

# Algoritma Genetika untuk menentukan kemiripan antar dokumen dalam *Information Retrieval* menggunakan nilai Dice Coefficient

Marice Hotnauli Simbolon<sup>1</sup>, Sartana<sup>2</sup>, Maradu Sihombing<sup>3</sup>

<sup>1,2,3</sup>Universitas Mandiri Bina Prestasi

Jl. Letjend. Djamin Ginting No.285-287, Padang Bulan, Medan Baru, Medan, Sumatera Utara, Indonesia - 20155

<sup>1</sup>simbolonice@gmail.com, <sup>2</sup>sartanasinurat@gmail.com, <sup>3</sup>maradus44@gmail.com

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

## Abstrak

Fungsi utama dari sistem manajemen basis data adalah untuk menemukan kembali informasi atau memilih sebagian dari basis data yang memenuhi kriteria tertentu. Tempat penyimpanan koleksi dokumen (*database*) untuk keperluan informasi sudah menjadi lebih mudah, tetapi usaha untuk mendapatkan kembali informasi (dokumen) yang dibutuhkan dan sesuai menjadi sulit karena data yang tersimpan sangat besar, terbagi – bagi dan tidak tersusun. Secara fisik kumpulan dokumen dapat disimpan dalam bentuk disket, hard disk, dan CD-ROM. Secara teknis setiap dokumen yang telah dianalisis subjeknya dan diindeks, disimpan dalam suatu pangkalan data. Di sisi lain, kebutuhan informasi pengguna dianalisis dan direpresentasikan dalam suatu istilah pencarian (*search terms*). Selanjutnya, representasi dokumen dan pertanyaan pengguna (*query*) dipadukan dalam suatu proses pencocokan (*matching*) hingga menemukan dokumen yang relevan. Maka dilakukanlah teknik probabilitas dengan sistem berbasis kecerdasan buatan yaitu suatu pembelajaran mesin (*machine learning*) yang diterapkan dalam sistem pencarian dokumen yaitu *Information Retrieval System*. Hasil dari sistem ini berupa algoritma genetika yang dapat mensimulasikan proses evolusi yang berfungsi untuk menciptakan suatu kesatuan solusi (kromosom) untuk memecahkan masalah terutama dalam hal pencarian dokumen yang diperlukan tanpa harus mendapatkan dolumen – dokumen yang tidak relevan. Algoritma Genetika untuk menentukan kemiripan antar dokumen dalam *Information Retrieval System* menggunakan nilai *Dice Coefficient* digunakan untuk pemilihan beberapa kata kunci yang merupakan perwakilan dari keseluruhan dokumen untuk mencari dokumen lain yang mirip dengan melakukan penghematan dalam pencarian karena tidak digunakan seluruh kata kunci. Keunggulan sistem ini, walau begitu banyaknya dokumen yang tersimpan dalam database, pencarian akan mudah dan cepat dan menunjukkan persentasi kemiripan nilai dokumen.

*Kata Kunci:* Information Retrieval System, Dice Coefficient, Algoritma Genetika.

## 1. Pendahuluan

Fungsi utama dari sistem manajemen basis data adalah untuk menemukan kembali informasi atau memilih sebagian dari basis data yang memenuhi kriteria tertentu.

Ketersediaan alat untuk sistem temu kembali informasi seperti penyimpanan yang efektif dan murah telah menjanjikan pertumbuhan yang cepat. Tempat penyimpanan koleksi dokumen (*database*) untuk keperluan informasi sudah menjadi lebih mudah, tetapi usaha untuk mendapatkan kembali informasi (dokumen) yang relevan menjadi sulit karena *database* yang tersimpan sangat besar, terbagi – bagi dan tidak tersusun.

Untuk teks yang sangat penuh (banyak), perolehan kembali informasi kurang dari hasil memuaskan. Teknik *retrieval* dengan *probabilistic* telah digunakan untuk meningkatkan kinerja dari sistem

temu kembali informasi. Pendekatan yang didasarkan pada dua parameter utama, *probability* (peluang) dari dokumen yang relevan dan *probability* yang tidak relevan dari dokumen.

*Information Retrieval System* (sistem temu kembali informasi) menggunakan teknik-teknik probabilitas telah menarik perhatian yang cukup signifikan pada sebagian peneliti – peneliti dalam ilmu komputer dan informatika selama beberapa dekade. Pada tahun 1980-an teknik-teknik berbasis pengetahuan juga menghasilkan kontribusi yang mengesankan pengindeksan dan *information retrieval*. Para peneliti ilmu informatika telah berpaling ke teknik-teknik pembelajaran berbasis kecerdasan buatan baru yang lain mencakup jaringan saraf, pembelajaran simbolis, dan algoritma genetika. Pada teknik – teknik yang baru ini, berdasarkan pada paradigma yang banyak, telah menyediakan kesempatan-kesempatan yang besar kepada peneliti untuk meningkatkan kemampuan pemrosesan dan pengambilan informasi dari penyimpanan informasi.

