

Implementasi *Relevant Feedback* Menggunakan Algoritma Genetika pada Dokumen Bahasa Indonesia

Implementation of Relevant Feedback Using Genetic Algorithm in Indonesian Documents

Ika Oktavia Suzanti¹, Fifin Ayu Mufarroha²

^{1,2}Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo Madura,
PO. BOX 2 Telang, Kamal – Bangkalan, Indonesia

¹iosuzanti@trunojoyo.ac.id, ²fifin.mufarroha@trunojoyo.ac.id

Naskah diterima: 6 Agustus 2021, direvisi: 9 September 2021, disetujui: 17 November 2021

Abstract

Search engines have been widely used to assist users in getting the information needed. Current search engines have limitations of the query. Often, the query is not correct, yet the user assumes that the information needed is in the search engine. As a result, the search engine will display results even though the query is less specific. Therefore, to be able to accommodate the user requirement in finding the necessary documents, users need new information search engine. In this study, relevance feedback method was applied using genetic algorithms to improve document search by improving queries. There were two selection methods used, namely roulette wheel selection and tournament selection. By using 1000 data, the results have indicated 58% average precision, 89% average recall, and 59% average F-Measure using roulette wheel selection. Meanwhile, the use of tournament selection has generated 57% average precision, 92% average recall, and 60% average F-measure. The average computation time for roulette wheel selection is 10.23 seconds, and tournament selection is 5.97 seconds.

Keywords: Search Engine, Relevance feedback, Roulette wheel Selection, Tournament Selection.

Abstrak

Mesin pencarian telah banyak digunakan untuk membantu user dalam mendapatkan informasi yang dibutuhkan. Pada mesin pencarian saat ini, terdapat keterbatasan pada query. Sering kali, query tidak tepat tetapi user beranggapan bahwa informasi yang dibutuhkan ada pada mesin pencarian. Hasilnya, mesin pencarian akan menampilkan hasil meskipun query kurang spesifik. Oleh karena itu, untuk dapat memenuhi kebutuhan pengguna dalam menemukan dokumen yang sesuai, dibutuhkan mesin pencarian informasi baru. Pada penelitian ini, diterapkan metode relevance feedback menggunakan algoritma genetika untuk meningkatkan pencarian dokumen dengan memperbaiki query. Terdapat dua metode seleksi yang digunakan, yaitu roulette wheel selection dan tournament selection. Dengan menggunakan 1000 data, diperoleh hasil 58% rata - rata precision, 89% rata - rata recall, dan 59% rata-rata F-Measure pada roulette wheel selection. Adapun pada tournament selection diperoleh 57% rata - rata precision, 92% rata-rata recall, dan 60% rata-rata f-measure. Waktu komputasi roulette wheel selection rata-rata 10,23 detik, sedangkan untuk tournament selection adalah 5,97 detik.

Kata kunci: Mesin Pencari, Relevance Feedback, Roulette Wheel Selection, Tournament Selection.

PENDAHULUAN

Kebutuhan informasi telah menjadi hal yang tidak bisa ditawar seiring dengan perkembangan teknologi informasi. Semakin bertambahnya informasi menuntut adanya mesin pencari yang mampu menyediakan berbagai informasi. Mesin pencari tersebut mampu menyediakan informasi yang dibutuhkan oleh *user* dan menghasilkan informasi yang akurat dan relevan. Penggunaan mesin pencari sudah lazim dijumpai di dunia *online*. Namun, hasil pencarian yang didapat seringkali berbeda tergantung dari metode yang digunakan. Mekanisme pencarian umumnya mengandalkan *query* yang dimasukkan oleh *user* dan komputer kemudian mencari kemiripan *query* tersebut dengan data yang tersedia (Halim, Nyoto, dan Safriadi 2017). Informasi yang dihasilkan oleh mesin pencari bisa saja tidak relevan dengan yang diinginkan oleh *user*. Hal tersebut bisa terjadi karena *user* tidak mengerti *query* yang harus dimasukkan. Akibatnya, *user* harus kembali memasukkan *query* yang lebih sesuai. Dengan demikian, dibutuhkan suatu upaya untuk meningkatkan hasil mesin pencarian yang didasarkan pada masukan *query* dari *user* di tengah keterbatasan waktu dan komputasi. Hal tersebut dimaksudkan agar proses pencarian menghasilkan informasi yang lebih relevan tanpa adanya keharusan melakukan *input query* baru oleh *user*.

Relevance feedback (RF) merupakan proses untuk meningkatkan jumlah dokumen yang sesuai dari perolehan tanggapan mesin, atau dapat dikatakan sebagai proses optimasi *query* (Azmi dan Kusumaningrum 2019). Cara kerja pada tahap *relevance feedback* ialah dengan mengubah *query* melalui umpan balik dari *user* ke mesin pencarian, ketika *user* memberikan tanggapan dari hasil pencarian yang dihasilkan oleh mesin. Dengan RF, *user* dapat memberikan *feedback* untuk mengganti *query* dengan yang lebih spesifik sehingga akan didapatkan hasil yang lebih relevan. Salah satu metode yang digunakan pada *relevance feedback* adalah metode Rocchio. Metode Rocchio *relevance feedback* merupakan metode yang sudah jamak digunakan dalam mengakomodasi *user* untuk mencari informasi dengan strategi reformulasi *query*, misalnya dalam penelitian similaritas pencarian dokumen tugas akhir menggunakan metode Rocchio (Dwiyantoro 2017). Dalam penelitian studi kelayakan pencarian buku, metode Rocchio telah diterapkan dalam desain sistem pencarian informasi berbasis *mobile*. Metode tersebut memanfaatkan *request* dari *user* untuk membantu proses pencarian dokumen relevan. Kesimpulan didapatkan setelah dilakukan kalkulasi derajat kemiripan antara *keyword* dan *term* pada judul. Hasilnya, metode Rocchio dapat menyelesaikan permasalahan *user* terkait *query* masukan dengan hasil yang didapatkan dari sistem pencarian informasi (Albert, Kristanda, dan Hansun 2016).

Peningkatan terhadap metode Rocchio dilakukan dalam melakukan kategorisasi teks (Gao dan Guan 2012). Caranya adalah dengan mengklasifikasikan segmen kata guna menemukan statistik jumlah kemunculan kata. Sebelum perhitungan TDIDF dilakukan, digunakanlah kamus *stop word* untuk mengurangi perhitungan TFIDF pada fungsi kata dan waktu komputasi. Akan tetapi, kumpulan data secara linier dapat mengakibatkan hilangnya hubungan antara kumpulan data yang serupa. Penelitian terkait kategorisasi teks disempurnakan dengan menggunakan pendekatan *hybrid* yang mengombinasikan metode Rocchio dan *random forest* (Thamarai Selvi *et al.* 2016). Metode Rocchio diterapkan dalam mengambil masukan *user* dan melakukan klasifikasi berdasarkan kesamaan kosinus. Untuk mengatasi keterbatasan dalam metode Rocchio, digunakanlah *stop remover* dan *word stemmer*. Metode *hybrid* yang diusulkan oleh peneliti akan menghasilkan efisiensi dalam proses kategorisasi dokumen. Keberhasilan metode Rocchio dalam melakukan klasifikasi teks telah dibuktikan dengan membandingkan performanya dengan KNN (Sowmya, Chetan, dan Srinivasa 2016). Hasilnya, metode Rocchio mendapatkan skor sebesar

0,28931, sedangkan metode KNN memperoleh skor sebesar 0,23088. Hal ini membuktikan bahwa metode Rocchio lebih akurat dan efisien dibandingkan dengan KNN.

