Tesis - KI142502

SELEKSI FITUR PADA PENGELOMPOKAN DOKUMEN DENGAN *RANDOM PROJECTION - GRAM SCHMIDT ORTHOGONALIZATION* DAN ALGORITMA *HARMONY SEARCH*

Muhammad Machmud
5114201035

DOSEN PEMBIMBING
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
Diana Purwitasari, S.Kom., M.Sc

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUAL
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

2016

Theses - KI142502

FEATURE SELECTION IN DOCUMENT CLUSTERING USING RANDOM PROJECTION – GRAM SCHMIDT ORTHOGONALIZATION AND HARMONY SEARCH ALGORITHM

Muhammad Machmud
5114201035

SUPERVISORS
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
Diana Purwitasari, S.Kom., M.Sc

MAGISTER PROGRAMME
FIELD OF EXPERTISE INTELLIGENT COMPUTATION AND VISUALISATION
DEPARTMENT OF INFORMATICS ENGINEERING
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

LEMBAR PENGESAHAN

TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember

oleh :
Muhammad Machmud
Nrp. 5114201035

Tanggal Ujian : 15 September 2016
Periode Wisuda : Gasal 2016

Disetujui oleh:

1. **Dr.Eng.Chastine Fatichah,S.Kom., M.Kom** .....
   NIP: 197512202001122002 **(Pembimbing I)**

2. **Diana Purwitasari, S.Kom., M.Sc** .....
   NIP: 197804102003122001 **(Pembimbing II)**

3. **Prof.Dr.Ir.Joko Lianto Buliali, M.Sc** .....
   NIP: 196707271992031002 **(Penguji)**

4. **Dr. Darlis Heru Murti, S.Kom, M.Kom** .....
   NIP: 197712172003121001 **(Penguji)**

5. **Dini Adni Navastara, S.Kom, M.Sc** .....
   NIP: 198510172015042001 **(Penguji)**

Direktur Program Pascasarjana,

Prof.Ir. Djauhar Manfaat, M.Sc, Ph.D
NIP. 196012021987011001

[*Halaman ini sengaja dikosongkan*]

SELEKSI FITUR PADA PENGELOMPOKAN DOKUMEN DENGAN *RANDOM PROJECTION – GRAM SCHMIDT ORTHOGONALIZATION* DAN ALGORITMA *HARMONY SEARCH*

Nama Mahasiswa	:	Muhammad Machmud
NRP	:	5114201035
Pembimbing 1	:	Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
Pembimbing 2	:	Diana Purwitasari, S.Kom., M.Sc.

ABSTRAK

Proses pengelompokan dokumen sangat tergantung pada keberadaan fitur kata tiap dokumen dan kemiripan antar fitur kata tersebut. Fitur kata pada suatu dokumen terkadang merupakan fitur *noise*, *redundant*, maupun fitur kata yang tidak relevan sehingga menyebabkan hasil akhir proses pengelompokan dokumen menjadi bias. Selain itu, pengelompokan dokumen dengan metode klasik kurang bisa menghasilkan kelompok dokumen yang mampu merepresentasikan kemiripan isi pada tiap-tiap kelompok dokumen.

Pada penelitian ini diusulkan seleksi fitur pada pengelompokan dokumen dengan *Random Projection Gram Schmidt Orthogonalization* (RPGSO) dan Algoritma *Harmony Search* (HS). Dengan metode RPGSO akan didapatkan tingkat kepentingan tiap-tiap fitur kata untuk semua dokumen. Pada algoritma HS, dilakukan proses pengelompokan dokumen berdasarkan urutan fitur-fitur kata dari RPGSO dengan *fitness function* berupa *Average Distance of Documents to the cluster Centroid* (ADDC). Untuk mendapatkan kelompok dokumen dengan kriteria evaluasi yang paling baik, proses pengelompokan dokumen dengan algoritma HS ini diiterasi untuk jumlah fitur yang berbeda sesuai urutan yang dihasilkan dari proses RP-GSO.

Uji coba dilaksanakan terhadap tiga buah dataset dokumen berita dengan evaluasi menggunakan kriteria F-Measure. Berdasarkan uji coba tersebut, metode usulan mampu menghasilkan kelompok dokumen dengan rata-rata F-Measure lebih tinggi 9.50% dibandingkan dengan menggunakan seluruh fitur. Uji coba juga menunjukkan bahwa kelompok dokumen yang dihasilkan dari metode usulan memiliki rata-rata F-Measure lebih tinggi 8.40% dibandingkan K-Means yang menggunakan *Cosine Similarity*, dan jika dibandingkan dengan K-Means yang menggunakan *Euclidean Distance*, metode usulan mampu menghasilkan kelompok dokumen dengan rata-rata F-Measure lebih tinggi sampai 120.05%.

Kata kunci: Seleksi Fitur, Pengelompokan Dokumen, *Random Projection Gram Schmidt Orthogonalization*, *Harmony Search*

[Halaman ini sengaja dikosongkan]

FEATURE SELECTION IN DOCUMENT CLUSTERING USING RANDOM PROJECTION – GRAM SCHMIDT ORTHOGONALIZATION AND HARMONY SEARCH ALGORITHM

Student Name	:	Muhammad Machmud
NRP	:	5114201035
Supervisor	:	Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
Co Supervisor	:	Diana Purwitasari, S.Kom., M.Sc.

ABSTRACT

Document clustering procceses are depends on the quality of its term features and the similarity between those features. Sometimes the features of the document is a noise, redundant, or irrelevant and its cause the result of document clustering is bias. Furthermore, classical clustering method was unable to generate clusters of documents that is represent the similarity of its contents.

In this study, we propose feature selection in document clustering using Random Projection Gram Schmidt Orthogonalization (RPGSO) and Harmony Search (HS) algorithm. With using RPGSO methods we will obtain the rank of term features of all documents. Then, we cluster the document based on the rank of features using HS algorithm with fitness function is *Average Distance of Documents to the cluster Centroid (ADDC)*. To produce the clusters of documents which the best evaluation criteria, the clustering algorithm will be iterated for different number of features based on RPGSO rank.

The methode has been tested to three datasets of news documents with F-Measure as evaluation criteria,. Based on the testing result, the proposed method generates clusters of documents with average of F-Measure criteria that 9.50% higher than use all features of documents in datasets.The testing result also shown that the proposed method generate clusters of documents with average of F-Measure criteria that 8.40% higher than K-Means method with *Cosine Similarity* and 120.05% higher than K-Means method with *Euclidean Distance*.

Keywords : Feature Selection, Document Clustering, Random Projection Gram Schmidt Orthogonalization, Harmony Search

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Alhamdulilahi robbil 'alamiin, segala puji bagi Allah SWT atas rahmatNya penulis dapat menyelesaikan Tesis ini meski prosesnya melebihi batas waktu ketentuan beasiswa BPPDN. Sholawat dan salam kepada Rasulullah SAW atas segala tauladan yang Beliau berikan sehingga hidup kita lebih bermakna.

Pada kesempatan ini penulis ingin mengucapkan banyak terimakasih kepada :

1. Kedua Orang tua yang telah bekerja keras membentuk putra-putrinya sehingga menjadi generasi emas yang *sholeh* dan *birrul walidain*.
2. Kakak-kakak yang telah memberikan bantuan moral dan material serta contoh dan bimbingan sehingga penulis lebih mengerti makna hidup yang sebenarnya.
3. Bunda Ruro, Kak Naila dan Adek Naura yang menjadi cahaya kehidupan dan sumber kebahagiaan penulis.
4. Ibu Dr. Eng. Chastine F., S.Kom, M.Kom selaku dosen pembimbing 1 dan dosen wali selama pengerjaan tesis, yang telah bersedia membimbing dan meluangkan waktu, memberikan saran dan kritik serta berbagai pengalaman hidup atau pun dalam bidang pendidikan dan penelitian.
5. Ibu Diana Purwitasari S.Kom, M.Sc selaku dosen pembimbing 2 yang telah bersedia meluangkan waktu, membimbing dengan penuh kesabaran, serta memberikan saran dan kritik sehingga tesis ini dapat diselesaikan.
6. Tim Penguji Proposal dan Tesis, Bapak Prof.Dr.Ir.Joko Lianto Buliali, M.Sc, Bapak Dr. Darlis Heru Murti, S.Kom, M.Kom, Ibu Dini Adni Navastara, S.Kom, M.Sc, dan Ibu Wijayanti Nurul Khotimah, S.Kom.,M.Sc. atas segala masukan dan kritik yang membangun untuk Tesis ini.
7. Bapak Washkito Wibisono, S.Kom, M.Eng, Ph.D selaku Ketua Program Studi Pascasarjana Informatika ITS dan dosen wali selama 2 tahun pertama, atas segala nasihat selama penulis menempuh pendidikan S2.
8. Koordinator Kopertis Wilayah 7 Bapak Prof. Dr. Ir. Suprapro, DEA (ITS), Sekretaris Pelaksana Kopertis Wilayah 7 Bapak Prof. Dr. Ali Maksum (UNESA), Kabag Kelembagaan dan Sistem Informasi Bapak Drs. Ec. Purwo Beksi, Msi, Kasie Sistem Informasi Bapak Drs. Supradono, MM, Kabag Umum Bapak (Calon Dr) Sulaksono, SH, MH, Kasie Kelembagaan Bapak Drs. Budi Hasan, SH, Msi, Kasubbag

Kepegawaian Ibu Drs. Ec. Indratiningsih, MM, dan Bapak/Ibu pimpinan yang lain, atas bantuan dan dukungan Beliau semua sehingga penulis dapat melaksanakan studi lanjut BPPDN.

9. Rekan-rekan Seksi Sistem Informasi dan subbag/ seksi lain di Kopertis 7, Pak Yono, Mas Tohari, Mas Indera, Mas Dhani, Pak Cahyono, Mbak Cindi, Mbak Vita, Bu Sutipak, Bu Puji, Bu Rina, Mas Agung Yundi, Mbak Airin, Mbak Rahma, Mas Rossi, dan rekan-rekan yang lain, Terima kasih atas supportnya.
10. Teman-teman S2 Teknik Informatika angkatan 2014, Mas Indera Zainul M, Mas Indra Gita, Mas Hanif Affandi, Mas Agri, Mas Farid, Mas Rifial, Mbak Hani, Mas Christian, Mbak Ratri, yang telah memberikan semangat dan segala bantuan lain. Serta rekan-rekan mahasiswa yang memberikan motivasi penulis untuk selalu belajar dan mencoba lebih demi pendidikan yang lebih maju.
11. Bapak dan Ibu administasi sarjana, pasca informatika, dan fakultas, Pak Yudi, Pak Soleh, Pak Pri, Mas Gayuh, Mas Murdiono, Mbak Lina, Mbak Rini, Bu Eva RBTC, dan Bapak/Ibu lain yang tidak bisa saya sebut satu-persatu, atas bantuan informasi dan administrasi yang berkaitan dengan perkuliahan dan Tesis selama menempuh pendidikan S2.
12. Mas Kunto atas kesediaan memberikan waktunya untuk menjaga Laboraturium Pascasarjana Informatika ITS sebagai tempat belajar dan diksusi mengenai segala hal yang berhubungan dengan pendidikan S2.
13. Teman-teman lain yang tidak bisa disebutkan satu-persatu atas dukungan doa serta kesediaanya untuk mendengarkan dan memberikan masukan kepada penulis.

Dengan segala keterbatasan, penulis menyadari bahwa tesis ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan saran atau kritik yang membangun dari para pembaca.

Akhir kata, penulis berharap semoga penelitian ini memberikan manfaat bagi berbagai pihak terutama bagi pengembangan ilmu pengetahuan dan teknologi di bidang Komputasi Cerdas dan Visualisasi

Surabaya, November 2016

Muhammad Machmud

DAFTAR ISI

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Clustering dokumen merupakan proses pengelompokan dokumen yang bertujuan memaksimalkan kemiripan antar dokumen pada *cluster* yang sama dan meminimalkan kemiripan antar dokumen pada *cluster* yang berbeda. Pada proses ini, kemiripan antar fitur kata pada tiap dokumen sangat menentukan hasil akhir proses pengelompokan dokumen. Semakin tinggi kemiripan antar kata pada beberapa dokumen, maka kecenderungan dokumen-dokumen tersebut untuk membentuk *cluster* yang sama semakin besar.

Keberadaan fitur kata pada suatu dokumen terkadang menyebabkan proses pengenalan dokumen menjadi bias (Guyon & Elisseeff, 2003). Fitur-fitur kata ini disebut sebagai *noisy features*, yakni fitur kata yang memiliki kecenderungan untuk membentuk kelompok yang berbeda dengan kata-kata lain pada dokumen tersebut (Kohavi & John, 1997) (Blum & Langley, 1997). Ada kalanya beberapa kata pada sebuah dokumen bersifat *redundant*, yakni beberapa kata menyampaikan hal yang sama sehingga keberadaan kata tersebut bisa saling menggantikan (Hall, 1999). Beberapa kata yang lain kurang relevan dengan pembentukan *cluster* dokumen yang ada (Kohavi & John, 1997) (Blum & Langley, 1997).

Untuk mengatasi *noisy features*, *redundant features*, dan *irrelevant features*, terdapat dua pendekatan reduksi dimensi yang bisa digunakan, yakni ekstraksi fitur dan seleksi fitur (Fodor, 2002). Ekstraksi fitur merupakan metode reduksi dimensi yang akan mentransformasikan ruang dimensi fitur asli ke dalam representasi dan ruang dimensi lain yang lebih kecil. Diantara metode ekstraksi fitur yang banyak diimplementasikan adalah metode *Principal Component Analysis* (Jolliffe, 1986) dan *Independent Component Analysis* (Hyvarinen, 1999).

Berbeda dengan ekstraksi fitur yang akan merubah fitur menjadi representasi lain, metode seleksi fitur tetap menggunakan fitur asal dengan memanfaatkan perhitungan tertentu untuk menentukan fitur mana yang paling sesuai. Dengan metode ini, representasi fitur asal tidak akan berubah sehingga hal

ini akan memudahkan proses kombinasi beberapa metode seleksi fitur maupun kombinasi dengan metode lain.

Berdasarkan metode penilaiannya, pendekatan dalam seleksi fitur dapat dikategorikan dalam tiga kelompok, yakni metode *wrapper*, metode *filter*, dan metode *hybrid* (Guyon & Elisseeff, 2003). Pada pendekatan *wrapper*, terlebih dahulu ditentukan sebuah model pembelajaran, kemudian fitur-fitur yang ada dipilih apabila memenuhi model pembelajaran yang ditentukan berdasarkan kriteria tertentu (Kohavi & John, 1997). Algoritma *wrapper* akan membutuhkan waktu yang lebih lama namun akurasi yang didapatkan lebih besar daripada pendekatan *filter*.

Dalam pendekatan *filter*, semua fitur dievaluasi misalnya dengan pendekatan statistik kemudian fitur terpilih ditentukan berdasarkan kriteria evaluasi yang telah dibuat (Chandrashekar & Sahin, 2014). Beberapa metode dengan pendekatan *filter* adalah *chi-square test*, pendekatan korelasi, *Wilcoxon Mann–Whitney test*, *t-test*, *mutual information* (MI), *normalized mutual information* (NMI). Pendekatan *filter* membutuhkan waktu yang lebih cepat dibandingkan metode *wrapper* dan fitur yang dihasilkan bersifat umum (Chandrashekar & Sahin, 2014).

Salah satu metode filter yang dapat diimplementasikan untuk seleksi fitur adalah *Random Projection Gram Schmidt Orthogonalization* (RPGSO) (Wang, Zhang, Liu, Liu, & Wang, 2016). Pendekatan ini dapat digunakan untuk memperoleh fitur kata yang paling penting pada sebuah dokumen. Pendekatan ini menggunakan prinsip orthogonalitas, bahwa sebuah vektor dapat dinyatakan sebagai kombinasi vektor-vektor lain yang saling orthogonal/tegak lurus. Pada sudut pandang dokumen, suatu dokumen dapat dinyatakan sebagai susunan dari beberapa fitur kata penting yang membentuk dokumen, sedangkan fitur kata-kata yang lain merupakan kombinasi dari fitur kata-kata penting tersebut.

Diantara kelebihan pendekatan RP-GSO ini adalah kemampuan untuk mendapatkan kata-kata penting pada sebuah dokumen yang memenuhi kriteria bukan fitur *noise* dan bukan fitur *irrelevant,* serta menghilangkan redundansi pada fitur kata (Wang, Zhang, Liu, Liu, & Wang, 2016).

Beberapa penelitian tentang metode *hybrid* yang menggabungkan metode *wrapper* dengan basis metode *filter* menjadi salah satu topik yang banyak dibahas. Idenya adalah menggunakan algoritma *filter* yang kemudian outputnya menjadi dasar pada algoritma *wrapper*. Dengan cara ini maka keuntungan dari pendekatan *wrapper* tetap dipertahankan sementara jumlah fitur evaluasi *wrapper* berkurang secara bertahap setelah diseleksi dengan metode *filter* (Unler, Murat, & Chinnam, 2011). Beberapa penelitian dengan pendekatan ini antara lain *ant colony optimization with mutual information* (Zhang & Hu, 2005), *mutual information and genetic algorithm* (Huang, Cai, & Xu, 2006), *ant colony optimization and Chi-square statistics with support vector machine* (Mesleh & Kanaan, 2008), dan *feature selection based on mutual information and particle swarm optimization with support vector machine* (Unler, Murat, & Chinnam, 2011).

Salah satu algoritma yang dapat digunakan pada pendekatan *hybrid* adalah algoritma *Harmony Search*. *Harmony Search* (HS) merupakan algoritma optimisasi yang banyak digunakan dalam proses penyelesaian permasalahan optimisasi (Lee & Geem, 2005). Diantara kelebihan algoritma ini adalah jumlah parameter yang dibutuhkan hanya sedikit, kemampuan memperbaiki solusi lama dan mencari solusi baru yang baik, serta fleksibilitas dibandingkan dengan proses optimisasi lainnya (Alia & Mandava, 2011). Algoritma ini juga dapat diimplementasikan pada persoalan pengelompokan dokumen sehingga menghasilkan *cluster* dokumen yang memiliki kriteria *global optimal* dan *local optimal* (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013).

Berdasarkan latar belakang yang telah dijelaskan diatas, maka penelitian ini mengusulkan metode baru yakni SELEKSI FITUR PADA PENGELOMPOKAN DOKUMEN DENGAN *RANDOM PROJECTION – GRAM SCHMIDT ORTHOGONALIZATION* DAN ALGORITMA *HARMONY SEARCH*. Dengan kombinasi *Random Projection – Gram Schmidt Orthogonalization* (RP-GSO) dan algoritma *Harmony Search* (HS) tersebut diharapkan dapat menyeleksi fitur pada dokumen berbahasa indonesia sehingga menghasilkan *cluster* dokumen dengan kriteria evaluasi yang lebih baik.

1.2 Perumusan Masalah

Berdasarkan uraian yang telah dijelaskan pada latar belakang, maka dirumuskan permasalahan sebagai berikut:

1. Bagaimana menentukan fitur penting berdasarkan hasil *Random Projection Gram Schmidt Orthogonalization* pada proses pengelompokan dokumen dengan Algoritma *Harmony Search*?
2. Bagaimana mengevaluasi hasil kombinasi *Random Projection-Gram Schmidt Orthogonalization* dan Algoritma *Harmony Search* pada proses pengelompokan dokumen?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini bahwa dataset yang digunakan adalah dokumen berita berbahasa Indonesia yang berasal dari situs berita kompas[1].

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk melakukan seleksi fitur dengan *Random Projection - Gram Schmidt Orthogonalization* dan Algoritma *Harmony Search* pada proses pengelompokan dokumen.

1.5 Kontribusi Penelitian

Kontribusi dari penelitian ini adalah mengusulkan *Random Projection - Gram Schmidt Orthogonalization* dan Algoritma *Harmony Search* untuk melakukan seleksi fitur pada pengelompokan dokumen.

1.6 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini adalah menghasilkan pendekatan kombinasi dalam seleksi fitur pada pengelompokan dokumen dan diharapkan metode yang dihasilkan dapat menjadi acuan bagi penelitian selanjutnya.

[1] www.kompas.com

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

Pada bagian kajian pustaka dan dasar teori ini akan dipaparkan konsep-konsep yang dipakai sebagai bahan acuan dalam melakukan penelitian. Adapun pemaparan teori tersebut dibagi menjadi beberapa sub bab yang meliputi konsep representasi dokumen, pengukuran kemiripan dokumen, *clustering*, *random projection -gram schmidt orthogonalization* dan algoritma *harmony search.*

2.1 Representasi Dokumen

Dalam model ruang vektor, misalkan dokumen j diwakili oleh vektor bobot kata ke n seperti representasi berikut:

$$D_j = (w_1, w_2, \dots, w_n) \tag{2.1}$$

dengan w_i adalah bobot dari fitur kata ke-i dan n adalah jumlah total fitur kata yang unik. Skema pembobotan yang paling banyak digunakan adalah kombinasi dari *Term Frequency - Invers Document Frequency* (TF-IDF) (Everitt, 1980) (Salton G. , 1989), yang dapat dihitung dengan rumus berikut (Salton & Buckley, 1988):

$$TF_{d,t} = f(d,t) \tag{2.2}$$

$$IDF_{d,t} = \log\left(1 + \frac{N}{N_{d,t}}\right) \tag{2.3}$$

$$W_{d,t} = TF_{d,t} \times IDF_{d,t} \tag{2.4}$$

dengan $TF_{d,t}$ adalah *term frequency,* yaitu jumlah keberadaan kata t dalam dokumen d, dan $IDF_{d,t}$ adalah *invers document frequency*, dengan N adalah jumlah dokumen dalam seluruh koleksi, dan $N_{d,t}$ adalah jumlah dokumen yang memiliki kata t. $IDF_{d,t}$ berkaitan dengan keberadaan sebuah kata dalam dokumen, semakin banyak dokumen yang memuat kata tersebut, maka bobotnya semakin kecil. Sebaliknya semakin sedikit dokumen yang memuat kata tersebut, maka bobotnya semakin besar.

2.2 Pengukuran Kemiripan Dokumen

Dalam proses pengelompokan dokumen, hal yang paling menentukan proses ini adalah kemiripan antar dokumen. Ada beberapa metode yang banyak digunakan untuk menghitung kemiripan antara dua dokumen D_1 dan D_2.

Metode pertama berdasarkan jarak Euclidean, misalkan terdapat pada dokumen, $D_1 = (w_{11}, w_{12}, \dots, w_{1n})$ dan $D_2 = (w_{21}, w_{22}, \dots, w_{2n})$, jarak Euclidean kedua dokumen didefinisikan sebagai:

$$Distance(D_1, D_2) = \left(\sum_{i=1}^{n} (w_{1i} - w_{2i})^2 \right)^{\frac{1}{2}} \quad (2.5)$$

Dengan w_{jn} menyatakan bobot term ke-n pada dokumen ke-j.

Metode kedua berdasarkan jarak Manhattan, yang didefinisikan sesuai Persamaan berikut:

$$Distance(D_1, D_2) = \left(\sum_{i=1}^{n} |w_{1i} - w_{2i}| \right) \quad (2.6)$$

Metode selanjutnya adalah jarak Minkowski (Boley, 1999). Jarak Minkowski merupakan bentuk umum dari jarak Euclidean dan Manhattan. Jarak Minkowski antara dua dokumen D_1 dan D_2 didefinisikan sebagai:

$$Distance_p(D_1, D_2) = \left(\sum_{i=1}^{n} |w_{1i} - w_{2i}|^p \right)^{\frac{1}{p}} \quad (2.7)$$

Dengan w_{jn} menyatakan bobot term ke-n pada dokumen ke-j. Persamaan 2.7 tersebut merupakan jarak Euclidean (Persamaan 2.5) untuk nilai $p = 2$, dan merupakan jarak Manhattan (Persamaan 2.6) untuk nilai $p = 1$.

