

## ANALISA OPTIMASI *QUERY DATABASE* DALAM PEMBELAJARAN BERBASIS WEB DENGAN METODE M2S *CROSSOVER* DAN *CHUNK CROSSOVER*

Eka Rahmadyani  
Program Studi Magister Teknik Informatika  
Fakultas Ilmu Komputer Dan Teknologi Informasi  
Universitas Sumatera Utara  
e-mail : ekaeetc@yahoo.co.id

### ABSTRAK

Agar performa web pembelajaran yang ada tetap stabil, maka diperlukan optimasi yang merupakan suatu langkah untuk mengoptimalkan waktu dan volume menjadi lebih efisien. Tujuan penelitian ini adalah untuk menganalisa optimasi *query database* dalam web pembelajaran dan untuk menganalisa perbandingan antara optimasi dengan metode M2S *Crossover* dan metode *CHUNK Crossover*, sehingga nantinya optimasi *query* yang terbaik antara dua metode tersebut dapat diimplementasikan pada web pembelajaran guna mengoptimalkan *query* pada *database* web pembelajaran tersebut. Hasil dari penelitian ini adalah metode terbaik untuk permasalahan optimasi *query database* web pembelajaran adalah metode M2S *crossover* dengan nilai *fitness* yang lebih tinggi dan waktu yang lebih cepat. Optimasi *query database* web pembelajaran adalah memindahkan tiap operasi *SELECT* sejauh mungkin ke bawah *query tree* selama diperbolehkan oleh attribute-attribute dalam kondisi *SELECT*, Optimasi *query database* dalam web pembelajaran dapat meningkatkan performa aplikasi dengan nilai optimum dan tidak membutuhkan banyak iterasi.

**Kata Kunci :** *Query Database, Web Pembelajaran, Metode M2S Crossover, Chunk Crossover*

### 1. Pendahuluan

Optimasi *query* dapat dilakukan dengan pendekatan algoritma genetik yaitu algoritma pencarian heuristic yang didasarkan atas mekanisme evolusi biologis. Algoritma Genetika ini merupakan sebuah metode untuk memisahkan satu populasi kromosom (terdiri dari bit-bit 1 dan 0) ke populasi baru dengan menggunakan “seleksi alam” dan operator genetik seperti *crossover*, *mutation*, *inversion*.

*Crossover* menukar bagian kecil dari dua kromosom, *mutation* mengganti secara acak nilai gen di beberapa lokasi pada kromosom, *inversion* membalikkan urutan beberapa gen yang berurutan dalam kromosom. Dasar teori inilah yang menjadi dasar kebanyakan program yang menggunakan algoritma genetika pada saat ini.

*Crossover* merupakan operator genetik yang digunakan untuk mengkombinasikan individu-individu yang terdapat di dalam populasi untuk menghasilkan individu pada generasi berikutnya. Untuk *crossover*, metode yang digunakan yaitu dimana kromosom-kromosom anak diperoleh dengan cara memilih sub-barisan suatu *tour* dari satu induk dengan tetap menjaga urutan dan posisi sejumlah relasi yang mungkin terhadap induk yang lain. Metode operator *crossover* yang diterapkan dan dapat digunakan untuk menyelesaikan masalah optimasi *query database* adalah M2S *crossover* dan *CHUNK crossover*. Penggunaan dua metode *crossover* ini diuji dan dianalisa hasilnya, untuk mengetahui metode *crossover* apa yang terbaik yang dapat digunakan untuk menyelesaikan masalah optimasi *query database*, dengan mencari nilai minimum.

### 2. Tinjauan Pustaka

#### 2.1. Aplikasi Pembelajaran Berbasis Web

Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu (Yuhfizar, 2009). Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya (Sutabri, 2012). Jadi aplikasi adalah kumpulan perintah-perintah dalam komputer yang dibuat untuk menyelesaikan tugas-tugas tertentu. Sebagai contoh, aplikasi *word processing*, adalah sebuah aplikasi yang di peruntukkan membuat dokumen tertulis, Aplikasi *web browser*, adalah aplikasi yang di peruntukkan untuk mencari sesuatu dan menampilkan halaman web.