Selain metode Rocchio, proses pencarian informasi juga dapat menggunakan algoritma genetika sebagai alternatif. Algoritma genetika ialah metode optimasi dengan melakukan seleksi genetik yang menghasilkan pemecahan masalah. Pada algoritma genetika, beberapa proses menggunakan sistem data yang diambil secara acak, namun dapat diatur tingkat pengacakan dan tingkat *control*-nya (Azmi dan Kusumaningrum 2019). Dengan performanya yang mampu mencari solusi yang optimal, algoritma genetika sudah diimplementasikan untuk menentukan jadwal dalam proses pembelajaran (Najib dan Grandis 2018). Dalam algoritma genetika, faktor penentu keberhasilan algoritma dalam mendapatkan solusi terbaik adalah *control*. Berdasarkan perhitungan nilai *fitness*, dapat ditetapkan derajat keberhasilan sebuah solusi. Algoritma genetika memiliki tujuan untuk mendapatkan solusi berdasarkan nilai *fitness* tertinggi.

Selain digunakan untuk penjadwalan mata pelajaran, algoritma genetika dapat juga dimanfaatkan untuk meningkatkan hasil dari *information retrieval*. Karena algoritma genetika telah menunjukkan hasil yang baik pada proses optimasi, metode ini dimasukkan sebagai bagian dari proses pencarian informasi (Azmi dan Kusumaningrum 2019). Algoritma genetika diterapkan dalam *information retrieval* untuk membentuk formula *query* baru melalui *relevance feedback* yang digunakan untuk pengambilan dokumen yang dianggap paling relevan. Penelitian Azmi dan Kusumaningrum (2019) telah mengimplementasikan algoritma genetika dengan menggunakan *roulette wheel selection* dan kombinasi *crossover* pada proses *relevance feedback*. Matriks evaluasi yang digunakan adalah *Mean Average Precision* (MAP) dan *recall* rata-rata berdasarkan penilaian *user*. Penelitian tersebut menunjukkan hasil peningkatan nilai MAP sebesar 15,2% dan 28,6% dengan menggunakan dua *dataset* dokumen bahasa Indonesia, yaitu tesis abstrak dan *dataset* berita. Dalam penelitian Lebib, Mellah, dan Drias (2017), penulis mengusulkan metode algoritma genetika untuk memilih sumber informasi melalui seleksi pencarian informasi. Penambahan *tag* pengguna memberikan deskripsi yang lebih akurat tentang sumber informasi. Penggunaan algoritma genetika dan penambahan *tag* pengguna dalam seleksi pencarian informasi telah menghasilkan efisiensi yang lebih baik dalam pencarian informasi terdistribusi.

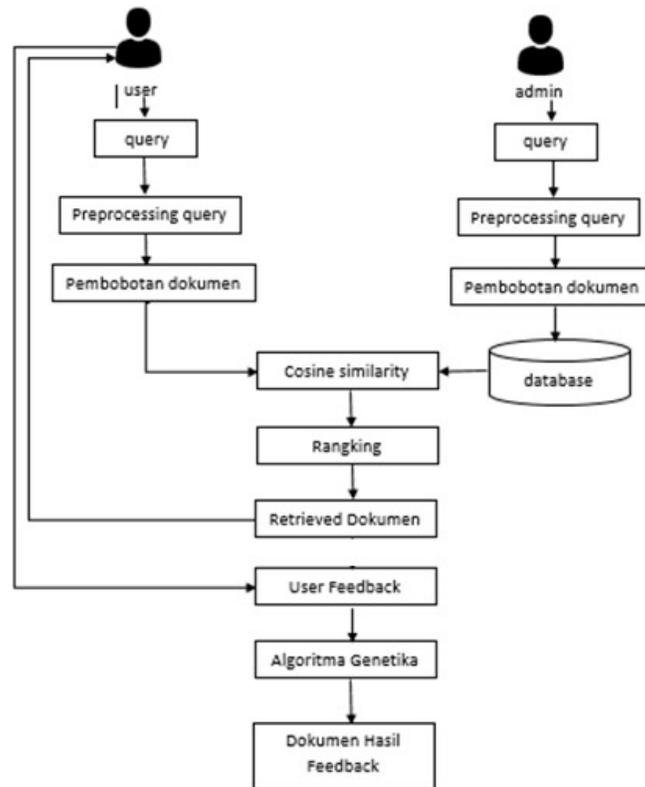
Penelitian yang dilakukan oleh Lilis Setiawati, Gani, dan Away (2019) mengenai *Travelling Salesman Problem* (TSP) bertujuan untuk mengetahui hasil dari penerapan metode *roulette wheel selection* dan *tournament selection* pada algoritma genetika. Hasil penerapan kedua metode seleksi tersebut menyatakan bahwa proses seleksi dengan metode *tournament* 44% lebih baik dibandingkan metode *roulette wheel*. Pada kasus lain, algoritma genetika bahkan dapat menyelesaikan tantangan pencarian yang terjadi pada informasi XML karena terdapat standar *de facto* di web (Bessai-Mechmache, Hammouche, dan Alimazighi 2020). Penelitian ini merumuskan permasalahan *retrieval* sebagai masalah optimasi kombinasi dalam menghasilkan *set* terbaik elemen XML yang relevan untuk *query* kunci yang diberikan. Algoritma genetika berperan untuk memaksimalkan kesamaan antara sekumpulan elemen XML dan *query* pengguna. Algoritma ini mampu memberikan hasil yang lebih tepat dibandingkan dengan model probabilitas.

Dengan demikian, untuk meningkatkan proses pencarian yang efektif dikembangkanlah mesin pencarian yang memanfaatkan *feedback* dari *user* dan menerapkan algoritma genetika pada proses *relevance feedback* guna memenuhi kebutuhan *user*. Selain itu, dilakukan pengukuran kemampuan dari *relevance feedback* dengan membandingkan *roulette wheel selection* dan *tournament selection* pada proses seleksi dalam algoritma genetika, guna melihat metode seleksi mana yang dapat menghasilkan pencarian yang handal. Pada *roulette wheel selection*, digunakan nilai probabilitas kumulatif dan *range* sebelum dilakukan proses *random*,

sedangkan pada *tournament selection* dilakukan proses *random* berulang. Terkait dengan hal tersebut, penelitian ini dilakukan dengan tujuan agar dapat menghasilkan sistem *relevance feedback* yang lebih akurat dalam pencarian kemiripan dokumen. Harapannya, sistem dapat melakukan interaksi dengan *user* untuk mendapatkan dokumen yang lebih relevan. *User* dalam hal ini adalah orang yang menggunakan atau mengakses mesin pencarian tanpa dibatasi oleh hak akses.

