JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 4
No. 2
Desember
2018

Optimasi Parameter pada *Fast Correlation Based Filter* Menggunakan Algoritma Genetika untuk Klasifikasi Metagenome

Hanif Bagus Guritno[#1], Toto Haryanto[*,+2], Aziz Kustiyo[*3], Irman Hermadi[*4]

[#]*PT Nusantara Baskara Jaya, Jl. Raya Kutisari Indah No.125-139, Kutisari, Tenggilis Mejoyo, Surabaya, Jawa Timur*
[1]hanifbg48@gmail.com
[*]*Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam,Institut Pertanian Bogor*
Jalan Meranti Wing 5 Kampus IPB Dramaga 16680
[3]azizku@apps.ipb.ac.id
[4]irmanhermadi@apps.ipb.ac.id
[+]*Fakultas Ilmu Komputer Universitas Indonesia, Kampus UI Depok, Indonesia 16424*
[2]totoharyanto@apps.ipb.ac.id

***Abstrak*— *Metagenome* merupakan mikroorganisme yang diambil secara langsung dari alam. Proses *sequencing* genom dari *metagenome* mengakibatkan bercampurnya berbagai organisme. Hal ini menyebabkan kesulitan pada proses perakitan DNA. Oleh karena itu, dibutuhkan proses pemilahan yang disebut *binning*. Pada proses *binning* dengan pendekatan komposisi, teknik yang dilakukan adalah dengan *supervised learning*. Salah satu tahapan dalam *supervised learning* yaitu ekstraksi fitur, penelitian ini menggunakan metode ektraksi fitur *n-mers*. Besarnya parameter *n* pada metode ekstraksi fitur *n-mers* akan mengakibatkan dimensi fitur yang tinggi. Penelitian ini bertujuan untuk menerapkan algorime *fast-correlation based filter (FCBF)* untuk mereduksi dimensi fitur yang dihasilkan *n-mers* dan mengoptimasi parameter *threshold* pada *fast-correlation based filter* menggunakan algoritme genetika. Penelitian ini diuji menggunakan klasifikasi *k-nearest neighbour*. Performa terbaik diperoleh ketika *n* = 7 dan k = 3 dengan akurasi mencapai 99.41% dengan nilai *threshold* 0.67788. Dengan optimasi, waktu komputasi menjadi lebih efisien karena jumlah fitur sudah tereduksi.**

Kata kunci*— Algoritme genetika, *Binning*, *Fast-correlation based filter*, *N-mers*, *K-nearest neighbour

I. PENDAHULUAN

Penelitian di bidang *metagenome*ika menjadi salah satu bidang kajian bioinformatika yang terus berkembang hingga saat ini. *Metagenome* merupakan sekumpulan mikroorganime yang diambil secara langsung dari lingkungan tanpa dilakukan kultur langsung [1]. Proses *sequencing* genom dari sekumpulan mikroorganime yang diambil secara langsung tersebut mengakibatkan bercampurnya berbagai organisme. Hal ini menyebabkan kesulitan pada proses perakitan (*assembly*). Oleh karena itu, dibutuhkan proses pemilahan yang disebut *binning*.

Proses *binning* adalah proses pemilahan suatu genom ke dalam beberapa kelompok yang merepresentasikan sekumpulan genom dari organisme memiliki keterkaitan [2]. Pada prinsipnya *binning* dapat dilakukan dengan pendekatan homologi dan komposisi. Pendekatan komposisi dilakukan dengan metode *supervised learning* [3], *semi supervised* [4], weakly supervised [5] dan *unsupervised learning*.

Penelitian terkait mengenai pengklasifikasian fragmen *metagenome* pernah dilakukan antara lain [6] yang menggunakan sekuens mikroba. Dari penelitian tersebut diperoleh bahwa penentuan *n-mers* memberikan kontribusi terhadap akurasi. Penelitian lain yang terkait dengan klasifikasi *metagenome* dilakukan pada [7]. Penggunaan varian dari Hidden Markov Model yaitu Abstraction Augmented Markov Model (AAMM). Hasil klasifikasi menunjukkan bahwa pada level genus, mampu mencapai 98, 62% secara rata-rata. Sementara pendekatan *unsupervised* yakni *clustering metagenome* telah dilakukan oleh beberapa penelitian di antaranya [8] yang mengusulkan proses *clustering* dengan waktu yang cepat dibandingkan dengan beberapa metode yang telah ada. Penelitian yang dilakukan pada [9] pun diklaim berhasil melakukan *clustering* pada data *metagenome* berskala besar dengan metode RAMMCAP. Growth Self Organizing Map (GSOM) telah berhasil dilakukan untuk melakukan *clustering metagenome* pada [10].

Dimensi fitur yang telalu tinggi menjadi permasalahan pada proses komputasi yang menyebabkan tidak efisien. Bukan hanya itu, bahkan terkandang tingginya dimensi ini tidak memberikan kontribusi yang signifikan terhadap hasil akurasi karena di antara variabel yang ada ternyata memiliki korelasi linear misalnya. Oleh karena itu, masalah tersebut dapat diatasi dengan metode reduksi dimensi fitur atau seleksi fitur. Penelitian terkait dengan seleksi fitur pada *metagenome* di antaranya [11], [12] dan [13].

Seleksi fitur adalah salah satu tahapan praproses dalam klasifikasi untuk memilih fitur-fitur yang relevan terhadap data. Salah satu algoritme seleksi fitur adalah Fast

Submitted 24-09-2018; Revised 21-11-2018; Accepted 30-12-2018

Correlation Based Filter (FCBF) yang dikembangkan oleh Yu dan Li [14].

