# Term Weighting Berbasis Indeks Buku dan Kelas untuk Perangkingan Dokumen Berbahasa Arab

**M. Ali Fauzi[1], Dr. Agus Zainal Arifin[2], S.Kom, M.Kom, Anny Yuniarti[3], S.Kom, M.Comp.Sc**
Institut Teknologi Sepuluh Nopember
e-mail: moch.ali.fauzi@gmail.com

***Abstrak***

*Information Retrieval berdasarkan query tertentu sudah jamak ditemukan pada sistem komputer saat ini. Salah satu metode yang populer digunakan adalah perangkingan dokumen menggunakan space vector model berbasis pada nilai term weighting TF.IDF. Pada penelitian ini, terdapat beberapa buku berbahasa Arab yang memiliki puluhan bahkan ratusan halaman. Masing-masing halaman dari buku tersebut adalah sebuah dokumen yang akan diranking berdasarkan query dari pengguna. TF.IDF hanya melakukan pembobotan berbasis pada dokumen tanpa memperhatikan indeks buku dan kelas yang merupakan induk dokumen tersebut sehingga kinerjanya kurang maksimal jika diimplementasikan pada kasus ini. Oleh karena itu, diusulkan metode baru term weighting yang berbasis pada indeks buku dan kelas. Metode ini memperhatikan frekuensi kemunculan term pada keseluruhan buku dan kelas. Metode yang disebut inverse class frequency (ICF) dan inverse book frequency (IBF) ini digabungkan dengan metode sebelumnya sehingga menjadi TF.IDF.ICF.IBF. Pengujian metode ini menggunakan dataset dari beberapa e-book berbahasa arab. Hasil penelitian menunjukkan bahwa metode yang diajukan terbukti dapat diaplikasikan pada perangkingan dokumen berbahasa arab dan memiliki performa yang lebih bagus dibanding metode sebelumnya dengan nilai F-Measure 75%, precision 76%, dan recall mencapai 74%.*

***Kata kunci:*** *Perankingan Dokumen, Term Weighting, IBF, Indeks Buku, Indeks Kelas*

***Abstract***

*Information Retrieval based on specific queries is common to the current computer systems. One of the popular methods used is the document ranking method using vector space models based on TF.IDF term weighting. In this study, there are several books in Arabic that has tens or even hundreds of pages. Each page of the book is a single document that will be ranked based on the user query. TF.IDF only performs term weighting based on the document without regard to the indexes of the book and class of the document. Therefore, a new method of term weighting that based on books and classes indexes proposed. This method favor the frequency of term in whole books and classes. This method that called inverse class frequency (ICF) and inverse book frequency (IBF) then combined with the previous method so that it becomes TF.IDF.ICF.IBF. This new method was tested using a dataset from some Arabic e-books. The experimental results show that the proposed method can be implemented on document ranking method and the performances are better than some previous methods with F-Measure value 75%, precision value 76%, dan recall value 74%.*

***Keywords:*** *Dokument Ranking, Term Weighting, IBF, Book Index, Class Index*

## 1. Pendahuluan

Tujuan dari sistem temu kembali informasi adalah menemukan informasi yang paling relevan untuk memenuhi kebutuhan informasi pengguna. Salah satu pembahasan temu kembali informasi yang biasa di teliti adalah tentang perangkingan dokumen. Perangkingan dokumen dilakukan untuk mendapatkan dokumen-dokumen yang relevan dengan *query* pengguna diurutkan dari tingkat relevansinya [1][2].

