

## OPTIMASI METODE *DISCRIMINATIVELY REGULARIZED LEAST SQUARE CLASSIFICATION* DENGAN ALGORITMA GENETIKA

\*Ariadi Retno Tri Hayati Ririd, \*\*Agus Zainal Arifin, \*\*\*Anny Yuniarti

\*Program Studi Manajemen Informatika, Gedung AH, Politeknik Negeri Malang  
Jl. Soekarno Hatta No.9 Malang 65141

\*\*&\*\*\*Program Pasca Sarjana Jurusan Teknik Informatika, ITS

Jl. Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111

E-Mail: \*ririd@cs.its.ac.id, \*\*agusza@cs.its.ac.id, \*\*\*anny@cs.its.ac.id

### Abstrak

Metode regularisasi dianalisa untuk mengklasifikasi data sehingga diperoleh hasil pengklasifikasian yang lebih tepat dalam pengenalan pola. Metode *Discriminatively Regularized Least Square Classification (DRLSC)* digunakan sebagai metode pengklasifikasian data berdasarkan informasi diskriminatif dengan menerapkan penghitungan matrik *Laplacian*. Kelemahan *DRLSC* adalah hasil dari pengklasifikasian data dipengaruhi oleh jumlah tetangga terdekat dari setiap data ( $K$ ) dan parameter regularisasi ( $\eta$ ). Penelitian ini bertujuan untuk menentukan jumlah tetangga terdekat dan parameter regularisasi secara otomatis dengan meminimalkan *fitness* berdasarkan kesalahan pembelajaran untuk menghasilkan bobot dengan kualitas yang baik. Penentuan nilai dilakukan dengan menggunakan Algoritma Genetika (*GA*) sebagai metode optimasi. *GA* menggunakan probabilitas *crossover* 100% dan mutasi *gaussian* dengan probabilitas 20%. Seleksi dalam operasi *crossover*, dilakukan berdasarkan urutan data yang memiliki nilai *fitness* terbaik hingga nilai *fitness* terburuk. Metode *GA* yang diterapkan untuk mengoptimalkan metode *DRLSC* menghasilkan nilai *fitness* yang lebih baik jika dibandingkan dengan metode *DRLSC* tanpa optimasi, dengan perbedaan nilai *fitness* antara  $1.0796e-004$  hingga  $0,0048$ .

Kata Kunci: Algoritma Genetika, *Discriminatively Regularized Least Square Classification*, Pengenalan Pola.

### Abstract

*Regularization method has been applied in pattern recognition which aims at producing of classification data in order to obtain a more accurate classification. The classification of data based on the information discriminating based on Laplacian matrix method has been applied with Discriminatively Regularized Least Squares Classification (DRLSC). The weakness of DRLSC is that the result of data classification is strongly influenced by the amount of the nearest neighbors on each data and regularization parameter. The purpose of this study is to determine the number of the nearest neighbors and the regularization parameters automatically by minimizing the fitness based on the learning error to produce good quality. Optimization method used in this study is the learning method of Genetic Algorithm (GA). GA uses crossover probability 100% and random gaussian mutation with a probability 20%. The selection in the crossover operation is based on sequence of the data from best to worst fitness value. GA method which is applied to optimize DRLSC method produces better fitness value if it is compared with DRLSC method without optimization. The difference fitness value is between  $1.0796e-004$  and  $0.004$ .*

Key words: Genetic Algorithm, *Discriminatively Regularized Least Squares Classification*, Pattern Recognition.

## PENDAHULUAN

Metode regularisasi pada pengenalan pola telah mengalami perkembangan misalkan untuk pengklasifikasian dan pengelompokan data (*clustering*). Tujuan dari pengklasifikasian data adalah kemampuan komputer dalam mengenali pola secara otomatis untuk menghasilkan data-data berdasarkan kategori yang berbeda [1]. Pada beberapa metode regularisasi seperti metode *Support Vector Machine (SVM)*, *Regularization Network (RN)*, *Least Square Support Vector Machine (LS SVM)*, *Generalized Radial Basis Function (GRBF)* dan *Manifold Regularization (MR)*, belum mengklasifikasikan data berdasarkan informasi diskriminatif. Selain itu, metode-metode tersebut membutuhkan lebih dari satu parameter regularisasi. Pada pengklasifikasian, informasi diskriminatif bertujuan untuk memaksimalkan jarak data-data yang berbeda kelasnya agar menghasilkan pengklasifikasian yang lebih akurat [2].

Metode *Discriminatively Regularized Least Square Classification (DRLSC)* menggunakan dua grafik berdasarkan konsep keterhubungan data-data pada kelas yang sama (*intraclass*) dan keterhubungan data-data pada kelas yang berbeda (*interclass*) yang bertujuan untuk lebih memaksimalkan pengklasifikasian data berdasarkan informasi diskriminatif. Sedangkan pada metode *SVM*, pengklasifikasian data yang terbagi pada banyak kelas (*multiclass*) dilakukan dengan beberapa kali pengklasifikasian data berdasarkan dua kelas (*binaryclass*). Kelebihan dari metode *DRLSC* selain berdasarkan informasi diskriminatif adalah dapat mengklasifikasikan data secara *multiclass* dengan serangkaian persamaan *linear*. *Multiclass* tidak perlu diklasifikasikan ke dalam bentuk beberapa *binaryclass* dengan menggunakan metode *Least Square* [2]. Metode *DRLSC* membutuhkan satu parameter regularisasi untuk mengklasifikasikan data, sedangkan pada metode sebelumnya, yaitu *MR* membutuhkan dua parameter regularisasi.