Metode lain yang umum digunakan untuk menghitung kemiripan antara dua dokumen pada proses pengelompokan dokumen adalah *Cosine similarity* (Salton G. , 1989) (Erkan & Randev, 2004), yang didefinisikan sebagai:

$$cos(D_1, D_2) = \frac{D_1 . D_2}{\|D_1\| \times \|D_2\|} \quad (2.8)$$

Dengan . menunjukkan perkalian titik antara dua dokumen, dan ‖ ‖ menunjukkan panjang dokumen. Pesamaan 2.8 ini akan menghasilkan nilai kemiripan 1 jika kedua dokumen identik dan akan menghasilkan nilai kemiripan nol jika kedua dokumen tidak memiliki kemiripan sama sekali. Menghitung kemiripan antara dua dokumen yang tegak lurus (*orthogonal*) satu sama lain juga akan menghasilkan ke nilai kemiripan nol.

Metode penghitungan jarak pada Persaman 2.5 sampai 2.8 tersebut banyak diimplementasikan pada proses pengelompokan dokumen. Namun, pada kasus di mana jumlah dimensi dari dua dokumen sangat berbeda, rumus c*osine similarity* yang lebih tepat digunakan. Sebaliknya, di mana dua dokumen memiliki dimensi yang hampir sama, jarak Minkowski yang lebih tepat digunakan.

2.3 *Clustering*

Clustering dokumen mempunyai peran penting dalam bidang *data mining* dan *Information Retrieval* (IR). *Clustering* berfungsi untuk mengelompokkan beberapa dokumen menjadi *cluster* dokumen sesuai dengan kemiripannya. *Cosine similarity* adalah metode pengukuran yang sering digunakan pada proses *clustering* (Erkan & Randev, 2004). Secara umum, metode *clustering* terbagi menjadi lima kelompok besar yaitu : *hierarchical, partitioning, density-based, grid-based,* dan *model-based* (Han & Kamber, 2011).

Hierarchical clustering merupakan pengelompokan data dimana sebuah data dapat menjadi anggota dari satu atau lebih *cluster* secara hirarkis berdasarkan nilai kemiripan tertentu. *Partitioning clustering* merupakan pengelompokan data secara ketat sehingga membagi data ke dalam *cluster-cluster* tertentu berdasarkan nilai kemiripan data. *Density-based clustering* merupakan pengelompokan data berdasarkan kepadatan daerah sekitar data tersebut. Pada metode ini selama data-data yang ada saling terhubung sesuai batas tertentu, maka data-data tersebut termasuk dalam satu *cluster*, meski jarak antara data pada ujung-ujung *cluster* saling berjauhan. *Grid-based clustering* merupakan pengelompokan data ke dalam *cluster-cluster* tertentu sehingga membentuk struktul sel yang saling terkait. *Grid-based clustering* menggunakan pendekatan *spatial* yang cukup efektif, sehingga

seringkali dikombinasikan dengan metode *clustering* lain, seperti *hierarchical* maupun d*ensity-based clustering* (Han & Kamber, 2011).

2.4 *Random Projection and Gram Schmidt Orthogonalization*

Random Projection – Gram Schmidt Orthogonalization (RP-GSO) (Wang, Zhang, Liu, Liu, & Wang, 2016) merupakan metode yang dapat digunakan untuk mendapatkan tingkat kepentingan fitur kata pada sebuah dokumen. Pendekatan ini menggunakan prinsip orthogonalitas, bahwa sebuah vektor dapat dinyatakan sebagai kombinasi vektor-vektor lain yang saling orthogonal.

Pada sudut pandang dokumen, suatu dokumen dapat dinyatakan sebagai susunan dari beberapa fitur kata penting yang membentuk dokumen, sedangkan fitur kata-kata yang lain merupakan kombinasi dari fitur kata-kata penting tersebut. Dengan pendekatan RP-GSO ini akan didapatkan urutan fitur kata-kata penting pada sebuah dokumen yang memenuhi kriteria bukan fitur *noise,* bukan fitur *redundant,* dan bukan fitur yang tidak *relevant* (Wang, Zhang, Liu, Liu, & Wang, 2016).

Prinsip utama dari pendekatan RP-GSO ini adalah proses ortogonalisasi dengan metode *Gram Schmidt* (Golub & Van Loan, 1989) (Chen, Billings, & Luo, 1989). Ortogonalisasi dengan metode *Gram Schmidt* merupakan proyeksi vektor v terhadap vektor u yang dirumuskan sebagai:

$$proj_u(v) = \frac{\langle u, v \rangle}{\langle u, u \rangle} u \tag{2.9}$$

Dimana $\langle u, v \rangle$ merupakan perkalian titik antara vektor u dan vektor v. Proses ini akan memproyeksikan vektor v secara ortogonal terhadap garis yang direntang oleh vektor u. Untuk n buah vektor, $v_1, v_2, \dots\dots, v_n$, akan didapatkan urutan vektor ortogonal $u_1, u_2, \dots\dots, u_n$, sesuai Persamaan berikut:

$$u_n = v_n - \sum_{i=1}^{n-1} proj_{u_i}(v_n) \tag{2.10}$$

Metode RP-GSO ini menggunakan teknik proyeksi acak (*random projections*) sebagai salah satu tahapannya. Teknik proyeksi acak ini dapat

dilakukan tanpa merubah karakter asal fitur kata-kata penting tersebut. Teknik ini menggunakan transformasi *Johnson and Lindenstrauss* yang bertujuan agar proses pembentukan matriks ortogonal berjalan lebih cepat sehingga meminimalisir sumber daya yang dibutuhkan (Achlioptas, 2003).

2.5 Algoritma *Harmony Search*

Algoritma *Harmony Search* (HS) merupakan algoritma optimisasi yang banyak digunakan dalam proses penyelesaian permasalahan optimisasi (Lee & Geem, 2005). Ide utama algoritma ini berasal dari ide improvisasi musik terhadap instrumen untuk mendapatkan nada yang harmoni. Algoritma ini memiliki kelebihan dan fleksibilitas dibandingkan dengan proses optimisasi tradisional lainnya

Langkah pertama penyelesaian masalah dengan algoritma HSC sesuai Gambar 2.1 adalah setting parameter HS yang terdiri dari *Uppper Bound* (UB) dan *Lower Bound* (LB), *Harmony Memory* (HM), *Harmony Memory Size* (HMS), *Harmony Memory Considering Rate* (HMCR), *Pitch Adjusting Rate* (PAR), serta *Number of Iterations* (NI) dan *Bandwidth*.

Harmony Memory (HM) identik dengan populasi pada Algoritma Genetika, merupakan kumpulan dari vektor solusi yang merepresentasikan permasalahan yang sedang diselesaikan. *Harmony Memory Size* (HMS) menunjukkan jumlah alternatif solusi yang mungkin diambil pada saat proses penyelesaian permasalahan. *Harmony Memory Considering Rate* (HMCR) merupakan probabilitas nilai sebagai acuan apakah vektor solusi yang baru, yakni *New Harmony Vector* (NHV), diambil dari HMS yang sudah ada ataukah NHV didapatkan dengan membuat nilai acak baru. *Pitch Adjusting Rate* (PAR) merupakan probabilitas untuk menentukan cara pemilihan NHV dari HMS yang sudah ada. *Number of Iterations* (NI) menyatakan banyaknya iterasi yang dilakukan untuk membentuk kelompok dokumen.

Input: *D*, yakni dataset.

1. Setting parameter : *UB, LB, HMS, HMCR, PAR, NI, Bandwidth*.
2. Menentukan *fitness function*.
   f(x) = fitness
3. Membangkitkan harmony inisial awal
   $x_i, LB \leq x_i \leq UB$
4. Iterasi Harmony Search

```
bestfit=0;
bestharm=0;
for i=1 to NI
    randval=rand()
    if(randval<=HMCR)
        harmony baru = pilih salah satu harmony inisial
        if(randval<=PAR)
            harmony baru = harmony inisial dengan penyesuaian
        endif
    else
        harmony baru = bangkitkan harmony, x_i = x_(i-1) + randval * (LB - UB)
    endif
    fitness = hitung fitness harmony baru
    if(fitness lebih baik drpd bestfit)
        bestfit=fitness
        bestharm= harmony baru
    endif
endfor
```

Gambar 2.1 Algoritma *Harmony Search*

Setelah setting parameter, langkah selanjutnya sesuai Gambar 2.1 adalah menentukan nilai *fitness* sebagai evaluasi hasil yang dihasilkan dari proses pembentukan vektor solusi baru.

Sesuai Gambar 2.1, langkah selanjutnya adalah membangkitkan *harmony* awal sesuai dengan rentang batas bawah LB dan batas UB yang telah ditentukan, yakni antara 1 sampai 4. Persamaan untuk membangkitkan harmony baru adalah:

$$x_i = LB + rand * (UB - LB) \quad (2.11)$$

Selanjutnya adalah proses iterasi penyelesaian masalah dengan algortima HS. Pada langkah pembentukan vektor solusi baru, yaitu *New Harmony Vector* (NHV), dihasilkan dari vektor solusi yang tersimpan di HM. Vektor harmoni baru yang dihasilkan harus sebanyak mungkin mewarisi informasi dari vektor solusi di HM.

Langkah pemilihan vektor solusi baru berdasarkan dua kemungkinan, dengan probabilitas sebesar HMCR untuk pemilihan vektor solusi baru berdasarkan vektor solusi yang telah ada pada HM dan dengan probabilitas 1-HMRC untuk pemilihan vektor solusi baru secara acak sesuai UB dan LB.

Parameter PAR sangat berpengaruh dalam mengontrol dan menyesuaikan vektor solusi sehingga menjaga tingkat konvergensi algoritma untuk menemukan solusi optimal. Apabila probabilitas nilai acak kurang dari parameter PAR, maka langkah selanjutnya adalah penyesuaian NHV sesuai persamaan berikut:

$$x_{new} = x_{old} + bw \times rand \quad (2.12)$$

Dimana x_{new} adalah nada baru setelah dilakukan penyesuaian,

x_{old} adalah nada lama yang tersimpan pada *harmony memory,*

bw adalah *bandwidth,*

$rand$ adalah bilangan *random* dengan interval [-1,1].

2.6 Evaluasi Hasil

Ada dua proses evaluasi dalam penelitian ini, yakni penghitungan nilai *fitness* pada saat proses pembentukan kelompok dokumen dan evaluasi hasil pengelompokan dokumen. Penghitungan nilai *fitness* pada saat proses pembentukan kelompok dokumen ini berfungsi sebagai acuan dalam pemilihan solusi dari beberapa kemungkinan alternatif solusi yang ada. Penentuan nilai *fitness* sebagai evaluasi pada saat proses *clustering* yang dihasilkan dari proses pembentukan vektor solusi baru yang digunakan dalam penelitian ini adalah *Average Distance of Documents to the cluster Centroid (ADDC).*

Misalkan $C = (c_1, c_2, \dots\ c_K)$ merupakan K *centroid* dari *cluster* untuk tiap baris HM, maka *centroid* dari *cluster* ke-k adalah $c_k = (c_{k1}, c_{k2}, \dots c_{kn})$ dengan persamaan:

$$c_{kj} = \frac{\sum_{i=1}^{m} a_{ki} d_{ij}}{\sum_{l=1}^{n} a_{ki}} \quad (2.13)$$

Tujuan dari penentuan nilai ADDC adalah untuk memperbesar nilai *intra-cluster similarity*. ADDC ini dirumuskan dengan persamaan:

$$f = \left[\sum_{i=1}^{K} \frac{1}{n_i} \sum_{j=1}^{m_i} \left(D(c_i, d_j)\right)^2\right] \Big/ K \quad (2.14)$$

dengan K adalah jumlah cluster yang ada, m_i adalah jumlah dokumen dalam cluster i, D adalah kemiripan dan d_{ij} adalah dokumen ke-j pada *cluster* i.

Untuk evaluasi pada hasil pembentukan kelompok dokumen menggunakan *F-Measure*. *F-Measure* merupakan ukuran akurasi berdasarkan nilai *recall* dan *precision* terhadap suatu mekanisme temu kembali informasi. *Precision* merupakan perbandingan *cluster* yang terdiri dari beberapa dokumen dari kelas tertentu terhadap *cluster* yang ada. *Recall* merupakan perbandingan *cluster* yang terdiri dari beberapa dokumen dari kelas tertentu terhadap kelas yang ada. *Precision* dan *recall* didefinisikan dengan:

$$Precision(\delta_i, C_j) = \frac{|\delta_i \cap C_j|}{|C_j|} \quad (2.15)$$

$$Recall(\delta_i, C_j) = \frac{|\delta_i \cap C_j|}{|\delta_i|} \quad 2.16)$$

F-Measure dari kelas i dan *cluster* C_j. tentukan dengan:

$$F(\delta_i, C_j) = \frac{2 * Recall|\delta_i \cap C_j| * Precision|\delta_i \cap C_j|}{Recall|\delta_i \cap C_j| + Precision|\delta_i \cap C_j|} \quad (2.17)$$

F-Measure dari sebuah kelas i adalah *F-Measure* maximum yang didapatkan dari proses *clustering*. *F-Measure* total adalah total bobot *F-Measure* dari semua kelas yang ada. *F-Measure* total dirumuskan dengan:

$$F - Measure(\delta_i) = \max F\left(\delta_i, C_j\right) \quad (2.18)$$

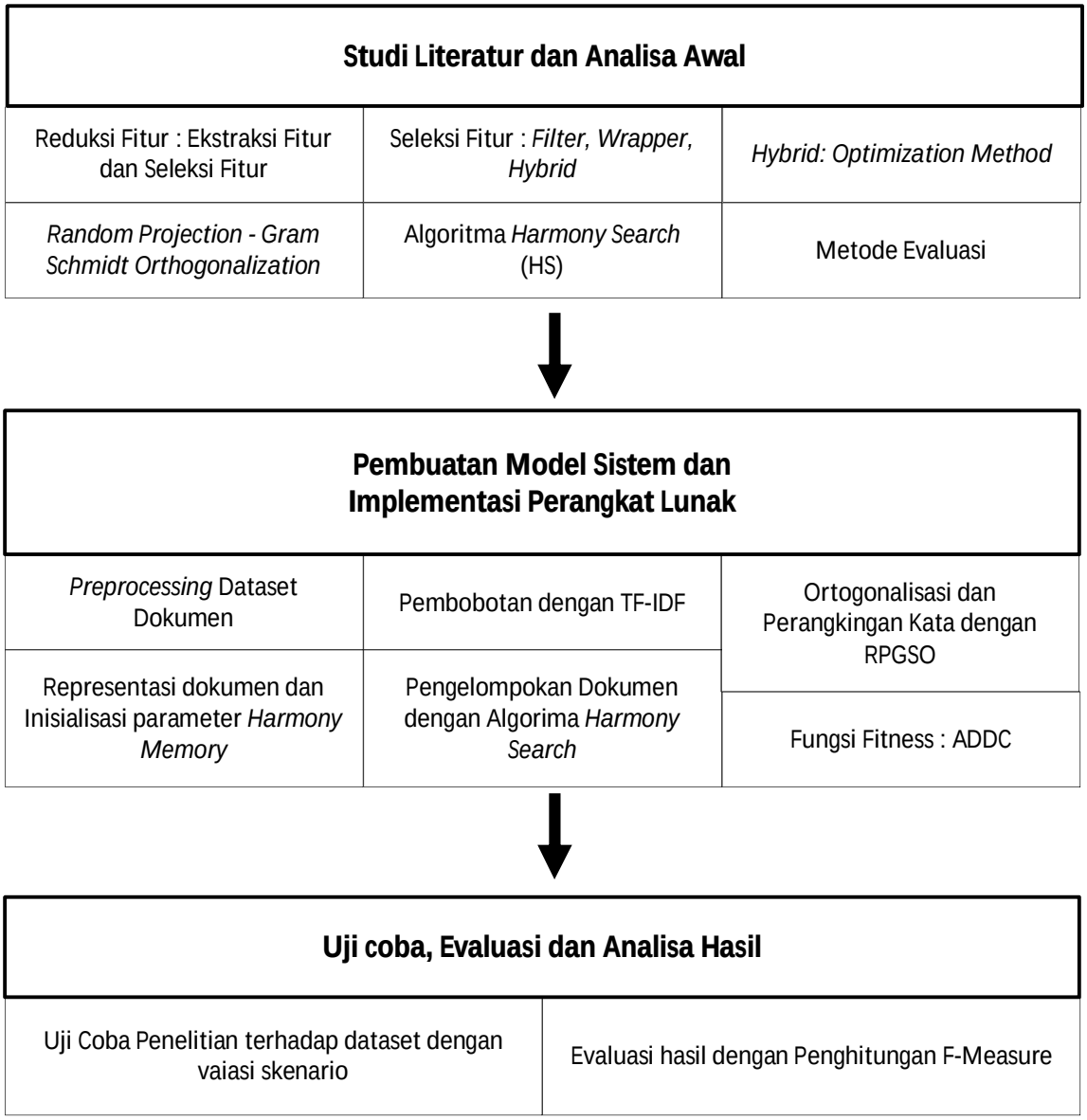
$$F - Measure = \sum_{i=1}^{K} \frac{|\delta_i|}{N} * F - Measure(\delta_i) \quad (2.19)$$

Semakin tinggi nilai F-Measure menunjukkan bahwa semakin bagus proses pengelompokan dokumen, sebaliknya proses pengelompokan dokumen yang kurang bagus akan menghasilkan nilai F-Measure yang rendah.

[*Halaman ini sengaja dikosongkan*]

BAB 3
METODE PENELITIAN

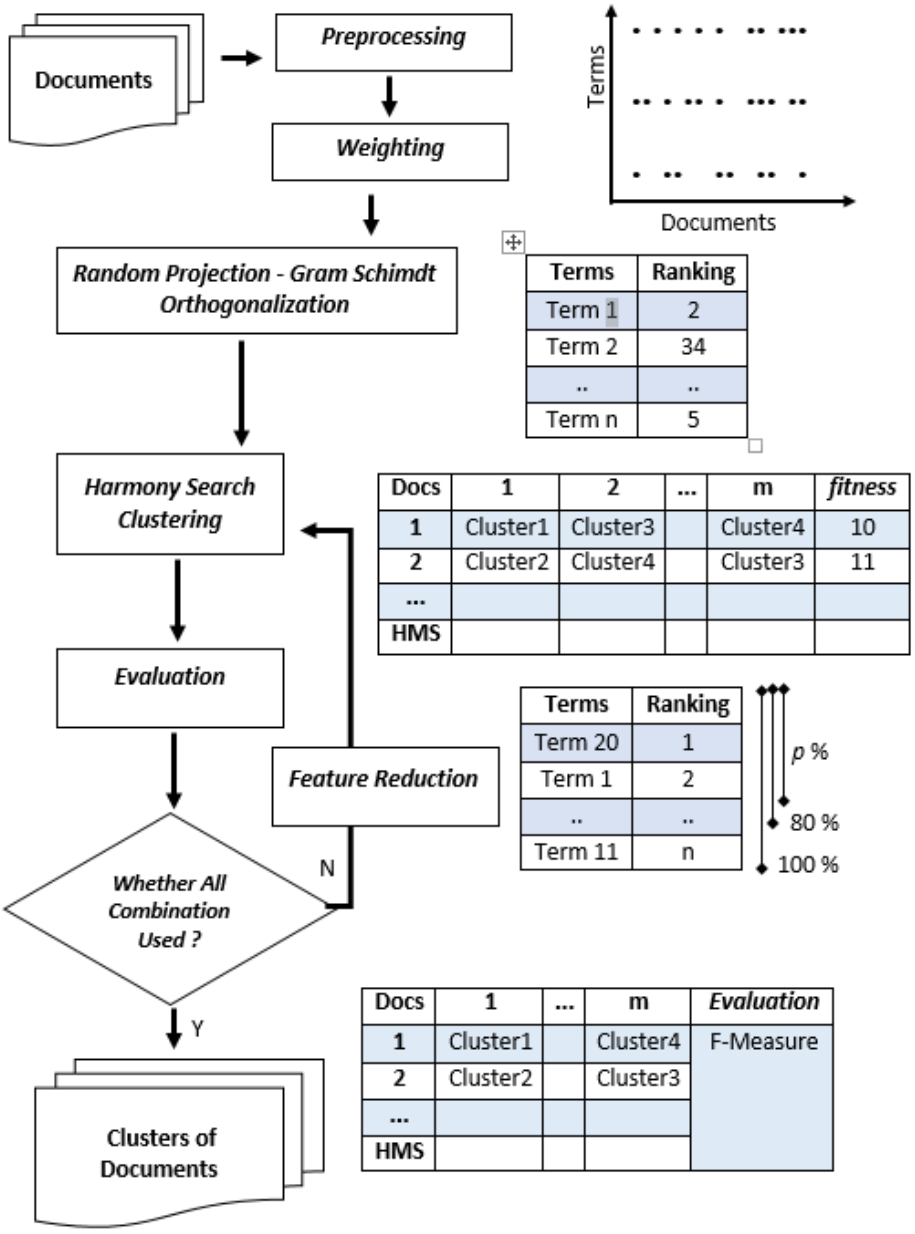
Tahapan dilakukan pada penelitian ini adalah (i) Studi Literatur dan Analisa Awal, (ii) Pembuatan Model Sistem, (iii) Pembuatan Perangkat Lunak, (iv) Uji Coba, dan (v) Evaluasi dan Analisa Hasil. Alur metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.1 Studi Literatur dan Analisa Awal

Pada tahap ini akan dipelajari segala informasi dan sumber pustaka yang disesuaikan dengan konteks penelitian yang dilakukan. Dalam penelitian ini, literatur yang dikaji secara garis besar meliputi konsep-konsep dasar yang berkaitan dengan metode *Random Projection - Gram Schmidt Orthogonalization*, seleksi fitur, algoritma *Harmony Search* dan pengelompokan dokumen serta perkembangan penelitiannya. Studi literatur yang dilakukan mencakup pencarian dan mempelajari referensi-referensi yang terkait.



Gambar 3.2 Model sistem yang diajukan

3.2 Desain Model Sistem

Pada bagian ini akan dipaparkan penggambaran alur proses yang terjadi dalam metode atau sistem untuk menghasilkan *output*. Secara global desain model sistem yang diajukan dapat dilihat pada Gambar 3.2.

