JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 6
No. 1
April 2020

Implementasi *Single Pass Clustering* pada *Preprocessing* Temu Kembali Koleksi Berita Teks

Faisal Rahutomo[#1], Dwi Puspitasari[#2], Trie Endah Sulistyoningrum[#3]

[#]*Teknologi Informasi, Politeknik Negeri Malang*
Jalan Soekarno Hatta No 9, Jatimulyo, Lowokwaru, Malang

[1]faisal@polinema.ac.id

[2]dwi_sti@yahoo.com

[3]trieendhs@gmail.com

Abstrak— **Berita saat ini masih menjadi sumber yang dipercaya untuk mendapatkan informasi. Namun seiring dengan perkembangan teknologi berita yang terbit menjadi semakin banyak jumlahnya. Akibat dari jumlah berita yang banyak diperlukan suatu sistem yang dapat dipergunakan untuk menemukan berita dengan cepat. Sistem Temu Kembali menjadi cara yang dapat dipergunakan untuk membantu menangani masalah tersebut. Sistem temu kembali yang ada masih terus dikaji efisiensinya jika berhubungan dengan jumlah informasi yang sangat besar. Makalah ini melakukan pengujian efektifitas dan efisiensi tambahan *preprocessing* pada sistem temu kembali. Langkahnya yaitu mengklasterkan informasi yang ada terlebih dahulu. Pada *preprocesing* ini diimplementasikan metode *single pass clustering*. Kemudian pencocokan *query* dengan dokumen disederhanakan kepada pencocokan *query* dengan *centroid* klaster. Hasil uji coba efisiensi menunjukkan bahwa sistem temu kembali yang mengimplementasikan *single pass clustering* mampu mencari berita dengan lebih cepat. Sedangkan pengujian efektifitas untuk mengetahui seberapa tepat berita yang bisa diketahui dari nilai pengujian *precision*, *recall*, dan *f-score*. Dari pengujian tersebut didapatkan hasil jika proses pencarian paling tepat dilakukan pada *cluster* dengan nilai *threshold* 0,1. Pengujian pada *cluster threshold* 0,1, *f-score* terbaik didapatkan ketika dilakukan proses temu kembali berita dengan *keyword* '4g lte' bernilai 0,732. Sedangkan pengujian *f-score* terburuk terdapat pada pengujian dengan *keyword* 'aplikasi whatsapp' dengan nilai 0,111. Sedangkan secara umum, sistem yang diusulkan selalu lebih cepat hanya saja lebih rendah nilai performa *precision*, *recall*, dan *f-score*-nya.**

Kata kunci— **Berita, Klastering, Sistem Temu Kembali, *Preprocessing*, *Single Pass Clustering*.**

I. PENDAHULUAN

Berita saat ini telah menjadi bagian yang tidak terpisahkan dari kehidupan setiap orang. Pengaruh perkembangan teknologi juga berpengaruh pada jumlah berita yang meningkat. Oleh karena itu dibutuhkan suatu sistem yang dapat digunakan untuk menemukan berita yang revelan dengan cepat. Sistem temu kembali atau *information retrieval* menjadi cara yang dapat digunakan untuk memenuhi kebutuhan tersebut [1][2].

Sistem temu kembali informasi adalah sistem yang digunakan untuk menemukan informasi, baik berbentuk teks, gambar, dan lainnya yang sesuai dengan keinginan pengguna. Sumber informasi bisa dalam jumlah yang sangat besar. Sistem temu kembali informasi sendiri telah menjadi cabang ilmu pada ilmu komputer. Namun terkadang muncul masalah efisiensi ketika sistem berjalan pada data yang sangat besar. Kurang efisiennya sistem dikarenakan waktu tunggu sistem menjadi lebih lama karena diperlukan untuk menghitung tingkat kemiripan *query* dengan masing-masing dokumen.

Metode *Clustering* dapat digunakan dalam pengelompokan dokumen [3]. Caranya dengan mengelompokan dokumen-dokumen ke dalam *cluster* berdasarkan kedekatan atau kemiripan antar dokumen [1][2]. Di dalam makalah ini, sistem yang dibangun adalah implementasi metode *single pass clustering* [4] untuk mengelompokkan dokumen berita terlebih dahulu. Tujuan pengelompokan dokumen untuk mengurangi jumlah pencocokan *query* dengan dokumen, dan hanya mencocokkan *query* dengan *centroid* kelompok dokumen. Pembentukkan klaster dilakukan sebagai langkah *preprocessing*. Sedangkan pengevaluasian tingkat kemiripan antar dokumen dengan kluster dilakukan dengan *cosine similarity*. Tujuan dari dilakukan perbandingan antar dua sistem temu kembali adalah untuk mengetahui apakah sistem yang mengimplementasikan proses klastering memang lebih baik hasilnya daripada sistem temu kembali biasa. Kemudian makalah ini melakukan evaluasi terhadap performa sistem *precision*, *recall*, dan *f-score* antara sistem tanpa *clustering* dan dengan *clustering*. Evaluasi waktu juga dilakukan untuk melihat perbedaannya.

Penelitian di dalam makalah ini melakukan pengembangan eksperimen terkait pemanfaatan

Submitted 17-07-2019; Revised 05-11-2019; Accepted 10-01-2020

pembelajaran mesin pada kasus teks berbahasa Indonesia. Penelitian yang dilakukan peneliti dan tim sebelumnya terkait analisa sentimen [5][6], deteksi hoax [7][8], klasifikasi lirik lagu [9], klasifikasi review film [10], dan klasifikasi artikel wikipedia [11]. Penelitian ini juga melanjutkan topik penelitian yang dilakukan tim terkait sistem temu kembali informasi yang dilakukan peneliti dan tim [12][13][14][15][16]. Harapannya makalah yang disusun ini dapat memnberikan kontribusi memperkaya khasanah pengetahuan sebagaimana yang telah dilakukan banyak peneliti yang lainnya di bidang ini.