Berdasarkan latar belakang yang telah dijelaskan di atas, penulisan akan menganalisa mengenai algoritma genetika menggunakan nilai Dice Coefficient dalam kemampuannya menentukan kemiripan antara dokumen yang dipilih dengan dokumen yang lainnya yang ada dalam *database* dan menganalisa apakah kemiripan dokumen sesuai dengan keinginan. Kemudian akan dilakukan perancangan sistem dalam hal pencarian dokumen yang mirip.

## 2. Landasan Teori

### 2.1. Algoritma Genetika

Algoritma Genetika adalah suatu algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Dalam hal ini populasi dari kromosom dihasilkan secara random dan memungkinkan untuk berkembang biak sesuai dengan hukum-hukum evolusi dengan harapan akan menghasilkan kromosom yang prima yang disebut dengan istilah generasi. Pada setiap generasi kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi fitness. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Jika nilai fitness semakin besar, maka solusi yang dihasilkan semakin baik. Generasi berikutnya dikenal dengan istilah anak (*offspring*) yang terbentuk dari dua gabungan kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi.

Algoritma genetika sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan. Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu:

1. Operator reproduksi.
2. Operator persilangan (*crossover*).
3. Operator mutasi.

### 2.2. Information Retrieval System

*Information Retrieval System* (Sistem Temu Kembali Informasi) adalah sebuah sistem yang memiliki kemampuan untuk memberikan informasi

sesuai kebutuhan informasi pengguna. Sumber informasi berasal dari data tekstual kumpulan dokumen dalam spesifikasi format dokumen yang beraneka ragam. Istilah sistem temu kembali informasi diperkenalkan pertama kali pada tahun 1952, dan mulai populer diteliti sejak tahun 1961. Saat itu disadari bahwa *Information Retrieval System* sangat dibutuhkan. Aktivitas *Information Retrieval System* tidak hanya terbatas pada bagaimana cara menyimpan dokumen, tetapi juga meliputi pemahaman tentang penempatan informasi yang telah dikatalog dan diindeks agar mudah ditemukan kembali.

Lancaster mendefinisikan *Information Retrieval System* adalah suatu proses pencarian dokumen dengan menggunakan istilah-istilah pencarian untuk mendefinisikan dokumen sesuai dengan subjek yang diinginkan. *Information Retrieval System* ini bertujuan untuk mendapatkan dokumen yang relevan bagi pengguna.

Pada intinya pada *Information Retrieval System* terdapat tiga komponen utama yang saling mempengaruhi, yaitu:

- 1) Kumpulan dokumen
- 2) Kebutuhan informasi pengguna
- 3) Proses pencocokan.

Secara fisik kumpulan informasi/dokumen dapat disimpan dalam bentuk disket, hardisk, dan CD-ROM. Secara teknis setiap dokumen yang telah dianalisis subjeknya dan diindeks, disimpan dalam suatu pangkalan data. Di sisi lain, kebutuhan informasi pengguna dianalisis dan direpresentasikan dalam suatu istilah pencarian (*search terms*). Selanjutnya, representasi dokumen dan pertanyaan pengguna (*query*) dipadukan dalam suatu proses pencocokan (*matching*) hingga menemukan dokumen yang relevan.

Bahasa pengindeksan dapat dibedakan menjadi tiga kategori, yaitu:

- 1) Bahasa terkendali (*controlled indexing language*),  
Dalam bahasa pengindeks-an terkendali, istilah-istilah yang menggambarkan informasi yang terkandung dalam dokumen sudah tersusun dalam suatu daftar bahasa pengindeksan terabjad seperti kamus;
- 2) Bahasa bebas (*free indexing language*),  
Bahasa pengindeks-an bebas adalah bahasa yang didasarkan pada penggunaan semua kata / istilah yang cocok;
- 3) Bahasa alami (*natural indexing language*).  
Bahasa pengindeks-an alami adalah bahasa pengindeksan yang didasarkan pada bahasa yang digunakan dalam dokumen, seperti istilah yang terdapat pada judul, abstrak, dan isi teks

lainnya. Secara umum, bahasa alami mempunyai kesamaan dengan bahasa bebas, sehingga banyak pakar yang menggabungkan keduanya.