Konsep dasar dari algoritme ini adalah menghilangkan fitur-fitur yang tidak relevan berdasarkan nilai *threshold* yang telah ditentukan serta menghilangkan fitur-fitur yang *redundant* terhadap fitur lain. Penelitian sebelumnya pernah dilakukan oleh Dinilhak [15] menggunakan Support Vector Machine (SVM) sebagai *classifier* dan FCBF sebagai seleksi fiturnya. Tanpa seleksi fitur FCBF dihasilkan akurasi berkisar 84.93%-99.01% dan dengan seleksi fitur FCBF dihasilkan akurasi berkisar 79.13%-96.68%. Dari Penelitian tersebut dapat dilihat akurasi menurun saat penggunaan seleksi fitur FCBF sehingga dibutuhkan algoritme untuk mengoptimalkan parameter *threshold* pada FCBF.

Algoritme genetika (GA) merupakan salah satu algoritme yang banyak digunakan untuk menyelesaikan permasalahan optimasi dan mencari pola baru yang diharapkan memiliki nilai *fitness* yang lebih baik dari seluruh kromosom. Keberhasilan GA untuk melakukan optimasi telah dilakukan pada beberapa penelitian antara lain [16], [17] dan [18].

Penelitian sebelumnya pernah dilakukan pada [19] untuk mengoptimasi parameter pada SVM. Penelitian tersebut menghasilkan akurasi sebesar 65.3% dengan menggunakan algoritme *grid search* dan 67.3% dengan menggunakan algoritme genetika untuk panjang fragmen 400 *base pair* (bp). Oleh karena itu, penelitian ini melakukan klasifikasi fragmen *metagenome* dengan metode k-nearest neighbour (KNN) dan FCBF sebagai peyeleksi fitur, serta algoritme genetika untuk mengoptimasi nilai *threshold* dari FCBF tersebut. Dibandingkan penelitian sebelumnya, seleksi fitur FCBF yang digunakan masih belum dioptimasi sehingga fitur optimal bisa jadi belum ditemukan. Selain itu, seleksi fitur tanpa proses algoritme optimasi justru menghasilkan akurasi yang tidak begitu baik.

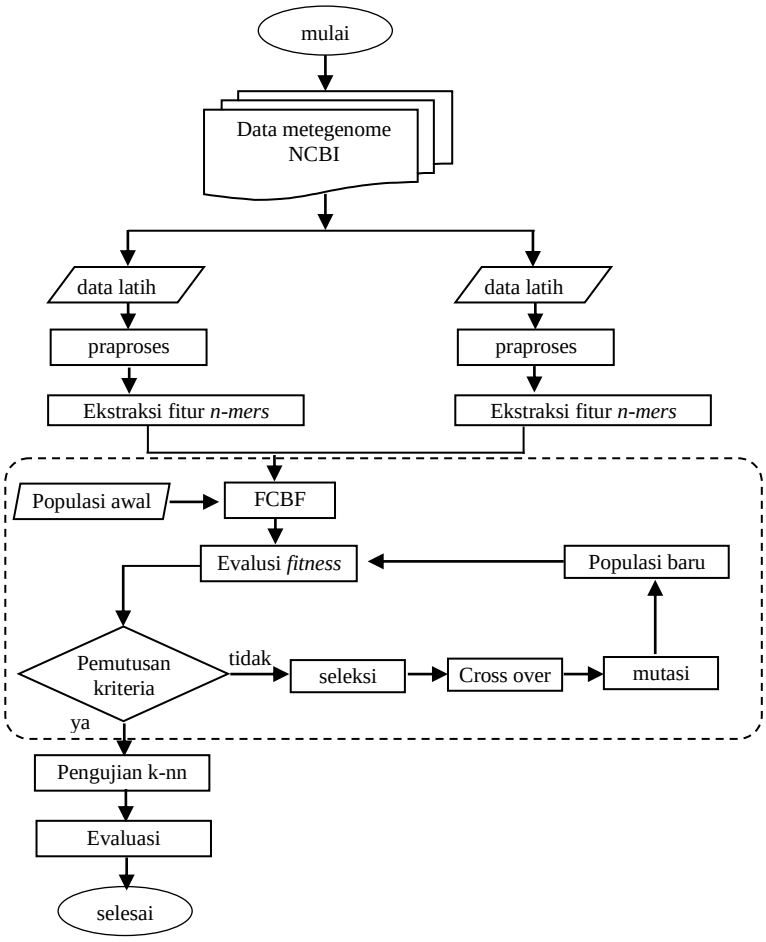
II. METODE PENELITIAN

A. Dataset

Secara keseluruhan data *metagenome* yang digunakan pada penelitian ini sebanyak 214 data organisme. Data tersebut terbagi atas 150 data mikroorganisme dari 25 genus sebagai data latih dan 64 data mikroorganisme dari 16 genus sebagai data uji. Data tersebut diunduh dari situs *national center for biotechnology information* (NCBI) melalui alamat website pada tautan berikut ini. ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/all.fna.tar.gz.

