Perangkingan Dokumen Berbahasa Arab Berdasarkan *Query* dengan Metode Klasifikasi *Naïve Bayes* dan *K-Nearest Neighbor*

Arabic Document Ranking based on Query using Naïve Bayes and K-Nearest Neighbor Classification Method

Usfita Kiftiyani[1], Suprapto[2], Novanto Yudistira[3]
[1] Teknik Informatika, Universitas Islam Negeri Sunan Kalijaga Yogyakarta
[1,2,3] Teknik Informatika, Universitas Brawijaya
E-mail: [1]usfita.kiftiyani@gmail.com, [2]spttif@ub.ac.id, [3]yudistira@ub.ac.id

Abstrak

Penelitian tentang perangkingan dokumen pada temu kembali informasi saat ini mudah ditemukan, hal ini terkait perkembangan keilmuan dibidang penggalian informasi yang bergerak sangat cepat. Namun, Walaupun sudah penelitian yang menggunakan Bahasa Arab sebagai objek masih terbatas. Karena keterbatasan penggunaan dokumen Bahasa Arab untuk penelitian bidang penggalian informasi maka penulis mencoba melakukan pendekatan sederhana, yaitu dengan mengimplementasikan metode klasifikasi *naïve bayes* dan *k-Nearest Neighbor* (k-NN). Tujuan dari penelitian ini adalah untuk mengetahui apakah metode klasifikasi terutama *naïve bayes* dan k-NN dapat digunakan untuk melakukan perangkingan, dan juga membandingkan akurasi dari kedua metode tersebut. Berdasarkan penelitian yang dilakukan, didapatkan hasil bahwa perangkingan dengan metode klasifikasi dapat dilakukan dengan tingkat akurasi metode *Naïve Bayes* lebih baik dibandingkan dengan metode k-NN dengan rata-rata nilai F1 Measure mencapai 72%, rata-rata nilai precision mencapai 75%, dan rata-rata nilai recall mencapai 80%. Sedangkan hasil dari metode k-NN diperoleh rata-rata nilai F1 Measure mencapai 70%, rata-rata nilai precision mencapai 76%, dan rata-rata nilai recall mencapai 79%. Namun penelitian ini masih kurang dari segi teknik yang dilakukan, yaitu dengan menghilangkan proses stemming. Sehngga penulis memberikan saran untuk penelitian selanjutnya supaya bisa dilakukan proses stemming dan menggunakan metode perangkingan yang lebih baru.

Kata kunci: perangkingan dokumen, *naïve bayes*, k-*nearest neighbor*

Abstract

Research about document ranking on information retrieval is now easy to find, this is related to scientific developments in the field of extracting information that is moving very fast. However, research that used Arabic documents as an object it is still limited. Due to the limited use of Arabic documents for research in the field of extracting information, the author tries to take a simple approach, by implementing the Naïve Bayes and the k-Nearest Neighbor (k-NN) classification method. The purpose of this study was to determine whether the classification methods, especially Naïve Bayes and k-NN, can be used to rank, and also compare the accuracy of the two methods. Based on this research, it was found that the ranking with the classification method can be done with the accuracy level of the Naïve Bayes method is better than the k-NN method with an average F1 Measure value reaching 72%, the average value of precision is 75%, and the average recall value reaches 80%. Meanwhile, the results of the k-NN method showed that the average value of F1 Measure reached 70%, the average value of precision was 76%, and the average recall value reached 79%. However, this research is still lacking in terms of the technique used, which is by eliminating the stemming process. So the authors provide suggestions for further research so that the stemming process can be carried out and using a newer ranking method.

Keywords: document ranking, naïve bayes, k-nearest neighbor

1. PENDAHULUAN

Perangkingan dokumen adalah salah satu *task* dari temu kembali informasi. Pada umumnya, perangkingan dokumen pada temu kembali informasi ini dilakukan berdasarkan tingkat relevansi dokumen terhadap suatu *query*[1] yang biasa digunakan sebagai *keyword* pencarian. Banyak *task* dari temu kembali informasi yang kemudian disajikan dalam bentuk sistem yang disebut dengan sistem temu kembali informasi.

