setiap atribut memiliki data bobot berbeda. Pada perhitungan manual ini distribusi frekuensi tidak dilakukan ke seluruh atribut karna jumlah dokumen terbatas, maka dilakukan distribusi frekuensi keseluruh data hasil pembobotan dari rata training. Berikut langkah2 untuk mebuat distribusi frekuensi.

1. Hasil pembobotan data training dilakukan pengurutan data dan mencari data terkecil dan data terbesar (X_{min}dan X_{max}). Dilakukan pencarian range anatara kedua data tersebut dengan persamaan 2.3.

   Hasil pengurutan pembobotan :

Tabel 3.5 Tabel hasil pengurutan data pembobotan

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	0.60206
0.60206	1.20412	1.80618	1.80618	1.80618	2.40824
2.70927	2.70927	2.70927	3.61236	6.32163	

dari tabel diatas diketahui nilai maksimum dari data adalah 6.32163 dengan nilai minimum dari data adalah 0 dengan 6.32163 jumlah data sampel 64. Perhitungan rentang niali minimum dengan maksimum:

R = 6.32163 - 0 = **6.32163**

2. Untuk menentukan banyak kelas yang akan digunakan dengan menggunakan persamaan 2.3 :

   $K = 1 + 3.3 \log 66$

   Hasil dari perhitungan kelas tersebut dapat dibuat **7 kelas.**

3. Menentukan panjang interval setiap kelas menggunakan persamaan :

   Interval $= \frac{6.32163}{7} =$ **0.9**

   maka interval kelas dengan panjang 1

4. Dipilih ujung paling bawah kelas interval pertama. Dengan begitu dapat diambil sama dengan data terkecil atau data lebih kecil, namum selisih harus kurang dari panjang kelasnya. Diambil 0 sebagai sampel data terkecil, menentukan batas kelas atas dengan menambahkan 0 dengan interval kemudian kurangi batas atas kelas dengan skala terkecil dari data. Skala terkecil data 0.1. maka kelas pertama 0 hingga 0,9, dan interval kelas kedua 1-1,9 dan seterusnya

5. Seetiap ramge interval diberikan nama agar memudahkan pemanggilan record data. Untuk mengatasi data dengan nilai bobot lebih dari Xmax dalam distribusi frekuensi, oleh karena itu data maksimum mendapat perlakuan berbeda dengan menggunakan kondisi data x > max. hasilnya pada tabel 3.5

6.

Tabel 3.6 tabel hasil pengurutan interval

Data interval	Nilai interval	Frekuensi
Int1	0 – 0.9	54
Int2	1 – 1.9	4
Int3	2 – 2.9	4
Int4	3 – 3.9	1
Int5	4 – 4.9	0
Int6	5 – 5.9	0
Int7	X >6	1
Jumlah data		66

Berikut hasil transformasi data dan label kategori dari setiap dokumen ditunjukkan pada tabel 3.6.

Tabel 3.7 Training data hasil transformasi data

	kesal	paksa	takut	ngeri	suka	senang	korban	bunuh	Kategori
D1	Int2	Int2	Int1	Int1	Int1	Int2	Int1	Int1	Marah
D2	Int1	Int3	Int1	Int1	Int1	Int1	Int1	Int1	Marah
D3	Int1	Int1	Int4	Int1	Int1	Int1	Int1	Int1	Marah
D4	Int1	Int1	Int1	Int2	Int1	Int1	Int1	Int1	Marah
D5	Int1	Int1	Int1	Int1	Int3	Int1	Int1	Int1	Senang
D6	Int1	Int1	Int1	Int1	Int1	Int3	Int1	Int1	Senang
D7	Int1	Int1	Int1	Int1	Int1	Int1	Int7	Int1	Marah
D8	Int1	Int1	Int1	Int1	Int1	Int1	Int1	Int3	Marah

3.5 Pembentukan Pohon Keputusan

Sub bab ini akan membahas perhitungan dalam pembetukkan pohon keputusan beserta contohnya. Dengan menggunakan data dari tabel 3.6 hasil proses transformasi data, dimana dat sudah dalam bentuk kategori sehingga lebih mudah untuk digunakan untuk perhitungan pada persamaan C 5.0. Berikut langkah-langkahnya dalam perhitungan pembetukan pohon\

1. Pembetukan Simpul akar utama

   Dilakukan perbandingan entropy atribut kategori dengan atribut kata dengan menggunakan persamaan 2.3.

