



## Operator Kawin Silang pada Algoritma Genetik Riil untuk Variabel Rencana Selalu Positif

**Yoyong Arfiadi**

Program Studi Teknik Sipil, Universitas Atma Jaya Yogyakarta

Jl. Babarsari 44 Yogyakarta

E-mail: yoyong@mail.uajy.ac.id

### Abstract

*Genetic algorithms have been used to solve various optimization problems. One of the advantages of genetic algorithms is that they have the ability to solve complex optimization problems in a simple way. By using genetic algorithms, the near global optimum can be achieved easily. Although in the early development, binary coded genetic algorithms are more popular, recently real coded genetic algorithms are widely used to solve engineering problem's optimization. The advantage of using real coded genetic algorithms is the ability of the crossover operator to explore a larger domain of interest. As a result the use of crossover in real coded genetic algorithms may have a detrimental effect, as it can explore the domain that is very far from the expected domain. In the civil engineering area, most variables are positive. Therefore, it is needed to develop a crossover operator that can produce positive-only offspring. In this paper an asymmetric crossover is proposed to solve this problem. It is shown in the experiments that this crossover has a good performance in achieving optimum results.*

**Keywords:** *Real-coded genetic algorithms, Optimization, Crossover operator, Global optimum.*

### Abstrak

*Algoritma genetik telah banyak digunakan untuk menyelesaikan permasalahan optimasi. Salah satu kelebihan algoritma genetik adalah dapat menyelesaikan permasalahan optimasi yang kompleks dengan cara yang mudah. Dengan algoritma genetik, nilai yang relatif dekat dengan optimum global dapat dicapai dengan mudah. Walaupun pada awal perkembangannya algoritma genetik biner yang banyak digunakan, akhir-akhir ini algoritma genetik riil banyak digunakan untuk menyelesaikan masalah optimasi dalam bidang teknik. Kelebihan algoritma genetik riil dibandingkan dengan algoritma genetik biner adalah pada kemampuan operator kawin silang yang dapat mengeksplorasi kemungkinan domain yang lebih luas. Sebagai akibatnya, penggunaan operator kawin silang pada algoritma genetik riil dapat mengeksplorasi domain yang berada sangat jauh di luar domain yang diharapkan, sehingga algoritma menjadi kurang efisien. Dalam bidang teknik sipil, variabel yang akan dioptimasi sering hanya bernilai positif. Untuk itu perlu dikembangkan operator kawin silang yang tepat, agar turunan dari hasil kawin silang berada pada domain yang sesuai, misal domain variabel yang selalu bernilai positif. Dalam tulisan ini diajukan operator kawin silang baru sebagai operator kawin silang asimetrik. Dalam contoh-contoh yang disajikan, dapat ditunjukkan bahwa operator kawin silang yang diajukan menunjukkan kinerja yang baik dalam mencapai nilai optimum yang diinginkan.*

**Kata-kata Kunci:** *Algoritma genetik riil, Optimasi, Operator kawin silang, Optimum global.*

### Pendahuluan

Algoritma genetik (AG) telah banyak digunakan untuk menyelesaikan permasalahan optimasi yang sulit diselesaikan oleh metode lain. AG pertama kali diajukan oleh Holland (1992) yang didasarkan pada prinsip yang paling kuat yang bertahan hidup

*(survival of the fittest)*. Perbedaan algoritma genetik dengan metode optimasi lain adalah jika pada metode optimasi lain kandidat penyelesaian adalah tunggal, pada AG kandidat penyelesaian berupa suatu populasi. Hal ini menyebabkan seolah-olah AG mendekati permasalahan dari berbagai arah. Setiap kandidat dalam populasi

mengalami mutasi dan kawin silang dari generasi ke generasi. Dengan operator mutasi, setiap gen yang telah mengalami mutasi diharapkan menjadi lebih baik dibandingkan dengan gen awal. Demikian juga dengan kawin silang, individu-individu yang terpilih untuk kawin silang diharapkan membawa sifat-sifat baik dari orang tua mereka untuk menghasilkan individu-individu yang lebih baik. Ketahanan (*fitness*) suatu individu merupakan nilai fungsi obyektif yang ingin dioptimasi. Karena proses mutasi dan kawin silang dilakukan terus menerus sampai generasi yang diinginkan, ketahanan suatu individu lama kelamaan menjadi semakin tinggi, sehingga pada generasi terakhir diperoleh nilai optimum yang diinginkan. Karena dalam AG tidak diperlukan turunan (derivatif) dari fungsi yang akan diminimumkan, maka proses ini merupakan salah satu kelebihan yang lain dari AG dibandingkan dengan metode optimasi yang ada.