### 2.3. Dice Coefficient

Teori keterhubungan sebagai jarak ukuran antar unit data  $j$  dan  $k$  yang pernah diusulkan oleh Lance dan Williams (1966) lalu dikembangkan oleh Czekanowski yang disebut sebagai *Dice Coefficient*, yang dinyatakan dengan rumus sebagai berikut :

$$S_{j,k} = \frac{2n(j \cap k)}{n(j) + n(k)}$$

Misalkan :

$$j = \{ a, b, c, d, e, f, g, h, i, j \}$$

$$k = \{ b, c, d, e, f \}$$

Maka :

$$(j \cap k) = \{ b, c, d, e, f \}$$

$$(j + k) = \{ a, b, c, d, e, f, g, h, i, j \}$$

$$\frac{2n(j \cap k)}{n(j) + n(k)} = \frac{10}{15} = 0.66$$

## 3. Metodologi Penelitian

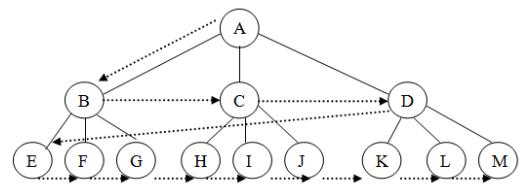
Hal terpenting dalam menentukan keberhasilan sistem berdasar kecerdasan adalah kesuksesan dalam pencarian dan pencocokan. Ada dua metode/teknik pencarian yaitu:

### 1) Pencarian Buta (*Blink Search*)

Pencarian Buta terdiri dari dua metode yaitu:

#### a. Pencarian Melebar Pertama (*Breadth-First Search*)

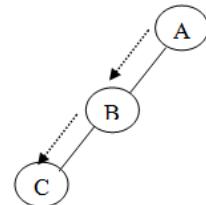
Pada metode *Breadth-First Search*, semua node pada level  $n$  akan dikunjungi terlebih dahulu sebelum mengunjungi node – node pada level  $n + 1$ . Pencarian dimulai dari node akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya kemudian dilakukan pencarian dari kiri ke kanan hingga ditemukannya solusi. Jika solusi ditemukan maka maka proses stop.



Gambar 1. Metode *Breadth-First Search*

### b. Pencarian Mendalam Pertama (*Dept-First Search*)

Dengan metode *Dept-First Search*, proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan dilakukan pencarian ke node-node yang selevel. Proses ini diulang sampai ditemukannya solusi.



Gambar 2. Metode *Dept-First Search*

### 2) Pencarian Heuristik (*Heuristic Search*)

Pencarian heuristik dibagi menjadi 4 (empat) metode yaitu:

#### 1. *Generate and Test*

*Generate and Test* (Pembangkit dan Pengujian)

Merupakan penggabungan antara *depth-first search* dengan pelacakan mundur, yaitu bergerak ke belakang menuju pada suatu keadaan awal. Nilai pengujian berupa jawaban ‘ya’ atau ‘tidak’.

#### 2. *Hill Climbing*

*Hill Climbing* (Pendakian Bukit Hill) hampir sama dengan metode pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada feedback dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan – keadaan lainnya

#### 3. *Best First Search*

*Best First Search* (Pencarian Terbaik Pertama) merupakan kombinasi dari metode *depth-first search* dan metode *breadth-first search* dengan mengambil kelebihan dari kedua metode tersebut. Pada metode best firs search, pencarian

diperbolehkan mengunjungi node yang ada pada level yang lebih rendah, jika ternyata pada node yang lebih tinggi ternyata memiliki nilai heuristik yang lebih buruk.

#### 4. Simulated Annealing

*Simulated Annealing* (Simulasi Annealing) terbentuk dari pemrosesan logam. Annealing (mamanaskan kemudian mendinginkan) dalam pemrosesan logam ini adalah suatu proses bagaimana membuat bentuk cair berangsur-angsur menjadi bentuk yang lebih padat seiring dengan penurunan temperatur. *Simulated annealing* biasanya digunakan untuk penyelesaian masalah yang mana perubahan keadaan dari suatu kondisi ke kondisi yang lainnya membutuhkan ruang yang sangat luas.