II. METODOLOGI

A. Single Pass Clustering

Single pass clustering merupakan suatu tipe pengelompokan yang mengelompokan data satu demi satu. Hal itu dilakukan seiring dengan evaluasi setiap data yang masuk ke dalam *cluster* [4]. Evaluasi tingkat kesamaan antar data dan *cluster* dapat dilakukan dengan berbagai macam cara termasuk fungsi jarak, *vectors similarity*, dan lainnya. Contoh vektor dokumen yang diproses dengan menggunakan s*ingle pass clustering* ditunjukkan pada Gambar 1.

	T1	T2	T3	T4	T5
Doc1	1	2	0	0	1
Doc2	3	1	2	3	0
Doc3	3	0	0	0	1
Doc4	2	1	0	3	0
Doc5	2	2	1	5	1

Gambar. 1 Contoh vektor dokumen

Algoritma *single pass clustering* bekerja dengan langkah-langkah sebagai berikut [4].

- Doc1 (dokumen pertama) yang masuk kedalam proses *clustering* akan secara otomatis digunakan sebagai representasi C1 (*cluster* pertama).
- Untuk Doci (dokumen ke-i) akan dilakukan perhitungan untuk mencari kesamaan (*similarity*) dengan setiap wakil dari masing-masing cluster yang telah terbentuk.
- Jika S_{max} (*maximum similarity*) lebih besar dari batas nilai S_T (*threshold value*), dokumen akan ditambahkan sebagai anggota pada cluster yang bersesuaian dan akan dilakukan perhitungan kembali titip pusat (*centroid*) rata-rata *vector terms cluster*. Sebaliknya Doci akan digunakan sebagai inisialisasi *cluster* baru apabila nilai *similarity* tidak lebih dari nilai S_T.
- Jika masih ada sebuah item Doci yang belum dikelompokkan, kembali dilakukan proses pada langkah ke-2.

B. Text Processing

Untuk bisa mengolah dokumen teks berita supaya bisa menghasilkan luaran yang diinginkan maka harus dilakukan pengolahan teks terlebih dahulu. Tahapan pengolahan teks yang dilakukan pada makalah ini sebagai berikut.

- *Case folding:* Dalam *case folding*, terjadi proses mengubah semua huruf teks menjadi huruf kecil semua atau menjadi huruf kapital semua. Pada penelitian ini semua huruf dirubah menjadi huruf kecil karena mayoritas teks berupa huruf kecil semua.
- *Tokenization*: Proses pemecahan kumpulan teks dalam dokumen atau kalimat menjadi bentuk satuannya. Pada makalah ini tanda baca spasi penanda pemisah teks sehingga menjadi kumpulan token yang biasanya berupa kata.
- *Stopword removal*: Proses penghapusan kata yang dianggap tidak penting dengan membandingkannya dengan daftar kata yang tidak penting (*stop word list*). Daftar kata ini bersifat unik, tiap-tiap bahasa memiliki daftar katanya tersendiri. Di dalam bahasa Indonesia terdapat beberapa versi daftar kata ini [14].
- *TF.IDF*: proses pembobotan pada pembentukan vektor istilah dari dokumen yang ada [17]. *TF* atau *Term Frequency* adalah jumlah kemunculan suatu kata pada dokumen. Sedangkan *TF.IDF* adalah salah satu metode terbaik untuk melakukan pengukuran berat dari suatu kata. Nilai *TF.IDF* bertambah sesuai dengan jumlah kemunculan kata dalam dokumen namun terhalangi nilainya dengan jumlah kata dalam *corpus*.

C. Cosine Similarity

Cosine similarity merupakan metode yang digunakan untuk menghitug tingkat kesamaan (*similarity*) antar dua buah vektor. Untuk tujuan klastering dokumen dapat digunakan fungsi *cosine similarity*. Cosine Similarity dipilih karena cara kerja metode ini efisien, mudah dalam representasi dan dapat diimplementasikan pada pencocokan dokumen [6][7].

$$Cosine(X,Y) = \frac{X.Y}{\sqrt{|x|^2|y|^2}} \quad (1)$$

Keterangan :

x : vektor istilah dari dokumen X
y : vektor istilah dari dokumen Y
$|x|$: normalisasi vektor istilah dari dokumen X
$|y|$: normalisasi vektor istilah dari dokumen Y

D. Pengukuran Performa

Pengukuran performa sistem temu kembali informasi yang digunakan pada penelitian ini adalah *precision*, *recall*, dan *f-score*.

1) *Precision*: *Precision* dipergunakan untuk mengukur tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Nilai *Precision* tertinggi adalah 1, yang berarti seluruh dokumen yang ditemukan adalah relevan.

$$precision = \frac{N2 \cap N1}{N2} \quad (2)$$

2) *Recall* : *Recall* dipergunakan untuk mengukur tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Nilai Recall tertinggi adalah 1, yang berarti seluruh dokumen dalam koleksi yang benar berhasil ditemukan.

$$recall = \frac{N2 \cap N1}{N1} \quad (3)$$

Dimana:
N1 : jumlah artikel yang sesuai dari keseluruhan artikel dengan *keyword* yang diujikan berdasar pengetahuan penguji
N2 : jumlah artikel yang ditampilkan sistem

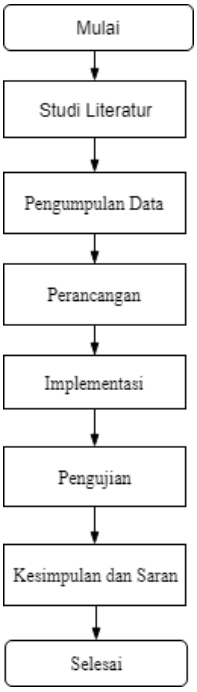
3) *F-Score*: *F-Score* merupakan rata-rata harmonik dari *precision* dan *recall*.

$$f - score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

4) Pengujian waktu: Uji waktu yang dilakukan adalah untuk menguji apakah waktu yang dibutuhkan untuk proses temu kembali oleh sistem yang dibuat lebih cepat dari pada sistem yang tidak menggunakan metode *single pass clustering* pada *preprocessing* temu kembali beritanya.