METODE

Information retrieval bertujuan untuk memenuhi kebutuhan *user* akan informasi yang sesuai. Sistem ini membutuhkan *query* dari *user* untuk mencari kemiripan dengan dokumen yang tersimpan sehingga mendapatkan hasil yang relevan (Agiyola dan Indriati 2019). Penelitian ini bertujuan untuk membangun mesin pencarian yang menghasilkan informasi sesuai dengan harapan *user*. Gambar 1 merupakan arsitektur *relevance feedback* pada *information retrieval* menggunakan algoritma genetika yang diusulkan.



Gambar 1. Arsitektur *Information Retrieval*

Pre-processing

Pre-processing merupakan proses pertama yang harus dilakukan dalam *information retrieval*. Tahap *pre-processing* digunakan untuk mengubah data mentah sebelum dilakukan

proses selanjutnya. *Pre-processing* mengubah isi dokumen menjadi kumpulan *terms*. Agiyola dan Indriati (2019) menjelaskan bahwa proses yang berlangsung pada *pre-processing*, yaitu:

1. *Case folding*
Case folding bertujuan untuk mengubah setiap isi dokumen dari huruf kapital menjadi huruf kecil semua. Pada tahap *case folding* juga dilakukan penghapusan karakter yang tidak penting dalam setiap isi dokumen.
2. *Tokenizing*
Tokenizing merupakan tahapan untuk memperoleh kata yang berasal dari dokumen.
3. *Stopword Removal*
Stopword removal merupakan tahap menghapus kata-kata tidak penting yang termasuk dalam *stopword list*. Kata-kata tersebut dihapus karena tidak mempunyai arti dan manfaat untuk proses berikutnya.
4. *Stemming*
Pada tahap ini dilakukan proses menghilangkan kata imbuhan yang melekat di kata dasar.

Pembobotan

Pada tahap ini diterapkan metode TF (*Term Frequency*) dan metode IDF (*Inversed Document Frequency*) dalam perhitungan bobot. Metode tersebut diterapkan dalam dokumen dengan menghitung bobot kata. Metode TF-IDF menggabungkan dua konsep untuk perhitungan bobot. TF menyatakan seberapa sering sebuah kata akan muncul dalam dokumen tertentu, sedangkan IDF adalah kebalikan dari TF. Seberapa sering sebuah kata muncul menunjukkan seberapa penting kata tersebut dalam dokumen. Dengan begitu, maka bobot kata akan semakin besar (Melita, 2018). Adapun IDF atau *Inversed Document Frequency* merupakan kebalikan frekuensi dari dokumen yang mengandung kata tersebut. Persamaan IDF adalah sebagai berikut:

$$IDF = \log \frac{N_{dok}}{DF_{dok}} \dots\dots\dots 1)$$

Keterangan:

N_{dok} = banyaknya dokumen

DF_{dok} = banyaknya dokumen berdasarkan kata yang dicari

Persamaan perhitungan bobot setiap dokumen dengan *query* dapat dilihat di bawah ini:

$$Wdt = tfdt \times idft \dots\dots\dots 2)$$

Keterangan:

Wdt = bobot dokumen ke-d pada kata ke-t

tf = jumlah kata yang dicari pada suatu dokumen

df = jumlah dokumen yang mengandung kata yang dicari

$idft = Inversed Document Frequency (\log(N/df))$

Perhitungan Kemiripan (*Cosine Similarity*)

Pada tahap ini dilakukan proses untuk mengukur kesamaan dengan menggunakan *cosine similarity*. *Cosine similarity* digunakan untuk menghitung tingkat kemiripan antara dua buah *vector* dalam ruang dimensi. Dua *vector* tersebut ialah *vector query* dan *vector* dokumen. *Cosine similarity* diperoleh dari perkalian dua buah *vector* yang akan dikatakan mirip apabila hasil perhitungan bernilai 1 (Aziz, 2017). Persamaan *cosine similarity* dapat dilihat pada Persamaan 3 berikut.

$$Similarity = \cos(\phi) = \frac{\sum_{i=1}^n Qi \cdot Di}{\sqrt{\sum_{i=1}^n (Qi)^2} \cdot \sqrt{\sum_{i=1}^n (Di)^2}} \dots\dots\dots 3)$$

Keterangan:

$\cos(\emptyset)$ = kemiripan Q terhadap dokumen D

Q = *query*

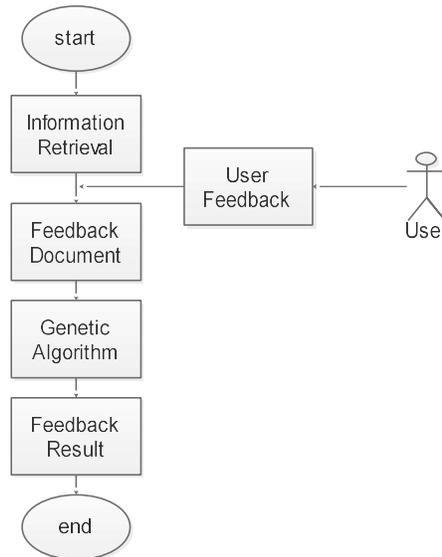
D = dokumen

n = jumlah data

Berdasarkan hasil *cosine*, dokumen akan melewati proses pemeringkatan. Dokumen akan diurutkan dengan menempatkan pembobotan terbaik di peringkat teratas. Setelah mendapatkan dokumen yang relevan, dokumen akan ditampilkan ke *user*. *User* akan memilih dokumen yang dianggap relevan menurut kebutuhannya. Dokumen tersebut nantinya akan menjadi *query* baru dan akan digunakan untuk mencari dokumen yang lebih relevan dengan menggunakan algoritma genetika.

Relevance Feedback Menggunakan Algoritma Genetika

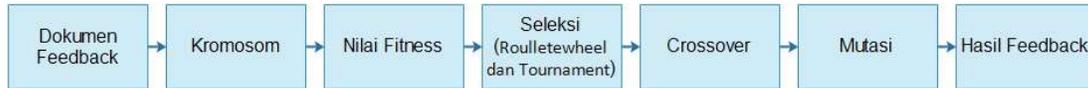
Relevance feedback merupakan hubungan antara *user* dan sistem yang bertujuan untuk meningkatkan *precision* dan *recall* yang sesuai dengan informasi dari dokumen yang relevan dan telah diidentifikasi oleh *user* (Agiyola dan Indriati, 2019). Pada tahap ini akan dilakukan perubahan pada *query* awal masukan *user* dan kemudian ditransformasi menjadi *query* baru sehingga mampu menampilkan hasil sesuai kebutuhan *user*. Modifikasi *query* pada penelitian ini menggunakan manual-*relevance feedback* dengan memanfaatkan *feedback* dari *user*. Pada manual-*relevance feedback*, algoritma genetika diterapkan sebagai perhitungan dokumen pada *relevance feedback*, yang menghitung kemiripan antara dokumen yang dijadikan *feedback* dengan dokumen yang tersedia. Diagram alur *relevance feedback*-algoritma genetika ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur *Relevance Feedback*-AG

Algoritma genetika menggunakan aturan probabilistik yang menghasilkan solusi perkiraan pada suatu masalah melalui seleksi genetik. Metode ini sering digunakan sebagai metode optimasi dalam memecahkan masalah. Ditambah lagi, algoritma genetika telah berhasil diterapkan pada berbagai bidang, baik dalam bidang teknik maupun bidang sains (McCall, 2005). Kemunculan algoritma genetika pada tahun 1975 diperkenalkan oleh John Holland. Algoritma genetika merupakan sebuah metode yang diimplementasikan untuk mencari solusi terbaik dalam pemecahan masalah dengan meniru proses evolusi kehidupan, yakni manipulasi kromosom pada

setiap individu. Pada tahap manipulasi, gen dari *parent* dicampur dan digabungkan kembali melalui *crossover* atau mutasi untuk menghasilkan keturunan di generasi berikutnya. Tahap demi tahap pada algoritma genetika ditunjukkan pada Gambar 3.