B. Diagram Alir Penelitian

Diagram alir penelitian mennjukkan proses penelitian ini secara umum dari mulai pengambilan data sampai dengan analisis hasil dan evaluasi sehingga penelitian ini dapat terselesaikan. Diagram alir penelitian ini disajikan pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

C. Praproses

Sekuen DNA *metagenome* yang telah dipilih akan di-*sequence* menggunakan perangkat lunak MetaSim dengan nilai coverage = 10. MetaSim merupakan perangkat lunak simulasi *sequencer* baik untuk data genom atau *metagenome* [20]. Data yang telah dipilih akan diproses menggunakan MetaSim untuk melakukan *sequencing* sepanjang 500 bp (*base pair*).

D. Ekstrasksi Ciri

Pada tahap ini dilakukan dengan pembacaan frekuensi dari kombinasi basa nukleotida ACGT (adenin, sitosin, guanin, timin) yang mungkin terbentuk. Tahap ini menggunakan ekstraksi ciri *n-mers* untuk k = 5 dan k = 7. Pola kemunculan k adalah pola yang menampilkan k pada suatu waktu dalam suatu *sequences*. Pola kemunculan k dihitung menggunakan kombinasi empat basa nukleotida. Pada penelitian ini, nilai k = 5 dan 7 akan berimplikasi pada sejumlah 4^5 (1024) dan 4^7 (16384) pola kemunculan fitur yang terbentuk.

E. Fast Correlation Based Filter (FCBF)

FCBF merupakan algoritme seleksi fitur yang dikembangkan oleh Yu dan Liu [14]. Prinsip algoritme ini adalah bahwa suatu fitur yang baik adalah fitur-fitur yang relevan terhadap kelas tapi tidak *redundant* terhadap fitur yang lain. Konsep dasar FCBF adalah bahwa suatu fitur yang baik adalah fitur yang relevan terhadap kelas tapi tidak *redundant* terhadap fitur relevan yang lain. Oleh karena itu, akan dilakukan suatu pendekatan dengan mengukur korelasi antara dua variabel acak menggunakan *symmetrical uncertainty* (SU). Nilai SU ada pada selang 0 sampai dengan 1. *Symmetrical uncertainty* dirumuskan pada persamaan (1).

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \quad (1)$$

H(X) adalah nilai *entropy* dari variabel (X) yang diformulasikan pada persamaan (2).

$$H(X) = -\sum_i P(X_i)\log_2(P(X_i)) \quad (2)$$

Sementara H(X|Y) adalah *entropy* dari variabel X apabila diketahui variabel Y yang didefinisikan dalam persamaan (3)

$$H(X \setminus Y) = -\sum_j P(Y_j)\sum_i P(X_i|Y_j)\log_2(P(X_i|Y_j)) \quad (3)$$

IG merupakan *information gain* dengan persamaan (4)

$$IG(X|Y) = H(X) - H(X|Y) \quad (5)$$

F. Representasi Solusi

Representasi solusi di dalam algoritme genetika dinyatakan dengan kromosom. Berbagai teknik dapat dilakukan untuk merepresentasikan suatu solusi. Pada GA, satu kromosom biasanya menyatakan satu buah variabel penyelesaian dan setiap kromosom terdiri atas beberapa gen. Nilai *threshold* yang dioptimasi akan direpresentasikan menjadi suatu bilangan biner. Adapun tahapan pengkodean *threshold* menjadi kode biner [21]:

- Menentukan batas bawah (a) dan batas atas (b)
- Menentukan tingkat ketelitian (d)
- Menentukan jumlah bit kromosom berdasarkan formulasi pada persamaan (6)

$$[^2\log(((b-a)*10^d)+1)] \quad (6)$$

G. Pembentukan Populasi Awal

Populasi awal dibentuk dengan membangkitkan 15 kromosom secara acak. Hal ini dikarenakan pembangkitan populasi awal tersebut sudah merepresentasikan ruang pencarian. Dari populasi awal tersebut akan diproses menggunakan algoritme genetika untuk mendapatkan populasi baru. Proses pembentukan populasi baru akan dilakukan sampai nilai *threshold* dan akurasi dianggap optimal.

H. Evaluasi Fitness

Pada tahap evaluasi *fitness*, setiap kromosom yang digunakan akan bergantung pada nilai *threshold* yang dipilih sehingga setiap kromosom memiliki nilai *fitness* masing-masing. Nilai *fitness* dipengaruhi oleh akurasi dan jumlah fitur yang terpilih. Kedua nilai tersebut diberi pembobotan, yaitu 75% untuk akurasi dan 25% untuk jumlah fitur yang terpilih. Nilai akuirasi dari hasil klasifikasi yang diperoleh masing-masing kromosom akan dievaluasi menggunakan fungsi *fitness* dengan persamaan (7)

$$fitness = \frac{3x + (1 - \frac{Y}{Y_t})}{4} \quad (7)$$

dengan X merupakan nilai akurasi, Y merupakan jumlah fitur yang terpilih, dan Y_t merupakan jumlah fitur total.

I. Kriteria Pemberhentian

Tahapan ini berfungsi untuk memeriksa kondisi populasi saat ini terhadap kriteria pemberhentian. Ketika kriteria telah terpenuhi, maka proses pembentukan populasi baru dihentikan. Beberapa kriteria pemeberhentian dalam GA untuk penelitian ini adalah

- Banyaknya generasi maksimum.
- Mencapai durasi maksimum (pada penelitian ini durasi maksimum adalah 45 iterasi)
- Nilai *fitness* terbaik sudah tidak mengalami peningkatan selama beberapa generasi terakhir.
- Sudah mencapai nilai *fitness* yang ditentukan sebelumnya.
- Menggunakan kriteria berhenti dinamis setelah tingkat keyakinan dari nilai *fitness* terbaik telah tercapai.