Beberapa penelitian yang membahas perangkingan dokumen berbahasa Arab telah dilakukan sebelumnya, seperti perangkingan dengan menggunakan pencocokan *N-gram* terhadap kata dari *query* dan dokumen [3][4], menggunakan modul *crawler* dokumen dengan *feedback* bentuk kata yang tepat [2], dan berdasarkan variasi *orthographic* [5]. Harrag dkk menggunakan *vector space model* berbasis *term weighting* TF.IDF untuk melakukan perangkingan pada dokumen berbahasa Arab. Pada metode ini dokumen direpresentasikan sebagai sebuah vektor yang dibentuk dari nilai-nilai term yang menjadi indeknya [6]. Nilai-nilai *term* tersebut dihitung dengan menggunakan *term weighting* TF.IDF. TF.IDF mengkombinasikan *term frequency* (TF) yang mengukur kepadatan term dalam sebuah dokumen dikalikan dengan inverse document frequency (IDF) yang mengukur keinformatifan sebuah *term* (kelangkaannya pada keseluruhan korpus) [7]. Akan tetapi, term weighting dengan TF.IDF yang hanya berbasis pada dokumen itu tidak cukup untuk menentukan indeks dari suatu dokumen. Penentuan indeks yang akurat juga bergantung pada keinformatifan *term* terhadap kelas (kelangkaanya pada keseluruhan kelas). Term yang sering muncul di banyak kelas seharusnya tidak menjadi *term* yang penting meskipun nilai TF.IDFnya tinggi. Oleh karena itu, Fuji Ren & Mohammad Golam Sohrab mengusulkan penggunaan pembobotan berbasis kelas untuk *term weighting* pada dokumen berbahasa Inggris yang dinamakan *inverse class frequency* (ICF) dan variasinya, *Inverse Class Space density Frequency* (ICSdF) [8]. Dengan ICF dan ICSdF ini *term* yang sering muncul pada banyak kelas akan memiliki nilai yang kecil. Metode ini terbukti memiliki *precision* dan *recall* yang lebih tinggi daripada TF.IDF [8].

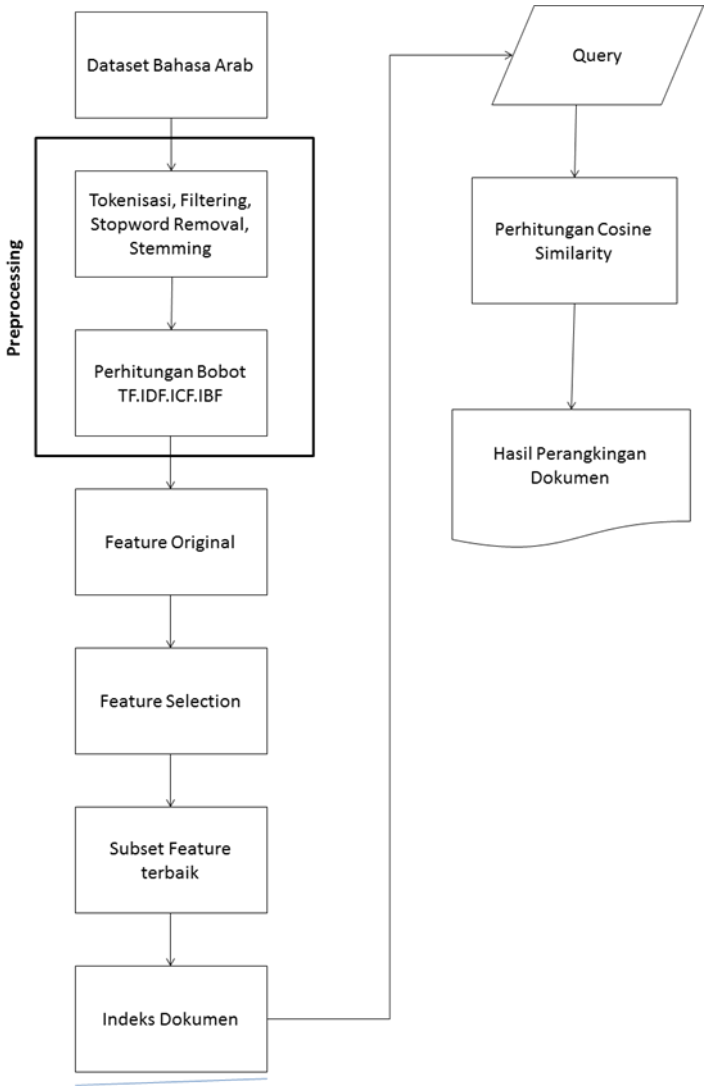
Dalam penelitian ini, dibutuhkan metode perangkingan halaman-halaman buku berbahasa Arab. Buku-buku tersebut memiliki jumlah halaman yang banyak, antara puluhan hingga ratusan halaman. Masing-masing halaman buku adalah sebuah dokumen. Hasil pencarian *query* dari pengguna akan menunjukkan dokumen halaman berapakah dan dari buku manakah yang sesuai dengan *query* pengguna. *Term weighting* yang hanya berbasis pada dokumen dan kelas semacam TF.IDF.ICF tidak cukup untuk menentukan indeks dari suatu dokumen halaman-halaman buku. Buku dapat dikatakan sebagai bentuk lain dari kelas atau kategori. Semua dokumen (halaman) dalam sebuah buku pasti membahas topik yang hampir sama. Seperti pada ICF, beberapa indeks buku seharusnya juga menjadi *term* kunci bagi dokumen-dokumen di dalam buku tersebut. Selain itu, keinformatifan term terhadap buku (kelangkaanya pada keseluruhan buku) juga perlu diperhatikan. Beberapa *term* yang sering muncul pada suatu buku pasti akan memiliki TF.IDF.ICF yang tinggi, akan tetapi *term* itu belum tentu bisa dikatakan sebagai term kunci sebelum dihitung kelangkaanya pada keseluruhan buku. *Term* yang sering muncul pada banyak ragam buku seharusnya tidak memiliki nilai yang tinggi karena tidak mencerminkan indeks buku tersebut.