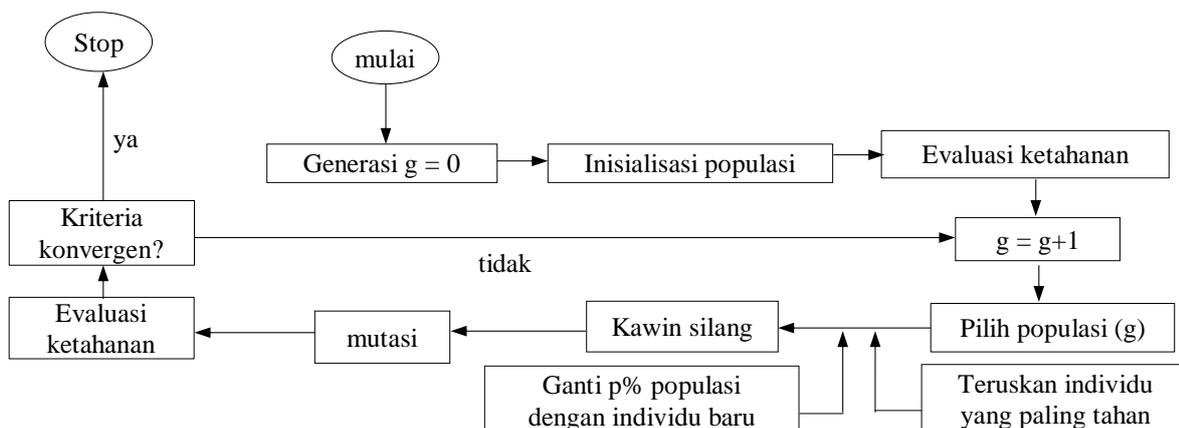
AG juga merupakan salah satu algoritma evolusioner (*evolutionary algorithm*) yang telah banyak digunakan untuk optimasi bidang teknik sipil. Salah satu tinjauan penggunaan AG, khususnya dalam bidang perencanaan struktur, dapat dilihat pada Kicinger dkk (2005). Di antara masalah optimasi yang menggunakan AG adalah tentang optimasi struktur baja (Rajeev dan Krishnamoorthy, 1992). Guerra dan Kiousis (2006) membahas optimasi struktur beton bertulang, sedangkan Alqedra dkk (2011) membahas optimasi balok beton bertulang dan prategang. Arfiadi dan Zebua (2015) melakukan optimasi beton prategangan statik tertentu dan statik tak tertentu. Aplikasi AG untuk mengoptimasi ukuran, bentuk, dan topologi struktur atap baja dibahas di antaranya oleh Frans dan Arfiadi (2014), dan Sesok dan Belevicius (2007). Dalam bidang kendali getaran AG telah digunakan untuk mengoptimasi gaya kontrol (Arfiadi dan Hadi,

2001), dan mengoptimasi sifat-sifat peredam (Arfiadi dan Hadi, 2011).

Dalam metode-metode terdahulu tidak ada perlakuan khusus untuk penyelesaian masalah sifat-sifat struktur yang selalu bernilai positif dalam operator mutasi dan kawin silang. Pada kebanyakan masalah teknik, variabel yang akan dioptimasi, seperti massa, kekakuan, redaman, jumlah tulangan, ukuran balok, selalu bernilai positif, sehingga penggunaan operator yang dapat mengakomodasi keadaan ini akan lebih cocok. Hal ini dapat lebih berguna terutama pada AG riil di mana operator AG dapat mengeksplorasi ruang domain yang terletak di luar domain awal (Arfiadi dan Hadi, 2001). Dalam tulisan ini dibahas operator kawin silang dalam algoritma genetik riil di mana operator ini akan selalu menghasilkan nilai variabel yang selalu positif, sehingga cocok digunakan pada optimasi masalah teknik.

### Langkah optimasi

Langkah-langkah optimasi pada AG dimulai dengan menginisiasi suatu populasi kandidat dari variabel desain secara random. Masing-masing kandidat dari variabel desain mempunyai nilai ketahanan (*fitness*) sendiri-sendiri sesuai dengan permasalahan optimasi yang sedang dianalisis. Beberapa individu dalam populasi terpilih untuk mutasi dan kawin silang untuk membentuk individu baru. Individu-individu baru ini membentuk populasi yang baru dengan harapan mempunyai nilai ketahanan yang lebih baik dari generasi sebelumnya. Proses ini dilakukan secara berulang sampai kriteria konvergen yang telah ditetapkan atau sampai generasi yang terakhir. Pada generasi terakhir, individu yang mempunyai ketahanan yang paling besar diambil sebagai nilai optimum dari variabel yang kita cari. Secara umum bagan alir langkah penyelesaian AG dapat dilihat pada Gambar 1.