#### 2.2. Metode Optimasi *Crossover* dan *CHUNK Crossover*

Metode *crossover two point* (M2S) mempunyai cara kerja yang sama dengan *one point*, perbedaannya terletak pada pemilihan titik *crossover*. Dalam metode ini dipilih secara acak dua titik *crossover*. Pada *one point* dan *two point*, letak titik *crossover* yang dipilih pada kedua kromosom orang tua harus sama. *K point* adalah metode *crossover* yang merupakan generalisasi dari *one point* dan *two point*. Sedangkan *cut and splice* (*CHUNK*) adalah suatu metode *crossover* dengan letak titik *crossover* pada kedua kromosom orang tua tidak harus sama. Akibatnya, dua kromosom anak yang dihasilkan bisa memiliki panjang yang berbeda.

Dalam penelitian ini menggunakan 2 metode operator *crossover* yang diterapkan pada operator *crossover* ini, yaitu :

1. Metode M2S (*Modified Two Swap*) Proses algoritmanya adalah sebagai berikut :

- a. Pilih secara acak 2 gen dalam X
- b. *Replace* dengan gen yang sesuai dari Y
- c. Pertahankan urutannya tetap seperti itu dari Y untuk menciptakan satu *offspring*.

Contoh crossover dengan metode M2S :  
Diberikan 2 induk kromosom, pilih secara acak label A dan D  
Hasil kromosom baru yang diperoleh dengan metode M2S adalah  
 $X = mA \ nC \ mB \ mD \ nF \ nE$   
 $Y = mB \ nC \ mD \ mF \ mE \ nA$

Offspring  $X = mD \ nC \ mB \ nA \ nF \ nE$  Kemudian, tukar peran antara X dan Y, menjadi  
 $Y = mB \ nC \ mD \ mF \ mE \ nA$   $X = mA \ nC \ mB \ mD \ nF \ nE$   
Offspring  $Y = mB \ nC \ mA \ mF \ mE \ mD$

## 2. Metode CHUNK

Proses algoritmanya adalah sebagai berikut :

- Bangkitkan secara acak CHUNK yang dipilih.
- Kemudian kromosom hasil dari X dicopy kedalam posisi yang sama dalam offspringnya.
- Hapus gen yang sesuai pada Y
- Gunakan gen Y untuk memenuhi sisa posisi pada offspring.

Contoh *crossover* dengan metode CHUNK :  
Diberikan 2 induk kromosom, pilih CHUNK secara acak urutan ke [3,5], kemudian proses.  
 $X = mA \ nC \ mB \ mD \ nF \ nE$   
 $Y = mB \ nC \ mD \ mF \ mE \ nA$

Hapus B dan D yang ada di Y, dan hasil kromosom baru yang diperoleh dengan metode CHUNK adalah:  
Offspring  $X = nC \ mD \ mB \ mE \ nF \ nA$  Kemudian, tukar peran antara X dan Y, menjadi  
 $Y = mB \ nC \ mD \ mF \ mE \ nA$

$X = mA \ nC \ mB \ mD \ nF \ nE$   
Hapus B dan D yang ada di X dan hasil kromosom baru yang diperoleh dengan metode CHUNK adalah:  
Offspring  $Y = mA \ nC \ mD \ mB \ mE \ nF$

## 3. Metode Penelitian

Menganalisa optimasi *query database* dalam web pembelajaran dengan perbandingan antara optimasi dengan metode M2S *Crossover* dan optimasi dengan metode CHUNK *Crossover* dengan tahapan-tahapan:

### 1. Representasi Kromosom

Elemen utama dalam pengoperasian algoritma genetik adalah kromosom. Pada optimasi query database ini diberikan sejumlah join dan relasi. Dari setiap tabel yang ada, untuk membuat query, field-fieldnya dapat direlasikan ke tabel mana saja. Dengan kata lain, setiap tabel dapat dijoinkan dengan tabel-tabel lainnya.

Contoh:

Dicari nilai a, b, c, d maka variabel a, b, c, d dijadikan sebagai gen-gen pembentuk chromosome. Batasan nilai variabel a adalah bilangan integer 0 sampai 30. Sedangkan batasan nilai variabel b, c, dan d adalah bilangan integer 0 sampai 10.

### 2. Menentukan populasi awal

Setelah ukuran populasi ditentukan, kemudian dilakukan inialisasi terhadap kromosom yang terdapat pada populasi tersebut dengan populasi awal dipilih secara acak.