Kelemahan dari metode *DRLSC* adalah pada hasil pengklasifikasiannya sangat dipengaruhi oleh jumlah tetangga terdekat setiap data sebanyak  $K$  dan parameter regularisasi. Berdasarkan hal ini maka perlu adanya optimasi pada penentuan jumlah tetangga terdekat pada setiap data ( $K$ ) dan parameter

regularisasi ( $\eta$ ). Penelitian ini mengoptimasikan parameter  $K$  dan  $\eta$  dengan Algoritma Genetika yang bertujuan untuk mendapatkan nilai dari kedua parameter ini secara otomatis dengan meminimumkan *fitness* pembelajaran.

## **DISCRIMINATIVE REGULARIZED LEAST SQUARE CLASSIFICATION (DRLSC)**

Penelitian ini dalam pengklasifikasiannya menggunakan metode *Discriminative Regularization Least Square Classification (DRLSC)* yang dibangun berdasarkan *graph* dan operasi matrik. Konsep *graph* pada metode *DRLSC* diterapkan ketika membangun matrik *adjacency* dan matrik *Laplacian* sedangkan konsep penghitungan matrik diterapkan sebagai dasar dalam penghitungan *Least Square*.

Konsep dasar pembangunan *graph (G)* dengan menggunakan metode *K Nearest Neighbor* digunakan untuk menentukan hubungan antara data. Pada setiap data  $x_i$ , maka dicari  $K$  data yang memiliki jarak yang terdekat kemudian dibangun *edge* yang menghubungkan data  $x_i$  dengan  $K$  data yang terdekat disimpan pada variabel  $ne(i) = \{x_i^1, \dots, x_i^k\}$ . Pada *DRLSC* dibangun dua *graph* berdasarkan *intraclass (G<sub>w</sub>)* dan *interclass (G<sub>b</sub>)*.

Sebelum membangun *graph* maka perlu dibagi data  $ne$  menjadi dua subset, yaitu seperti yang ditunjukkan dalam Persamaan (1) dan (2).

$$ne_w(i) = \{x_i^j \mid \text{jika } x_i^j \text{ dan } x_i \text{ masuk pada kelas yang sama, } 1 \leq j \leq K\} \quad (1)$$

$$ne_b(i) = \{x_i^j \mid \text{jika } x_i^j \text{ dan } x_i \text{ masuk pada kelas yang berbeda, } 1 \leq j \leq K\} \quad (2)$$

Pembangunan *graph G<sub>w</sub>* dan *G<sub>b</sub>* berdasarkan matrik *adjacency* dengan keanggotaan dari  $ne_w$  dan  $ne_b$ . Setelah pembangunan matrik  $G_w$  dan  $G_b$ , maka dilanjutkan dengan pembangunan matrik *Laplacian*. Matrik *Laplacian* dibangun sebagaimana Persamaan (3) dan (4).

$$L_w = D_w - G_w \quad (3)$$

$$L_b = D_b - G_b \quad (4)$$

dimana  $D_w$  adalah matrik diagonal dengan elemen diagonalnya merupakan jumlah seluruh

$j$  elemen dari  $G_w$ . Sedangkan  $D_b$  merupakan matrik diagonal dimana elemen diagonalnya merupakan jumlah seluruh  $j$  elemen dari  $G_b$ .

Pada pengklasifikasian *binary class* dimana parameter  $\gamma \in R^N$ , dapat dirumuskan sebagaimana Persamaan (5).

$$\begin{bmatrix} 0 & 1_N^T \\ 1_N \Omega_\eta + I_N \end{bmatrix} \begin{bmatrix} b \\ \gamma \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (5)$$

Dimana  $b$  adalah variabel bias,  $\gamma_i$  merupakan variabel perkalian *Langrange (Langrange multiplier)*, dan  $S_\eta$  didapatkan dari Persamaan (6).

$$S_\eta = X[\eta L_w - (1 - \eta)L_b]X^T \quad (6)$$

Dimana

$\Omega_{\eta ij} = x_j^T (S_\eta)^+ x_i$ ,  $1_N = [1, \dots, 1]^T$ ,  $\gamma = [\gamma_1, \dots, \gamma_N]^T$ ,  $I_N \in R^{N \times N}$ ,  $I$  merupakan matrik identitas dan  $X$  adalah kumpulan dari data-data  $x$ . Jika data diproyeksikan berdasarkan *kernel*, maka perlu pembangunan *matrik kernel* dengan cara data diproyeksikan ke dalam fungsi *kernel* kemudian dilanjutkan dengan penyelesaian persamaan *linier* [2]. Kernel yang digunakan adalah kernel *gaussian* [2]. Persamaan (7) adalah persamaan untuk *kernel gaussian* [3][4][5].