E. Tahapan Penelitian

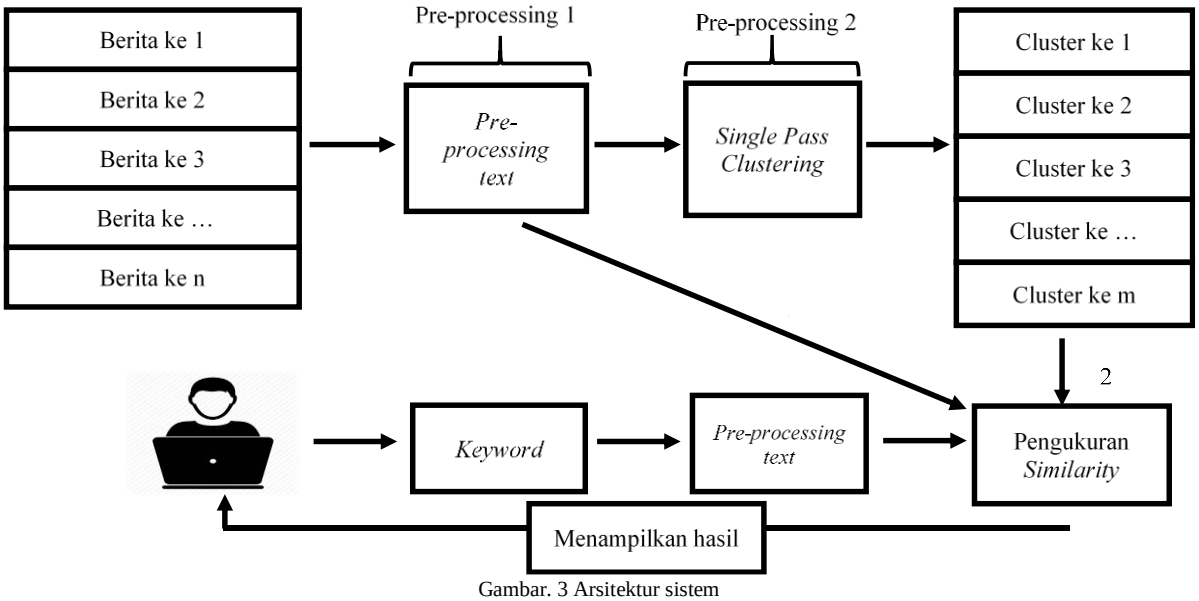
Tahapan penelitian dalam implementasi *single pass clustering* pada *preprocessing* temu kembali koleksi teks dilakukan seperti pada Gambar 2 sebagai berikut.



Gambar. 2 Tahapan penelitian

1) *Studi Literatur*: Studi literatur dilakukan untuk mengumpulkan informasi mengenai data-data yang diperlukan untuk analisa sistem termasuk *dataset*. Informasi mengenai perhitungan dalam metode algoritma *single pass clustering* dan *cosine similarity*, yang digunakan pada aplikasi/ sistem yang dibangun. Studi literatur ini dilakukan dengan cara menggunakan internet, jurnal serta buku untuk mendapatkan referensi yang terkait dengan apa yang dibutuhkan. Sumber studi literatur diperoleh baik dari dalam maupun luar negeri.

2) *Pengumpulan Data:* Data yang digunakan dalam penelitian ini berupa berita yang berasal dari *corpus*. *Corpus* yang dipergunakan adalah *Indonesian news corpus* [13]. *Corpus ini* telah dibuat di dalam penelitian sebelumnya. Corpus *berita* tersebut bisa didapatkan



Gambar. 3 Arsitektur sistem

melalui internet pada situs Mendeley data. *Corpus* yang dipergunakan berformat xml dan json. Sedangkan data berita yang diambil adalah berita yang berasal dari beberapa portal berita online. *Corpus* ini berasal dari kumpulan berita mulai dari bulan Juli 2015 hingga Desember 2015. Data yang dipilih pada penelitian ini adalah data dari portal kompas.com dan republika.co.id dengan kategori "teknologi".

3) *Perancangan.* Dalam tahap perancangan, ditentukan arsitektur dari sistem dan perancangan antarmuka dari sistem yang akan dibuat. Proses dalam sistem yang dibangun diterapkan pada *corpus* yang berisi berita dari beberapa situs *online*. Proses yang berjalan dalam sistem berkerja seperti pada Gambar 3. Di dalam penelitian ini diterapkan sebuah *preprocessing* yang ditujukan untuk mengklasterkan berita. Hasil dari proses klastering inilah dipakai untuk proses temu kembali informasi berdasarkan *query* pengguna. Berita sebelum diklaster diproses pada *text processing*. Tujuan proses ini untuk mendapatkan nilai vektor dokumen yang bisa diolah lebih lanjut pada proses klaster. Dalam pemrosesan teks ini dilakukan proses *case folding*, *stopword removal*, dan *tokenization*. Hasil *preprocessing text* kemudian akan dilakukan pehitungan untuk mendapatkan nilai *TF* dan *TF-IDF*. Dalam proses pengelompokan dengan *single pass clustering* dibutuhkan nilai *threshold*. Perbedaan nilai *threshold* yang dipergunakan akan mempengaruhi jumlah *cluster* yang terbentuk. Semakin besar suatu nilai *threshold* maka jumlah *cluster* yang terbentuk akan semakin banyak. Berita yang telah terklaster inilah yang dipergunakan dalam proses temu kembali. Proses temu kembali pada sistem yang dibangun akan dilakukan dengan cara mencari tingkat kemiripan antara *query* dari pengguna dengan *centroid* masing-masing klaster. Kecepatan dan ketepatan hasil temuan sistem ini nantinya akan dibandingkan dengan sistem temu kembali yang tidak mengimplementasikan proses *clustering*. *Implementasi.* Pada tahapan implementasi, dilakukan pembuatan modul-modul yang telah dirancang dalam tahap perancangan kedalam bahasa pemrograman.