Kriteria di atas menandakan telah didapatkan *threshold* yang optimal. Apabila kriteria belum tercapai maka dilanjutkan pembentukan populasi baru.

J. Seleksi, crossover, mutasi dan elitisme

Seleksi, *crossover*, mutasi dan elitisme adalah rangkaian iterasi di dalam dalam proses optimasi menggunakan algoritme genetika. Pada proses seleksi ini, teknik yang digunakan adalah metode *roulette wheel*. Untuk *crossover*, penelitian ini menggunakan peluang *crossover* sebesar 60%. Selanjutnya akan dibangkitkan bilangan acak dari 1 sampai 16 untuk setiap kromosom. Bilangan ini berguna untuk menentukan posisi pada saat

akan dilakukan *crossover*. Untuk menjaga keberagaman individu, mutasi yang dilakukan pada penelitian ini sebesar 10% dengan pemilihan gen secara acak. Untuk mempertahankan generasi yang baik, dua kromoson tetap dipertahankan di generasi berikutnya. Proses ini yang dikenal dengan elistisme.

K. Evaluasi

Untuk menguji hasil klasifikasi dilakukan dengan menggunakan akurasi, sensitifitas, spesifisitas berdasarkan matrik konfusi pada Tabel 1.

TABEL I
MATRIK KONFUSI

Kelas Aktual	**Kelas Prediksi**	
	A	~A
A	*True Positive* (TP)	*False Negative* (FN)
~A	*False Positive* (FP)	*True Negative* (TN)

Nilai akurasi, sensitifitas dan spesifisitas didefinisikan pada persamaann (8), (9) dan (10).

$$akurasi = \sum \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (8)$$

$$sensitifitas = \sum \frac{TP}{TP + FN} \times 100\% \quad (9)$$

$$spesifisitas = \sum \frac{TN}{TN + FP} \times 100\% \quad (10)$$

III. HASIL PENELITIAN

A. Dataset

Data hasil simulasi menggunakan perangkat lunak Metasim dalam format FastA dengan menggunakan coverage 10 dan panjang pasang basa yang digunakan adalah 500 atau 500 bp. Contoh data yang diperoleh dari perangkat lunak tersebut seperti dilihat pada Gambar 2.

```
>r1.1 | SOURCES={GI=50196905,bw,1016309-
1016766}|ERRORS={}|SOURCES_1 + "STR. 'Ames
Ancestor' choromosoe"
(2b301d2cec11c944b70447bada91610998
ACTTCATTAAGAATTATTTTTAACATGATCGTTTTAACCCCC
TTTTTGCTATTTTTTTTATAGGATTAAATTTAGGATTAGTAG
GGTAGATAGTAGTAGCCCCCCAAAAAGGGGGATAAAATTATT
ATTATTAGGGGCTCTGGCGGGCTTCGGCTGCGGGGGGGGCTT
TTTTGGGGGGGATTGGATTTTGGGATTTAGGAGAGGGGATTT
```

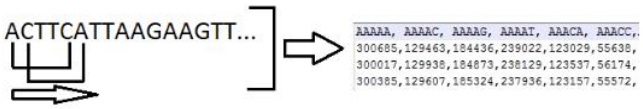
Gambar 2. Contoh Ilustrasi Fail Hasil Simulasi Metasim

Data hasil simulasi tersebut kemudian diproses melalui tekniik *parsing* untuk mendapatkan fitur dengan metode *n-mers*.

B. Ekstraksi Fitur n-mers

n-mers merupakan salah satu satu teknik untuk mendapatkan fitur dari sekuens *metagenom* hasil simulasi Metasim. Pada prinsipnya *n-mers* akan menghitung frekuensi kemuculan suatu basa A, C, G atau T. Banyaknya fitur yang terbentuk akan bergantung pada nilai n yang dipilih. Penelitian ini menggunakan nilai n = 5 dan n = 7 sehingga jumlah fitur yang terbentuk adalah sebanyak 4^n dengan n ={5,7}. Dengan demikian, jumlah fitur yang terbentuk adalah sebanyak 1024 untuk n = 5 dan 16384 untuk n = 7. Jumlah ini sudah cukup banyak untuk diseleksi menggunakan FCBF yang akan dioptimasi dengan menggunakan GA.

Untuk memudahkan pemahaman mengenai ekstraksi fitur menggunakan *n-mers*, dapat dilihat pada ilustrasi yang terdapat di Gambar 3.



Gambar 3. Ilustrasi hasil ekstraksi fitur = *n-mers* dengan k = 5

C. Fast Correlation Based Filter (FCBF)

Setelah dilakukan ekstraksi *n-mers*, proses berikutnya adalah melakukan proses reduksi dimensi fitur dengan FCBF. FCBF memiliki parameter *threshold* yang harus dioptimasi untuk menghasilkan akurasi yang baik. Pengoptimalan parameter *threshold* menggunakan algoritme genetika. Parameter yang dihasilkan dari optimasi menggunakan algoritme genetika akan menjadi input dalam FCBF. Reduksi dimensi fitur FCBF menggunakan perangkat lunak Weka versi 3.7. Hasil fitur yang terpilih pada 5-mers dengan nilai *threshold* = 0 ditunjukkan pada Tabel 2.

