BAB 3
METODOLOGI PENELITIAN

3.1 Bahan dan Perangkat Penelitian

3.1.1 Bahan Penelitian

Pada penelitian ini bahan yang akan digunakan merupakan data yang diperoleh dari media sosial *Twitter* yang merupakan *tweet* dengan jumlah dataset sebanyak 1309 *tweet* terkait opini mengenai *childfree* yang diambil dari bulan 26 Februari 2023 hingga 08 Maret 2023.

3.1.2 Perangkat Penelitian

Perangkat penelitian merupakan perangkat yang digunakan dalam penelitian ini, meliputi perangkat keras dan perangkat lunak sebagai berikut:

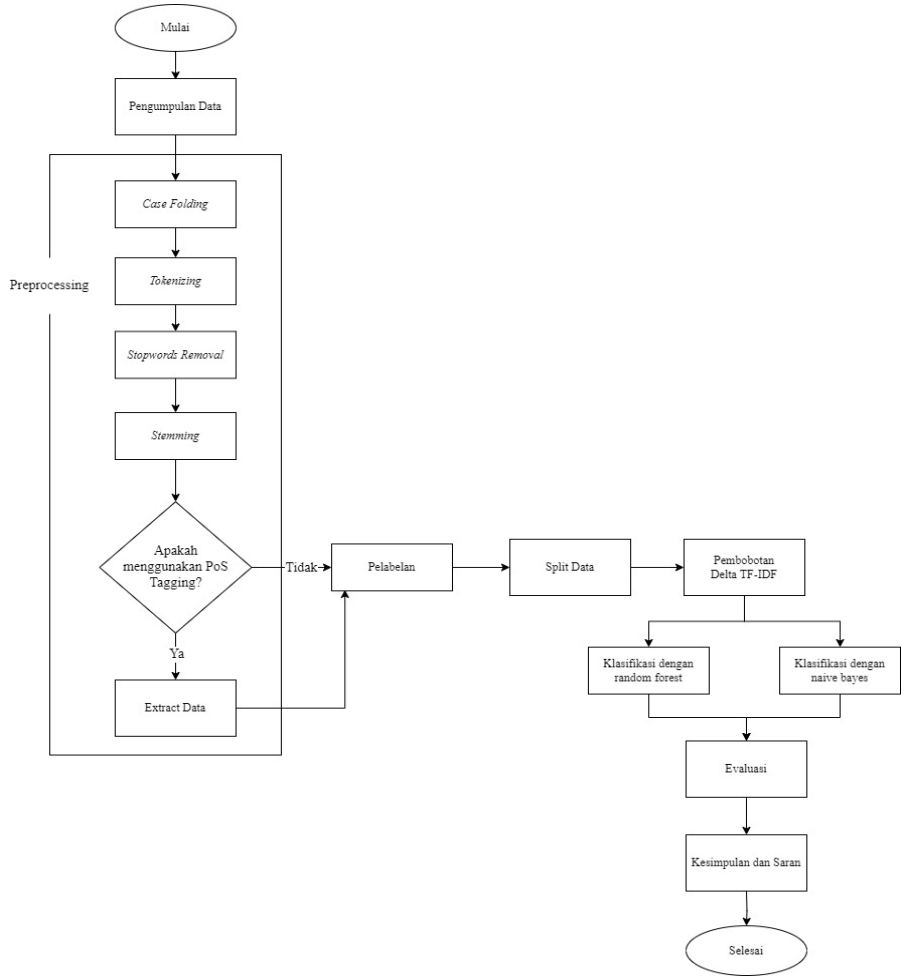
1. Perangkat Keras (*Hardware*)
   a. *Device* : Laptop
   b. *Processor* : Intel(R) Core (TM) i7-8650U CPU @ 1.90GHz
   c. *Harddisk* : 256 GB SSD
   d. RAM : 16.00 GB
2. Perangkat Lunak (*Software*)
   a. *Operating System* : Windows 10
   b. *Text Editor* : Google Colaboratory/Jupyter Notebook

3.2 Objek Penelitian

Objek penelitian ini merupakan masyarakat yang memiliki opini terkait *childfree* pada media sosial *Twitter* dengan rentang waktu pada 26 Februari 2023 hingga 08 Maret 2023.

3.3 Tahapan Penelitian

Dalam melakukan penelitian, dibuat diagram alir penelitian berupa tahapan-tahapan dalam melakukan penyelesaian terhadap suatu permasalahan. Berikut Gambar 3.1 merupakan tahapan penelitian yang akan dilakukan.



Gambar 3.1 Tahapan Penelitian

3.3.1 Pengumpulan Data

Pengumpulan data dilakukan dengan mengumpulkan dalam dua jenis data yaitu data primer dan data sekunder. Data primer dalam penelitian ini berupa pengambilan data dari *Twitter* yang berisi *tweet* opini masyarakat terkait *childfree*. Data yang digunakan dalam penelitian ini berjumlah 1309 *tweet* yang diambil dari tanggal 26 Februari 2023 hingga 08 Maret 2023. Sedangkan data sekunder pada penelitian ini berupa studi literatur terhadap jurnal-jurnal penelitian sebelumnya yang terkait dengan penelitian ini.

Teknik pengumpulan data dari *Twitter* dilakukan menggunakan *scrapping* yang nantinya disimpan dalam format .csv dan data tersebut berbasis lokasi karena berupa *tweet* dari masyarakat Indonesia mengenai *childfree,* selain itu *tweet* tersebut mengandung Bahasa Indonesia.

3.3.2 *Preprocessing*

Preprocessing merupakan tahapan pembersihan dan pengelolaan data yang nantinya akan dilakukan pembobotan dan analitis. Tahapan *text* preprocessing yang digunakan disini adalah *casefolding, tokenize, stopwords removal, stemming,* proses menggunakan dan tanpa menggunakan *PoS Tagging*, dan *extract data*.
Berikut merupakan penjabaran terkait tahapan *preprocessing* menggunakan 6 *dataset.*

a. *Case Folding*

Case folding adalah proses mengubah semua huruf dalam bentuk dokumen semua huruf menjadi huruf kecil. Berikut pada Tabel 3.1 merupakan hasil *case folding* dari data yang didapatkan.

Tabel 3.1 *Case Folding*

	original	**casefolding**
1	`~`kalau aku sendiri sih suka *childfree* licioust karena buat apa punya anak di dunia yang cuman hanya sebentar sementara sekejap saja sekedipan mata dan dalam tempo yang sangat singkat ini nanti anaknya malah tambah kasihan banget kita juga semakin capek repot pusing semuanyaΦ`~`	`~`kalau aku sendiri sih suka *childfree* licioust karena buat apa punya anak di dunia yang cuman hanya sebentar sementara sekejap saja sekedipan mata dan dalam tempo yang sangat singkat ini nanti anaknya malah tambah kasihan banget kita juga semakin capek repot pusing semuanyaφ`~`
2	niat makin yakin buat *childfree* atau satu deh maksimal, setelah buanyaaaak hal🤯	niat makin yakin buat *childfree* atau satu deh maksimal, setelah buanyaaaak hal🤯
3	@AREAJULID Yg kaya gini nih yg bikin berpikir utk *childfree*.. bkn krn biar awet muda, tp krn situasi dan kondisi diluaran semakin mengkhawatirkan.. anak ga salah aja bisa jd korban kaya begitu.. 🥺😭	@areajulid yg kaya gini nih yg bikin berpikir utk *childfree*.. bkn krn biar awet muda, tp krn situasi dan kondisi diluaran semakin mengkhawatirkan.. anak ga salah aja bisa jd korban kaya begitu.. 🥺😭

b. *Tokenize*

Tokenizing adalah proses memisahkan teks menjadi kata. Tabel 3.2 merupakan hasil yang didapatkan.

Tabel 3.2 *Tokenizing*

	casefolding	***tokenize***
0	`~`kalau aku sendiri sih suka *childfree* licioust karena buat apa punya anak di dunia yang cuman hanya sebentar sementara sekejap saja sekedipan mata dan dalam tempo yang sangat singkat ini nanti anaknya malah tambah kasihan banget kita juga semakin capek repot pusing semuanyaφ`~`	['kalau', 'aku', 'sendiri', 'sih', 'suka', 'childfre', 'licioust', 'karena', 'buat', 'apa', 'punya', 'anak', 'di', 'dunia', 'yang', 'cuman', 'hanya', 'sebentar', 'sementara', 'sekejap', 'saja', 'sekedipan', 'mata', 'dan', 'dalam', 'tempo', 'yang', 'sangat', 'singkat', 'ini', 'nanti', 'anaknya', 'malah', 'tambah', 'kasihan', 'banget', 'kita', 'juga', 'semakin', 'capek', 'repot', 'pusing', 'semuanya']
1	niat makin yakin buat *childfree* atau satu deh maksimal, setelah buanyaaaak hal🤯	['niat', 'makin', 'yakin', 'buat', 'childfre', 'atau', 'satu', 'deh', 'maksimal', 'setelah', 'buanyak', 'hal']
2	@areajulid yg kaya gini nih yg bikin berpikir utk *childfree*.. bkn krn biar awet muda, tp krn situasi dan kondisi diluaran semakin mengkhawatirkan.. anak ga salah aja bisa jd korban kaya begitu.. 🥺😭	['areajulid', 'yg', 'kaya', 'gini', 'nih', 'yg', 'bikin', 'berpikir', 'utk', 'childfre', 'bkn', 'krn', 'biar', 'awet', 'muda', 'tp', 'krn', 'situasi', 'dan', 'kondisi', 'diluaran', 'semakin', 'mengkhawatirkan', 'anak', 'ga', 'salah', 'aja', 'bisa', 'jd', 'korban', 'kaya', 'begitu']

c. *Stopwords Removal*

Stopwords removal atau *filtering* adalah proses menghilangkan kata-kata yang dianggap tidak memiliki makna (*stopwords removal*). Berikut pada Tabel 3.3 merupakan hasil yang didapatkan.