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (7)$$

Penghitungan bobot ( $w$ ) berdasarkan Persamaan (8).

$$w = \sum_{i=1}^N \gamma_i (S_\eta)^+ x_i \quad (8)$$

Untuk pengklasifikasian *multiclass*, maka pada label kelas dibangun secara *vektor* dengan tujuan dapat mengatasi permasalahan *multiclass*. Jika data  $x_i$  termasuk pada kelas  $k$  maka target kelas adalah  $Y_i = [0, \dots, 1, \dots, 0]^T \in R^c$ , dimana kelas ke- $k$  bernilai 1 dan yang lain bernilai 0. Penghitungan *actual output* pada *multiclass* dapat dilakukan dengan menggunakan Persamaan (9).

$$f(x) = w^T x + b \quad (9)$$

Dimana  $w \in R^{n \times c}$ ,  $b \in R^c$  Persamaan *multiclass* didefinisikan dalam Persamaan (10).

$$\begin{bmatrix} b & \gamma \end{bmatrix} \begin{bmatrix} 0 & 1_N^T \\ 1_N \Omega_\eta + I_N \end{bmatrix} = \begin{bmatrix} 0_N & Y \end{bmatrix} \quad (10)$$

Dimana

$\gamma = [\gamma_1, \dots, \gamma_N]$ ,  $0_c = [0, \dots, 0]^T$ ,  $Y = [y_1, \dots, y_N]$ , parameter yang lain termasuk  $\Omega_\eta$  memiliki penyelesaian yang sama dengan penyelesaian pada *binary class*.

### Algoritma Genetika (GA)

Algoritma Genetika memiliki konsep *natural selection* untuk permasalahan *search* dan optimasi. Dalam hal ini mencari data dengan nilai terbaik dari suatu kumpulan data. Ruang dari semua kumpulan nilai ini dinamakan *search space*. Setiap *point* pada *search space* mampu memberikan suatu solusi. Dengan demikian setiap *point* pada *search space* dapat diberikan nilai *fitness* tergantung dari definisi permasalahan. GA berusaha mencari satu *point* terbaik dari *search space*. Permasalahan dalam hal ini adalah *local minima* dan nilai awal dari pencarian data.

Penggunaan *random* pada GA merupakan hal yang sangat penting, karena pada proses seleksi dan pada reproduksi membutuhkan *random*. Setiap iterasi pada pembelajaran dengan metode GA akan selalu menyimpan populasi yang memiliki informasi yang terbaik. Kelebihan dari algoritma ini adalah tidak adanya pembatasan dalam menentukan permasalahan pada setiap genya. Dengan kelebihan ini, maka GA mampu menyelesaikan semua permasalahan. Kelemahan GA adalah pada *search space*. Ia dapat mengalami kegagalan jika *search space* tidak tepat atau tidak adanya *point* solusi yang tepat pada *search space*. Maka perlu adanya batasan pada *search space* yaitu batas atas dan batas bawah pada setiap gen.

GA tidak hanya menggunakan proses mutasi tetapi juga *recombination* atau dikenal dengan istilah *crossover*. Tujuan dari *crossover* adalah reproduksi dua kromosom untuk menghasilkan kromosom baru dengan karakteristik dari kedua *parent*. Pada ilmu biologi, metode *recombination* yang umum dengan *crossover* adalah dengan menentukan satu titik potong pada setiap kromosom dan dilanjutkan dengan menggabungkan pada setiap setengah kromosom. Hasil dari *crossover* adalah mendapatkan individu baru yang dihasilkan

dari kedua *parent* yaitu bapak dan ibu. Pembentukan individu baru ini sangat penting, karena kedua individu baru akan memiliki karakteristik dari kedua *parent*. Jika kedua *parent* memiliki karakteristik yang baik maka diharapkan individu baru yang terbentuk juga memiliki karakteristik yang baik. Pada *GA*, *parent* adalah data-data yang dibelajarkan yang telah terpilih melalui proses seleksi.

Metode mutasi pada *GA* merupakan cara lain untuk mendapatkan perubahan dalam individu baru dalam hal perubahan gen. Kegunaan dari mutasi adalah membantu mempercepat proses pencarian nilai optimal. Selain itu, mutasi bertujuan untuk menghindari terjadinya *local minima* [6]. Jika *crossover* bertujuan untuk mendapatkan kemungkinan individu yang baru dengan nilai *fitness* yang lebih baik, maka mutasi meningkatkan wilayah *search point* setelah individu baru terbentuk [6]. *GA* berusaha menangani permasalahan dengan menganalisa setiap solusi pada masing-masing populasi. Langkah pertama adalah dengan melakukan kode pada setiap kromosom sesuai dengan permasalahan yang akan diselesaikan. Berikutnya adalah persiapan pembelajaran dengan operator *GA*. Operator reproduksi pada *GA* dilakukan dengan mutasi dan *crossover*.

