

Deteksi Kanker Berdasarkan Klasifikasi Data *Microarray* Menggunakan *Genetic Bee Colony* (GBC) dan *Artificial Neural Network* (ANN)

Tugas Akhir
diajukan untuk memenuhi salah satu syarat
memperoleh gelar sarjana
dari Program Studi Teknik Informatika
Fakultas Informatika
Universitas Telkom

1301144363
Melati Suci Pratiwi



Program Studi Sarjana Teknik Informatika
Fakultas Informatika
Universitas Telkom
Bandung

2018

LEMBAR PENGESAHAN

Deteksi Kanker Berdasarkan Klasifikasi Data *Microarray* Menggunakan *Genetic Bee Colony* (GBC) dan *Artificial Neural Network* (ANN)

Cancer Detection Based on Microarray Data Classification using Genetic Bee Colony (GBC) and Artificial Neural Network (ANN)

NIM: 1301144363

Melati Suci Pratiwi

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Teknik Informatika

Fakultas Informatika

Universitas Telkom

Bandung, 28 Juli 2018

Menyetujui

Pembimbing I

Pembimbing II

Prof. Dr. Adiwijaya, S.Si., M.Si.

NIP: 00740046

Annisa Aditsania, S.Si., M.Si.

NIP: 15900046

Ketua Program Studi
Sarjana Teknik Informatika,

Said Al Faraby, S.T., M.Sc.

NIP: 15890019

LEMBAR PERNYATAAN

Dengan ini saya, Melati Suci Pratiwi, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul "**Deteksi Kanker Berdasarkan Klasifikasi Data *Microarray* Menggunakan *Genetic Bee Colony* (GBC) dan *Artificial Neural Network* (ANN)**" beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika dikemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 28 Juli 2018

Yang Menyatakan,

Melati Suci Pratiwi

Deteksi Kanker Berdasarkan Klasifikasi Data *Microarray* Menggunakan *Genetic Bee Colony* (GBC) dan *Artificial Neural Network* (ANN)

Melati Suci Pratiwi¹, Adiwijaya², Annisa Aditsania³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹melatisp@student.telkomuniversity.ac.id, ²adiwijaya@telkomuniversity.ac.id,

³aaditsania@telkomuniversity.ac.id

Abstrak

Kanker merupakan salah satu masalah utama bagi kesehatan seluruh manusia di dunia ini. Oleh karena itu, pendeteksian kanker harus dilakukan sejak dini. Seiring berkembangnya teknologi, terdapat teknologi *microarray* yang dapat membantu peneliti untuk mengumpulkan informasi dari ribuan gen seorang manusia secara bersamaan yang nantinya akan berguna untuk pendeteksian penyakit kanker. Setiap fitur data *microarray* mempunyai dimensi yang tinggi sehingga dilakukan pemilihan dimensi untuk meningkatkan akurasi dari klasifikasi data *microarray* apakah seorang individu menderita kanker atau tidak menggunakan algoritma *Genetic Bee Colony* (GBC) dan *Artificial Neural Network-Conjugate Gradient Polak Ribiere* (ANN-CGP). GBC merupakan algoritma *hybrid* metaheuristik yang terinspirasi berdasarkan algoritma *Artificial Bee Colony* dan *Genetic Algorithm*. ANN-CGP merupakan modifikasi dari *Artificial Neural Network* (ANN) untuk mempercepat pelatihan *backpropagation*. Dengan implementasi GBC dan ANN-CGP sebagai seleksi fitur dan *classifier*, sistem yang dibangun mampu menyeleksi fitur hingga 49% untuk keseluruhan data serta meningkatkan akurasi klasifikasi sekitar 18-33% dibandingkan tanpa kombinasi dari GBC dan ANN-CGP.

Kata kunci : Data *Microarray*, *Genetic Bee Colony*, *Artificial Neural Network*, *Conjugate Gradient Polak Ribiere*, Deteksi Kanker, Kanker

Abstract

Cancer is one of the major problems for the health of all humans in the world. Therefore, the detection of cancer should be done early. As technology develops, there is a *microarray* technology that can help researchers to gather information from thousands of genes of a human being simultaneously which will be useful for the detection of cancer. Each feature of *microarray* data has a high dimension so that dimensional selection is done to improve the accuracy of the *microarray* data classification whether an individual has cancer or has not using *Genetic Bee Colony* (GBC) algorithm and *Artificial Neural Network-Conjugate Gradient Polak Ribiere* (ANN-CGP). GBC is a metaheuristic hybrid algorithm which inspired based on *Artificial Bee Colony* algorithm and *Genetic Algorithm*. ANN-CGP is a modification of *Artificial Neural Network* (ANN) to accelerate *backpropagation* training. By implementation of GBC and ANN-CGP as a selection of features and classifier, the system is capable of selecting features up to 49% for all datasets and increasing accuracy by 18-33% compared with no combination of GBC and ANN-CGP.

Keywords: Data *Microarray*, *Genetic Bee Colony*, *Artificial Neural Network*, *Conjugate Gradient Polak Ribiere*, Cancer, Cancer Detection

1. Pendahuluan

Pada bab pendahuluan ini terdapat beberapa sub-bab, diantaranya adalah sub-bab latar belakang yang memaparkan tentang alasan masalah terkait penelitian dan metode yang digunakan, sub-bab perumusan dan batasan masalah yang menjelaskan tentang batasan-batasan yang digunakan pada penelitian, sub-bab tujuan yang memaparkan tujuan dari penelitian, serta sub-bab organisasi penulisan yang berisi struktur penulisan dari jurnal ini.

Latar Belakang

Kanker merupakan suatu penyakit yang disebabkan oleh pertumbuhan sel-sel dalam tubuh secara abnormal atau tidak terkontrol, yang dicontohkan dengan pembentukan jaringan ekstra yang disebut dengan *mass* dan kemudian menyerang organ lain pada tubuh [9]. Berdasarkan pernyataan *World Health Organization* (WHO), kanker merupakan salah satu penyakit mematikan kedua di dunia karena menyebabkan 8,8 juta kasus kematian pada tahun 2015 dan secara global hampir satu dari enam manusia meninggal dunia karena penyakit kanker [17]. Dalam

proses diagnosa kanker, dokter masih menggunakan data rekam medis pasien-pasiennya. Metode ini dianggap kurang efektif karena merujuk kepada *human error prediction* dan berdasarkan ketidakpastian, dikarenakan data rekam medis seorang pasien dapat berubah-ubah seiring dengan perubahan gaya hidup pasien yang bersangkutan [19]. Berdasarkan kemajuan teknologi, saat ini terdapat teknologi biomedis yang berisi data-data 'OMIC' seperti genomik, transkriptomik, dan proteomik yang dikumpulkan dalam satu waktu yang bersamaan [24].

Teknologi *microarray* adalah teknologi yang dapat membantu peneliti untuk mengumpulkan informasi dari banyak ribuan gen dalam satu waktu yang sama [24] dan dapat membantu prediksi dokter terhadap pendiagnosaan penyakit kanker [18]. Setiap fitur data *microarray* dapat memiliki variabel (*probe gen*) hingga 450.000 dan pengolahan mayoritas fitur yang melibatkan proses komputasi yang tinggi [14]. Banyaknya variabel atau dimensi yang terdapat dalam data *microarray* akan meningkatkan kesulitan dalam proses pembuktian hasil (dalam hal ini diagnosa atau klasifikasi kanker itu sendiri). Oleh karena itu, dibutuhkan metode seleksi fitur dan klasifikasi untuk fitur data *microarray* sehingga dapat mendeteksi kanker. Secara umum, seleksi fitur dapat dikelompokkan menjadi tiga kategori yaitu: metode *filter*, *wrapper*, dan *embedded* [7]. Metode *wrapper* digunakan untuk masalah pemilihan subset fitur (gen) yang selanjutnya akan digunakan untuk proses pelatihan dan pengujian algoritma *classifier* yang digunakan. Menurut penelitian sebelumnya pada tahun 2015, dilakukan penelitian menggunakan *Artificial Bee Colony* (ABC) dan klasifier *Artificial Neural Network* (ANN) dalam hal ini menggunakan *Multilayer Perceptron* (MLP) yang menyimpulkan bahwa ABC merupakan kandidat yang kompetitif untuk mereduksi dimensi dari data *microarray* serta MLP mencapai akurasi tertinggi saat dilatih menggunakan gen-gen yang sudah tereduksi (dalam hal ini gen-gen yang terbaik) dan memiliki hasil performansi yang lebih baik dibandingkan *Radial Basis Function* (RBF) [10].

Pada penelitian ini, digunakan sebuah pendekatan *wrapper* dengan menggabungkan *Genetic Bee Colony* (GBC) untuk menyeleksi fitur dan *Artificial Neural Network* (ANN) dengan algoritma *learning backpropagation* untuk mengklasifikasi data *microarray*. Algoritma *Genetic Bee Colony* (GBC) dipilih sebagai seleksi fitur karena mencapai akurasi tertinggi dan rata-rata terendah dari pemilihan gen ketika diuji menggunakan enam *binary* dan *multi-class* dataset *microarray* dibandingkan menggunakan algoritma ABC asli dan metode mRMR-ABC menggunakan pendekatan *cross validation* yang sama [6]. Adapun ANN dipilih karena ANN dapat mengenali entitas biologis yang berbeda secara halus dan model komputasi yang luar biasa yang telah diterapkan untuk memecahkan berbagai jenis masalah [7] seperti klasifikasi pola, peramalan, dan masalah regresi [13]. Namun, pada pelatihan ANN standar menggunakan *backpropagation* ditemukan beberapa masalah yakni lamanya proses pelatihan apabila pemilihan parameternya tidak tepat, misalnya penentuan untuk *learning rate* yang mempengaruhi perubahan bobot [4]. Oleh karena itu, diperlukan algoritma yang dapat mempercepat pelatihan *backpropagation*, salah satunya adalah algoritma *Conjugate Gradient Polak-Ribiere*. Algoritma *Conjugate Gradient* merupakan metode optimasi yang dapat meminimasi suatu fungsi, dimana arah pencariannya berdasarkan arah konjugasi yang nilainya ortogonal [4]. Salah satu jenis metode *Conjugate Gradient* adalah *Polak Ribiere* yang dapat diimplementasikan sebagai algoritma *learning backpropagation* pada klasifikasi data *microarray*.