Adapun algoritma yang digunakan untuk menghasilkan dokumen yang mirip pada sistem yang dirancang ini adalah sebagai berikut:

Misalkan kita akan memilih dokumen sebanyak n, dengan panjang kromosom = L maka langkah-langkahnya :

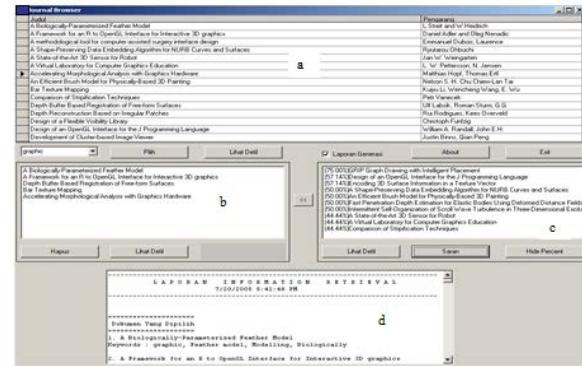
- 1) Bentuk model kromosom
  - 2) Bentuk populasi awal
  - 3) Tentukan nilai fitness pada setiap kromosom
  - 4) Seleksi kromosom untuk mendapatkan kandidat induk.
  - 5) Lakukan dengan metode roda roulette
  - 6) Proses crossover
    - Memilih kromosom yang akan melakukan crossover

Bangkitkan bilangan random antara [0 1] sebanyak n. Jika bilangan random  $\leq 0,25$ , kromosom akan melakukan crossover dan jika lebih besar tidak melakukan apa – apa.

    - Memilih posisi crossover
    - Ambil bilangan random antara [1 L-1] = x, lakukan crossover pada posisi x.
  - 7) Proses mutasi
  - 8) Lakukan evaluasi fitness
- Jika sudah mencapai solusi yang diharapkan maka proses selesai, jika tidak kembali ke langkah 4.

#### 4. Hasil dan Pembahasan

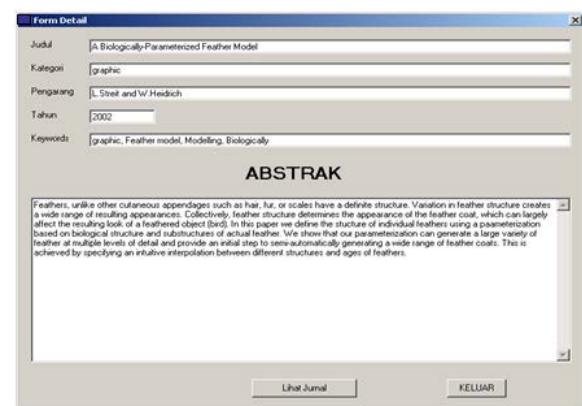
Pada penelitian ini akan dicoba mengimplementasikan Algoritma Genetika untuk menentukan kemiripan antar dokumen dalam *Information Retrieval System* menggunakan nilai *Dice Coefficient*



Gambar 3. Tampilan form utama

Keterangan gambar:

Sorot satu judul dokumen (a), kemudian klik tombol Pilih, maka judul dokumen akan berpindah ke b, ulangi sampai maksimal 10 kali Misalkan kita memilih 5 judul dokumen. Kemudian klik tombol Saran, maka judul dokumen dan persentase kemiripannya akan tampil (c). Klik pada check box Laporan Generasi, akan tampil laporan *information retrieval* (d) yaitu laporan proses algoritma genetika dari generasi ke generasi sampai menemukan solusi akhir. Klik tombol Lihat Detail, maka form detail akan tampil seperti gambar 4 di bawah ini.



Jika diklik tombol Lihat Jurnal maka isi jurnal akan ditampilkan

LAPORAN INFORMATION RETRIEVAL  
 7/22/2009 9:55:47 PM

Dokumen Yang Dipilih

1. A Biologically-Parameterized Feather Model  
 Keywords : graphic, Feather model, Modelling, Biologically
2. A Framework for an R to OpenGL Interface for Interactive 3D graphics  
 Keywords : graphic, OpenGL, Interface, 3D, C++
3. Depth Buffer Based Registration of Free-form Surfaces  
 Keywords : graphic, Buffer, Registration, Algorithm
4. Bar Texture Mapping  
 Keywords : graphic, Bar, mapping, Image based render, Acceleration, Hardware, Pixel Buffer, 3D, Texture
5. Accelerating Morphological Analysis with Graphics Hardware  
 Keywords : graphic, Hardware, Morphological, 3D

Himpunan Kata Kunci Keseluruhan

graphic, Feather model, Modelling, Biologically, OpenGL, Interface, 3D, C++, Buffer, Registration, Algorithm, Bar, mapping, Image based render, Acceleration, Hardware, Pixel Buffer, Texture, Morphological

Populasi Sesudah Persilangan :

Generasi ke-1

Populasi Awal :

11110000000000000000000000000000 [0.375213675213675]  
 10000000000000000000000000000000 [0.434920684920685]  
 10000000000000000000000000000000 [0.375213675213675]  
 10000000000000000000000000000000 [0.410989010989011]  
 10000000000000000000000000000000 [0.481196581196581]

Populasi Sesudah Seleksi :