Gambar 3. Blok Diagram Algoritma Genetika

1. Kromosom

Kromosom merupakan suatu cara untuk mengubah suatu nilai tertentu, yang hasil pemberian nilainya merupakan representasi dari tiap variabel. Hal ini dilakukan untuk mewakili gen yang ada dari suatu populasi (Keikha, Ensan, dan Bagheri, 2018). Proses pemberian nilai kromosom dilakukan dengan mengganti setiap gen dengan nilai 1 atau 0. Populasi awal akan membentuk matriks bilangan biner yang terdiri dari baris dan kolom.

2. *Fitness*

Setelah menentukan kromosom pada tiap dokumen, tahap selanjutnya adalah menghitung *fitness* dari masing-masing kromosom tersebut. Nilai *fitness* digunakan untuk menentukan kualitas suatu kromosom atau individu. Semakin besar nilai *fitness* suatu kromosom berarti kromosom tersebut semakin baik, begitu pula sebaliknya. Rumus menghitung nilai *fitness* dapat dilihat pada Persamaan 4 berikut.

$$Fitness = \frac{dijk}{dij} \dots \dots \dots \dots \dots \dots 4)$$

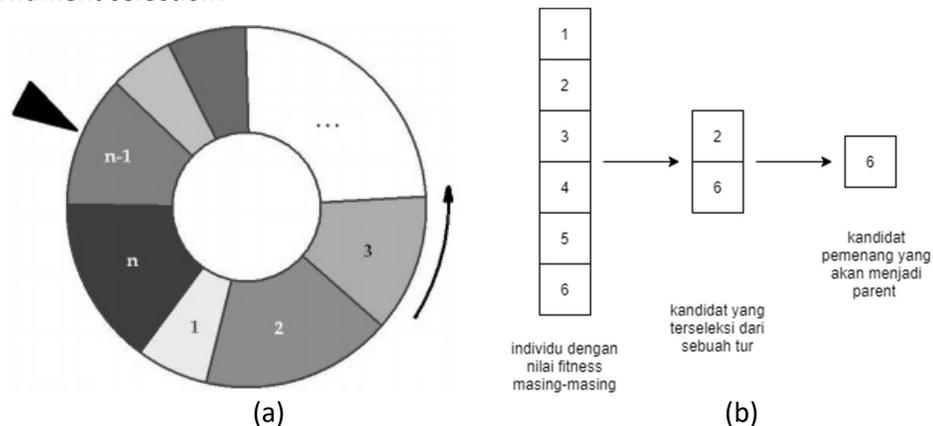
Keterangan:

dijk = gen dari kromosom populasi dengan nilai kromosom populasi 1 yang merepresentasikan kode kromosom populasi.

dij = gen dari kromosom solusi yang diwakili dengan nilai 1.

3. Seleksi

Berdasarkan perhitungan nilai *fitness* yang telah didapatkan, ditentukan induk (*parent*) dari kromosom yang terseleksi. Tujuan dari tahap ini ialah untuk memilih individu yang akan digunakan pada tahap *crossover*. Pada penelitian ini akan digunakan *roulette wheel selection* dan *tournament selection*.



Gambar 4. Ilustrasi (a) *Roulette Wheel Selection* (Sharapov, 2007), (b) *Tournament Selection*

Pada seleksi *roulette wheel*, *parent* dipilih berdasarkan nilai *fitness* pada tiap individu (Pencheva, Atanassov, dan Shannon, 2009; Ballera, Lukandu, dan Radwan, 2014) dengan cara mencari total *fitness* pada keseluruhan individu dan probabilitas masing-masing individu (Ho-Huu *et al.*, 2018; Zhang, Chang, dan Xu, 2012). Gambar 4 (a) merupakan ilustrasi cara kerja *roulette wheel*, yakni memutar sebuah roda dengan sebuah jarum penghenti. Roda yang berputar tersebut berisikan nilai-nilai yang mewakili individu yang ada. Individu dari kromosom dipilih dari *range* nilai yang ditunjukkan saat jarum penunjuk roda itu berhenti. Adapun cara kerja seleksi turnamen ialah dengan menghasilkan satu *set* kromosom awal secara acak dari populasi (Anton V. Eremeev, 2000; A. V. Eremeev, 2012). Langkah selanjutnya adalah memilih individu terbaik yang dilihat dari nilai *fitness* (Andrzej dan Stanislaw, 2000). Lakukan kembali proses tersebut sebanyak jumlah populasi yang diinginkan. Penjabaran lebih jelas terkait seleksi turnamen dapat dilihat pada Gambar 4 (b).

4. Crossover

Pada tahap ini akan dilakukan persilangan induk (*parent*), yang didapat dari proses seleksi dengan sesama induk. Proses ini akan menghasilkan kromosom turunan (*offspring*) berdasarkan elemen dari gabungan kromosom induk terpilih. Kromosom baru yang telah terbentuk diharapkan dapat memberikan solusi terbaik. Pada proses *crossover*, dibutuhkan nilai probabilitas untuk menentukan bagian yang akan dilakukan *crossover*. *Crossover* pada umumnya bernilai mendekati 1 (Janata dan Haerani, 2015). Umumnya, metode *one-point cut crossover* atau metode satu titik dipilih karena lebih sederhana dan mudah untuk diterapkan (Kom, 2016).

5. Mutasi

Mutasi adalah proses penggantian nilai suatu gen dalam suatu kromosom dengan nilai yang diperoleh melalui tahap *crossover*. Apabila gen terpilih bernilai 1, maka nilai gen tersebut selanjutnya diganti dengan 0 dan demikian sebaliknya. Mutasi akan menghasilkan individu yang baru dengan melakukan variasi suatu gen dalam individu yang sama.

