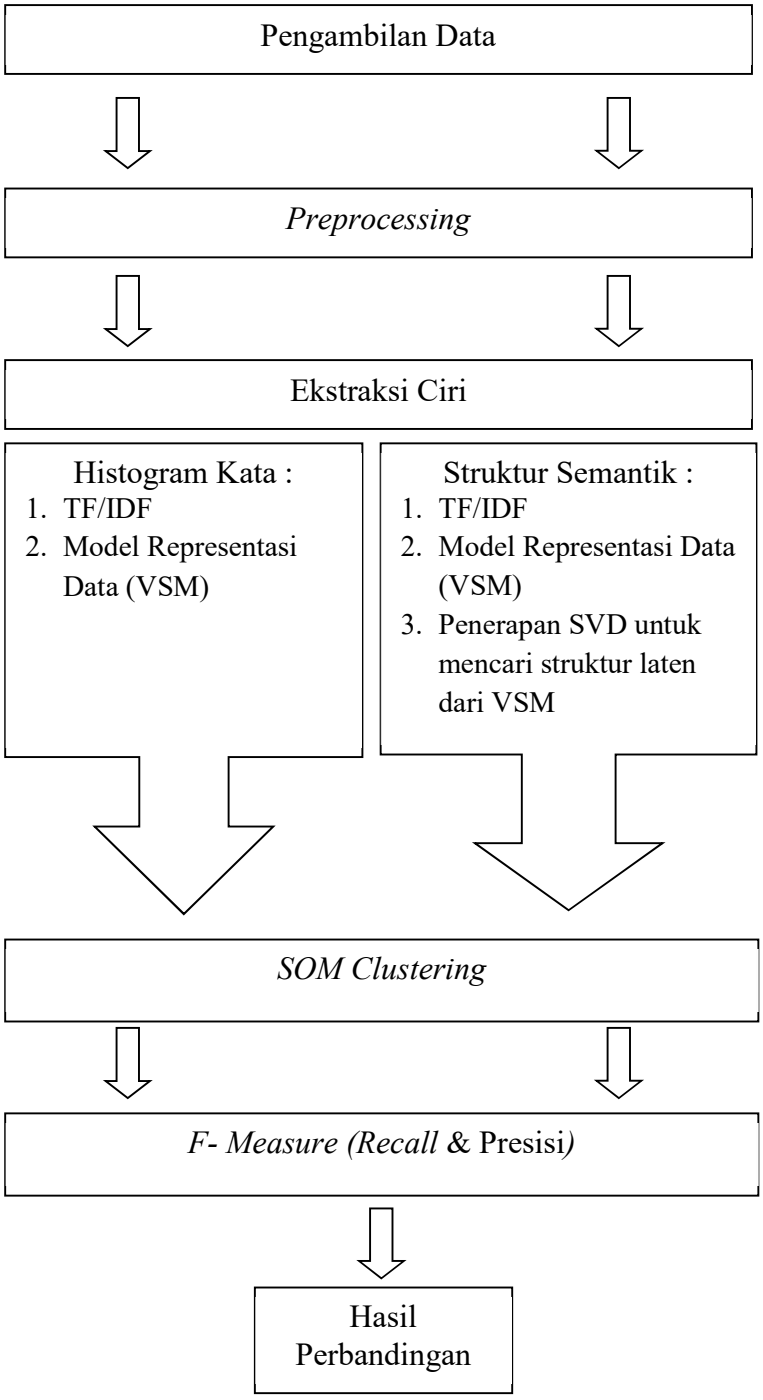
BAB 4
METODOLOGI PENELITIAN

4.1 Metodologi Penelitian

Metodologi penelitian yang digunakan pada penelitian ini secara rinci dapat dilihat pada gambar di bawah ini :



Gambar 4.1 : Metodologi Penelitian

Secara keseluruhan proses pengelompokan dilakukan dengan menggunakan bantuan bahasa pemrograman python. Versi python yang digunakan adalah python versi 2.7.12. Dalam rangka mempermudah proses pelacakan variabel, digunakan salah satu IDE Python yang populer yaitu, Spyder 3.1.2.