   Hitung *entropy* kategori (S)

   Dimana S adalah 8 data dari tabel kategori pada tabel 3.6 dengan prediksi 6 kategori marah, 2 kategori senang.

$$I(6,2) = \left(-\frac{6}{8}log2\left(\frac{6}{8}\right)\right) + \left(-\frac{2}{8}log2\left(\frac{2}{8}\right)\right)$$

$$= 0.31127 + 0.49997$$

$$= \mathbf{0.8113}$$

3.5.1 Menghitung *entropy* atribut dan menetukan *information gain*

Berikut contoh berhitungan entropy dan information gain untuk atribut **korban**, *information gain* dengan persamaan 2.4. Dengan jumlah data 8, record data int7 dengan jumlah 1 dalam kategori marah, record data int1 dengan jumlah data 7 (dalam kategori marah 5, kategori senang berjumlah 2).

Entropy korban int7 1 (1.0) , dan int1 7 (5,2)

Entropy int7

$$(1,0) = \left(-\frac{1}{1}\log 2\left(\frac{1}{1}\right)\right) + \left(-\frac{0}{1}\, log2\left(\frac{0}{1}\right)\right) = 0$$

Entropy int1

$$(5,2) = \left(-\frac{5}{7}\log 2\left(\frac{5}{7}\right)\right) + \left(-\frac{2}{7}\, log2\left(\frac{2}{7}\right)\right)$$

$$= 0.34672 + 0.51638$$

$$= \mathbf{0.8632}$$

Sebelum menghitung nilai gain, terlebih dahulu menghitung entropy total pada korban, dengan menggunakan persamaan 2.5 sebagai berikut:

$$\text{E(korban)} = \frac{1}{8}(0) + \frac{7}{8}(0.8631)$$

$$= 0 + 0.875\,(0.8631)$$

$$= \mathbf{0.7553}$$

Information gain korban

Menghitung nilai gain kata korban menggunakan persamaan 2.6

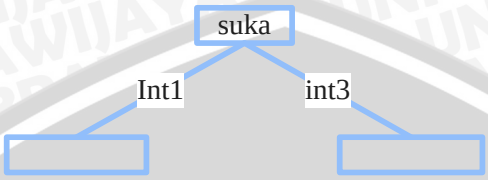
$$G(korban) = 0.8113 - 0.7553 = \mathbf{0.0561}$$

Berikut hasil perhitungan entropi serta information gain (IG) pada seluruh atribut pada tabel dibawah ini:

Tabel 3.8 Hasil perhitungan nilai *Gain*

	kesal	Paksa	Takut	ngeri	suka	senang	bunuh	korban
Int1	0.8632	0.9183	0.8632	0.8632	0.5916	0.6500	0.8632	0.8632
Int2	0	0	-	0	-	0	-	-
Int3	-	0	-	-	0	0	0	-
Int4	-	-	0	-	-	-	-	-
Int5	-	-	-	-	-	-	-	-
Int6	-	-	-	-	-	-	-	-
Int7	-	-	-	-	-	-	-	0
IG	0.0561	0.1225	0.0561	0.0561	**0.3675**	0.3237	0.0561	0.0561

Hasil tabel perhitungan gain di atas menunjukkan nilai gain pada atribut suka adalah yang terbesar dibanding dengan atribut lainnya yaitu dengan nilai **0.3675**.
Maka berikut gambar tree yang terbentuk dari perhitungan root.



Gambar 3.16 Tree perhitungan root

2. Cabang akar Int1 dari atribut suka.

Pengecekan atribut yang berada pada cabang Int1, data atribut dapat dilihat pada tabel

Tabel 3.9 Data atribut untuk cabang Int1 atribut suka

	kesal	paksa	takut	ngeri	suka	senang	korban	bunuh	Kategori
D1	Int2	Int2	Int1	Int1	Int1	Int2	Int1	Int1	**Marah**
D2	Int1	Int3	Int4	Int1	Int1	Int1	Int1	Int1	**Marah**
D4	Int1	Int1	Int1	Int2	Int1	Int1	Int1	Int1	**Marah**
D6	Int1	Int1	Int1	Int1	Int1	Int3	Int1	Int1	**Senang**
D7	Int1	Int1	Int1	Int1	Int1	Int1	Int7	Int1	**Marah**
D8	Int1	Int1	Int1	Int1	Int1	Int1	Int1	Int3	**Marah**

Tabel diatas mempunyai data untuk kelas kategori, maka dapat dilakukan perhitungan entropy. Jumlah data 6 dengan 4 data berkategori marah dan 1 pada kategori senang.