TABEL II
HASIL FITUR TERPILIH FCBF DENGAN 5-MERS

Ranked Attributes :		
0.796	590	GCACT
0.779	231	ATGCG
0.778	79	ACATG
0.766	902	TGACC
0.766	1010	TTTAC
0.766	571	GATGG
0.762	102	ACGCC
0.761	338	CCCAC
0.760	300	CAGTT
0.759	330	CCAGC
0.759	636	GCTGT

D. Representasi Solusi Algoritme Gentika

FCBF diterapkan pada 5-mers dan 7-mers untuk mendapatkan batas atas sebagai salah satu parameter untuk merepresentasikan solusi dengan algoritme genetika. Representasi batas bawah dan batas atas disajikan pada Tabel 3.

TABELIII
BATAS BAWAH DAN BATAS ATAS SETIAP *N-MERS*

n-mers	batas bawah	Batas atas
5-mers	0	0.79621
7-mers	0	0.74378

Batas atas diperoleh berdasarkan nilai SU tertinggi pada masing-masing *n-mers* sementara batas bawah merupakan nilai terkecil dari *threshold*. Adapun tingkat ketelitian yang digunakan sejumlah lima angka di belakang koma. Dengan penggunaan lima angka di belakang koma ini, perbedaan nilai *threshold* sudah dapat terlihat.

Tahap selanjutnya adalah penentuan jumlah bit kromosom masing-masing *n-mers* dengan persamaan (6). Berdasarkan persamaan tersebut, jumlah bit yang dihasilkan untuk setiap *n-mers* adalah 17 bit

$$[^{2}\log(((0.79621-0)*10^{4})+1)] = 17{,}27 \cong 17$$

E. Pembentukan Populasi Awal

Populasi awal pada GA didapatkan dengan cara membangkitkan 15 kromosom yang setiap individu terdiri atas 17 bit gen secara acak Pada penelitian ini populasi direpresentasikan dalam bentuk *bit-string*. Tabel 4 menunjukan proses pembentukan populasi awal yang telah dibangkitkan secara acak.

TABEL IV
HASIL PEMBANGKITAN POPULASI AWAL PADA ALGORITME GENETIKA

Individu	Gen1	Gen2	…	…	Gen 17
Ind. 1	1	0	1	0	1
Ind. 2	1	1	0	0	1
…	0	1	1	1	0
…	1	0	0	1	1
Ind. 15	0	1	0	1	0

F. Penentuan nilai threshold

Nilai *threshold* akan ini akan digunakan di dalam FCBF. Setiap individu memiliki nilai *threshold* tertentu yang akan berimplikasi pada fitur FCBF yang terpilih untuk kemudian fitur tersebut digunakan sebagai input bagi KNN. Sebelum mendapatkan nilai *threshold*, terlebih dahulu dilakukan proses *decoding* melalui persamaan (11)

$$th = a + decimal(kromosom) \times \frac{(a+b)}{2^{m}-1} \qquad (11)$$

Dengan a dan b masing-masing batas bawah dan batas atas serta m adalah jumlah bit kromosom. Berikut ini adalah salah satu contoh *threshold* yang diperoleh

$$th = 0 + decimal(10001001011101110) \times \frac{(0+0.79621)}{2^{17}-1}$$

$$th = 0{,}42755$$

Hasil *threshold* tersebut akan digunakan sebagain input pada FCBF untuk mendapatkan fitur fitur relevan yang digunakan untuk proses klasifkasi pada KNN. Tabel 5 adalan contoh potongan hasil *threshold* dengan 5-mers dan 3-NN.

TABEL V
POTONGAN HASIL PERHITUNGAN *THRESHOLD* 5-MERS DAN 3-NN

	Ind.1	Ind/2	...	Ind.15
Threshold	0.09985	0.61575		0.60777
Fitur terpilih	4192	1689	...	1896
Akurasi	85.9375	89.0625		90.625
Fitness	83.0566	89.2197		90.0757

G. Seleksi dengan Roulette Wheel

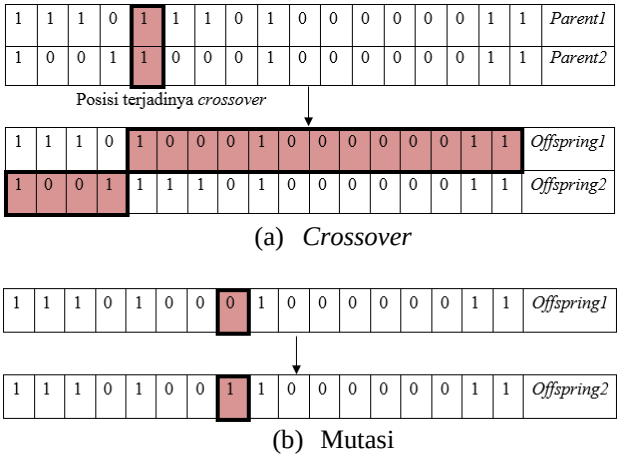
Hasil seleksi dengan menggunakan metode *roulette wheel* dengan pemilihan *parrent* berdasarkan nilai *fitness* yang paling baik. Adapun hasil proporsi kumulatif setiap individu disajikan pada Tabel 6.

TABEL VI
PROPORSI KUMUATIF SETIAP INDIVIDU

Individu	*Fitness*	Proporsi	Proporsi kumulatif
1	93.70	0.070	0.070
2	93.70	0.070	0.140
3	88.77	0.066	0.206
4	81.37	0.061	0.267
5	91.67	0.069	0.336
6	91.48	0.068	0.404
7	81.67	0.061	0.465
8	82.45	0.062	0.527
9	93.60	0.070	0.597
10	89.31	0.067	0.663
11	93.60	0.070	0.733
12	88.77	0.066	0.800
13	89.55	0.067	0.867
14	88.77	0.066	0.933
15	89.72	0.067	1.000

H. Crossover dan mutasi

Seperti layakanya dalam hal ilmu genetika, proses *crossover* dan mutasi terjadi antara individu. Hasil *crossover* dan mutasi ini ditunjukkan pada Gambar 4.