Oleh karena itu, diusulkan metode baru pembobotan *term* berbasis buku untuk perangkingan dokumen bahasa Arab yang dinamakan *inverse book frequency* (IBF) untuk meningkatkan performa perangkingan dokumen yang memiliki hierarki berupa buku-buku yang memiliki banyak halaman. Perhitungan IBF ini akan dikombinasikan juga dengan metode sebelumnya sehingga menjadi TF.IDF.ICF.IBF. Metode ini dapat diterapkan pada dokumen semua bahasa secara umum yang memiliki hierarki berupa buku-buku yang memiliki banyak halaman. Akan tetapi, dilihat dari keperluan penerapan metode ini pada aplikasi pencarian kitab berbahasa Arab serta sumber dataset dan ground truth dari *expert* yang dimiliki adalah dokumen-dokumen berbahasa Arab maka metode ini akan diterapkan pada *Information Retrieval* dokumen berbahasa Arab. Metode TF.IDF.ICF.IBF ini diharapkan *precision* dan *recall* yang lebih tinggi pada perangkingan halaman-halaman buku berbahasa Arab dibandingkan dengan metode sebelumnya.

## 2. Metodologi Penelitian

Secara garis besar, skema metode perankingan dokumen dalam penelitian ini terdiri dari dua tahapan utama, yaitu penentuan indeks dokumen dan perangkingan dokumen berdasarkan *query* dari pengguna. Perangkingan dokumen dilakukan berdasarkan perhitungan *similarity* antara *vector* indeks dokumen dan *query* yang berbasis pada pembobotan *term* TF.IDF.ICF.IBF. Bagan besar proses perangkingan ini seperti terlihat pada Gambar 1.

Sebelum dilakukan proses perangkingan perlu dilakukan tahapan indexing seperti terlihat pada Gambar 1. Pada tahapan ini terdapat beberapa proses yang saling berkesinambungan. proses-proses dalam tahap ini diantaranya *tokenization*, *filtering*, *stopwords removal*, *stemming* dan penghitungan bobot.




**Gambar 1. Skema Proses Perangkingan Dokumen**

Untuk *stemming* akan digunakan *light stemmer* yang sering digunakan dalam *information retrieval* teks Arab [9]. Setelah itu, akan didapatkan sebuah set fitur original dari semua dokumen. Melalui metode *feature selection*, set fitur original tersebut akan dipilih sebuah subset yang berisi beberapa fitur terbaik sesuai dengan kriteria tertentu yang dalam penelitian ini adalah nilai TF.IDF.ICF.IBF. Subset terbaik inilah yang disebut sebagai indeks dari dokumen tersebut.

Indeks dari dokumen-dokumen tersebut akan dihitung kemiripannya dengan *query* yang dimasukkan oleh pengguna. Perhitungan kemiripan ini dilakukan dengan menggunakan perhitungan *cosine similarity* yang berbasis pada TF.IDF.ICF.IBF. Dokumen-dokumen yang didapatkan akan diurutkan secara *descending* sesuai dengan nilai *cosine similarity*nya. Hasil ini menunjukkan hasil perangkingan dokumen sesuai tingkat kemiripannya dengan *query* pengguna.