4.2 Pengambilan Data

Penelitian diawali dengan cara mengumpulkan data yang akan diproses. Data diambil dari beberapa situs penyedia berita, antara lain :

1. http://detik.com
2. http://www.kompas.com
3. http://www.liputan6.com

Data mentah yang diperoleh berupa data dengan format html. Format tersebut merupakan format standar halaman web. Umumnya sebuah halaman web terdiri dari banyak bagian antara lain : iklan, menu utama, gambar, video, judul , konten dan lain sebagainya. Untuk mempermudah proses pengelompokan, maka data yang diambil adalah data yang terkandung pada bagian judul dan kontennya saja.

Bagian konten yang diambil masih merupakan data yang berformat html, mengingat terkadang pada bagian tersebut terkadang masih terdapat tautan URL atau tag *anchor* yang mengarah pada halaman lainnya. Oleh sebab itu harus dilakukan proses pembuangan tag html yang tidak diperlukan, sehingga hasil akhir yang diperoleh adalah data mentah yang berformat teks biasa (tidak mengandung tag html). Untuk keperluan pengambilan teks mentah dari beberapa situs berita , digunakan pustaka perayapan scrapy versi 1.2.2 yang merupakan salah satu pustaka perayapan *web* dalam bahasa pemrograman python.

4.3 *Preprocessing*

Terdapat beberapa tahapan pada *preprocessing,* yang secara terurut adalah sebagai berikut :

1. Memecah seluruh bagian dokumen berdasarkan tanda baca (karakter selain alfabet) menjadi vektor *string*.
2. Membuang elemen vektor yang berisi tanda baca seperti titik, koma, garis miring, tanda petik dan tanda baca lainnya.

3. Membuang elemen vektor yang merupakan bagian dari kata hubung. Daftar kata hubung disimpan pada kamus berbentuk *file* teks, seluruh elemen vektor yang terdapat pada kamus kata hubung dibuang dari proses.
4. Membuang imbuhan. Pada tahap ini digunakan sastrawi, sebuah pustaka *stemmer* berbahasa PHP (Librian 2016). Pembuangan imbuhan menghasilkan vektor kata dasar. Selanjutnya vektor kata dasar ini dikembalikan lagi menjadi sekumpulan artikel yang hanya berbentuk kata dasar saja.
5. Tahap terakhir adalah menyimpan artikel yang sudah menjadi kumpulan kata dasar ke dalam database MySQL. Hal ini dilakukan untuk mempermudah pengambilan dalam proses uji coba, sehingga tidak perlu lagi koneksi internet jika diperlukan data artikel.

4.4 Ekstraksi Ciri

Terdapat dua jenis ciri yang akan digunakan dalam mengelompokkan artikel, antara lain :

4.4.1 Kemunculan Kata

Ciri kemunculan kata dikemas dalam bentuk matriks, dimana baris mewakili dokumen dan kolom mewakili kata. Setiap sel matriks berisi bobot *TF/IDF* suatu kata pada suatu dokumen. Satu dokumen bisa dianggap sebagai satu vektor kumpulan bobot kata. Lebih jelas mengenai representasi ciri kemunculan kata dapat dilihat pada tabel di bawah ini :

Tabel 4.1 : Contoh matriks bobot *TF/IDF*

	Kata1	Kata2	Kata3	Kata4	KataN
Dok1	$\frac{tf}{idf}_{1,1}$	$\frac{tf}{idf}_{2,1}$	$\frac{tf}{idf}_{3,1}$	$\frac{tf}{idf}_{4,1}$	$\frac{tf}{idf}_{n,1}$
Dok2	$\frac{tf}{idf}_{1,2}$	$\frac{tf}{idf}_{2,2}$	$\frac{tf}{idf}_{3,2}$	$\frac{tf}{idf}_{4,2}$	$\frac{tf}{idf}_{n,2}$
DokN	$\frac{tf}{idf}_{1,n}$	$\frac{tf}{idf}_{2,n}$	$\frac{tf}{idf}_{3,n}$	$\frac{tf}{idf}_{4,n}$	$\frac{tf}{idf}_{n,n}$

Proses pembangkitan matriks kemunculan kata dilakukan dengan bantuan pustaka TfidfVectorizer yang merupakan bagian dari pustaka sklearn.