Dalam Sistem Temu Kembali informasi, pengguna membutuhkan informasi yang sangat beragam. Keberagaman itu juga terjadi dalam keberagaman Bahasa yang digunakan sebagai sumber informasinya. Sumber informasi dalam Bahasa Inggris banyak digunakan sebagai acuan dan objek penelitian dalam bidang temu kembali informasi. Namun, saat ini sudah semakin banyak penelitian yang menggunakan teks berbahasa lain seperti Bahasa Arab, Cina, ataupun Bahasa Indonesia sebagai objek penelitiannya.

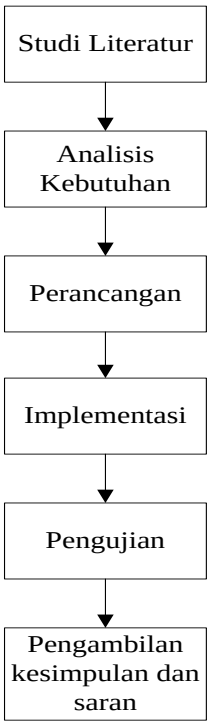
Sebagai salah satu objek penelitian dalam bidang temu kembali informasi, Bahasa Arab mempunyai keunikan dibandingkan dengan Bahasa lainnya yaitu morfologi Bahasa Arab yang lebih kaya dan kompleks [2]. Penelitian sebelumnya yang membahas tentang perangkingan dokumen berbahasa inggris telah dilakukan, beberapa diantaranya adalah perangkingan berdasarkan susunan posisi kata dari *query* [3], perangkingan dengan pencocokan kata berbasis N-gram dan SVM [4], perangkingan dokumen berdasarkan konteks [5]. Sedangkan penelitian tentang dokumen berbahasa Arab yang telah dilakukan sebelumnya antara lain perangkingan informasi atau dokumen berdasarkan pembobotan kata [6], [7].

Metode yang digunakan untuk melakukan perangkingan dokumen bermacam-macam salah satunya dalam [4] perangkingan dokumen dilakukan dengan meghitung kemiripan katadengan N-gram dan SVM. Sedangkan pada penelitian ini penulis melakukan pendekatan yang lebih sederhana yaitu dengan mengimplementasikan metode klasifikasi *Naïve Bayes* dan *K-Nearest Neighbor* (k-NN). Pada topik klasifikasi, metode *Naïve Bayes* dikenal sebagai metode klasifikasi yang meggunakan algoritma probabilitas sederhana [8] dan sudah banyak digunakan baik untuk klasifikasi dokumen, pola, maupun gambar. Dokumen yang diklasifikasikan pun bervariasi dalam Bahasa. Penelitian sebalumnya yang membandingkan algoritma-algoritma klasifikasi untuk teks berbahasa arab [9] menyatakan bahwa tingkat akurasi metode *Naïve Bayes* lebih tinggi daripada metode k-NN tetapi *running time* yang dibutuhkan metode *Naïve Bayes* lebih lamadari metode k-NN. Namun dalam hal ini metode Naïve Bayes dan k-NN belum pernah digunakan untuk melakukan perangkingan dokumen.

Sehingga pada penelitian ini penulis melakukan implementasi metode *Naïve Bayes* dan metode k-NN untuk perangkingan pada dokumen berbahasa arab berdasarkan *query*. Dalam proses ekstraksi kata pada pemrosesan teks dalam Bahasa latin baik pada Bahasa Indonesia maupun Bahasa Inggris terdapat beberapa metode stemming, namun dalam bahasa arab belum ada metode *stemming* yang baku sehingga proses stemming tidak digunakan dalam penelitian ini. Dengan implementasi metode *Naïve Bayes* dan k-NN ini diharapkan dapat diketahui keakuratan kedua metode tersebut-yang pada dasarnya merupakan metode klasifikasi-jika digunakan untuk perangkingan dokumen.