Perumusan dan Batasan Masalah

Rumusan masalah yang ada pada penelitian ini yaitu bagaimana menganalisis dan mengimplementasikan seleksi fitur pada data *microarray*, bagaimana membangun sistem yang mampu mendeteksi kanker dengan proses pengklasifikasian berdasarkan data *microarray*, bagaimana menganalisa performansi yang dihasilkan dari seleksi fitur dan *classifier* yang dibangun berdasarkan parameter-parameter yang digunakan. Adapun batasan masalah yang digunakan pada penelitian ini adalah data yang digunakan berjumlah tiga yaitu data *Colon Tumor*, *Leukemia (ALL-AML)* dan *Lung Cancer* yang diambil dari *Kent Ridge Biomedical Dataset Repository* yang dapat diakses melalui situs <http://leo.ugr.es/elvira/DBCRepository/>.

Tujuan

Berdasarkan perumusan masalah yang telah disebutkan sebelumnya, maka tujuan dari penelitian ini adalah mengaplikasikan *Genetic Bee Colony* (GBC) untuk menyeleksi fitur-fitur pada data *microarray*, membangun model klasifikasi dengan menggunakan metode *Artificial Neural Network* (ANN) termodifikasi *Conjugate Gradient Polak Ribiere* (CGP), menganalisis performansi dan efisiensi dari penggabungan kedua metode seleksi fitur dan klasifikasi *Genetic Bee Colony* (GBC) dan ANN-CGP berdasarkan parameter-parameter pada GBC dan ANN-CGP itu sendiri.

Organisasi Tulisan

Setelah bab Pendahuluan dipaparkan, bab selanjutnya pada jurnal ini adalah sebagai berikut. Bab Studi Terkait yang menjelaskan tentang studi komparatif terkait penelitian sebelumnya yang berkaitan dengan penelitian yang sedang dikerjakan serta beberapa teori dan penjelasan tentang metode yang diimplementasikan pada penelitian ini. Bab Pembangunan Sistem yang menjelaskan tentang sistem yang dibangun pada penelitian ini yaitu deteksi kanker berdasarkan klasifikasi data *microarray* menggunakan *Genetic Bee Colony* dan *Artificial Neural Network* termodifikasi *Conjugate Gradient Polak Ribiere* (CGP). Kemudian bab Evaluasi yang akan memaparkan hasil pengujian serta analisis hasil yang didapatkan dari pengujian tersebut. Selanjutnya bab Kesimpulan yang akan

menjelaskan tentang kesimpulan yang didapatkan dari penelitian ini.

2. Studi Terkait

Pada bab Studi Terkait ini terdapat dua sub-bab yaitu studi komparatif yang berkaitan dengan penelitian ini dan teori-teori tentang metode yang digunakan pada penelitian ini.

2.1 Studi Komparatif

Terhitung sejak sepuluh tahun terakhir, teknik-teknik pembelajaran mesin telah diimplementasikan untuk menganalisis data *microarray* seperti pada [3]. Beberapa pendekatan telah diimplementasikan untuk membedakan antara kanker dan non-kanker sampel serta mengklasifikasikan jenis kanker. Kedua pendekatan tersebut mempunyai satu tujuan yang sama yaitu menghasilkan interpretasi biologis bermakna dari data *microarray* yang memiliki fitur yang kompleks. Berdasarkan hal tersebut, pada penelitian klasifikasi data *microarray*, seperti pada [10], [7], [12] yang memakai ANN dalam hal ini menggunakan *Multilayer Perceptron* sebagai metode klasifikasi menyebutkan bahwa MLP mengungguli metode lainnya dengan menunjukkan performansi yang lebih baik. Terbukti pada dataset ALL-AML *Multilayer Perceptron* (MLP) menghasilkan akurasi sebesar 0,941 dibandingkan *Radial Basis Function* (RBF) yang hanya menghasilkan akurasi sebesar 0,559 dan 0,507 [10]. Pemilihan subset fitur-fitur gen menggunakan *Artificial Bee Colony* (ABC) juga sudah pernah dilakukan oleh beberapa peneliti, [10] melakukan perbandingan beberapa algoritma klasifikasi yang digabungkan dengan ABC, [5] yang menyimpulkan bahwa akurasi dari metode filter mRMR ketika dikombinasikan dengan ABC secara umum mengungguli akurasi klasifikasi dari algoritma ABC tanpa mRMR, dan pada [6] dilakukan beberapa modifikasi terhadap algoritma ABC seperti menambahkan *uniform crossover* sebagai step baru pada fase *onlooker* dan menambahkan jumlah *scout bee* menjadi dua dibandingkan pada ABC asli yang hanya menggunakan satu *scout bee* untuk meningkatkan kecepatan gerakan serta untuk mencapai keseimbangan antara eksploitasi dan eksplorasi diadopsikan operator mutasi dari *Genetic Algorithm* selama proses penggantian solusi *exhausted* pada fase *scout bee*. Algoritma GBC yang digunakan [6] dapat mengungguli hasil-hasil yang telah dilaporkan sebelumnya, misalnya saja pada data *Leukemia 1* menghasilkan akurasi terbaik dengan hasil 93,05% dibandingkan mRMR-ABC dan *Artificial Bee Colony* (ABC) sendiri yang hanya menghasilkan 91,66% dan 88,88%.

2.2 DNA Microarray

Teknologi *microarray* DNA dikembangkan untuk memungkinkan peneliti mengumpulkan sejumlah besar ekspresi gen pada saat yang bersamaan [24]. Saat ini, analisis dari data ekspresi gen telah menjadi topik utama di antara bioinformatik, biostatistik, dokter, dan ilmuwan. Hal ini dikarenakan analisis data ekspresi gen telah memungkinkan ditemukannya informasi tersembunyi yang memberikan pengetahuan biologis [20]. Data dari *microarray* terdiri dari banyak fitur dimana masing-masing fitur merepresentasikan sebuah ekspresi gen [11]. Data *point* pada data *microarray* merupakan data rekam medis seorang pasien. Kelas pada setiap ekspresi gen akan menjadi penentu untuk sistem klasifikasi yang dibangun. Masalah yang terdapat dalam data *microarray* adalah banyaknya fitur yang menyebabkan *curse of dimensionality*, dimana mengacu pada berbagai fenomena yang akan dihadapi saat menganalisis dan mengatur data pada ruang dimensi yang tinggi (yakni berkisar dari ratusan hingga ribuan dimensi). Oleh karena itu diperlukan adanya pemilihan dan klasifikasi gen untuk analisis dan interpretasi data *microarray* [20].

2.3 Genetic Bee Colony (GBC) Algorithm

GBC merupakan sebuah metode algoritma *hybrid* metaheuristik yang terinspirasi berdasarkan algoritma *Artificial Bee Colony* (ABC) dan *Genetic Algorithm* (GA) [6]. ABC sendiri terinspirasi dari perilaku *honeybee* ketika mencari sumber makanan yang berkualitas. Tujuan dari GBC adalah untuk memilih gen yang lebih informatif untuk mengoptimalkan akurasi dari *classifiers* yang digunakan. Pada GBC, GA diadopsi pada eksploitasi yang terdapat pada algoritma ABC pada fase *onlooker bee* untuk mengimprovisasi informasi yang dibagikan antara *employee bee* dan *onlooker bee* untuk menemukan solusi optimal dan pada fase *scout bee* untuk meningkatkan proses solusi yang terpakai [6]. Adapun fase-fase pada algoritma GBC dapat dilihat pada Gambar 2.

2.3.1 Fase Representasi dan Inisialisasi

Pada fase Representasi dan Inisialisasi, representasi dari *solution space* (*foods*) direpresentasikan dengan representasi biner di mana 0 merupakan fitur yang tidak terpilih dan 1 merupakan fitur yang terpilih. Setiap solusi

dinotasikan sebagai x_i dimana i merupakan solusi tertentu ($i = 1, 2, \dots, SN$) dan setiap solusi merupakan D -dimensi vektor $j = (1, 2, \dots, D)$ dimana D merepresentasikan bilangan dari gen informatif untuk optimasi disetiap solusi. Sehingga dapat dikatakan bahwa setiap x_{ij} adalah koresponding indeks gen. Jumlah nektar pada sumber makanan sesuai untuk nilai *fitness* dari asosiasi solusi pada algoritma ABC, dimana jumlah *employeeed bees* atau *onlooker bees* sama dengan jumlah solusi di populasi. Setiap x_{ij} diinisialisasikan secara acak sejumlah populasi yang akan digunakan dan jumlah *solution space* yang dipakai pada algoritma GBC.