3.2.1. Dokumen Dataset

Dataset yang digunakan pada penelitian ini adalah dokumen berita berbahasa indonesia dari situs berita kompas[2]. Dataset dokumen berita ini terbagi menjadi 5 kategori, yakni ekonomi, entertainment, politik, olahraga dan teknologi. Uji coba dilaksanakan terhadap tiga buah dataset dokumen, yakni 100 dokumen, 150 dokumen dan 200 dokumen yang masing-masing terdiri dari 20 dokumen, 30 dokumen dan 40 dokumen untuk tiap kategori. Dokumen berita dataset ini tertanggal mulai tahun 2010 sampai tahun 2015. Contoh dokumen untuk tiap kategori dapat dilihat pada Tabel 3.1

Tabel 3.1. Contoh dokumen per kategori

No	Isi Dokumen	Kategori
1	Bank **Indonesia** mencatat total rata-rata pengeluaran wisatawan mancanegara sebesar 125,93 **dollar** AS atau sekitar Rp 1,76 juta (kurs Rp 14.000 per dollar AS) per hari. Hal itu didasarkan pada hasil **survei** Perilaku wisatawan mancanegara. Selain itu, BI juga mencatat bahwa rata-rata lama tinggal wisatawan mancanegara mencapai 7,66 hari. "Angka pengeluaran ini lebih rendah dibandingkan pengeluaran wisman pada **periode** yang sama pada tahun 2014 lalu, yaitu mencapai 190,07 dollar AS per hari. Sementara rata-rata lama tinggal mencapai 8,19 hari," kata Dewi Setyowati, Kepala Kantor Perwakilan Bank **Indonesia** Provinsi Bali, Rabu (9/9/2015)	Ekonomi
2	**Artis** multi talenta kelahiran Chicago, Illinois, AS, pada 21 Juli 1951, Robin McLaurin **Williams** (63) atau Robin **Williams** meninggal **dunia** di kediamannya di bagian utara California, AS, pada 11 Agustus 2014 waktu setempat. Ia diduga bunuh diri. Dicatat oleh Wikipedia, **Williams**, yang sebelumnya tampil dalam **film-film televisi**, mengawali **karier**nya sebagai **aktor** dalam **industri film layar lebar** dengan bermain dalam	Entertainment

[2] www.kompas.com

	film Popeye pada 1980. Setelah itu **Williams** bersinar lewat **film-film** yang melibatkannya sebagai pemain atau pengisi suara, antara lain Hook, Aladdin, Jumanji, Night at the Museum, dan Happy Feet. ...	
3	Calon **presiden** independen Rizal Ramli akan menyelidiki kemungkinan adanya kecurangan daftar pemilih tetap (DPT) dalam Pemilu 2009. Ia mengatakan akan melakukan *mapping* pada masing-masing **daerah** untuk mengetahui kecenderungan adanya permainan DPT. "Kami akan lakukan *mapping* dan melihat di **daerah** kemenangan **partai** tertentu dengan hilangnya **suara rakyat**. Akan dilihat yang banyak DPT dengan kemenangan **partai**, dan daerah dengan jumlah DPT yang ditolak dengan kekalahan **partai** tertentu. Itu akan kami koreksi dulu," ujarnya. Lebih jauh Rizal mengatakan, selain dengan blok perubahan, ia juga akan melakukan komunikasi dengan parpol besar terkait hal ini. ...	Politik
4	Pemain Malaysia, Lee Chong Wei, mungkin harus melupakan ambisinya menjadi **juara dunia** setelah dikalahkan Chen Long pada **final** Total BWF World Championships, Minggu (16/8/2015). Dalam **final** yang seperti mengulang **final** kejuaraan **dunia** tahun lalu, Chen Long mempertahankan **gelar juara** dengan **menang** 21-14, 21-17. Pertandingan yang berlangsung penuh dengan demonstrasi teknik yang tinggi ini berlangsung dalam 1 jam 5 menit. Setelah **lolos** ke **final** dengan menyisihkan pemain Denmark, Jan O Jorgensen pada semifinal, Chong Wei menyatakan tekadnya untuk mewujudkan impian menjadi pemain Malaysia pertama yang menjadi **juara dunia** bulu tangkis. "Tidak mungkin untuk menunggu hingga 2017," katanya. ...	Olahraga
5	Setelah tersedia di **sistem operasi Android** dan iOS, Facebook resmi merilis aplikasi pesan instan Facebook Messenger untuk **perangkat** bersistem operasi **Windows** Phone, Senin (21/4/2014). **Aplikasi** yang bisa diunduh secara gratis di **Windows** Phone Store ini akan menampilkan foto profil dari orang-orang yang terhubung dengan akun Facebook pengguna. Ada tiga layar tab dalam aplikasi Facebook Messenger di Windows Phone yang dapat dinavigasi dengan cara digeser. Pertama, tab 'Recent' yang memperlihatkan daftar perbincangan terakhir yang terjadi dengan teman di Facebook. Tab 'Messenger' memperlihatkan kepada **pengguna** tentang siapa saja yang bisa dihubungi ...	Teknologi

3.2.2. Preprocessing

Tahapan pertama pada penelitian ini adalah *preprocessing* dokumen. Pada tahapan ini dilakukan proses *tokenizing* kata dari dataset dokumen. Setelah proses *tokenizing* kata dilanjutkan dengan proses *non-character removal*, yakni menghilangkan *token* yang bukan kata, seperti tanda baca dan angka. Selanjutnya adalah *stopword removal*, yakni menghilangkan kata-kata yang tidak memiliki makna khusus, seperti kata 'yang', 'tidak', 'dan', dan lainnya. Proses *stopword removal* menggunakan *stopword list* tertentu sebagai acuan untuk menentukan apakah kata yang sedang diproses merupakan *stopword* atau bukan. Proses selanjutnya adalah proses *stemming* untuk membentuk kata dasar dari kata berimbuhan dan sisipan yang ada. Proses ini bertujuan agar kata-kata yang memiliki bentuk dasar sama tidak tersimpan dan terindeks sebagai kata yang berbeda.Pada Tabel 3.2 dapat dilihat contoh dokumen sebelum melalui tahap *preprocessing* dan setelah melalui tahap *preprocessing*. Kata-kata yang ditebalkan adalah kata-kata yang mengalami perubahan, baik itu yang mengalami proses *stemming* maupun yang dihilangkan karena dianggap sebagai *stopword*.

Tabel 3.2. Contoh dokumen sebelum dan setelah *preprocessing*

Dokumen sebelum *preprocessing*	**Dokumen setelah *preprocessing***
Bank Indonesia **mencatat** total **rata-rata pengeluaran** wisatawan mancanegara **sebesar 125,93** dollar **AS atau sekitar Rp 1,76** juta **(**kurs **Rp 14.000 per** dollar **AS) per** hari**. Hal itu didasarkan pada** hasil survei Perilaku wisatawan mancanegara. **Selain itu, BI juga mencatat bahwa rata-rata lama** tinggal wisatawan mancanegara **mencapai 7,66** hari**. "**Angka **pengeluaran ini lebih** rendah **dibandingkan pengeluaran** wisman **pada** periode **yang sama pada** tahun **2014** lalu**, yaitu mencapai 190,07** dollar **AS per** hari. **Sementara rata-rata** lama tinggal **mencapai 8,19** hari**,"** kata Dewi Setyowati**,** Kepala Kantor **Perwakilan** Bank Indonesia Provinsi **Bali,** Rabu **(9/9/2015)**	Bank indonesia catat total rata keluar wisatawan mancanegara besar dollar juta kurs dollar hari dasar hasil survei perilaku wisatawan bi catat rata tinggal wisatawan mancanegara capai hari angka keluar rendah banding keluar wisman periode tahun capai dollar hari rata tinggal capai hari kata dewi setyowati kepala kantor wakil bank indonesia provinsi bal rabu

3.2.3. Pembobotan (*Weighting*)

Proses selanjutnya adalah pembobotan terhadap kata berdasarkan metode pembobotan tertentu. Skema pembobotan yang digunakan pada penelitian ini adalah kombinasi dari *Term Frequency - Invers Document Frequency* (TF-IDF) (Everitt, 1980) (Salton G. , 1989), yang dapat dihitung dengan rumus berikut (Salton & Buckley, 1988):

$$TF_{d,t} = f(d,t) \tag{3.1}$$

$$IDF_{d,t} = \log\left(1 + \frac{N}{N_{d,t}}\right) \tag{3.2}$$

$$W_{d,t} = TF_{d,t} \times IDF_{d,t} \tag{3.3}$$

dimana $TF_{d,t}$ adalah *term frequency*, yaitu jumlah keberadaan kata t dalam dokumen d, dan $IDF_{d,t}$ adalah *invers document frequency*, dengan N adalah jumlah dokumen dalam seluruh koleksi, dan $N_{d,t}$ adalah jumlah dokumen yang memiliki kata t. $IDF_{d,t}$ berkaitan dengan keberadaan sebuah kata dalam dokumen, semakin banyak dokumen yang memuat kata tersebut, maka bobotnya semakin kecil. Sebaliknya semakin sedikit dokumen yang memuat kata tersebut, maka bobotnya semakin besar.

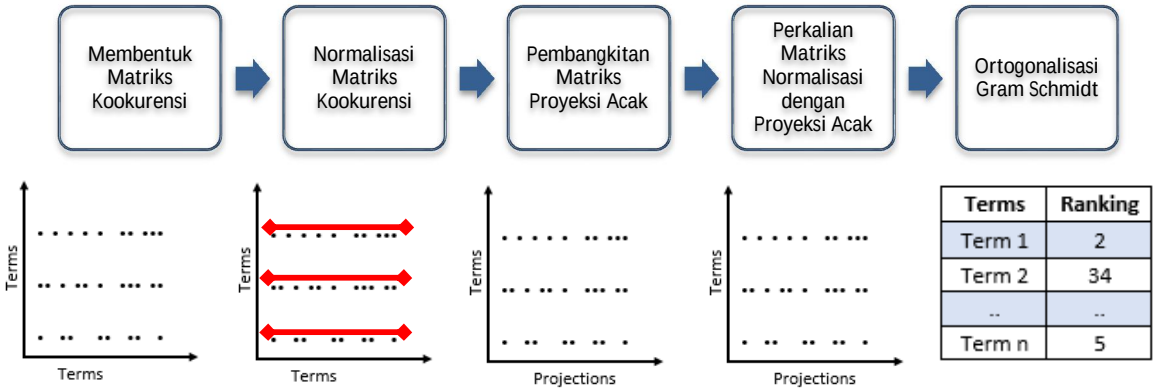
Dari proses ini dihasilkan matriks D *document-term* yang memetakan dokumen dengan kata-kata yang ada masing-masing dokumen sesuai bobot hasil perhitungan TF-IDF pada Persamaan 3.3. Representasi dokumen setelah *preprocessing* sebagai berikut:

$$D_i = \{w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}\} \tag{3.4}$$

Dengan w_{in} menunjukkan bobot kata ke-n pada dokumen ke-i.

3.2.4. *Random Projection and Gram–Schmidt Orthogonalization*

Tahapan selanjutnya adalah proses perangkingan kata menggunakan ortogonalisasi untuk mendapatkan kata-kata yang relevan. Proses ini menggunakan metode *Random Projection – Gram Schmidt Orthogonalization* (RP-GSO), sebagaimana alur proses pada Gambar 3.3 dan algoritma pada Gambar 3.4.



Gambar 3.3 Tahapan Algoritma RPGSO dan *output* tiap tahapan.

Langkah pertama pada proses RP-GSO sesuai Gambar 3.3 dan Gambar 3.4 adalah pembentukan matriks kookurensi kata berdasarkan matriks *document-term* yang terbentuk dari *preprocessing* dokumen. Pembentukan matriks kookurensi kata ini bertujuan untuk mengatasi *sparseness* yang menjadi karakter utama dari representasi matriks *document-term* serta dapat menghasilkan kata-kata yang lebih baik sebagai fitur yang paling penting (Wang, Zhang, Liu, Liu, & Wang, 2016).

Pada matriks kookurensi kata, untuk set dokumen D sebanyak m dokumen dan n kata, $D_i = \{w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}\}$, misalkan B_i merupakan vektor kolom pada R^n, maka matriks kookurensi kata Q berdimensi $n \times n$ dirumuskan dengan:

$$Q = \frac{1}{m} \times \left(\bar{B} . \bar{B}^T - \tilde{B}\right) \tag{3.5}$$

$$\bar{B}_i = \frac{B_i}{\sqrt{\langle D_i \rangle \times (\langle D_i \rangle - 1)}} \tag{3.6}$$

$$\tilde{B}_i = \frac{diag(B_i)}{\sqrt{\langle D_i \rangle \times (\langle D_i \rangle - 1)}} \tag{3.7}$$

dengan, $\bar{B}$ adalah vektor kolom yang dinormalisasi, $\tilde{B}$ adalah diagonal vektor kolom yang dinormalisasi, dan $\langle D_i \rangle$ merupakan panjang dokumen D.

Berikutnya adalah proses normalisasi. Matriks kookurensi kata Q yang telah terbentuk kemudian dinormalisasi per baris menjadi sehingga didapatkan distribusi per baris dalam unit yang sama.

Langkah selanjutnya adalah pembangkitan matriks proyeksi acak. Proses ini bertujuan untuk mengurangi dimensi matriks tanpa merubah karakter asal matriks

Input: *D*, yakni dataset sejumlah *m* dokumen dan total *n* kata.
p, yakni dimensi baris pada matriks proyeksi acak.

1. Membentuk *Q*, yakni matriks kookurensi kata berdimensi *n x n*.

```
for i=1 to m
    sumrow = jumlah perbaris
    mul =(sumrow==1)? sumrow^2: sumrow*( sumrow-1)
    for j=1 to n
        diagD = diagD + (w_{i,j} /mul)
        w_{i,j} = w_{i,j} /sqrt(mul)
    end for
end for
Q = [(D'*D)-diag(diagD)]/m
```

2. Normalisasi matriks *Q* menjadi $\bar{Q}$.

$\bar{Q} = normr(Q)$

3. Pembangkitan matriks proyeksi acak *R* berdimensi *p x n*, dengan *p << n*.

```
R = matriks proyeksi
R (find(rand<=1/6) = -1;
R (find(rand>1/6 & rand<=5/6)) = 0;
R (find(rand>5/6 & rand<=1)) = 1;
R = sqrt(3)*(R);
```

4. Perkalian matrix proyeksi acak R dengan matriks $\bar{Q}$.

$\overline{Q_{rp}} = R \times \bar{Q}$.

5. Ortogonalisasi Gram Schmidt terhadap matriks $\overline{Q_{rp}}$ dan memberi ranking sesuai jarak dari titik pusat, mulai dari yang terjauh.

u_n = *GramSchmidt* ($w_{rp\ (i,j)}$)

Gambar 3.4 Algoritma RPGSO

Teknik ini menggunakan transformasi *Johnson and Lindenstrauss* (Johnson, 1984) yang kemudian dikembangkan lebih lanjut untuk permasalahan basis data (Achlioptas, 2003). Transformasi *Johnson and Lindenstrauss* menjelaskan bahwa sejumlah titik pada suatu dimensi Euclidean *D* dapat diproyeksikan pada dimensi Euclidean lain yang lebih kecil. Sesuai penelitian bahwa pada permasalahan basis data, matriks proyeksi yang dihasilkan berada dalam nilai {-1, 0, 1} (Achlioptas, 2003). Pembangkitan matriks proyeksi acak *R* berdimensi *p x n* , dengan *p << n* mengikuti persamaan:

$$R_{i,j} = \sqrt{3} \times \begin{cases} -1 & dengan\ probabilitas\ \ 1/6 \\ 0 & dengan\ probabilitas\ \ 2/3 \\ 1 & dengan\ probabilitas\ \ 1/6 \end{cases} \quad (3.8)$$

Dengan nilai dimensi p pada matriks proyeksi acak R merupakan parameter setting tertentu. Berdasarkan penelitian lain, nilai dimensi p pada matriks proyeksi acak R tidak terlaku signifikan terhadap keluaran yang dihasilkan (Wang, Zhang, Liu, Liu, & Wang, 2016).

Matriks proyeksi ini kemudian dikalikan dengan matriks kookurensi kata yang telah dinormalisasi $\bar{Q}$. Perkalian matriks R yang berdimensi $p \times n$ dengan matriks $\bar{Q}$ berdimensi $n \times n$ menghasilkan matrks baru $\overline{Q_{rp}}$ berdimensi $n \times p$ yang menjadi dasar dalam ortogonalisasi dengan metode *Gram Schmidt.*

Langkah terakhir adalah proses ortogonalisasi dengan metode *Gram Schmidt* terhadap matriks $\overline{Q_{rp}}$ sesuai Persamaan 2.9 dan 2.10, lalu menghitung jarak masing-masing titik dari titik pusat. Mengacu pada hasil ortogonalisasi tersebut, kemudian didapatkan urutan berdasarkan jaraknya dari titik pusat. Urut-urutan matriks inilah yang kemudian menjadi dasar dalam proses seleksi fitur untuk proses pengelompokan dokumen.

3.2.5. *Harmony Search Clustering* (HSC)

Tahapan pengelompokan dokumen dilakukan dengan algoritma *Harmony Search* (HS) untuk mendapatkan *cluster* dokumen berdasarkan kata terpilih. Algortima pengelompokan dokumen dengan *Harmony Search* seperti Gambar 3.5.

Langkah pertama penyelesaian masalah dengan algoritma HSC sesuai Gambar 3.5 adalah setting parameter HS yang terdiri dari jumlah *cluster* K, *Uppper Bound* (UB) dan *Lower Bound* (LB) yakni nilai batas atas dan batas bawah label *cluster* dokumen, *Harmony Memory* (HM) yakni representasi label *cluster* dokumen, *Harmony Memory Size* (HMS) yakni banyaknya generasi dari representasi label *cluster* dokumen yang menjadi acuan bagi pembangkitan generasi-generasi berikutnya, *Harmony Memory Considering Rate* (HMCR) yakni nilai yang menentukan kecenderungan sistem untuk membangkitan generasi baru atau memperbaiki generasi yang sudah ada, *Pitch Adjusting Rate* (PAR) yakni nilai yang menentukan mekanisme perbaikan generasi yang sudah ada, serta *Number of Iterations* (NI) yakni jumlah iterasi yang dilakukan untuk mendapatkan nilai evaluasi yang optimal.

Input: *D*, yakni dataset sejumlah *m* dokumen dan total *n* kata.

1. Setting parameter : *K, UB, LB, HMS, HMCR, PAR, NI.*
2. Menentukan *fitness function*.
   f(x) = jarak rata-rata dokumen terhadap centroid cluster
3. Membangkitkan harmony inisial awal
   $x_i, LB \leq x_i \leq UB, = \{1, 2 \dots, K\}, untuk\ i\ bernilai\ 1 \leq i \leq m$
4. Pengelompokan dokumen dengan HS

```
bestfit=0;
bestharm=0;
for i=1 to NI
    randval=rand()
    if(randval<=HMCR)
       pilih klaster pada harmony inisial
       if(randval<=PAR)
            pilih label klaster dengan jarak dok-klaster terdekat
       else
            pilih label klaster dengan probabilitas sesuai jarak dok-
            klaster relatif terhadap jarak dok dengan semua klaster
       endif
    else
       bangkitkan label klaster secara acak, x_i = x_{i-1} + randval *
       (LB − UB)
    endif
    tentukan masing-masing center klaster baru
    kelompokkan dokumen dengan center terdekat
    fitness = hitung fitness harmony baru
    if(fitness lebih baik drpd bestfit)
       bestfit=fitness
       bestharm= harmony baru
    endif
endfor
```

Gambar 3.5 Algoritma HSC

Sesuai dengan label kelas pada dataset dokumen yang digunakan, maka pada penelitian ini jumlah *cluster* K sama dengan label kelas dokumen berita, yakni 5 *cluster*. Parameter *Uppper Bound* (UB) dan *Lower Bound* (LB) merupakan batas atas dan batas bawah nilai yang digunakan pada label dokumen. Mengacu pada nilai *cluster* K tersebut, maka nilai *Uppper Bound* (UB) dan *Lower Bound* (LB) masing-masing adalah 5 dan 1.

Harmony Memory (HM) identik dengan populasi pada Algoritma Genetika, merupakan kumpulan dari vektor solusi yang merepresentasikan permasalahan yang sedang diselesaikan. *HM* pada permasalahan ini merepresentasikan dokumen dan label *cluster*nya. Posisi vektor solusi menunjukkan urutan dokumen dan label pada vektor solusi menunjukkan *cluster* yang memuat dokumen tersebut seperti Gambar 3.6.

Harmony Memory Size (HMS) menunjukkan jumlah alternatif solusi yang mungkin diambil pada saat proses pengelompokan dokumen. Pada permasalahan ini HMS merupakan alternatif-alternatif label *cluster* yang berbeda-beda pada masing-masing dokumen. Berdasarkan penelitian pada permasalahan serupa (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013), pada penelitian ini nilai HMS menggunakan nilai 2 kali jumlah *cluster*, yakni 10.

Harmony Memory Considering Rate (HMCR) merupakan probabilitas nilai sebagai acuan apakah vektor solusi yang baru, yakni *New Harmony Vector* (NHV), diambil dari HMS yang sudah ada ataukah NHV didapatkan dengan membuat nilai acak baru. *Pitch Adjusting Rate* (PAR) merupakan probabilitas untuk menentukan cara pemilihan NHV dari HMS yang sudah ada. *Number of Iterations* (NI) menyatakan banyaknya iterasi yang dilakukan untuk membentuk kelompok dokumen.

Pada tahapan ini, digunakan representasi vektor dokumen dengan panjang m, yakni sejumlah dokumen. Setiap elemen dari vektor ini adalah label dari *cluster* dokumen. Jika terdapat K *cluster*, maka setiap elemen dari vektor solusi adalah nilai integer dalam kisaran $K = \{1,. . . , K\}$. Vektor.

Pada Gambar 3.6 terlihat pada *Harmony Memory* (HM) pertama, *cluster* 1 memiliki anggota dokumen dengan label {1,2,6,10}, *cluster* 2 memiliki anggota dokumen dengan label {3,7}, *cluster* 3 memiliki anggota dokumen dengan label {4,8}, dan *cluster* 4 memiliki anggota dokumen dengan label {5}. Nilai fitness untuk HM pertama sebesar 10.

Sedangkan pada *Harmony Memory* (HM) kedua, *cluster* 1 memiliki anggota dokumen dengan label {2,9,10}, *cluster* 2 memiliki anggota dokumen dengan label {1,8}, *cluster* 3 memiliki anggota dokumen dengan label {3,4,7}, dan *cluster* 4

memiliki anggota dokumen dengan label {5,6}. Nilai *fitness* untuk HM kedua ini sebesar 11.

Dok. / HM	1	2	3	4	5	6	7	8	9	10	*fitness*
1	1	1	2	3	4	1	2	3	3	1	10
2	2	1	3	3	4	4	3	2	1	1	11
...											
HMS-1											
HMS											

Gambar 3.6 Representasi *Harmony Memory* pada pengelompokan dokumen.

Setelah setting parameter, langkah selanjutnya sesuai Gambar 3.5 adalah menentukan nilai *fitness* sebagai evaluasi hasil *clustering* yang dihasilkan dari proses pembentukan vektor solusi baru. Nilai *fitness* yang digunakan dalam penelitian ini adalah *Average Distance of Documents to the cluster Centroid* (ADDC). Misalkan $C = (c_1, c_2, \dots\ c_K)$ merupakan K *centroid* dari *cluster* untuk tiap baris HM, maka *centroid* dari *cluster* ke-k adalah $c_k = (c_{k1}, c_{k2}, \dots\ c_{kn})$ dengan persamaan:

$$c_{kj} = \frac{\sum_{i=1}^{m} a_{ki} d_{ij}}{\sum_{l=1}^{n} a_{ki}} \tag{3.9}$$

Tujuan dari penentuan nilai ADDC adalah untuk memperbesar nilai *intra-cluster similarity*. ADDC dirumuskan dengan persamaan:

$$f = \left[\sum_{i=1}^{K} \frac{1}{n_i} \sum_{j=1}^{m_i} \left(D(c_i, d_j)\right)\ \right] \Big/ K \tag{3.10}$$

dengan K adalah jumlah cluster yang ada, m_i adalah jumlah dokumen dalam cluster i, D adalah kemiripan dan d_{ij} adalah dokumen ke-j pada *cluster* i.