4.4.2 Struktur Laten

Matriks Latensi merupakan ciri turunan dari matriks kemunculan kata. Latensi dapat dipandang sebagai hubungan antara dua buah artikel yang tidak memiliki kesamaan kemunculan kata. Hubungan tersebut muncul karena terdapat artikel lain yang memiliki kesamaan kemunculan kata dengan artikel ke satu dan ke dua. Misal terdapat 9 dokumen singkat dengan konten sebagai berikut :

1. Romeo dan juliet.
2. Romeo terbunuh oleh belati.
3. Juliet terbunuh oleh racun.
4. Racun dan belati merupakan benda mematikan.
5. Pak roni seorang montir mobil.
6. Pak roni bekerja di sebuah dealer.
7. Dealer A menjual mobil ferrari.
8. Kendaraan yang dijual di dealer A sangat bagus.
9. Mobil ferrari merupakan kendaraan yang nyaman.

Jika seluruh dokumen ini dikenakan operasi preprocessing dan diekstrak ciri kemunculan katanya (dikonversi menjadi matriks TF/IDF), maka akan menghasilkan sebuah matriks dengan dimensi 9 x 17. Masing-masing baris pada matriks tersebut mewakili satu objek dokumen, sedangkan kolom mewakili ciri kemunculan kata, sehingga objek dokumen dapat dianggap sebagai vektor baris pada matriks TF/IDF.

Kita dapat mengetahui hubungan antara satu vektor dengan vektor lainnya dengan cara menghitung koefisien korelasi. Jika suatu vektor memiliki hubungan dengan vektor lain, maka koefisien korelasi akan bernilai mendekati 1, sebaliknya koefisien korelasi akan bernilai mendekati -1 jika tidak ada keterkaitan antara 2 buah vektor. Dengan kata lain, dokumen juga dapat kita anggap demikian, bahwa dua dokumen dapat dianggap memiliki hubungan kedekatan makna ketika nilai koefisien korelasi mendekati 1 serta sebaliknya.

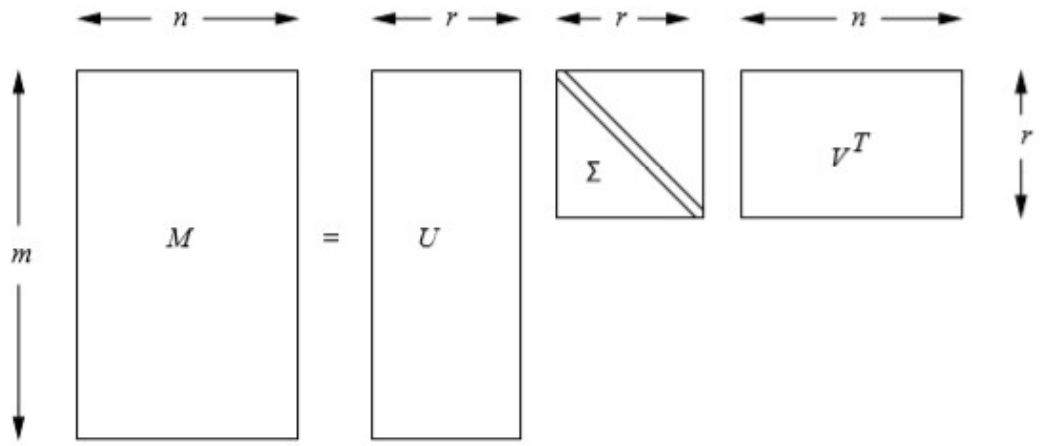
Pada matriks TF/IDF dokumen 1 dan dokumen 2, koefisien korelasi bernilai 0,31, hal ini wajar mengingat pada dokumen 1 dan setidaknya ada kemiripan

kemunculan kata khususnya pada kata ‘Romeo’. Namun bagaimana bila kita melihat korelasi antara dokumen 5 dengan dokumen 8, korelasi antara keduanya bernilai -0.20. Seolah-olah antara keduanya tidak memiliki hubungan keterkaitan. Padahal jika dicermati dokumen 5 dan 8 sama-sama membahas topik otomotif.