Tabel 3.3 *Stopwords*

	tokenize	***stopwords***
1	['kalau', 'aku', 'sendiri', 'sih', 'suka', 'childfre', 'licioust', 'karena', 'buat', 'apa', 'punya', 'anak', 'di', 'dunia', 'yang', 'cuman', 'hanya', 'sebentar', 'sementara', 'sekejap', 'saja', 'sekedipan', 'mata', 'dan', 'dalam', 'tempo', 'yang', 'sangat', 'singkat', 'ini', 'nanti', 'anaknya', 'malah', 'tambah', 'kasihan', 'banget', 'kita', 'juga', 'semakin', 'capek', 'repot', 'pusing', 'semuanya']	['sih', 'suka', 'childfre', 'licioust', 'anak', 'dunia', 'cuman', 'sebentar', 'sekejap', 'sekedipan', 'mata', 'tempo', 'singkat', 'anaknya', 'kasihan', 'banget', 'capek', 'repot', 'pusing']
2	['niat', 'makin', 'yakin', 'buat', 'childfre', 'atau', 'satu', 'deh', 'maksimal', 'setelah', 'buanyak', 'hal']	['niat', 'childfre', 'deh', 'maksimal', 'buanyak']
3	['areajulid', 'yg', 'kaya', 'gini', 'nih', 'yg', 'bikin', 'berpikir', 'utk', 'childfre', 'bkn', 'krn', 'biar', 'awet', 'muda', 'tp', 'krn', 'situasi', 'dan', 'kondisi', 'diluaran', 'semakin', 'mengkhawatirkan', 'anak', 'ga', 'salah', 'aja', 'bisa', 'jd', 'korban', 'kaya', 'begitu']	['areajulid', 'kaya', 'gini', 'bikin', 'berpikir', 'childfre', 'biar', 'awet', 'muda', 'situasi', 'kondisi', 'diluaran', 'mengkhawatirkan', 'anak', 'ga', 'salah', 'korban', 'kaya']

d. *Stemming*

Stemming merupakan penghilangan kata imbuhan dalam sebuah kata kedalam bentuk dasar. Berikut pada Tabel 3.4 merupakan hasil yang didapatkan.

Tabel 3.4 *Stemming Data Train*

	stopwords	***stemming***
1	['sih', 'suka', 'childfre', 'licioust', 'anak', 'dunia', 'cuman', 'sebentar', 'sekejap', 'sekedipan', 'mata', 'tempo', 'singkat', 'anaknya', 'kasihan', 'banget', 'capek', 'repot', 'pusing']	sih suka childfre licioust anak dunia cuman sebentar kejap kedip mata tempo singkat anak kasihan banget capek repot pusing
2	['niat', 'childfre', 'deh', 'maksimal', 'buanyak']	niat childfre deh maksimal buanyak
3	['areajulid', 'kaya', 'gini', 'bikin', 'berpikir', 'childfre', 'biar', 'awet', 'muda', 'situasi', 'kondisi', 'diluaran', 'mengkhawatirkan', 'anak', 'ga', 'salah', 'korban', 'kaya']	areajulid kaya gin bikin pikir childfre biar awet muda situasi kondisi luar khawatir anak ga salah korban kaya

e. *PoS Tagging*

Part of Speech Tagging atau *PoS Tagging* dilakukan untuk mengelompokkan kelas kata untuk setiap kata dalam sebuah kalimat. *PoS tagging* terdiri dari kata benda, kata kerja, kata sifat, keterangan dan lain-lain. Fungsi *PoS Tagging* tersebut untuk menghapus perbedaan yang tidak relevan, menghapus ambiguitas, dan membantu pencarian kata benda. Pada penelitian ini akan dilakukan proses dengan menggunakan dan tanpa menggunakan *PoS Tagging*. Apabila tidak menggunakan *PoS Tagging*, maka proses akan langsung ke tahap pelabelan, namun jika menggunakan *PoS Tagging*, maka ke proses *extract* data. Berikut pada Tabel 3.5 merupakan penjabaran dari proses menggunakan PoS Tagging.

Tabel 3.5 Hasil *PoS Tagging*

	stopwords	***POS Tagger***
1	sih suka childfre licioust anak dunia cuman sebentar kejap kedip mata tempo singkat anak kasihan banget capek repot pusing	[('sih', 'RP'), ('suka', 'VB'), ('childfre', 'FW'), ('licioust', 'FW'), ('anak', 'NN'), ('dunia', 'NN'), ('cuman', 'NN'), ('sebentar', 'NN'), ('kejap', 'NN'), ('kedip', 'NN'), ('mata', 'NN'), ('tempo', 'NN'), ('singkat', 'JJ'), ('anak', 'NN'), ('kasihan', 'NN'), ('banget', 'ADV'), ('capek', 'JJ'), ('repot', 'JJ'), ('pusing', 'JJ')]
2	niat childfre deh maksimal buanyak	[('niat', 'NN'), ('childfre', 'FW'), ('deh', 'UH'), ('maksimal', 'JJ'), ('buanyak', 'JJ')]
3	areajulid kaya gin bikin pikir childfre biar awet muda situasi kondisi luar khawatir anak ga salah korban kaya	[('areajulid', 'NN'), ('kaya', 'JJ'), ('gin', 'NN'), ('bikin', 'VB'), ('pikir', 'VB'), ('childfre', 'FW'), ('biar', 'VB'), ('awet', 'JJ'), ('muda', 'JJ'), ('situasi', 'NN'), ('kondisi', 'NN'), ('luar', 'NN'), ('khawatir', 'JJ'), ('anak', 'NN'), ('ga', 'NN'), ('salah', 'JJ'), ('korban', 'NN'), ('kaya', 'JJ')]

f. *Extract* Data

Tahapan *extract data* merupakan tahap penyeleksian kata-kata, kata-kata yang termasuk:

1) Kata kerja yang diikuti oleh keterangan.
2) Kata kerja diikuti kata sifat.
3) Kata benda diikuti kata kerja dan keterangan.
4) Kata benda diikuti kata kerja.
5) Kata kerja diikuti kata benda dan keterangan.

Berikut pada Tabel 3.6 merupakan hasil dari *extract data.*

Tabel 3.6 Hasil *Extract Data*

	POS Tagger	***Extract Data***
1	[('sih', 'RP'), ('suka', 'VB'), ('childfre', 'FW'), ('licioust', 'FW'), ('anak', 'NN'), ('dunia', 'NN'), ('cuman', 'NN'), ('sebentar', 'NN'), ('kejap', 'NN'), ('kedip', 'NN'), ('mata', 'NN'), ('tempo', 'NN'), ('singkat', 'JJ'), ('anak', 'NN'), ('kasihan', 'NN'), ('banget', 'ADV'), ('capek', 'JJ'), ('repot', 'JJ'), ('pusing', 'JJ')]	childfre licioust anak dunia cuman sebentar kejap kedip mata tempo anak kasihan banget
2	[('niat', 'NN'), ('childfre', 'FW'), ('deh', 'UH'), ('maksimal', 'JJ'), ('buanyak', 'JJ')]	niat childfre
3	[('areajulid', 'NN'), ('kaya', 'JJ'), ('gin', 'NN'), ('bikin', 'VB'), ('pikir', 'VB'), ('childfre', 'FW'), ('biar', 'VB'), ('awet', 'JJ'), ('muda', 'JJ'), ('situasi', 'NN'), ('kondisi', 'NN'), ('luar', 'NN'), ('khawatir', 'JJ'), ('anak', 'NN'), ('ga', 'NN'), ('salah', 'JJ'), ('korban', 'NN'), ('kaya', 'JJ')]	areajulid gin childfre situasi kondisi luar anak ga korban

Berikut pada Tabel 3.7 merupakan hasil *extract data* lengkap dari *sample dataset* yang digunakan dalam penelitian ini.

Tabel 3.7 Hasil *Extract* Data

	Extract Data
1	childfre licioust anak dunia cuman sebentar kejap kedip mata tempo anak kasihan banget
2	niat childfre
3	areajulid gin childfre situasi kondisi luar anak ga korban

	Extract Data
4	bule childfre
5	worksfes childfre ga

3.3.3 Pelabelan

Pada tahap ini dilakukan pelabelan data untuk memisahkan sentimen. Dalam melakukan pelabelan data menggunakan *Corpus*. *Corpus* merupakan kumpulan data dalam bentuk teks yang digunakan untuk mencocokan data dalam menentukan kelas labelnya (Hidayah, et al., 2021). Berikut merupakan penjabaran terkait proses pelabelan dari hasil menggunakan *PoS Tagging* dan tanpa menggunakan *PoS Tagging* menggunakan persamaan (2.1).

a. Menggunakan PoS Tagging

Dari persamaan rumus (2.1) maka didapatkan hasil sebagai berikut.

$$s_{positif}(Data\ ke-2) = \sum 1\ (niat) + 1\ (childfre) = 2$$

$$s_{negatif}(Data\ ke-2) = \sum -1\ (niat) + (-1)(childfre) = -2$$

$$P_{total}(Data\ ke-2) = \sum s_{positif} + s_{negatif} = 2 + (-2) = 0$$

Sepertinya dijelaskan diatas, bahwasana jika hasil polaritas yang didapatkan adalah 0, maka masuk kedalam label netral (0). Hasil *labelling* menggunakan PoS *tagging* dapat dilihat pada Tabel 3.8.