2.3.2 Fase Employee Bee

Pada fase ini, *employee bees* dikirimkan untuk menjadi kandidat solusi (*food source*) dan mengevaluasi nilai *fitness*nya menggunakan *classifier* yang digunakan. Dengan demikian, *employee bee* mencari di sekitar solusi pada x_i yang akan mencari untuk solusi indeks gen yang lebih baik pada lokasi baru v_i yang didefinisikan seperti dengan Persamaan (1) [6]:

$$v_{ij} = x_{ij} + R_{ij}(x_{ij} - x_{kj}) \quad (1)$$

Dimana $v_{ij} = [v_{i1}, v_{i2}, \dots, v_{in}]$ representasi indeks gen baru (lokasi vektor dari *bee*), $x_{ij} = [x_{i1}, x_{i2}, \dots, x_{in}]$ adalah lokasi indeks gen saat ini (lokasi vektor dari ke- i *bee*), $k(k \neq j)$ adalah bilangan acak di $[1, SN]$, dan SN adalah jumlah solusi (*artificial bees*), R_{ij} adalah bilangan acak *uniform* yang didistribusikan pada $[0, 1]$. Bilangan acak x_{ij} dipilih dari indeks gen *microarray* yang telah di-generate pada fase Representasi dan Inisialisasi.

2.3.3 Fase Onlooker Bee

Pada GBC, operasi *crossover* digunakan untuk berbagi informasi antara *employee* dan *onlooker* bee pada ruang pencarian optimasi. *Onlooker bees* mempelajari lokasi dari solusi dengan melihat *waggle dance* dari *employee bee*. Pada GBC, *onlooker bees* menggunakan lokasi dari solusi dengan nilai *fitness* tertinggi (*Queen Bee*) pada fase *Employee* untuk pemilihan tetangga dimana pada ABC asli tidak menggunakan ini. Setiap *onlooker bee* memilih secara acak *employee bee* sebagai tetangga. GBC menggunakan *uniform crossover* untuk pemilihan tetangga *onlooker bees*, dimana *Queen Bee* dan pilihan secara acak diperoleh dari (1) dimana tetangga *employee bee* dipilih berdasarkan *rating fitness* dari *employee bees* yang digunakan untuk operasi *uniform crossover*. Pilihan tetangga secara acak bergantung pada nilai probabilitas pemenang yang sama dengan *Roulette Wheel* pada GA yang didefinisikan sebagai P_i dari pemilihan *food source* pada *onlooker bee* menggunakan (2) [6].

$$P_i = \frac{f_i}{\sum_{j=1}^{SN} f_j} \quad (2)$$

Dengan demikian D variabel acak dari distribusi *uniform crossover* $[0,1]$ dibuat. Setelah itu, digunakan *Crossover Probability Rate* (CPR) pada 0,6 yang merupakan sebuah parameter kontrol pada GBC yang sangat sensitif pada setiap perubahan. Pada setiap posisi, jika nilai dari string acak dibawah CPR, *OffSpring₁* mengambil indeks gen dari *Parent₁* dan *OffSpring₂* mengambil indeks gen dari *Parent₂*. Sebaliknya, *OffSpring₁* mengambil indeks gen dari *Parent₂* dan *OffSpring₂* yang diambil dari indeks gen *Parent₁*. Pada algoritma ABC asli, solusi baru dihasilkan dari solusi saat ini dan satu solusi acak sehingga solusi baru di dalam pencarian *space* yang terbesar, memaksimalkan perbedaan. GBC hanya mengizinkan *Queen Bee* sebagai solusi yang baru, sehingga mampu mereduksi perbedaan yang akan muncul dimana solusi yang baru dapat dihasilkan.

2.3.4 Fase Scout Bee

Pada algoritma ABC asli, jika nilai *fitness* diasosiasikan dengan sebuah solusi tidak mengimprovisasi untuk sebuah bilangan terbatas dari percobaan spesifik *employee bee* menjadi sebuah *scout* untuk sebuah nilai acak yang ditujukan untuk menemukan solusi baru. Dengan mempelajari algoritma ABC dari [8], didapatkan sebuah kekurangan untuk proses pencarian solusi, dimana setelah jumlah signifikan dari iterasi ketika solusi optimal hampir ditemukan, *scout bee* yang melakukan proses eksploitasi tidak digunakan. Untuk mencapai keseimbangan antara kemampuan eksploitasi dan eksplorasi dari algoritma ABC, GBC meningkatkan gerakan *scout bee* pada algoritma ABC asli dengan mengaplikasikan dua modifikasi yaitu: (a) meningkatkan jumlah dari *scout bee* dari satu menjadi dua untuk meningkatkan kecepatan gerakan dengan meningkatkan tingkat pergantian (b) yang mengadopsi operator mutasi dari GA selama proses dari penggantian solusi *exhausted* untuk mengimprovisasi proses eksploitasi pada tahap selanjutnya dari algoritma.

Scout bee pertama akan bekerja sebagaimana pada algoritma ABC, yang akan me-reset proses pencarian dan eksplorasi sebuah area solusi baru. Selanjutnya, sebuah operator mutasi akan mengambil posisi pada penempatan *scout bee* kedua, dimana kita mencari sebuah tempat di sekitar solusi nilai *fitness* tertinggi dihasilkan (*Queen Bee*).

Dengan demikian, parameter pada *Queen Bee* dimutasi dengan probabilitas kecil menggunakan (3). *Mutation Probability Rate* (MPR) diatur menjadi 0.01, yang juga sebuah parameter kontrol pada GBC yang sangat sensitif terhadap perubahan. Adapun perhitungan terhadap nilai MPR didefinisikan sebagai berikut [6]:

$$ScoutB_{ij} = QueenB_{ij} + R_{ij}(RandB_{ij} - QueenB_{ij}) \quad (3)$$

Dimana *QueenB* adalah solusi terbaik, *i* adalah indeks solusi ke-*i* dan proses mutasi diaplikasikan ke semua gen *j* di indeks-*i*, dimana *j* diantara [1 dan *D*], *RandB* adalah pemilihan solusi secara acak yang diperoleh menggunakan (1), dan *R_{ij}* adalah sebuah distribusi *uniform* acak pada [0, 1].

2.4 Artificial Neural Network-Conjugate Gradient Polak Ribiere (ANN-CGP)

Artificial Neural Network merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu [21]. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut *neuron*, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi [21]. Salah satu metode yang sering digunakan dalam pendekatan ANN adalah pelatihan *backpropagation*. Algoritma ini melakukan dua tahap perhitungan maju untuk menghitung galat antara keluaran aktual dan target; dan perhitungan mundur yang mempropagasikan balik galat tersebut untuk memperbaiki bobot-bobot sinaptik pada semua neuron yang ada [21]. Namun, *backpropagation* klasik sensitif terhadap *learning rate* dan *momentum rate* [15]. Sebagai contoh, nilai *learning rate* sangat penting dalam artian nilai yang terlalu kecil akan membuat konvergensi semakin lambat dan nilai yang terlalu besar akan membuat arah pencarian melonjak tidak stabil dan tidak pernah menyatu [15]. Oleh karena itu, diperlukan algoritma yang dapat mempercepat pelatihan *backpropagation*, salah satunya adalah algoritma *Conjugate Gradient Polak-Ribiere*. Algoritma *Conjugate Gradient* (CG) merupakan metode optimasi yang dapat meminimasi suatu fungsi, dimana arah pencariannya berdasarkan arah konjugasi yang nilainya ortogonal [4] sehingga menghasilkan konvergensi yang umumnya lebih cepat [22]. CG menggunakan vektor tidak nol yang ortogonal dan bebas linier [4][1][2]. Dua buah vektor *d_i* dan *d_j* dikatakan ortogonal (*G-conjugate*) jika perkalian dalamnya bernilai nol, dan dapat dituliskan sebagai berikut [4]:

$$d_i^T d_j = 0 \quad (4)$$

Salah satu jenis metode *Conjugate Gradient* adalah *Polak Ribiere* yang dapat diimplementasikan sebagai algoritma *learning backpropagation* pada klasifikasi data *microarray*. Sebelum masuk ke algoritma *Conjugate Gradient* (CG) harus ditentukan fungsi objektif yang akan dioptimasi [4]. Apabila CG digunakan sebagai algoritma pelatihan ANN, maka tujuannya adalah meminimasi *error* yang bergantung pada bobot-bobot yang menghubungkan antar neuron [4]. Fungsi objektif adalah fungsi *error* yaitu [4]:

$$f(w) = \frac{1}{2N} \sum_n \sum_j (t_{nj} - y_{nj}(w))^2 \quad (5)$$

Dimana *N* merupakan jumlah pola pada data *training*, *w* adalah matriks bobot, dan *t_{nj}* dan *y_{nj}(w)* adalah data *target* dan *output* neuron untuk *n* pola. CG merupakan metode untuk meminimasi fungsi turunan dengan menghitung pendekatan *w_{k+1}* secara iteratif berdasarkan [4]:

$$w_{k+1} = w_k + \alpha_k d_k \quad (6)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (7)$$

Dimana α dan β adalah parameter untuk menghindari konvergen lokal [4]. Dari penelitian [4] dan [16] langkah-langkah dalam algoritma ANN-CGP dapat dituliskan sebagai berikut:

1. Inisiasi nilai-nilai bobot sinaptik dengan pembangkitan bilangan nilai acak dengan interval nilai bebas.
2. Jika kondisi berhenti belum terpenuhi, lakukan langkah 2-13.
3. Setiap unit *input* menerima sinyal dan meneruskannya ke unit *hidden* di atasnya.
 - Perhitungan Maju
4. Menghitung semua *output* pada unit *hidden* $z_j (j = 1, 2, \dots, p)$.