4) *Pengujian.* Pengujian merupakan tahapan dimana sistem akan dijalankan. Tahap pengujian diperlukan untuk menjadi ukuran bahwa sistem dapat dijalankan sesuai dengan tujuan. Pengujian yang dilakukan menggunakan metrik *precision*, *recall*, *f-score*, dan waktu.

5) *Kesimpulan dan Saran.* Tahap ini dilakukan untuk mendapatkan kesimpulan yang bisa diambil dari penelitian yang dilakukan dan pemberian saran yang bisa dikembangkan untuk penelitian selanjutnya.

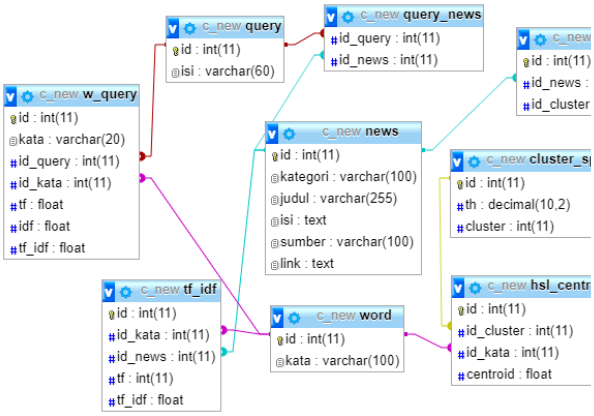
III. HASIL DAN PEMBAHASAN

Setelah dilakukan analisa dan perancangan terhadap sistem yang akan dibangun, selanjutnya adalah implementasi. Tahap implementasi ini dilakukan terhadap dua bagian dari sistem, yakni untuk database dan sistem itu sendiri.

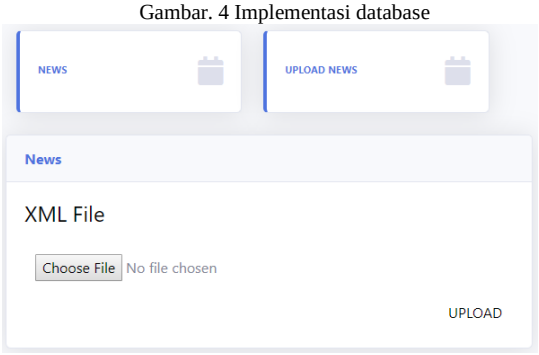
A. *Implementasi Basis Data*

Basis data yang dibangun disusun sebagaimana tampak pada Gambar 4. Database yang dibangun akan memiliki tabel-tabel sebagai berikut.

- Tabel query. Table query ini berisi data query yang digunakan sebagai query yang nantinya digunakan untuk mencari informasi.
- Tabel w_query. Tabel ini digunakan untuk menyimpan data hasil *preprocessing text* query yang tersimpan pada table query.
- Tabel news. Tabel news digunakan untuk menyimpan data dari *corpus* berita. Data berita sendiri pada sistem akan menjadi data utama yang



Gambar. 4 Implementasi database



Gambar. 5 Tampilan *upload corpus*

diproses sedemikian rupa sehingga bisa dikelompokkan.

- Tabel query_news. Tabel query_news digunakan untuk menyimpan penilian manual dari berita mana saja yang sesuai dengan *query* yang terdata.
- Tabel word. Tabel ini berisi daftar kata-kata semua berita yang telah melalui langkah *preprocessing*.
- Tabel tf_idf. Tabel ini digunakan untuk menyimpan data nilai dari masing-masing kata hasil *preprocessing* teks. Nilai tiap kata yang terdata adalah nilai *TF* dan *TF.IDF* setiap kata di masing-masing berita.
- Tabel cluster_spc. Tabel cluster_spc ini berisi data hasil pengklasteran berita menggunakan *single pass clustering*. Pada tabel ini disimpan *cluster* yang

terbentuk untuk setiap nilai *threshold* berbeda yang dipergunakan.

- Tabel hsl_spc. Tabel hsl_spc ini berisi data berita menjadi anggota dari cluster yang mana. Dari tabel ini bisa diketahui anggota dari setiap *cluster* yang terbentuk.
- Tabel hsl_centroid. Tabel hsl_centroid ini berisi data *centroid* dari masing-masing cluster yang terbentuk. Dari tabel bisa diketahui hasil perhitungan centroid nilai *TF.IDF* dari semua anggota yang menyusun cluster.

B. *Implementasi Sistem*

Sistem yang dibuat nantinya akan memiliki tampilan dan fungsi seperti dibawah ini.

- Tampilan *upload corpus:* Tampilan ini adalah antarmuka pengguna yang bisa digunakan oleh pengguna untuk meng-*import* data berita yang digunakan pada sistem. Data berita yang bisa di-*import* adalah kumpulan berita yang berformat xml.
- Tampilan hasil *preprocessing text:* Gambar 6 menunjukkan antarmuka pengguna hasil dari *preprocessing* berita yang ada hingga menghasilkan kata-kata yang tersimpan pada table word beserta nilai perhitungan *TF* dan *TF.IDF*.
- Tampilan hasil klaster berita: Hasil dari proses tersebut ditampilkan seperti pada Gambar 7.
- Tampilan sistem temu kembali: Gambar 8 adalah antarmuka pengguna yang bisa digunakan oleh pengguna untuk mencari berita sesuai dengan *keyword* yang dimasukkan oleh pengguna. Selain pengguna memasukkan *keyword*, pengguna juga diminta untuk memilih nilai *threshold*. Nilai *threshold* ini digunakan untuk mengetahui pengguna ingin melakukan pencarian terhadap klaster hasil dengan nilai *threshold* yang mana.
- Tampilan Hasil Temu Kembali Berita: Ini adalah *user interface* yang nantinya akan menampilkan berita sesuai dengan pencarian oleh sistem.