Entropy kategori (4,1)

Entropy Kategory S_{int1}

$$(5,1) = \left(-\frac{5}{6}\log 2\left(\frac{5}{6}\right)\right) + \left(-\frac{1}{6}\, log2\left(\frac{1}{6}\right)\right)$$

$$= 0.2191 + 0.4308$$

$$= \mathbf{0.6501}$$

Dengan perhitungan yang sebelumnya dilakukan untuk perhitungan yang sama entropy dan information gain pada seluru atribut dapat dilihat tabel dibawah ini.

Contoh perhitungan entropy dan gain untuk node pada atribut takut.

Jumlah Int1 ada 5 terdapat 4 dalam kategori marah dan 1 dalam kategori takut.

Int1 (5)

$$(4,1) \quad = \left(-\frac{4}{5}\log 2\left(\frac{4}{5}\right)\right) + \left(-\frac{1}{5}\, log2\left(\frac{1}{5}\right)\right)$$

$$= 0.2575 + 0.4643$$

$$= \mathbf{0.7219}$$

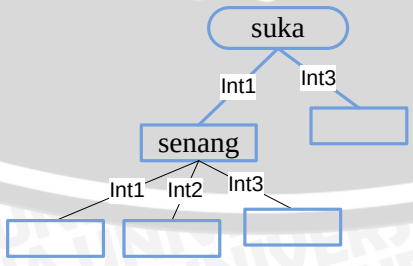
Hitung Gain kesal dari data dalam tabel diatas:

$G(takut) = 0.6501 - 0.7219 = \mathbf{0.0484}$

Tabel 3.10 hasil perhitungan *Entropy* dan *IG* untuk cabang Int1

	Kesal	Paksa	takut	ngeri	senang	bunuh	korban
Int1	0.6501	0.6501	0.6501	0.6501	0	0.6501	0.6501
Int2	0	0	-	0	0	-	0
Int3	-	-	-	-	0	0	-
Int4	-	-	-	-	-	-	-
Int5	-	-	-	-	-	-	-
Int6	-	-	-	-	-	-	-
IG	0.0484	0.0484	0.0484	0.0484	**0.6501**	0.0484	0.0484

Dari hasil perhitungan pada tabel atribut yang memiliki nilai IG terbesar adalah atribut senang dengan **0.6501**, maka atribut tersebut digunakan sebagai node dalam cabang atribut suka pada gambar 3.17



Gambar 3.17 Tree hasil cabang Int1 untuk atribut suka

2.1 Node cabang Int1 dari aribut senang.

Berikut contoh salah satu node cabang Int1 dari atribut senang. Pemanggilan fungsi untuk atribut senang dengan record data Int1, data berikut yang berada pada cabang Int1 atribut senang pada tabel dibawah ini.

Tabel 3.11 Data atribut untuk cabang Int1 atribut senang

	kesal	paksa	takut	ngeri	senang	korban	bunuh	**Kategori**
D2	Int1	Int3	Int4	Int1	Int1	Int1	Int1	**Marah**
D4	Int1	Int1	Int1	Int2	Int1	Int1	Int1	**Marah**
D7	Int1	Int1	Int1	Int1	Int1	Int7	Int1	**Marah**
D8	Int1	Int1	Int1	Int1	Int1	Int1	Int3	**Marah**

Melakukan perhitungan *entropy* untuk kelas kategori, 3 jumlah data yaitu 3 dalam berkategoi marah.

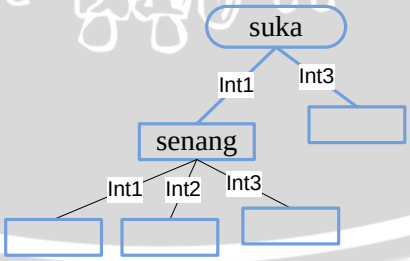
Entropy kategori 3 (3,0)

$$(4,0) = \left(-\frac{4}{4}\log 2\left(\frac{4}{4}\right)\right) + \left(-\frac{0}{4}\ log 2\left(\frac{0}{4}\right)\right)$$

$$= 0 + 0 = 0$$

Hasil *entropy* kelas nilai = 0

Dengan demikian sampel data yang berada pada cabang Int1 untuk atribut senang sudah dalam atribut yang sama, oleh karena itu tidak perlu dilakukan pemanggilan rekursif lagi dan proses kembali ke node awal. Cabang Int1 untuk atribut senang berkategori emosi marah. Berikut gambar tree dari hasil cabang ini.