Contoh:

Misalkan ditentukan jumlah populasi adalah 6, maka:

Chromosome[1] = [a;b;c;d] = [12;05;03;08]  
Chromosome[2] = [a;b;c;d] = [02;01;08;03]  
Chromosome[3] = [a;b;c;d] = [10;04;03;04]  
Chromosome[4] = [a;b;c;d] = [20;01;10;06]  
Chromosome[5] = [a;b;c;d] = [01;04;03;09]  
Chromosome[6] = [a;b;c;d] = [20;05;07;01]

### 3. Seleksi

Metode seleksi yang digunakan pada algoritma ini adalah metode Roulette Wheel (piringan rolet). Seleksi ini bertujuan untuk memberikan kesempatan seleksi yang lebih besar bagi anggota populasi yang memiliki fitness tinggi untuk melakukan seleksi. Permasalahan yang ingin diselesaikan adalah nilai variabel a, b, c, dan d yang memenuhi persamaan  $a+2b+3c+4d = 30$ , maka fungsi\_objektif yang dapat digunakan untuk mendapatkan solusi adalah:

$fungsi\_objektif(chromosome) = | (a+2b+3c+4d) - 30 |$

Hitung fungsi\_objektif dari chromosome yang telah dibangkitkan:

$fungsi\_objektif(chromosome[1]) = Abs((12 + 2*5 + 3*3 + 4*8) - 30)$

$= Abs((12 + 10 + 9 + 32) - 30)$

$= Abs(63 - 30)$

$= 33$

$fungsi\_objektif(chromosome[2]) = Abs((2 + 2*1 + 3*8 + 4*3) - 30)$

$= Abs((2 + 2 + 24 + 12) - 30)$

$= Abs(40 - 30)$

$= 10$

$fungsi\_objektif(chromosome[3]) = Abs((10 + 2*4 + 3*3 + 4*4) - 30)$

$= Abs((10 + 8 + 9 + 16) - 30)$

$= Abs(43 - 30)$

$= 13$

$fungsi\_objektif(chromosome[4]) = Abs((20 + 2*1 + 3*10 + 4*6) - 30)$

$= Abs((20 + 2 + 30 + 24) - 30)$

$= Abs(76 - 30)$

$= 46$

$fungsi\_objektif(chromosome[5]) = Abs((1 + 2*4 + 3*3 + 4*9) - 30)$

$= Abs((1 + 8 + 9 + 36) - 30)$

$= Abs(54 - 30)$

$= 24$

$fungsi\_objektif(chromosome[6]) = Abs((20 + 2*5 + 3*7 + 4*1) - 30)$

$$\begin{aligned}
 &= \text{Abs}((20 + 10 + 21 + 4) - 30) \\
 &= \text{Abs}(55 - 30) \\
 &= 25 \\
 &\text{Rata-rata dari fungsi objektif adalah:} \\
 &\text{rata-rata} = (33+10+13+46+24+25)/6 \\
 &= 151 / 6 \\
 &= 25.167
 \end{aligned}$$

Proses seleksi dilakukan dengan cara membuat chromosome yang mempunyai fungsi objektif kecil mempunyai kemungkinan terpilih yang besar atau mempunyai nilai probabilitas yang tinggi. Algoritma dari seleksi roda roulette wheel adalah sebagai berikut :

- Hitung total fitness (F) :  
 $\text{TotFitness} = \sum F_k; k=1,2,\dots,\text{popsize}$   
 Untuk itu dapat digunakan fungsi fitness =  $(1/(1+\text{fungsi\_objektif}))$ , fungsi\_objektif perlu ditambah 1 untuk menghindari kesalahan program yang diakibatkan pembagian oleh 0.

$$\begin{aligned}
 \text{fitness}[1] &= 1 / (\text{fungsi\_objektif}[1]+1) \\
 &= 1 / 34 \\
 &= 0.0294 \\
 \text{fitness}[2] &= 1 / (\text{fungsi\_objektif}[2]+1) \\
 &= 1 / 11 \\
 &= 0.0909 \\
 \text{fitness}[3] &= 1 / (\text{fungsi\_objektif}[3]+1) \\
 &= 1 / 14 \\
 &= 0.0714 \\
 \text{fitness}[4] &= 1 / (\text{fungsi\_objektif}[4]+1) \\
 &= 1 / 47 \\
 &= 0.0212 \\
 \text{fitness}[5] &= 1 / (\text{fungsi\_objektif}[5]+1) \\
 &= 1 / 25 \\
 &= 0.0400 \\
 \text{fitness}[6] &= 1 / (\text{fungsi\_objektif}[6]+1) \\
 &= 1 / 26 \\
 &= 0.0385 \\
 \text{total\_fitness} &= 0.0294 + 0.0909 + 0.0714 + 0.0212 \\
 &\quad + 0.04 + 0.0385 \\
 &= 0.2914
 \end{aligned}$$