C. *Pengujian*

Pengujian dilakukan secara keseluruhan terhadap hasil dari temu kembali menggunakan sistem. Pengujian dilakukan dengan cara membandingkan hasil dari pencarian berita secara manual sebagai standar emas dengan hasil berita yang ditampilkan sistem yang telah dibuat. Kemudian akan dilakukan perhitungan pengujian menggunakan *precision*, *recall*, *f-score* dan waktu.

TABEL I
JUMLAH KELAS HASIL CLUSTERING

Threshold	Jumlah Cluster
0.1	206
0.2	360
0.3	451
0.4	549
0.5	633
0.6	703
0.7	758

1) Kelas Hasil Clustering: Tabel 1 menunjukkan banyaknya kelompok berdasarkan *threshold* yang diberikan. Dari tabel tersebut terbukti jika semakin besar nilai *threshold* yang dipergunakan maka jumlah *cluster* yang dihasilkan dari proses klastering juga semakin banyak. Setiap proses temu kembali akan dilakukan pada setiap *cluster-cluster* yang telah terbentuk dengan cara mencari tingkat kemiripan antar *keyword* pengguna dan *centroid* dari masing-masing *cluster*.

2) Pengujian *precision*, *recall*, dan *f-score*: Percobaan dilakukan berulang kali pada beberapa *threshold*. Beberapa *threshold* tersebut adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, dan 0,7. Nilai *threshold* tersebut merupakan nilai *threshold* yang dipergunakan pada saat proses *clastering* menggunakan *single pass clustering*. Sedangkan nilai *threshold* yang dipergunakan pada sistem temu kembali sendiri adalah 0.1.

Kata	Judul Berita	TF	TF-IDF
acara	Pre-Order R7 Di Ofanstore kembali Bisa Diakses pada 6 Juli 2015	1	1.36173
acer	Oto Tek	3	4.08518
achmad	Oto Tek	3	4.08518
ada	Apple Music Bikin iPhone Boros Baterai	1	1.36173
adanya	Juni, Elevenia Klaim Transaksi Capai Rp100 Miliar	1	0.759668
adanya	Layanan StarOne Berakhir Sepenuhnya	1	0.759668
adanya	Indosat Resmi Hentikan Layanan StarOne	1	0.759668
adanya	Ponsel Android BlackBerry Bakal Mirip Galaxy S6 Edge?	2	1.51934

Gambar. 6 Tampilan hasil preprocessing text

No	Threshold	Cluster
1	0.10	0
2	0.10	1
3	0.10	2
4	0.10	3
5	0.10	4

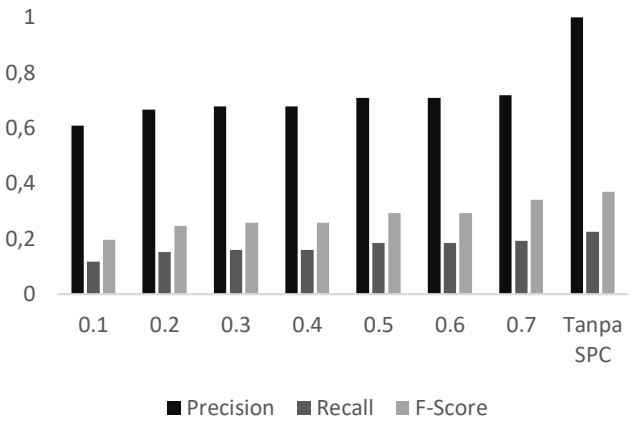
Gambar. 7 Tampilan Hasil Klastering Berita

Gambar. 8 Tampilan Sistem Temu Kembali

Gambar. 9 Tampilan Hasil Temu Kembali Berita

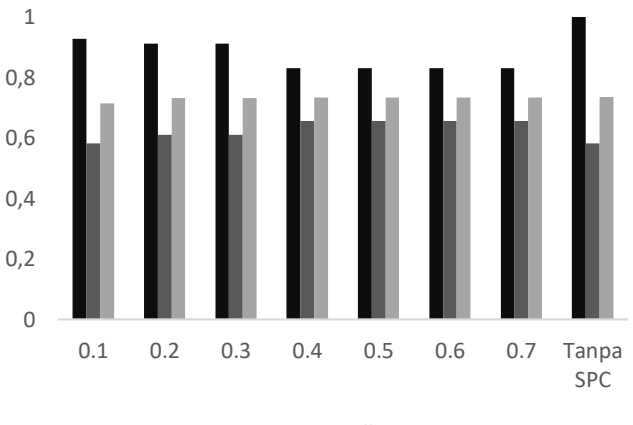
Penggunaan *threshold* pada sistem temu kembali ini bertujuan untuk menentukan *cluster* dengan tingkat kemiripan yang lebih dari *threshold* saja yang akan ditampilkan kepada pengguna. Alasan dipilihnya nilai 0.1 dikarenakan pada nilai ini pengujian *precision*, *recall*, dan *f-score* didapatkan hasil pada semua *cluster*.

- Pada *keyword* 'android'. Dari pengujian yang dilakukan pada *keyword* 'android, didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap klaster hasil *clustering* dengan nilai *threshold* 0,7. Pada pengujian ini *precision* hasilnya sebesar 0,718, pengujian *recall* sebesar 0,193, dan pengujian *f-score* sebesar 0,304. Pengujian pada *threshold* 0,7 inilah yang memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 1, pengujian *recall* sebesar 0,226, dan pengujian *f-score* 0,369.