## 3. Kajian Pustaka

### 3.1 Pembobotan *Term*

Perangkingan dokumen menggunakan representasi *vector space model* dari kumpulan dataset. Dokumen dalam *vector space model* direpresentasikan dalam *matriks* yang berisi bobot kata pada dokumen. Bobot tersebut menyatakan kepentingan/kontribusi kata terhadap suatu dokumen dan kumpulan dokumen. Kepentingan suatu kata dalam dokumen dapat dilihat dari frekuensi kemunculannya terhadap dokumen. Biasanya kata yang berbeda memiliki frekuensi yang berbeda. Dibawah ini terdapat beberapa metode pembobotan :

1. *Term Frequency* (TF)
   *Term frequency* merupakan metode yang paling sederhana dalam membobotkan setiap *term*. Setiap *term* diasumsikan memiliki kepentingan yang proporsional terhadap jumlah kemunculan *term* pada dokumen. Bobot dari *term* t pada dokumen d yaitu:

$$TF(d,t) = f(d,t,), \quad (1)$$

   dimana *f(d,t)* adalah frekuensi kemunculan *term t* pada dokumen *d*.

2. *Inverse Document Frequency* (IDF)
   Bila *term frequency* memperhatiakan kemunculan term di dalam dokumen, maka IDF memperhatikan kemunculan term pada kumpulan dokumen. Latar belakang pembobotan ini adalah term yang jarang muncul pada kumpulan dokumen sangat bernilai. Kepentingan tiap *term* diasumsikan memilki proporsi yang berkebalikan dengan jumlah dokumen yang mengandung *term*. Faktor IDF dari *term* t yaitu:

$$IDF(t) = 1 + log(Nd / df(t)), \quad (2)$$

   dimana *Nd* adalah jumlah seluruh dokumen, dan *df(t)* jumlah dokumen yang mengandung *term t*.

3. *Inverse Class Frequency* (ICF)
   Jika IDF memperhatikan kemunculan *term* pada kumpulan dokumen, maka ICF memperhatikan kemunculan *term* pada kumpulan kategori/kelas. *Term* yang jarang muncul pada banyak kelas adalah *term* yang bernilai untuk klasifikasi. Kepentingan tiap *term* diasumsikan memilki proporsi yang berkebalikan dengan jumlah kelas yang mengandung *term*. Faktor ICF dari *term* t yaitu:

$$ICF(t) = 1 + log(Nc / cf(t)), \quad (3)$$

   dimana *Nc* adalah jumlah seluruh kelas, *cf(t)* jumlah kelas yang mengandung *term t*.

4. *Inverse Book Frequency* (IBF)
   Jika ICF memperhatikan kemunculan term pada kumpulan kelas, maka IBF memperhatikan kemunculan *term* pada kumpulan kitab/buku. *Term* yang jarang muncul pada banyak buku adalah *term* yang sangat bernilai. Kepentingan tiap *term* diasumsikan memilki proporsi yang berkebalikan dengan jumlah buku yang mengandung *term*. Faktor IBF dari *term* t yaitu:

$$IBF(t) = 1 + log(Nb / bf(t)), \quad (4)$$

   dimana *Nb* adalah jumlah seluruh buku, *bf(t)* jumlah buku yang mengandung *term t*

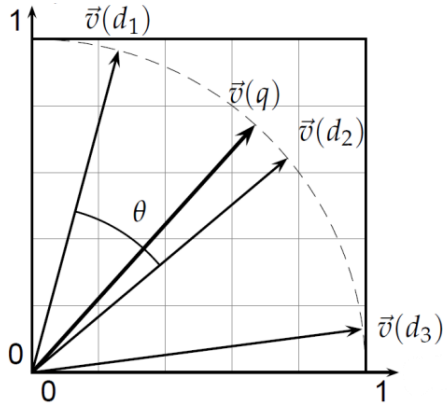
5. TF.IDF.ICF.IBF
   TF.IDF.ICF.IBF merupakan perkalian antara TF, IDF, ICF dan IBF. Kombinasi bobot dari term *t* pada dokumen *d* yaitu:

$$TF.IDF.ICF.IBF\ (d,t) = TF(d,t)\ x\ IDF(t)\ x\ ICF(t)\ x\ IBF(t), \quad (5)$$

dimana *TF(d,t)* adalah nilai TF *term t* pada dokumen *d*, IDF(t) adalah nilai IDF *term t*, ICF(t) adalah nilai ICF *term t* dan IBF(t) adalah nilai IBF *term t.*

### 3.2 *Cosine Similarity*

Hasil pembobotan kata pada dokumen digunakan sebagai representasi vektor. Dari representasi bobot tersebut dapat dihitung nilai kemiripan suatu dokumen dengan *query*. Nilai kemiripan ini biasa dihitung dengan rumusan *cosine similarity*, perhitungan tingkat kemiripan ini dibuat dengan berdasar pada besar sudut kosinus antara dua vektor, dalam hal ini adalah vektor dokumen. Representasi perumusan ini dalam bidang kartesian seperti diperlihatkan pada Gambar 2.