Gambar 1. Langkah dalam algoritma genetik

Dalam Gambar 1, prosedur dalam AG standar telah dimodifikasi di mana dalam setiap generasi, sejumlah individu baru disisipkan untuk mengganti individu lama dalam populasi. Hal ini dimaksudkan untuk menambah variabilitas dalam suatu populasi (Arfiadi dan Hadi, 2001). Selain itu strategi elitis (Grefenstette, 1986) diberlakukan dengan selalu meneruskan individu yang paling tinggi ketahanannya pada generasi berikutnya.

**AG biner dan riil**

Dalam tahap awal perkembangannya, AG menggunakan bilangan biner yang terdiri dari kombinasi angka 0 dan 1. Kombinasi angka tersebut lalu dikonversi menjadi suatu integer, yang selanjutnya dikonversi menjadi bilangan riil. Dengan demikian panjang string dari kombinasi 0 dan 1 membatasi batas bawah dan batas atas dari suatu bilangan yang akan dicari.

Berbeda dengan AG biner, dalam AG riil, variabel desain dinyatakan dengan bilangan riil. Sebagai contoh untuk masalah optimasi dengan 3 desain variabel maka suatu individu dinyatakan dengan 20,1; 5,9; 102,3, yang mewakili suatu individu  $i$ :  $G_i = [g_1 \ g_2 \ g_3]$  Untuk jumlah populasi 5 dengan jumlah variabel desain 3 untuk suatu generasi tertentu dapat mempunyai data seperti terlihat pada Gambar 2. Dalam banyak hal, penggunaan AG riil lebih menguntungkan dibandingkan dengan AG biner.

Jumlah populasi

|      |       |       |
|------|-------|-------|
| 20,1 | 5,9   | 102,3 |
| 7,4  | 123,6 | 3,9   |
| 10,2 | 111,2 | 5,7   |
| 11,9 | 122,6 | 4,9   |
| 10,9 | 111,6 | 6,1   |

**Gambar 2. Representasi desain variabel dalam populasi dalam AG riil**

**Rekombinasi dalam AG riil**

Setiap individu dalam populasi mempunyai ketahanan sendiri-sendiri. Individu-individu kemudian dipilih untuk proses mutasi dan kawin silang. Dalam proses mutasi, individu yang terpilih mengalami perubahan struktur. Ada berbagai macam metode mutasi dalam AG riil. Salah satu metode adalah mutasi sederhana. Sebagai contoh untuk suatu individu tertentu yang mengalami mutasi sederhana dilakukan sebagai berikut ini. Sebelum mutasi:  $G_i = [g_1 \ g_2 \ g_3]$

Setelah mutasi:  $G'_i = [g'_1 \ g'_2 \ g'_3]$

dengan  $g'_2 = \bar{\alpha} a g_2$ ,  $\bar{\alpha} > 1$  dan  $a =$  variabel random antara 0 dan 1, jika variabel desain  $g_2$  terpilih untuk proses mutasi.

Pada proses kawin silang, terdapat beberapa metode kawin silang yang diajukan pada AG riil, yaitu kawin silang aritmetik dan kawin silang seimbang (Michalewicz, 1996). Pada kawin silang aritmetik, proses kawin silang diperoleh sebagai berikut ini. Jika dua individu  $\tilde{G}$  dan  $\bar{G}$  terpilih untuk kawin silang, maka hasil kawin silang menghasilkan turunan:

$$\tilde{G}' = a\tilde{G} + (1-a)\bar{G} \dots\dots\dots (1)$$

$$\bar{G}' = (1-a)\tilde{G} + a\bar{G} \dots\dots\dots (2)$$

dimana:

$a =$  variabel random antara 0 dan 1.

Pada kawin silang aritmetik, turunan yang dihasilkan akan selalu berada di antara batas bawah dan batas atas orang tuanya, seperti yang dapat dilihat pada Tabel 1 untuk  $a = 0,4$ . Hal ini kurang menguntungkan jika domain dari variabel yang akan dicari belum diketahui sebelumnya.

**Tabel 1. Kawin silang aritmetik**

| Sebelum kawin silang |       | Setelah kawin silang |        |
|----------------------|-------|----------------------|--------|
| $G_1$                | $G_2$ | $G_1'$               | $G_2'$ |
| 2                    | 10    | 6,8                  | 5,2    |
| 3                    | 10    | 7,2                  | 5,8    |
| 5                    | 10    | 8,0                  | 7,0    |
| 6                    | 10    | 8,4                  | 7,6    |
| 8                    | 10    | 9,2                  | 8,8    |
| 9                    | 10    | 9,6                  | 9,4    |
| 10                   | 10    | 10,0                 | 10,0   |
| 11                   | 10    | 10,4                 | 10,6   |
| 12                   | 10    | 10,8                 | 11,2   |

Metode kawin silang yang lain adalah kawin silang seimbang. Pada kawin silang seimbang, individu  $\tilde{G}$  dan  $\bar{G}$  yang terpilih untuk kawin silang akan menghasilkan turunan:

$$\tilde{G}' = a(\tilde{G} - \bar{G}) + \tilde{G} \dots\dots\dots (3)$$

$$\bar{G}' = a(\bar{G} - \tilde{G}) + \bar{G} \dots\dots\dots (4)$$

dimana:

$a =$  variabel random antara 0 dan 1.