2. METODE PENELITIAN

Secara garis besar tahapan-tahapan yang dilakukan dalam untuk penelitian ini diperlihatkan pada gambar 1.



Gambar 1 Tahapan metode penelitian

2.1 Study Literatur

Kajian terhadap teori-teori yang dibutuhkan untuk melakukan penelitian dilakukan pada tahap ini. Tahapan ini berguna untuk menghimpun informasi sehingga tahapan-tahapan selanjutnya dapat dilakukan sesuai teori dan meningkatkan efektifitas proses penelitian secara keseluruhan. Teori-teori yang dibutuhkan diantaranya adalah teori tentang perangkingan dokumen, metode klasifikasi *Naïve Bayes* dan K-*Nearest Neighbor*.

Perangkingan Dokumen

Perangkingan dokumen adalah salah satu task pada temu kembali informasi. Model konvensional dari perangkingan bisa dikategorikan menjadi dua bagian yaitu *query-dependent* dan *query-independent* model [10]. Pada penelitian ini akan menggunakan *query-dependent model*. Model ini merupakan model awal dari temu kembali informasi dimana dokumen yang dikembalikan hanyalah dokumen yang mempunyai nilai kemunculan *query* dalam dokumen tersebut. Hal tersebut menandakan bahwa dokumen yang dikembalikan mempunyai keterkaitan dengan *query*.

Metode-metode yang digunakan dalam perangkingan dokumen sangatlah banyak, akan tetapi pada penelitian ini penulis melakukan pendekatan sederhana pada perangkingan dokumen berbahasa Arab dengan menerapkan metode klasifikasi yaitu *naïve bayes* dan *k*-NN.

Text Mining

Text mining adalah teknik menggali informasi informasi dari sekumpulan dokumen menggunakan metode analisis tertentu. *Text mining* dan data mining, keduanya adalah teknik menggali informasi dari data teks structural maupun semi-struktural [11]. Perbedaannya data mining mengatasi penggalian data structural, sedangkan text mining mengatasi penggalian data text yang mempunyai struktur data yang tidak terstruktur.

Text Preprocessing

Tujuan dari dilakukannya *preprocessing* adalah untuk mengoptimalkan tahapan pemrosesan teks selanjutnya [12]. Hasil akhir dari tahapan ini adalah teks awal yang sudah dipecah menjadi kata perkata dan juga sudah mengalami perubahan struktur. Pada penelitian ini tahapan-tahapan dari *text preprocessing* yang dilakukan adalah:

- *Part Of Speech Tagging* adalah proses memberikan tanda pada masing-masing kata dalam kalimat berdasarkan jenis kata [13] pada tata bahasa pada umumnya misalnya kata kerja, kata benda, kata sifat, kata depan, dan sebagainya.
- Tokenisasi adalah pemecahan kalimat menjadi kumpulan kata-kata.
- *Case Folding* yaitu proses perubahan case huruf, dalam hal ini kami melakukan penyamaan semua huruf menjadi huruf kecil.
- *Stopword removal* atau penghapusan kata-kata yang tidak mempunyai makna, misalnya kata penghubung, dll.
- *Stemming* adalah tahap mencari bentuk dasar dari setiap kata yang muncul. Pada penelitian ini penulis tidak melakukan proses stemming karena proses stemming untuk Bahasa Arab masih jarang sekali dilakukan.

Term Weighting

Term Weighting adalah teknik untuk menghitung jumlah kemunculan setiap kata pada dokumen sebagai ukuran keterkaitan antara kata dan dokumen tersebut [6]. Pada penelitian ini akan digunakan teknik pembobotan TF-IDF untuk melakukan pembobotan pada setiap kata. TF-IDF menerapkan pembobotan kombinasi antara perkalian bobot lokal(*term frequency*) dan bobot global (*global inverse document frequency*).

Algoritma Naïve Bayes

Naive bayes Classifier adalah metode pengklasifikasian paling sederhana dari model pengklasifikasian dengan peluang, dimana diasumsikan bahwa setiap atribut contoh (data sampel) bersifat saling lepas satu sama lain berdasarkan atribut kelas.