Seleksi berdasarkan *fitness* digunakan untuk mengetahui kualitas dari data yang terseleksi. Semakin besar nilai *fitness* maka semakin baik kualitas data. Namun dalam permasalahan *minimum cost*, konsep tersebut dibalik menjadi bahwa semakin kecil nilai *fitness* maka kualitas data semakin baik. Berikut adalah penjelasan dari proses-proses yang merupakan pembelajaran *GA* pada setiap iterasinya (lihat Gambar 1):

1. Dilakukan *random* sebesar  $n$  populasi yang merupakan  $n$  kromosom.
2. Mengevaluasi *fitness* dari setiap kromosom.
3. Membangun populasi baru hingga dicapai nilai optimal dengan perulangan sebagaimana berikut:
  - a. *Selection*: memilih dua *parent* kromosom berdasarkan *fitness*. Semakin bagus nilai *fitness*, maka kemungkinan untuk dipilih semakin besar.
  - b. *Crossover*: *crossover* kedua *parent* untuk mendapatkan generasi baru.
  - c. *Mutation*: dengan adanya *mutation probability*, setiap *locus* dilakukan mutasi.

- d. *Accepting*: populasi lama diganti dengan populasi baru.
- e. *Replace*: populasi baru digunakan untuk pembelajaran iterasi berikutnya.
- f. *Test*: jika telah memenuhi hasil yang diinginkan proses berhenti, jika tidak maka lakukan *Loop*.
- g. *Loop*: kembali ke langkah dua untuk mengevaluasi *fitness*.

*Crossover* adalah proses penggabungan genetik dari kedua *parent* untuk mendapatkan satu atau lebih anak yang merupakan generasi baru. Sedangkan mutasi adalah perubahan genetik dari individu dengan cara beberapa genetik awal dari individu mengalami perubahan secara *random*.

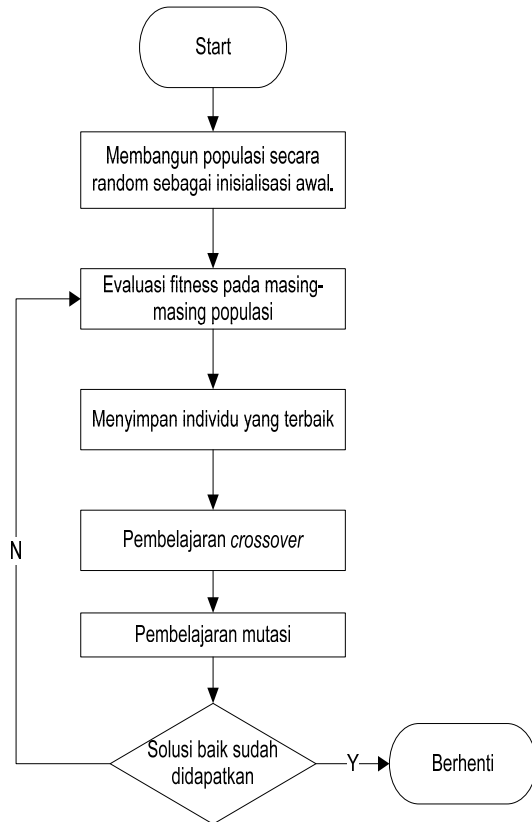
Berdasarkan analisa proses dari *GA*, maka dapat disimpulkan beberapa hal yang penting dalam pembelajaran *GA*, yaitu:

1. Permasalahan yang akan diselesaikan.
2. Penghitungan *fitness*.
3. Variasi dari variabel dan parameter pada kromosom.
4. Hasil akhir dan cara untuk pemberhentian pembelajaran (pemberhentian iterasi).

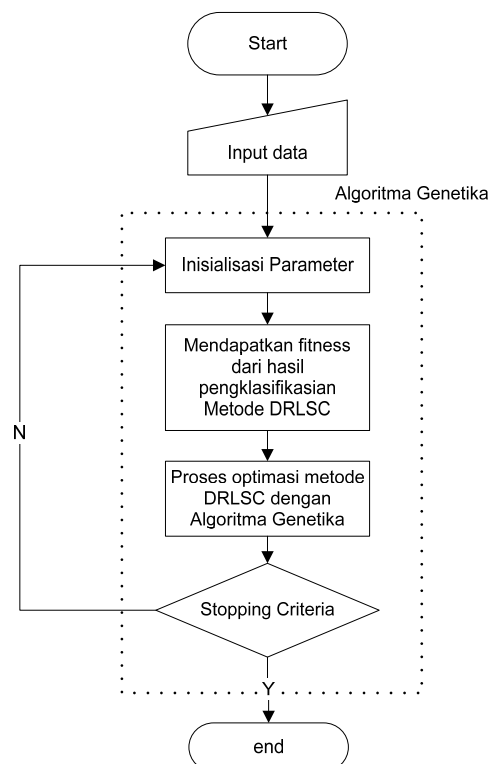
Penghitungan *fitness* adalah cara pembelajaran *GA* mendapatkan solusi untuk mendapatkan data yang menghasilkan suatu nilai yang mendekati optimal. Penentuan nilai *fitness* berdasarkan permasalahan yang akan dipecahkan. Penentuan nilai *fitness* memiliki tujuan untuk mengoptimasi parameter pembelajaran sebagaimana pada Gambar 1.