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (8)$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (9)$$

5. Hitung semua keluaran di unit *output* $y_k (k = 1, 2, \dots, m)$.

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (10)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (11)$$

- Perhitungan Mundur

6. Hitung faktor kesalahan di unit *output* berdasarkan *error* pada nilai *actual* dan nilai prediksi (*output* dari unit *output*).

$$\sigma_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (12)$$

7. Hitung faktor kesalahan di unit *hidden* berdasarkan faktor kesalahan pada unit di atasnya [4].

$$\sigma_{net_j} = \sum_{k=1}^m \sigma_k w_{kj} \quad (13)$$

$$\sigma_j = \sigma_{net_j} f'(z_{net_j}) = \sigma_{net_j} z_j (1 - z_j) \quad (14)$$

8. Hitung *gradient* di unit *output* dari fungsi objektif yang sudah ditentukan.

$$g_{k+1} = \frac{1}{N} \sum_{n=1}^p \sigma_{nk} y_{nk} \quad (15)$$

9. Hitung *gradient* di unit *hidden*.

$$g_{j+1} = \frac{1}{N} \sum_{n=1}^p \sigma_{nj} z_{nj} \quad (16)$$

10. Hitung parameter β untuk semua neuron di unit *hidden* dan unit *output*. Karena pada penelitian ini menggunakan *Polak Ribiere* maka persamaannya adalah:

$$\beta_{k+1} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k} \quad (17)$$

Dimana β_{k+1} adalah nilai parameter β saat ini, g_{k+1} adalah *gradient* pada iterasi saat ini, dan g_k adalah *gradient* pada iterasi sebelumnya.

11. Hitung *direction* untuk semua neuron di unit *hidden* dan unit *output*.

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (18)$$

Dimana d_{k+1} adalah *direction* pada iterasi saat ini, g_{k+1} adalah *gradient* pada iterasi saat ini, β_k nilai parameter β pada iterasi sebelumnya, dan d_k *direction* pada iterasi sebelumnya. Untuk *direction* awal yaitu: $d_1 = -g_1$.

12. Hitung parameter α untuk semua neuron di unit *hidden* dan *output* yang merupakan besar langkah yang diambil untuk setiap *direction*, parameter ini dapat dicari dengan teknik *line search*.

- Perbaharui Bobot

13. Perbaharui bobot dengan menggunakan persamaan berikut:

$$w_{k+1} = w_k + \alpha_{k+1} d_{k+1} \quad (19)$$

Dimana w_{k+1} adalah bobot yang akan diperbaharui, w_k adalah bobot sebelumnya, α_{k+1} adalah nilai *alfa* saat ini, dan d_{k+1} merupakan *direction* pada iterasi saat ini.

2.5 K-Fold Cross Validation

Cross Validation adalah metode yang menggunakan data yang sama untuk melatih model yang akan dibangun guna mendapatkan perkiraan kesalahan prediksi yang lebih sedikit dibandingkan dengan estimasi secara langsung [23]. Ide dasarnya adalah membagi data pelatihan menjadi dua subset - satu subset yang digunakan untuk melatih aturan prediksi dan kemudian subset lainnya untuk menilai kesalahan prediksi [23]. Untuk menggunakan data secara efisien, *cross validation* ini diulang dengan beberapa pemecahan terhadap data yang digunakan [23].

Teori statistik dan simulasi telah menemukan bahwa *cross validation* bukanlah perkiraan kesalahan prediksi yang baik untuk setiap jenis prediktor [23]. Selain dikarenakan masalah dalam menentukan jumlah kluster, ada pula dalam beberapa kasus tidak ada metode yang efisien secara komputasional [23]. *K-Fold Cross Validation* membagi seluruh dataset menjadi k himpunan bagian yang setiap iterasinya sebagian digunakan sebagai data validasi dan sebagian lainnya digunakan sebagai data pelatihan [23]. Sebuah teori statistik menunjukkan bahwa pilihan k yang tepat bergantung pada n dan jenis prediktornya [23]. Namun, dalam prakteknya dependensinya lemah dan $k = 10$ bekerja dengan baik untuk berbagai ukuran data dan dalam masalah yang bervariasi [23].

3. Pembangunan Sistem

Pada bagian ini, pembangunan sistem memuat beberapa sub bab yang berkaitan dengan sistem yang dibangun baik secara umum maupun secara detail.

3.1 Gambaran Umum Sistem

Secara umum, data *microarray* yang digunakan untuk sistem terlebih dahulu dinormalisasi sehingga menghasilkan data baru hasil normalisasi data. Kemudian, *k-fold cross validation* diimplementasikan pada data hasil normalisasi tersebut sehingga data terbagi menjadi dua yakni data *training* dan data *testing*. Setelah data sudah terbagi, sistem melakukan seleksi fitur menggunakan algoritma *Genetic Bee Colony*. Data hasil seleksi fitur selanjutnya digunakan untuk tahap klasifikasi ANN-CGP dimana hasil akurasi dari klasifikasi akan menjadi nilai *fitness* sebagai parameter selanjutnya hingga mendapatkan akurasi tertinggi dari sistem yang dibangun. Setelah akurasi didapatkan, karena menggunakan *k-fold cross validation* maka dilakukan iterasi sebanyak k lalu hasil akurasi yang telah didapatkan dihitung rata-ratanya dengan membaginya dengan sejumlah k yang digunakan pada sistem dan akurasi sistem secara keseluruhan pun akan diperoleh. Gambar 1 merupakan representasi gambaran umum sistem yang dibangun.

3.2 Dataset

Langkah awal dalam perancangan sistem adalah menentukan data apa saja yang ingin digunakan pada sistem, sehingga dilakukan pengumpulan data *microarray* tentang berbagai jenis data penyakit kanker. Adapun data yang digunakan dalam penelitian ini adalah data *Colon Tumor*, *Leukemia ALL-AML* dan *Lung Cancer* yang diperoleh dari *Kent Ridge Biomedical Dataset Repository* yang dapat diakses melalui situs <http://leo.ugr.es/elvira/DBCRepository>. Tabel 1 merepresentasikan karakteristik untuk masing-masing *dataset* yang digunakan:

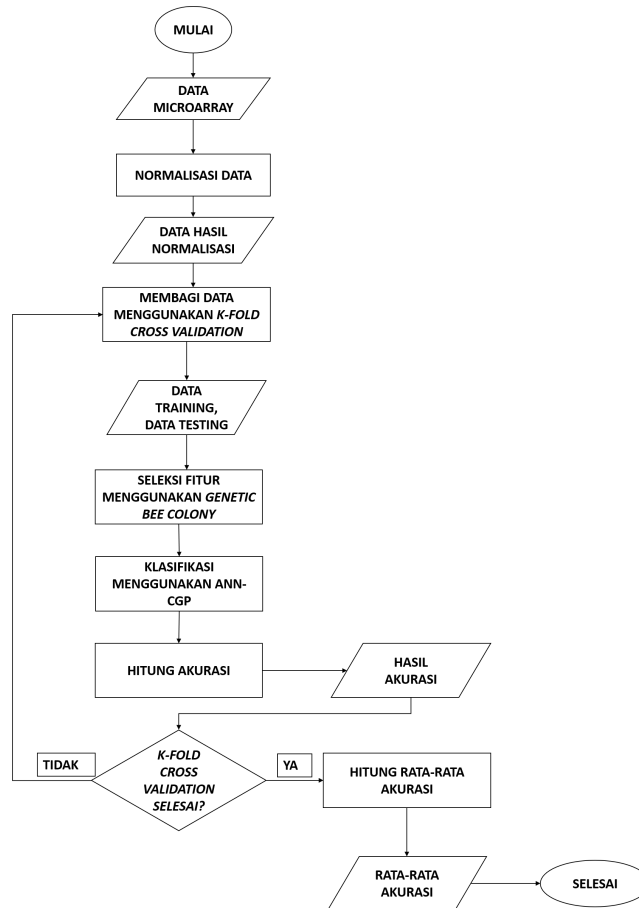
Tabel 1. Deskripsi *Dataset*

<i>Dataset</i>	Jumlah Fitur	Jumlah Sampel	Kelas	Kelas 1	Kelas 0
<i>Colon Tumor</i>	2.000	62	2	Positive	Negative
<i>Leukemia ALL-AML</i>	7.130	38	2	ALL	AML
<i>Lung Cancer</i>	12.534	32	2	ADCA	Mesothelioma

Adapun pembagian data *training* dan data *testing* adalah dengan menggunakan *k-fold cross validation* yang akan dilanjutkan pada sub bab selanjutnya.

3.3 Normalisasi

Data *microarray* memiliki jangkauan nilai antar fitur yang sangat jauh, hal ini dikarenakan dimensi yang ada pada data *microarray* sangat banyak. Oleh karena itu, sebelum klasifikasi dilakukan nilai antar fitur pada data harus diubah ke dalam jangkauan nilai yang lebih kecil. Normalisasi adalah suatu metode yang dapat diimplementasikan pada data yang memiliki jangkauan nilai yang berbeda, sehingga jangkauan nilai pada suatu data berada pada



Gambar 1. Gambaran Umum Sistem

interval yang sama. Dalam penelitian ini data awal akan diubah ke dalam interval $[0, 1]$ dengan menggunakan metode *min-max normalization* berikut:

$$d' = \frac{d - \min_A}{\max_A - \min_A} \quad (20)$$

Dimana, d merupakan data awal, \min_A dan \max_A merupakan nilai minimum dan maksimum dari data awal.