Kelebihan kawin silang seimbang adalah bahwa proses ini akan menghasilkan individu yang dapat mengeksplorasi domain di luar domain awal. Kawin silang seimbang cocok digunakan jika domain

yang akan dicari tidak diketahui. Perlu dicatat pula bahwa untuk masalah teknik yang selalu bernilai positif, penggunaan kawin silang seimbang dapat menghasilkan individu yang bernilai negatif. Hasil kawin silang dengan cara kawin silang seimbang dapat dilihat pada Tabel 2 untuk  $a = 0,4$ .

**Tabel 2. Kawin silang seimbang**

| Sebelum kawin silang |       | Setelah kawin silang |        |
|----------------------|-------|----------------------|--------|
| $G_1$                | $G_2$ | $G_1'$               | $G_2'$ |
| 2                    | 10    | -1,2                 | 13,2   |
| 3                    | 10    | 0,2                  | 12,8   |
| 5                    | 10    | 3,0                  | 12,0   |
| 6                    | 10    | 4,4                  | 11,6   |
| 8                    | 10    | 7,2                  | 10,8   |
| 9                    | 10    | 8,6                  | 10,4   |
| 10                   | 10    | 10,0                 | 10,0   |
| 11                   | 10    | 11,4                 | 9,6    |
| 12                   | 10    | 12,8                 | 9,2    |

Memperhatikan kekurangan operator kawin silang yang ada, maka perlu diusahakan operator kawin silang untuk masalah dalam bidang teknik, yaitu untuk variabel desain yang selalu positif dan dengan domain yang belum diketahui. Dalam tulisan ini dicoba operator kawin silang asimetrik untuk mengatasi kekurangan operator yang ada.

**Operator kawin silang untuk variabel selalu positif**

Dalam bidang teknik, variabel desain dalam banyak kasus selalu bernilai positif. Sebagai contoh sifat-sifat struktur (seperti massa, redaman, dan kekakuan), ukuran struktur (seperti lebar dan tinggi), dan lain-lain. Dalam tulisan ini diajukan operator kawin silang asimetrik (Arfiadi, 2016). Dalam kawin silang asimetrik, individu  $G_1$  dan  $G_2$  yang terpilih untuk kawin silang menghasilkan:

$$G_1' = aG_1 + (1+a)^2G_2 \dots\dots\dots (5)$$

$$G_2' = a^2G_2 + aG_1 \dots\dots\dots (6)$$

dimana:  
 $a$  = variabel random antara 0 dan 1.

Dengan operator kawin silang asimetrik, maka untuk domain awal positif akan selalu menghasilkan turunan yang bernilai positif pula. Selain itu operator kawin silang asimetrik dapat mengeksplorasi domain di luar domain awal sehingga merupakan kelebihan dari operator kawin silang asimetrik. Pengaruh kawin silang asimetrik dapat dilihat pada Tabel 3.

Pada Tabel 3 terlihat bahwa operator kawin silang asimetrik selalu positif untuk orang tua yang bernilai positif. Selain itu operator kawin silang

asimetrik mempunyai kemampuan untuk mengeksplorasi domain yang lebih luas dari pada domain semula.

**Tabel 3. Kawin silang asimetrik**

| Sebelum kawin silang |       | Setelah kawin silang |        |
|----------------------|-------|----------------------|--------|
| $G_1$                | $G_2$ | $G_1'$               | $G_2'$ |
| 2                    | 10    | 20,4                 | 2,4    |
| 3                    | 10    | 20,8                 | 2,8    |
| 5                    | 10    | 21,6                 | 3,6    |
| 6                    | 10    | 22,0                 | 4,0    |
| 8                    | 10    | 22,8                 | 4,8    |
| 9                    | 10    | 23,2                 | 5,2    |
| 10                   | 10    | 23,6                 | 5,6    |
| 11                   | 10    | 24,0                 | 6,0    |
| 12                   | 10    | 24,4                 | 6,4    |

**Aplikasi**

**Eksperimen 1**

Sebagai eksperimen pertama ditinjau suatu fungsi sebagai berikut (Michalewicz, 1996):

$$f(x_1, x_2) = 4x_1 + 3x_2$$

yang akan dicari nilai maksimumnya, dengan kendala:

$$2x_1 + 3x_2 \leq 6$$

$$-3x_1 + 2x_2 \leq 3$$

$$2x_1 + x_2 \leq 4$$

$$0 \leq x_i \leq 6, i = 1, 2$$

Nilai optimum fungsi adalah  $f(1,5;1,0) = 9$ .