11110000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000

Proses Persilangan :  
 Kromosom 4 dengan kromosom 3 , posisi 11  
 Kromosom 1 dengan kromosom 2 , posisi 12

Populasi Sesudah Persilangan :

11110000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000

Populasi Sesudah Seleksi :

10000000000000000000000000000000 [0.2716666666666667]  
 10000000000000000000000000000000 [0.35232785820211]  
 10010000000000000000000000000000 [0.446060606060606]  
 10000000000000000000000000000000 [0.3716666666666667]  
 10000000000000000000000000000000 [0.375213675213675]

Populasi Sesudah Seleksi :

11110000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10010000000000000000000000000000

Proses Mutasi :  
 Kromosom ke-5 posisi ke-4

Populasi Sesudah Mutasi :

11110000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10010000000000000000000000000000

Populasi Awal :

11110000000000000000000000000000 [0.375213675213675]  
 10000000000000000000000000000000 [0.375213675213675]

Proses Mutasi :

10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000  
 10000000000000000000000000000000

Generasi Akhir

Solusi : 11011000000000000000000000000000  
 Kata kunci : graphic, Feather model, Biologically, OpenGL, Bar, mapping, Acceleration, Hardware, Pixel Buffer, Texture

Pada penelitian ini akan dicoba mengimplementasikan Algoritma Genetika untuk menentukan kemiripan antar dokumen dalam sistem temu kembali informasi (*information retrieval system*) menggunakan nilai *Dice Coefficient* dengan beberapa langkah sebagai berikut:

#### 4.1. Membentuk Model Kromosom

Kromosom pada *information retrieval* adalah serangkaian gen ataupun bit kata kunci dari dokumen-dokumen yang dipilih. Jadi, langkah awal adalah menentukan jumlah macam kata kunci dari keseluruhan dokumen yang dipilih, juga menentukan posisi gen/bit dengan kata kunci pada rangkaian gen tersebut. Jika gen/bit bernilai 1, maka pada dokumen terdapat kata kunci, dan sebaliknya jika gen/bit

bernilai 0, maka pada dokumen tidak terdapat kata kunci tersebut.

Misalkan dipilih lima buah dokumen yaitu: Dokumen A, Dokumen B, Dokumen C, Dokumen D dan Dokumen E, dengan kata kunci sebagai berikut : Dokumen A : Graphic, Feather model, Modelling, Biologically Dokumen B : Graphic, OpenGL, Interface, 3D, C++ Dokumen C : Graphic, Buffer, Registration, Algorithm Dokumen D : Graphic, Bar, mapping, Image based render, Acceleration, Hardware, Pixel Buffer, 3D, Texture Dokumen E : Graphic, Hardware, Morphological, 3D

Didapatkan himpunan kata kunci dengan jumlah kata kunci sebanyak 19 buah sebagai berikut:

*Graphic, Feather model, Modelling, Biologically, OpenGL, Interface, 3D, C++, Buffer, Registration, Algorithm, Bar, mapping, Image based render, Acceleration, Hardware, Pixel Buffer, Texture, Morphological.*

Himpunan kata kunci inilah yang menjadi model kromosom dengan panjang kromosom 19 bit. Maka model kromosom dalam contoh ini adalah:

Kromosom A	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kromosom B	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Kromosom C	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Kromosom D	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	0
Kromosom E	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0

#### 4.2. Membentuk populasi awal

Populasi awal berdasarkan kata kunci – kata kunci dari dokumen yang dipilih. Jadi dengan menggunakan model kromosom yang didapatkan dalam langkah diatas maka populasi awal adalah sebagai berikut:

Kromosom A	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kromosom B	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Kromosom C	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Kromosom D	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	0
Kromosom E	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0

#### 4.3. Mencari Nilai Fitness dengan Rumus Dice Coefficient

Nilai fitness suatu kromosom adalah rata – rata nilai dice kromosom tersebut dengan setiap kromosom populasi awal.

Pada contoh di atas nilai fitness dari kromosom A adalah rata – rata nilai dice antara kromosom A dengan kromosom B, kromosom C, kromosom D, dan kromosom E, yaitu:

$$\text{kromosom A dengan kromosom A} \quad \frac{2 \times (A \cap A)}{A + A} = \frac{2 \times (4)}{8} = 1$$

$$\text{kromosom A dengan kromosom B} \quad \frac{2 \times (A \cap B)}{A + B} = \frac{2 \times (1)}{9} = 0,2222$$

$$\text{kromosom A dengan kromosom C} \quad \frac{2 \times (A \cap C)}{A + C} = \frac{2 \times (1)}{8} = 0,25$$

$$\text{kromosom A dengan kromosom D} : \quad \frac{2 \times (A \cap D)}{A + D} = \frac{2 \times (1)}{13} = 0,1538$$

$$\text{kromosom A dengan kromosom E} \quad \frac{2 \times (A \cap E)}{A + E} = \frac{2 \times (1)}{8} = 0,25$$