Gambar. 10 Grafik pengujian *keyword* 'android'

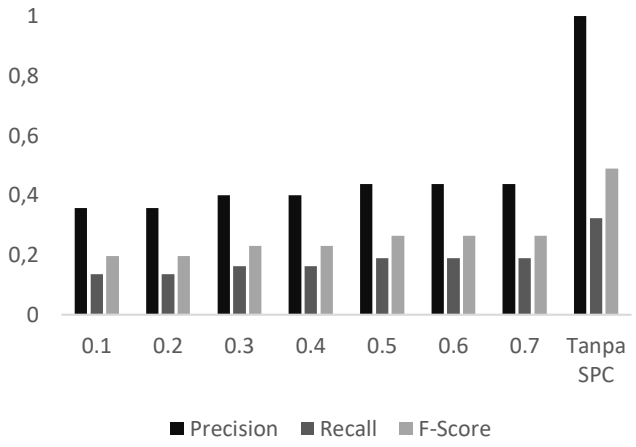
- Pada *keyword* 'iphone'. Dari pengujian yang dilakukan pada *keyword* 'iphone', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap kluster hasil *clustering* dengan nilai *threshold* 0,1. Pada pengujian ini *precision* hasilnya sebesar 0,928, pengujian *recall* sebesar 0,582, dan pengujian *f-score* sebesar 0,715. Pengujian pada *threshold* 0,7 inilah yang memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 1, pengujian *recall* sebesar 0,582, dan pengujian *f-score* 0,735.



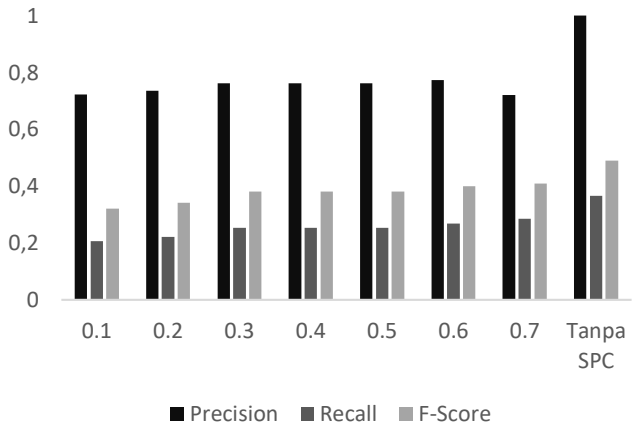
Gambar. 11 Grafik pengujian *keyword* 'iphone'

- Pada *keyword* 'game'. Dari pengujian yang dilakukan pada *keyword* 'game', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap klaster hasil *clustering* dengan nilai *threshold* 0,5, 0,6, dan 0,7. Pada pengujian ini *precision* hasilnya sebesar 0,437, pengujian *recall* sebesar 0,189, dan pengujian *f-score* sebesar 0,264. Pengujian pada *threshold-threshold* tersebut

memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 1, pengujian *recall* sebesar 0,324, dan pengujian *f-score* 0,489.

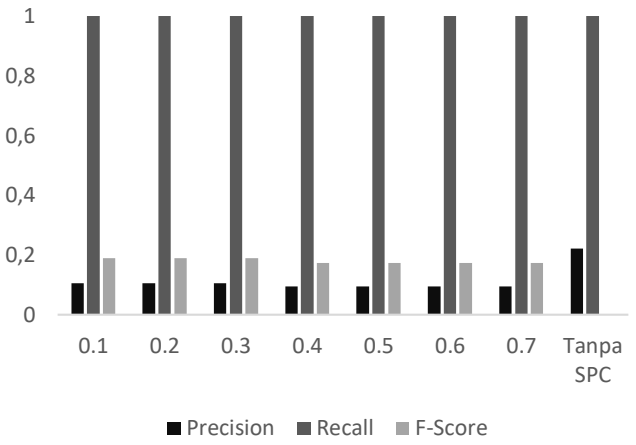


Gambar. 12 Grafik pengujian *keyword* 'game'



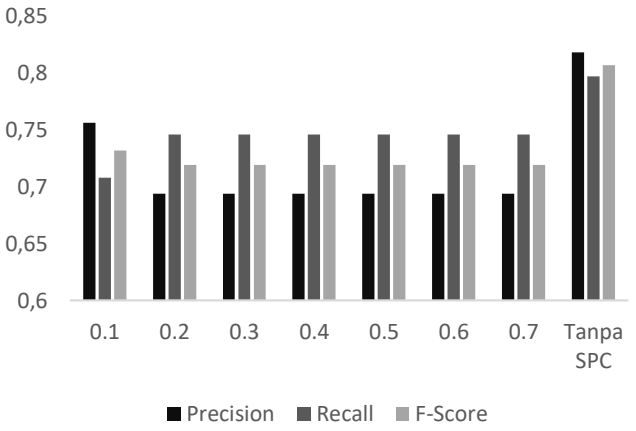
Gambar. 13 Grafik pengujian *keyword* 'facebook'

- Pada *keyword* 'facebook'. Dari pengujian yang dilakukan pada *keyword* 'game', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap klaster hasil *clustering* dengan nilai *threshold* 0,6. Pada pengujian ini *precision* hasilnya sebesar 0,772, pengujian *recall* sebesar 0,269, dan pengujian *f-score* sebesar 0,4. Pengujian pada *threshold* 0,6 memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 1, pengujian *recall* sebesar 0,365, dan pengujian *f-score* 0,535.