Selanjutnya GA riil dengan strategi elitis seperti yang telah ditunjukkan pada Gambar 1 digunakan untuk mengoptimasi fungsi tersebut. Kawin silang asimetrik digunakan dalam hal ini. Empat kali run dilakukan dengan:

- jumlah populasi = 30;
- generasi maksimum = 1000;
- probabilitas mutasi = 0,4;
- probabilitas kawin silang = 0,8;
- persentase populasi yang diganti per generasi = 0,1.

Dalam setiap run, batas atas dan batas bawah variabel desain diambil berbeda. Untuk mengakomodasi kendala, maka setiap individu yang melanggar kendala diberi nilai ketahanan yang sangat kecil yang dapat diterima oleh komputer. Hasil nilai ketahanan maksimum AG riil dengan kawin silang asimetrik dapat dilihat pada Gambar 3.

Dari empat kali *run* diperoleh nilai maksimum fungsi berdasarkan GA adalah: 8,9826; 8,9937; 8,9713; dan 8,963. Hasil terbaik diperoleh untuk  $x_1 = 1,4970$ ;  $x_2 = 1,0019$ , dengan nilai optimum = 8,9937. Hasil lengkap optimasi dapat dilihat pada Tabel 4.

Tampak bahwa walaupun dengan batas atas dan batas bawah yang berbeda dalam setiap *run*, menghasilkan variabel desain yang hampir sama.

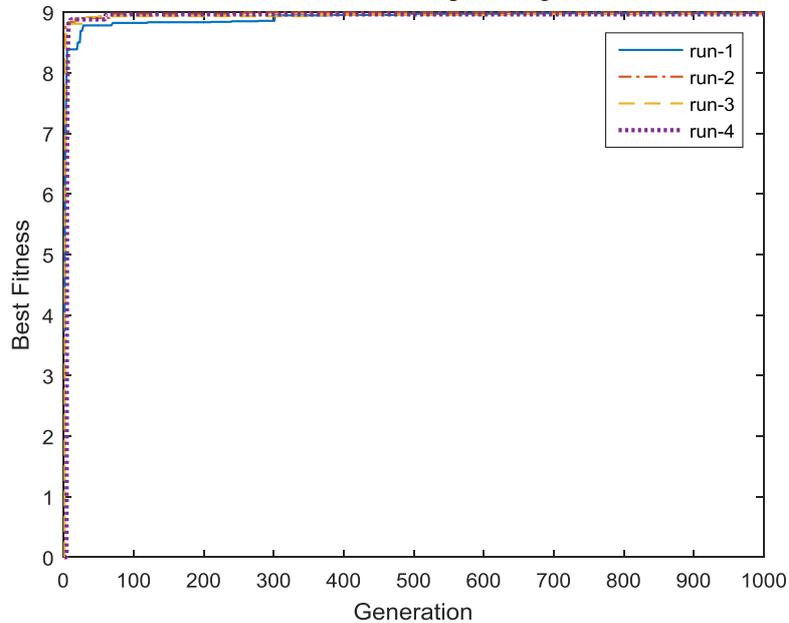
**Eksperimen 2**

Eksperimen kedua adalah meminimumkan fungsi Beale seperti terlihat pada Gambar 4 (Wikipedia.org):

$$f(x, y) = (1,5 - x + xy)^2 + (2,25 - x + xy^2)^2 + (2,625 - x + xy^3)^2$$

Karena AG selalu memaksimalkan fungsi maka fungsi obyektif untuk masalah ini diambil sebagai  $100/f(x, y)$ . Nilai minimum fungsi adalah= 0 untuk  $x = 3$  dan  $y = 0,5$ , yaitu  $f(3, 0,5) = 0$ . AG yang sama digunakan untuk mendapatkan nilai minimum fungsi ini. Dari 4 kali *run* diperoleh nilai  $100/f(x, y)$  berturut-turut 5,957e+06, 1,0067e+07, 9,7686e+06, dan 1,2537e+08. Hasil terbaik diperoleh untuk  $x = 3$  dan  $y = 0,50018$ . Nilai ketahanan digambarkan pada Gambar 5. Sedangkan nilai  $x$  dan  $y$  yang diperoleh ditunjukkan pada Tabel 5.

Tampak bahwa walaupun dengan batas atas dan batas bawah awal yang berbeda dalam setiap *run*, menghasilkan variabel desain yang mendekati optimum global.