Sesuai Gambar 3.5, langkah selanjutnya adalah membangkitkan *harmony* awal sesuai dengan rentang batas bawah LB dan batas UB yang telah ditentukan, yakni antara 1 sampai 5. Persamaan untuk membangkitkan harmony baru adalah:

$$x_i = LB + rand * (UB - LB) \tag{3.11}$$

Selanjutnya adalah proses pengelompokan dokumen dengan algortima HS. Pada langkah pembentukan vektor solusi baru, yaitu *New Harmony Vector* (NHV), dihasilkan dari vektor solusi yang tersimpan di HM. Vektor harmoni baru yang dihasilkan harus sebanyak mungkin mewarisi informasi dari vektor solusi di HM.

Langkah pemilihan label *cluster* dokumen berdasarkan dua kemungkinan, dengan probabilitas sebesar HMCR untuk pemilihan vektor solusi baru berdasarkan vektor solusi yang telah ada pada HM dan dengan probabilitas 1-HMRC untuk pemilihan vektor solusi baru secara acak dari himpunan *{1, 2,. . . , K}* yang merupakan kemungkinan label *cluster*.

Setelah menghasilkan solusi baru, dilakukan proses penyesuaian nilai (*pitch adjustment*) yang telah diperoleh. Parameter PAR sangat berpengaruh dalam mengontrol dan menyesuaikan vektor solusi sehingga menjaga tingkat konvergensi algoritma untuk menemukan solusi optimal. Untuk setiap dokumen yang pemilihan labelnya berdasarkan vektor solusi yang telah ada pada HM, ada dua kemungkinan yang dapat dilakukan, pertama dengan merubah label *cluster* yang telah ada dengan label *cluster* baru yang memiliki jarak minimum menurut persamaan:

$$NVH[i] = \arg\min_{j \in [K]} D(d_i, c_j) \tag{3.12}$$

Dimana d_i menyatakan dokumen ke-i dan c_j menyatakan *cluster* j,

Kemungkinan kedua dengan merubah label *cluster* yang telah ada dengan label *cluster* baru yang dipilih secara random berdasarkan distribusi berikut:

$$p_j = \frac{D_{max} - D(d_i, c_j)}{NF}\left(1 - \frac{gn}{NI}\right) \tag{3.14}$$

$$NF = KD_{max} - \sum_{j=1}^{K} D(d_i, c_j) \tag{3.15}$$

$$D_{max} = max_i D(NHV, c_i) \tag{3.16}$$

dengan NI merupakan total iterasi, dan gn merupakan iterasi saat ini.

3.2.6. Evaluasi

Tahapan selanjutnya adalah evaluasi apakah pengelompokan yang dihasilkan dari algoritma *Harmony Search* dengan fitur yang telah diseleksi merupakan solusi optimal atau bukan. Untuk mengevaluasi hasil tersebut digunakan *F-measure* berdasarkan Persamaan 2.15 sampai Persamaan 2.19.

F-measure merupakan nilai yang menunjukkan kualitas *cluster* yang terbentuk dibandingkan dengan kategori dokumen asal. Semakin banyak dokumen yang dikenali sebagai cluster yang sama dengan kategori asal, maka nilai *F-measure* semakin tinggi. *F-measure* merupakan rata-rata harmonik antara *precison* dan *recall* cluster yang terbentuk terhadap kategori dokumen asal. *Precision* merupakan perbandingan dokumen yang tercluster sama dengan kategori asal dengan semua dokumen hasil cluster. Sedangkan *recall* merupakan perbandingan dokumen yang tercluster sama dengan kategori asal dengan semua dokumen kategori asal.

Semakin tinggi nilai *F-measure* suatu *cluster* menunjukkan bahwa semakin bagus *cluster* yang dihasilkan. Sebaliknya apabila nilai *F-measure* suatu *cluster* semakin rendah, maka hal ini menunjukkan semakin kurang bagus *cluster* yang dihasilkan. Berdasarkan kriteria tersebut, maka kondisi solusi optimal dapat ditentukan dengan nilai *F-measure* pada proses seleksi fitur.

Dengan kriteria *F-measure* diharapkan dapat menentukan solusi optimal. Solusi tersebut merupakan *cluster* terbaik dari hasil pengelompokan dokumen dengan kombinasi fitur yang paling bagus.

3.2.7. Pemeriksaan Kombinasi Fitur

Pada tahapan ini akan dilakukan pemeriksaan apakah semua kombinasi fitur pada dataset dokumen telah digunakan pada proses pengelompokan dokumen atau belum. Proses pengelompokan dokumen dilakukan berdasarkan urutan kepentingan fitur dari proses *Random Projection - Gram Schmidt Orthogonalization* (RP-GSO). Berdasarkan urutan kepentingan fitur tersebut, maka dilakukan pengelompokan dokumen menggunakan kombinasi fitur yang mungkin.

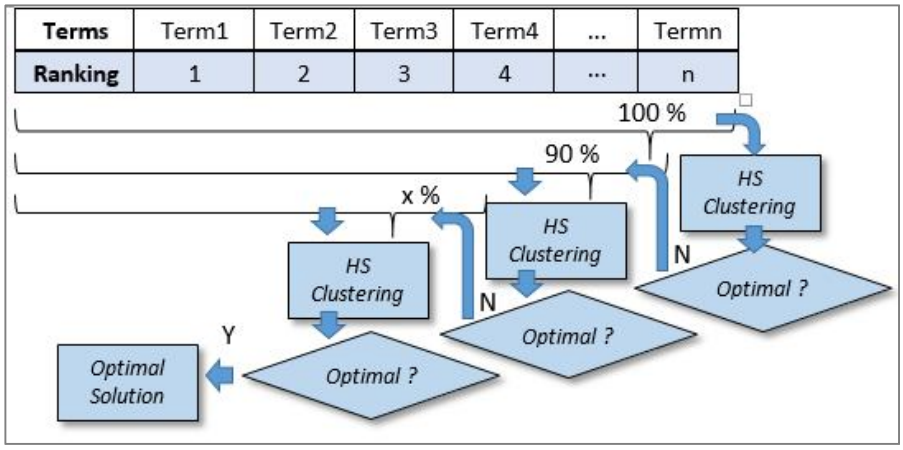
Pada iterasi pertama, seluruh fitur yang ada (100%) digunakan sebagai kombinasi awal. Iterasi berikutnya menggunakan kombinasi 90% dari seluruh fitur

yang ada, kemudian 80%, 70% sampai 0%, yakni tidak ada kombinasi fitur lagi yang dapat digunakan. Pada tahapan ini dilakukan pemeriksaan apakah semua kombinasi fitur telah digunakan dalam proses pengelompokan dokumen. Jika tidak ada lagi kombinasi fitur yang bisa digunakan, maka proses akan berakhir dan didapatkan kombinasi fitur optimal berdasarkan evaluasi nilai *F-Measure*. Jika masih ada kombinasi fitur yang digunakan, maka akan dilakukan proses pengelompokan dokumen menggunakan kombinasi fitur berikutnya.

3.2.8. Reduksi Fitur

Tahapan ini adalah tahapan untuk melakukan reduksi fitur kata pada dataset dokumen seperti Gambar 3.7. Pada tahapan ini, fitur kata yang telah terurut berdasarkan tingkat kepentingannya hasil dari proses *Random Projection - Gram Schmidt Orthogonalization* (RP-GSO) dikombinasikan kemudian digunakan untuk proses pengelompokan dokumen dengan algoritma *Harmony Search* (HS). Mekanisme reduksi fitur pada tahapan ini seperti algoritma pada Gambar 3.8.

Sebagai kombinasi awal adalah menggunakan seluruh fitur yang ada (100%). Kombinasi fitur-fitur tersebut kemudian digunakan pada proses pengelompokan dokumen dengan Algoritma *Harmony Search* sehingga membentuk *cluster-cluster* dokumen. *Cluster* yang terbentuk lalu dievaluasi untuk menentukan solusi optimal berdasarkan *F-measure*.



Gambar 3.7 Reduksi Fitur Pada Proses Pengelompokan Dokumen.

- Input: *T*, yakni *n* fitur *term* (kata) yang terurut.
  x, yakni jumlah fitur yang digunakan.
  p, yakni prosentase fitur yang digunakan
- Melakukan seleksi fitur pada fitur *T* sebanyak *x* persen.

```
opt = 0; bestFeat = 0;
newOpt =0;
x = n; p=x*100/n;
while  (p <> 0)
    feat = seleksi(p,T);
    hs = harmonySearch(feat);
    newOpt = checkOptimal(hs);
    if newOpt > opt
        opt = newOpt;
        bestFit = p;
    end if
    p=p-10
end while
```

Gambar 3.8 Mekanisme reduksi fitur pada proses pengelompokan dokumen.

Iterasi berikutnya menggunakan kombinasi 90% dari seluruh fitur yang ada sebagai masukan pada proses pengelompokan dokumen kemudian keluarannya dievaluasi dengan *F-Measure*. Iterasi berikutnya kemudian menggunakan 80% fitur, 70% sampai dengan 0% yakni ketika semua kombinasi fitur telah digunakan pada proses pengelompokan dokumen.

Mekanise reduksi fitur mengacu pada Gambar 3.8 sebagai berikut, pada setiap iterasi, jumlah fitur yang digunakan adalah *x*, prosentase fitur yang digunakan adalah *p* dari total *n* fitur kata yang ada pada set dokumen. Fitur sebanyak *x* atau *p* persen ini kemudian menjadi input bagi proses pengelompokan dokumen, dan hasilnya dievaluasi dengan kriteria *F-Measure*.

Proses reduksi fitur ini menggunakan nilai pengurangan 10% dari total seluruh fitur yang ada. Sehingga apabila ada 3.461 fitur, maka pada awal iterasi menggunakan 3.461 fitur, pada iterasi kedua menggunakan 90% yakni sebanyak 3.115 fitur, iterasi ketiga menggunakan 80% fitur yakni 2769, iterasi berikutnya 2423 fitur, berikutnya 2077 fitur dan seterusnya sampai semua kombinasi digunakan.

3.3 Pembuatan Perangkat Lunak

Tahapan pembuatan perangkat lunak merupakan suatu fase implementasi dari model sistem yang telah dirancang ke dalam suatu bahasa pemrograman. Tahapan ini akan menghasilkan suatu program sabagai media representatif terhadap hasil dari metode yang diusulkan. Perangkat lunak pemrograman yang digunakan untuk penelitian ini adalah Matlab.

3.4 Skenario Uji Coba

Setelah tahapan implementasi algoritma dan pembuatan perangkat lunak selesai, maka tahapan penelitian ini akan dilanjutkan dengan melakukan suatu uji coba terhadap sistem yang telah dibuat untuk melihat hasil dan melakukan evaluasi. Uji coba dimaksudkan untuk mengetahui apakah penelitian yang dilakukan telah dapat memenuhi tujuan penelitian sebagaimana yang telah direncanakan. Sebagaimana disebutkan di atas, tujuan dari penelitian ini adalah untuk menyeleksi fitur penting dari dokumen berita sehingga mampu menghasilkan pengelompokan dokumen dengan hasil evaluasi yang lebih baik

Ujicoba akan dilakukan dengan beberapa tahap untuk mendapatkan data pembanding. Uji coba pertama adalah menentukan tingkat kepentingan fitur kata dengan metode RP-GSO sehingga didapatkan urutan kepentingan masing-masing fitur. Uji coba kedua adalah menentukan jumlah fitur optimal berdasarkan hasil dari metode RP-GSO yang akan digunakan pada proses pengelompokan dokumen dengan Algoritma *Harmony Search.*

Uji coba ketiga adalah menentukan parameter optimal yang akan digunakan pada proses pengelompokan dokumen dengan Algoritma *Harmony Search.* Pada uji coba ini menggunakan jumlah fitur optimal dari hasil uji coba kedua. Parameter yang akan ditentukan pada uji coba ketiga ini adalah *Harmony Memory Size* (HMS) dan *Harmony Memory Consideration Rate* (HMCR).

Uji coba keempat adalah membandingkan metode usulan dengan metode RPGSO dan kombinasi metode *clustering* standard. Pada penelitian ini metode *clustering* yang digunakan sebagai pembanding adalah metode *clustering* K-Means. Untuk memperoleh hasil perbandingan lebih banyak, ada dua metode penentuan

jarak yang digunakan pada metode K-Means ini, yakni menggunakan *cosine similarity* dan *euclidean distance*.

Uji coba kelima adalah membandingkan metode usulan dengan metode RPGSO dan *harmony search clustering* dengan fungsi *fitness Sum of Squared Error (SSE)*. Masing-masing metode ini dijalankan sebanyak dua kali, pertama menggunakan parameter optimal dari hasil uji coba ketiga, kedua menggunakan parameter optimal yang dihasilkan dari penelitian sebelumnya tentang *Harmony Search Clustering* (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013).

3.5 Evaluasi dan Analisa Hasil

Evaluasi dilakukan untuk menguji apakah metode yang diusulkan menghasilkan hasil yang lebih baik jika dibandingkan dengan mekanisme serupa yang tidak menggunakan kombinasi RP-GSO dan Algoritma *Harmony Search*. Pada uji coba akan dilakukan beberapa kali dengan kombinasi setting parameter untuk mengetahui nilai parameter yang optimal sehingga menghasilkan pengelompokan dokumen yang baik. Kualitas hasil pengelompokan pada penelitian ini dievaluasi dengan kriteria *F-measure* dan *Average Distance of Documents to the cluster Centroid (ADDC)*.

BAB 4
UJI COBA DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai tahapan uji coba untuk proses seleksi fitur dengan *Random Projection - Gram Schmidt Orthogonalization* (RPGSO) pada pengelompokan dokumen dengan algoritma *Harmony Search* (HS) serta proses analisa terhadap hasil uji coba yang telah dilakukan.

4.1 Lingkungan Implementasi

Pada penelitian ini, uji coba dilaksanakan pada lingkungan : (a) Processor Intel(R) Core i5 CPU M540 @ 2,53 GHZ, RAM 4,00 GB,(b) Sistem Operasi Windows 7 Home Premium, (c) Perangkat lunak Matlab versi R2013a.

4.2 Hasil dan Uji Coba

Uji coba dilaksanakan pada dataset dokumen berita yang berasal dari situs berita kompas[3]. Dataset yang digunakan pada penelitian ini ada tiga buah, yakni 100 dokumen, 150 dokumen dan 200 dokumen. Pada masing-masing dataset terdiri dari 5 kategori, yakni kategori ekonomi, entertainment, olahraga, politik dan teknologi. Banyaknya dokumen pada tiap dataset untuk masing-masing kategori memiliki jumlah yang sama. Pada dataset 100 dokumen terdiri dari 20 dokumen berita untuk masing-masing kategori, untuk dataset 150 dokumen masing-masing terdiri dari 30 dokumen untuk tiap kategori, dan untuk dataset 200 dokumen masing-masing terdiri dari 40 dokumen untuk tiap kategori.

4.2.1. Uji Coba *Preprocessing*

Pada uji coba ini dilakukan proses *tokenizing* kata dari dataset dokumen. Setelah proses *tokenizing* kata dilanjutkan dengan proses *non-character removal*, yakni menghilangkan *token* yang bukan kata, seperti tanda baca dan angka. Selanjutnya adalah *stopword removal*, yakni menghilangkan kata-kata yang tidak memiliki makna khusus, seperti kata ‘yang’, ‘tidak’, ‘dan’, dan lainnya. Proses *stopword removal* menggunakan *stopword list* tertentu sebagai acuan untuk

[3] www.kompas.com

menentukan apakah kata yang sedang diproses merupakan *stopword* atau bukan. Pada uji coba ini menggunakan *stopword list* dari penelitian tentang efek *stemming* pada bahasa indonesia (Tala, 2003). Proses selanjutnya adalah proses *stemming* untuk membentuk kata dasar dari kata berimbuhan dan sisipan yang ada. Pada uji coba ini proses *stemming* menggunakan *library* sastrawi[4] yang menggunakan kamus kata dasar dari kateglo[5] dengan beberapa penyesuaian. *Library* ini menggunakan gabungan dari beberapa algoritma stemming, yakni Algoritma Nazief dan Andriani (Nazief & Adriani, 1996), Algoritma Confix Stripping (Asian, 2007), Algoritma Enhanded Confix Stripping (Arifin, Mahendra, & Ciptaningtyas, 2009), serta Modifikasi Enhanced Confix Stripping (Tahitoe & Purwitasari, 2010).

Pada Tabel 4.1 dapat dilihat contoh dokumen sebelum melalui tahap *preprocessing* dan setelah melalui tahap *preprocessing* dengan *stopword list* dan *stemming* dengan *library* sastrawi.

Tabel 4.1. Dokumen sebelum dan setelah tahap *preprocessing*

Dokumen sebelum *preprocessing*	**Dokumen setelah *preprocessing***
Bank Indonesia mencatat total rata-rata pengeluaran wisatawan mancanegara sebesar 125,93 dollar AS atau sekitar Rp 1,76 juta (kurs Rp 14.000 per dollar AS) per hari. Hal itu didasarkan pada hasil survei Perilaku wisatawan mancanegara. Selain itu, BI juga mencatat bahwa rata-rata lama tinggal wisatawan mancanegara mencapai 7,66 hari. "Angka pengeluaran ini lebih rendah dibandingkan pengeluaran wisman pada periode yang sama pada tahun 2014 lalu, yaitu mencapai 190,07 dollar AS per hari. Sementara rata-rata lama tinggal mencapai 8,19 hari," kata Dewi Setyowati, Kepala Kantor Perwakilan Bank Indonesia Provinsi Bali, Rabu (9/9/2015)	Bank indonesia catat total rata keluar wisatawan mancanegara besar dollar as rp juta kurs rp dollar as hari dasar hasil survei perilaku wisatawan bi catat rata tinggal wisatawan mancanegara capai hari angka keluar rendah banding keluar wisman periode tahun capai dollar as hari rata tinggal capai hari kata dewi setyowati kepala kantor wakil bank indonesia provinsi bal rabu

[4] https://github.com/sastrawi/sastrawi
[5] http://kateglo.com/

4.2.2. Uji Coba RPGSO

Random Projection – Gram Schmidt Orthogonalization (RP-GSO) merupakan proses perangkingan kata yang menggunakan teknik ortogonalisasi untuk mendapatkan kata-kata yang relevan serta dapat mewakili dokumen. Pada metode ini terdapat sebuah parameter yang perlu disetting pada awal proses, yakni dimensi baris matriks proyeksi acak.

Berdasarkan percobaan sebelumnya, penentuan nilai dimensi baris matriks proyeksi acak tidak berpengaruh terhadap urutan fitur yang dihasilkan (Wang, Zhang, Liu, Liu, & Wang, 2016). Pada uji coba ini, dimensi baris matriks proyeksi acak yang digunakan adalah 200 sesuai dengan parameter pada penelitian sebelumnya (Wang, Zhang, Liu, Liu, & Wang, 2016) (Arora, et al., 2013).

Pada akhir metode *Random Projection – Gram Schmidt Orthogonalization* (RP-GSO) akan didapatkan urut-urutan fitur yang menunjukkan tingkat kepentingan fitur terhadap dataset dokumen secara keseluruhan. Pada dataset pertama yang terdiri dari 100 dokumen, total fitur yang ada sebanyak 3461 fitur. Urut-urutan 10 fitur teratas beserta banyaknya dokumen yang memuat fitur tersebut beserta bobot rata-rata untuk dataset ini seperti pada Tabel 4.2.

Tabel 4.2 Urutan 10 fitur penting teratas pada dataset pertama

Urutan	Fitur	Norm Hasil Ortogonalisasi	Dokumen yang memiliki	Kategori Dokumen	Bobot Rata-Rata
1	Film	62280.92	20	Entertainment	100.50
2	Windows	25578.84	19	Teknologi	28.43
3	Silat	23199.95	10	Entertainment	78.10
4	Aktor	16761.54	19	Entertainment	23.27
5	Best	15973.71	7	Entertainment	48.22
6	Williams	9458.80	2	Entertainment	97.33
7	Partai	9250.17	16	Politik	17.26
8	Freeport	8561.70	9	Ekonomi	22.51
9	Kontrak	8380.47	10	Ekonomi, Olahraga	16.66
10	Cortana	8102.83	4	Teknologi	33.96

Pada Tabel 4.2 dapat dilihat bahwa fitur kata yang memiliki tingkat kepentingan tertinggi adalah fitur kata “film” yang dimiliki oleh 20 dokumen pada

kategori entertainment. Ini berarti dari 20 dokumen pada kategori tersebut, semuanya mengandung fitur “film”. Bobot rata-rata untuk fitur “film” pada 20 dokumen yang memuatnya pun cukup tinggi, yakni 100.50. Fitur “windows” dan “aktor” yang berada pada urutan 2 dan 4 dimiliki oleh 19 dokumen pada kategori entertainment dan teknologi, yang berarti hanya 1 dokumen pada kategori tersebut yang tidak mengandung fitur kata kata “windows” dan “aktor”. Fitur kata “williams” dan “cortana” hanya dimiliki oleh 2 dan 4 dokumen pada kategori entertainment dan teknologi, namun bobot rata-ratanya cukup tinggi yakni 97.33 dan 33.36.

Urut-urutan 10 fitur terbawah beserta banyaknya dokumen yang memuat fitur tersebut dan bobot rata-rata untuk dataset pertama seperti terlihat pada Tabel 4.3.

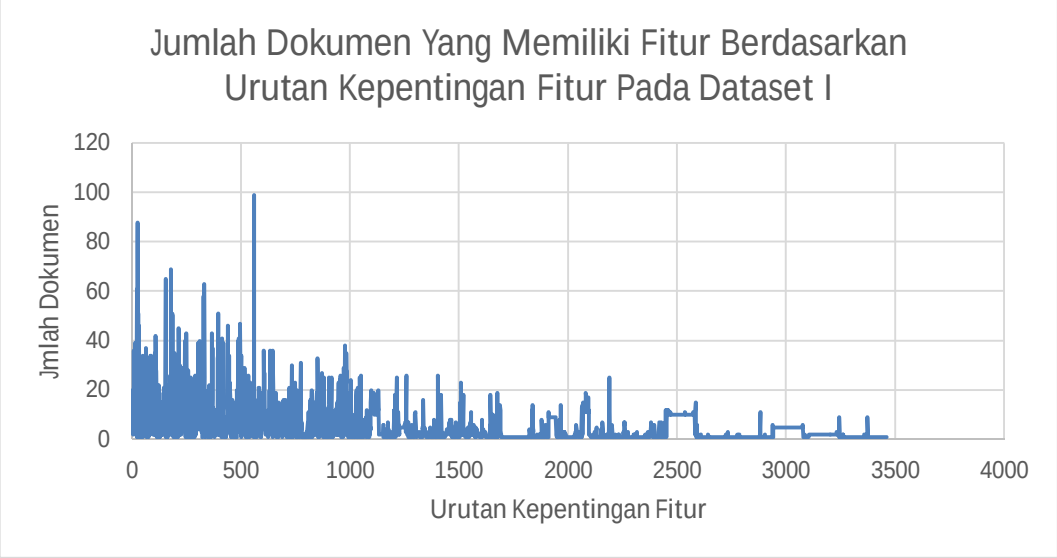
Tabel 4.3 Urutan 10 fitur penting terbawah pada dataset pertama

Urutan	**Fitur**	**Norm Hasil Ortogonalisasi**	**Dokumen yang memiliki**	**Kategori Dokumen**	**Bobot Rata-Rata**
3461	Soetomo	1.00	1	Entertainment	2.00
3460	Reuters	2.78	1	Teknologi	2.00
3459	Dahono	2.97	1	Entertainment	2.00
3458	Yazid	3.08	1	Ekonomi	4.00
3457	Taipei	3.36	1	Teknologi	2.00
3456	Taqiyyah	4.25	1	Ekonomi	2.00
3455	Denpasar	4.85	1	Ekonomi	2.00
3454	Welter	5.65	1	Politik	2.00
3453	Handoyo	5.80	1	Ekonomi	2.00
3452	Sungkur	6.64	1	Ekonomi	2.00

Pada tabel 4.3 dapat dilihat bahwa sepuluh fitur kata yang memiliki tingkat kepentingan terendah hanya dimiliki oleh sebuah dokumen saja pada kategori sebuah kategori. Bobot rata-rata fitur kata tersebut hanya 2.00 dan 4.00, dimana nilai ini sangat kecil jika dibandingkan dengan bobot fitur kata yang memiliki tingat kepentingan tinggi tertinggi, yakni 100.50.