Naive bayes classifier banyak digunakan dalam melakukan klasifiksi dokumen teks. Pada penerapannya, setiap posisi kata dalam dokumen harus diposisikan atau dianggap sebagai atribut. Persamaan yang digunakan untuk mengklasifikasikan dokumen dengan metode *Naive bayes* adalah sebagai berikut [14]:

$$P(kategori \mid kata) = \frac{P(kata \mid kategori)P(kategori)}{P(kata)} \quad (1)$$

Keterangan :

P(*kategori*|*kata*) : Peluang kategori tertentu untuk kemunculan sebuah kata.

Jika data yang digunakan merupakan data kontinyu maka persamaan yang digunakan adalah:

$$\varphi_{\mu,\sigma}(kata) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

Dengan asumsi bahwa teorema bayes bersifat *independence* (saling bebas) maka menyebabkan setiap kata pada setiap kategori menjadi independen antara satu dengan yang lainnya [14]. Sehingga persamaan menjadi :

$$P(a_1, a_2, a_3, ..., a_n \mid v_j) = \prod_i P(a_i \mid v_j) \quad (3)$$

Himpunan set dari nilai-nilai probabilitas ini berkorespondensi dengan hipotesa yang ingin dipelajari. Hipotesa kemudian digunakan untuk mengklasifikasi data-data baru. Pada pengklasifikasian teks, perhitungan rumus probabilitas dapat didefinisikan [14] :

$$P(v_j) = \frac{docs_j}{examples} \quad (4)$$

$$P(w_k \mid v_j) = \frac{n_k+1}{n+|kata|} \quad (5)$$

Keterangan :

$P(v_j)$: Peluang jumlah dokumen kategori tertentu terhadap seluruh dokumen

$P(w_k \mid v_j)$: Peluang kategori w_k ketika terdapat kemunculan sebuah kata v_j.

$docs_j$: dokumen dengan nilai target v_j

examples: jumlah dokumen yang digunakan pada proses pelatihan

n: jumlah semua kata yang terdapat di dalam dokumen dengan nilai fungsi target yang sesuai.

n_k: jumlah munculnya kata w_k pada semua dokumen dengan nilai fungsi target yang sesuai.

|kata|: jumlah unik kata yang muncul pada seluruh dokumen yang digunakan.

K-Nearest Neighbor

K-NN merupakan *instance-based learning*, dimana data training disimpan sehingga klasifikasi untuk record baru yg belum diklasifikasi dapat ditemukan dengan membandingkan kemiripan yang paling banyak dalam *data training* atau *data learning*. Metode ini memanfaatkan mekanisme *voting* dari k buah objek terdekat dan bila hasil *voting* seri, maka label untuk objek akan dipilih berdasarkan pengurutan[12].

Terdapat dua macam mekanisme *voting* yaitu *simple unweighted voting* dan *weighted voting*. *Simple unweighted voting* dilakukan dengan menentukan k yaitu jumlah rekord yang memiliki suara dalam pengklasifikasian rekord baru, kemudian membandingkan rekord baru ke k-nn. *Weighted voting* merupakan kebalikan proporsi jarak dari record baru dengan klasifikasi. Vote dari *weighted voting* dibobotkan dengan *inverse square* dari nilai jarak. Pada penerapannya, setiap posisi kata dalam dokumen diposisikan atau dianggap sebagai atribut. Jarak antara dua *query* dengan data *learning* dihitung dengan rumus *Euclidean Distance* [15]. Persamaan *Euclidean distance* sebagai berikut :

$$D_{euc}(P,Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (6)$$

Keterangan :

P dan Q: titik pada ruang vector *n*-dimensi.

p_i dan q_i: besaran scalar untuk dimensi ke-i dalam ruang vector *n*-dimensi.