- Hitung fitness relatif tiap individu :

$$\begin{aligned}
 p_k &= F_k / \text{TotFitness} \\
 \text{Rumus untuk mencari probabilitas: } P[i] &= \text{fitness}[i] / \text{total\_fitness}
 \end{aligned}$$

$$\begin{aligned}
 P[1] &= 0.0294 / 0.2914 \\
 &= 0.1009 \\
 P[2] &= 0.0909 / 0.2914 \\
 &= 0.3119 \\
 P[3] &= 0.0714 / 0.2914 \\
 &= 0.2450 \\
 P[4] &= 0.0212 / 0.2914 \\
 &= 0.0728 \\
 P[5] &= 0.04 / 0.2914 \\
 &= 0.1373 \\
 P[6] &= 0.0385 / 0.2914 \\
 &= 0.1321
 \end{aligned}$$

- c. Hitung fitness kumulatif :

- $q_1 = p_1$
- $q_k = q_{k-1} + p_k; k=2,3,\dots,\text{popsize}$

Dari probabilitas diatas dapat kita lihat kalau chromosome ke 2 yang mempunyai fitness paling besar maka chromosome tersebut mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari chromosome lainnya. Untuk proses seleksi kita gunakan roulette wheel, untuk itu kita harus mencari dahulu nilai kumulatif probabilitasnya:

$$\begin{aligned}
 C[1] &= 0.1009 \\
 C[2] &= 0.1009 + 0.3119 \\
 &= 0.4128 \\
 C[3] &= 0.1009 + 0.3119 + 0.2450 \\
 &= 0.6578 \\
 C[4] &= 0.1009 + 0.3119 + 0.2450 + \\
 &\quad 0.0728 \\
 &= 0.7306 \\
 C[5] &= 0.1009 + 0.3119 + 0.2450 + 0.0728 + \\
 &\quad 0.1373 \\
 &= 0.8679 \\
 C[6] &= 0.1009 + 0.3119 + 0.2450 + 0.0728 + \\
 &\quad 0.1373 + 0.1321 \\
 &= 1
 \end{aligned}$$

- d. Pilih induk yang akan menjadi kandidat untuk di-crossover dengan cara :

- Bangkitkan bilangan random r.
- Jika  $q_k \leq r$  dan  $q_{k+1} > r$ , maka pilih kromosom ke (k+1) sebagai kandidat induk. Jika  $R[k] < C[1]$  maka pilih chromosome 1 sebagai induk, selain itu pilih chromosome ke-k sebagai induk dengan syarat  $C[k-1] < R < C[k]$ . Kita putar roulette wheel sebanyak jumlah populasi yaitu 6 kali (bangkitkan bilangan acak R) dan pada tiap putaran, kita pilih satu chromosome untuk populasi baru. Misal:  
 $R[1] = 0.201$   
 $R[2] = 0.284$   
 $R[3] = 0.009$   
 $R[4] = 0.822$   
 $R[5] = 0.398$   
 $R[6] = 0.501$

Angka acak pertama  $R[1]$  adalah lebih besar dari  $C[1]$  dan lebih kecil daripada  $C[2]$  maka pilih chromosome[2] sebagai chromosome pada populasi baru, dari bilangan acak yang telah dibangkitkan diatas maka populasi chromosome baru hasil proses seleksi adalah:

$$\begin{aligned}
 \text{chromosome}[1] &= \text{chromosome}[2] \\
 \text{chromosome}[2] &= \text{chromosome}[2] \\
 \text{chromosome}[3] &= \text{chromosome}[1] \\
 \text{chromosome}[4] &= \text{chromosome}[5] \\
 \text{chromosome}[5] &= \text{chromosome}[2] \\
 \text{chromosome}[6] &= \text{chromosome}[3]
 \end{aligned}$$