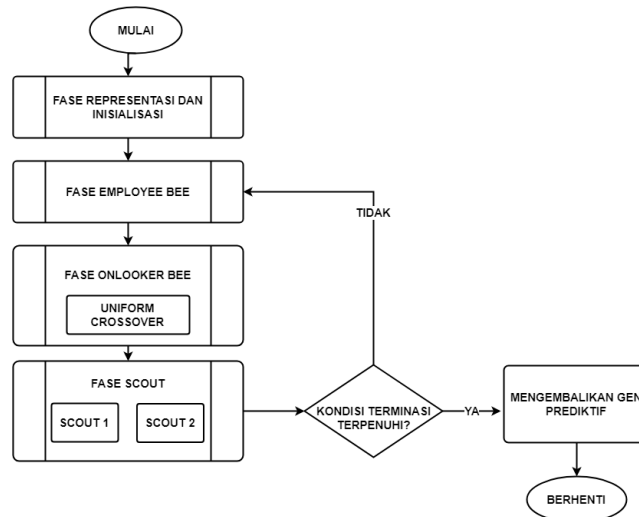
3.4 Validasi Sistem menggunakan *K-Fold Cross Validation*

Untuk melakukan validasi terhadap sistem yang dibangun digunakan *k-fold cross validation*. Dimana pada penelitian ini digunakan $k = 5$ yang nantinya akan menjadi acuan k subset dalam pembagian data. Apakah data tersebut akan menjadi data *training* atau data *testing*. Perbandingan 80 : 20 digunakan untuk pembagian data, dimana data yang dipakai dibagi 20% dari jumlah data awal. Skema *k-fold cross validation* ini dimulai dengan pembagian untuk data *testing* terlebih dahulu pada bagian pertama lalu bagian sisanya akan menjadi data *training*. Setelah itu, hal tersebut diulang-ulang sejumlah k subset yang digunakan sehingga pada data ke-lima pun akan bergantian menjadi data *testing* (dalam penelitian ini digunakan $k = 5$).

3.5 Seleksi Fitur menggunakan *Genetic Bee Colony* (GBC)

Seleksi fitur pada penelitian ini adalah dengan memilih fitur mana saja yang memberikan akurasi tertinggi pada *classsifier* yang digunakan. *Genetic Bee Colony* (GBC) digunakan untuk menyeleksi fitur yang mempunyai nilai *fitness* tertinggi (dalam hal ini akurasi) atau merupakan solusi terbaik yang selanjutnya disebut dengan *Queen Bee*. Gambar 2 merupakan fase-fase pada algoritma GBC.

Untuk alur proses lebih *detail* dapat dilihat pada Gambar 5 sampai dengan Gambar 8 di bagian Lampiran. Pada seleksi fitur ini tahap pertama yang dilakukan adalah menentukan paramater yang diperlukan dalam algoritma GBC seperti: populasi dari *bee colony*, *maximum cycle* yang sama dengan maksimum jumlah generasi, nilai *limit* yang menunjukkan jumlah iterasi yang diperbolehkan ketika calon solusi tidak diimprovisasi (jika calon solusi melebihi



Gambar 2. Fase utama dari algoritma *Genetic Bee Colony* (GBC)
[6]

Tabel 2. Inisial solusi pada sistem

Fitur	Bee ke-1	Bee ke-2	Bee ke-3	Bee ke-4	Bee ke-n
x1	1	0	1	1	1
x2	0	1	0	0	0
x3	1	1	0	1	1
...
x-n	1	0	0	1	0

batas *limit* maka ia akan dijadikan *scout bee*), nilai *Crossover Probability Rate* (CPR) dan *Mutation Probability Rate*. Pada tahap kedua dilakukan inisial solusi yang terdiri dari sejumlah *bee*. Setiap *bee* direpresentasikan dengan representasi biner yakni 0 dan 1. Setiap *bee* memiliki nilai *fitness* (akurasi yang didapatkan dari hasil klasifikasi ANN-CGP, dihitung dengan menggunakan Persamaan 21 dan 22) yang selanjutnya akan menjadi hasil akurasi fase 1 dan nilai *fitness* tertinggi pada fase ini akan menjadi *queenBee*. Adapun representasi calon solusi dinyatakan pada Tabel 2:

$$\text{if } \text{hasilTes} == \text{targetTes} \text{ then } \text{jumlahBenar} + 1 \quad (21)$$

$$\text{HasilAkurasi} = \frac{\text{jumlahBenar}}{\text{jumlahTotalData}} \times 100\% \quad (22)$$

Pada langkah ketiga, selama *maximum cycle* belum terpenuhi maka dilakukan evaluasi *fitness* per-*bee*. Pada tahap ketiga ini, dipilih tetangga secara acak setelah itu akan dilakukan *update* posisi bilamana *bee* yang ada pada tahap kedua bernilai 0. Setelah solusi hasil *update* didapatkan, akurasi dihitung kembali. Bila akurasi dari solusi *update* lebih besar dibandingkan akurasi solusi tahap kedua maka solusi tahap kedua akan digantikan dengan solusi *update* dan *abondementCounter* (sebagai pembanding untuk nilai *limit*) akan di-*reset* menjadi 0, jika tidak lebih besar maka *abondementCounter* akan ditambah dengan 1 dan kita mendapatkan *bestFitness* sementara dari tahap ketiga. Lalu dilakukan perbandingan apakah *bestFitness* pada tahap ketiga ini lebih besar dari hasil *queen bee* tahap kedua maka *queenBee* akan digantikan dengan *bestFitness* tahap ketiga yang selanjutnya menjadi hasil *queenBee* pada fase 2, bila tidak lebih besar maka *queenBee* tetap.

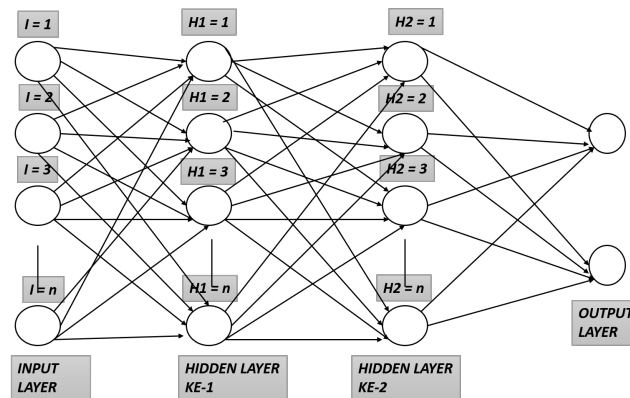
Pada langkah ke-empat selama *maximum cycle* belum terpenuhi maka dilakukan pemilihan *employee bee* dengan nilai *fitness* tertinggi dan dipilih tetangga secara acak yang selanjutnya akan digunakan untuk operasi *crossover*. Pada operasi *crossover*, akan dihitung akurasi per *OffSpring1* dan *OffSpring2* yang selanjutnya akan dipilih akurasi yang lebih besar untuk dijadikan *new solution*. Apabila akurasi dari *new solution* lebih besar dibandingkan *old solution* (sebelum masuk ke operasi *crossover*) maka *old solution* akan digantikan dengan *new solution* dan *abondementCounter* akan menjadi 0, jika tidak *abondementCounter* akan ditambah dengan 1 dan akan dihasilkan *bestFitness* fase 3. Lalu dilakukan perbandingan apakah *bestFitness* pada tahap ke-empat ini lebih besar dari hasil *queen bee* tahap ketiga maka *queenBee* akan digantikan dengan *bestFitness* tahap ke-empat yang selanjutnya menjadi hasil *queenBee* pada fase 3, bila tidak lebih besar maka *queenBee* tetap.

Pada tahap ke-lima, bila *abondementCounter* lebih besar dari batas nilai *limit* dan selama *maximum cycle* belum terpenuhi maka akan dilakukan dua pencarian solusi: dimana pada pencarian pertama, solusi akan di-*generate* secara random lalu dilakukan penghitungan akurasi terhadap solusi tersebut. Pada pencarian kedua, solusi akan dimutasikan dengan *queenBee* yang ada bila bilangan *random* yang kita tentukan lebih kecil dari MPR lalu dilakukan perhitungan akurasi terhadap hasil mutasi tersebut. Bila akurasi pada pencarian pertama lebih besar dari pencarian kedua maka solusi pada pencarian pertama akan menjadi *new solution*, bila tidak maka solusi pada pencarian kedua-lah yang menjadi *new solution*. Didapatkan hasil *bestFitness* fase 4 yang selanjutnya akan dibandingkan dengan *bestFitness* fase 3, bila *bestFitness* pada tahap ke-empat ini lebih besar dari hasil *queen bee* tahap ketiga maka *queenBee* akan digantikan dengan *bestFitness* tahap ke-empat yang selanjutnya menjadi hasil *queenBee* pada fase 4, bila tidak lebih besar maka *queenBee* tetap.

Langkah-langkah tersebut diulang kembali hingga mencapai *maximum cycle* yang telah ditentukan serta menghasilkan calon solusi (*bee*) dengan nilai *fitness* terbaik.

3.6 Klasifikasi menggunakan Artificial Neural Network-Conjugate Gradient Polak Ribiere (ANN-CGP)

Pada penelitian ini digunakan arsitektur ANN-CGP dengan dua lapisan *hidden layer* seperti Gambar 3.



Gambar 3. Arsitektur ANN-CGP

Dimana, dari neuron *input* ke neuron *hidden* merupakan sinyal yang datang dari neuron *input*. Lalu, pada kedua *hidden layer* digunakan fungsi aktivasi Sigmoid pada Persamaan 9 dan 11. Setelah itu, diaplikasikan CGP untuk optimasi algoritma *backpropagation* pada ANN yang diimplementasikan pada sistem. Adapun proses CGP dapat dihitung dari Persamaan 8 sampai dengan Persamaan 18 dan direpresentasikan dengan Gambar 4. Gambar 4 merupakan proses pelatihan CGP yang diimplementasikan pada satu *hidden layer*.