2.2 Analisis Kebutuhan

Setelah mengetahui teori dan kebutuhan untuk melakukan penelitian, selanjutnya bisa dianalisa apa yang harus dilakukan untuk memenuhi kebutuhan tersebut pada tahap ini. Analisis kebutuhan ini juga termasuk analisis kebutuhan sistem, data, dan metode.

2.3 Perancangan

Tahap perancangan ini dilakukan setelah semua kebutuhan sistem didapatkan melalui analisa kebutuhan. tahap perancangan yang dilakukan terdiri dari perancangan *preprocessing* dokumen, perancangan transformasi teks, perancangan perhitungan frekuensi kata, perancangan *pattern discovery*. Hasil perancangan ini akan diimplementasikan menjadi sistem perangkingan dokumen.

2.4 Implementasi

Pada tahap ini akan dilakukan implementasi pembuatan sistem dengan mengacu pada perancangan sistem pada tahap sebelumnya. Uraian implementasi pembuatan sistem ini dibagi mejadi dua bagian yaitu uraian tentang lingkungan sistem dan uraian tentang implementasi

sistem.

2.5 Pengujian

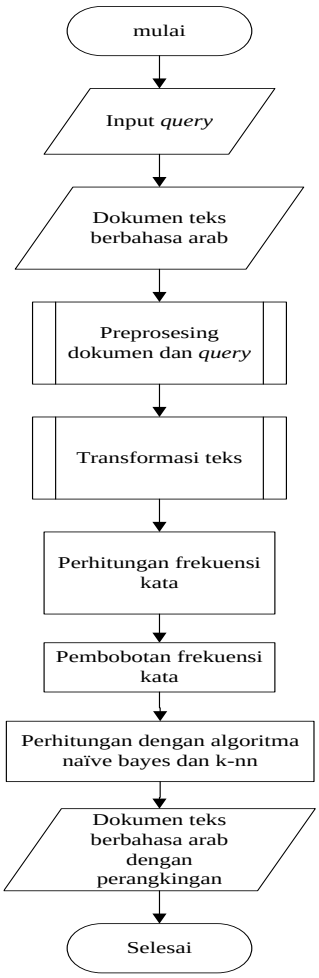
Pengujian hasil kerja sistem yang telah dibuat sekaligus evaluasi terhadap sistem dilakukan pada tahap ini. Sehingga dapat diketahui hasil perangkingan dokumen pada sistem yang nantinya dijadikan sebagai kesimpulan untuk pelaporan pembuatan sistem perangkingan dokumen berbahasa arab. Pengujian ini dilakukan untuk membandingkan nilai *precision recall* dan *f-measure* antara hasil perangkingan dengan metode *naive bayes* dan perangkingan dengan metode k-NN.

2.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah proses pengujian sistem selesai sehingga dapat diketahui efektifitas kinerja sistem. Tahap terakhir yaitu penulisan saran yang dapat membantu dalam pengembangan sistem selanjutnya

3. HASIL DAN PEMBAHASAN

Diagram alir untuk menggambarkan arus data yang membantu dalam proses memahami jalannya implementasi sistem ini digambarkan dalam gambar 2.



Gambar 2 Diagram alir implementasi system

Proses-proses yang terjadi dalam sistem seperti yang ditunjukkan pada gambar 2 dapat diuraikan sebagai berikut :

3.1 Preprosesing dokumen dan query

Preprocessing dilakukan dua kali yaitu pada dokumen sebagai data set yang akan dirangking dan pada *query* yang digunakan sebagai *keyword* pencarian atau perangkingan dalam sistem ini.

Dokumen dataset yang diolah dalam penelitian ini didapatkan dari aplikasi open source kitab *Maktabah Syamilah*. Proses *preprocessing* terdiri dari tokenisasi dan *parsing.* Proses tokenisasi dilakukan untuk menghilangkan karakter selain abjad yang selanjutnya hasil dari proses tokenisasi ini akan diparsing atau dipisahkan per kata untuk setiap dokumen. Proses memecah kalimat-kalimat menjadi satuan kata ini dipisahkan dengan berdasarkan spasi. Sama halnya dengan proses tokenisasi, proses *parsing* ini juga dilakukan terhadap dokumen satu per satu.