**Gambar 2. Representasi Perumusan Cosine Similarity**

Dalam Gambar 2. terdapat tiga vektor dokumen *d1, d2* dan *d3* dan satu vektor *query q*. *cosine similarity* menghitung nilai kosinus θ dari *query* dan tiga dokumen lain. Nilai ini menunjukkan derajat kemiripan dokumen dengan *query*.

Karena berdasarkan kosinus sudut antara dua vektor, maka nilainya berkisar pada 0 sampai dengan 1, dimana 0 menandakan bahwa kedua dokumen tidak mirip sama sekali, dan 1 menandakan bahwa antara *query* dan dokumen benar-benar identik. *Cosine* dinyatakan sebagai berikut [10]:

$$\cos(q, d_j) = \frac{\sum_{t_k} [TFIDF(t_k, q)] \bullet [TFIDF(t_k, d_j)]}{\sqrt{\sum |TFIDFq|^2} \bullet \sqrt{\sum |TFIDFd_j|^2}}, \quad (6)$$

dimana $cos(q,d_j)$ merupakan nilai kosinus antara *query* dan dokumen *j*, sedangkan $TFIDF(t_k,q)$ dan $TFIDF(t_k,d_j)$ adalah pembobotan *TFIDF* kata $t_k$ pada *query* dan dokumen *j*. |*TFIDF*q| dan $|TFIDFd_j|$ adalah panjang dari vektor *query* q dan dokumen. Sebagai contoh $||d_i||^2 = (TFIDFt_1^2 + TFIDFt_2^2 + TFIDFt_3^2 + ... + TFIDFt_k^2)^{1/2}$, dimana $TFIDFt_k$ adalah bobot kata ke-$t_k$ pada vektor dokumen $d_i$.

## 4. Hasil dan Pembahasan

Data yang digunakan dalam uji coba ini merupakan *corpus* atau kumpulan dokumen teks berbahasa Arab, yang diambil dari 13 kitab dalam perangkat lunak *Maktabah Syamilah.* halaman kitab-kitab sebagai suatu dokumen. Jumlah total dokumen dari seluruh kitab tersebut adalaha 6996 dokumen yang tersebar dalam 5 kategori. Dan dari seluruh dokumen dataset tersebut terdapat 47.447 kata bebeda (*distinct term*).

Pengujian dilakukan pada 7 *query* yang memiliki lebih dari satu dokumen hasil pencarian yang relevan. Pengujian ini juga dilakukan dengan memakai beberapa variasi *feature selection*, yaitu 1000, 500, dan 250 fitur terbaik. *Ground Truth* yang dipakai pada pengujian ini berasal dari data

*expert* yang berisi daftar *query* beserta dokumen-dokumen hasil pencarianya yang relevan. Dokumen yang dimaksud di sini adalah halaman tertentu dari sebuah buku.
Pada pengujian ini dilakukan pengukuran *precision*, *recall*, dan *F-Measure*. Hasil uji coba dengan menggunakan metode *term weighting* TF.IDF.ICF.IBF dan dibandingkan dengan beberapa metode *term weighting* yang ada sebelumnya. Metode-metode *term weighting* ini bukan hanya diterapkan pada perhitungan *cosine similarity*nya, akan tetapi diterapkan juga pada waktu melakukan *feature selection*. Untuk metode TF.IDF, *feature selection* yang digunakan adalah metode *mean* TF.IDF, sedangkan untuk TF.IDF.ICF *feature selection* yang digunakan adalah metode *mean* TF.IDF.ICF dan seterusnya.