Tabel 3.8 Hasil *Labeling* Menggunakan PoS Tagging

	***Extract* Data**	**Label**
1	childfre licioust anak dunia cuman sebentar kejap kedip mata tempo anak kasihan banget	-1
2	niat childfre	0
3	areajulid gin childfre situasi kondisi luar anak ga korban	1
4	bule childfre	-1
5	worksfes childfre ga	0
6	childfre gak ngerasain anak matin kran aer kamar mandi mager	1

b. Tanpa Menggunakan PoS Tagging

Dari persamaan rumus (2.1) maka didapatkan hasil sebagai berikut.

$$s_{positif}(Data\ ke-2) = \sum 1\ (niat) + 1\ (childfre) + 0(deh) + 1\ (maksimal) + 1\ (banyak) = 4$$

$$s_{negatif}(Data\ ke-2) = \sum -1\ (niat) + (-1)(childfre) + 0(deh) + (-1)(maksimal) + (-1)(banyak) = -4$$

$$P_{total}(Data\ ke-2) = \sum s_{positif} + s_{negatif} = 4 + (-4) = 0$$

Sepertinya dijelaskan diatas, bahwasana jika hasil polaritas yang didapatkan adalah 0, maka masuk kedalam label netral (0).

Tabel 3.9 Hasil Labeling Tanpa Menggunakan PoS Tagging

	stemming	**Label**
1	sih suka childfre licioust anak dunia cuman sebentar kejap kedip mata tempo singkat anak kasihan banget capek repot pusing	-1
2	niat childfre deh maksimal buanyak	0
3	areajulid kaya gin bikin pikir childfre biar awet muda situasi kondisi luar khawatir anak ga salah korban kaya	1
4	cari bule mah childfre	-1
5	worksfes dibilangin childfre ga	0
6	childfre gak ngerasain nyuruh anak matin kran aer kamar mandi pas mager	1

3.3.4 *Split* Data

Pada tahap ini dilakukan pembagian data yang dibagi menjadi dua yaitu data *training* dan data *testing. Data training* digunakan untuk melatih model dalam analisis sentimen, sedangkan *data testing* digunakan untuk menguji model serta mengetahui performa model. Jumlah sampel data yang digunakan ada 6 dengan masing-masing kelas mewakil 2 data, dan dataset tersebut akan dibagi menjadi 2 bagian yaitu train dan test. Berikut merupakan penjabaran terkait proses split data dari hasil menggunakan PoS Tagging dan tanpa menggunakan PoS Tagging.

a. Menggunakan PoS Tagging

Berikut pada Tabel 3.10 merupakan *data train,* dan Tabel 3.11 merupakan *data test.*

Tabel 3.10 Data Train

username	***tweet***
HanieDimasWisht	`~`kalau aku sendiri sih suka *childfree* licioust karena buat apa punya anak di dunia yang cuman hanya sebentar sementara sekejap saja sekedipan mata dan dalam tempo yang sangat singkat ini nanti anaknya malah tambah kasihan banget kita juga semakin capek repot pusing semuanyaΦ`~`
haypedia	niat makin yakin buat *childfree* atau satu deh maksimal, setelah buanyaaaak hal
tukangcepuw	@AREAJULID Yg kaya gini nih yg bikin berpikir utk *childfree*.. bkn krn biar awet muda, tp krn situasi dan kondisi diluaran semakin mengkhawatirkan.. anak ga salah aja bisa jd korban kaya begitu..

Tabel 3.11 Data Test

username	***tweet***
lastcookieinjcr	Cari bule mah buat apa kalo *childfree*
Incelesque	@worksfess Makanya jg dibilangin *childfree* pada ga mau
lha_iki_lho	*Childfree*? Kalian gak bakal ngerasain nyuruh anak matiin kran aer kamar mandi pas kita lagi mager

b. Tanpa Menggunakan PoS Tagging

Berikut pada Tabel 3.12 merupakan *data train,* dan Tabel 3.13 merupakan *data test.*

Tabel 3.12 Data *Train*

username	***tweet***
HanieDimasWisht	`~`kalau aku sendiri sih suka *childfree* licioust karena buat apa punya anak di dunia yang cuman hanya sebentar sementara sekejap saja sekedipan mata dan dalam tempo yang sangat singkat ini nanti anaknya malah tambah kasihan banget kita juga semakin capek repot pusing semuanyaΦ`~`
haypedia	niat makin yakin buat *childfree* atau satu deh maksimal, setelah buanyaaaak hal
tukangcepuw	@AREAJULID Yg kaya gini nih yg bikin berpikir utk *childfree*.. bkn krn biar awet muda, tp krn situasi dan kondisi diluaran semakin mengkhawatirkan.. anak ga salah aja bisa jd korban kaya begitu..

Tabel 3.13 Data *Test*

username	***tweet***
lastcookieinjcr	Cari bule mah buat apa kalo *childfree*
Incelesque	@worksfess Makanya jg dibilangin *childfree* pada ga mau
lha_iki_lho	*Childfree*? Kalian gak bakal ngerasain nyuruh anak matiin kran aer kamar mandi pas kita lagi mager

3.3.5 Pembobotan Delta TF-IDF

Pada tahap ini dilakukan ketika tanpa menggunakan *Pos Tagging,* pembobotan dilakukan untuk merubah data yang bersifat tekstual menjadi numerik untuk memudahkan perhitungan pada tahap berikutnya menggunakan TF-IDF. TF-

IDF adalah cara yang dipakai mencari sebanyak mana hubungan dari kata (*term*) ke dokumen yang akan diberikan bobot. Dengan menyatukan kedua konsep frekuensi adanya hubungan antar kata dan *inverse* frekuensi yang ada di dalam dokumen tersebut. Perhitungan bobot ini memerlukan dua hal yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Namun TF-IDF memiliki kelemahan, sehingga terdapat sebuah metode penyempurna yang dinamakan Delta TF-IDF. Berikut merupakan penjabaran dari proses pembobotan TF-IDF menggunakan PoS Tagging dan tanpa menggunakan PoS Tagging.

a. Menggunakan PoS *Tagging*

Untuk menghitung melakukan perhitungan dengan menggunakan Delta TF-IDF dapat dilakukan dengan menggunakan formula 2.2.

$$tf - idf_{anak,D1} = 2 * \log_2\left(\frac{1 * 1 + 0.5}{1 * 1 + 0.5}\right) = 2 * 0 = 0$$

Pada perhitungan diatas N_p = 1 dimana total dokumen kelas positif adalah 1 dan hal tersebut juga berlaku untuk N_n. Selanjutnya Nilai A didapatkan dari total kemunculan kata anak dalam D1 (yang berlaku sebagai kelas positif) dengan kemunculan lebih dari 0 (dianggap 1) dan Nilai C didapatkan total kemunculan kata anak dalam D3 (yang berlaku sebagai kelas negative) dengan kemunculan lebih dari 0 (dianggap 1).

Tabel 3.14 Delta TFIDF

Corpus	**TF**			**A**	**C**	**IDF**	**TF-IDF**		
	D1	**D2**	**D3**				**D1**	**D2**	**D3**
anak	2	0	1	1	1	0	0	0	0
areajulid	0	0	1	0	1	1.585	0	0	1.585
awet	0	0	1	0	1	1.585	0	0	1.585
banget	1	0	0	1	0	0	0	0	0
biar	0	0	1	0	1	1.585	0	0	1.585
bikin	0	0	1	0	1	1.585	0	0	1.585
buanyak	0	1	0	0	0	0	0	0	0
capek	1	0	0	1	0	0	0	0	0
childfre	1	1	1	1	1	0	0	0	0
cuman	1	0	0	1	0	0	0	0	0

Corpus	TF			A	C	IDF	TF-IDF		
	D1	D2	D3				D1	D2	D3
deh	0	1	0	0	0	0	0	0	0
dunia	1	0	0	1	0	0	0	0	0
ga	0	0	1	0	1	1.585	0	0	1.585
gin	0	0	1	0	1	1.585	0	0	1.585
kasihan	1	0	0	1	0	0	0	0	0
kaya	0	0	2	0	1	1.585	0	0	3.17
kedip	1	0	0	1	0	0	0	0	0
kejap	1	0	0	1	0	0	0	0	0
khawatir	0	0	1	0	1	1.585	0	0	1.585
kondisi	0	0	1	0	1	1.585	0	0	1.585
korban	0	0	1	0	1	1.585	0	0	1.585
licioust	1	0	0	1	0	0	0	0	0
luar	0	0	1	0	1	1.585	0	0	1.585
maksimal	0	1	0	0	0	0	0	0	0
mata	1	0	0	1	0	0	0	0	0
muda	0	0	1	0	1	1.585	0	0	1.585
niat	0	1	0	0	0	0	0	0	0
pikir	0	0	1	0	1	1.585	0	0	1.585
pusing	1	0	0	1	0	0	0	0	0
repot	1	0	0	1	0	0	0	0	0
salah	0	0	1	0	1	1.585	0	0	1.585
sebentar	1	0	0	1	0	0	0	0	0
sih	1	0	0	1	0	0	0	0	0
singkat	1	0	0	1	0	0	0	0	0
situasi	0	0	1	0	1	1.585	0	0	1.585
suka	1	0	0	1	0	0	0	0	0
tempo	1	0	0	1	0	0	0	0	0

b. Tanpa Menggunakan PoS Tagging

Untuk menghitung melakukan perhitungan dengan menggunakan Delta TF-IDF dapat dilakukan dengan menggunakan formula 2.2.

$$tf - idf_{anak,D1} = 2 * \log_2\left(\frac{1 * 1 + 0.5}{1 * 1 + 0.5}\right) = 2 * 0 = 0$$

Pada perhitungan diatas N_p = 1 dimana total dokumen kelas positif adalah 1 dan hal tersebut juga berlaku untuk N_n. Selanjutnya Nilai A didapatkan dari total kemunculan kata anak dalam D1 (yang berlaku sebagai kelas positif) dengan kemunculan lebih dari 0 (dianggap 1) dan Nilai C didapatkan total kemunculan kata anak dalam D3 (yang berlaku sebagai kelas negative) dengan kemunculan lebih dari 0 (dianggap 1).