3.2 Transformasi teks

Proses transformasi dokumen dilakukan untuk learning dokumen setelah *preprosesing*, sedangkan proses transformasi pada *query* akan menghasilkan fitur-fitur yang digunakan untuk melakukan perangkingan dokumen.

proses transformasi dilakukan tanpa proses *stemming* sehingga hanya akan dilakukan penghapusan *stopword.* Proses penghapusan *stopword* dilakukan terhadap data hasil *parsing* yang telah tersimpan dalam *database*, untuk setiap kata dalam stopword akan dilakukan penghapusan pada data hasil *parsing* hingga kata dalam *stopword* habis atau telah dicek.

3.3 Perhitungan dan pembobotan frekuensi kata

Perhitungan frekuensi kata digunakan untuk mendapatkan frekuensi kata yang terkandung dalam setiap dokumen. Proses ini dilakukan dengan menghitung daftar term hasil *parsing* masing-masing dokumen yang telah disimpan ke dalam database untuk selanjutnya digunakan sebagai master dataset.

3.4 Perhitungan dengan Algoritma Naïve Bayes dan k-NN

Proses perangkingan dokumen dengan algoritma *naïve bayes* atau k-NN akan menghasilkan dokumen teks berbahasa arab dengan perangkingan sesuai yang didapatkan dengan algoritma *naïve bayes* ataupun k-NN.

Implementasi algoritma *naïve bayes* diawali dengan perhitungan prior ($P(v_j)$) Kemudian dilakukan perhitungan variansi dan rerata dari setiap fitur atau variable. Jika prior, varian dan rerata sudah dihitung, kemudian dilakukan perhitungan *likelihood* $\varphi_{\mu,\sigma}(kata)$. Yang terakhir dilakukan perhitungan probabilitas dokumen terhadap kemunculan semua kata (*posterior*) yang mana posterior akan digunakan sebagai acuan nilai perangkingan dokumen.

Implementasi algoritma k-NN yang dilakukan yaitu menghitung jarak antara dokumen dengan *query* berdasarkan frekuensi kata tertentu yang terpilih sebagai fitur dalam dokumen. Kata-kata yang digunakan sebagai fitur adalah kata-kata hasil *parsing* dari *query.* Implementasi algoritma k-NN diawali tanpa proses normalisasi sehingga lagsung dilakukan perhitungan jarak query terhadap seluruh dokumen. hasil perhitungan jarak tersebut akan digunakan untuk acuan perangkingan dokumen dengan metode k-NN.

3.5 Hasil dan Pengujian

Pengujian efektifitas terhadap sistem ini akan menggunakan metode *precision, recall*, dan *F1 measure*. Pengujian dilakukan dengan dua macam pengolahan yaitu dengan pembobotan term (*term weighted*) dan tanpa pembobotan term.

Untuk mempelajari pengaruh threshold terhadap hasil perangkingan dokumen oleh sistem maka dilakukan 3 kali uji coba dengan nilai threshold yang berbeda yaitu 0.5, 0.8 dan 1. Nilai threshold tersebut dihitung dari jumlah data yang didapatkan ketika proses pencarian.

Tabel 1 berikut merupakan jumlah data yang dihasilkan dari proses pencarian :

Tabel 1 Data Hasil Pencarian

No	Query	Sistem	Expert
1	قنوت	67	67
2	حَقِيْقَةِ الْقَبْضِ	39	8
3	كُسُوفِ	68	68
4	الْمَاءُ الْمُطْلَقُ	8	10
5	أَحْكَامِ الإِمَامِ وَفِيْهِ مَسَائِلٌ	1	1
6	يُكْرَهُ الدَّفْنُ بِاللَّيْلِ	4	3
7	غَايَةٌ فِي وُجُوْبِ الْعِدَّةِ	12	2
8	أَنَّ لِزَكَاةِ التِّجَارَةِ شُرُوْطًا	16	16
9	اِمْرَأَةً حَامِلاً مِنَ الزِّنَا صَحَّ نِكَاحُهُ بِلاَخِلاَفٍ	8	8
10	الرجل استعمال الحرير	11	11