Perbandingan nilai *precision, recall*, dan *F-Measure* masing-masing metode dengan menggunakan 1000 fitur terbaik dapat dilihat pada Tabel 1. Sedangkan hasil pengujian untuk feature selection 500 fitur terbaik dapat dilihat pada Tabel 2 dan hasil pengujian untuk feature selection 250 fitur terbaik dapat dilihat pada Tabel 3. Dari Tabel 1, 2, dan 3 dapat dilihat bahwa metode *term weighting* TF.IDF.ICF.IBF terbukti bisa diimplementasikan untuk pencarian query yang memiliki lebih dari satu dokumen relevan. Dibandingkan dengan tiga metode yang lain, metode term weighting TF.IDF.ICF.IBF memiliki *precision, recall*, dan *F-Measure* yang lebih tinggi pada semua variasi *feature selection*. Nilai evaluasi terbaik dari metode ini didapatkan ketika menggunakan 1000 feature terbaik yaitu *precision* sebesar 76%, *recall* sebesar 74%, dan *F-Measure* 75%. Sedangkan metode TF.IDF.IBF menempati posisi kedua dengan nilai evaluasi terbaik ketika menggunakan 1000 feature terbaik yaitu *precision* sebesar 68%, *recall* sebesar 62%, dan *F-Measure* 65%. Dari Tabel 1, 2, dan 3 juga dapat dilihat bahwa metode TF.IDF mengalami penurunan performa yang signifikan pada penggunaan jumlah fitur yang sangat sedikit. Hal ini menunjukkan bahwa metode TF.IDF banyak kehilangan fitur-fitur penting ketika hanya sedikit jumlah fitur yang digunakan.

**Tabel 1. Hasil Pengujian Kedua dengan Menggunakan 1000 Fitur**

| No. | TF.IDF | | TF.IDF.ICF | | TF.IDF.IBF | | TF.IDF.ICF.IBF | |
|---|---|---|---|---|---|---|---|---|
| | P | R | P | R | P | R | P | R |
| Q1 | 1.00 | 1.00 | 0.50 | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 |
| Q2 | 0.5 | 0.25 | 0.5 | 0.25 | 0.5 | 0.25 | 0.75 | 0.75 |
| Q3 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 |
| Q4 | 0.1 | 0.33 | 0.167 | 0.33 | 0.167 | 0.33 | 0.29 | 0.67 |
| Q5 | 1.00 | 0.5 | 1.00 | 0.5 | 1.00 | 0.5 | 1.00 | 0.5 |
| Q6 | 0.33 | 0.5 | 0.33 | 0.5 | 0.33 | 0.5 | 0.5 | 0.5 |
| Q7 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| **Rata-rata** | **67%** | **62%** | **61%** | **55%** | **68%** | **62%** | **76%** | **74%** |
| **F1** | **64%** | | **58%** | | **65%** | | **75%** | |

**Tabel 2. Hasil Pengujian Kedua dengan Menggunakan 500 Fitur**

| Nilai | TF.IDF | | TF.IDF.ICF | | TF.IDF.IBF | | TF.IDF.ICF.IBF | |
|---|---|---|---|---|---|---|---|---|
| | P | R | P | R | P | R | P | R |
| Rata-rata | 56% | 58% | 59% | 58% | 60% | 58% | 66% | 65% |
| F1 | 57% | | 58% | | 59% | | 66% | |

Dari semua hasil pengujian, dapat dilihat bahwa metode baru term weighting TF.IDF.ICF.IBF terbukti berhasil diimplementasikan dalam perangkingan dokumen berbahasa arab dengan tingkat akurasi, *precision* dan *recall* yang tinggi. Metode ini juga terbukti memiliki nilai evaluasi

yang lebih baik dibandingkan dengan beberapa metode lain. Metode ini mampu mencari dokumen yang relevan terhadap *query* yang dimasukkan dengan memperhatikan bukan hanya indeks dokumen, tetapi juga indeks buku dan kelas. Hal ini memungkinkan metode ini untuk mendapatkan dokumen yang relevan dari buku dan kategori yang tepat sesuai dengan karakteristik *query* yang dimasukkan sehingga hasil pencarianya pun semakin akurat. Nilai terbaik metode ini didapatkan ketika menggunakan 1000 feature terbaik yaitu *precision* sebesar 76%, *recall* sebesar 74%, dan *F-Measure* 75%.

**Tabel 3. Hasil Pengujian Kedua dengan Menggunakan 250 Fitur**

| Nilai | TF.IDF | | TF.IDF.ICF | | TF.IDF.IBF | | TF.IDF.ICF.IBF | |
|---|---|---|---|---|---|---|---|---|
| | P | R | P | R | P | R | P | R |
| Rata-rata | 51% | 51% | 55% | 51% | 57% | 51% | 54% | 63% |
| F1 | 51% | | 53% | | 54% | | 58% | |

Berdasarkan hasil ujicoba pada Tabel 1, 2, dan 3 juga dapat dilihat bahwa metode TF.IDF.IBF (tanpa ICF) memiliki *precision* dan *recall* yang lebih tinggi dibandingkan dengan dua metode yang lain. Hal ini menunjukkan bahwa penambahan IBF memberikan dampak yang lebih bagus daripada ICF. Nilai evaluasi terbaik metode ini didapatkan ketika menggunakan 1000 feature terbaik yaitu *precision* sebesar 68%, *recall* sebesar 62%, dan *F-Measure* 65%.