Tabel 3.15 *Delta TFIDF*

Corpus	TF			A	C	IDF	TF-IDF		
	D1	D2	D3				D1	D2	D3
anak	2	0	1	1	1	0	0	0	0
areajulid	0	0	1	0	1	1.585	0	0	1.585
awet	0	0	1	0	1	1.585	0	0	1.585
banget	1	0	0	1	0	0	0	0	0
biar	0	0	1	0	1	1.585	0	0	1.585
bikin	0	0	1	0	1	1.585	0	0	1.585
buanyak	0	1	0	0	0	0	0	0	0
capek	1	0	0	1	0	0	0	0	0
childfre	1	1	1	1	1	0	0	0	0
cuman	1	0	0	1	0	0	0	0	0
deh	0	1	0	0	0	0	0	0	0
dunia	1	0	0	1	0	0	0	0	0
ga	0	0	1	0	1	1.585	0	0	1.585
gin	0	0	1	0	1	1.585	0	0	1.585
kasihan	1	0	0	1	0	0	0	0	0
kaya	0	0	2	0	1	1.585	0	0	3.17
kedip	1	0	0	1	0	0	0	0	0
kejap	1	0	0	1	0	0	0	0	0

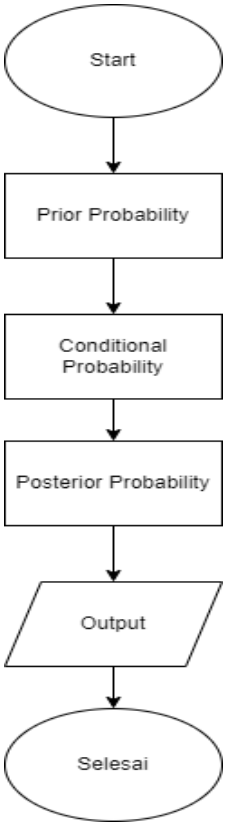
Corpus	TF			A	C	IDF	TF-IDF		
	D1	D2	D3				D1	D2	D3
khawatir	0	0	1	0	1	1.585	0	0	1.585
kondisi	0	0	1	0	1	1.585	0	0	1.585
korban	0	0	1	0	1	1.585	0	0	1.585
licioust	1	0	0	1	0	0	0	0	0
luar	0	0	1	0	1	1.585	0	0	1.585
maksimal	0	1	0	0	0	0	0	0	0
mata	1	0	0	1	0	0	0	0	0
muda	0	0	1	0	1	1.585	0	0	1.585
niat	0	1	0	0	0	0	0	0	0
pikir	0	0	1	0	1	1.585	0	0	1.585
pusing	1	0	0	1	0	0	0	0	0
repot	1	0	0	1	0	0	0	0	0
salah	0	0	1	0	1	1.585	0	0	1.585
sebentar	1	0	0	1	0	0	0	0	0
sih	1	0	0	1	0	0	0	0	0
singkat	1	0	0	1	0	0	0	0	0
situasi	0	0	1	0	1	1.585	0	0	1.585
suka	1	0	0	1	0	0	0	0	0
tempo	1	0	0	1	0	0	0	0	0

3.3.6 Klasifikasi dengan *Random Forest*, dan *Naïve Bayes*

Berikut merupakan hasil penjabaran proses klasifikasi *random forest* dan *naïve bayes*.

3.3.6.1 Klasifikasi *Naïve Bayes*

Berikut pada Gambar 3.2 merupakan tahapan yang digunakan klasifikasi *naïve bayes*.



Gambar 3.2 Tahapan *Naive Bayes*

a. Menggunakan *PoS Tagging*

1) *Prior Probability*

Untuk menghitung *prior probability* dapat menggunakan persamaan (2.8). Sehingga didapatkan probabilitas untuk setiap kelas adalah sebagai berikut.

$$P(0) = \frac{1}{3} = 0.33$$

$$P(-1) = \frac{1}{3} = 0.33$$

$$P(1) = \frac{1}{3} = 0.33$$

2) *Conditional Probability*

Untuk menghitung *conditional probability* dapat menggunakan persamaan (2.9). Sebelum melakukan perhitungan di atas perlu

mengetahui total TF-IDF setiap kelas dan Total IDF. Hasil total IDF dengan hasil 61.54783339 dan untuk jumlah bobot TF-IDF setiap kelas dapat dilihat pada Tabel 3.16.

Tabel 3.16 Jumlah Bobot TF-IDF Setiap Kelas

Jumlah Total Bobot Pada Kelas 0	2.201329
Jumlah Total Bobot Pada Kelas -1	4.192513
Jumlah Total Bobot Pada Kelas 1	3.98827

Setelah mengetahui nilai yang dibutuhkan, didapatkan hasil sebagai berikut.

$$P(Childfree|kelas\ 0) = \frac{1\ (Nilai\ TF - IDF\ Childfree) + 1}{2.201329 + \ 61.5478}$$
$$= 0.031372961$$
$$P(Childfree|kelas - 1) = \frac{1\ (Nilai\ TF - IDF\ Childfree) + 1}{4.192513 + \ 61.5478}$$
$$= 0.030422718$$
$$P(Childfree|kelas\ 1) = \frac{1\ (Nilai\ TF - IDF\ Childfree) + 1}{3.98827 + \ 61.5478}$$
$$= 0.03051753$$

Nilai 1 (TF-IDF *Childfree*) merupakan nilai TF-IDF pada data testing yang dapat dilihat pada Tabel 3.16 di dokumen 1. Nilai 2.201, 4.1925, dan 3.988 merupakan nilai total tf-idf setiap kelas yang dihitung berdasarkan pada hasil total IDF yaitu 61.54783339. Untuk nilai 61.5478 merupakan total nilai dari IDF. Hasil data test 1, 2 dan 3 dapat dilihat pada Tabel 3.17 hingga 3.19.

Tabel 3.17 Data test 1

Kelas	**Probabilitas**
Kelas 0	0.031372961
Kelas -1	0.030422718
Kelas 1	0.03051753

Tabel 3.18 Data test 2

Kelas	*childfree*	ga
Kelas 0	0.02366372	0.029193121
Kelas -1	0.022946979	0.028308902
Kelas 1	0.023018493	0.028397126

Tabel 3.19 Data test 3

Kelas	*childfree*	anak
Kelas 0	0.028076	0.02530787
Kelas -1	0.027225	0.02454133
Kelas 1	0.02731	0.024617813

Pada Tabel 3.17 dan Tabel 3.19, memiliki 2 kata yang memiliki bobot berdasarkan pada data train, artinya terdapat 2 kata yang akan dilakukan untuk menghitung probabilitas pada data test ke-2 dan ke-3 (lihat Tabel 3.15). Untuk mendapatkan nilai pada data test 2 dan 3 dihitung berdasarkan probabilitas yang sama pada data test 1 yang hanya memiliki 1 kata saja yang memiliki bobot

3) ***Posterior Probability***

Untuk menghitung *posterior probability* dapat menggunakan persamaan rumus (2.10). Sehingga didapatkan hasil sebagai berikut.

$$P(Data\ test\ 1|kelas\ 0) = 0.33 * 0.031372961 = 0.010457654$$

$$P(Data\ test\ 1|kelas\ 1) = 0.33 * 0.030422718 = 0.010140906$$

$$P(Data\ test\ 1|kelas\ 2) = 0.33 * 0.03051753 = 0.01017251$$

Tabel 3.20 Hasil Posteriror Probabilitas Data Test 1

Probabilitas Kelas		Prediksi Kelas
Kelas 0	0.010457654	Kelas 0
Kelas -1	0.010140906	
Kelas 1	0.01017251	

Tabel 3.21 Hasil Posterior Probilitas Data Test 2

Probabilitas Kelas		Prediksi Kelas
Kelas 0	0.000230273	Kelas 0
Kelas -1	0.000216535	
Kelas 1	0.000217886	

Tabel 3.22 Hasil Posterior Probabilitas Data Test 3

Probabilitas Kelas		Prediksi Kelas
Kelas 0	0.000237	Kelas 0
Kelas -1	0.000223	
Kelas 1	0.000224	

4) *Output*

Hasil prediksi pada *naïve bayes* akan tergantung dari nilai tertinggi *posterior probability* pada suatu kelas. Jika perhatikan, D1 nilai tertinggi nya adalah 0.104 yang dimana nilai tersebut pada kelas 0. Sehingga bisa disimpulkan bahwasanya D1 diprediksi 0. Hasil lengkap prediksi dengan *naïve bayes* dapat dilihat pada Tabel 3.23.