Gambar 4. Hasil Proses *Crossover* (a) dan Mutasi (b) pada Dua Individu

I. Hasil Optimasi Threshold

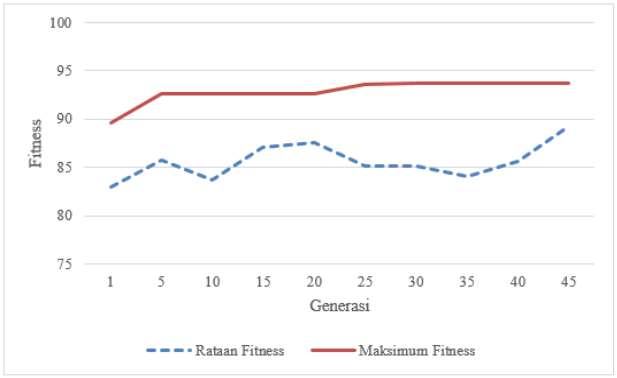
Untuk mendapatkan optimasi hasil *threshold* tersebut, dilakukan beberapa skenario percobaan. Terdapat empat skenario percobaan yang dilakukan sebagaimana disajikan pada Tabel 7.

TABEL VII
SKENARIO PERCOBAAN

	n-mers	**k-nn**
skenario 1	5-mers	3-nn
skenario 2	5-mers	5-nn
skenario 3	7-mers	3-nn
skenario 4	7-mers	5-nn

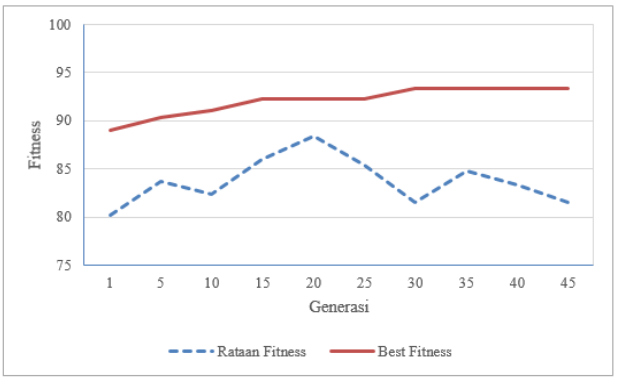
Setiap masing-masing skenario tersebut dilakukan iterasi sampai pada generasi tertentu yang dalam penelitian ini dilakukan sampai pada generasi ke 45. Untuk setiap generasi dihitung rata-rata nilai *fitness* dan nilai *fitness* maksimumnya. Hasil percobaan skenario 1 disajikan pada gambar 5.

Gambar 5 menunjukkan hasil optimasi nilai *threshold* pada 5-mers dengan nilai k = 3 pada klasifikasi KNN. Nilai rata-rata fitness tertinggi pada nilai 89.1%. Generasi pertama memiliki nilai *fitness* tertinggi sebesar 89.5% mengalami peningkatan secara terus menerus hingga nilai *fitness* yang tertinggi pada generasi ke-45 sebesar 93.70%.



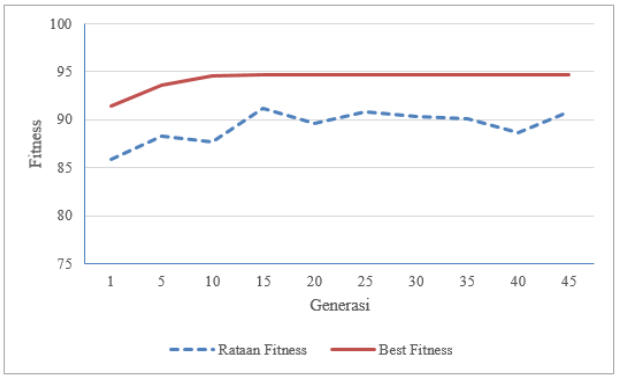
Gambar 5. Perolehan nilai *fitness* untuk 5-mers dan 3-nn

Percobaan pada skenario 2 disajikan pada Gambar 6.



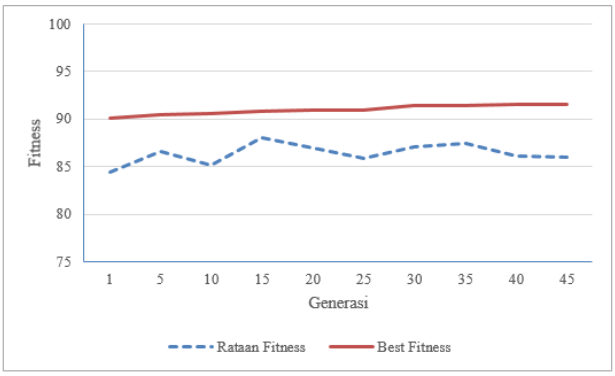
Gambar 6. Perolehan nilai *fitness* untuk 5-mers dan 5-nn

Pada percobaan skenario 2, nilai rata-rata akurasi tertinggi pada nilai 88.5%. Generasi pertama nilai *fitness* tertinggi sebesar 88.9% mengalami peningkatan secara terus menerus hingga nilai *fitness* tertinggi pada generasi ke-45 sebesar 93.3%. Hasil percobaan pada skenario 3 diperlihatkan pada Gambar 7.