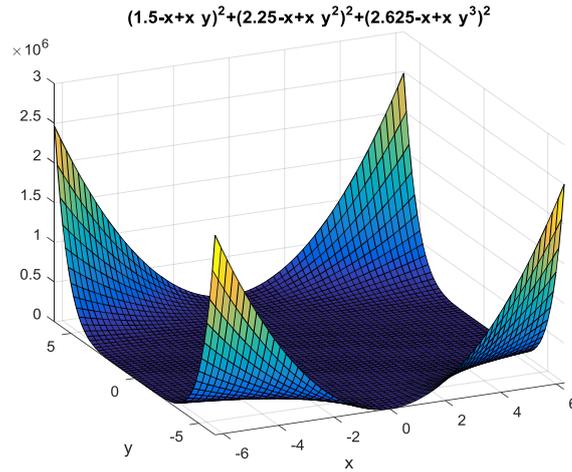
**Gambar 3. Ketahanan terbaik eksperimen 1**

**Tabel 4. Hasil optimasi eksperimen 1**

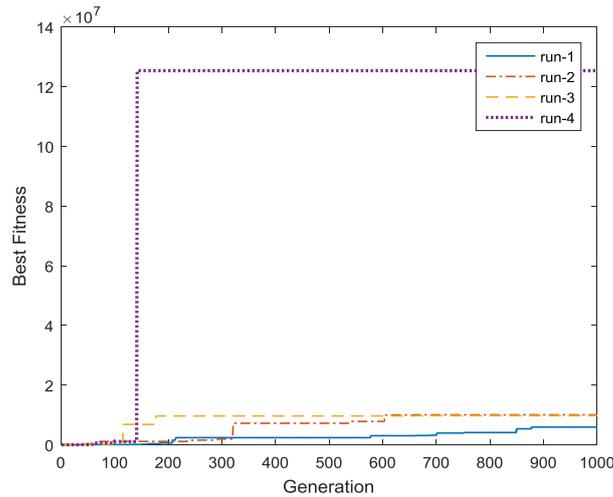
| Run | Batas atas awal | Batas bawah awal | $x_1$  | $x_2$   | Nilai maksimum |
|-----|-----------------|------------------|--------|---------|----------------|
| 1   | 100,0           | 0                | 1,4931 | 1,00340 | 8,9826         |
| 2   | 1,6             | 1                | 1,4970 | 1,00190 | 8,9937         |
| 3   | 7,0             | 2                | 1,4965 | 0,99506 | 8,9713         |
| 4   | 1000,0          | 100              | 1,4815 | 1,01230 | 8,9630         |

**Tabel 5. Hasil optimasi eksperimen 2**

| Run | Batas atas awal | Batas bawah awal | $x$    | $y$     | Nilai maksimum<br>$100/f(x, y)$ |
|-----|-----------------|------------------|--------|---------|---------------------------------|
| 1   | 100,0           | 0                | 3,0103 | 0,50256 | 5,957e+06                       |
| 2   | 1,6             | 1                | 3,0008 | 0,49955 | 1,0067e+07                      |
| 3   | 7,0             | 2                | 3,0036 | 0,50148 | 9,7686e+06                      |
| 4   | 1000,0          | 100              | 3,0000 | 0,50018 | 1,2537e+08                      |



Gambar 4. Fungsi Beale



Gambar 5. Ketahanan terbaik eksperimen 2

**Eksperimen 3**

Eksperimen ketiga adalah identifikasi sifat-sifat struktur. Suatu struktur berderajat kebebasan tunggal dengan massa  $m = 16,274$  ton, redaman  $c = 5.4$  kN-det/m, dan kekakuan  $k = 1050$  kN/m (Arfiadi, 2016) seperti ditunjukkan pada Gambar 6, akan diidentifikasi dengan AG. Dalam hal ini sebagai fungsi obyektif adalah selisih kuadratik terkecil antara respons sesungguhnya dan simulasi, menurut:

$$err = (a_m - a_s)^2 \dots\dots\dots (7)$$

dimana:

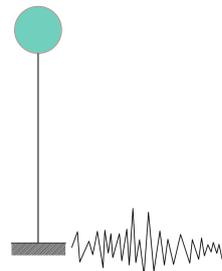
$a_m$  = pengukuran percepatan di lapangan

$a_s$  = percepatan dari simulasi.

Karena AG selalu mencari nilai maksimum maka fungsi obyektif diubah menjadi ketahanan individu menurut:

$$f = \frac{1}{(a_m - a_s)^2} \dots\dots\dots (8)$$

AG yang sama, seperti pada contoh sebelumnya, digunakan untuk mendapatkan sifat-sifat struktur agar respons yang terjadi menyerupai respons dari hasil pengukuran. Dalam hal ini struktur dikenai guncangan dengan 100 titik data dari gempa El Centro 1940 NS. Hasil AG dapat dilihat pada Tabel 6. Nilai ketahanan dalam setiap generasi ditunjukkan pada Gambar 7.