Total fitness dari kromosom A adalah : 1,8760

Nilai fitness dari kromosom A adalah :  $1,8760 / 5 = 0,5976$

Hal yang sama dilakukan pada kromosom lain, sehingga didapatkan nilai fitness kelima buah kromosom tersebut seperti yang terlihat pada tabel 1 di bawah ini :

Tabel 1 Nilai fitness dari tiap kromosom

Kromosom	A	B	C	D	E	Total	Fitness
A	1	0,2222	0,25	0,1538	0,25	1,8760	0,3752
B	0,2222	1	0,2222	0,2857	0,4444	2,1745	0,4349
C	0,25	0,2222	1	0,1538	0,25	1,8760	0,3752
D	0,1538	0,2857	0,1538	1	0,4615	2,0548	0,4109
E	0,25	0,4444	0,25	0,4615	1	2,4059	0,4811
Nilai fitness					2,0773		

#### 4.4. Seleksi Kromosom dengan Roda Roulette untuk Mendapatkan Kandidat Induk

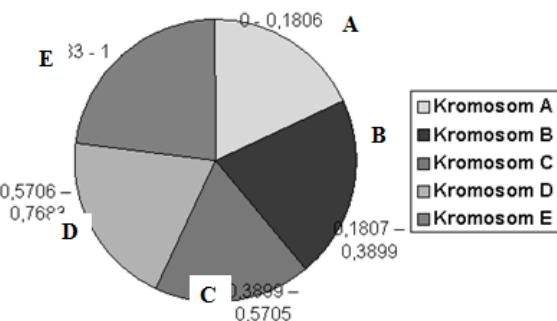
- Langkah – langkah yang harus dikerjakan adalah :
- Nilai fitness adalah : 2,0773
  - Sehingga nilai fitness relatif tiap – tiap kromosom dapat dicari sebagai berikut :  
 Untuk kromosom A = FA / Nilai fitness =  $0,3752 / 2,0773 = 0,1806$   
 Untuk kromosom B = FB / Nilai fitness =  $0,4349 / 2,0773 = 0,2093$   
 Untuk kromosom C = FC / Nilai fitness =  $0,3752 / 2,0773 = 0,1806$   
 Untuk kromosom D = FD / Nilai fitness =  $0,4109 / 2,0773 = 0,1978$   
 Untuk kromosom E = FE / Nilai fitness =  $0,4811 / 2,0773 = 0,2315$
  - Nilai fitness kumulatif tiap – tiap kromosom dapat dicari sebagai berikut:  
 Untuk kromosom A = **0,1806**  
 Untuk kromosom B =  $0,1806 + 0,2093 = 0,3899$

Untuk kromosom C =  $0,3899 + 0,1806 = 0,5705$

Untuk kromosom D =  $0,5705 + 0,1978 = 0,7683$

Untuk kromosom E =  $0,7683 + 0,2315 = 0,9998 (1)$

Dapat dipresentasikan pada diagram lingkaran pada gambar di bawah ini :



Gambar 3. Diagram lingkaran fitness

- Bangkitkan bilangan acak sebanyak 5 antara [0 1], dan bilangan acak yang dihasilkan adalah sebagai berikut:

- Bil 1 : 0,8381
- Bil 2 : 0,2537
- Bil 3 : 0,4168
- Bil 4 : 0,1725
- Bil 5 : 0,0921

Bil 1 berada di wilayah E

Bil 2 berada di wilayah B

Bil 3 berada di wilayah C

Bil 4 berada di wilayah A

Bil 5 berada di wilayah A

Maka kandidat induk pada generasi berikutnya adalah :

Kromosom A 100000100000001001 asal kromosom E

Kromosom B 1000111000000000000 asal kromosom B

Kromosom C 1000000111000000000 asal kromosom C

Kromosom D 1111000000000000000 asal kromosom A

Kromosom E 1111000000000000000 asal kromosom A

#### 4.5. Membentuk Crossover

Untuk memilih kromosom – kromosom mana saja yang akan dilakukan crossover, bangkitkan bilangan random antara [0 1] sebanyak 5 buah dan bilangan random yang dihasilkan adalah sebagai berikut :

Bil 1 = 0,1833

Bil 2 = 0,5517

Bil 3 = 0,1095

Bil 4 = 0,2453

Bil 5 = 0,0945

Kromosom dengan bilangan random yang lebih besar dari  $P_c$  (0,25) tidak dilakukan crossover, dan kromosom dengan nilai bilangan random yang lebih kecil dari  $P_c$  (0,25) akan melakukan crossover.