Tabel 3.23 Hasil Prediksi *Naive Bayes*

Dokumen	Prediksi	Aktual
D1	0	-1
D2	0	0
D3	0	1

b. Tanpa Menggunakan *PoS Tagging*

1) *Prior Probability*

Untuk menghitung *prior probability* dapat menggunakan persamaan (2.8). Sehingga didapatkan probabilitas untuk setiap kelas adalah sebagai berikut.

$$P(0) = \frac{1}{3} = 0.33$$

$$P(-1) = \frac{1}{3} = 0.33$$

$$P(1) = \frac{1}{3} = 0.33$$

2) *Conditional Probability*

Untuk menghitung *conditional probability* dapat menggunakan persamaan (2.9). Sebelum melakukan perhitungan di atas perlu mengetahui total TF-IDF setiap kelas dan Total IDF. Hasil total IDF dengan hasil 61.54783339 dan untuk jumlah bobot TF-IDF setiap kelas dapat dilihat pada Tabel 3.24.

Tabel 3.24 Jumlah Bobot TF-IDF Setiap Kelas

Jumlah Total Bobot Pada Kelas 0	2.201329
Jumlah Total Bobot Pada Kelas -1	4.192513
Jumlah Total Bobot Pada Kelas 1	3.98827

Setelah mengetahui nilai yang dibutuhkan, didapatkan hasil sebagai berikut.

$$P(Childfree|kelas\ 0) = \frac{1\ (Nilai\ TF - IDF\ Childfree) + 1}{2.201329 + \ 61.5478}$$

$$= 0.031372961$$

$$P(Childfree|kelas - 1) = \frac{1\ (Nilai\ TF - IDF\ Childfree) + 1}{4.192513 + \ 61.5478}$$

$$= 0.030422718$$

$$P(Childfree|kelas\ 1) = \frac{1\ (Nilai\ TF - IDF\ Childfree) + 1}{3.98827 + \ 61.5478}$$

$$= 0.03051753$$

Nilai 1 (TF-IDF *Childfree*) merupakan nilai TF-IDF pada data testing yang dapat dilihat pada Tabel 3.26 di dokumen 1. Nilai 2.201, 4.1925, dan 3.988 merupakan nilai total tf-idf setiap kelas yang dihitung berdasarkan pada Tabel 3.24. Untuk nilai 61.5478 merupakan total nilai dari IDF.

Tabel 3.25 Data Test 1

Kelas	Probabilitas
Kelas 0	0.031372961
Kelas -1	0.030422718
Kelas 1	0.03051753

Tabel 3.26 Data Test 2

Kelas	*childfree*	ga
Kelas 0	0.02366372	0.029193121
Kelas -1	0.022946979	0.028308902
Kelas 1	0.023018493	0.028397126

Tabel 3.27 Data Test 3

Kelas	*childfree*	anak
Kelas 0	0.028076	0.02530787
Kelas -1	0.027225	0.02454133
Kelas 1	0.02731	0.024617813

Pada Tabel 3.26 dan Tabel 3.27, memiliki 2 kata yang memiliki bobot berdasarkan pada data train, artinya terdapat 2 kata yang akan dilakukan untuk menghitung probabilitas pada data test ke-2 dan ke-3 (lihat Tabel 3.18). Untuk mendapatkan nilai pada data test 2 dan 3 dihitung berdasarkan probabilitas yang sama pada data test 1 yang hanya memiliki 1 kata saja yang memiliki bobot.

3) ***Posterior Probability***

Untuk menghitung *posterior probability* dapat menggunakan persamaan rumus (2.10). Sehingga didapatkan hasil sebagai berikut.

$$P(Data\ test\ 1|kelas\ 0) = 0.33 * 0.031372961 = 0.010457654$$

$$P(Data\ test\ 1|kelas\ 1) = 0.33 * 0.030422718 = 0.010140906$$

$$P(Data\ test\ 1|kelas\ 2) = 0.33 * 0.03051753 = 0.01017251$$

Tabel 3.28 Hasil Posterior Probabilitas Data Test 1

Probabilitas Kelas		Prediksi Kelas
Kelas 0	0.010457654	Kelas 0
Kelas -1	0.010140906	
Kelas 1	0.01017251	

Tabel 3.29 Hasil Posterior Probilitas Data Test 2

Probabilitas Kelas		Prediksi Kelas
Kelas 0	0.000230273	Kelas 0
Kelas -1	0.000216535	
Kelas 1	0.000217886	

Tabel 3.30 Hasil Posterior Probabilitas Data Test 3

Probabilitas Kelas		Prediksi Kelas
Kelas 0	0.000237	Kelas 0
Kelas -1	0.000223	
Kelas 1	0.000224	

4) *Output*

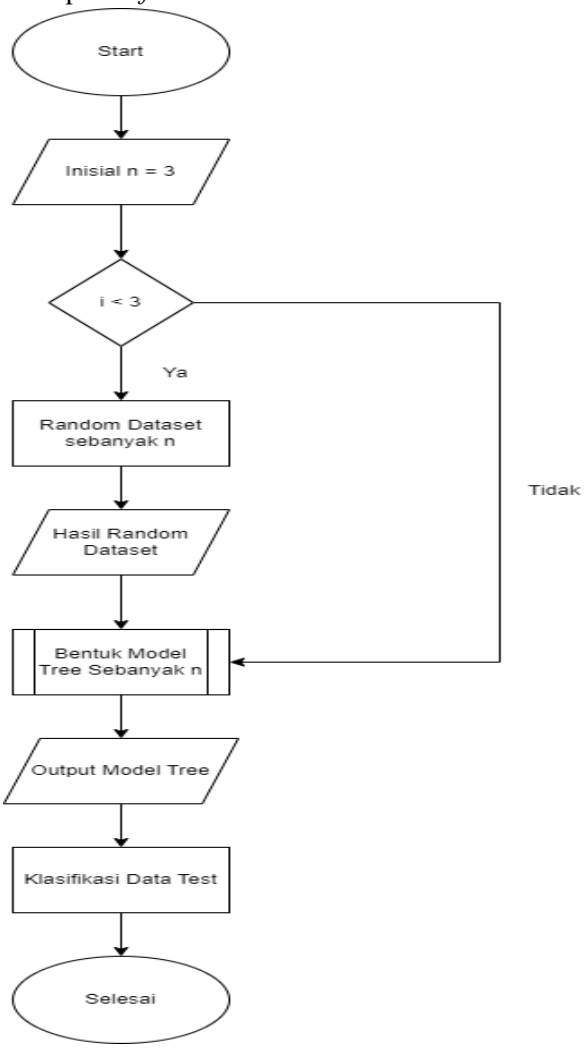
Hasil prediksi pada *naïve bayes* akan tergantung dari nilai tertinggi posterior probability pada suatu kelas. Jika perhatikan, D1 nilai tertinggi nya adalah 0.104 yang dimana nilai tersebut pada kelas 0. Sehingga bisa disimpulkan bahwasanya D1 diprediksi 0. Hasil lengkap prediksi dengan *naïve bayes* dapat dilihat pada Tabel 3.31.

Tabel 3.31 Hasil Prediksi *Naive Bayes*

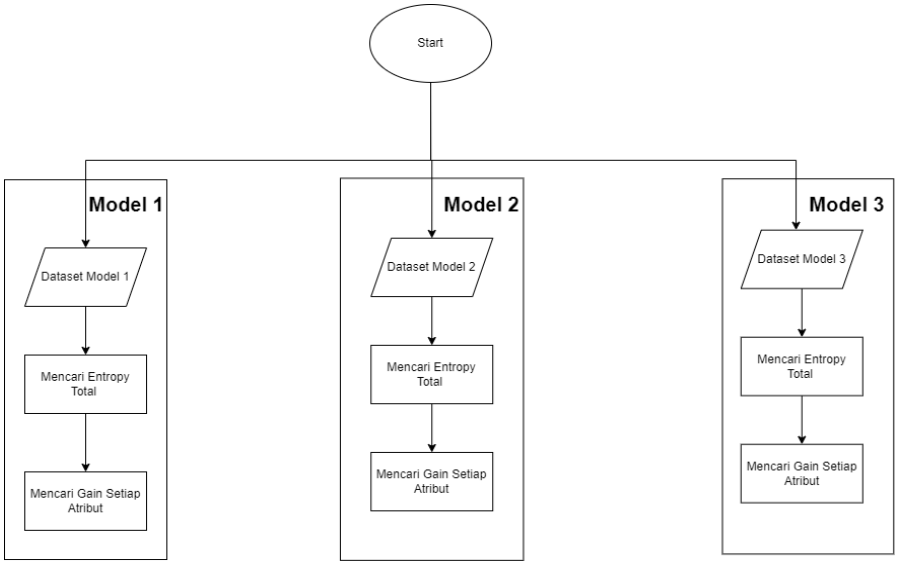
Dokumen	Prediksi	Aktual
D1	0	-1
D2	0	0
D3	0	1

3.3.6.2 Klasifikasi *Random Forest*

Berikut pada Gambar 3.4 merupakan *flowchart* dari random forest dan Gambar 3.5 merupakan *flowchart tree*.



Gambar 3.3 *Flowchart Random Forest*



Gambar 3.4 *Flowchart Tree*

a. Menggunakan PoS Tagging

1) Inisial N

Dari Gambar 3.3 atau 3.4, dapat dijelaskan bahwa *random forest* merupakan sekumpulan *tree*. Jika *random forest* akan membentuk 3 model *tree* dikarenakan N = 3. *(K)* dimana untuk membentuk model tree akan menggunakan formula *decision tree*, maka *random forest* akan melakukan pengacakan (*random*) dataset untuk 3 model dimana setiap model memiliki dataset nya tersendiri. Sehingga *flowchart* yang terbentuk dapat dilihat pada Gambar 3.4.