Evaluasi

Tahap evaluasi digunakan untuk menghitung persentase penerapan algoritma yang diusulkan dengan mencari nilai *precision*, *recall*, *f-measure*, dan *run time*. Buckland dan Gey (1994) menjelaskan bahwa *recall* adalah perbandingan jumlah dokumen yang didapat dari suatu proses pencarian (Persamaan 5). Sementara itu, Goutte dan Gaussier (2005) menyatakan bahwa *precision* adalah perbandingan jumlah dokumen terkait yang diperoleh berdasarkan kebutuhan pengguna (Persamaan 6). *F-Measure* memberikan skor tunggal yang menyeimbangkan antara *precision* dan *recall* dalam satu angka (Persamaan 7) (Brief 2005). Adapun *run time* adalah waktu yang dibutuhkan oleh program untuk berjalan dengan menggunakan *roulette wheel selection* dan *tournament selection*.

Tabel 1. Matriks Akurasi

	Relevan	Tidak Relevan	Total
Ditemukan	a_{dok}	b_{dok}	$a_{dok} + b_{dok}$
Tidak ditemukan	c_{dok}	d_{dok}	$c_{dok} + d_{dok}$
Total	$a_{dok} + c_{dok}$	$b_{dok} + d_{dok}$	$a_{dok} + b_{dok} + c_{dok} + d_{dok}$

Keterangan:

a_{dok} = Dokumen relevan

b_{dok} = Dokumen yang tidak relevan

c_{dok} = Dokumen relevan yang tidak ditemukan

d_{dok} = Dokumen yang tidak relevan yang tidak ditemukan.

$$Precision = \frac{a_{dok}}{a_{dok} + b_{dok}} \times 100 \dots \dots \dots 5)$$

$$Recall = \frac{a_{dok}}{a_{dok} + c_{dok}} \times 100 \dots \dots \dots 6)$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \dots \dots \dots 7)$$

HASIL DAN PEMBAHASAN

Pada sesi ini akan dijabarkan hasil penelitian dari metode yang telah diusulkan. Penulis mengimplementasikan metode algoritma genetika pada proses *relevance feedback* untuk mendapatkan mesin pencarian yang relevan. Pada tahap seleksi algoritma genetika dilakukan 2 jenis seleksi, yakni *roulette wheel selection* dan *tournament selection*. Tujuan dari diterapkannya metode *roulette wheel selection* dan *tournament selection* dalam proses seleksi pada algoritma genetika adalah untuk membandingkan tingkat akurasi dan waktu komputasi dari kedua metode sehingga didapatkan hasil terbaik di antara keduanya. *Dataset* yang digunakan dalam penelitian ini adalah jurnal tugas akhir yang diunduh dari Portal Tugas Akhir Universitas Trunojoyo Madura. Data yang dipilih untuk digunakan sebagai uji coba hanya judul dan abstrak bahasa Indonesia sebanyak 1000 data.

Pengujian dilakukan menggunakan 6 *query* dengan panjang 1-3 kata. *Query* yang digunakan pada pengujian dapat dilihat pada Tabel 2. Pada penelitian ini, ditetapkan nilai probabilitas *crossover* (Pc) dan nilai probabilitas mutasi (Pm) pada *roulette wheel selection* dan *tournament selection* sebagai berikut, yakni nilai Pc = 0.5 dan Pm = 0.02, Pc = 0.5 dan Pm 0.2, Pc = 0.7 dan Pm = 0.02, Pc = 0.7 dan Pm = 0.2. Setelah *query* dimasukkan dan dilakukan *feedback*, sistem akan melakukan perhitungan persentase kelayakan dokumen dengan menggunakan rumus *precision*, *recall*, dan *F-measure* yang sebelumnya telah dijelaskan pada subbab evaluasi. Perhitungan persentase kelayakan dokumen digunakan sebagai parameter kualitas metode *relevance feedback* dengan algoritma genetika sesuai dengan *feedback* yang diberikan oleh *user*. Hasil pengujian untuk *query* dapat dilihat pada Tabel 3.

Tabel 2. Query Pengujian

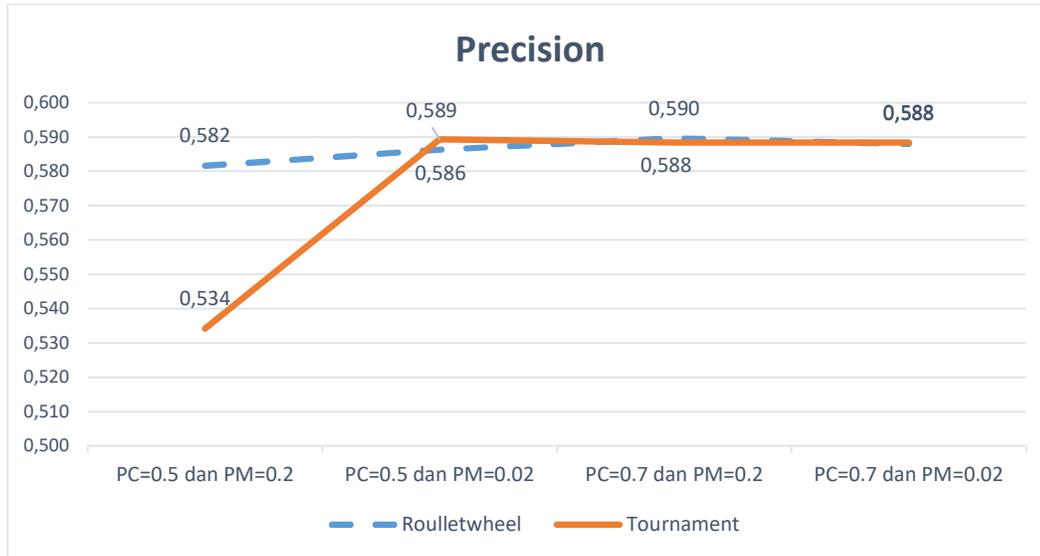
No	Query	Feedback
1	Citra	Sistem Perolehan Citra Berbasis Isi Menggunakan Gray Level Difference Method Berdasarkan Ciri Tekstur pada Pola Batik
2	Genetika	Sistem Pendukung Keputusan Optimasi Penjadwalan Mata Kuliah untuk Penggunaan Ruang Kuliah Bersama di Universitas Trunojoyo Madura Menggunakan Algoritma Genetika
3	Naïve Bayes	Sistem Penentuan Status Gizi Pasien Rawat Inap Menggunakan Metode Naïve Bayes Classifier (Studi Kasus: RSUD Dr. H. Slamet Martodirdjo Pamekasan)
4	Game Edukasi	Implementasi Game Edukasi untuk Media Pembelajaran Matematika Berbasis HTML5 pada Web Browser
5	Jaringan Syaraf tiruan	Sistem Peramalan Penjualan Menggunakan Metode Jaringan Syaraf Tiruan dengan Algoritma Backpropagation
6	Rancang Bangun Game	Rancang Bangun Game Perawatan Sapi Karapan Menggunakan Metode Fuzzy Logic