Pada klasifikasi menggunakan ANN-CGP langkah awal yang dilakukan adalah inisialisasi bobot, maksimum *epoch*, *learning rate*, *Mean Squared Error* (MSE). Setelah inisialisasi selesai, langkah kedua yang dilakukan adalah perhitungan maju menggunakan Persamaan 9, Persamaan 10, dan Persamaan 11. Setelah perhitungan maju dilakukan, langkah ketiga adalah menghitung MSE. Bila nilai MSE lebih besar dari maksimum MSE yang ditentukan dan nilai *epoch* lebih kecil dari maksimum *epoch* yang ditentukan maka lakukan perhitungan mundur dengan menghitung: faktor kesalahan menggunakan Persamaan 12, Persamaan 13, dan 14, menghitung *gradient* dengan Persamaan 15 dan Persamaan 16, menghitung parameter β dengan menggunakan Persamaan 17, menghitung *direction* dengan menggunakan Persamaan 18 dan yang terakhir menghitung parameter α .

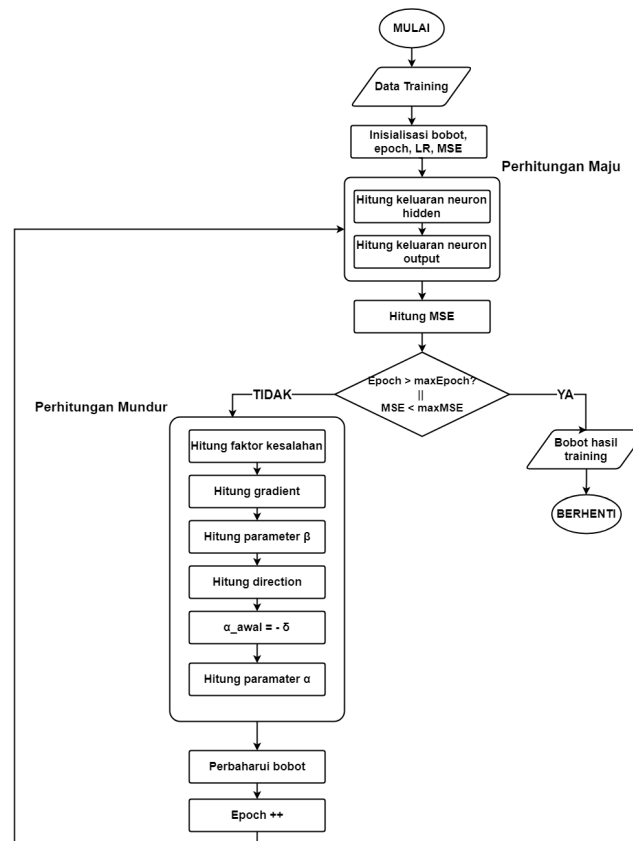
Setelah perhitungan maju dan mundur dilakukan (*backpropagation*) bobot diperharui dan nilai *epoch* bertambah. Proses tersebut akan terus berulang sampai dengan nilai MSE lebih kecil dari maksimum MSE yang ditentukan dan nilai *epoch* lebih besar dari maksimum *epoch* yang ditentukan.

4. Evaluasi

Pada bagian ini akan dimuat beberapa sub bab yang meliputi hasil pengujian dan analisis pengujian.

4.1 Hasil Pengujian berdasarkan parameter dari Genetic Bee Colony (GBC)

Pada pengujian ini, diuji beberapa paramater yang telah digunakan pada algoritma GBC yakni nColony (jumlah populasi) sebesar 10, 20 dan 30 serta maxCycle (jumlah generasi) sebesar 60, 80, dan 100. Adapun arsitektur



Gambar 4. Proses algoritma CGP
[4]

ANN-CGP yang digunakan adalah jumlah epoch sebesar 10.000 dan untuk kedua *hidden layer* digunakan 25 neuron pada *hidden layer* pertama serta 10 neuron pada *hidden layer* kedua. Jumlah *k-fold* yang digunakan adalah $k = 5$. Hasil yang diperoleh akan dipaparkan pada sub bab selanjutnya, dimana untuk setiap pengujian menggunakan *nColony* dan *maxCycle* fase yang paling berpengaruh terhadap perubahan akurasi adalah fase *Employee* dan fase *Onlooker bee*.

4.1.1 Menggunakan data *Colon Tumor* dengan uji parameter *nColony*

Tabel 3. Hasil pengujian parameter *nColony* pada data *Colon Tumor*

<i>Colon Tumor</i> (2000 fitur)	Jumlah <i>nColony</i> yang diuji		
	10	20	30
Σ Fitur <i>QueenBee</i>	1018	1004	1014
Seleksi (%)	50,9	50,2	50,7
Akurasi (%)	88,33	90	90

Dengan menggunakan algoritma GBC untuk menyeleksi fitur pada data *Colon Tumor* dapat dilihat pada Tabel 3 hasil pengujian menggunakan parameter *nColony* = 10, 20, dan 30 menghasilkan hasil yang bervariasi. Dimana, fitur *queenBee* paling sedikit terdapat pada *nColony* = 20 dengan akurasi sebesar 90% dan fitur *queenBee* paling banyak terdapat pada *nColony* = 10 dengan akurasi hanya 88,33%. Dari hasil yang didapatkan meskipun telah menambahkan jumlah populasi itu tidak berarti mengurangi jumlah fitur *queenBee* yang dihasilkan pada data *Colon Tumor*. Hal ini dikarenakan pencarian solusi dilakukan secara *random* sehingga tidak bisa mendapatkan jumlah fitur *queenBee* secara pasti (dalam artian jumlah fitur yang nantinya menjadi *queenBee*), terbukti saat dilakukan penambahan populasi sebanyak 10 pada *nColony* = 20, seleksi fitur mengalami penurunan sekitar 0,7 dan mengalami kenaikan sebesar 0,5 saat populasi ditambahkan kembali sebanyak 10 *bee* pada *nColony* = 30. Akurasi yang dihasilkan saat melakukan penambahan populasi pun tidak naik secara signifikan untuk setiap *nColony* yang diuji, dapat dilihat saat menggunakan *nColony* sebanyak 10 dan 20 akurasi naik sekitar 1,67 dan cenderung stagnan pada

saat menggunakan nColony sebanyak 20 dan 30 dengan arsitektur ANN-CGP serta jumlah generasi (*maxCycle*) yang sama. Adapun akurasi yang stagnan ini menandakan untuk data *Colon Tumor* dengan nColony sebanyak 20 dan 30 sudah mencapai akurasi yang paling optimal yang mampu dihasilkan oleh sistem yang dibangun, adapun untuk pembuktiannya bisa dengan menambahkan jumlah generasi yang ada pada sistem.

4.1.2 Menggunakan data *Leukemia ALL-AML* dengan uji parameter nColony

Tabel 4. Hasil pengujian parameter nColony pada data *Leukemia ALL-AML*

<i>Leukemia ALL-AML</i> (7130 fitur)	Jumlah nColony yang diuji		
	10	20	30
Σ Fitur QueenBee	3602	3591	3535
Seleksi (%)	50,5	50,3	49,5
Akurasi (%)	100	100	100

Dengan menggunakan algoritma GBC untuk menyeleksi fitur pada data *Leukemia ALL-AML* dapat dilihat pada Tabel 4 hasil pengujian menggunakan parameter nColony = 10, 20, dan 30 menghasilkan hasil yang bervariasi. Dimana, fitur *queenBee* paling sedikit terdapat pada nColony = 30 dengan akurasi sebesar 100% dan fitur *queenBee* paling banyak terdapat pada nColony = 10 dengan akurasi 100%. Dari hasil yang didapatkan dengan melakukan penambahan populasi (dalam hal ini *bee*) untuk setiap pengujiannya, penurunan fitur *queenBee* pun dapat terlihat dan menunjukan bahwa seiring bertambahnya jumlah populasi, jumlah fitur yang paling optimal pada data *Leukemia ALL-AML* berkurang. Dimana pada saat menggunakan jumlah nColony sebanyak 10 dan 20 fitur *queenBee* mengalami penurunan sebanyak 11 fitur dan mengalami penurunan sebanyak 56 fitur pada nColony 20 dan 30 dengan arsitektur ANN-CGP serta jumlah generasi (*maxCycle*) yang sama. Adapun hasil terbaik dan fitur *queenBee* paling sedikit dihasilkan menggunakan jumlah nColony = 30 dengan jumlah fitur *queenBee* = 3535 dan akurasi 100%. Untuk akurasi sendiri, setiap penambahan populasi pun cenderung tidak naik bahkan stagnan karena sudah mencapai hasil yang optimal yaitu 100%.