Chromosome baru hasil proses seleksi:

$$\begin{aligned}
 \text{chromosome}[1] &= [02;01;08;03] \\
 \text{chromosome}[2] &= [02;01;08;03]
 \end{aligned}$$

chromosome[3] = [12;05;03;08]  
 chromosome[4] = [01;04;03;09]  
 chromosome[5] = [02;01;08;03]  
 chromosome[6] = [10;04;03;04]

4. Crossover  
Melakukan optimasi query dengan metode M2S crossover dan CHUNK crossover.
5. Mutasi  
Mutasi yang digunakan dalam penelitian ini adalah mutasi permutasi. Mutasi yang dapat dilakukan yaitu dengan memilih dua nilai gen dari kromosom dan kemudian nilai tersebut saling dipertukarkan. Penukaran untuk setiap nilai dilakukan secara acak dengan masing-masing nilai gen yang menyatakan relasi mempunyai peluang yang sama untuk terpilih sebagai nilai gen penukar.
6. Parameter Genetik  
Parameter genetik berguna dalam pengendalian operator-operator genetik. Beberapa parameter yang digunakan adalah : jumlah relasi, ukuran populasi, maksimum generasi, probabilitas crossover (Pc), dan probabilitas mutasi (Pm).

#### 4. Hasil Dan Pembahasan

##### 4.1. Hasil

Bahasa query dalam web pembelajaran adalah sebagai berikut :

```
SELECT * FROM `tb_admin` WHERE
hari=1
SELECT * FROM `tb_guru` WHERE
hari=1 SELECT * FROM
`tb_matapelajaran` WHERE hari=1
SELECT * FROM `tb_siswa` WHERE
hari=1 SELECT * FROM
`tb_jadwal` WHERE hari=1
SELECT * FROM `tb_porsi` WHERE
hari=1 SELECT * FROM `tb_waktu`
WHERE hari=1
```

Query tersebut meminta semua record dari basis data yang sedang digunakan. Optimasi query yang dilakukan dalam web pembelajaran ini adalah sebagai berikut:

Query 1: Untuk setiap pembelajaran yang diberikan pada hari senin, diambil id, jam, hari, nama guru, kelas, dan mata pelajaran.

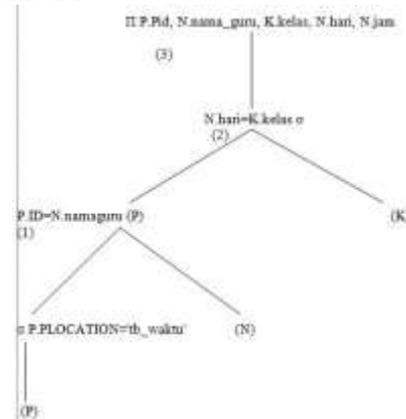
Maka dapat digambarkan :  
 $\Pi$  PId, jam, hari :  
 $(\sigma$ Plocation=tb\_waktu(PROJECT))  
 Nguru>Nama\_guru(tb\_guru)  
 Kkelas=kelas(tb\_kelas)

Persamaan ini mengikuti SQL query berikut:

```
SELECT P.id, N>Nama_guru, K.kelas,
N.Jam, N.hari FROM PROJECT AS P,
Nama_guru AS N, Kelas AS K WHERE
P.id=N>Nama-Guru AND
N.hari=K.kelas AND
P.Location='tb_waktu'
```

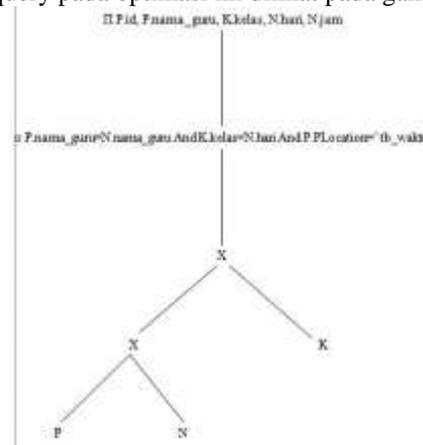
Maka Proses Parsing nya adalah sebagai berikut:

- a. Query tree yang sesuai dengan ekspresi relasi aljabar untuk Optimasi query dilihat pada gambar 1