Pada dataset pertama ini, berdasarkan Tabel 4.2 dan Tabel 4.3 dapat dilihat bahwa fitur-fitur yang dianggap penting oleh algortima RP-GSO adalah fitur yang termuat tidak hanya dalam sebuah dokumen saja, tapi fitur-fitur tersebut berada dalam beberapa dokumen dan memiliki bobot rata-rata yang cukup tinggi. Dari Tabel 4.1 dan Tabel 4.2 dapat dilihat juga bahwa tiap fitur kata tersebut mewakili dokumen pada kategori tertentu. Hal ini menunjukkan bahwa fitur-fitur tersebut memiliki keterkaitan yang cukup tinggi dengan kategori dokumen.

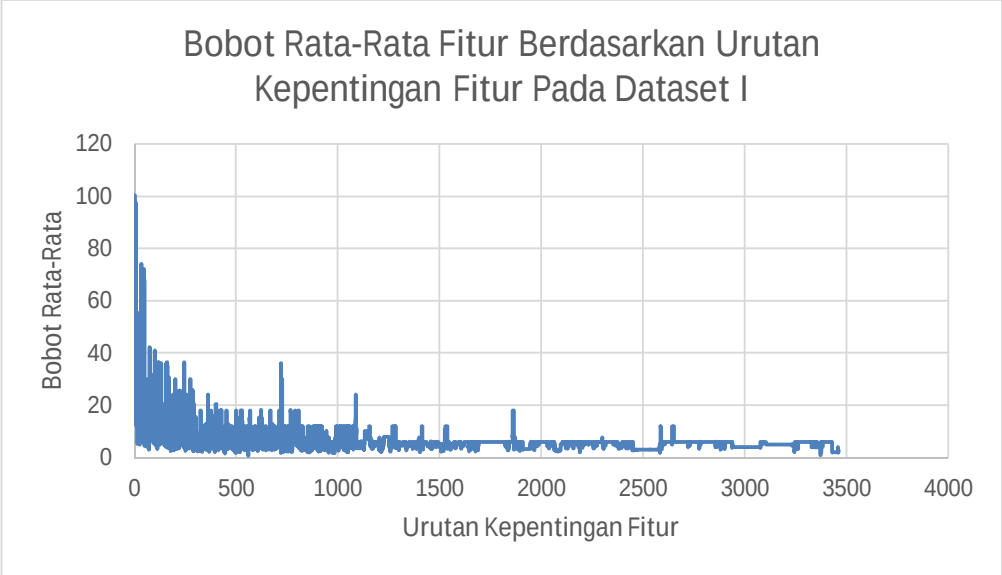
Pada Gambar 4.1 dapat dilihat urutan kepentingan fitur beserta banyaknya dokumen yang memiliki fitur tersebut pada dataset pertama. Pada dataset pertama ini terdapat total 100 dokumen pada 5 kategori dengan 3461 fitur kata.



Gambar 4.1 Jumlah dokumen yang memiliki fitur pada dataset pertama.

Dari Gambar 4.1 terlihat bahwa secara umum, fitur- fitur yang dimiliki oleh banyak dokumen, ternyata cenderung memiliki kepentingan yang lebih tinggi dibandingkan dengan fitur yang dimiliki oleh sedikit dokumen.

Pada Gambar 4.2 dapat dilihat urutan kepentingan fitur beserta bobot rata-rata fitur pada dokumen yang memiliki fitur tersebut pada dataset pertama.



Gambar 4.2 Bobot rata-rata fitur pada dataset pertama.

Dari Gambar 4.2 terlihat bahwa secara umum, fitur- fitur yang memiliki bobot tinggi cenderung memiliki kepentingan yang lebih tinggi dibanding fitur-fitur yang memiliki bobot lebih kecil.

Pada Tabel 4.4 dapat dilihat urutan kepentingan fitur beserta banyaknya dokumen yang memiliki fitur tersebut pada dataset kedua. Pada dataset kedua ini terdapat total 150 dokumen pada 5 kategori dengan 4692 fitur kata.

Tabel 4.4 Urutan 10 fitur penting teratas pada dataset kedua

Urutan	Fitur	Norm Hasil Ortogonalisasi	Dokumen yang memiliki	Kategori Dokumen	Bobot Rata-Rata
1	Film	54305.67	29	Entertainment	91.26
2	Windows	27985.66	20	Teknologi	31.79
3	Best	19406.34	13	Entertainment	37.26
4	Williams	18525.93	8	Entertainment	66.07
5	Silat	17742.78	10	Entertainment	90.31
6	Partai	15420.81	25	Politik	15.25
7	Sandra	13985.23	1	Entertainment	163.42
8	Chong	13848.37	5	Olahraga	32.21
9	Xiaoice	10294.21	1	Teknologi	104.59
10	Freeport	9802.19	10	Ekonomi	22.88

Pada tabel 4.4 dapat dilihat bahwa fitur kata yang memiliki tingkat kepentingan tertinggi adalah fitur “film” yang dimiliki oleh 29 dokumen. Ini berarti hanya 1 dokumen saja pada kategori entertainment yang tidak memuat kata “film”. Bobot rata-rata fitur “film” pada 29 dokumen tersebut pun cukup tinggi, yakni sebesar 91.26. Fitur kata selanjutnya yang memiliki tingkat kepentingan tinggi adalah “windows”, “best”, ”silat”, “partai”, dan ”freeport” yang dimiliki oleh 10 sampai 25 dokumen. Sedangkan fitur kata “williams”, “sandra”, “chong”, dan “xiaoice” memang dimiliki oleh beberapa atau bahkan 1 dokumen saja, namun bobot rata-rata fitur tersebut tergolong tinggi.

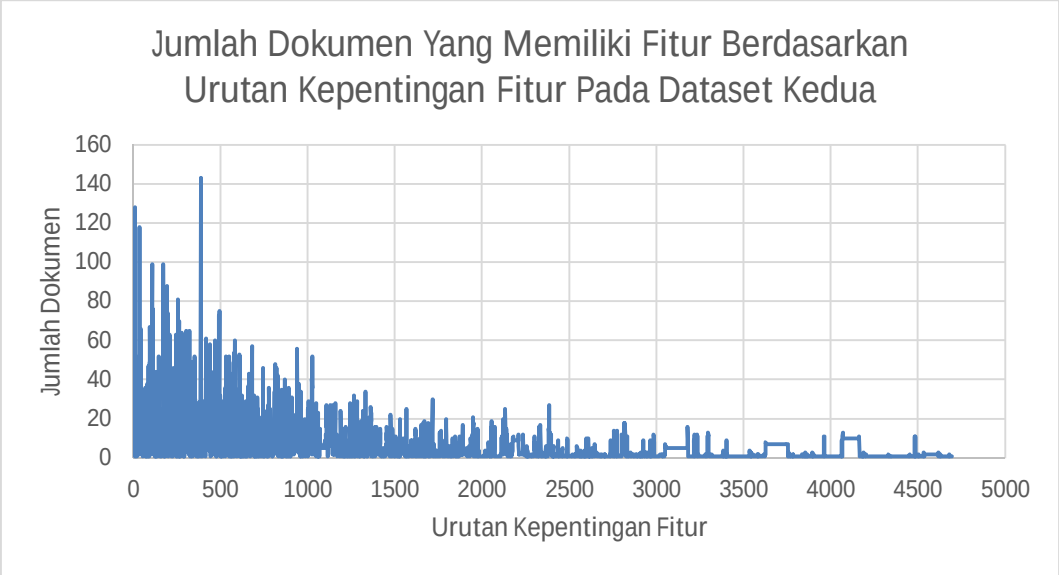
Urut-urutan 10 fitur terbawah beserta banyaknya dokumen yang memuat fitur tersebut beserta bobot rata-rata pada dokumen yang memuatnya untuk dataset kedua seperti terlihat pada Tabel 4.5.

Tabel 4.5 Urutan 10 fitur penting terbawah pada dataset kedua

Urutan	Fitur	Norm Hasil Ortogonalisasi	Dokumen yang memiliki	Kategori Dokumen	Bobot Rata-rata
4692	Soetomo	0.75	1	Entertainment	2.18
4691	Andrew	2.07	1	Teknologi	2.18
4690	Ubergizmo	2.25	1	Teknologi	4.36
4689	Canberra	2.52	1	Ekonomi	2.18
4688	Sulung	2.75	1	Entertainment	2.18
4687	Commerzbank	3.00	1	Ekonomi	2.18
4686	Shazam	3.57	1	Teknologi	2.18
4685	Yazid	3.57	1	Ekonomi	2.18
4684	Welter	3.87	1	Politik	2.18
4683	Shutterstock	3.91	1	Teknologi	2.18

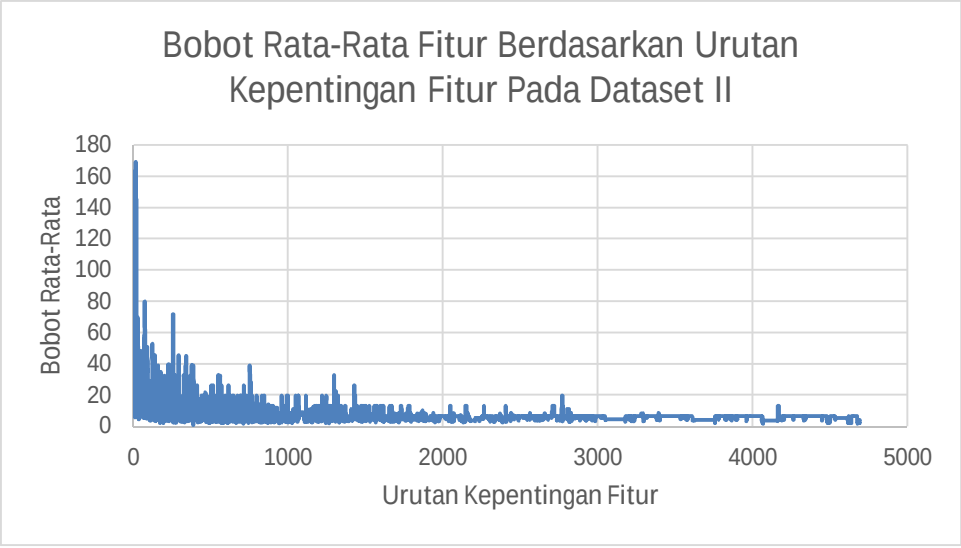
Pada tabel 4.5 dapat dilihat bahwa sepuluh fitur kata yang memiliki tingkat kepentingan terendah hanya dimiliki oleh sebuah dokumen saja pada sebuah kategori. Bobot rata-rata fitur kata tersebut hanya 2.18 dan 4.36, dimana nilai ini sangat kecil jika dibandingkan dengan bobot fitur kata yang memiliki tingat kepentingan tertinggi, yakni 91.26.

Pada Gambar 4.3 dapat dilihat urutan kepentingan fitur beserta banyaknya dokumen yang memiliki fitur tersebut pada dataset kedua. Pada dataset kedua ini terdapat total 150 dokumen pada 5 kategori dengan 4692 fitur kata.



Gambar 4.3 Jumlah dokumen yang memiliki fitur pada dataset kedua.

Gambar 4.3 memiliki karakter yang sama dengan Gambar 4.1, bahwa secara umum fitur-fitur yang dimiliki oleh banyak dokumen cenderung memiliki kepentingan yang lebih dibanding fitur-fitur yang dimiliki dokumen yang lebih sedikit. Sedangkan pada Gambar 4.4 dapat dilihat urutan kepentingan fitur beserta bobot rata-rata fitur pada dokumen yang memiliki fitur tersebut pada dataset kedua.



Gambar 4.4 Bobot rata-rata fitur pada dataset kedua.

Pada dataset kedua ini, berdasarkan Tabel 4.4 dan Tabel 4.5 serta Gambar 4.3 dan Gambar 4.4, dapat dilihat bahwa fitur-fitur yang dianggap penting oleh algortima RP-GSO adalah fitur yang termuat tidak hanya dalam sebuah dokumen saja, tapi fitur-fitur tersebut berada dalam beberapa dokumen dan memiliki bobot rata-rata yang cukup tinggi. Tabel 4.4 dan Tabel 4.5 juga memperlihatkan bahwa tiap fitur kata tersebut mewakili dokumen pada kategori tertentu. Hal ini menunjukkan bahwa fitur-fitur tersebut memiliki keterkaitan yang cukup tinggi dengan kategori dokumen.

Sedangkan pada dataset ketiga yang terdiri dari 200 dokumen, total fitur yang ada pada dataset ini sebanyak 5536 fitur. Dengan algoritma RPGSO akan didapatkan rangking fitur yang menunjukkan tingkat kepentingan fitur tersebut. Urut-urutan 10 fitur teratas untuk dataset ketiga terlihat seperti pada Tabel 4.6.

Tabel 4.6 Urutan 10 fitur penting teratas pada dataset ketiga

Urutan	Fitur	Norm Hasil Ortogonalisasi	Dokumen yang memiliki	Kategori Dokumen	Bobot Rata-rata
1	Film	46596.30	36	Entertainment	76.83
2	Windows	25174.05	21	Teknologi	33.44
3	Lawak	18235.05	7	Entertainment	121.66
4	Sandra	13278.05	1	Entertainment	172.73
5	Aplikasi	10115.17	24	Teknologi	16.12
6	Partai	9836.65	30	Politik, Entertainment	15.86
7	Apple	8780.34	18	Teknologi, Ekonomi	19.25
8	Silat	8304.76	10	Entertainment	99.16
9	Chong	8300.40	7	Olahraga	32.15
10	Masiv	7892.35	5	Entertainment	28.49

Pada tabel 4.6 terlihat bahwa fitur-fitur tersebut termuat dalam dokumen yang cukup banyak, yakni lebih dari 10 dokumen atau memiliki bobot rata-rata yang cukup tinggi, yakni lebih dari 20.

Sedangkan urut-urutan 10 fitur terbawah untuk dataset ketiga seperti terlihat pada Tabel 4.7.

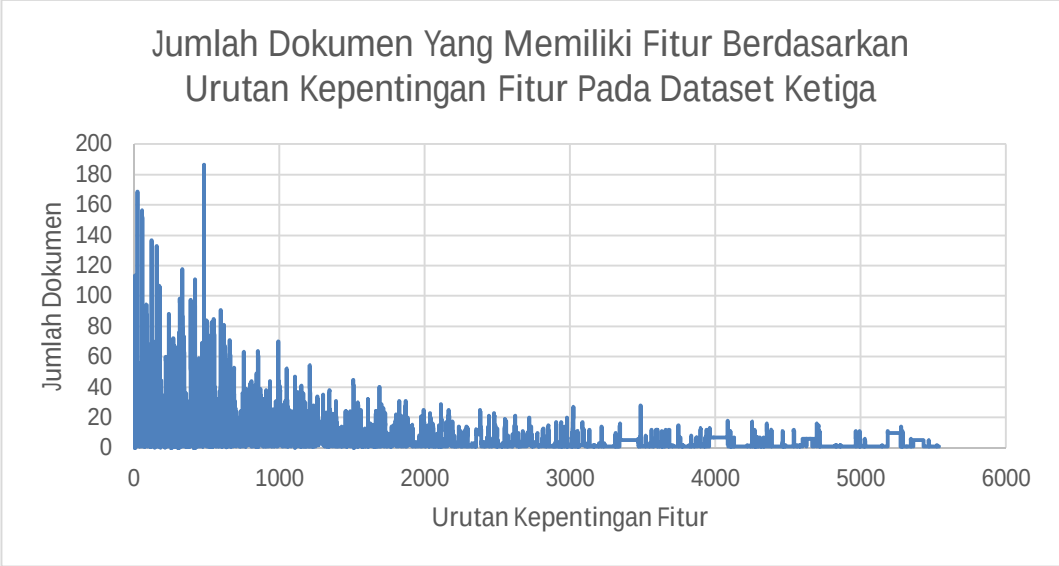
Selanjutnya pada Tabel 4.7 dapat dilihat banyaknya dokumen yang memiliki fitur berdasarkan urutan kepentingan fitur beserta bobot rata-rata pada dokumen yang memuatnya pada dataset ketiga.

Tabel 4.7 Urutan 10 fitur penting terbawah pada dataset ketiga

Urutan	Fitur	Norm Hasil Ortogonalisasi	Dokumen yang memiliki	Kategori Dokumen	Bobot Rata-Rata
5536	Soetomo	0.55	1	Entertainment	2.30
5535	Ubergizmo	2.02	1	Teknologi	2.30
5534	Babos	2.26	1	Olahraga	4.61
5533	Dahono	2.33	1	Entertainment	2.30
5532	Maniak	2.36	1	Teknologi	2.30
5531	Sulung	2.41	1	Entertainment	2.30
5530	Handoyo	2.94	1	Ekonomi	2.30
5529	Canberra	3.22	1	Ekonomi	2.30
5528	Yazid	3.34	1	Ekonomi	2.30
5527	Mustopa	3.37	1	Politik	2.30

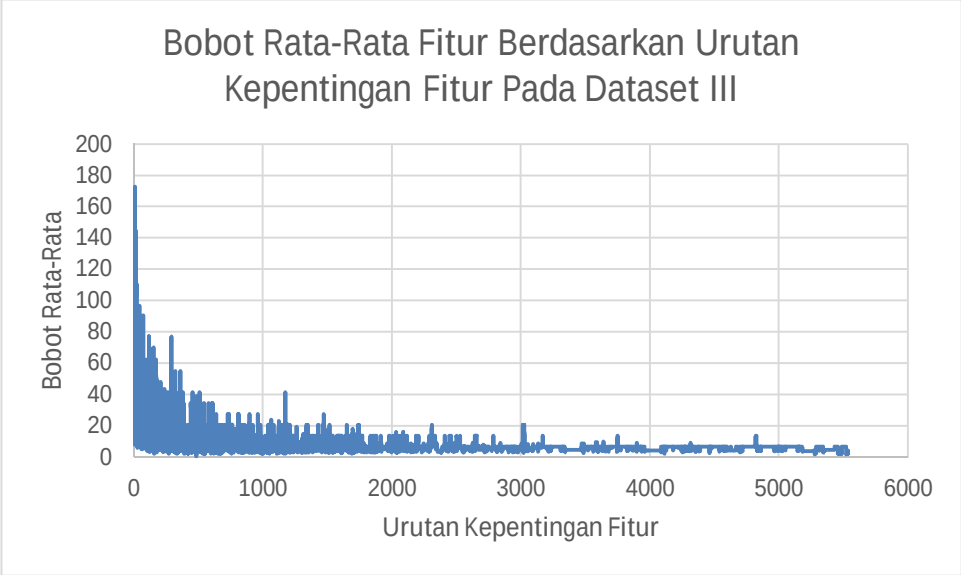
Pada dataset ketiga ini, berdasarkan Tabel 4.6 dan Tabel 4.7 telihat bahwa fitur-fitur yang dianggap penting oleh algortima RP-GSO adalah fitur yang termuat tidak hanya dalam sebuah dokumen saja, tapi fitur-fitur tersebut berada dalam beberapa dokumen. Selain keberadaan fitur pada dokumen, faktor lain yang berpengaruh pada tingkat kepentingan fitur adalah bobot fitur-fitur tersebut pada dokumen yang memuatnya.

Pada Gambar 4.5 dapat dilihat urutan kepentingan fitur beserta banyaknya dokumen yang memiliki fitur tersebut pada dataset ketiga. Pada dataset ketiga ini terdapat total 200 dokumen pada 5 kategori dengan 5536 fitur kata.



Gambar 4.5 Jumlah dokumen yang memiliki fitur pada dataset ketiga.

Pada Gambar 4.6 dapat dilihat bobot rata-rata fitur pada dokumen yang memilikinya berdasarkan urutan kepentingan fitur tersebut pada dataset ketiga.



Gambar 4.6 Jumlah dokumen yang memiliki fitur pada dataset ketiga.

Dari Tabel 4.2 sampai Tabel 4.7, serta dari Gambar 4.1 sampai Gambar 4.6 dapat dilihat hasil proses *Random Projection – Gram Schmidt Orthogonalization* (RPGSO) terhadap dataset pertama, kedua dan ketiga. Secara umum keberadaan fitur dalam sejumlah dokumen menjadi salah satu hal yang menentukan apakan fitur tersebut berada pada urutan sangat penting atau tidak. Semakin banyak jumlah

dokumen yang memiliki fitur tersebut, maka kemungkinan fitur tersebut semakin penting. Sebaliknya, semakin sedikit jumlah dokumen yang memiliki fitur tersebut, maka kemungkinan fitur tersebut semakin kurang penting.

Hal lain yang menentukan apakan fitur tersebut berada pada urutan sangat penting atau tidak adalah bobot fitur pada dokumen yang mengandung fitur tersebut. Semakin besar bobot fitur tersebut, maka kemungkinan fitur tersebut semakin penting. Sebaliknya, semakin kecil bobot fitur tersebut, maka kemungkinan fitur tersebut semakin kurang penting. Padahal bobot fitur tergantung pada jumlah fitur kata tersebut pada dokumen yang mengandungnya. Artinya, semakin banyak jumlah fitur kata tersebut pada sebuah dokumen, maka kecenderungan fitur kata tersebut semakin penting. Sebaliknya semakin sedikit jumlah fitur kata tersebut pada sebuah dokumen, maka kecenderungan fitur kata tersebut semakin kurang penting.

Dari Tabel 4.2 sampai Tabel 4.7 juga dapat dilihat bahwa secara umum, fitur-fitur yang penting dalam dokumen tersebut dapat menjadi pembeda antara dokumen dalam sebuah kategori dengan dokumen dalam kategori lain. Hal ini menunjukkan bahwa fitur-fitur tersebut dapat menjadi penentu pada proses berikutnya, yakni proses pengelompokan dokumen.

4.2.3. Uji Coba Penentuan Jumlah Fitur Terbaik

Pada uji coba ini dilakukan proses pengelompokan dokumen dataset dengan Algoritma *Harmony Search*. Proses ini dilakukan menggunakan fitur yang telah diseleksi pada tahap sebelumnya dengan metode RPGSO. Tujuan uji coba ini adalah untuk menentukan jumlah prosentase fitur terbaik yang akan digunakan pada proses pengelompokan dokumen. Uji coba ini akan dilakukan terhadap ketiga dataset yang ada, yakni dataset 100, 150 dan 200 dokumen.