Pada umumnya pembaca akan menganggap dokumen 5 memiliki keterkaitan dengan dokumen 8 salah satunya karena pada dokumen 5 memiliki komposisi kata yang juga terdapat pada dokumen 7 (kata ‘mobil’), selanjutnya pada dokumen 7 memiliki komposisi kata yang juga terdapat pada dokumen 8 (kata ‘dealer’). Secara sederhana terjadinya proses induksi antara dokumen 5 dan 8 terlihat pada gambar 4.2. Dengan adanya proses induksi kemunculan kata, seharusnya dokumen 5 berpotensi memiliki hubungan dengan dokumen 8 (memiliki nilai korelasi > 0).

Gambar 4.2 : Proses pembentukan latensi

Induksi korelasi yang terjadi antara dokumen 5 dengan dokumen 8 dapat dianggap sebagai latensi yang terjadi antara dokumen 5 dengan dokumen 8. Untuk memunculkan latensi tersebut diperlukan sebuah pendekatan yang mampu meningkatkan korelasi antara dokumen 5 dengan dokumen 8. Salah satu pendekatan dalam aljabar linier yang dapat digunakan untuk meningkatkan potensi korelasi antar vektor dalam suatu matriks adalah *Singular Value Decomposition* (SVD). SVD memungkinkan kita untuk dapat melihat potensi korelasi antar variabel sekaligus memperkecil korelasi antar variabel yang tidak berpotensi memiliki hubungan korelasi. SVD merupakan salah satu teknik pemfaktoran matriks A menjadi 3 buah matriks, yaitu matriks ortogonal U, matriks diagonal S dan transpose dari matriks ortogonal V. Gambaran sederhana SVD dapat dilihat pada gambar 4.3.



Gambar 4.3 : Dampak faktorisasi secara SVD

Sumber : (Ullman, Rajaraman and Leskovec 2014)

Bila matriks TF/IDF dengan dimensi 9 x 17 dikenakan operasi SVD, maka akan terbentuk tiga buah matriks yang masing-masing adalah matriks U dengan dimensi 9 x 9, matriks diagonal S (9 x 9) dan matriks V (9 x 17). Selanjutnya, jika kita hanya mengambil 2 kolom dari matriks U, maka akan terbentuk matriks baru yang memiliki dimensi yang lebih sederhana dari matriks TF/IDF. Jika masing-masing matriks TF/IDF dan matriks U dihitung korelasi antar dokumennya, maka perubahan korelasi terlihat pada gambar 3. Gambar 4.4 menunjukkan terjadinya penguatan korelasi antar dokumen 5 dengan dokumen 8 yang semula -0,2 meningkat menjadi 1,0, karena memang sebenarnya antara kedua dokumen tersebut berpotensi memiliki kedekatan makna.

Pembuangan kolom pada matriks *U* tentunya harus sejalan dengan pembuangan nilai *singular* (diagonal dari matriks *S*) terurut mulai dari yang terbesar. Artinya jumlah kolom U yang dibuang sama dengan jumlah nilai singular yang dibuang. Pola yang terbentuk dari hubungan antara pembuangan nilai singular dengan *F-Measure* selanjutnya akan dipelajari. Pada beberapa referensi disarankan untuk mempertahankan energi dari *S* agar tetap berada di atas 90% (Ullman, Rajaraman and Leskovec 2014), hal ini dilakukan agar nilai aproksimasi tidak terlalu jauh dengan matriks M yang merupakan matriks aslinya.

Korelasi antar dokumen pada matriks TF/IDF

	1	2	3	4	5	6	7	8	9
1	1.0	0.3	0.3	-0.2	-0.1	-0.2	-0.2	-0.2	-0.2
2	0.3	1.0	0.2	0.1	-0.2	-0.2	-0.2	-0.3	-0.2
3	0.3	0.2	1.0	0.1	-0.2	-0.2	-0.2	-0.3	-0.2
4	-0.2	0.1	0.1	1.0	-0.2	-0.3	-0.3	-0.3	-0.3
5	-0.1	-0.2	-0.2	-0.2	1.0	0.3	-0.2	-0.2	-0.2
6	-0.2	-0.2	-0.2	-0.3	0.3	1.0	0.1	0.0	-0.2
7	-0.2	-0.2	-0.2	-0.3	-0.2	0.1	1.0	0.0	0.2
8	-0.2	-0.3	-0.3	-0.3	-0.2	0.0	0.0	1.0	0.1
9	-0.2	-0.2	-0.2	-0.3	-0.2	-0.2	0.2	0.1	1.0

Korelasi antar dokumen pada matriks struktur laten

	1	2	3	4	5	6	7	8	9
1	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
2	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
3	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
4	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
5	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
6	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
7	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
8	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
9	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0

Gambar 4.4 : Visualisasi perubahan korelasi sebelum dan sesudah operasi SVD.