2) *Random* Dataset Sebanyak N

Pengambilan akan dilakukan secara *random* dengan kemungkinan terdapat kemunculan data yang pada hasil tiap bootstrap dan bahkan tidak sama sekali. Pengambilan disini akan menggunakan index data untuk memudahakan pembacaan alur.

Tabel 3.32 Hasil *Random Dataset*

No	Dataset Random 1	Dataset Random 2	Dataset Random 3
1	D1	D1	D1
2	D2	D2	D3
3	D3	-	-

Tabel 3.33 Dataset Random 1

	anak	areajulid	…	**Label**
D1	0.352095029	0		-1
D2	0	0		0
D3	0.174811679	0.22985635		1

Tabel 3.34 *Dataset* Random 2

	anak	areajulid	…	**Label**
1	0.352095029	0		-1
2	0	0		0

Tabel 3.35 Dataset Random 3

	anak	areajulid	…	**Label**
1	0.352095029	0		-1
3	0.174811679	0.22985635		1

Pada Tabel 3.33, Tabel 3.34, Tabel 3.35 merupakan hasil Tabel random dataset dimana data tersebut dilakukan pada data hasil preprocessing di TF-IDF. Setelah itu, sebelum melakukan klasifikasi, dataset pada feature independent yang bernilai numerik kontinus akan dirubah menjadi interval untuk mendapatkan kategorinya. Dikarenakan data telah dilakukan TF-IDF dan data berada pada rentang paling besar adalah 0.5, maka interval akan berada pada < 0.2 dan >= 0.2. Perhitungan Tree akan dilakukan transformasi untuk bertujuan memudahkan pembacaan hasil.

3) Mencari *Entropy* Total

Untuk menghitung *entropy* perlu mengetahui probabilitas label terhadap total kasus. Pada dataset ini memiliki 2 label yaitu 1 dan -1. Untuk total kemunculan kedua label pada total kasus dapat dilihat pada Tabel 3.36.

Tabel 3.36 Proporsi kemunculan kelas pada setiap bootstrap

Model Ke-	**0**	**-1**	**1**
1	1	1	1
2	1	1	-
3	-	1	1

Setelah mengetahui kemunculan setiap label pada datasetnya, maka dapat dihitung Entropy setiap model menggunakan persamaan (2.5).

$$Entropy(Model\ 1) = \sum\left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) = 1.584962501$$

$$Entropy(Model\ 2) = \sum\left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) = 1.056641667$$

$$Entropy(Model\ 3) = \sum\left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) = 1.056641667$$

Tabel 3.37 Entropy total setiap model

Model Ke-	***Entropy***
1	1.584962501
2	1.056641667
3	1.056641667

4) Mencari Gain Setiap Atribut

Berikut merupakan contoh perhitungan model ke-1. Untuk mencari gain, diperlukan mencari entropy setiap fitur terhadap setiap kriteria pada fitur tersebut dengan menggunakan persamaan (2.6). Contoh pada fitur pada **anak** terdapat 2 kriteria < 0.2 dan >= 0.2 maka perlu diketahui total kemunculan setiap kriteria terhadap terhadap label. Perhitungan *entropy* dan *gain* pada *feature* anak pada model 1 dapat dilihat pada Tabel 3.38.

Tabel 3.38 Perhitungan Entropy dan Gain pada Feature anak Pada Model 1

Atribut	Partisi	Total Data	0	-1	1	Entropy	Gain
Total		**3**	**1**	**1**	**1**	**1.584962501**	
anak	< 0.2	2	1	0	1	1	0.918295834
	>= 0.2	1	0	1	0	0	

Pada nilai 2 pada fitur total data artinya bahwasanya kemunculan nilai < 0.2 pada fitur **anak** dan nilai >= 0.2 adalah 1. Pada label 0 kemunculan kriteria < 0.2 pada fitur **anak** adalah 1 sedangkan pada label 1 adalah 0 dan pada label 2 adalah 1, selanjutnya pada kriteria >=0.2 pada label 0 dan 2 kemunculannya adalah 1 dan pada label 1 kemunculannya adalah 1. Setelah mengetahui kemunculan tiap kriteria terhadap label, maka dapat dihitung entropy per kriteria pada fitur **anak.**

$$Entropy(Model\ 1, anak, < 0.2) = \sum\left(-\frac{1}{2} * log_2\left(\frac{1}{2}\right)\right) + \left(-\frac{0}{2} * log_2\left(\frac{0}{2}\right)\right) + \left(-\frac{1}{2} * log_2\left(\frac{1}{2}\right)\right) = 1$$

$$Entropy(Model\ 1, anak, \geq 0.2) = \sum\left(-\frac{0}{1} * log_2\left(\frac{0}{1}\right)\right) + \left(-\frac{1}{1} * log_2\left(\frac{1}{1}\right)\right) + \left(-\frac{0}{1} * log_2\left(\frac{0}{1}\right)\right) = 0$$

Setelah mendapatkan nilai *entropy* setiap kriteria pada setiap atribut, maka selanjutnya dapat menghitung information gain fitur terkait dengan formula di atas. Sebelum itu harus menghitung entropy dua atribut, maksud dua atribut tersebut adalah entropy total dari dari kedua kriteria

dengan formula (2.7). Jika diperhatikan formula di atas adalah formula information Gain untuk setiap kriteria pada atribut terkait.

$$Entropy\ 2\ Atribut\ (Model\ 1, anak, Kriteria < 0.2\ dan \geq 0.2) = \sum\left(-\frac{2}{3} * 1\right) + \left(\frac{1}{3} * 0\right) = 0.666666667$$

$$Information\ Gain\ (Model\ 1, A1) = \mathbf{1.584962501} - 0.666666667 = 0.918295834$$

Perhitungan di atas berlaku untuk seluruh model berdasarkan dengan kriteria yang terdapat.

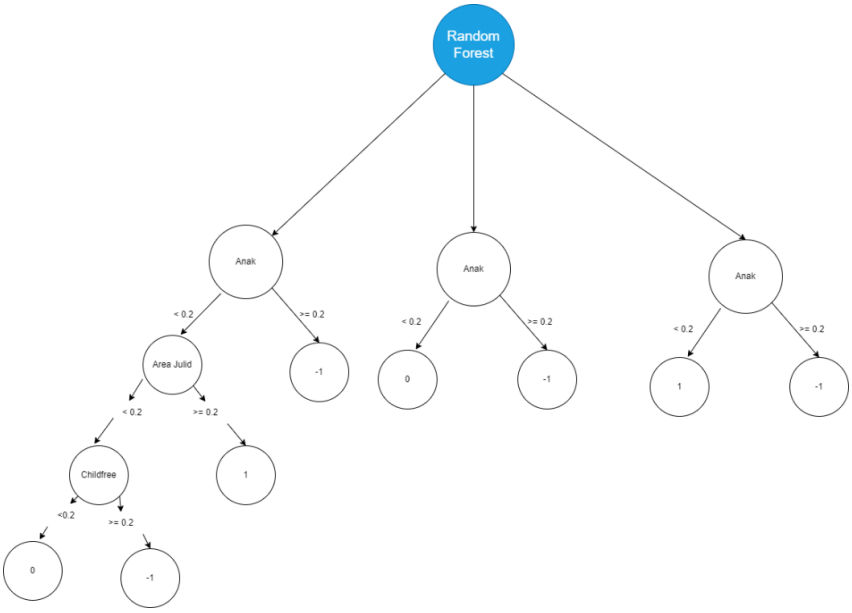
Tabel 3.39 Hasil Gain tertinggi setiap model

Feature	Gain Model Ke-1	Gain Model Ke-2	Gain Model Ke-3
anak	0.918296	1	1
areajulid	0.918296	0.333333333	0
awet	0.918296	0.333333333	0
banget	0.918296	1	1
biar	0.918296	0.333333333	0
bikin	0.918296	0.333333333	0
buanyak	0.918296	1	1
capek	0.918296	1	1
childfre	0.918296	1	1
cuman	0.918296	1	1
deh	0.918296	1	1
dunia	0.918296	1	1
ga	0.918296	0.333333333	0
gin	0.918296	0.333333333	0
kasihan	0.918296	1	1
kaya	0.918296	0.333333333	0
kedip	0.918296	1	1
kejap	0.918296	1	1
khawatir	0.918296	0.333333333	0

Feature	Gain Model Ke-1	Gain Model Ke-2	Gain Model Ke-3
kondisi	0.918296	0.333333333	0
korban	0.918296	0.333333333	0
licioust	0.918296	1	1
luar	0.918296	0.333333333	0
maksimal	0.918296	1	1
mata	0.918296	1	1
muda	0.918296	0.333333333	0
niat	0.918296	1	1
pikir	0.918296	0.333333333	0
pusing	0.918296	1	1
repot	0.918296	1	1
salah	0.918296	0.333333333	0
sih	0.918296	1	1
singkat	0.918296	1	1
situasi	0.918296	1	1
suka	0.918296	0.333333333	0
tempo	0.918296	1	1

5) ***Output***

Dari Information Gain di atas didapatkan model tree yang nantinya akan dilakukan penggabungan menjadi sebuah forest. Jika divisualkan menjadi Gambar 3.5.



Gambar 3.5 Visual *Random Forest*

6) Prediksi Data *Test*

Pada hasil *Random Forest* diatas, didapatkan bahwasanya setiap model tree memiliki 1 akar dan 2 leaf (daun) pada model tree dan 2 dan 3, namun berbeda dengan model tree 1. Hal ini dikarenakan distribusi data yang tidak merata sehingga harus melakukan iterasi hingga menemukan distribusi yang normal. Dengan setiap model tree memiliki feature root dan leaf yang berbeda. Untuk menentukan kelas yang akan diprediksi akan menggunakan sistem **majorit vactory** yang dimana jika 2 model memprediksi 1 dan 1 model memprediksi 0 maka hasil prediksi data testing adalah 1. Hasil prediksi *data testing* dapat dilihat pada Tabel 3.40.