Dalam penelitian ini, analisa *fitness* berdasarkan *Mean Square Error* digunakan dalam menganalisa optimasi kualitas bobot dan *real code* diterapkan dalam proses pembelajaran. Kelebihan dari *real code* [7] adalah:

1. Meningkatkan tingkat efisiensi dengan tidak perlu adanya proses pengubahan dari nilai *binary* ke nilai *real* ataupun sebaliknya.
2. Meningkatkan presisi nilai.



Gambar 1. Flowchart Pembelajaran GA.



Gambar 2. Konsep Optimasi Metode DRLSC dengan GA.

## PERANCANGAN SISTEM

Rancangan sistem pengklasifikasian data dengan metode *DRLSC*, penentuan jumlah *K* tetangga terdekat pada metode *KNN*, dan penentuan nilai *regularization parameter* secara otomatis ditunjukkan dalam Gambar 2. Garis titik-titik pada Gambar 2. merupakan proses *GA* untuk mengoptimasi *DRLSC*. Data yang dimasukkan adalah data-data yang akan dianalisa, yaitu berupa *dataset IRIS*, *WINE* dan *LENSA*. Sebelum data dibelajarkan perlu adanya proses normalisasi data sebagaimana Persamaan (11).

$$x_{ij} = \frac{x_{ij} - \text{minimum nilai feature}(f)}{\text{maximum nilai feature}(j) - \text{minimum nilai feature}(j)} \quad (11)$$

Normalisasi data dilakukan dengan cara setiap fitur dikurangi nilai minimum pada fitur tersebut kemudian dibagi dengan nilai maksimum fitur dan dikurangi dengan nilai minimum fitur. Tujuan dari proses normalisasi data adalah menghindari nilai yang terlalu besar antara data pembelajaran dan data uji coba.

### Inisialisasi Parameter

Langkah awal untuk memperoleh nilai parameter *K* dan parameter *regularization* yang optimum adalah dengan inisialisasi. Inisialisasi pada penelitian ini dilakukan secara *random* sebagaimana pada *GA*. Inisialisasi awal adalah sebesar *N* data untuk parameter *K* dan *regularization parameter*. Pada penelitian ini jumlah masukan data sebesar 20 data. Setiap data terdiri dua bagian, berupa parameter yang akan dioptimasi, yaitu nilai *K* dan *Ita* ( $\eta$ ). Langkah inisialisasi data adalah sebagai berikut:

1. *Random* 20 data dengan perintah berikut:

$$\text{Populasi} = \text{rand}(20, 2)$$

Maka akan dihasilkan 20 data dimana setiap data memiliki dua parameter yang akan dioptimasi dengan rancangan kromosom/partikel sebagaimana Tabel 1.

2. Pengubahan menjadi nilai sebenarnya pada 20 data berdasarkan batas atas dan batas bawah sebagaimana perintah berikut:

$$X(:, 1) = \text{round}(\text{Batas Bawah} + (\text{Batas Atas} - \text{Batas Bawah}) * \text{Populasi}(:, 1));$$

$$X(:, 2) = \text{Batas Bawah} + (\text{Batas Atas} - \text{Batas Bawah}) * \text{Populasi}(:, 2);$$

Pengertian dari  $X(:,1)$  adalah variabel yang menyimpan hasil dari pemetaan nilai *random populasi* pada kolom pertama terhadap *Batas Atas* dan *Batas Bawah* yang sesuai dengan parameter  $X(:,1)$ . Nilai  $X(:,1)$  adalah data-data yang memiliki nilai  $K$ . Pengertian dari  $X(:,2)$  adalah variabel yang menyimpan hasil pemetaan pada parameter kedua. Nilai  $X(:,2)$  adalah parameter *Ita* yang memiliki batasan lebih kecil atau sama dengan 1 (Tabel 2).

Pengklasifikasian *DRLSC* dilakukan setelah penentuan inisialisasi ke-20 parameter.

**Mendapatkan *Fitness* Hasil Pembelajaran *DRLSC***

Hasil pembelajaran *DRLSC* adalah penentuan data-data pembelajaran yang dikelompokkan pada kelas ke- $i$ , dimana  $i=1$  hingga  $C$  yang merupakan jumlah kelas dengan menerapkan Persamaan (1) hingga (10). Nilai *actual output* yang merupakan nilai keluaran sebenarnya digunakan sebagai dasar menghitung *error* minimum dengan cara mencari selisih target kelas dikurangkan dengan nilai *actual output* pada seluruh data kelas. *Fitness* yang digunakan adalah *Mean Square Error (MSE)*. Berdasarkan nilai *MSE*, akan didapatkan nilai parameter  $K$  dan *Ita* yang paling tepat dengan memperkecil *error* untuk mendapatkan kualitas bobot. Hasil dari pembelajaran *DRLSC* adalah mendapatkan *fitness* berdasarkan *actual output*, dengan persamaan *fitness* sebagaimana Persamaan (12).