4.1.3 Menggunakan data *Lung Cancer* dengan uji parameter nColony

Tabel 5. Hasil pengujian parameter nColony pada data *Lung Cancer*

<i>Lung Cancer</i> (12534 fitur)	Jumlah nColony yang diuji		
	10	20	30
Σ Fitur QueenBee	6301	6249	6327
Seleksi (%)	50,2	49,8	50,4
Akurasi (%)	100	100	100

Dengan menggunakan algoritma GBC untuk menyeleksi fitur pada data *Lung Cancer* dapat dilihat pada Tabel 5 hasil pengujian menggunakan parameter nColony = 10, 20, dan 30 menghasilkan hasil yang bervariasi. Dimana, fitur *queenBee* paling sedikit terdapat pada nColony = 20 dengan akurasi sebesar 100% dan fitur *queenBee* paling banyak terdapat pada nColony = 30 dengan akurasi sebesar 100%. Dari hasil yang didapatkan meskipun telah menambahkan jumlah populasi itu tidak berarti mengurangi jumlah fitur *queenBee* yang dihasilkan pada data *Lung Cancer*, terbukti saat dilakukan penambahan populasi sebanyak 10 *bee* pada nColony = 20, seleksi fitur mengalami penurunan sekitar 0,4 dan mengalami kenaikan sebesar 0,6 saat populasi ditambahkan kembali sebanyak 10 pada nColony = 30. Akurasi yang dihasilkan saat melakukan penambahan populasi pun tidak naik cenderung stagnan untuk setiap nColony yang diuji, dapat dilihat saat menggunakan nColony sebanyak 10, 20, dan 30 akurasi tetap sama menggunakan arsitektur ANN-CGP serta jumlah generasi (*maxCycle*) yang sama. Hal ini dikarenakan akurasi yang telah didapatkan sudah optimal yakni sebesar 100%.

Sehingga dapat diambil beberapa benang merah dalam hal pengujian terhadap parameter nColony ini yakni, akurasi untuk data *Leukemia ALL-AML* dan data *Lung Cancer* mencapai akurasi maksimum untuk beberapa pengujian yang dilakukan. Dari hasil pengujian, untuk skema pengujian berdasarkan parameter GBC nColony seiring bertambahnya jumlah populasi (nColony) jumlah fitur *queenBee* pada data *Leukemia ALL-AML* cenderung menurun, namun hal ini tidak berlaku untuk data *Colon Tumor* dan *Lung Cancer* yang mengalami kenaikan serta penurunan jumlah fitur *queenBee* yang didapatkan. Dari sisi akurasi pun penambahan jumlah populasi (nColony) menaikkan akurasi pada data *Colon Tumor* sebanyak 1,67, namun tidak untuk kedua data lainnya. Hal ini dikarenakan kedua data tersebut sudah mencapai akurasi optimum yakni 100%.

4.1.4 Menggunakan uji parameter *maxCycle* untuk keseluruhan data

Selain menggunakan parameter uji nColony, penelitian ini juga melakukan pengujian terhadap parameter *maxCycle* sebesar 60, 80, dan 100. Adapun grafik untuk setiap pengujiannya dapat dilihat pada Gambar 9, Gambar 10, dan Gambar 11. Dimana pada Gambar 9 terlihat perbandingan akurasi dalam skala (%) untuk data *Colon Tumor* yang cenderung menurun lalu naik, namun kenaikan akurasi ini tidak jauh berbeda dengan pengujian sebelumnya. Akurasi untuk data *Leukemia ALL-AML* dan *Lung Cancer* pun cenderung stagnan karena sudah mencapai hasil akurasi yang optimal yakni sebesar 100%. Hal ini menandakan penambahan jumlah *maxCycle* tidak terlalu mempengaruhi akurasi yang dihasilkan untuk data *Lung Cancer* dan *Leukemia ALL-AML*, namun mempengaruhi data *Colon Tumor* yang mengalami kenaikan dan penurunan yang menjadi parameter bahwa pada saat menggunakan *maxCycle* = 80 akurasi optimal yang mampu dihasilkan oleh sistem adalah sebesar 88,33.

Pada Gambar 10 terlihat perbandingan jumlah fitur *queenBee* yang dihasilkan untuk setiap pengujiannya. Dimana untuk data *Colon Tumor* dan *Leukemia ALL-AML* jumlah fitur *queenBee* untuk setiap *maxCycle* nya cenderung menurun meskipun penurunan jumlah fitur *queenBee* tidak terlalu besar. Sedangkan untuk data *Lung Cancer* jumlah fitur *queenBee* cenderung menaik untuk setiap pengujiannya. Hal ini menandakan bahwa penambahan jumlah *maxCycle* mempengaruhi jumlah fitur *queenBee* yang dihasilkan untuk data *Colon Tumor* dan *Leukemia ALL-AML*, namun tidak mempengaruhi data *Lung Cancer*.

Pada Gambar 11 terlihat perbandingan jumlah seleksi fitur dalam skala (%) untuk setiap pengujiannya. Dimana untuk data *Colon Tumor* dan *Leukemia ALL-AML* terjadi penurunan jumlah seleksi fitur, terbukti penurunan sebesar 1,2 untuk seluruh pengujian data *Colon Tumor* sedangkan untuk data *Leukemia ALL-AML* terjadi penurunan sebesar 0,4 untuk seluruh pengujiannya. Namun, untuk data *Lung Cancer* mengalami kenaikan seleksi fitur sebanyak 0.6 untuk seluruh pengujiannya. Sehingga penambahan jumlah *maxCycle* mempengaruhi jumlah fitur *queenBee* yang dihasilkan untuk data *Colon Tumor* dan *Leukemia ALL-AML*, namun tidak mempengaruhi data *Lung Cancer*. Adapun untuk setiap pengujian menggunakan nColony dan *maxCycle* fase yang paling berpengaruh terhadap perubahan akurasi adalah fase *Employee* dan fase *Onlooker bee*.

4.2 Hasil Pengujian berdasarkan arsitektur *Artificial Neural Network-Conjugate Gradient Polak Ribiere* (ANN-CGP)

Tabel 6. Hasil pengujian ANN-CGP tanpa seleksi fitur

<i>Dataset</i>	Jumlah fitur	Akurasi (%)
Colon Tumor	2000	70
Leukemia ALL-AML	7130	74,28
Lung Cancer	12534	66,67

Pada pengujian ini, ANN-CGP diujikan berdasarkan variasi arsitektur pada kedua *hidden layer* yaitu pada *hidden layer* pertama yang berjumlah 30 neuron dan 50 neuron, sedangkan pada *hidden layer* kedua diuji menggunakan 15, 20, dan 25 neuron. Adapun nilai epoch yang digunakan sebesar 10.000 dengan menggunakan nColony = 20 dan *maxCycle* sebesar 40. Jumlah *k-fold* yang digunakan adalah $k = 5$. Pada skenario pengujian ini pun dilakukan pengujian terhadap data asli (tanpa seleksi fitur GBC) dan arsitektur ANN-CGP serta jumlah k yang telah ditentukan sebelumnya.

4.2.1 Pengujian ANN-CGP tanpa seleksi fitur

Berikut merupakan pengujian ANN-CGP menggunakan data asli tanpa seleksi fitur dengan menggunakan jumlah $k = 5$ yang dapat dilihat pada Tabel 6.

Berdasarkan Tabel 6, dapat dilihat rata-rata akurasi untuk ketiga data yang digunakan sudah melebihi 50%, hal ini dapat dijadikan parameter bahwa ANN-CGP cukup direkomendasikan dalam melakukan proses klasifikasi menggunakan data *microarray* secara khususnya.

4.2.2 Menggunakan data *Colon Tumor*

Dengan menggunakan klasifikasi *Artificial Neural Network-Conjugate Gradient Polak Ribiere* (ANN-CGP) untuk menjadi *classifier* pada data *Colon Tumor* dapat dilihat pada Tabel 7 terdapat dua pengujian berbeda yakni saat menggunakan $h_1 = 30$ dan $h_1 = 50$, di mana untuk h_2 digunakan 15, 20, dan 25 neuron sebagai parameter pengujian terhadap arsitektur ANN-CGP. Dimana saat menggunakan $h_1 = 30$, terlihat jelas bahwa fitur *queenBee*

Tabel 7. Hasil pengujian *hidden layer* ke-1 = 30 dan 50 data *Colon Tumor*

<i>Colon Tumor</i> (2000 fitur)	Jumlah neuron <i>Hidden Layer</i> ke-1					
	$h_1 = 30$			$h_1 = 50$		
	15	20	25	15	20	25
Σ Fitur <i>QueenBee</i>	995	1005	1015	1011	1036	1028
Seleksi (%)	49,7	50,2	50,7	50,5	51,8	51,4
Akurasi (%)	88,33	88,33	90	90	88,33	88,33

terus bertambah seiring penambahan jumlah *hidden layer* ke-2, adapun jumlah fitur bertambah untuk setiap pengujian h_2 pada penelitian ini berjumlah 10 (nilai 10 ini bisa berubah-ubah bergantung dengan proses pelatihan dan pengujian ANN-CGP ini sendiri). Akurasi terbaik yang dihasilkan pada $h_1 = 30$ dan $h_2 = 25$ adalah 90% yang naik sekitar 1,67 dari akurasi sebelumnya saat menggunakan $h_2 = 20$ dan $h_2 = 15$, namun cenderung stagnan saat $h_2 = 20$ karena akurasi yang dihasilkan sama dengan akurasi pada $h_2 = 15$. Hal ini menandakan bahwa meskipun jumlah neuron pada h_2 bertambah tidak menjamin hasil yang lebih baik dari parameter uji h_2 sebelumnya (dalam hal seleksi fitur maupun hasil akurasinya).