Kromosom yang akan disilangkan adalah kromosom A, kromosom C, kromosom D dan kromosom E.

Silangkan kromosom A dengan kromosom C dan kromosom D dengan E. Pada penyilangan kromosom A dengan kromosom C, ambil bilangan random antara 1 sampai 18 ( 19 – 1 ).

Dan nilai random yang diambil adalah 7, maka :

Lakukan crossover posisi 7:

Kromosom A 1000001000000001001

Kromosom C 1000000111000000000

Hasil crossover :

Kromosom A **1000001011000000000**

Kromosom C **10000000000000001001**

Crossover antara kromosom D dengan kromosom E:

Kromosom D 11110000000000000000000000000000

Kromosom E 11110000000000000000000000000000

Ambil nilai random antara 1 sampai 18.

Dan nilai random yang diambil adalah 13, lakukan crossover posisi 13:

Kromosom D 11110000000000000000000000000000

Kromosom E 11110000000000000000000000000000

Hasil crossover

Kromosom D 11110000000000000000000000000000

Kromosom E 11110000000000000000000000000000

Kromosom – kromosom setelah crossover:

Kromosom A 1000001011100000000

Kromosom B 10001111000000000000000000000000

Kromosom C 10000000000000001001

Kromosom D 11110000000000000000000000000000

Kromosom E 11110000000000000000000000000000

#### 4.6. Melakukan Mutasi

Untuk memilih bit-bit mana saja yang akan dilakukan mutasi, bangkitkan bilangan random antara [0 1] sebanyak 95 ( 19 x 5 ).

Dan bilangan random yang dihasilkan terlihat pada tabel 2.

Tabel 2. Bilangan acak pada proses mutasi

Bil	Kromosom				
	A	B	C	D	E
1	0,4925	0,3254	0,8876	0,1112	0,3128
2	0,2971	0,2683	0,4321	0,9651	0,0254
3	0,4877	0,1578	0,0532	0,2897	0,6519
4	0,1839	0,3322	0,7735	0,8862	0,4936
5	0,9333	0,4589	0,6354	0,1361	0,5681
6	0,4487	0,5254	0,4875	0,8267	0,3518
7	0,7543	0,4521	0,3680	0,6497	0,5549
8	0,2896	0,5332	0,9518	0,7612	0,5382
9	0,9354	0,0558	0,8547	0,6359	0,0941
10	0,1928	0,7641	0,4572	0,5237	0,2864
11	0,7889	0,0854	0,1354	0,4281	0,0038
12	0,8235	0,4433	0,1965	0,3249	0,8249
13	0,1063	0,1124	0,3654	0,5467	0,0548
14	0,4389	0,1257	0,1679	0,5138	0,9659
15	0,6325	0,9524	0,1234	0,3942	0,8618
16	0,2798	0,8504	0,5647	0,1672	0,2153
17	0,2223	0,6052	0,3256	0,6421	0,5557
18	0,3506	0,8596	0,5478	0,3254	0,6868
19	0,2567	0,0667	0,7854	0,4875	0,1246

Pada kromosom A bit pertama, bilangan random yang terbentuk adalah: 0,4925, bilangan ini lebih besar jika dibanding dengan  $p_m$  (0,01), ini berarti pada kromosom A bit pertama tidak terjadi mutasi.

Kromosom yang terkena mutasi adalah kromosom E posisi 4, karena 0,0036 lebih kecil dibanding dengan  $p_m$  (0,01).

Maka pada kromosom E bit 11 dilakukan mutasi.

Kromosom E 11110000000000000000000000000000

Kromosom E setelah mutasi 111100000010000000

Hasil setelah mutasi :

Kromosom A 1000001011100000000

Kromosom B 10001111000000000000000000000000

Kromosom C 10000000000000001001

Kromosom D 11110000000000000000000000000000

Kromosom E 11110000001000000000

Hasil setelah mutasi ini akan dijadikan sebagai populasi awal untuk generasi kedua :

Tabel 3. Populasi awal generasi kedua

Kromosom	Gen
A	1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0
B	1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
C	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
D	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
E	1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

#### 4.7. Evaluasi generasi berikutnya

Dengan cara yang sama langkah 3 sampai langkah 6 dilakukan pada generasi kedua sampai generasi keenam, algoritma ini akan berhenti jika pada nilai fitness tidak terdapat kemajuan, keseluruhan populasi

memiliki bentuk kromosom yang sama, jika tidak, algoritma ini akan berlanjut pada generasi berikutnya dan kembali ke langkah 3 dengan cara yang sama.