$$MeanSquare\ Error(MSE) = \frac{1}{N} \sum (y - f(x))^2 \quad (12)$$

**Optimasi Metode *DRLSC* dengan *GA***

Operator *GA* berperan penting dalam mendapatkan optimasi kromosom. Penelitian ini menggunakan *crossover* dan mutasi sebagai operator *GA*. Persamaan *crossover*  $M$  data didefinisikan dalam Persamaan (13).

$$\begin{aligned} x_i &= \alpha * x_i + (1-\alpha)x_{i+1} & i=1,2,\dots,M-1 \\ x_i &= \alpha * x_i + (1-\alpha)x_i & i=M, \end{aligned} \quad (13)$$

Nilai  $\alpha$  dari Persamaan (13) adalah nilai yang dapat bernilai konstan atau *random*.  $\alpha$  dan  $M$  didapatkan dari jumlah inisialisasi parameter. Berdasarkan Persamaan (13), maka kedua *parent* menghasilkan satu anak. Dimulai dari data pertama *crossover* dengan pola kedua,

dilanjutkan data kedua *crossover* dengan data ketiga demikian berlanjut hingga pola ke- $M$  *crossover* dengan pola pertama. Metode ini sesuai dengan tujuan seleksi yaitu mendapatkan *parent* yang baik untuk generasi berikutnya.

Tabel 1. Rancangan Kromosom pada Penelitian.

Parameter 1	Parameter 2
K	$\eta$

Tabel 2. Rancangan 20 Partikel yang Digunakan.

Data	Populasi (data,1)	Populasi (data,2)
1	$K_1$	$\eta_1$
2	$K_2$	$\eta_2$
3	$K_3$	$\eta_3$
4	$K_4$	$\eta_4$
5	$K_5$	$\eta_5$
6	$K_6$	$\eta_6$
7	$K_7$	$\eta_7$
8	$K_8$	$\eta_8$
9	$K_9$	$\eta_9$
10	$K_{10}$	$\eta_{10}$
11	$K_{11}$	$\eta_{11}$
12	$K_{12}$	$\eta_{12}$
13	$K_{13}$	$\eta_{13}$
14	$K_{14}$	$\eta_{14}$
15	$K_{15}$	$\eta_{15}$
16	$K_{16}$	$\eta_{16}$
17	$K_{17}$	$\eta_{17}$
18	$K_{18}$	$\eta_{18}$
19	$K_{19}$	$\eta_{19}$
20	$K_{20}$	$\eta_{20}$

Dengan pengurutan *fitness*, pola pertama dan kedua adalah populasi dengan nilai *fitness* yang terbaik. Diharapkan generasi berikutnya

akan mendapatkan *search point* yang lebih baik dibandingkan kedua *parent*. Untuk menyimpan individu terbaik pada generasi sebelumnya, maka proses *crossover* dimulai pada data ke-2 (Gambar 3). Jika tidak menyimpan generasi terbaik sebelumnya, maka dapat terjadi naik turunnya nilai *fitness*.

Proses selanjutnya adalah *mutasi*. Proses mutasi adalah mengubah satu atau seluruh gen berdasarkan probabilitas *random* yang bertujuan untuk menghindari *local minimum*. Jika probabilitas sebesar 100%, maka seluruh data dan seluruh gen pada data dimutasi. Jika probabilitas mutasi 0%, maka populasi berikutnya sama dengan populasi sebelumnya. Metode mutasi yang digunakan berdasarkan *random gaussian*. Metode ini didefinisikan dalam Persamaan (14).

$$x'_k = x_k \text{normrand}(0,1,[ \text{jumlah gen}]) \quad (14)$$

Setiap gen ditambahkan dengan bilangan *random* dan dikalikan dengan nilai *gaussian*.

Tabel 3. Hasil Percobaan 1 Tanpa Optimasi.

Data Set	Fitness	Jumlah iterasi	Prosentase uji coba	K	Ita
Iris	1,1e-004	24	93,33%	6	1
Wine	0,0012	24	98,89%	3	0,95
Lensa	0,0049	2	81,82%	3	0,95

Tabel 4. Hasil Percobaan 1 dengan GA.

Data Set	Fitness	Jumlah iterasi	Prosentase uji coba	K	Ita
Iris	2,04e-006	14	93,33%	8	0,9984
Wine	0,002	20	98,89%	11	0,9757
Lensa	5,6e-005	18	81,82%	3	0,9937

Tabel 5. Hasil Percobaan 2 Tanpa Optimasi.