Gambar 7. Perolehan nilai *fitness* untuk 7-mers dan 3-nn

Pada percobaan ini, generasi pertama nilai *fitness* tertinggi di atas 90 persen yaitu sebesar 91.4%. Secara iteratif mengalami peningkatan secara terus menerus seiring dengan bertambahnya generasi hingga nilai *fitness* tertinggi berhasil dicapai pada generasi ke-45 sebesar 94.6%. Adapun percobaan skenario ke-4 diperlihatkan pada Gambar 8.



Gambar 8. Perolehan nilai *fitness* untuk 7-mers dan 5-nn

Pada percobaan skenario 4 ini diperoleh nilai rata-rata *fitness* tertinggi pada nilai 88.6%. Pada generasi pertama memiliki nilai *fitness* tertinggi sebesar 90% mengalami peningkatan secara terus menerus hingga nilai *fitness* tertinggi pada generasi ke-45 sebesar 91.48%.

Apabila dilihat dari keempat skenario tersebut, terlihat bahwa secara umum terjadi tren peningkatan nilai *fitness* seiring dengan bertambahnya generasi, meskipun dalam setiap genrasi memang terjadi fluktuasi nilai *fitness*.

J. Analisis Performa

Di dalam klasifikasi *metagenome*, ketepatan hasil bining sangat penting. Untuk itulah pada penelitian ini diukur beberapa parameter untuk menilai performa dan pengklasifikasi yang digunakan dalam hal ini adalah KNN. Untuk meperlihatkan hasil tersebut, Tabel 8 dan Tabel 9 menyajikan performa dari KNN berdasarkan penggunaan 3 tetangga terdekat dan 5 tetangga terdekat yang dilakukan secara eksperimental.

TABEL VIII
PERFORMA HASIL KLASIFIKASI 3-NN

N-mers	*Threshold*	Fitur	3-NN		
			Akurasi	Sensitivitas	Spesifisitas
5-*mers*	-	1024	98.92%	85.93%	99.79%
	0.71934	66	**99.31%**	**93.75%**	99.79%
7-*mers*	-	16.384	99.02%	87.5%	99.79%
	0.67788	410	**99.41%**	**93.75%**	99.79%

Pada Tabel 8 terlihat bahwa penggunaan algoritme untuk optimasi nilai *threshold* FCBF mampu menseleksi fitur dan mereduksi jumlah fitur bahkan mampu untuk menghasilkan akurasi yang lebih baik dibandingkan dengan tanpa dilakukan proses optimasi parameter FCBF.

Jumlah fitur pada 5-mers yang semula 1026 menjadi 66 dengan meningkatkan akurasi menjadi sebesar 99.41%. Tidak hanya itu, berdasarkan hasil penelitian ini waktu komputasi untuk mendapatkan hasil akurasi ini juga mampu untuk diefisienkan. Penggunaan 1024 fitur, membutuhkan waktu komputasi sebanyak 5980 detik atau sekitar 1 jam 36 menit sedangkan dengan penggunaan 66 fitur hanya dibutuhkan waktu sebesar 0.473 detik.

Begitu juga pada fitur dengan 7-mers, optimasi dengan menggunakan GA pada ternyata mampu meningkatkan akurasi dan waktu komputasi pada saat proses perhitungan KNN. Berdasarkan penelitian ini, waktu komputasi dengan menggunaan 16.384 fitur membutuhkan waktu sebanyak 92.150 detik atau sekitar 24 jam lebih. Adapun setelah mengalami proses optimasi dengan algoritme genetika, waktu yang dibutuhkan sebesar 2600 detik atau sekitar 43 menit.

Sementara performa hasil klasifikasi dengan 5 tetangga terdekat disajikan pada Tabel 9.

TABEL IX
PERFORMA HASIL KLASIFIKASI 5-NN

N-mers	*Threshold*	Fitur	5-NN		
			Akurasi	sensitifitas	spesifisitas
5-*mers*	-	1024	99.02%	87.50%	99.79%
	0.73528	33	**99.31%**	**92.19%**	99.79%
7-*mers*	-	16384	98.92%	85.93%	99.79%
	0.69393	202	**99.12%**	**89.06%**	99.79%

Berdasarkan tabel 9 di atas penggunaan algoritme genetika ternyata juga dapat meningkatkan performasi hasil akurasi pada KNN dengan lima tetangga terdekat. Dari sisi waktu komputasi juga dapat dipastikan sangat berpengaruh secara signifikan.

Hasil penelitan menunjukkan bahwa penggunaan seluruh fitur pada 5-mers dengan lima tetangga terdekat ini mengkonsumi waktu sebanyak 5820 detik sementara dengan menggunakan 3 fitur mengkonsumi waktu hanya sebanyak 0.225 detik. Adapun pada 7-mers, waktu yang dibutuhkan untuk melakukan komputasi saat perhitungan dengan KNN adalah sebanyak 91.480 detik atau sekitar 25 jam. Sementara penggunaan 202 fitur hasil optimasi dengan GA hanya membutuhkan waktu sebanyak 1200 detik atau 20 menit.

Secara umum, nilai sensitifitas tertinggi pada 7-mers. Penggunaan n = 7 pada 7-mers akan menjadikan tingkat keunikan pada kombinasi basa nukleotida yang terbentuk sehingga secara teori harusnya memilki akurasi yang lebih baik. Hanya saja akan menjadi *trade-off* dengan tingginya dimensi fitur yang terbentuk. Untuk itulah proses optimasi dengan GA pada FCBF ini diperlukan.