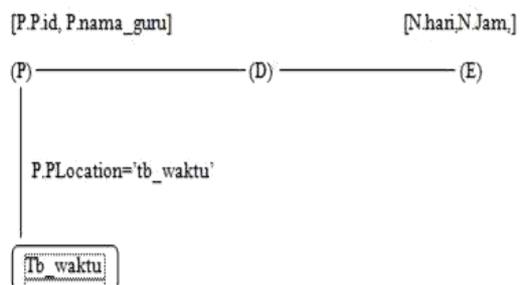
Gambar 1. Optimasi Query Tree

- b. Inisial ( Canonical) query tree untuk SQL query pada optimasi ini dilihat pada gambar 2



Gambar 2. Inisial Query Tree

- c. Query graph untuk optimasi query ini dapat dilihat gambar 3



Gambar 3. Query Graph

##### 4.2. Pembahasan

Sesuai rancangan penelitian yang penulis lakukan dengan menguji hasil optimasi *query database* dalam web pembelajaran dengan menganalisa perbandingan antara optimasi dengan metode M2S *Crossover* dan optimasi dengan metode CHUNK *Crossover* dengan parameter-parameter sebagai berikut:

- Ukuran populasi = 50
- Maksimum generasi = 100
- Probabilitas crossover (Pc) = 10,20,30,40,50
- Probabilitas mutasi (Pm) = 7

Dengan hasil pengujian sebagai berikut:  
Hasil pengujian lengkap perbandingan metode M2S crossover dengan CHUNKs crossover dijelaskan dalam tabel berikut:

**Tabel 1. Hasil Pengujian M2S Crossover**

| Probabilitas Crossover | Kromosom Yang Dibangkitkan | Maksimum Generasi | Probabilitas Mutasi | Fitness |
|------------------------|----------------------------|-------------------|---------------------|---------|
| 10                     | 50                         | 100               | 7                   | 1,570   |
| 20                     | 50                         | 100               | 7                   | 1,419   |
| 30                     | 50                         | 100               | 7                   | 1,668   |
| 40                     | 50                         | 100               | 7                   | 1,915   |
| 50                     | 50                         | 100               | 7                   | 2,082   |
| Fitness Rata-rata      |                            |                   |                     | 1,7308  |

**Tabel 2. Hasil Pengujian CHUNK Crossover**

| Probabilitas Crossover | Kromosom Yang Dibangkitkan | Maksimum Generasi | Probabilitas Mutasi | Fitness |
|------------------------|----------------------------|-------------------|---------------------|---------|
| 10                     | 50                         | 100               | 7                   | 1,530   |
| 20                     | 50                         | 100               | 7                   | 1,316   |
| 30                     | 50                         | 100               | 7                   | 1,915   |
| 40                     | 50                         | 100               | 7                   | 1,292   |
| 50                     | 50                         | 100               | 7                   | 1,575   |
| Fitness rata-rata      |                            |                   |                     | 1,5256  |

Berdasarkan hasil pengujian pada tabel diatas dapat dilihat bahwa nilai fitness pengujian dengan metode M2S cross over adalah lebih tinggi pada setiap probabilitas crossovernya yakni pada nilai probabilitas 10, 20, 30, 40 dan 50 dibanding dengan CHUNK crossover, dimana masing-masing kromosom yang dibangkitkan 50 dan maksimum generasi 100. Nilai fitness tertinggi pada metode m2s adalah pada probabilitas 50 dengan nilai fitness 2,082. Sedangkan pada metode CHUNK, nilai fitness tertinggi ada pada probabilitas 30 dengan nilai fitness 1,915. Nilai fitness rata-rata pada probabilitas 10,20,30,40 dan 50 untuk metode M2S adalah 1,7308. Sedangkan nilai fitness rata-rata untuk metode CHUNK cross over adalah 1,5256. Sehingga selisih fitness rata-rata metode M2S lebih baik senilai 0,2052 dibanding metode CHUNK Crossover.

**5. Kesimpulan**

Berdasarkan hasil penelitian dan analisa pembahasan yang peneliti lakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Optimasi query database web E-learning SMK Baramun adalah memindahkan tiap operasi SELECT sejauh mungkin ke bawah *query tree* selama diperbolehkan oleh attribute-attribute dalam kondisi SELECT
2. Optimasi *query database* dalam web pembelajaran dapat meningkatkan performa aplikasi dengan nilai optimum dan tidak membutuhkan banyak iterasi.
3. Metode terbaik untuk permasalahan optimasi query database web Pembelajaran pada percobaan yang dilakukan antara Metode M2S *crossover* dengan CHUNK *crossover* adalah metode M2S *crossover* dengan nilai fitness yang lebih tinggi dan waktu yang lebih cepat.