Pada Algoritma *Harmony Search* terdapat beberapa parameter yang perlu disetting, yakni *Harmony Memory Size* (HMS), *Harmony Memory Considering Rate* (HMCR), *Pitch Adjusting Rate* (PAR), serta jumlah total iterasi. Berdasarkan penelitian sebelumnya, nilai parameter HMS ditentukan senilai dua kali jumlah *cluster* dokumen dan parameter HMCR paling optimal pada saat bernilai 0.6 (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013). Untuk parameter PAR

ditentukan sesuai dengan penelitian sebelumnya yang nilainya berubah sesuai dengan jumlah iterasi saat ini dan total iterasi yang akan dilakukan (M. Mahdavi, M. Fesanghary, & E. Damangir, 2007). Sedangkan jumlah iterasi yang digunakan pada penelitian ini adalah 100 iterasi. Tabel 4.8 menunjukkan setting parameter yang digunakan pada uji coba penentuan prosentase fitur terbaik untuk ketiga dataset yang ada.

Tabel 4.8 Setting Parameter Uji Coba

Parameter	**Nilai**
Harmony Memory Size (HMS)	10
Harmony Memory Considering Rate (HMCR)	0.6
Pitch Adjusting Rate (PAR)	Sesuai iterasi
Jumlah iterasi	100

Nilai F-Measure hasil pengelompokan dokumen pada dataset pertama dengan Algoritma *Harmony Search* untuk tiap-tiap prosentase fitur tertentu dapat dilihat pada Tabel 4.9.

Tabel 4.9 F-Measure Dataset Pertama

Prosentase Jumlah Fitur	**F-Measure Hasil Percobaan**					
	1	**2**	**3**	**4**	**5**	**Rata-rata**
100	0.7382	0.7747	0.7485	0.7496	0.7559	0.7534
90	0.7845	0.7782	0.7467	0.7542	**0.7873**	0.7704
80	0.7582	0.7767	0.7863	0.7737	0.7768	0.7743
70	0.7851	0.7676	0.7646	**0.7851**	0.7649	0.7735
60	0.7622	0.7604	0.7927	0.7703	0.7427	0.7657
50	0.7756	0.7302	0.7433	0.7436	0.7392	0.7464
40	0.7522	0.7618	0.7258	0.7305	0.7508	0.7442
30	**0.8031**	**0.8745**	**0.8354**	0.7203	0.7064	**0.7879**
20	0.7524	0.662	0.7043	0.6822	0.713	0.7028
10	0.7409	0.7323	0.7256	0.7242	0.7336	0.7313
5	0.4839	0.5488	0.5783	0.5093	0.5559	0.5352

Pada Tabel 4.9 terlihat bahwa pada percobaan pertama, kedua, dan ketiga, proses *clustering* dengan 30% jumlah fitur memiliki nilai F-Measure yang lebih besar, bahkan jika dibandingkan dengan menggunakan keseluruhan fitur. Sedangkan pada percobaan keempat dan kelima, nilai F-Measure terbesar pada saat proses *clustering* menggunakan 70% dan 90% dari keseluruhan fitur yang ada. Dari

kelima percobaan yang dilakukan, dapat disimpulkan bahwa prosentase jumlah fitur terbaik untuk proses *clustering* dengan Algoritma *Harmony Search* pada dataset pertama adalah 30% dari semua fitur yang ada pada dataset. Pada saat menggunakan 30% dari keseluruhan fitur, rata-rata nilai F-Measure yang dihasilkan dari proses *clustering* adalah 0.7879, yang berarti lebih besar sebanyak 4.6% daripada menggunakan keseluruhan fitur, yakni 0.7534.

Nilai F-Measure hasil pengelompokan dokumen pada dataset kedua dengan Algoritma *Harmony Search* untuk tiap-tiap prosentase fitur tertentu dapat dilihat pada Tabel 4.10.

Tabel 4.10. F-Measure Dataset Kedua

Prosentase Jumlah Fitur	**F-Measure Hasil Percobaan**					
	1	**2**	**3**	**4**	**5**	**Rata-rata**
100	0.5448	0.7313	0.7273	0.7318	0.6611	0.6793
90	0.6911	**0.7533**	0.7285	**0.7501**	0.5572	0.6960
80	0.7096	0.6502	**0.7299**	0.7256	**0.6937**	**0.7018**
70	0.6949	0.6136	0.6585	0.7199	0.6438	0.6661
60	0.7046	0.5449	0.6949	0.6871	0.6673	0.6598
50	**0.7181**	0.6926	0.5976	0.6015	0.6442	0.6508
40	0.6852	0.4749	0.5027	0.6499	0.6887	0.6003
30	0.5834	0.654	0.6591	0.5959	0.5155	0.6016
20	0.6443	0.6611	0.6779	0.5775	0.6779	0.6477
10	0.5779	0.6315	0.6112	0.6373	0.5911	0.6098
5	0.6455	0.6272	0.6457	0.6522	0.6387	0.6419

Pada Tabel 4.10 terlihat bahwa pada percobaan pertama, proses *clustering* dengan 50% jumlah fitur memiliki nilai F-Measure yang paling besar. Sedangkan pada percobaan kedua dan keempat, nilai F-Measure terbesar pada saat proses *clustering* menggunakan 90% dari keseluruhan fitur yang ada. Pada percobaan ketiga, kelima, serta rata-rata dari kelima percobaan yang dilakukan, nilai F-Measure terbesar adalah pada saat menggunakan 80% fitur yang ada. Dapat disimpulkan bahwa prosentase jumlah fitur terbaik untuk proses *clustering* dengan Algoritma *Harmony Search* pada dataset kedua adalah 80% dari semua fitur. Pada saat tersebut rata-rata nilai F-Measure yang dihasilkan dari proses *clustering* adalah 0.7018 yang lebih besar 3.3% daripada menggunakan keseluruhan fitur, yakni 0.6793.

Nilai F-Measure hasil pengelompokan dokumen dengan Algoritma *Harmony Search* pada dataset ketiga untuk tiap-tiap prosentase fitur tertentu dapat dilihat pada Tabel 4.11.

Tabel 4.11. F-Measure Dataset Ketiga

Prosentase Jumlah Fitur	**F-Measure Hasil Percobaan**					
	1	**2**	**3**	**4**	**5**	**Rata-rata**
100	0.527	0.5567	0.5411	0.5469	0.5589	0.5461
90	0.5493	0.5515	0.5626	0.5726	0.7265	0.5925
80	0.5518	0.6975	0.556	0.6832	0.5426	0.6062
70	0.5533	0.7166	0.5707	0.7523	**0.7631**	0.6712
60	0.6799	**0.7672**	0.5466	**0.7539**	0.5729	0.6641
50	0.7331	0.7546	0.6943	0.7446	0.6806	**0.7214**
40	0.5418	0.7322	**0.7316**	0.6387	0.5417	0.6372
30	0.5512	0.5662	0.6021	0.5305	0.5274	0.5555
20	**0.7381**	0.5778	0.4423	0.6804	0.7083	0.6294
10	0.5077	0.5259	0.6628	0.6262	0.6109	0.5867
5	-	-	-	-	-	-

Pada Tabel 4.11 terlihat pada saat prosentase fitur 5%, tidak dapat dilakukan pengelompokan karena fitur terlalu sedikit sehingga kemiripan antar dokumen sangat kecil.

Pada Tabel 4.11 juga terlihat bahwa rata-rata nilai F-Measure terbesar terjadi saat proses menggunakan 50% dari keseluruhan fitur yang ada, meski tidak mendominasi tiap percobaan yang dilakukan. Dapat disimpulkan bahwa prosentase jumlah fitur terbaik untuk proses *clustering* dengan Algoritma *Harmony Search* pada dataset ketiga adalah 50% dari semua fitur. Pada saat tersebut rata-rata nilai F-Measure yang dihasilkan dari proses *clustering* adalah 0.7214 yang lebih besar 32% daripada menggunakan keseluruhan fitur yang nilainya 0.5461.

Dari Tabel 4.9, Tabel 4.10 dan Tabel 4.11 dapat disimpulkan bahwa dari 5 buah percobaan yang dilakukan untuk masing-masing dataset, didapatkan rata-rata F-Measure tertinggi untuk dataset pertama terjadi pada saat 30% fitur digunakan pada proses pengelompokan dokumen. Sedangkan untuk dataset kedua, nilai rata-rata F-Measure tertinggi terjadi pada saat 80% fitur digunakan, dan untuk dataset ketiga nilai rata-rata F-Measure tertinggi terjadi pada saat 50% fitur digunakan. Tabel 4.12 memperlihatkan prosentase optimal untuk masing-masing dataset.

Tabel 4.12. Prosentase Fitur Optimal Dibandingkan Seluruh Fitur

Dataset	Prosentase Optimal (%)	F-Measure Prosentase Optimal	F-Measure Dengan Seluruh Fitur
Pertama	30	0.7879	0.7534
Kedua	80	0.7018	0.6793
Ketiga	50	0.7214	0.5461
Rata-rata	__	0.7370	0.6596

Dari Tabel 4.12 dapat dilihat bahwa untuk ketiga dataset, pengelompokan dokumen dengan Algoritma *Harmony Search* menggunakan prosentase fitur optimal memiliki F-Measure yang rata-rata lebih tinggi dibandingkan dengan menggunakan seluruh fitur. Hasil ini kemudian digunakan untuk uji coba dalam menentukan variasi parameter terbaik.

4.2.4. Uji Coba Penentuan Variasi Parameter Terbaik

Pada uji coba ini akan dilakukan proses pengelompokan dokumen dengan Algoritma *Harmony Search* dengan variasi parameter (*Harmony Memory Size*) HMS dan *Harmony Memory Consideration Rate* (HMCR). Tujuan dari uji coba ini adalah untuk mengetahui nilai parameter terbaik sehingga menghasilkan hasil kelompok dokumen dengan nilai F-Measure yang besar. Uji coba ini akan dilakukan terhadap ketiga dataset yang ada, yakni dataset 100, 150 dan 200 dokumen. Prosentase fitur yang digunakan pada uji coba ini mengacu pada uji coba sebelumnya seperti pada Tabel 4.12.

Pada Tabel 4.13 berikut ini dapat dilihat nilai rata-rata F-Measure yang dihasilkan dari variasi HMS dan HMCR pada dataset pertama. Pada Tabel 4.13 tersebut, nilai HMS yang menghasilkan rata-rata nilai F-Measure terbesar adalah 1, kemudian 2 lalu 5. HMS adalah jumlah pilihan kombinasi yang dijadikan sumber untuk penyesuaian sehingga menghasilkan nilai baru yang lebih baik berdasarkan nilai *fitness* yang digunakan. Semakin besar nilai HMS, semakin banyak kombinasi yang memungkinkan untuk menghasilkan nilai baru, sebaliknya semakin kecil nilai HMS, semakin sedikit kombinasi yang memungkinkan untuk menghasilkan nilai

baru. Apabila menggunakan nilai HMS 1 atau 2, maka kombinasi nilai baru hanya berasal dari 1 atau 2 nilai sebelumnya sehingga F-Measure sangat tergantung pada nilai random awal. Mengacu hasil percobaan pada tabel 1 dan tabel 2, serta memperhatikan kemungkinan jumlah kombinasi baru, maka nilai HMS yang sebaiknya digunakan sebagai parameter terbaik adalah 5.

Tabel 4.13. F-Measure Variasi HMS dan HMCR pada Dataset Pertama

HMS	**Rata-Rata Nilai F-Measure**									**Rerata**
	HMCR									
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	
1	0.6757	0.6378	0.6756	0.7400	0.7514	0.8244	0.8067	0.854	0.7499	0.7462
2	0.6241	0.6292	0.6642	0.6599	0.7501	0.8459	0.8348	0.7884	0.7594	0.7284
3	0.5955	0.618	0.6976	0.6986	0.7379	0.7835	0.7704	0.7798	0.7968	0.7198
4	0.6048	0.5603	0.6365	0.6812	0.7451	0.7415	0.7777	0.7528	0.8433	0.7048
5	0.5561	0.6358	0.6143	0.6999	0.7487	0.7804	0.8205	0.781	0.8591	**0.7218**
6	0.5464	0.5748	0.608	0.6735	0.7278	0.8421	0.769	0.8208	0.8013	0.7071
7	0.5566	0.6373	0.5946	0.6716	0.729	0.7687	0.7829	0.8841	0.7757	0.7112
8	0.6029	0.5308	0.5869	0.7033	0.6898	0.7317	0.8162	0.8011	0.7839	0.6941
9	0.5643	0.5788	0.5899	0.6665	0.7596	0.7524	0.7718	0.7996	0.7491	0.6924
10	0.5578	0.6001	0.5935	0.623	0.6537	0.7618	0.7911	0.7806	0.7796	0.6824
11	0.5846	0.5211	0.5962	0.5968	0.7053	0.741	0.7441	0.7798	0.7950	0.6738
12	0.5513	0.5538	0.5616	0.6276	0.6921	0.7131	0.7948	0.7967	0.7730	0.6738
13	0.5553	0.5497	0.5657	0.641	0.6906	0.7786	0.8088	0.7572	0.7900	0.6819

Pada tabel tersebut, dapat dilihat bahwa nilai F-Measure terbesar diperoleh pada saat nilai HMCR mulai 0.6 sampai 0.9, sedangkan Nilai HMCR lainnya akan menghasilkan nilai F-Measure yang relatif lebih kecil. Mengacu pada kesimpulan diatas, bahwa nilai HMS terbaik adalah 5, maka nilai HMCR yang menghasilkan rata-rata nilai F-Measure terbesar untuk HMS 5 adalah 0.9. Dari hasil tersebut dapat disimpulkan bahwa pada dataset pertama ini nilai HMCR terbaik adalah 0.9.

Pada Tabel 4.14 berikut ini dapat dilihat rata-rata nilai F-Measure yang dihasilkan dari variasi HMS dan HMCR pada dataset kedua. Pada Tabel 4.14 tersebut terlihat bahwa nilai HMCR yang menghasilkan rata-rata nilai F-Measure terbesar adalah 0.8. yang lebih besar dibanding nilai HMCR yang lain. Dari hasil tersebut dapat disimpulkan bahwa pada dataset kedua nilai parameter HMCR terbaik adalah 0.8.

Tabel 4.14 F-Measure Variasi HMS dan HMCR pada Dataset Kedua

HMS	Rata-Rata Nilai F-Measure				
	HMCR				Rata-rata
	0.6	0.7	0.8	0.9	
1	0.7468	0.7815	0.7603	0.7223	**0.7527**
2	0.756	0.7527	0.7649	0.7273	**0.7502**
3	0.7188	0.7085	0.7628	0.6775	0.7169
4	0.7392	0.7549	0.7343	0.7022	0.7327
5	0.7292	0.7635	0.7226	0.7529	**0.7421**
6	0.7266	0.7274	0.7619	0.7303	0.7366
7	0.712	0.7415	0.7461	0.7282	0.7320
8	0.7197	0.7492	0.7223	0.7224	0.7284
9	0.6829	0.7347	0.7108	0.7293	0.7144
10	0.7216	0.698	0.7451	0.7332	0.7245
11	0.7036	0.707	0.7021	0.6444	0.6893
12	0.7133	0.7269	0.753	0.7391	0.7331
Rata-rata	0.7225	0.7372	**0.7405**	0.7174	0.7294

Pada Tabel 4.14 tersebut, nilai HMS yang menghasilkan rata-rata nilai F-Measure terbesar adalah 1, kemudian nilai 2, dan nilai 5. Berdasarkan penjelasan seperti pada dataset pertama, bahwa nilai HMS merupakan jumlah pilihan kombinasi yang dijadikan sumber untuk penyesuaian sehingga menghasilkan nilai baru yang lebih baik serta mengacu hasil percobaan pada tabel diatas, maka nilai HMS yang digunakan sebagai parameter terbaik adalah 5.

Tabel 4.15. F-Measure Variasi HMS dan HMCR pada Dataset Ketiga

HMS	Rata-Rata Nilai F-Measure				
	HMCR				Rata-rata
	0.6	0.7	0.8	0.9	
4	0.7257	0.6089	0.5925	0.6239	0.6378
5	0.6305	0.6462	0.6883	0.6325	0.6494
6	0.6697	0.6608	0.5897	0.6746	0.6487
7	0.6247	0.5699	0.6088	0.6534	0.6142
8	0.6340	0.6955	0.6311	0.6148	0.6439
9	0.6637	0.6745	0.6290	0.6398	**0.6518**
10	0.6681	0.6400	0.6569	0.6243	0.6473
Rata-rata	**0.6595**	0.6423	0.6280	0.6376	0.6419

Selanjutnya pada Tabel 4.15 berikut ini dapat dilihat rata-rata nilai *F-Measure* yang dihasilkan dari variasi HMS dan HMCR pada dataset ketiga.

Pada Tabel 4.15 tersebut terlihat bahwa nilai HMCR yang menghasilkan rata-rata nilai F-Measure terbesar adalah 0.6. Nilai ini merupakan nilai yang paling besar dibanding nilai HMCR yang lain. Dari hasil tersebut dapat disimpulkan bahwa pada dataset kedua nilai parameter HMCR terbaik adalah 0.6.

Pada Tabel 4.15 tersebut, nilai HMS yang menghasilkan rata-rata nilai F-Measure terbesar adalah 9. Berdasarkan hasil percobaan pada tabel diatas, maka nilai HMS yang digunakan sebagai parameter terbaik adalah 9.

Dari Tabel 4.13, Tabel 4.14 dan Tabel 4.15 dapat disimpulkan bahwa dari 5 buah percobaan yang dilakukan untuk masing-masing dataset dengan variasi HMS dan HMCR, didapatkan rata-rata *F-Measure* tertinggi untuk dataset pertama terjadi pada saat HMS sebesar 5 dan HMCR sebesar 0.9. Sedangkan untuk dataset kedua, nilai rata-rata *F-Measure* tertinggi terjadi pada saat HMS sebesar 5 dan HMCR sebesar 0.8, dan untuk dataset ketiga nilai rata-rata *F-Measure* tertinggi terjadi pada saat HMS sebesar 9 dan HMCR sebesar 0.6 digunakan. Tabel 4.16 memperlihatkan variasi parameter optimal untuk masing-masing dataset.

Tabel 4.16. Hasil Uji Coba Variasi Parameter Optimal

Dataset	**HMS**	**HMCR**
Pertama	5	0.9
Kedua	5	0.8
Ketiga	9	0.6

4.2.5. Uji Coba Menggunakan Parameter Terbaik

Berdasarkan uji coba untuk mendapatkan parameter optimal pada masing-masing dataset seperti pada Tabel 4.16, pada uji coba ini akan dilakukan proses pengelompokan dokumen dengan Algoritma *Harmony Search*. Untuk parameter PAR ditentukan sesuai dengan penelitian sebelumnya yang nilainya berubah sesuai dengan jumlah iterasi saat ini dan total iterasi yang akan dilakukan (M. Mahdavi, M. Fesanghary, & E. Damangir, 2007). Sedangkan jumlah iterasi yang digunakan pada penelitian ini adalah 100 iterasi.

Tujuan dari uji coba ini adalah untuk mengetahui nilai *F-Measure* tertinggi untuk masing-masing dataset dengan prosentase fitur tertentu. Uji coba ini akan dilakukan terhadap ketiga dataset yang ada, yakni dataset 100, 150 dan 200 dokumen.

Nilai *F-Measure* hasil pengelompokan dokumen pada dataset pertama dengan Algoritma *Harmony Search* dengan parameter optimal dapat dilihat pada Tabel 4.17.

Tabel 4.17. Hasil Uji Coba Parameter Optimal Dataset Pertama

Prosentase Fitur	**Nilai F-measure Percobaan Ke**					
	1	**2**	**3**	**4**	**5**	**Rata-rata**
100	0.8000	0.8000	0.7994	0.7968	0.7927	0.7978
90	0.8000	0.7778	0.7786	0.7843	0.7921	0.7866
80	0.9079	0.7739	0.7927	0.8870	0.7976	0.8318
70	0.7903	0.8968	0.8981	0.9067	0.9699	**0.8924**
60	0.7903	0.7846	0.8281	0.7748	0.7873	0.7930
50	0.7846	0.7843	0.7846	0.7846	0.7912	0.7859
40	0.7775	0.7799	0.7730	0.7946	0.7873	0.7825
30	0.7735	0.9286	0.7754	0.7191	0.9167	0.8227
20	0.8682	0.6022	0.7409	0.6304	0.8756	0.7435
10	0.6775	0.7385	0.7566	0.7518	0.7495	0.7348
5	0.5162	0.4964	0.5451	0.5248	0.5969	0.5359

Dari lima kali percobaan yang dilakukan terhadap dataset pertama, diketahui bahwa nilai rata-rata F-Measure tertinggi didapatkan pada saat prosentase fitur 70%. Pada saat tersebut menghasilkan nilai F-Measure sebesar 0.8924. Nilai rata-rata F-Measure tertinggi selanjutnya adalah 0.8318 yang didapatkan pada saat prosentase fitur 80%. Rata-rata F-Measure tertinggi berikutnya didapatkan pada saat prosentase fitur yang digunakan untuk pengelompokan dokumen sebesar 30%. Pada saat tersebut rata-rata nilai F-Measure yang dihasilkan sebesar 0.8227.

Ketiga nilai rata-rata F-Measure tersebut bahkan lebih tinggi dibandingkan dengan nilai rata-rata F-Measure pada saat menggunakan keseluruhan fitur yang ada. Jika menggunakan 100% fitur, rata-rata F-Measure yang dihasilkan sebesar 0.7978. Hal ini menunjukkan bahwa pengelompokan dokumen yang menggunakan

fitur terpilih berdasarkan RPGSO mampu menghasilkan kelompok dokumen yang lebih baik jika dibandingkan menggunakan keseluruhan fitur.

Nilai *F-Measure* hasil pengelompokan dokumen pada dataset kedua dengan Algoritma *Harmony Search* dengan parameter optimal dapat dilihat pada Tabel 4.18.

Tabel 4.18. Hasil Uji Coba Parameter Optimal Dataset Kedua

Prosentase Fitur	**Nilai F-measure Percobaan Ke**					
	1	**2**	**3**	**4**	**5**	**Rata-rata**
100	0.7521	0.7587	0.7704	0.7781	0.7658	0.7650
90	0.7392	0.6586	0.7291	0.8128	0.7147	0.7309
80	0.7450	0.9054	0.7657	0.7574	0.7395	**0.7826**
70	0.6937	0.5688	0.7513	0.5886	0.6216	0.6448
60	0.7352	0.7693	0.7373	0.5251	0.6917	0.6917
50	0.7139	0.7045	0.5787	0.7363	0.7403	0.6947
40	0.6853	0.7235	0.7307	0.6032	0.7391	0.6964
30	0.6247	0.6066	0.5466	0.6682	0.5677	0.6028
20	0.7279	0.7299	0.6975	0.5516	0.7119	0.6838
10	0.5987	0.6210	0.6409	0.6167	0.6427	0.6240
5	0.6868	0.6644	0.6425	0.6480	0.6430	0.6569

Dari tabel 4.18 tersebut dapat diketahui bahwa nilai rata-rata F-Measure tertinggi didapatkan pada saat prosentase fitur 80%. Pada saat tersebut menghasilkan nilai F-Measure sebesar 0.7826. Nilai rata-rata F-Measure tertinggi selanjutnya adalah 0.7650 yang didapatkan pada saat prosentase fitur 100%. Rata-rata F-Measure tertinggi berikutnya didapatkan pada saat prosentase fitur yang digunakan untuk pengelompokan dokumen sebesar 90%. Pada saat tersebut rata-rata nilai F-Measure yang dihasilkan sebesar 0.7309.