Tabel 3.40 Hasil Prediksi *Data Testing*

Index Data	Model Tree 1	Model Tree 2	Model Tree 3	Prediksi	Label Sebenarnya
D1 Test	-1	0	1	-1	-1
D2 Test	-1	0	1	-1	0
D3 Test	-1	-1	-1	-1	1

b. Tanpa Menggunakan PoS Tagging

1) Inisial N

Dari Gambar 3.5, dapat dijelaskan bahwa *random forest* merupakan sekumpulan *tree*. Jika *random forest* akan membentuk 3 model *tree* dikarenakan N = 3. *(K)* dimana untuk membentuk model tree akan menggunakan formula *decision tree*, maka *random forest* akan melakukan pengacakan (*random*) dataset untuk 3 model dimana setiap model memiliki dataset nya tersendiri. Sehingga *flowchart* yang terbentuk dapat dilihat pada Gambar 3.4.

2) Random Dataset Sebanyak N

Pengambilan akan dilakukan secara *random* dengan kemungkinan terdapat kemunculan data yang pada hasil tiap bootstrap dan bahkan tidak sama sekali. Pengambilan disini akan menggunakan index data untuk memudahakan pembacaan alur.

Tabel 3.41 Hasil *Random Dataset*

No	Dataset Random 1	Dataset Random 2	Dataset Random 3
1	D1	D1	D1
2	D2	D2	D3
3	D3	-	-

Tabel 3.42 Dataset Random 1

	anak	areajulid		**Label**
D1	0.352095029	0	…	-1
D2	0	0		0
D3	0.174811679	0.22985635		1

Tabel 3.43 *Dataset* Random 2

	anak	areajulid		**Label**
1	0.352095029	0	…	-1
2	0	0		0

Tabel 3.44 Dataset Random 3

	anak	areajulid		**Label**
1	0.352095029	0	…	-1
3	0.174811679	0.22985635		1

Pada Tabel 3.42, Tabel 3.43, Tabel 3.44 merupakan hasil Tabel random dataset dimana data tersebut dilakukan pada data hasil preprocessing di TF-IDF. Setelah itu, sebelum melakukan klasifikasi, dataset pada feature independent yang bernilai numerik kontinus akan dirubah menjadi interval untuk mendapatkan kategorinya. Dikarenakan data telah dilakukan TF-IDF dan data berada pada rentang paling besar adalah 0.5, maka interval akan berada pada < 0.2 dan >= 0.2. Perhitungan Tree akan dilakukan transformasi untuk bertujuan memudahkan pembacaan hasil.

3) Mencari Entropy Total

Untuk menghitung *entropy* perlu mengetahui probabilitas label terhadap total kasus. Pada dataset ini memiliki 2 label yaitu 1 dan -1. Untuk total kemunculan kedua label pada total kasus dapat dilihat pada Tabel 3.45.

Tabel 3.45 Proporsi kemunculan kelas pada setiap bootstrap

Model Ke-	**0**	**-1**	**1**
1	1	1	1
2	1	1	-
3	-	1	1

Setelah mengetahui kemunculan setiap label pada datasetnya, maka dapat dihitung Entropy setiap model menggunakan persamaan (2.5).

$$Entropy(Model\ 1) = \sum\left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) = 1.584962501$$

$$Entropy(Model\ 2) = \sum\left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) = 1.056641667$$

$$Entropy(Model\ 3) = \sum\left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) + \left(-\frac{1}{3} * log_2\left(\frac{1}{3}\right)\right) = 1.056641667$$

Tabel 3.46 Entropy total setiap model

Model Ke-	**Entropy**
1	1.584962501
2	1.056641667
3	1.056641667

4) Mencari Gain Setiap Atribut

Berikut merupakan contoh perhitungan model ke-1. Untuk mencari gain, diperlukan mencari entropy setiap fitur terhadap setiap kriteria pada fitur tersebut dengan menggunakan persamaan (2.6). Contoh pada fitur pada **anak** terdapat 2 kriteria < 0.2 dan >= 0.2 maka perlu diketahui total kemunculan setiap kriteria terhadap terhadap label.

Tabel 3.47 Perhitungan Entropy dan Gain pada Feature anak Pada Model 1

Atribut	Partisi	Total Data	0	-1	1	Entropy	Gain
Total		**3**	**1**	**1**	**1**	**1.584962501**	
anak	< 0.2	2	1	0	1	1	0.918295834
	>= 0.2	1	0	1	0	0	

Pada nilai 2 pada fitur total data artinya bahwasanya kemunculan nilai < 0.2 pada fitur **anak** dan nilai >= 0.2 adalah 1. Pada label 0 kemunculan kriteria < 0.2 pada fitur **anak** adalah 1 sedangkan pada label 1 adalah 0 dan pada label 2 adalah 1, selanjutnya pada kriteria >=0.2 pada label 0 dan 2 kemunculannya adalah 1 dan pada label 1 kemunculannya adalah 1. Setelah mengetahui kemunculan tiap kriteria terhadap label, maka dapat dihitung entropy per kriteria pada fitur **anak.**

$$Entropy(Model\ 1, anak, < 0.2) = \sum\left(-\frac{1}{2} * log_2\left(\frac{1}{2}\right)\right) + \left(-\frac{0}{2} * log_2\left(\frac{0}{2}\right)\right) + \left(-\frac{1}{2} * log_2\left(\frac{1}{2}\right)\right) = 1$$

$$Entropy(Model\ 1, anak, \geq 0.2) = \sum\left(-\frac{0}{1} * log_2\left(\frac{0}{1}\right)\right) + \left(-\frac{1}{1} * log_2\left(\frac{1}{1}\right)\right) + \left(-\frac{0}{1} * log_2\left(\frac{0}{1}\right)\right) = 0$$

Setelah mendapatkan nilai entropy setiap kriteria pada setiap atribut, maka selanjutnya dapat menghitung information gain fitur terkait dengan formula di atas. Sebelum itu harus menghitung entropy dua atribut, maksud dua atribut tersebut adalah entropy total dari dari kedua kriteria dengan formula (2.7). Jika diperhatikan formula di atas adalah formula information Gain untuk setiap kriteria pada atribut terkait.

$$Entropy\ 2\ Atribut\ (Model\ 1, anak, Kriteria < 0.2\ dan \geq 0.2) = \sum\left(-\frac{2}{3} * 1\right) + \left(\frac{1}{3} * 0\right) = 0.666666667$$

$$Information\ Gain\ (Model\ 1, A1) = \mathbf{1.584962501} - 0.666666667 = 0.918295834$$

Perhitungan di atas berlaku untuk seluruh model berdasarkan dengan kriteria yang terdapat.

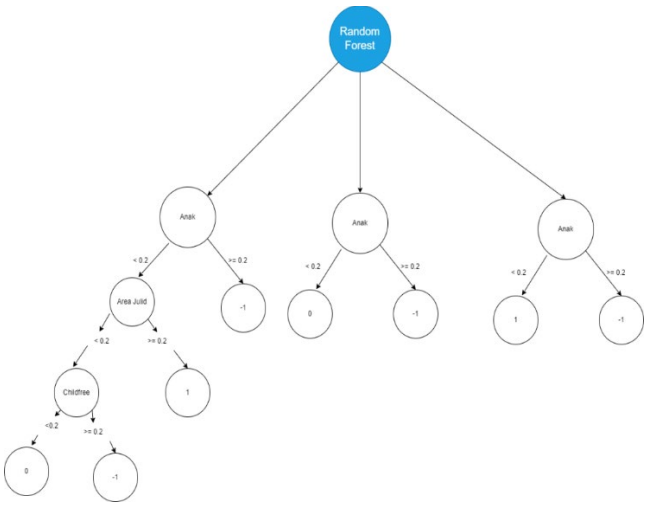
Tabel 3.48 Hasil Gain tertinggi setiap model

Feature	Gain Model Ke-1	Gain Model Ke-2	Gain Model Ke-3
anak	0.918296	1	1
areajulid	0.918296	0.333333333	0
awet	0.918296	0.333333333	0
banget	0.918296	1	1
biar	0.918296	0.333333333	0
bikin	0.918296	0.333333333	0
buanyak	0.918296	1	1
capek	0.918296	1	1
childfre	0.918296	1	1
cuman	0.918296	1	1
deh	0.918296	1	1
dunia	0.918296	1	1
ga	0.918296	0.333333333	0
gin	0.918296	0.333333333	0
kasihan	0.918296	1	1
kaya	0.918296	0.333333333	0
kedip	0.918296	1	1
kejap	0.918296	1	1
khawatir	0.918296	0.333333333	0
kondisi	0.918296	0.333333333	0
korban	0.918296	0.333333333	0
licioust	0.918296	1	1
luar	0.918296	0.333333333	0
maksimal	0.918296	1	1
mata	0.918296	1	1

Feature	Gain Model Ke-1	Gain Model Ke-2	Gain Model Ke-3
muda	0.918296	0.333333333	0
niat	0.918296	1	1
pikir	0.918296	0.333333333	0
pusing	0.918296	1	1
repot	0.918296	1	1
salah	0.918296	0.333333333	0
sih	0.918296	1	1
singkat	0.918296	1	1
situasi	0.918296	1	1
suka	0.918296	0.333333333	0
tempo	0.918296	1	1

5) *Output*

Dari *Information Gain* di atas didapatkan model tree yang nantinya akan dilakukan penggabungan menjadi sebuah forest. Jika divisualkan menjadi Gambar 3.6.