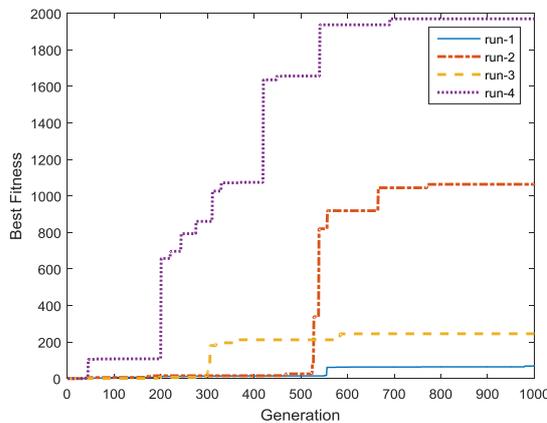


Gambar 6. Eksperimen 3: Identifikasi struktur

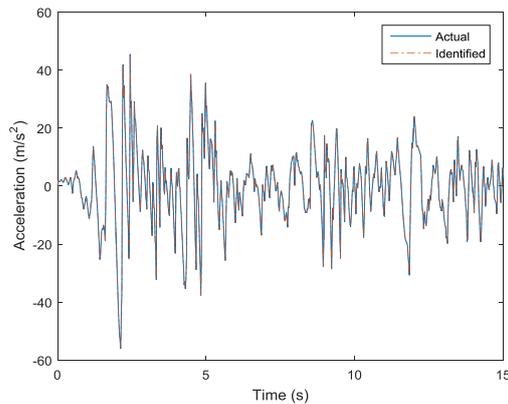
**Tabel 6. Hasil optimasi eksperimen 3**

| Run | Batas atas awal | Batas bawah awal | $m$ (ton) | $c$ (kN-det/m) | $k$ (kN/m) | Nilai maksimum $1/(a_m - a_s)^2$ |
|-----|-----------------|------------------|-----------|----------------|------------|----------------------------------|
| 1   | 100             | 0                | 16,304    | 7,3448         | 1042,9     | 68,499                           |
| 2   | 1,6             | 1                | 16,281    | 5,8587         | 1048,8     | 1067                             |
| 3   | 7               | 2                | 16,289    | 6,3917         | 1045       | 245,32                           |
| 4   | 1000            | 100              | 16,279    | 5,7453         | 1048,2     | 1970,1                           |

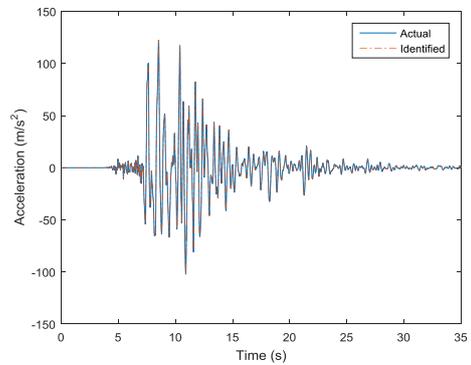
Sifat-sifat struktur yang diperoleh dari hasil AG kemudian digunakan dalam simulasi untuk menilai seberapa jauh hasil yang diperoleh dapat mengidentifikasi sifat-sifat struktur. Sifat-sifat struktur dengan nilai ketahanan paling maksimum (paling baik) diambil dan dikenai guncangan tanah akibat gempa El Centro 1940 NS. Hasil simulasi dibandingkan dengan respons dari struktur sesungguhnya akibat guncangan yang sama. Hasil simulasi ditunjukkan pada Gambar 8. Dari Gambar 8 dapat terlihat bahwa kedua respons mempunyai kesesuaian yang sangat baik, sehingga dapat disimpulkan bahwa AG dapat mengidentifikasi sifat-sifat struktur. Demikian pula jika struktur disimulasi terhadap gempa Kobe 1995; hasil identifikasi memberikan kesesuaian yang sangat baik, seperti terlihat pada Gambar 9.



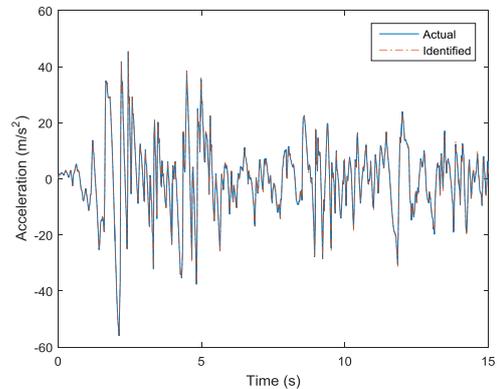
**Gambar 7. Ketahanan setiap run**



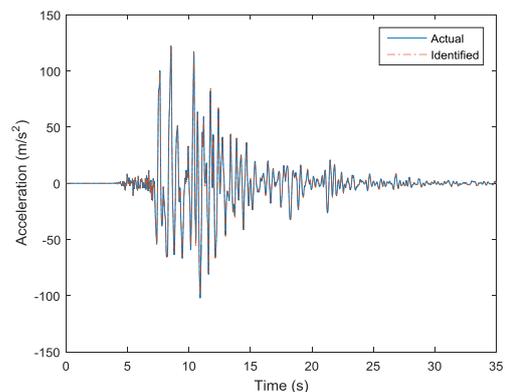
**Gambar 8. Perbandingan respons berdasarkan sifat-sifat struktur aktual dan teridentifikasi terbaik oleh gempa El Centro 1940 NS**