Nilai rata-rata F-Measure tertinggi bahkan lebih tinggi dibandingkan dengan nilai rata-rata F-Measure pada saat menggunakan keseluruhan fitur yang ada. Jika menggunakan 100% fitur, rata-rata F-Measure yang dihasilkan sebesar 0.7650.

Nilai *F-Measure* hasil pengelompokan dokumen pada dataset ketiga dengan Algoritma *Harmony Search* dengan parameter optimal dapat dilihat pada Tabel 4.19.

Tabel 4.19. Hasil Uji Coba Parameter Optimal Dataset Ketiga

Prosentase Fitur	Nilai F-measure Percobaan Ke					
	1	2	3	4	5	Rata-rata
100	0.5613	0.5666	0.7195	0.7147	0.5382	0.6201
90	0.7628	0.7578	0.5614	0.5543	0.5913	0.6455
80	0.7621	0.7376	0.7381	0.7552	0.5568	0.7100
70	0.7605	0.6976	0.5590	0.5470	0.7681	0.6664
60	0.6152	0.5555	0.7296	0.7522	0.7235	0.6752
50	0.7481	0.7416	0.7286	0.7875	0.5707	**0.7153**
40	0.7354	0.7193	0.5649	0.5694	0.7484	0.6675
30	0.7428	0.5450	0.6330	0.5923	0.5645	0.6155
20	0.7260	0.5953	0.7158	0.6241	0.7115	0.6745
10	0.6675	0.6551	0.6885	0.5140	0.6751	0.6400

Dari tabel 4.19 tersebut dapat diketahui bahwa nilai rata-rata F-Measure tertinggi didapatkan pada saat prosentase fitur 50%. Pada saat tersebut menghasilkan nilai F-Measure sebesar 0.7153. Nilai rata-rata F-Measure tertinggi selanjutnya adalah 0.7100 yang didapatkan pada saat prosentase fitur 80%. Rata-rata F-Measure tertinggi berikutnya didapatkan pada saat prosentase fitur yang digunakan untuk pengelompokan dokumen sebesar 60%. Pada saat tersebut rata-rata nilai F-Measure yang dihasilkan sebesar 0.6752.

Dari lima percobaan yang dilakukan terhadap dataset ketiga ini, hampir semua nilai rata-rata F-Measure dengan prosentase fitur kurang dari 100% bahkan lebih tinggi dibandingkan dengan nilai rata-rata F-Measure pada saat menggunakan keseluruhan fitur yang ada. Jika menggunakan 100% fitur, rata-rata F-Measure yang dihasilkan sebesar 0.6201. Nilai tersebut hanya lebih baik jika dibandingkan dengan rata-rata F-Measure pada saat 30% fitur. Hal ini menunjukkan bahwa pengelompokan dokumen yang menggunakan fitur terpilih berdasarkan RPGSO mampu menghasilkan kelompok dokumen yang lebih baik jika dibandingkan menggunakan keseluruhan fitur.

Dari Tabel 4.17, Tabel 4.18 dan Tabel 4.19 dapat disimpulkan bahwa dari 5 buah percobaan yang dilakukan untuk masing-masing dataset, didapatkan rata-rata F-Measure tertinggi untuk dataset pertama terjadi pada saat 70% fitur digunakan

pada proses pengelompokan dokumen. Sedangkan untuk dataset kedua, nilai rata-rata F-Measure tertinggi terjadi pada saat 80% fitur digunakan, dan untuk dataset ketiga nilai rata-rata F-Measure tertinggi terjadi pada saat 50% fitur digunakan. Tabel 4.20 memperlihatkan prosentase dengan parameter optimal untuk masing-masing dataset.

Tabel 4.20. Prosentase Fitur Dan Parameter Optimal Dibandingkan menggunakan Seluruh Fitur

Dataset	Prosentase Fitur(%)	F-Measure		
		Fitur dan Parameter Optimal	Seluruh Fitur	Perbandingan Optimal dengan Seluruh Fitur (%)
Pertama	70	0.8924	0.7978	11.86
Kedua	80	0.7826	0.7650	2.30
Ketiga	50	0.7153	0.6201	15.35
Rata-rata		0.7968	0.7276	9.50

Dari Tabel 4.12 dapat dilihat bahwa untuk ketiga dataset, Algoritma *Harmony Search* menggunakan prosentase fitur optimal memiliki F-Measure yang rata-rata lebih tinggi 9.50% dibandingkan dengan menggunakan seluruh fitur.

4.2.6. Perbandingan dengan Metode K-Means

Pada uji coba ini dilakukan clustering dokumen dengan algoritma K-Means dan Algoritma *Harmony Search* menggunakan fitur-fitur yang telah diseleksi berdasarkan urutan yang dihasilkan dari proses RPGSO. Ada dua metode Algoritma K–Means yang digunakan pada uji coba, pertama Algoritma K–Means menggunakan *Cosine Similarity*, kedua Algoritma K–Means menggunakan *Euclidean Distance*. Setting yang digunakan pada Algoritma K–Means dengan *Cosine Similarity* sebagai berikut:

('Distance','cosine','emptyaction','singleton','onlinephase','off','start','sample');,

sedangkan setting yang digunakan pada Algoritma K–Means dengan *Euclidean distance* sebagai berikut:

('Distance', 'sqEuclidean', 'emptyaction', 'singleton', 'onlinephase', 'off', 'start', 'sample');,

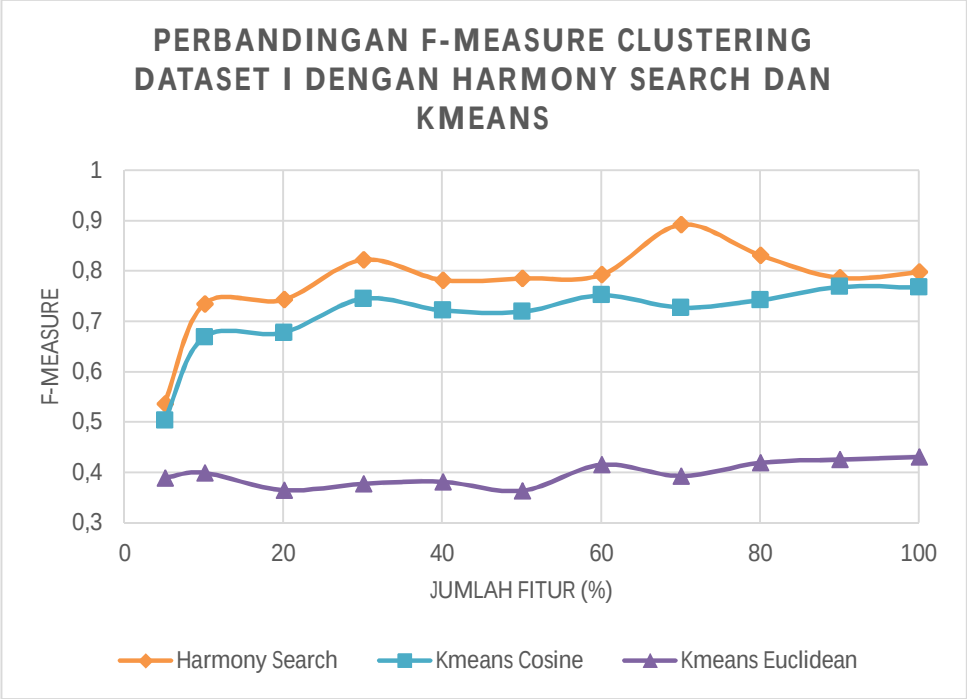
Untuk mendapatkan nilai F-Measure pada penelitian ini, metode K-Means dijalankan sebanyak 100 kali sehingga didapatkan nilai rata-rata F-Measure dari percobaan tersebut.

Pada Algoritma *Harmony Search* terdapat beberapa parameter yang perlu diinisialisasi, yakni *Harmony Memory Size* (HMS), *Harmony Memory Considering Rate* (HMCR), *Pitch Adjusting Rate* (PAR), serta jumlah total iterasi. Berdasarkan percobaan sebelumnya, nilai parameter HMS dan HMCR optimal untuk tiap-tiap dataset dapat dilihat pada tabel 4.21. Untuk parameter PAR ditentukan sesuai dengan penelitian sebelumnya yang nilainya berubah sesuai dengan jumlah iterasi saat ini dan total iterasi yang akan dilakukan (M. Mahdavi, M. Fesanghary, & E. Damangir, 2007). Sedangkan jumlah iterasi yang digunakan pada penelitian ini adalah 100 iterasi.

Tabel 4.21 Variasi Parameter Optimal Tiap Dataset

Dataset	**Parameter**	
	***Harmony Memory Size* (HMS)**	***Harmony Memory Considering Rate* (HMCR)**
I	5	0.9
II	5	0.8
III	9	0.6

Nilai F-Measure hasil pengelompokan dokumen pada dataset pertama dengan Metode K-Means dan Algoritma Harmony Search dapat dilihat pada Gambar 4.7.



Gambar 4.7 F-Measure Hasil Pengelompokan Pada Dataset Pertama.

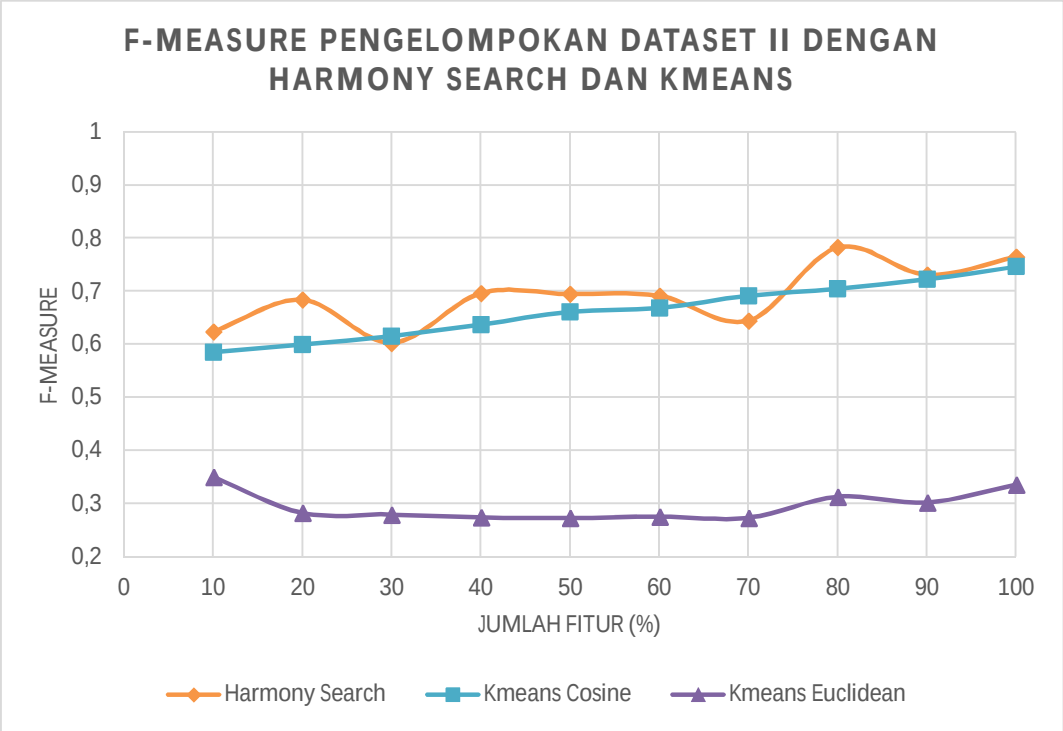
Nilai F-Measure tertinggi pada dataset pertama untuk metode K-Means dengan *Euclidean Distance* berturut-turut adalah 0.4313 pada saat semua fitur digunakan, 0.4263 pada saat 90% fitur, dan 0.4189 pada saat 80% fitur digunakan pada proses pengelompokan dokumen. Sedangkan F-Measure terendah untuk metode K-Means dengan *Euclidean Distance* pada dataset pertama adalah 0.3639 yakni pada saat 50% fitur digunakan untuk proses pengelompokan dokumen.

Pada metode K-Means dengan *Cosine Similarity* berturut-turut nilai F-Measure tertinggi adalah 0.7686 pada saat 90% fitur digunakan, 0.7676 pada saat 100% fitur, dan 0.7522 pada saat 60% fitur digunakan pada proses pengelompokan dokumen. Sedangkan F-Measure terendah untuk metode K-Means dengan *Cosine Similarity* pada dataset pertama adalah 0.5033 yakni pada saat 5% fitur digunakan untuk proses pengelompokan dokumen.

Pada algoritma *Harmony Search* untuk dataset pertama, nilai F-Measure tertinggi adalah 0.8924, lalu 0.8318 dan 0.8227 yakni pada saat 70%, 80% dan 30% fitur digunakan untuk proses pengelompokan dokumen. Nilai F-Measure terendah adalah 0.5359 yakni pada saat 5% fitur digunakan untuk proses pengelompokan dokumen.

Pada dataset pertama ini, berdasarkan uji coba dapat dilihat bahwa pengelompokan dokumen dengan algoritma *Harmony Search* yang menggunakan 70% fitur memiliki F-Measure 0.8924 yang lebih besar 16% daripada F-Measure terbesar pada metode K-Means dengan *Cosine Similarity*, yakni 0.7686 dan lebih besar 106% dibanding F-Measure terbesar pada metode K-Means dengan *Euclidean Distance*, yakni 0.4313. Hal ini menunjukkan bahwa algoritma *Harmony Search* yang dikombinasikan dengan seleksi fitur menggunakan algoritma *Random Projection–Gram Schmidt Orthogonalization* (RPGSO) dapat menghasilkan pengelompokan dokumen yang lebih baik dibandingkan dengan metode KMeans yang menggunakan *Cosine Similarity* maupun dengan metode KMeans yang menggunakan *Euclidean Distance*.

Nilai F-Measure hasil pengelompokan dokumen pada dataset kedua dengan Metode K-Means dan Algoritma *Harmony Search* dapat dilihat pada Gambar 4.8.



Gambar 4.8 F-Measure Hasil Pengelompokan Pada Dataset Kedua.

Nilai F-Measure tertinggi pada dataset kedua pada metode K-Means dengan *Euclidean Distance* berturut-turut adalah 0.3502 pada saat 10% fitur digunakan, 0.3355 pada saat 100% fitur, dan 0.3133 pada saat 80% fitur digunakan pada proses

pengelompokan dokumen. Sedangkan F-Measure terendah untuk metode K-Means dengan *Euclidean Distance* pada dataset kedua adalah 0.2732 yakni pada saat 50% fitur digunakan untuk proses pengelompokan dokumen.

Pada metode K-Means dengan *Cosine Similarity* berturut-turut nilai F-Measure tertinggi adalah 0.7464 pada saat 100% fitur digunakan, 0.7224 pada saat 90% fitur, dan 0.7044 pada saat 80% fitur digunakan pada proses pengelompokan dokumen. Sedangkan F-Measure terendah untuk metode K-Means dengan *Cosine Similarity* pada dataset pertama adalah 0.585 yakni pada saat 5% fitur digunakan untuk proses pengelompokan dokumen.

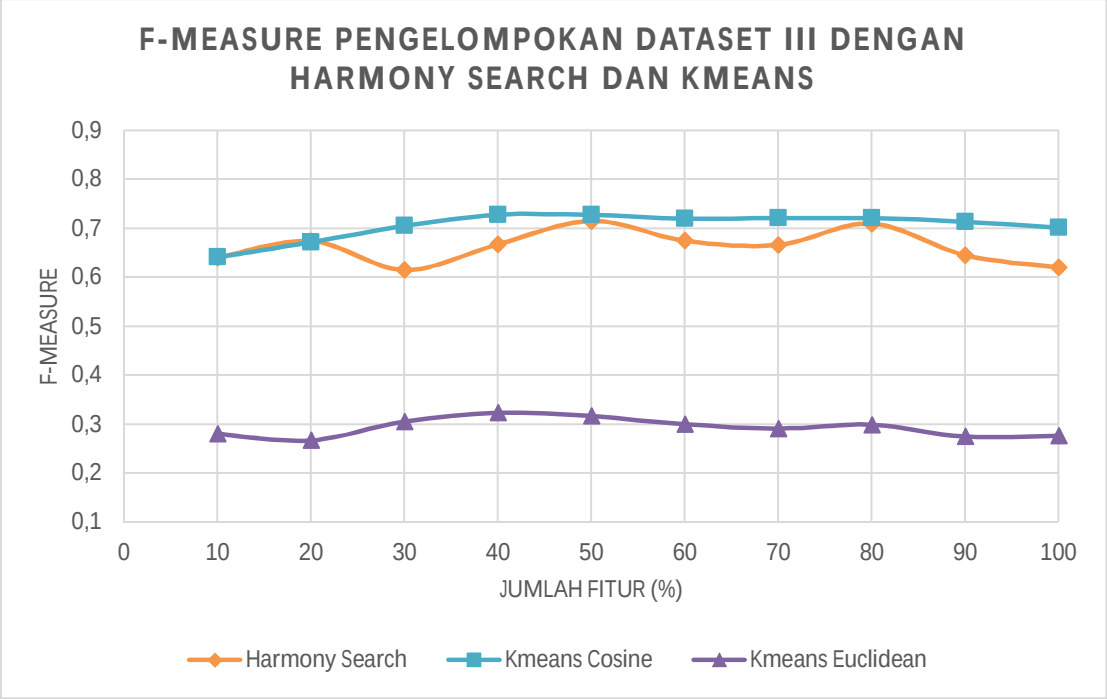
Pada algoritma *Harmony Search* untuk dataset kedua, nilai F-Measure tertinggi adalah 0.7826, lalu 0.765 dan 0.7309 yakni pada saat 80%, 100% dan 90% fitur digunakan untuk proses pengelompokan dokumen. Nilai F-Measure terendah adalah 0.6028 yakni pada saat 30% fitur digunakan untuk proses pengelompokan dokumen.

Pada dataset kedua ini, berdasarkan uji coba dapat dilihat bahwa pengelompokan dokumen dengan algoritma *Harmony Search* yang menggunakan 80% fitur memiliki F-Measure 0.7826 yang lebih besar 4.8% daripada F-Measure terbesar pada metode K-Means dengan *Cosine Similarity*, yakni 0.7464 dan lebih besar 123% dibanding F-Measure terbesar pada metode K-Means dengan *Euclidean Distance*, yakni 0.3502.

Nilai F-Measure hasil pengelompokan dokumen pada dataset ketiga dengan Metode K-Means dan Algoritma *Harmony Search* dapat dilihat pada Gambar 4.9. Nilai F-Measure tertinggi pada dataset ketiga pada metode K-Means dengan *Euclidean Distance* berturut-turut adalah 0.3235 pada saat 40% fitur digunakan, 0.3165 pada saat 50% fitur, dan 0.3058 pada saat 30% fitur digunakan pada proses pengelompokan dokumen. Sedangkan F-Measure terendah untuk metode K-Means dengan *Euclidean Distance* pada dataset kedua adalah 0.2669 yakni pada saat 20% fitur digunakan untuk proses pengelompokan dokumen.

Pada metode K-Means dengan *Cosine Similarity* berturut-turut nilai F-Measure tertinggi adalah 0.7282 pada saat 40% fitur digunakan, 0.7276 pada saat 50% fitur, dan 0.7212 pada saat 70% fitur digunakan pada proses pengelompokan dokumen. Sedangkan F-Measure terendah untuk metode K-Means dengan *Cosine*

Similarity pada dataset pertama adalah 0.6413 yakni pada saat 10% fitur digunakan untuk proses pengelompokan dokumen.



Gambar 4.9 F-Measure Hasil Pengelompokan Pada Dataset Ketiga.

Pada algoritma *Harmony Search* untuk dataset ketiga, nilai F-Measure tertinggi adalah 0.7153, lalu 0.71 dan 0.6752 yakni pada saat 50%, 80% dan 60% fitur digunakan untuk proses pengelompokan dokumen. Nilai F-Measure terendah adalah 0.6201 yakni pada saat 100% fitur digunakan untuk proses pengelompokan dokumen.

Pada dataset ketiga ini, berdasarkan uji coba dapat dilihat bahwa pengelompokan dokumen dengan algoritma *Harmony Search* yang menggunakan 80% fitur memiliki F-Measure 0.7153 yang lebih kecil 1.8% daripada F-Measure terbesar pada metode K-Means dengan *Cosine Similarity*, yakni 0.7282 namun lebih besar 121% dibanding F-Measure terbesar pada metode K-Means dengan *Euclidean Distance*, yakni 0.3235.

Pada uji coba dengan 100 iterasi diatas, algoritma *Harmony Search* ternyata memiliki F-measure yang lebih kecil dibanding dengan metode K-Means dengan *Cosine Similarity*. Untuk mengetahui lebih jauh perbandingan algoritma *Harmony*

Search dan metode K-Means dengan *Cosine Similarity* untuk dataset ketiga, maka dilakukan dilakukan percobaan lain dengan iterasi lebih dari 100 seperti pada Tabel 4.22.

Tabel 4.22 F-Measure Pengelompokan Dataset Ketiga untuk iterasi lebih dari 100

Iterasi	**F-Measure Percobaan Ke**					
	1	**2**	**3**	**4**	**5**	**Rata-rata**
100	0.5400	0.7586	0.6829	0.6557	0.7044	0.6683
200	0.5104	0.7647	0.6829	0.6651	0.7283	0.6703
300	0.7579	0.7666	0.6829	0.6402	0.7283	0.7152
400	0.7654	0.7666	0.7409	0.7483	0.7308	0.7504
500	0.7654	0.7666	0.7651	0.7456	0.7403	**0.7566**

Tabel 4.22 tersebut memperlihatkan bahwa nilai F-Measure pengelompokan dokumen dengan Algoritma *Harmony Search* untuk dataset ketiga ini memiliki F-measure 0.7504, yang lebih besar 3.04% dibanding metode K-Means dengan *Cosine Similarity*, yakni 0.7282 pada saat iterasi ke 400. Sedangkan pada saat iterasi ke 500, nilai F-Measure pengelompokan dokumen dengan Algoritma *Harmony Search* untuk dataset ketiga ini memiliki F-measure 0.7560, yang lebih besar 3.90% dibanding metode K-Means dengan *Cosine Similarity*, yakni 0.7282.

Dari Gambar 4.7, Gambar 4.8, Gambar 4.9, serta Tabel 4.22 dapat disimpulkan secara umum bahwa Algoritma *Harmony Search* memiliki F-Measure yang lebih tinggi dibandingkan Metode K-Means dengan *Cosine Similarity* maupun dibandingkan Metode K-Means dengan *Euclidean Distance*. Semakin besar iterasi yang dijalankan pada Algoritma *Harmony Search*, maka semakin tinggi nilai F-Measure yang dihasilkan. Tabel 4.23 memperlihatkan perbandingan tiap metode pada masing-masing dataset.

Pada Tabel 4.23 dapat dilihat bahwa untuk ketiga dataset, Algoritma *Harmony Search* memiliki F-Measure yang rata-rata lebih tinggi 8.40% dibandingkan Metode K-Means dengan *Cosine Similarity* serta 120.05% dibandingkan Metode K-Means dengan *Euclidean Distance*.