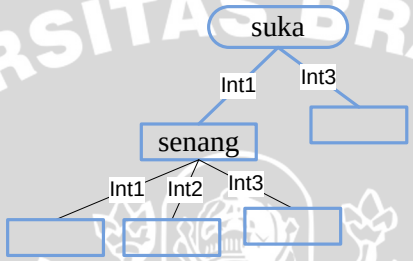
Gambar 3.18 Tree cabang Int1 atribut senang.

2.1 Node cabang Int2 dari atribut senang

Pemanggilan fungsi cabang Int2 pada atribut senang. Sampel data tersebut berada dalam atribut kelas utama, sehingga proses selanjutnya tidak perlu dilakukan. Cabang Int3 yang dimiliki atribut senang adalah marah.

Tabel 3.12 Data atribut untuk cabang Int2 atribut senang

	kesal	Paksa	takut	ngeri	Senang	korban	bunuh	**Kategori**
D1	Int2	Int2	Int1	Int1	Int2	Int1	Int1	**Marah**



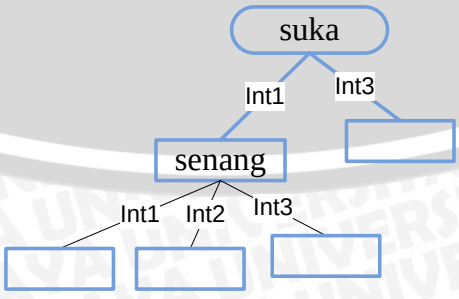
Gambar 3.19 Tree cabang Int2 atribut senang

3. Node cabang Int3 dari atribut senang

Cabang Int3 dalam atribut senang sudah berada pada atribut target, tidak perlu dilakukan pemanggilan kembali dan label masing-masing cabang sesuai atribung te yt yang tersedia pada atribut utama atau target yaitu dengan kater emosi senang. Seperti yang terlihat pada gambar *tree* di bawah ini

Tabel 3.13 Data atribut senang cabang Int3

	kesal	paksa	Takut	Ngeri	Senang	bunuh	korban	**Kategori**
D6	Int1	Int1	Int1	Int1	Int3	Int1	Int1	**Senang**



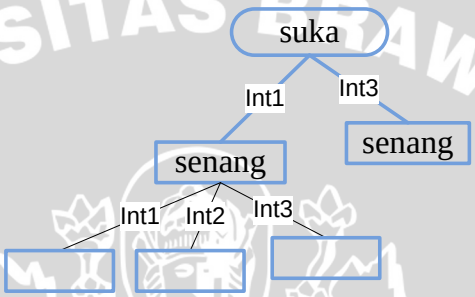
Gambar 3.20 Tree atribut senang pada cabang Int3

3. Node cabang Int3 dari atribut suka

Cabang Int3 dalam atribut suka sudah berada pada atribut target, tidak perlu dilakukan pemanggilan kembali dan label masing-masing cabang sesuai atribung te yt yang tersedia pada atribut utama atau target yaitu dengan kategori emosi senang. Seperti yang terlihat pada gambar *tree* di bawah ini

Tabel 3.14 Data atribut suka cabang Int3

	kesal	paksa	Takut	Ngeri	suka	senang	bunuh	korban	**Kategori**
D5	Int1	Int1	Int1	Int1	Int3	Int1	Int1	Int1	**Senang**



Gambar 3.21 Tree atribut suka pada cabang Int3

3.5.2 Perancangan Rule Tree

Dari data yang dihasilkan decision tree, selanjutnya untuk mendapatkan aturan dalam pengklasifikasian maka dilakukan dari hasil pengekstrakan *decision tree* tersebut.

Aturan klasifikasi yang didapat dari pembentukan pohon yang telah di hasilkan :

1. **IF** atribut suka = Int1 **AND** atribut senang = Int1 **THEN** dokumen teks berita berkategori Marah.
2. **IF** atribut suka = Int1 **AND** atribut senang = Int2 **THEN** dokumen teks berita berkategori Marah.
3. **IF** atribut suka = Int1 **AND** atribut senang = Int3 **THEN** dokumen teks berita berkategori Senang.
4. **IF** atribut suka = Int3 **THEN** dokumen berita berkategori Senang.

3.6 Perancangan Proses Klasifikasi

Dari hasil decision tree yang telah dibentuk pada sub bab sebelumnya dan telah dilakukan pengekstrakan sehingga menghasilkan aturan. Dalam tahap ini

proses klasifikasi dilakukan sesuai dengan hasil aturan klasifikasi yang telah terbentuk dalam dokumen pembelajaran.