**Gambar 9. Perbandingan respons berdasarkan sifat-sifat struktur aktual dan teridentifikasi terbaik oleh gempa Kobe 1995 NS**



**Gambar 10. Perbandingan respons berdasarkan sifat-sifat struktur aktual dan teridentifikasi terjelek oleh gempa El Centro 1940 NS**



**Gambar 11. Perbandingan respons berdasarkan sifat-sifat struktur aktual dan teridentifikasi terjelek oleh gempa Kobe 1995 NS**

Dalam hal ini ingin diketahui pula hasil identifikasi dengan AG yang mempunyai ketahanan yang terjelek (ketahanan paling rendah). Hasil simulasi dengan sifat-sifat struktur yang teridentifikasi dengan ketahanan terendah dapat dilihat pada Gambar 10 dan 11 oleh rekaman gempa El Centro 1940 NS dan Kobe 1995 NS, berturut-turut. Dari hasil simulasi tampak bahwa AG dapat mengidentifikasi struktur dengan baik, ditinjau dari perbandingan respons yang ada.

## Kesimpulan

AG riil dengan operator kawin silang asimetrik telah dibahas dalam makalah ini. Dengan menggunakan operator kawin silang asimetrik dapat dieksplorasi domain penyelesaian yang terletak di luar domain awal. Hal ini sangat berguna untuk domain yang tidak diketahui di awal optimasi. Selain itu operator kawin silang yang diajukan cocok digunakan untuk penyelesaian masalah optimasi dalam bidang teknik dengan variabel desain yang selalu positif. Tampak pula bahwa walaupun dengan nilai awal yang berbeda-beda, selalu dihasilkan nilai global optimum yang mendekati nilai global yang sesungguhnya.

## Daftar Pustaka

- Alqeedra, M., Arafa, M., dan Ismail, M., 2011. Optimum Cost of Prestressed and Reinforced Concrete Beams Using Genetic Algorithms, *Journal of Artificial Intelligence*, 4 (1), 76-88.
- Arfiadi, Y., 2016. Reducing Response of Structures by Using Optimum Composite Tuned Mass Dampers, *World Multidisciplinary Civil Engineering-Architecture-Urban Planning Symposium*, Prague, Czech Republic.
- Arfiadi, Y., dan Hadi, M., N., S., 2001. Optimal Direc (Static) Output Feedback Controller Using Genetic Algorithms, *Computers and Structures*, 79 (17) 1625-1634.
- Arfiadi, Y., dan Hadi, M., N., S., 2011. Optimum Placement and Properties of Tuned Mass Dampers Using Hybrid Genetic Algorithms. *International Journal of Optimization in Civil Engineering*, 1, 39059.
- Arfiadi, Y. dan Zebua, A., W., 2015. Cross Sections and Prestressing Forces Optimizations of Prestressed Concrete Beams, *11th World Congress on Structural and Multidisciplinary Optimisation*. Sydney, Australia.
- Frans, R., dan Arfiadi, Y., 2014. Sizing, Shape, and Topology Optimizations of Roof Trusses Using Hybrid Genetic Algorithms, *Procedia Engineering*, 185-195.
- Grefenstette, J., J., 1986. Optimization of Control Parameters for Genetic Algorithms, *IEEE Transaction on System, Man, and Cybernetics*, 16 (1), 122-128.
- Guerra, A., dan Kioussis, D., P., 2006. Design Optimization of Reinforced Concrete Structures, *Computers and Concrete*, 3(5), 313-334.
- Holland, J., 1992. *Adaptation in Natural and Artificial Systems*, MIT Press, Mass.
- Kicinger, R., Arciszewski, T., dan De Jong, K., A. 2005. Evolutionary Computation and Structural Design: a Survey of The State of The Art, *Computers and Structures*, 83 (23-24), 1943-1978.
- Michalewicz, Z., 1996. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin.
- Rajeev, S., dan Krishnamoorthy, C., S., 1992. Discrete Optimization of Structures Using Genetic Algorithms, *Journal of Structural Engineering*, 118 (5), 1233-1250.
- Sesok, D. dan Belevicius, R., 2007. Use of Genetic Algorithms in Topology Optimization of Truss Structure, *Mechanika*, 2 (64).
- Wikipedia.org. 2016. [https://en.wikipedia.org/wiki/Test\\_function\\_for\\_optimization](https://en.wikipedia.org/wiki/Test_function_for_optimization), diakses 28 Agustus 2016.