Tabel 3. Hasil Pengujian Dokumen Relevan yang Dihasilkan

query	Dokumen Relevan							
	Roulette wheel selection				Tournament Selection			
	Pc : 0.5, Pm : 0.02	Pc : 0.5, Pm : 0.2	Pc : 0.7, Pm : 0.02	Pc : 0.7, Pm : 0.2	Pc : 0.5, Pm : 0.02	Pc : 0.5, Pm : 0.2	Pc : 0.7, Pm : 0.02	Pc : 0.7, Pm : 0.2
Q1	76	76	77	77	77	77	77	77
Q2	5	5	5	5	6	4	6	5
Q3	17	18	18	18	18	19	19	19
Q4	19	19	20	20	20	20	20	20
Q5	9	8	9	9	9	9	9	9
Q6	14	14	14	14	14	14	14	14

Pada prinsipnya, sistem akan menampilkan dokumen yang dianggap mirip apabila hasil mutasi memiliki nilai yang sama dengan *fitness* pada populasi sebelumnya. Jika nilai tidak sama, maka tidak akan ditampilkan. Selanjutnya, dilakukan pemeringkatan terhadap dokumen yang dinilai mirip tersebut. Dokumen dengan nilai *fitness* paling besar akan ditampilkan di urutan teratas dan merupakan dokumen yang paling mirip dengan *feedback* yang dipilih oleh *user*. Proses penghitungan dokumen berpatokan pada *query* pertama yang dimasukkan oleh *user*. Berdasarkan jumlah kata pada *query*, semakin banyak kata pada *query* maka nilai dari *precision* akan semakin kecil. Hal tersebut disebabkan karena jauhnya nilai perbandingan antara dokumen yang ditemukan dan dokumen relevan yang didapatkan. Dokumen yang ditampilkan merupakan hasil dari *cosine similarity*, yang akan muncul jika dokumen mengandung salah satu kata dari *query*.

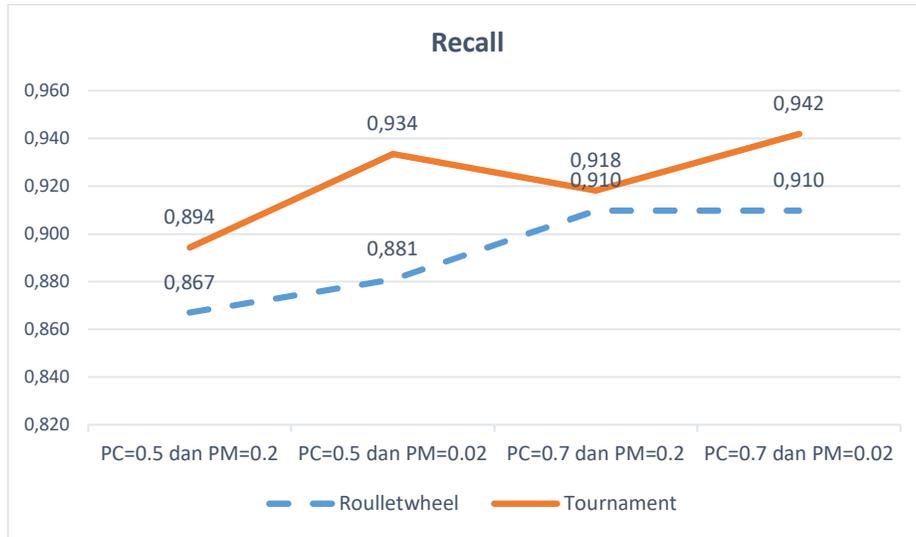
Berdasarkan skenario yang telah ditetapkan, hasil pengujian dengan beberapa probabilitas *crossover* dan mutasi dengan nilai rata-rata dari seluruh *query* dapat digambarkan dengan *trend* hasil percobaan. Gambar 5 menunjukkan *trend precision* yang telah diuji dengan menerapkan skenario uji coba. Nilai *precision* yang lebih baik didapatkan dengan menggunakan *roulette wheel selection* dibandingkan dengan menggunakan *tournament selection*. Ini menunjukkan bahwa sistem akan memberikan informasi yang lebih tepat sesuai permintaan *user* dengan menerapkan metode *roulette wheel selection*. Presisi merupakan perhitungan selisih antara jumlah dokumen yang diterima dan relevan dibagi dengan jumlah dokumen yang dikembalikan oleh sistem. Ini menandakan bahwa metode *roulette wheel selection* memiliki hasil terbaik dalam pengembalian hasil dokumen yang relevan kepada *user*. Pada *roulette wheel selection* diperoleh nilai *precision* tertinggi dengan menggunakan nilai Pc = 0.5 dan nilai Pm = 0.02.



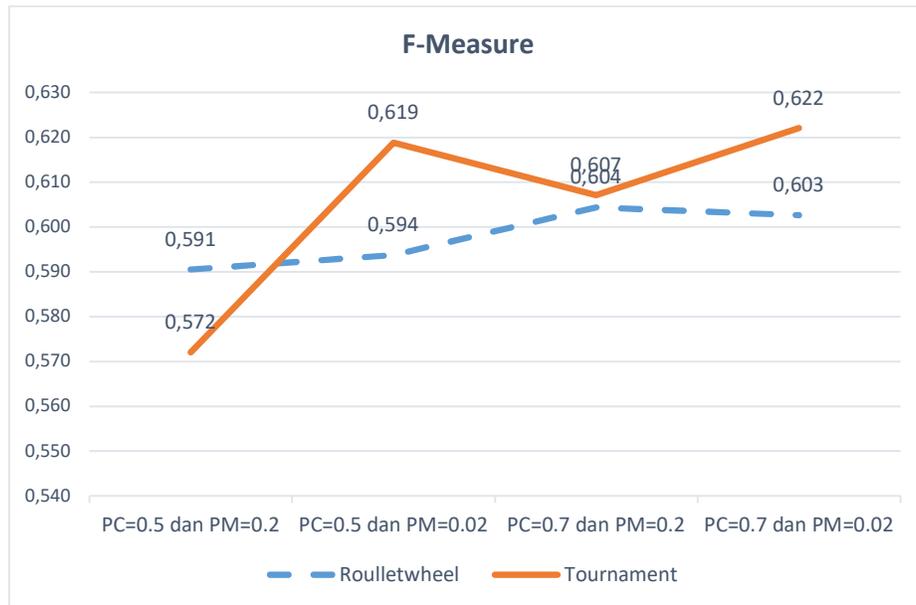
Gambar 5. Grafik *Trend Precision*

Gambar 6 menunjukkan hasil perhitungan nilai *recall*. Grafik menunjukkan bahwa nilai *recall* yang lebih baik akan didapatkan jika menggunakan *tournament selection* dibandingkan dengan menggunakan *roulette wheel selection*. Berdasarkan hal tersebut, keberhasilan sistem dalam menemukan kembali sebuah informasi secara akurat diperoleh dengan menggunakan metode *tournament selection*. *Recall* merupakan perhitungan selisih antara jumlah dokumen yang diterima dan relevan dibagi dengan keseluruhan dokumen yang relevan. Dalam hal ini, metode *tournament selection* mampu memberikan hasil terbaik dalam ketepatan sistem untuk mengembalikan dokumen. Pada *tournament selection* diperoleh nilai *recall* tertinggi dengan menggunakan nilai $P_c = 0.7$ dan nilai $P_m = 0.2$.