Data Set	Fitness	Jumlah iterasi	Prosentase uji coba	K	Ita
Iris	6,7e-005	24	74,67%	6	1
Wine	2,9e-004	24	93,33%	2	0,95
Lensa	0,005	2	92,31%	2	0,95

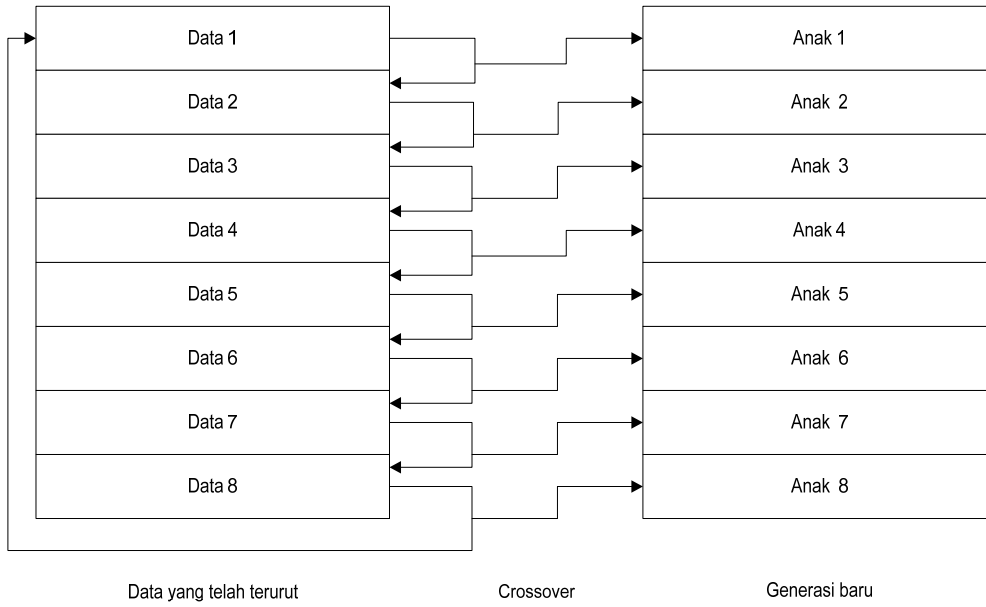
Tabel 6. Hasil Percobaan 2 dengan GA.

Data Set	Error pembelajaran	Jumlah iterasi	Prosentase uji coba	K	Ita
Iris	6,7e-005	16	74,67%	6	1
Wine	1,4e-007	17	93,33%	9	0,9004
Lensa	2,8e-004	21	92,31%	3	0,9915

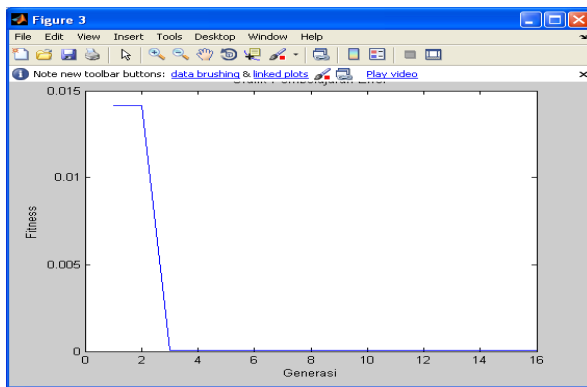
Nilai *gaussian* didapatkan sepanjang jumlah gen yang dibelajarkan. Pada gen pertama, maka gen ditambahkan dengan nilai *gaussian* yang pertama, dan demikian seterusnya. Penentuan nilai *gaussian* dilakukan berdasarkan *normrand*, yaitu penghitungan nilai *gaussian* dari sejumlah data sepanjang jumlah gen yang telah di-*random*. Proses optimasi dilakukan dengan melakukan proses *crossover* dan mutasi pada setiap iterasinya hingga mendapatkan nilai dari kedua parameter, yaitu parameter *K* dan  $\eta$ , untuk menghasilkan nilai *fitness* yang terbaik (meminimumkan *fitness*).

### Stopping Criteria

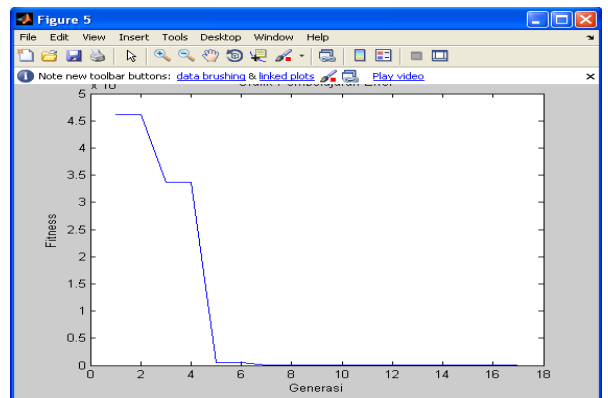
Berdasarkan Gambar 2, proses akan kembali pada penentuan nilai *K* dan  $\eta$  jika belum memenuhi proses *stopping criteria*. Jika telah memenuhi *stopping criteria*, maka proses berhenti (pembelajaran telah konvergen).



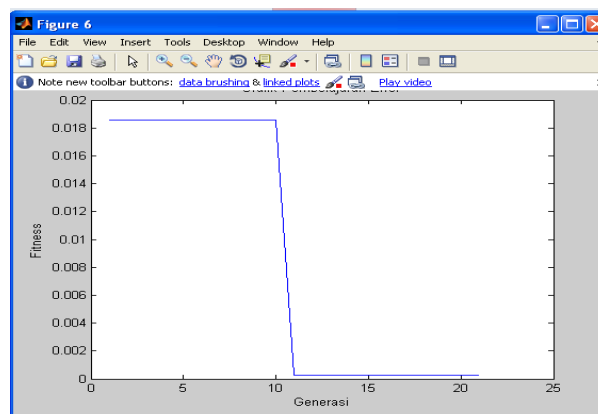
Gambar 3. Proses *Crossover*.