IV. KESIMPULAN

Algoritm Genetika (GA) mampu memberikan hasil yang baik untuk melakukan optimasi parameter FCBF pada klasifikasi *metagenome*. Kemampuan GA ditunjukan dengan meningkatnya nilai akurasi dan sensitifitas serta mampu menjalankan algoritme KNN menjadi lebih efisien dalam proses komputasinya

REFERENSI

[1] Y.W. Wu, Y. Ye, A novel abundance-based algorithm for binning metagenomic sequences using l-tuples, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6044 LNBI (2010) 535–549. doi:10.1007/978-3-642-12683-3_35.

[2] T. Thomas, J. Gilbert, F. Meyer, Metagenomics - a guide from sampling to data analysis, Microb. Inform. Exp. 2 (2012) 3. doi:10.1186/2042-5783-2-3.

[3] S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, Informatica. 31 (2007) 249–268. doi:10.1115/1.1559160.

[4] M. Hajighorbani, S. Mohammad, R. Hashemi, A. Broumandnia, M. Faridpour, A Review of Some Semi-Supervised Learning Methods, J. Knowledge- Based Eng. Innov. 2 (2016) 250–259. http://aeuso.org/jkbei/wp-content/uploads/2016/06/27-A-Review-of-Some-Semi-Supervised-Learning-Methods.pdf.

[5] Z.-H. Zhou, A Brief Introduction to Weakly Supervised Learning, Natl. Sci. Rev. (2017) 44–53. doi:10.1093/nsr/nwx106.

[6] S. Higashi, A. Barreto, M. Cantão, A. de Vasconcelos, Analysis of composition-based metagenomic classification, BMC Genomics. 13 (2012) S1. doi:10.1186/1471-2164-13-S5-S1.

[7] X. (Sylvia) Zhu, M. McGee, Metagenomic Classification Using an Abstraction Augmented Markov Model, J. Comput. Biol. 23 (2015) cmb.2015.0141. doi:10.1089/cmb.2015.0141.

[8] W. Li, L. Fu, B. Niu, S. Wu, J. Wooley, Ultrafast clustering algorithms for metagenomic sequence analysis, Brief. Bioinform. 13 (2012) 656–668. doi:10.1093/bib/bbs035.

[9] W. Li, Analysis and comparison of very large metagenomes with fast clustering and functional annotation, BMC Bioinformatics. 10 (2009) 1–9. doi:10.1186/1471-2105-10-359.

[10] M.V. Overbeek, W.A. Kusuma, A. Buono, Clustering metagenome fragments using growing self organizing map, 2013 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2013. (2013) 285–289. doi:10.1109/ICACSIS.2013.6761590.

[11] N. Pookhao, M.B. Sohn, Q. Li, I. Jenkins, R. Du, H. Jiang, L. An, A two-stage statistical procedure for feature selection and comparison in functional analysis of metagenomes, Bioinformatics. 31 (2015) 158–165. doi:10.1093/bioinformatics/btu635.

[12] G. Ditzler, J.C. Morrison, Y. Lan, G.L. Rosen, Fizzy: Feature subset selection for metagenomics, BMC Bioinformatics. 16 (2015) 1–8. doi:10.1186/s12859-015-0793-8.

[13] A. Al-ajlan, Feature selection for gene prediction in metagenomic fragments, BioData Min. 11 (2018) 1–12. doi:10.1186/s13040-018-0170-z.

[14] L. Yu, H. Liu, Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution, Int. Conf. Mach. Learn. (2003) 1–8. doi:citeulike-article-id:3398512.

[15] A. Dinilhak, Klasifikasi fragmen metagenome menggunakan metode SVM dan fast correlation based filter sebagai penyeleksi fitur, Bogor Agricultural Universisty, 2015. https://repository.ipb.ac.id/handle/123456789/74945.

[16] D. Zeng, S. Wang, Y. Shen, C. Shi, A GA-based feature selection and parameter optimization for support tucker machine, Procedia Comput. Sci. 111 (2017) 17–23. doi:10.1016/j.procs.2017.06.004.

[17] G. Nagarajan, R.I. Minu, B. Muthukumar, V. Vedanarayanan, S.D. Sundarsingh, Hybrid Genetic Algorithm for Medical Image Feature Extraction and Selection, Procedia Comput. Sci. 85 (2016) 455–462. doi:10.1016/j.procs.2016.05.192.

[18] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, Z. Gao, A hybrid feature selection algorithm for gene expression data classification, Neurocomputing. 256 (2017) 56–62. doi:10.1016/j.neucom.2016.07.080.

[19] I.S. Karima, Optimasi parameter pada support vector machine untuk klasifikasi fragmen metagenome menggunakan algoritme genetika, Institut Pertanian Bogor, 2014. https://repository.ipb.ac.id/handle/123456789/71332.

[20] D.C. Richter, F. Ott, A.F. Auch, R. Schmid, D.H. Huson, MetaSim: A Sequencing Simulator for Genomics and Metagenomics, Handb. Mol. Microb. Ecol. I Metagenomics Complement. Approaches. 3 (2011) 417–421. doi:10.1002/9781118010518.ch48.

[21] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Comput. Stat. Data Anal. 24 (1996) 372–373. doi:10.1007/978-3-662-03315-9.