Selain itu, dari Tabel 1, 2, dan 3 juga dapat dilihat bahwa pengurangan fitur juga berpengaruh pada performa masing-masing metode. Semakin sedikit fitur yang digunakan, semakin menurun pula performa metode-metode tersebut. TF.IDF memiliki penurunan performa yang sangat signifikan seiring berkurangnya jumlah fitur yang digunakan. Hal ini dikarenakan banyak fitur-fitur penting yang hilang ketika dilakukan pengurangan fitur. Fitur-fitur yang hilang tersebut memiliki nilai TF.IDF yang lebih kecil daripada beberapa fitur lain sehingga harus dihilangkan meski sebenarnya beberapa fitur-fitur tersebut memiliki peranan yang lebih penting. Berbeda dengan TF.IDF.ICF.IBF yang tetap memiliki performa cukup bagus walaupun hanya menggunakan sedikit fitur karena tetap bisa mempertahankan fitur-fitur yang memiliki peranan penting.

## 5. Kesimpulan

*Term Weighting* TF.IDF.ICF.IBF dapat diaplikasikan pada perangkingan dokumen berbahasa Arab yang memiliki hierarki berupa buku-buku yang memiliki banyak halaman. Hasil ujicoba menunjukkan bahwa metode ini memiliki rata - rata nilai *F-Measure* sebesar 75% , rata-rata *precision* 76% dan rata-rata *recall* mencapai 74%. Dibandingkan dengan perangkingan dokumen menggunakan metode *term weighting* yang lain meliputi TF.IDF, TF.IDF.ICF, dan TF.IDF.IBF, metode yang diusulkan memiliki *precision, recall*, dan *F-Measure* yang lebih tinggi. Metode *Term Weighting* TF.IDF.ICF.IBF terbukti berhasil digunakan dalam seleksi fitur dan perangkingan dokumen hasil pencarian dengan hierarki berupa buku-buku yang memiliki banyak halaman. Oleh karena itu pada penelitian selanjutnya, metode ini dapat diaplikasikan pada klasifikasi dokumen dengan hierarki yang sama.

## Daftar Pustaka

[1] Esraa E.A., B.L. Nagma, M.F. Tolba, An Efficient Rangking Module for an Arabic Search Engine,International Journal of Computer Science and Network Security. 2010; 10(2): 1-3.
[2] Suleiman H.M., Character Contiguity in N-gram-based Word Matching: the Case for Arabic Text Searching,Information Processing and Management, 2005; 20(4): 2-4.
[3] Suleiman H.M., Arabic String Searching in the Context of Character Code Standards and Orthographic Variations,Computer Standards and Interfaces. 1998; 4(1): 3-10.
[4] Fuji R., G.S. Mohammad, Class-indexing-based term weighting for automatic text classification,Journal of Informetrics. 2009; 3(1):2-5.

[5] Larkey, Leah S., Lisa Ballesteros, Margaret E Connell, Light Stemming for Arabic Information Retrieval,Springer Link: Text, Speech and Language Technology, 2007; 38(1):7-12.
[6] Harrag F., A. Hamdi-Cherif, E. El-Qawasmeh. Vector space model for Arabic information retrieval - application to Hadith indexing. Proceedings of the First IEEE Conference on the Applications of Digital Information and Web Technologies. ICADWIT. 2008: 107-112.
[7] Manning C.D., R. Prabhakar, S. Hinrich. An Introduction to Information Retrieval. Cambridge, England: Cambridge University Press. 2009.
[8] Salton G. Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer. New York: Addison-Wesly. 1989.
[9] Ahmad N., Z.A. Agus, Diana P., Implementasi N-Gram Dalam Pencarian Teks Sebagai Penunjang Aplikasi Perpustakaan Kitab Berbahasa Arab. Under Graduate Thesis. Surabaya: Under Graduate ITS.
[10] http://www.miislita.com/term-vector/term-vector-3.html, diakses tanggal 5 Mei 2013.