Tabel 4.23 F-Measure Tertinggi Pada Tiap Dataset

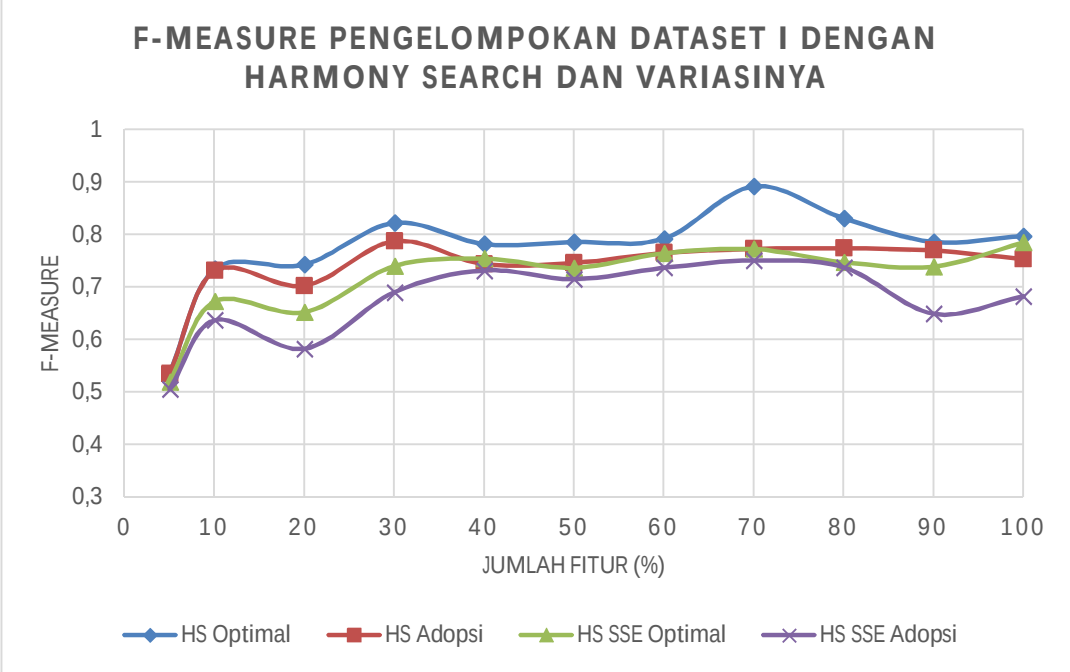
Dataset	F-Measure Tertinggi		
	Harmony Search	**KMeans *Cosine***	**KMeans *Euclidean***
Pertama	0.8924	0.7686	0.4313
Kedua	0.7826	0.7464	0.3502
Ketiga	0.7566	0.7282	0.3235
Rata-rata	0.8105	0.7477	0.3683

4.2.7. Perbandingan dengan Parameter dan Fungsi Fitness Yang Lain.

Pada uji coba ini, pengelompokan dokumen dengan Algoritma *Harmony Search* menggunakan fitur-fitur yang telah diseleksi berdasarkan urutan yang dihasilkan dari proses *Random Projection – Gram Schmidt Orthogonalization* (RPGSO). Ada dua fungsi *fitness* Algoritma *Harmony Search* yang digunakan pada uji coba, pertama menggunakan fungsi *fitness Average Distance of Documents to the cluster Centroid* (ADDC), kedua menggunakan fungsi *fitness Sum of Squared Error* (SSE). Pada kedua algoritma tersebut, masing-masing menggunakan dua macam parameter, yakni parameter optimal yang dihasilkan pada uji coba penelitian ini, dan parameter adopsi yang didapatkan dari penelitian sebelumnya (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013), yakni nilai parameter HMS ditentukan senilai dua kali jumlah *cluster* dokumen, parameter HMCR ditentukan dengan nilai 0.6.

Algoritma *Harmony Search* yang menggunakan fungsi *fitness* ADDC dan memakai parameter optimal yang dihasilkan pada uji coba penelitian ini selanjutnya disebut HS optimal. Algoritma *Harmony Search* yang menggunakan fungsi *fitness* ADDC dan memakai parameter adopsi yang didapatkan dari penelitian sebelumnya disebut HS adopsi. Algoritma *Harmony Search* yang menggunakan fungsi *fitness* SSE dan memakai parameter optimal yang dihasilkan pada uji coba penelitian ini selanjutnya disebut HS SSE optimal. Sedangkan algoritma *Harmony Search* yang menggunakan fungsi *fitness* SSE dan memakai parameter adopsi yang didapatkan dari penelitian sebelumnya disebut HS SSE adopsi.

Nilai F-Measure hasil pengelompokan dokumen pada dataset pertama dengan variasi Algoritma *Harmony Search* dapat dilihat pada Gambar 4.10.



Gambar 4.10 F-Measure Hasil Pengelompokan Pada Dataset Pertama.

Berdasarkan Gambar 4.10, nilai F-Measure terbesar pada dataset pertama untuk algoritma HS optimal adalah 0.8924 yang dicapai pada saat 70% fitur digunakan pada proses pengelompokan dokumen. Nilai F-Measure terbesar selanjutnya adalah 0.8318 pada saat 80% fitur digunakan dan 0.8227 pada saat 30% fitur digunakan pada proses pengelompokan dokumen.

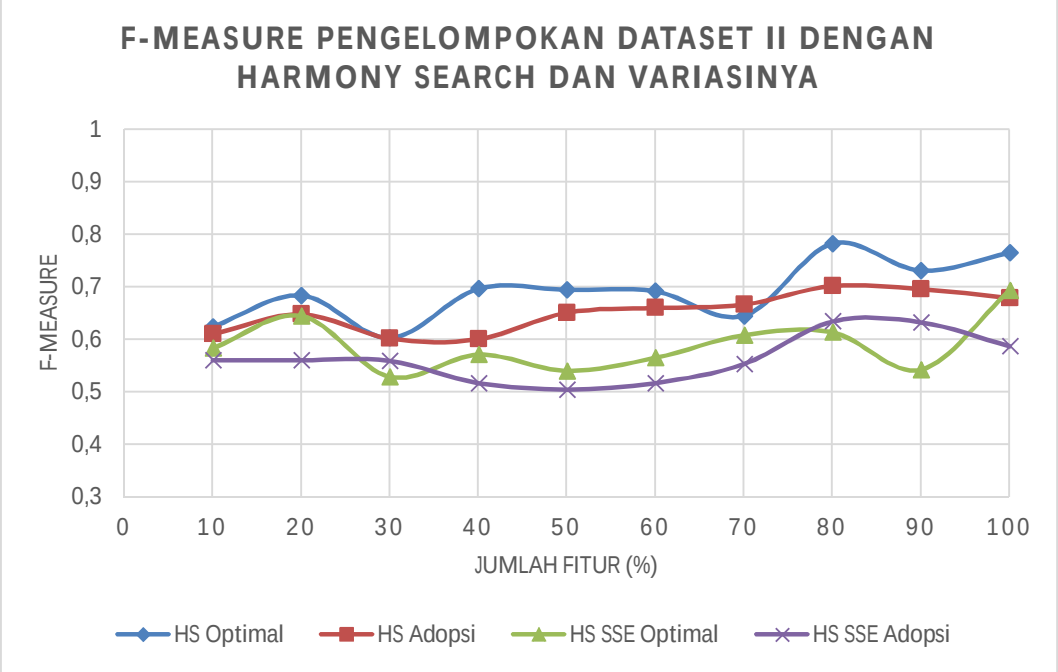
Pada algoritma HS adopsi, nilai F-Measure terbesar adalah 0.7879 yang dicapai pada saat 30% fitur digunakan. Nilai F-Measure terbesar selanjutnya adalah 0.7743 pada saat 80% fitur digunakan dan 0.7735 pada saat 70% fitur digunakan pada proses pengelompokan dokumen.

Sedangkan pada algoritma HS SSE optimal dan HS SSE adopsi, nilai F-Measure terbesar adalah 0.7852 dan 0.7512 yang dicapai pada saat 100% dan 70% fitur digunakan. Nilai F-Measure terbesar selanjutnya adalah 0.7729 dan 0.738 pada saat 70% dan 80% fitur digunakan serta 0.7647 dan 0.7372 yang keduanya dicapai pada saat 30% fitur digunakan pada proses pengelompokan dokumen.

Berdasarkan Gambar 4.10 diatas, dapat disimpulkan bahwa Algoritma Harmony Search pada dataset pertama menggunakan fungsi *fitness* ADDC dan

memakai parameter optimal memiliki nilai F-Measure 0.8924 yang lebih besar 13.26% dari Algoritma Harmony Search dengan parameter adopsi (0.7879), lebih besar 13.65% dari Algoritma *Harmony Search* dengan SSE dan parameter optimal (0.7852), serta lebih besar 18.80% dari Algoritma Harmony Search dengan SSE dan parameter adopsi (0.7512).

Nilai F-Measure hasil pengelompokan dokumen pada dataset kedua dengan variasi Algoritma *Harmony Search* dapat dilihat pada Gambar 4.11.



Gambar 4.11 F-Measure Hasil Pengelompokan Pada Dataset Kedua.

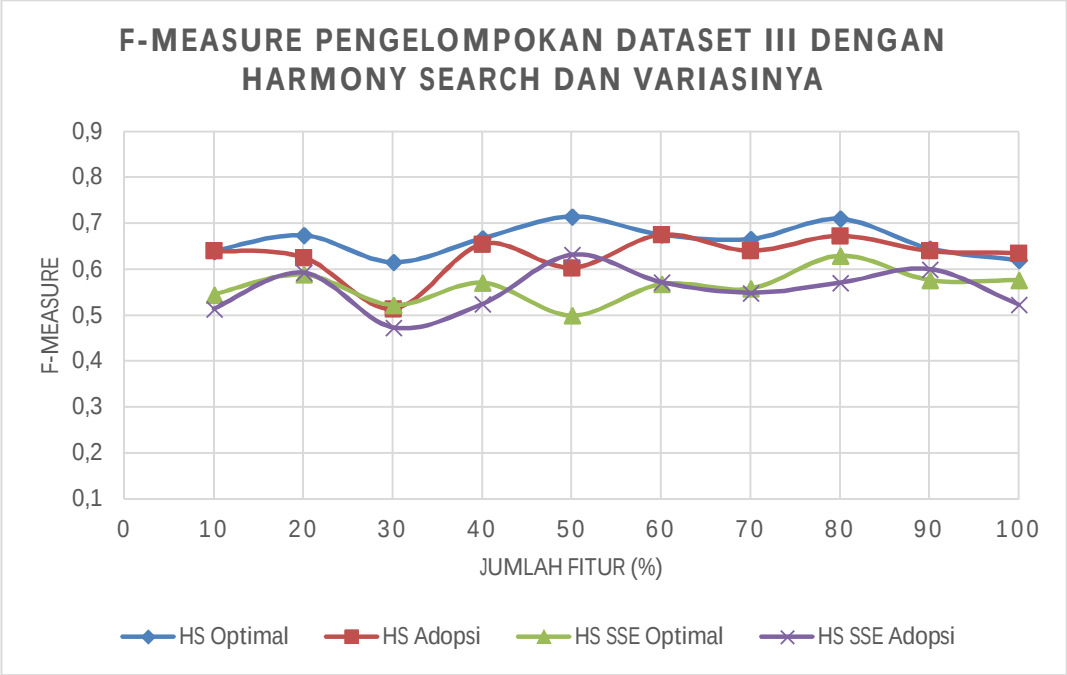
Berdasarkan Gambar 4.11, nilai F-Measure terbesar pada dataset pertama untuk algoritma HS optimal adalah 0.7826 yang dicapai pada saat 80% fitur digunakan pada proses pengelompokan dokumen. Nilai F-Measure terbesar selanjutnya adalah 0.765 pada saat 100% fitur digunakan dan 0.7309 pada saat 90% fitur digunakan pada proses pengelompokan dokumen.

Pada algoritma HS adopsi, nilai F-Measure terbesar adalah 0.7018 yang dicapai pada saat 80% fitur digunakan. Nilai F-Measure terbesar selanjutnya adalah 0.696 pada saat 90% fitur digunakan dan 0.6792 pada saat 100% fitur digunakan pada proses pengelompokan dokumen.

Sedangkan pada algoritma HS SSE optimal dan HS SSE adopsi, nilai F-Measure terbesar 0.6932 dan 0.6341 yang dicapai pada saat 100% dan 80% fitur digunakan. Nilai F-Measure terbesar selanjutnya adalah 0.6449 dan 0.6327 pada saat 20% dan 90% fitur digunakan serta 0.6136 dan 0.5868 yang dicapai pada saat 80% dan 100% fitur digunakan pada proses pengelompokan dokumen.

Berdasarkan Gambar 4.11 diatas, dapat disimpulkan bahwa Algoritma Harmony Search pada dataset kedua menggunakan fungsi *fitness* ADDC dan memakai parameter optimal memiliki nilai F-Measure 0.7826 yang lebih besar 11.51% dari Algoritma Harmony Search dengan parameter adopsi (0.7018), lebih besar 12.90% dari Algoritma Harmony Search dengan SSE dan parameter optimal (0.6932), serta lebih besar 23.42% dari Algoritma Harmony Search dengan SSE dan parameter adopsi (0.6341).

Nilai F-Measure hasil pengelompokan dokumen pada dataset ketiga pada variasi Algoritma Harmony Search dapat dilihat pada Gambar 4.12.



Gambar 4.12 F-Measure Hasil Pengelompokan Pada Dataset Ketiga.

Berdasarkan Gambar 4.12, nilai F-Measure terbesar pada dataset ketiga untuk algoritma HS optimal adalah 0.7153 yang dicapai pada saat 50% fitur digunakan pada proses pengelompokan dokumen. Nilai F-Measure terbesar

selanjutnya adalah 0.71 pada saat 80% fitur digunakan dan 0.6752 pada saat 60% fitur digunakan pada proses pengelompokan dokumen.

Pada algoritma HS adopsi, nilai F-Measure terbesar adalah 0.6757 yang dicapai pada saat 60% fitur digunakan. Nilai F-Measure terbesar selanjutnya adalah 0.6732 pada saat 80% fitur digunakan dan 0.6552 pada saat 40% fitur digunakan pada proses pengelompokan dokumen.

Sedangkan pada algoritma HS SSE optimal dan HS SSE adopsi, nilai F-Measure terbesar 0.6299 dan 0.6321 yang dicapai pada saat 80% dan 50% fitur digunakan. Nilai F-Measure terbesar selanjutnya adalah 0.5886 dan 0.601 pada saat 20% dan 90% fitur digunakan serta 0.5771 dan 0.5933 yang dicapai pada saat 100% dan 30% fitur digunakan pada proses pengelompokan dokumen.

Berdasarkan Gambar 4.12 diatas, dapat disimpulkan bahwa Algoritma *Harmony Search* pada dataset ketiga menggunakan fungsi *fitness* ADDC dan memakai parameter optimal memiliki nilai F-Measure 0.7153 yang lebih besar 5.86% dari Algoritma *Harmony Search* dengan parameter adopsi (0.6757), lebih besar 13.56% dari Algoritma *Harmony Search* dengan SSE dan parameter optimal (0.6299), serta lebih besar 13.16% dari Algoritma *Harmony Search* dengan SSE dan parameter adopsi (0.6321).

Dari Gambar 4.10, Gambar 4.11 dan Gambar 4.12, dapat disimpulkan bahwa Algoritma *Harmony Search* menggunakan fungsi *fitness* ADDC dan memakai parameter optimal memiliki F-Measure yang lebih tinggi dibandingkan Algoritma *Harmony Search* dengan variasi parameter dan fungsi fitness yang lain. Tabel 4.24 memperlihatkan perbandingan tiap metode pada masing-masing dataset.

Tabel 4.24 F-Measure Tertinggi Variasi *Harmony Search* Tiap Dataset

Dataset	**F-Measure Tertinggi**			
	HS Opt	**HS Adop**	**HS SSE Opt**	**HS SSE Adop**
Pertama	0.8924	0.7879	0.7852	0.7512
Kedua	0.7826	0.7018	0.6932	0.6341
Ketiga	0.7153	0.6757	0.6299	0.6321
Rata-rata	0.7968	0.7218	0.7028	0.6725

Pada Tabel 4.24 dapat dilihat bahwa untuk ketiga dataset, Algoritma *Harmony Search* memiliki F-Measure yang rata-rata lebih tinggi 10.39% dibandingkan Algoritma *Harmony Search* dengan parameter adopsi, lebih besar 13.38% dari Algoritma *Harmony Search* dengan SSE dan parameter optimal, serta lebih besar 18.48% dari Algoritma *Harmony Search* dengan SSE dan parameter adopsi.

[Halaman ini sengaja dikosongkan]

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uji coba serta analisa terhadap metode usulan yakni seleksi fitur pada pengelompokan dokumen dengan *Random Projection Gram Schmidt Orthogonalization* (RPGSO) dan Algoritma *Harmony Search* (HS) menggunakan fungsi fitness berupa *Average Distance of Documents to the cluster Centroid* (ADDC), maka didapatkan kesimpulan sebagai berikut :

1. Pengelompokan dokumen dengan metode usulan dapat menghasilkan kelompok dokumen yang memiliki kriteria F-Measure rata-rata lebih tinggi 9.50% dibandingkan dengan pengelompokan dokumen menggunakan seluruh fitur yang ada.
2. Pengelompokan dokumen dengan metode usulan mampu menghasilkan kelompok dokumen yang memiliki kriteria rata-rata F-Measure lebih tinggi 8.40% dibandingkan Metode K-Means dengan *Cosine Similarity* serta lebih tinggi 120.05% dibandingkan Metode K-Means dengan *Euclidean Distance*.
3. Pengelompokan dokumen dengan metode usulan mampu menghasilkan kelompok dokumen yang memiliki kriteria rata-rata F-Measure lebih tinggi 13.38% dibandingkan Algoritma *Harmony Search* dengan fungsi *fitness* berupa *Sum of Squared Error* (SSE).

5.2 Saran

Berdasarkan uji coba yang telah dilakukan dan kesimpulan yang didapatkan, maka saran untuk pengembangan penelitian ini antara lain :

1. Penggunaan karakter fitur sebagai informasi tambahan pada proses RPGSO, sehingga tidak perlu membandingkan satu persatu fitur untuk memperoleh fitur yang paling penting.
2. Penggunakan fungsi fitness yang lain untuk mempercepat proses konvergensi pada iterasi algoritma *Harmony Search*.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

Achlioptas, D. (2003). Database-Friendly Random Projections: Johnson-Lindenstrauss With Binary Coins. *Journal of Computer and System Sciences*, 671-687.

Alia, O., & Mandava, R. (2011). The variants of the harmony search algorithm: an overview. *Artificial Intelligence Review 36*, pp. 49-68.

Arifin, A. Z., Mahendra, I., & Ciptaningtyas, H. (2009). Enhanced Confix Stripping Stemmer and Ants Algorithm for Classifying News Document in Indonesian Language. *International Conference on Information & Communication Technology and Systems (ICTS)* . Surabaya.

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., . . . Zhu, M. (2013). A Practical Algorithm for Topic Modeling with Provable Guarantees. *30th International Conference on Machine Learning, ICML 2013* (pp. 939-947). United States: International Machine Learning Society (IMLS).

Asian, J. (2007). *Effective Techniques for Indonesian Text Retrieval.* PhD thesis, School of Computer Science and Information Technology, RMIT University.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 245-271.

Chandrashekar, G., & Sahin, F. (2014). A Survey On Feature Selection Methods. *Computers And Electrical Engineering*, 16-28.

Chen, S., Billings, S., & Luo, W. (1989). Orthogonal Least Squares Methods And Their Application. *International Journal of Control*, 1873 - 1896.

Everitt, B. S. (1980). *Cluster Analysis.* New York: Halsted Press.

Fodor, I. (2002). *A survey of dimension reduction techniques.* Center for Applied Scientific Computing, Lawrence Livermore National Laboratory: Technical Report UCRL-ID-148494.

Forsati, R., Mahdavi, M., Shamsfard, M., & Meybodi, M. R. (2013). Efficient Stochastic Algorithms For Document Clustering. *Information Sciences*, 269–291.

Golub, G., & Van Loan, C. (1989). *Matrix Computations (Johns Hopkins Studies in the Mathematical Sciences), 3rd ed.* London: Johns Hopkins University.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*, 1157-1182.

Hall, M. (1999). *Correlation-based Feature Selection for Machine Learning.* Department of Computer Science, Thesis for Doctor of Philosophy, The University of Waikato.

Han, J., & Kamber, M. (2011). *Data Mining : Concepts and Techniques.* Elsevier.

Huang, J., Cai, Y., & Xu, X. (2006). A Wrapper for Feature Selection Based on Mutual Information. *18th International Conference on Pattern Recognition (ICPR'06)*, (pp. 618-621). Hong Kong.

Jolliffe, I. (1986). *Principal Component Analysis.* Springer-Verlag.

Kohavi, R., & John, G. H. (1997). Wrappers For Feature Subset Selection. *Artificial Intelligence*, 273-324.

Lee, K. S., & Geem, Z. W. (2005). A New Meta-heuristic Algorithm For Continuous Engineering Optimization: Harmony Search Theory And Practice. *Computer Methods in Applied Mechanics and Engineering*, 3902–3933.

M. Mahdavi, M. Fesanghary, & E. Damangir. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 1567-1579.

Mesleh, A., & Kanaan, G. (2008). Support vector machine text classification system: Using Ant Colony Optimization based feature subset selection. *ICCES 2008. International Conference on Computer Engineering & Systems*, (pp. 143-148). Cairo, Egypt.

Nazief, B., & Adriani, M. (1996). *Confixstripping : Approach to Stemming Algorithm for Bahasa Indonesia.* Internal publication, Faculty of Computer Science, University of Indonesia, Depok, Jakarta.

Salton, G. (1989). *Automatic Text Processing.* Boston: Addison-Wesley.

Salton, G., & Buckley, C. (1988). Term-Weighting Approaches In Automatic Text Retrieval. *Information Processing And Management*, 513-523.

Tahitoe, A., & Purwitasari, D. (2010). *Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia dengan Metode Corpus Based Stemming.* Surabaya: Institut Teknologi Sepuluh Nopember (ITS).

Tala, F. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.* M.Sc. Thesis. Master of Logic Project. Institute for Logic, Language and Computation. Universiteti van Amsterdam The Netherlands.

Unler, A., Murat, A., & Chinnam, R. B. (2011). mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, 4625-4641.

Wang, D., Zhang, H., Liu, R., Liu, X., & Wang, J. (2016). Unsupervised Feature Selection Through Gram–Schmidt Orthogonalization — A Word Co-occurrence Perspective. *Neurocomputing*, 845–854.

Zhang, C.-K., & Hu, H. (2005). Feature selection using the hybrid of ant colony optimization and mutual information for the forecaster. *International Conference on Machine Learning and Cybernetics*, (pp. 1728-1732). Guangzhou, China.

BIOGRAFI PENULIS

Terlahir di desa Morobakung Manyar Gresik, penulis menamatkan pendidikan dasar di MI Roudlotut Tholibin Morobakung pada 1994. Selanjutnya menamatkan pendidikan menengah pertama di MTs Assa'adah I Bungah Gresik pada 1997, dan pendidikan menengah atas di SMU Insan Cendekia Boarding School Serpong Tangerang pada tahun 2000. Pada tahun 2006, penulis menamatkan pendidikan sarjana di Program Studi Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Tahun 2014, penulis berkesempatan memulai studi di Program Studi Magister Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) Surabaya melalui program beasiswa BPP-DN Tendik. Saat ini, penulis bekerja pada Seksi Sistem Informasi Bidang Kelembagaan dan Sistem Informasi di Kopertis Wilayah VII. Selain itu, penulis juga mengajar sebagai dosen luar biasa di beberapa perguruan tinggi swasta di daerah Gresik. Untuk menghubungi penulis, silakan kirim email ke muh.machmud@gmail.com atau 08563011516.