Hasil *trend* nilai *F-Measure* ditunjukkan pada Gambar 7. *F-measure* merupakan salah satu perhitungan evaluasi yang menggabungkan *recall rate* dan *precision rate* dalam *information retrieval*. Tingkat *recall* dan nilai presisi dalam sebuah kasus dapat memiliki bobot yang berbeda. Matriks yang menunjukkan hubungan antara tingkat *recall* dan tingkat presisi adalah matriks *F*, yang merupakan bobot rata-rata harmonik dari tingkat *recall* dan tingkat presisi. Berdasarkan nilai presisi dan *recall* yang telah diperoleh sebelumnya, perhitungan *F-measure* akan melihat perbandingan antara metode *roulette wheel selection* dan *tournament selection*. Hasilnya, nilai *F-Measure* yang lebih baik akan didapatkan jika menggunakan *tournament selection* dibandingkan dengan menggunakan *roulette wheel selection*. Pada *tournament selection* diperoleh nilai *F-Measure* tertinggi dengan menggunakan nilai $P_c = 0.7$ dan nilai $P_m = 0.02$. Rendahnya nilai *F-Measure* yang didapatkan disebabkan oleh perbandingan nilai yang sangat jauh antara dokumen yang ditemukan dan dokumen relevan yang didapat pada *query* dengan 3 kata.

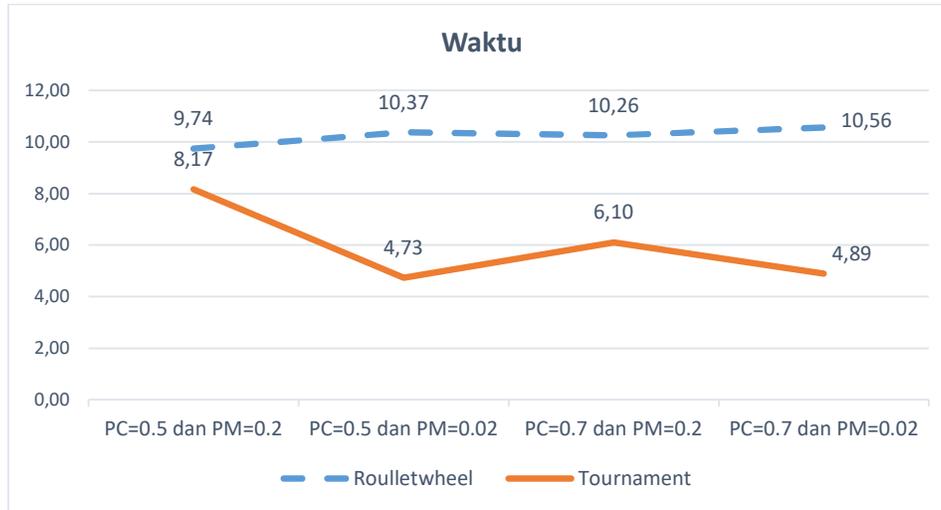


Gambar 6. Grafik Trend Recall



Gambar 7. Grafik Trend F-measure

Gambar 8 menunjukkan hasil pengujian waktu yang dibutuhkan untuk melakukan proses pencocokan pada tiap *selection* dalam algoritma genetika. Waktu tersebut dihitung dari saat *user* melakukan *feedback* sampai dokumen ditemukan. Hasil menunjukkan bahwa waktu yang dibutuhkan *tournament selection* lebih sedikit dibandingkan dengan *roulette wheel selection*. Waktu yang dibutuhkan oleh *tournament* tidak menunjukkan kestabilan berapapun probabilitas yang digunakan. Namun pada *roulette wheel selection*, waktu yang dibutuhkan cenderung stabil.



Gambar 8. Grafik *Trend Waktu*

. Mengacu pada *trend* percobaan yang dilakukan, metode *roulette wheel selection* menghasilkan nilai *precision* rata-rata sebesar 58%, *recall* rata-rata 89%, dan *F-Measure* rata-rata 59%. Adapun dengan menggunakan *tournament selection* didapatkan nilai *precision* rata-rata 57%, *recall* rata-rata 92%, dan *F-Measure* rata-rata 60%. Waktu rata-rata yang dibutuhkan pada *roulette wheel selection* adalah 10,23 detik, sedangkan pada *tournament* adalah 5,97 detik.

KESIMPULAN

Penelitian ini bertujuan untuk mengimplementasikan algoritma genetika pada proses *relevance feedback* agar mendapatkan hasil informasi yang sesuai kebutuhan *user* pada *information retrieval*. Untuk mendapatkan mesin yang handal, dilakukan perbandingan antara metode *roulette wheel selection* dan *tournament selection* pada proses seleksi dalam algoritma genetika. Berdasarkan jumlah kata pada *query*, semakin banyak jumlah kata pada *query* maka nilai *precision* akan semakin kecil. Hal tersebut disebabkan sangat jauhnya nilai perbandingan antara dokumen yang ditemukan dan dokumen relevan yang didapatkan. Dalam hal ini, dokumen yang ditampilkan merupakan hasil dari *cosine similarity* yang mengandung salah satu kata dari *query*.

Dengan demikian, tingkat ketepatan sistem dalam menemukan dokumen dengan informasi yang diminta *user* memiliki nilai yang lebih baik pada *roulette wheel selection*. Akan tetapi, tingkat keberhasilan sistem menemukan dokumen yang relevan memiliki nilai yang lebih baik dengan menggunakan *tournament selection*. Pada proses perhitungan waktu, *tournament selection* lebih unggul daripada *roulette wheel selection* dalam proses pencarian dokumen. Berdasarkan perhitungan *F-measure* dan komputasi waktu dapat disimpulkan bahwa penerapan algoritma genetika pada *relevance feedback* dalam proses pencarian dokumen mendapatkan hasil terbaik dengan menggunakan *tournament selection* pada proses seleksinya. Dari hasil pengujian yang telah dilakukan, sistem mampu menampilkan dokumen sesuai harapan dengan menggunakan nilai probabilitas pada masing-masing metode *selection* dan hal ini menghasilkan perbandingan antara kedua metode *selection* tersebut.