Pada tabel 1 diperlihatkan bahwa hasil pencarian yang dilakukan oleh sistem mendapatkan hasil yang beragam berdasarkan *query* yang dimasukkan. *Query*-query tersebut didapatkan dari *index bhatsul masail*. Hasil tersebut didapatkan dari pencarian oleh sistem dan oleh *expert* pada data yang sama, namun perbedaan term dalam *query* yang dimasukkan akan mempengaruhi hasil pencarian.

Rata-rata perhitungan *precision, recall* dan *F1 measure* dari algoritma *Naïve Bayes* dapat dilihat pada tabel 2.

Tabel 2 Rata-rata Hasil Perangkingan algoritma Naïve Bayes

No	Threshold	Naïve Bayes					
		Weighted(%)			Without Weighting(%)		
		R	P	F1	R	P	F1
1	0.5	82	50	60	76	46	55
2	0.8	81	81	76	81	80	75
3	1	81	96	83	81	96	83

Rata-rata perhitungan *precision, recall* dan *F1 measure* dari algoritma *k-NN* dapat dilihat pada tabel 3.

Tabel 3 Rata-rata Hasil Perangkingan algoritma K-NN

No	Threshold	K-NN					
		Weighted(%)			Without Weighting(%)		
		R	P	F1	R	P	F1
1	0.5	77	58	61	75	48	42
2	0.8	81	81	76	80	76	74
3	1	81	96	83	81	96	83

Berikut perbandingan rata-rata nilai *precision, recall* dan F1 *Measure* Algoritma *Naïve Bayes* dan k-NN yang diperlihatkan pada tabel 5.10. Pergerakan nilai *precision, recall* dan F1 *Measure* pada tabel 5.10 digambarkan dengan grafik pada gambar 5.5 dan 5.6.

Tabel 4 Perbandingan Rata-rata precision, recall dan F1 Measure Algoritma Naïve Bayes dan k-NN

No	Threshold	Naïve Bayes (%)			k Nearest Neighbour		
		R	P	F1	R	P	F1
1	0.5	79	48	57	76	53	51
2	0.8	81	81	76	81	79	75
3	1	81	96	83	81	96	83
Rata-rata		80	75	72	79	76	70

Perbandingan akurasi perangkingan dokumen dengan algoritma *naïve bayes* dan k-NN dapat dilihat pada tabel 4 kolom f1 *measure*. Pada tabel 4 ditunjukkan bahwa nilai f1 *measure* terendah untuk kedua metode sama-sama pada threshold 0.5 dan nilai tertinggi terdapat pada threshold 1. Rata-rata nilai f1 *measure* menunjukkan tingkat akurasi secara keseluruhan dari masing-masing metode dengan beragam nilai threshold, dengan pembobotan atau tanpa pembobotan didapatkan bahwa algoritma *naïve bayes* mendapatkan nilai rata-rata f1 *measure* lebih baik dari nilai f1 *measure* algoritma k-NN. Nilai rata-rata f1 *measure* algoritma *naïve bayes* mencapai nilai akurasi 72%. Sedangkan untuk algoritma k-NN nilai f1 measure hanya mencapai mencapai nilai akurasi 70%. Hal tersebut menunjukkan bahwa perangkingan dokumen berbahasa arab dengan algoritma *naïve bayes* lebih baik daripada algoritma k-NN.

4. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil setelah melakukan pengujian terhadap hasil perangkingan dokumen berbahasa arab menggunakan metode *Naïve Bayes* dan k-NN adalah metode *Naïve Bayes* menunjukkan tingkat akurasi 72% yang mana lebiih tinggi dibandingkan dengan metode k-NN yang menunjukkan tingkat akurasi 70%.