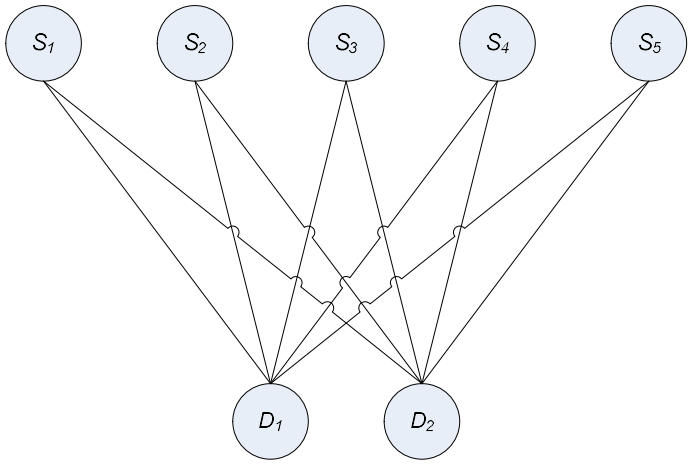
Proses dekomposisi matriks kemunculan kata secara SVD dilakukan dengan memanfaatkan pustaka TruncatedSVD yang merupakan bagian dari paket pustaka sklearn.

4.5 Pengelompokan

Terdapat 2 jenis matriks yang mewakili karakteristik dokumen observasi, yakni matriks kemunculan kata dan matriks latensi. Masing-masing matriks dibagi menjadi 2 bagian, yakni sebagai data latih dan sebagai data uji (testing). Untuk setiap matriks, akan dilakukan proses pengelompokan yang tahapannya adalah sebagai berikut :

4.5.1 Pembentukan Topologi

Topologi merupakan gambaran luaran yang diharapkan. Dalam kasus *clustering*, luaran yang diharapkan adalah himpunan Kluster. Topologi diwujudkan dalam kumpulan simpul, sehingga jumlah simpul sama dengan jumlah Kluster yang diharapkan. Masing-masing simpul akan berhubungan tepat satu persatu dengan objek dokumen. Jika S adalah simpul dan D adalah dokumen yang ingin dikluster, maka hubungan antara simpul dengan dokumen dapat dilihat pada gambar di bawah :



Gambar 4.5 : Hubungan antara simpul (Kluster **yang diharapkan) dengan dokumen**

Mula-mula setiap simpul akan diberikan bobot secara acak sebanyak jumlah ciri dokumen. Jika n adalah jumlah ciri dokumen, maka tiap-tiap simpul akan berisi bobot W_1, W_2, W_3 ... W_n .

4.5.2 Menentukan *Best Matching Unit (BMU)*

Setiap dokumen akan dianggap sebagai masukan untuk seluruh simpul. Karena setiap dokumen memiliki sejumlah ciri, maka setiap dokumen bisa dianggap sebagai sebuah vektor. BMU merupakan simpul dengan jarak terdekat dengan suatu dokumen. BMU bisa dikatakan sebagai simpul yang memiliki kemiripan tertinggi dengan data masukan. Untuk mencari kemiripan antara simpul dengan data masukan digunakan fungsi jarak *cosine similarity*. Selanjutnya simpul yang dianggap paling mirip adalah simpul dengan jarak *cosine* terkecil.