Untuk data testing perlakuan yang sama dengan data training yakni dilakukan proses pembobotan terlebih dahulu. Setelah pada masing-masing atribut telah diketahui nilai bobotnya, akan dilakukan data transformasi. Setelah data transformasi tidak dilakukan perhitungan distribusi frekuensi seperti pada data training, data bobot pada testing cukup mengikuti hasil distribusi frekuensi yang didapat dari hasil data training.

Tabel 3.15 Hasil generalisasi Data testing

	Kesal	Paksa	takut	Dosa	suka	senang	korban	Bunuh
D1	Int4	Int2	Int4	Int1	Int1	Int1	Int1	Int1
D2	Int1	Int1	Int2	Int1	Int3	Int6	Int1	Int1
D3	Int1	Int1	Int1	Int7	Int1	Int1	Int3	Int1
D4	Int1	Int1	Int1	Int1	Int2	Int1	Int1	Int3

3.7 Perancangan Interface

Sistem klasifikasi emosi pada teks berita berbahasa Indonesia ini dibuat basis visual, bahasa yang digunakan pemrograman visual C#. Bagian sistem adalah satu form utama yang memiliki bagian-bagian form seperti yang dapat dilihat pada gambar

Gambar 3.22 Interface Sistem

Pada gambar 3.20 antarmuka sistem terdapat 7 bagian dari *form* tersebut yg memiliki kegunaan berbeda, berikut keterangan dari bagian gambar 3.23.

1. *Form* utama dari sistem klasifikasi berita.

2. Data, terdapat dua *button*, digunakan untuk mengambil dokumen berita. Keterangan teks di bawah *button* digunakan sebagai informasi lokasi data berita.
3. Proses data, terdapat empat *button* yaitu :
   - *Button* training digunakan setelah data melalui *preprocessing* dan menampilkan hasil preprocessing dalam *form* Data Berita.txt. Sistem akan melalui proses *preprocessing* pada tahap ini.
   - *Button* Testing digunakan setelah data testing selesai diambil menampilkan hasil preprocessing untuk data testing yang telah dipilih dalam *text area* Data Berita dan hasil proses klasifikasi penentuan kategori pada setiap dokumen testing.
4. Form Informasi Data, untuk menampilkan informasi jumlah data latih, jumlah data uji,waktu training, waktu testing, waktu perbaikan dan keterangan jumlah masing-masing kategori.
5. *Progress bar*, digunakan sebagai informasi untuk mengtahui brapa lama proses berlangsung.
6. *Form* ini digunakan untuk menampilkan data, terdapat 5 tab menu diantaranya:
   - Dokumen Berita, untuk menampilkan jumlah dokumen berita dalam bentuk text , dan kata apa saja yang dimasukkan kedalam sistem.
   - Frekuensi, untuk menampilkan hasil dari *frequency*.
   - DecisionTree, untuk menampilkan hasil pembentukan dan pemangkasan pohon yang dihasilkan sistem.
   - Hasil, menampilkan hasil dari klasifikasi sesuai data training atau data testing yang di pilih pada form data training dan data test.

3.8 Metode Pengujian

Setelah selesai melakukan pembuatan sistem, dilakukan pengujian terhadap sistem. Dalam penelitian ini metode pengujian dengan mengunakan perbandingan jumlah data terhadap perubahan prosentase. Pada tabel perbandingan data pengujian dengan prosentase data *training* terhadap data *testing* dengan jumlah data berita keseluruhan yang beragam. Pada bagian kolom

data pengujian berisi perbandingan data antara data *training* dengan data *testing*. Perbandingan data *training* dengan data *testing* dimulai dari 60% : 40%, 50% : 50%, 70% : 30%, 75% : 25%, 85% : 15% sampai dengan pengujian dengan perbandingan 90% : 10%. Berikut tabel hasil untuk penelitian ditunjukan pada tabel 3.15.

3.8.1 Perhitungan *Accuracy*

Telah didapat jumlah dari data pada nilai dan total dokumen pada pembahasan sebelumnya. Pada sub bab 2.12 dilakukan perhitungan *precision* dengan menggunakan rumus 2.10. berikut perancangan tabel penelitian untuk perhitungan precision terdapat pada tabel 3.16

. **Tabel 3.16** Contoh Tabel hasil pengujian

Pengujian training : testing		Total Data training	Total Data testing	Total Data pengujian	Ketepatan document	Akurasi
60%	40%					
50%	50%					
70%	30%					
75%	25%					
85%	15%					
90%	10%					