Pada contoh ini setelah berlanjut beberapa generasi akan didapatkan bentuk kromosom tunggal pada keseluruhan populasi yaitu:

**1 1 1 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1**

Maka hasil dari proses algoritma genetika keseluruhan dokumen yang dipilih, yaitu : graphic, Feather model, Modelling, openGL, Interface, 3D, Algorithm, Morphological.

Dari hasil tersebut didapatkan dokumen yang mirip yaitu :

[57.14%]GRIP Graph Drawing with Intelligent Placement  
[46.15%]A Shape-Preserving Data Embedding Algorithm for NURB Curves and Surfaces  
[46.15%]An Efficient Brush Model for Physically-Based 3D Painting  
[46.15%]Fast Penetration Depth Estimation for Elastic Bodies Using Deformed Distance Fields  
[37.50%]An Ontology-Driven Similarity Algorithm  
[37.50%]Real-time Terrain Rendering using Smooth Hardware Optimized Level of Detail  
[36.36%]Using Local Maxs Algorithm for the Extraction of Contiguous and Non-contiguous  
[35.29%]Rendering Techniques for Hardware-Accelerated Image-Based CSG  
[33.33%]Design of an OpenGL Interface for the J Programming Language  
[33.33%]Encoding 3D Surface Information in a Texture Vector

## 5. Kesimpulan dan Saran

Algoritma genetika dalam *information retrieval* digunakan untuk pemilihan beberapa kata kunci yang merupakan perwakilan dari keseluruhan dokumen untuk mencari dokumen lain yang mirip dengan melakukan penghematan dalam pencarian karena tidak digunakan seluruh kata kunci.

Keunggulan sistem ini, walau begitu banyaknya dokumen yang tersimpan dalam database, pencarian akan mudah dan cepat dan menunjukkan persentasi kemiripan nilai dokumen.

Kedepannya, penelitian mengenai algoritma dari langkah 4 sampai langkah 8 menjadi bahan penelitian lebih lanjut.

## Referensi

- [1] Azis, RI., 2017, Sistem Temu Kembali Informasi Abstrak Tugas Akhir Menggunakan Algoritma Genetika Dengan Faktor Dicesimilarity, Universitas Islam Negeri Sultan Syarif Kasim Riau, <https://repository.uin-suska.ac.id/18495/>
- [2] Budiharto dan Suhartono, 2014, Artificial Intelligence : Konsep dan Penerapannya, Andi Publisher , Yogyakarta
- [3] Cholissodin, 2016, Dasar-Dasar Algoritma Genetika, Universitas Brawijaya, Malang [http://imamcs.lecture.ub.ac.id/files/2016/09/3.-Dasar-Dasar-Algoritma-Genetika\\_ALEV\\_BCD\\_L1617\\_v4.03.pdf](http://imamcs.lecture.ub.ac.id/files/2016/09/3.-Dasar-Dasar-Algoritma-Genetika_ALEV_BCD_L1617_v4.03.pdf)
- [4] Fathansyah, 2012, Basis Data Cetakan Pertama (edisi revisi), Penerbit Informatika, Bandung.
- [5] Kadir, A, 2012, Algoritma dan Pemrograman, Penerbit Andi, Yogyakarta.
- [6] Prasetyo, 2019, Penentuan Pembimbing Tugas Akhir Mahasiswa Jurusan Teknik Informatika Universitas Surabaya Dengan Metode Dice Coefficient, TEKNIKA, Pusat PPKM Institut Informatika Indonesia, Surabaya [http://repository.ubaya.ac.id/35038/1/Jurnal\\_Teknik\\_Full\\_Vol %208\\_No%201.pdf](http://repository.ubaya.ac.id/35038/1/Jurnal_Teknik_Full_Vol %208_No%201.pdf)
- [7] Purwanto, 2008, Perancangan dan Analisis Algoritma, Graha Ilmu, Yogyakarta.
- [8] Sri Kusumadewi, 2003, Artificial Intelligence : Teknik dan Aplikasinya, Graha Ilmu, Yogyakarta.
- [9] Suharjito, 2021, Algoritma Genetika dengan Python, Binus Online Learning, Jakarta Barat. <https://onlinelearning.binus.ac.id/computer-science/post/algoritma-genetika-dengan-python>
- [10] Sukmadinata, NS, 2010, Metode Penelitian Pendidikan, Penerbit PT. Remaja Rosdakarya, Bandung.
- [11] Valacich, et al.,2015, Essentials of Systems Analysis and Design (6th ed.), Pearson Education Limited, England.