Gambar. 14 Grafik pengujian *keyword* 'transaksi online'

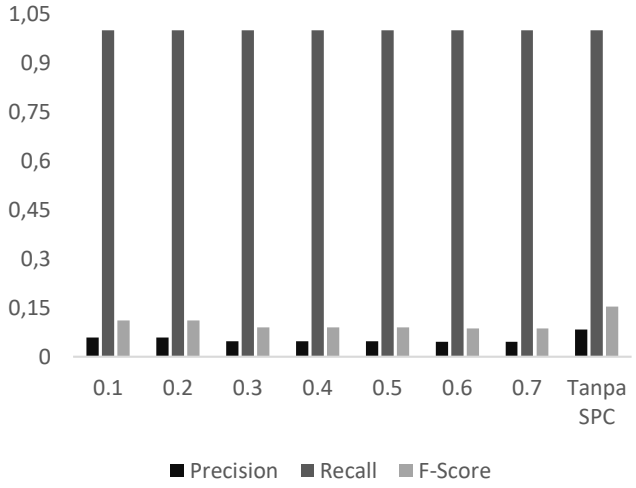
- Pada *keyword* 'transaksi online'. Dari pengujian yang dilakukan pada *keyword* 'transaksi online', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap *cluster* hasil klastering dengan nilai *threshold* 0,1, 0,2, dan 0,3. Pada pengujian ini *precision* hasilnya sebesar 0,105, pengujian *recall* sebesar 1, dan pengujian *f-score* sebesar 0,190. Pengujian pada *threshold-threshold* tersebut memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 0,222, pengujian *recall* sebesar 1, dan pengujian *f-score* 0,363.



Gambar. 15 Grafik pengujian *keyword* '4g lte'

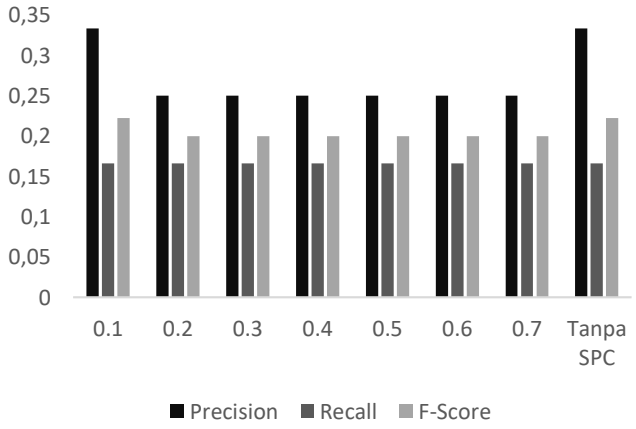
- Pada *keyword* '4g lte'. Dari pengujian yang dilakukan pada *keyword* '4g lte', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap klaster hasil *clustering* dengan nilai *threshold* 0,1. Pada pengujian ini *precision* hasilnya sebesar 0,756, pengujian *recall* sebesar 0,708, dan pengujian *f-score* sebesar 0,732. Pengujian pada *threshold* tersebut memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada

sistem tersebut nilai pengujian *precision* sebesar 0,818, pengujian *recall* sebesar 0,797, dan pengujian *f-score* 0,807.



Gambar. 16 Grafik pengujian *keyword* 'aplikasi whatsapp'

- Pada *keyword* 'aplikasi whatsapp'. Dari pengujian yang dilakukan pada *keyword* 'aplikasi whatsapp', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap klaster hasil *clustering* dengan nilai *threshold* 0,1. Pada pengujian ini *precision* hasilnya sebesar 0,058, pengujian *recall* sebesar 1, dan pengujian *f-score* sebesar 0,111. Pengujian pada *threshold* tersebut memiliki hasil paling mendekati pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 0,083, pengujian *recall* sebesar 1, dan pengujian *f-score* 0,153.



Gambar. 17 Grafik pengujian *keyword* 'virtual reality'

- Pada *keyword* 'virtual reality'. Dari pengujian yang dilakukan pada *keyword* 'virtual reality', didapatkan hasil terbaik ketika proses temu kembali dilakukan terhadap klaster hasil *clustering* dengan nilai *threshold* 0,1. Pada pengujian ini *precision* hasilnya sebesar 0,333, pengujian *recall* sebesar 0,166 dan pengujian *f-score* sebesar 0,222. Pengujian pada *threshold* tersebut memiliki hasil sama dengan pengujian pada sistem yang tidak mengimplementasikan *single pass clustering*. Pada sistem tersebut nilai pengujian *precision* sebesar 0,333, pengujian *recall* sebesar 0,166, dan pengujian *f-score* 0,222.

3) *Pengujian Waktu:* Tabel 3 menunjukkan pengujian waktu eksekusi yang dilakukan pada semua *keyword* yang terdata kepada setiap klaster hasil *clustering* dengan nilai *threshold* yang berbeda-beda. Hasil dari pengujian waktu yang didapatkan menggunakan satuan detik. Dari hasil dua pengujian waktu pada kedua sistem temu kembali yang dibuat menunjukkan jika memang waktu eksekusi yang diperlukan oleh sistem temu kembali yang mengimplementasikan *single pass clustering* memiliki waktu eksekusi yang lebih cepat dari pada sistem temu kembali tanpa *clustering*.

TABEL III
PENGUJIAN WAKTU

k	Waktu							
	Sistem Temu Kembali Dengan *Single Pass Clustering*							Tanpa *SPC*
	0,1	**0,2**	**0,3**	**0,4**	**0,5**	**0,6**	**0,7**	
1	4,3	4,7	4,8	5,1	5,5	5,6	5,9	7,2
2	4,8	5,5	5,9	6	6,4	6,5	6,7	7
3	4,8	5,2	5,8	5,9	6	6,3	6,5	6,6
4	4,8	5,3	5,3	5,7	5,9	6,1	6,4	8,3
5	4,6	5,2	5,4	5,6	5,9	6,1	6,4	7,6
6	4,7	5,2	5,4	5,7	6	6,3	6,7	9,1
7	5	5,7	5,9	6,1	6,4	6,7	7,2	8,5
8	4,9	5,4	5,5	5,8	5,9	6,2	6,5	7,4