DAFTAR PUSTAKA

[1] R. Deveaud, J. Mothe, Z. Ullah, and J.-Y. Nie, "Learning to Adaptively Rank Document Retrieval System Configurations Learning to Adaptively Rank Docu-ment Retrieval System Configurations," *ACM Trans. Inf. Syst.*, vol. 37, no. 1, pp. 1–41, 2018, doi: 10.1145/3231937ï.

[2] A. Khemakhem, B. Gargouri, A. Ben Hamadou, and G. Francopoulo, "ISO standard modeling of a large Arabic dictionary," *Nat. Lang. Eng.*, vol. 22, no. 6, pp. 849–879, Nov. 2016, doi: 10.1017/S1351324915000224.

[3] A. Montazeralghaem, R. Rahimi, and J. Allan, "Relevance Ranking Based on Query-

Aware Context Analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12035 LNCS, pp. 446–460, doi: 10.1007/978-3-030-45439-5_30.

[4] N. Othman and R. Faiz, "Question answering passage retrieval and re-ranking using n-grams and SVM," *Comput. y Sist.*, vol. 20, no. 3, pp. 483–494, 2016, doi: 10.13053/CyS-20-3-2470.

[5] W. U. Ahmad, K.-W. Chang, and H. Wang, "Context Attentive Document Ranking and Query Suggestion," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, p. 10, Accessed: Sep. 23, 2020. [Online]. Available: https://doi.org/10.1145/3331184.3331246.

[6] M. A. Fauzi, A. Z. Arifin, and A. Yuniarti, "Arabic Book Retrieval using Class and Book Index Based Term Weighting," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, p. 3705, Dec. 2017, doi: 10.11591/ijece.v7i6.pp3705-3710.

[7] K. F. H. Holle, A. Z. Arifin, and D. Purwitasari, "PREFERENCE BASED TERM WEIGHTING FOR ARABIC FIQH DOCUMENT RANKING," *J. Ilmu Komput. dan Inf.*, vol. 8, no. 1, p. 45, 2015, doi: 10.21609/jiki.v8i1.283.

[8] A. H. Aliwy and E. H. A. Ameer, "Comparative Study of Five Text Classification Algorithms with their Improvements," *Int. J. Appl. Eng. Res.*, vol. 12, pp. 4309–4319, 2017, Accessed: Sep. 19, 2020. [Online]. Available: http://www.ripublication.com.

[9] G. Raho, G. Kanaan, R. Al-Shalabi, and A. ' Anassar, "Different Classification Algorithms Based on Arabic Text Classification: Feature Selection Comparative Study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 2, 2015.

[10] J. Devezas and S. Nunes, "Hypergraph-of-entity: A unified representation model for the retrieval of text and knowledge," *Open Comput. Sci.*, vol. 9, no. 1, pp. 103–127, Jan. 2019, doi: 10.1515/comp-2019-0006.

[11] B. Nie and S. Sun, "Using Text Mining Techniques to Identify Research Trends: A Case Study of Design Research," *Appl. Sci.*, vol. 7, no. 4, p. 401, Apr. 2017, doi: 10.3390/app7040401.

[12] R. Alhutaish and N. Omar, "Arabic Text Classification using K-Nearest Neighbour Algorithm," *Int. Arab J. Inf. Technol.*, vol. 12, no. 2, 2015.

[13] A. A. Suryani and W. Maharani, "Part of Speech Tagging for Javanese Language with Hidden Markov Model," *J. Comput. Sci. Informatics Eng.*, vol. 4, no. 1, pp. 84–91, Jun. 2020, doi: 10.29303/jcosine.v4i1.346.

[14] S. Karthina and N. Sairam, "A Naïve Bayesian Classifier for Educational Qualification," *Indian J. Sci. Technol.*, vol. 8, no. 16, pp. 1–5, Jul. 2015.

[15] L. Y. Hu, M. W. Huang, S. W. Ke, and C. F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *Springerplus*, vol. 5, no. 1, Dec. 2016, doi: 10.1186/s40064-016-2941-7.