4.5.3 Perubahan Bobot Simpul

Setelah diperoleh BMU, langkah selanjutnya adalah melakukan penyesuaian bobot dengan menggunakan persamaan 2.9. Bobot pada BMU dan simpul yang menjadi tetangga BMU akan mengalami perubahan pula. Penentuan tetangga BMU dapat diketahui melalui fungsi ketetanggaan yang nilainya akan meluruh setiap putaran waktu. Setelah bobot berubah, proses selanjutnya adalah mengulangi proses 4.5.2 sampai dengan 4.5.3 untuk dokumen lainnya sampai seluruh dokumen terpenuhi. Satu proses tersebut dianggap sebagai satu *epoch*. Proses akan dijalankan sampai beberapa *epoch* sampai dengan nilai bobot konvergen atau sejumlah *epoch* yang diinginkan telah terpenuhi. Bobot yang telah mencapai nilai konvergen dapat dijadikan sebagai pedoman dalam menentukan suatu dokumen layak berada pada Kluster yang mana berdasarkan jarak dokumen terhadap simpul.

Untuk menghindari proses pelatihan secara berulang, bobot yang telah mencapai nilai konvergen disimpan ke dalam *file* tersendiri. Bobot konvergen yang tersimpan ini kemudian dapat dianggap sebagai model. Model inilah yang selanjutnya akan digunakan pada proses pengujian.

4.5.4 Proses Pengujian.

Setelah diperoleh model yang berisi bobot konvergen, tahap selanjutnya adalah proses pengujian. Proses pengujian dapat dianggap sebagai proses penerapan model untuk menentukan data masukan berada pada kluster yang mana. Sejumlah data uji dimasukkan untuk dicari simpul mana yang memiliki jarak yang terdekat dengan masing-masing data uji. Simpul yang menjadi jarak terdekat dengan objek data uji akan menjadi kelompok (Kluster) pada dokumen yang bersangkutan.

Seluruh proses pembelajaran SOM dilakukan dengan memanfaatkan pustaka yang diperoleh dari (Vettigli 2016). Pustaka ini merupakan salah satu pustaka penerapan SOM standar dengan menggunakan fungsi ketetanggaan gaussian.

4.6 Evaluasi

Proses evaluasi didasarkan pada data kategori yang sebenarnya. Hasil pengelompokan yang dilakukan oleh proses clustering akan dibandingkan dengan hasil pengelompokan yang didasarkan pada kategori artikel sebenarnya (*Ground Truth*). Selanjutnya istilah *Ground Truth* akan disebut **GT**. Perlu diingat bahwa pengelompokan yang dilakukan dalam penelitian ini adalah merupakan salah satu bentuk dari *unsupervised learning*, sehingga dalam menentukan kualitas pengelompokan tidak cukup hanya didasarkan pada label yang terkandung pada GT saja, misalkan tabel perbandingan antara clustering dengan Ground Truth adalah sebagai berikut :

Tabel 4.2 : Contoh perbandingan label GT dengan label cluster

Dokumen	Label GT	Label Kluster
1	Bola	Ekonomi
2	Bola	Ekonomi
3	Ekonomi	Bola
4	Ekonomi	Bola

Jika dilihat dari aspek labelnya, jelas contoh hasil pengelompokan di atas merupakan pengelompokan yang sama sekali tidak akurat. Namun jika dilihat dari aspek *unsupervised learning*, hasil di atas merupakan proses clustering yang sepenuhnya akurat, karena telah mampu mengelompokkan subjek yang sejenis dalam kelompok yang sama, terlepas dari label Kluster yang dihasilkan.

Untuk dapat mencapai evaluasi dari aspek *unsupervised learning*, maka diperlukan evaluasi yang tidak hanya didasarkan pada label saja, namun juga diukur dari pasangan antar subjek dokumen (pairwise). Seluruh subjek yang diteliti akan dipasangkan satu sama lain sehingga akan membentuk N(N-1)/2 pasangan, dimana :

1. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang sama dan di Kluster juga memiliki label yang sama, maka pasangan tersebut dihitung sebagai True Positif (TP).
2. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang berbeda dan di Kluster juga memiliki label berbeda, maka pasangan tersebut dihitung sebagai True Negatif (TN).

3. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang berbeda namun di Kluster memiliki label yang sama, maka pasangan tersebut dihitung sebagai False Positif (FP).
4. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang sama namun di Kluster memiliki label berbeda, maka pasangan tersebut dihitung sebagai False Negatif (FN).

Masing-masing nilai TP, TN, FP dan FN akan digunakan sebagai dasar untuk menghitung presisi dan *recall*, selanjutnya nilai presisi dan *recall* akan digunakan untuk menghitung F-*Measure*.