IV. KESIMPULAN DAN SARAN

Pengujian dilakukan sebanyak delapan kali terhadap sistem temu kembali pada delapan *keyword* yang berbeda-beda. Untuk hasil pengujian waktu, sistem temu kembali yang mengimplementasikan metode bisa mencari berita lebih cepat dari pada sistem temu kembali yang tidak mengimplemen-tasikan *Single Pass Cl*ustering. Sedangkan pengujian *precision*, *recall*, dan *f-score* terbaik didapatkan ketika proses temu kembali dilakukan pada klaster dengan nilai *threshold* 0.1. Hasil tersebut didapatkan ketika pengujian dilakukan menggunakan *keyword* 'iphone', 'transaksi online', '4g lte', 'aplikasi whatsapp', dan 'virtual reality'. Dari kelima pengujian tersebut hasil terbaik didapatkan pada pengujian dengan *keyword* '4g lte'. Yang mana pada pengujian tersebut nilai pengujian *f-score* sebesar 0,732. Nilai tersebut adalah nilai pengujian *f-score* yang paling baik dari pengujian lainnya. Sedangkan untuk nilai pengujian *precision* sebesar 0,756 dan pengujian *recall* sebesar 0,708.

Untuk penelitian selanjutnya bias dilakukan pengujian sistem temu kembali pada lebih banyak nilai *threshold*. Nilai *threshold* yang bisa digunakan kurang dari 0,1. Nilai *threshold* ini ditujukan untuk memilih seberapa banyak berita yang akan ditampilkan pada pengguna. Selain bisa

juga menggunakan lebih banyak lagi kondisi nilai *threshold* diatas 0,7 atau di bawah 0,1 yang digunakan pada proses klasterisasi *single pass clustering* untuk mendapatkan jumlah klaster berita yang lebih beragam lagi.

REFERENSI

[1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*, 2nd ed. USA: Addison-Wesley Publishing Company, 2008.

[3] J. Zhang, J. Gao, M. Zhou, and J. Wang, "Improving the Effectiveness of Information Retrieval with Clustering and Fusion," in *International Journal of Computational Linguistics & {C}hinese Language Processing, Volume 6, Number 1, {F}ebruary 2001: Special Issue on Natural Language Processing Researches in {MSRA}*, 2001, pp. 109–125.

[4] S. Rieber and U. P. Marathe, "The Single Pass Clustering Method," 1969.

[5] F. Rahutomo, I. F. Rozi, and H. Setiyono, "Implementasi Support Vector Machine pada Analisa Sentimen Twitter Berdasarkan Waktu," *J. TAM (Technology Accept. Model.*, vol. 10, no. 2, pp. 83–88, 2019.

[6] F. Rahutomo, Z. H. R. Adi, I. F. Rozi, and P. Y. Saputra, "Implementasi Text Mining Pada Website/Blog Di Internet Untuk Menilai Kinerja Suatu Organisasi," *Inovtek Polbeng Seri Inform.*, vol. 3, no. 2, pp. 101–109, 2018.

[7] I. Y. R. Pratiwi, R. A. Asmara, and F. Rahutomo, "Study of Hoax News Detection using Naïve Bayes Classifier in Indonesian Language," in *2017 11th International Conference on Information Communication Technology and System (ICTS)*, 2017, pp. 73–78.

[8] F. Rahutomo, I. Y. R. Pratiwi, and D. M. Ramadhani, "Eksperimen Naïve Bayes pada Deteksi Berita Hoax Berbahasa Indonesia," *J. Penelit. Komun. dan Opini Publik*, vol. 23, no. 1, pp. 1–15, 2019.

[9] M. F. Shadiqin Thirafi and F. Rahutomo, "Implementation of Naïve Bayes Classifier Algorithm to Categorize Indonesian Song Lyrics Based on Age," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2018, pp. 106–109.

[10] F. Rahutomo, P. Y. Saputra, and M. A. Fidyawan, "Implementasi Twitter Sentiment Analysis untuk Review Film Menggunakan Algoritma Support Vector Machine," *J. Inform. Polinema*, vol. 4, no. 2, p. 93, 2018.

[11] E. Hardiyanto and F. Rahutomo, "Studi Awal Klasifikasi Artikel Wikipedia Bahasa Indonesia Dengan Menggunakan Metoda K Nearest Neighbor," *Pros. Sentrinov (Seminar Nas. Terap. Ris. Inov.*, vol. 2, no. 1, pp. 158–165, 2016.

[12] F. Rahutomo and E. Rohadi, "Pengembangan Piranti Penelitian Sistem Temu Kembali Informasi Bahasa Indonesia," in *Seminar Nasional Sistem Informasi Indonesia (SESINDO)*, 2015, pp. 313–319.

[13] A. Muzad and F. Rahutomo, "Korpus Berita Daring Bahasa Indonesia Dengan Depth First Focused Crawling," *Pros. Sentrinov (Seminar Nas. Terap. Ris. Inov.*, vol. 2, no. 1, pp. 11–20, 2016.

[14] F. Rahutomo and A. R. T. H. Ririd, "Evaluasi Daftar Stopword Bahasa Indonesia," *J. Teknol. Inform. dan Ilmu Komput.*, vol. 6, no. 1, pp. 41–48, 2019.

[15] P. Kharismadita and F. Rahutomo, "Implementasi Tokenizing Plus pada Sistem Pendeteksi Kemiripan Jurnal Skripsi," *J. Inform. Polinema*, vol. 2, no. 1, p. 24, 2017.

[16] D. Zamzami, F. Rahutomo, and D. Puspitasari, "Aplikasi Wordnet Indonesia Berdasarkan Kamus Thesaurus Bahasa Indonesia menggunakan Algoritma Rule Based Text Parsing," in *Seminar Informatika Aplikatif Polinema*, 2016.

[17] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Aug. 1988.