(a)



(b)



(c)

Gambar 4. Grafik Hasil Pembelajaran GA Pada Tabel 6. (a) Grafik Metode GA Data *IRIS*, (b) Grafik Metode GA Data *WINE*, dan (c) Grafik Metode GA Data *LENSEA*.



## HASIL DAN PEMBAHASAN

Penelitian ini menggunakan *dataset* uji coba *IRIS*, *WINE*, dan *LENSA*. Hasil uji coba tanpa optimasi didapatkan dengan jumlah iterasi dipengaruhi oleh jumlah data terkecil dari kelas yang ada dengan lambang  $N$ . Sehingga nilai  $K$  dimulai dari 2 hingga  $N + 1$ . Untuk nilai  $\eta$ , uji coba tanpa optimasi menggunakan batas bawah bernilai 0 dan batas atas bernilai 1. Tabel 3 dan Tabel 4 menunjukkan hasil uji coba dengan data pembelajaran sebesar 50% dan data uji coba sebesar 50% pada setiap kelas.

*Fitness* yang didapatkan dari hasil tanpa optimasi memiliki *error* pembelajaran yang lebih besar jika dibandingkan dengan hasil pembelajaran yang dioptimasi dengan *GA* (Tabel 4). Maka, untuk meningkatkan kualitas bobot yang lebih baik dibutuhkan optimasi untuk mendapatkan nilai parameter  $K$  dan  $\eta$  terbaik. Hasil pembelajaran dengan menggunakan data yang sama sebagaimana data-data dari Tabel 3 yang telah dioptimasi dengan *GA* ditampilkan pada Tabel 4.

Hasil *fitness* pada Tabel 4 memiliki kecenderungan lebih kecil jika dibandingkan dengan uji coba pada Tabel 3. Hasil *fitness* pada Tabel 5 diperoleh dengan menggunakan data pembelajaran yang merupakan data uji coba pada Tabel 3 dan Tabel 4. Nilai *fitness* memiliki kecenderungan menghasilkan nilai *fitness* yang lebih besar jika dibandingkan

*error* pembelajaran pada Tabel 6. Hal ini disebabkan parameter  $K$  dan  $\eta$  pada Tabel 5 belum dioptimasi dengan metode *GA*.

Hasil *fitness* pembelajaran data *IRIS*, *WINE* dan *LENSA* pada Tabel 6 memiliki kecenderungan menghasilkan nilai *fitness* yang lebih kecil jika dibandingkan dengan hasil pada Tabel 3 yang belum dioptimasi. Gambar 4 menunjukkan terjadi penurunan nilai *fitness* pada setiap iterasinya. Proses pembelajaran akan berhenti jika nilai *fitness* telah mendapatkan  $n$  iterasi dengan nilai *fitness* yang sama. Nilai *fitness* pada penelitian adalah *Mean Square Error*, dimana semakin kecil nilai *error* maka diharapkan kualitas bobot yang didapatkan semakin baik.

## SIMPULAN

Berdasarkan hasil uji coba dan pembahasan, dapat diambil simpulan sebagaimana berikut:

1. Parameter  $K$  dan  $\eta$  dapat secara otomatis didapatkan dengan menerapkan metode optimasi *GA*.
2. *GA* yang diterapkan untuk mengoptimasi metode *DRLSC* memiliki kecenderungan mampu menghasilkan nilai *fitness* yang lebih baik jika dibandingkan metode *DRLSC* tanpa optimasi yaitu antara  $1.0796e-004$  dan  $0.0048$ .

## DAFTAR PUSTAKA

- [1] Bishop CM. *Pattern Classification and Machine Learning*. New York: Springer Berlin Heidelberg. 2006.
- [2] Xue H, Chen S, and Yang Q. Discriminatively Regularized Least-Squares Classification. *Science Direct, Pattern Recognition*. 42: 93-104. 2009.
- [3] Rifkin RM and Lippert RA. *Computer Science and Artificial Intelligence Laboratory Technical Report*. Notes on Regularized Least Squares. 2007. URL: <http://cbcl.mit.edu/projects/cbcl/publications/ps/MIT-CSAIL-TR-2007-025.pdf>, diakses tanggal 10 Agustus 2009.
- [4] Arune K and Gopal M. Least Squares Twin Support vector machines for pattern classification. *Science Direct, Expert Systems with Applications*. 36:7535-7543.2009.
- [5] Schölkopf B, Guyon I and Weston J. Statistical Learning and Kernel Methods in Bioinformatics. *Proceedings NATO Advanced Studies Institute on Artificial Intelligence and Heuristics Methods for Bioinformatics*. 1-21. 2001.
- [6] Sivanandam SN and Deepa SN. *Introduction to Genetic Algorithms*. New York: Springer Berlin Heidelberg. 2008.
- [7] Wright AH. *Genetic Algorithms for Real Parameter Optimization*. Montana Missoula: Department of Computer Science, University of Montana Missoula. 1991.