DAFTAR PUSTAKA

- Agiyola, Tasya, and Bayu Rahayudi Indriati. 2019. "Relevance Feedback Pada Sistem Temu Kembali Informasi Dokumen E-Book Berbahasa Indonesia Menggunakan Metode BM25." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* E-ISSN 2548: 964X.
- Albert, Albert, Marcel Bonar Kristanda, and Seng Hansun. 2016. "Tudi Kelayakan Dan Perancangan Aplikasi Pencarian Buku Pada Katalog Perpustakaan Menggunakan Rocchio Relevance Feedback." *Ultimatics: Jurnal Teknik Informatika* 8 (1): 37–43.
- Andrzej, O, and K Stanislaw. 2000. "A New Constraint Tournament Selection Method for Multicriteria Optimization Using Genetic Algorithm." In *Proceedings of the 2000 Congress on Evolutionary Computation*. CEC00 (Cat. No.00TH8512), 1:501–8 vol.1. <https://doi.org/10.1109/CEC.2000.870338>.
- Aziz, Muliadi. 2017. "Pemodelan Algoritma Genetika Pada Sistem Penjadwalan Perkuliahan Prodi Ilmu Komputer Universitas Lambungmangkurat." *KLIK-KUMPULAN JURNAL ILMU KOMPUTER* 1 (1): 67–79.
- Azmi, Salman Dziyaul, and Retno Kusumaningrum. 2019. "Relevance Feedback Using Genetic Algorithm on Information Retrieval for Indonesian Language Documents." *Journal of Information Systems Engineering and Business Intelligence* 5 (2): 171–82.
- Ballera, M, I A Lukandu, and A Radwan. 2014. "Personalizing E-Learning Curriculum Using: Reversed Roulette Wheel Selection Algorithm." In *2014 International Conference on Education Technologies and Computers (ICETC)*, 91–97. <https://doi.org/10.1109/ICETC.2014.6998908>.
- Bessai-Mechmache, Fatma Zohra, Karima Hammouche, and Zaia Alimazighi. 2020. "A Genetic Algorithm-Based XML Information Retrieval Model." *Proceedings - 2020 21st International Arab Conference on Information Technology, ACIT 2020*. <https://doi.org/10.1109/ACIT50332.2020.9300048>.
- Brief, Technical. 2005. "Agreement , the F-Measure , and Reliability in Information Retrieval," 296–98. <https://doi.org/10.1197/jamia.M1733.Informatics>.
- Buckland, Michael, and Fredric Gey. 1994. "The Relationship between Recall and Precision." *Journal of the American Society for Information Science* 45 (1): 12–19. [https://doi.org/https://doi.org/10.1002/\(SICI\)1097-4571\(199401\)45:1<12::AID-ASIS2>3.0.CO;2-L](https://doi.org/https://doi.org/10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASIS2>3.0.CO;2-L).
- Dwiyantoro, Dwiyantoro. 2017. "Sistem Temu Kembali Dengan Keyword (Deskriptif Menggunakan Recall Dan Precision Pada Judul, Subjek OPAC Perpustakaan Universitas Gadjah Mada)." *Khizanah Al-Hikmah: Jurnal Ilmu Perpustakaan, Informasi, Dan Kearsipan* 5 (2): 164–75.
- Eremeev, A V. 2012. "A Genetic Algorithm with Tournament Selection as a Local Search Method." *Journal of Applied and Industrial Mathematics* 6 (3): 286–94. <https://doi.org/10.1134/S1990478912030039>.
- Eremeev, Anton V. 2000. "Modeling and Analysis of Genetic Algorithm with Tournament Selection BT - Artificial Evolution." In , edited by Cyril Fonlupt, Jin-Kao Hao, Evelyne Lutten, Marc Schoenauer, and Edmund Ronald, 84–95. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Gao, Guanyu, and Shengxiao Guan. 2012. "Text Categorization Based on Improved Rocchio Algorithm." *2012 International Conference on Systems and Informatics, ICSAI 2012*, no. Icsai: 2247–50. <https://doi.org/10.1109/ICSAI.2012.6223499>.
- Goutte, Cyril, and Eric Gaussier. 2005. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation BT - Advances in Information Retrieval." In , edited by David E Losada and Juan M Fernández-Luna, 345–59. Berlin, Heidelberg: Springer Berlin Heidelberg.

- Halim, Agustino, Rudy Dwi Nyoto, and Novi Safriadi. 2017. "Perancangan Aplikasi Web Crawler Untuk Menghasilkan Dokumen Teks Pada Domain Tertentu." *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)* 5 (2): 114–17.
- Ho-Huu, V, T Nguyen-Thoi, T Truong-Khac, L Le-Anh, and T Vo-Duy. 2018. "An Improved Differential Evolution Based on Roulette Wheel Selection for Shape and Size Optimization of Truss Structures with Frequency Constraints." *Neural Computing and Applications* 29 (1): 167–85. <https://doi.org/10.1007/s00521-016-2426-1>.
- Janata, Ari, and Elin Haerani. 2015. "Sistem Penjadwalan Outsourcing Menggunakan Algoritma Genetika (Studi Kasus: PT. Syarikatama)." *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi* 1 (1): 17–24.
- Keikha, Andisheh, Faezeh Ensan, and Ebrahim Bagheri. 2018. "Query Expansion Using Pseudo Relevance Feedback on Wikipedia." *Journal of Intelligent Information Systems* 50 (3): 455–78.
- Kom, Eva Desiana M. 2016. "Performance Algoritma Genetika (GA) Pada Penjadwalan Mata Pelajaran." *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan* 1 (1): 56–60.
- Lebib, Fatma Zohra, Hakima Mellah, and Habiba Drias. 2017. "Enhancing Information Source Selection Using a Genetic Algorithm and Social Tagging." *International Journal of Information Management* 37 (6): 741–49. <https://doi.org/10.1016/j.ijinfomgt.2017.07.011>.
- Lilis Setiawati, Cut, Taufiq Abdul Gani, and Yuwaldi Away. 2019. "Studi Pengaruh Hibridisasi Seleksi Roulette Wheel Dengan Tournament Selection Menggunakan Algoritma Berevolusi Pada TSP." *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)* 2 (1): 26–30.
- McCall, John. 2005. "Genetic Algorithms for Modelling and Optimisation." *Journal of Computational and Applied Mathematics* 184 (1): 205–22.
- Melita, Ria. 2018. "Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)." *Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta*.
- Najib, Abdul, and Textianis Grandis. 2018. "Similaritas Dokumen Tugas Akhir Menggunakan Metode Rocchio." In *Prosiding SAKTI (Seminar Ilmu Komputer Dan Teknologi Informasi)*, 3:124–28.
- Pencheva, T, K Atanassov, and A Shannon. 2009. "Modelling of a Roulette Wheel Selection Operator in Genetic Algorithm Using Generalized Nets." *Bio Automation* 13 (4): 257–64.
- Sharapov, R R. 2007. "Genetic Algorithms: Basic Ideas, Variants and Analysis." In *Vision Systems*, edited by Goro Obinata and Ashish Dutta. Rijeka: IntechOpen. <https://doi.org/10.5772/4971>.
- Sowmya, B. J., Chetan, and K. G. Srinivasa. 2016. "Large Scale Multi-Label Text Classification of a Hierarchical Dataset Using Rocchio Algorithm." *2016 International Conference on Computation System and Information Technology for Sustainable Solutions, CSITSS 2016*, 291–96. <https://doi.org/10.1109/CSITSS.2016.7779373>.
- Thamarai Selvi, S, P Karthikeyan, A Vincent, V Abinaya, G Neeraja, and R Deepika. 2016. "Text Categorization Using Rocchio Algorithm and Random Forest Algorithm." In *2016 IEEE Eighth International Conference on Advanced Computing (ICoAC)*, 7–12. IEEE.
- Zhang, L, H Chang, and R Xu. 2012. "Equal-Width Partitioning Roulette Wheel Selection in Genetic Algorithm." In *2012 Conference on Technologies and Applications of Artificial Intelligence*, 62–67. <https://doi.org/10.1109/TAAI.2012.21>.