Gambar 3.6 Visual *Random Forest*

6) Prediksi Data Test

Pada hasil Random Forest diatas, didapatkan bahwasanya setiap model tree memiliki 1 akar dan 2 leaf (daun) pada model tree dan 2 dan 3, namun

berbeda dengan model tree 1. Hal ini dikarenakan distribusi data yang tidak merata sehingga harus melakukan iterasi hingga menemukan distribusi yang normal. eDngan setiap model tree memiliki feature root dan leaf yang berbeda. Untuk menentukan kelas yang akan diprediksi akan menggunakan sistem **majorit vactory** yang dimana jika 2 model memprediksi 1 dan 1 model memprediksi 0 maka hasil prediksi data testing adalah 1. Hasil prediksi *data testing* dapat dilihat pada Tabel 3.49.

Tabel 3.49 Hasil prediksi *data testing*

Index Data	**Model Tree 1**	**Model Tree 2**	**Model Tree 3**	**Prediksi**	**Label Sebenarnya**
D1 Test	-1	0	1	-1	-1
D2 Test	-1	0	1	-1	0
D3 Test	-1	-1	-1	-1	1

3.3.7 Evaluasi

Setelah dilakukan klasifikasi menggunakan algoritma *Random Forest* dan *Naïve Bayes* selanjutnya dilakukan proses evaluasi dengan menggunakan metode *confusion matrix* dengan menggunakan persamaan (2.13) hingga (2.16). Berikut merupakan penjabaran mengenai evaluasi tersebut.

a. Menggunakan PoS Tagging

1) *Confusion Matrix Naïve Bayes*

Berikut pada Tabel 3.50 merupakan *confusion matrix naïve bayes*.

Tabel 3.50 *Confusion Matrix Naive Bayes*

		Prediksi		
		1	0	-1
Actual	1	24	9	5
	0	8	74	20
	-1	1	18	100

$$Akurasi = \frac{((24+74+100)+(0+0+0))}{((24+74+100)+(9+27+25)+(14+28+19)+(0+0+0))}\ x\ 100\% =$$

76.45 %

$$Presisi = \frac{\frac{24}{(24+8+1)} \text{ x } 38 \text{ x } 100\% + \frac{74}{(74+9+18)} \text{ x } 102 \text{ x } 100\% + \frac{100}{(100+20+5)} \text{ x } 119 \text{ x } 100\%}{(38+102+119)} = 76.28\ \%$$

$$Recall = \frac{\frac{24}{(24+9+5)} \text{ x } 38 \text{ x } 100\% + \frac{74}{(74+8+20)} \text{ x } 102 \text{ x } 100\% + \frac{100}{(100+1+18)} \text{ x } 119 \text{ x } 100\%}{(38+102+119)} = 76.45\ \%$$

$$F1 - Score = \frac{2 * \frac{\frac{24}{(24+8+1)} x \frac{24}{(24+9+5)}}{\frac{24}{(24+8+1)} + \frac{24}{(24+9+5)}} \text{x } 38 \text{ x } 100\% + 2 * \frac{\frac{74}{(74+9+18)} x \frac{74}{(74+8+20)}}{\frac{74}{(74+9+18)} + \frac{74}{(74+8+20)}} \text{ x } 102 \text{ x } 100\% + 2 * \frac{\frac{100}{(100+20+5)} x \frac{100}{(100+1+18)}}{\frac{100}{(100+20+5)} + \frac{100}{(100+1+18)}} \text{ x } 119 \text{ x } 100\%}{(38+102+119)} =$$

76.29 %

2) *Confusion Matrix Random Forest*

Berikut pada Tabel 3.51 merupakan *confusion matrix random forest*.

Tabel 3.51 *Confusion matrix Random Forest*

		Prediksi		
		1	0	-1
Actual	1	19	17	2
	0	1	93	8
	-1	0	23	96

$$Akurasi = \frac{((19+93+96)+(0+0+0))}{((19+93+96)+(1+40+10)+(19+9+23)+(0+0+0))}\ x\ 100\% =$$

80.31 %

$$Presisi = \frac{\frac{19}{(19+1+0)} \text{ x } 38 \text{ x } 100\% + \frac{93}{(93+17+23)} \text{ x } 102 \text{ x } 100\% + \frac{96}{(96+2+8)} \text{ x } 119 \text{ x } 100\%}{(38+102+119)} = 83.09\ \%$$

$$Recall = \frac{\frac{19}{(19+17+2)} \text{ x } 38 \text{ x } 100\% + \frac{93}{(93+1+8)} \text{ x } 102 \text{ x } 100\% + \frac{96}{(96+0+23)} \text{ x } 119 \text{ x } 100\%}{(38+102+119)} =$$

80.31 %

$$F1 - Score = \frac{2 * \frac{\frac{19}{(19+1+0)} x \frac{19}{(19+17+2)}}{\frac{19}{(19+1+0)} + \frac{19}{(19+17+2)}} \text{x } 38 \text{ x } 100\% + 2 * \frac{\frac{93}{(93+17+23)} x \frac{93}{(93+1+8)}}{\frac{93}{(93+17+23)} + \frac{93}{(93+1+8)}} \text{ x } 102 \text{ x } 100\% + 2 * \frac{\frac{96}{(96+2+8)} x \frac{96}{(96+0+23)}}{\frac{96}{(96+2+8)} + \frac{96}{(96+0+23)}} \text{ x } 119 \text{ x } 100\%}{(38+102+119)} =$$

79.99 %

b. Tanpa Menggunakan PoS Tagging

1) *Confusion Matrix Naïve Bayes*

Berikut pada Tabel 3.52 merupakan *confusion matrix naïve bayes*.

Tabel 3.52 *Confusion Matrix Naive Bayes*

		Prediksi		
		1	0	-1
Actual	1	40	18	7
	0	15	85	20
	-1	2	12	52

$$Akurasi = \frac{((40+85+52)+(0+0+0))}{((40+85+52)+(17+30+27)+(25+35+14)+(0+0+0))} \; x \; 100\% =$$

70.52 %

$$Presisi = \frac{\frac{40}{(40+15+2)} \text{ x } 65 \text{ x } 100\% + \frac{85}{(85+18+12)} \text{ x } 120 \text{ x } 100\% + \frac{52}{(52+7+20)} \text{ x } 66 \text{ x } 100\%}{(65+120+66)} = 70.82\ \%$$

$$Recall = \frac{\frac{40}{(40+18+7)} \text{ x } 65 \text{ x } 100\% + \frac{85}{(85+15+20)} \text{ x } 120 \text{ x } 100\% + \frac{52}{(52+2+12)} \text{ x } 66 \text{ x } 100\%}{(65+120+66)} =$$

70.52 %

$$F1 - Score = \frac{2 * \frac{\frac{40}{(40+15+2)} x \frac{40}{(40+18+7)}}{\frac{40}{(40+15+2)} + \frac{40}{(40+18+7)}} \text{x } 65 \text{ x } 100\% + 2 * \frac{\frac{85}{(85+18+12)} x \frac{85}{(85+15+20)}}{\frac{85}{(85+18+12)} + \frac{85}{(85+15+20)}} \text{ x } 120 \text{ x } 100\% + 2 * \frac{\frac{52}{(52+7+20)} x \frac{52}{(52+2+12)}}{\frac{52}{(52+7+20)} + \frac{52}{(52+2+12)}} \text{ x } 66 \text{ x } 100\%}{(65+120+66)} = 76.43\ \%$$

2) *Confusion Matrix Random Forest*

Berikut pada Tabel 3.53 merupakan *confusion matrix random forest*.

Tabel 3.53 *Confusion matrix Random Forest*

		Prediksi		
		1	0	-1
Actual	1	18	46	1
	0	0	118	2
	-1	1	35	30

$$Akurasi = \frac{((18+118+30)+(0+0+0))}{((18+118+30)+(1+81+3)+(47+2+36)+(0+0+0))} x\ 100\% =$$

66.14 %

$$Presisi = \frac{\frac{18}{(18+0+1)} \text{x } 65 \text{ x } 100\% + \frac{118}{(118+46+35)} \text{x } 120 \text{ x } 100\% + \frac{30}{(30+1+2)} \text{x } 66 \text{ x } 100\%}{(65+120+66)} =$$

76.79 %

$$Recall = \frac{\frac{18}{(18+46+1)} \text{x } 65 \text{ x } 100\% + \frac{118}{(118+0+2)} \text{x } 120 \text{ x } 100\% + \frac{30}{(30+1+35)} \text{x } 66 \text{ x } 100\%}{(65+120+66)} =$$

66.14 %

$$F1 - Score = \frac{2 * \frac{\frac{18}{(18+0+1)} x \frac{18}{(18+46+1)}}{\frac{18}{(18+0+1)} + \frac{18}{(18+46+1)}} \text{x } 65 \text{ x } 100\% + 2 * \frac{\frac{118}{(118+46+35)} x \frac{118}{(118+0+2)}}{\frac{118}{(118+46+35)} + \frac{118}{(118+0+2)}} \text{ x } 120 \text{ x } 100\% + 2 * \frac{\frac{30}{(30+1+2)} x \frac{30}{(30+1+35)}}{\frac{30}{(30+1+2)} + \frac{30}{(30+1+35)}} \text{ x } 66 \text{ x } 100\%}{(65+120+66)} =$$

62.4 %