**6. Daftar Pustaka**

Abo-Hamad, Waleed & Arisha, Amr. Simulation–Optimisation Methods in Supply Chain Applications: A review. Irish journal management, pp97-124. 2011.

Coupey , Eloise. Marketing and the Internet: Conceptual foundation. Prentice-Hall, Upper Saddle River. New Jersey. 2001.

Fariza, A. Peramalan Time Series menggunakan Hybrid Algoritma Genetika Simulated Annealing. Hartanto, A.A. & Purbo, O.W. E-Learning Berbasis PHP dan MySQL, Elex Media Komputindo, Jakarta. 2003.

ISO 9003-3-1991 and ISO 8204. Quality Management and Quality Assurance Standards, Part 3:Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software. International Standards Organization.

Jananto, A. Meningkatkan Kecepatan Akses Data dengan Teknologi Optimalisasi Query Rushmore, Jurnal Teknologi Informasi DINAMIK Volume XI, No. 1, pp 47-56. 2006.

Koran, J.K.C. Aplikasi E-Learning dalam Pengajaran dan pembelajaran di Sekolah Malaysia, 2002.

Kusumadewi, S. Artificial Intelligence (Teknik dan Aplikasinya). Graha Ilmu. Yogyakarta. 2003.

Ladjamudin, Al-Bahra. Analisis dan Desain Sistem Informasi. Graha Ilmu. Yogyakarta. 2013.

Murniati. Penerapan Algoritma Genetik Pada Dna Sequencing By Hybridization. Skripsi. Universitas Indonesia. Jakarta. 2009.

Pandao, I. Multi Query Optimization Using Heuristic Approach. Internatio Journal of Computer Science and Network (IJCSN) Volume 1, Issue 4, August 2012 www.ijcsn.org ISSN 2277-5420.

Perangin-angin, K. Aplikasi Web dengan PHP dan MySQL. Andi : Yogyakarta. 2006.

Prasetyo, D.D. Aplikasi Database Client/Server Menggunakan Delphi dan MySQL. PT.Elex Media Komputindo. Jakarta 2004.

Saputro. A. Implementasi Algoritma Genetika Untuk Optimasi Penggunaan Lahan Pertanian.

DORO: Repository Jurnal Mahasiswa PTIIK  
Universitas Brawijaya, vol. 5, no. 12. 2015.

- Siallagan, M. Optimasi Query Database Menggunakan Algoritma Genetik. Jogjakarta : UII. 2008.
- Sutarman. Pengantar Teknologi Informasi. Jakarta: Bumi Aksara. 2012.
- Suyanto, A.H. Web Design Theory and Practices. Yogyakarta : Andi. 2010.
- Syaifudin, R. Implementasi Holap Untuk Optimasi Query Sistem Basis Data Terdistribusi Dengan Pendekatan Algoritma Genetik. Jurnal Sistem Informasi Bisnis 03. [http://ejournal.undip.ac.id/index.php/jsin\\_bis](http://ejournal.undip.ac.id/index.php/jsin_bis). 2014.
- Tata, S. Analisis Sistem Informasi. Andi. Yogyakarta. 2012.
- Wardiana, W. Perkembangan Teknologi Informasi di Indonesia. Makalah disampaikan pada seminar dan Pameran Teknologi Informasi, 9 Juli 2002, Universitas Komputer Indonesia, Jakarta. 2002.
- Yan-Fu, B. Wang, X. Shi,X-W. Yang, Y-V & Li, P. The Optimization Design Of Power Filter By Genetic Algorithm.National Key Laboratory Of Antennas And Microwave Technology Xidian. University. China 2009.
- Yuhefizar, Cara Mudah Membangun website interaktif menggunakan CMS Joomla edisi revisi. PT.Elex Media komputindi. Jakarta. 2009.
- Yunantara, I.G.S.A.M.D. Analisis dan Implementasi Penjadwalan dengan Menggunakan Pengembangan Model Crossover dalam Algoritma Genetika, Elektron. Ilmu Komputer., Vol. 1, No. 2, pp. 14–23, 2012.