Untuk $h_1 = 50$ dengan jumlah neuron pada h_2 15, 20, dan 25 akurasi yang dihasilkan cenderung menurun atau stagnan serta fitur *queenBee* yang dihasilkan pun cukup bervariasi. Dimana saat menggunakan $h_2 = 25$ fitur *queenBee* menurun yakni sekitar 8 fitur dan menaik saat menggunakan $h_2 = 20$ dengan bertambahnya fitur sebanyak 25 fitur. Akurasi yang stagnan menandakan untuk data *Colon Tumor* dengan $h_1 = 30$ dan $h_2 = 15$ dan $h_2 = 20$ sudah mencapai akurasi optimal yang mampu dihasilkan oleh sistem yang dibangun, begitupula dengan $h_1 = 50$ dengan $h_2 = 20$ dan $h_2 = 25$. Hal ini menandakan bertambahnya jumlah h_1 pun tidak menjamin bahwa hasil yang didapatkan lebih baik dari parameter uji sebelumnya yakni $h_1 = 30$ (dalam hal seleksi fitur maupun akurasinya).

4.2.3 Menggunakan data *Leukemia ALL-AML*

Tabel 8. Hasil pengujian *hidden layer* ke-1 = 30 dan 50 data *Leukemia ALL-AML*

<i>Leukemia ALL-AML</i> (7130 fitur)	Jumlah neuron <i>Hidden Layer</i> ke-1					
	$h_1 = 30$			$h_1 = 50$		
	15	20	25	15	20	25
Σ Fitur <i>QueenBee</i>	3568	3588	3508	3565	3540	3574
Seleksi (%)	50,0	50,3	49,2	50,0	49,6	50,1
Akurasi (%)	100	100	100	100	100	100

Dengan menggunakan klasifikasi *Artificial Neural Network-Conjugate Gradient Polak Ribiere* (ANN-CGP) untuk menjadi *classifier* pada data *Leukemia ALL-AML* dapat dilihat pada Tabel 8 terdapat dua pengujian berbeda yakni saat menggunakan $h_1 = 30$ dan $h_1 = 50$, di mana untuk h_2 digunakan 15, 20, dan 25 neuron sebagai parameter pengujian terhadap arsitektur ANN-CGP. Dimana saat menggunakan $h_1 = 30$, dengan $h_2 = 25$ fitur *queenBee* yang digunakan paling sedikit dibandingkan parameter h_2 lainnya yakni 3508 dengan akurasi sebesar 100%. Saat penambahan jumlah neuron pada h_2 terjadi penurunan dan kenaikan seleksi fitur, terbukti kenaikan terjadi saat $h_2 = 15$ dan $h_2 = 20$ naik sekitar 0,3 dan penurunan seleksi terjadi saat $h_2 = 25$ yakni selisih 1,1 dari parameter h_2 sebelumnya. Akurasi yang didapatkan pun cenderung stagnan untuk setiap pengujian parameter h_2 karena sudah mencapai akurasi maksimal yakni 100%. Hal ini menandakan bahwa dengan diubahnya parameter h_2 tidak menjamin pula parameter tersebut lebih baik dari parameter uji h_2 sebelumnya (dalam hal seleksi fitur maupun akurasinya).

Untuk $h_1 = 50$ dengan jumlah neuron pada h_2 15, 20, dan 25 akurasi yang dihasilkan cenderung stagnan serta fitur *queenBee* yang dihasilkan pun cukup bervariasi. Dimana saat menggunakan $h_2 = 15$ fitur *queenBee* menurun yakni sekitar 25 fitur dan menaik saat menggunakan $h_2 = 25$ dengan bertambahnya fitur sebanyak 34 fitur. Hal ini menandakan bertambahnya jumlah h_1 pun tidak menjamin bahwa hasil yang didapatkan lebih baik dari parameter uji sebelumnya yakni $h_1 = 30$ (dalam hal seleksi fitur maupun akurasinya).

4.2.4 Menggunakan data *Lung Cancer*

Dengan menggunakan klasifikasi *Artificial Neural Network-Conjugate Gradient Polak Ribiere* (ANN-CGP) untuk menjadi *classifier* pada data *Lung Cancer* dapat dilihat pada Tabel 9 terdapat dua pengujian berbeda yakni

Tabel 9. Hasil pengujian *hidden layer* ke-1 = 30 dan 50 data *Lung Cancer*

<i>Lung Cancer</i> (12534 fitur)	Jumlah neuron <i>Hidden Layer</i> ke-1					
	$h_1 = 30$			$h_1 = 50$		
	15	20	25	15	20	25
Σ Fitur <i>QueenBee</i>	6204	6278	6234	6255	6357	6169
Seleksi (%)	49,4	50,0	49,7	49,9	50,7	49,2
Akurasi (%)	100	100	100	100	100	100

saat menggunakan $h_1 = 30$ dan $h_1 = 50$, di mana untuk h_2 digunakan 15, 20, dan 25 neuron sebagai parameter pengujian terhadap arsitektur ANN-CGP. Dimana saat menggunakan $h_1 = 30$, dengan $h_2 = 15$ fitur *queenBee* yang digunakan paling sedikit dibandingkan parameter h_2 lainnya yakni 6204 dengan akurasi sebesar 100%. Saat penambahan jumlah neuron pada h_2 terjadi penurunan dan kenaikan seleksi fitur, terbukti kenaikan terjadi saat $h_2 = 15$ dan $h_2 = 20$ naik sekitar 0,6 dan penurunan seleksi terjadi saat $h_2 = 25$ yakni selisih 0,3 dari parameter h_2 sebelumnya. Akurasi yang didapatkan pun cenderung stagnan untuk setiap pengujian parameter h_2 karena sudah mencapai akurasi maksimal yakni 100%. Hal ini menandakan bahwa dengan diubahnya parameter h_2 tidak menjamin pula parameter tersebut lebih baik dari parameter uji h_2 sebelumnya (dalam hal seleksi fitur maupun akurasinya).

Untuk $h_1 = 50$ dengan jumlah neuron pada h_2 15, 20, dan 25 akurasi yang dihasilkan cenderung stagnan serta fitur *queenBee* yang dihasilkan pun cukup bervariasi. Dimana saat menggunakan $h_2 = 25$ fitur *queenBee* menurun yakni sekitar 228 fitur dan menaik saat menggunakan $h_2 = 20$ dengan bertambahnya fitur sebanyak 102 fitur. Hal ini menandakan bertambahnya jumlah h_1 pun tidak menjamin bahwa hasil yang didapatkan lebih baik dari parameter uji sebelumnya yakni $h_1 = 30$ (dalam hal seleksi fitur maupun akurasinya).

Dari beberapa hasil pengujian yang telah dipaparkan, dapat diambil benang merah untuk skema pengujian berdasarkan parameter arsitektur ANN-CGP *hidden layer* yang menghasilkan dengan penambahan jumlah neuron pada *hidden layer₂* dan *hidden layer₁* pun tidak menjamin bahwa akurasi maupun jumlah fitur *queenBee* akan naik secara kontinu, hal ini berlaku untuk data *Colon* yang mengalami penurunan akurasi saat jumlah neuron h_1 ditambahkan. Adapun akurasi terbaik terdapat pada (*Leukemia ALL-AML* dan *Lung Cancer* dimana seleksi fitur yang dihasilkan mencapai 49,2% dengan menggunakan $h_1 = 30$ dan $h_2 = 25$ serta $h_1 = 50$ dan $h_2 = 25$).

5. Kesimpulan

Skema pengujian pada penelitian ini dibagi menjadi dua yakni menggunakan parameter observasi pada GBC (dalam hal ini nColony dan maxCycle) serta menggunakan parameter arsitektur pada ANN-CGP. Dari penelitian yang dilakukan, performansi sistem yang dibangun untuk ketiga data yang digunakan (tanpa seleksi fitur) memperoleh akurasi tertinggi sebesar 74,26%, sedangkan hasil akurasi menggunakan klasifikasi ANN-CGP berkisar di antara 88,33-100%.

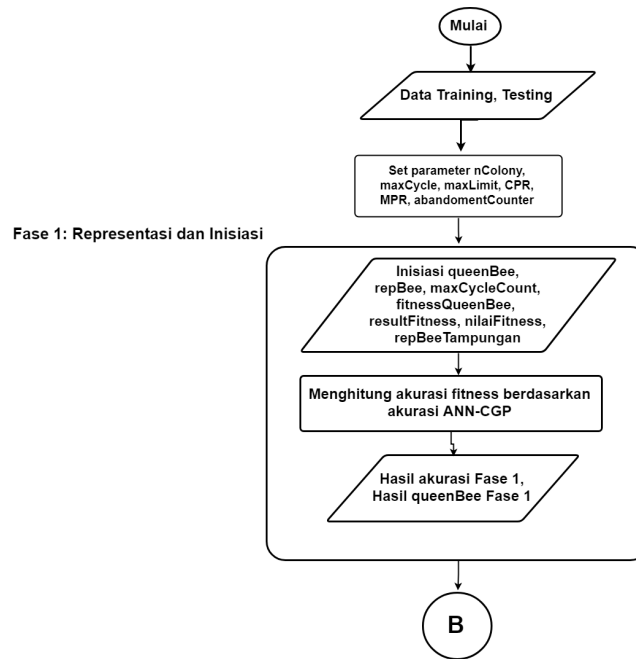
Sehingga dapat disimpulkan kombinasi GBC dan ANN-CGP memiliki performansi (dalam hal ini akurasi) yang lebih baik sekitar 18-33% dibandingkan tanpa mengkombinasikan ANN-CGP dengan GBC. Serta dengan kombinasi ini, metode yang digunakan mampu menyeleksi fitur hingga 49% untuk data *Colon Tumor* dengan mengubah parameter observasi di sisi GBC dan mampu menyeleksi fitur hingga 49,2% untuk data *Leukemia ALL-AML* dan data *Lung Cancer* dengan menggunakan perubahan arsitektur dari sisi ANN-CGP. Adapun saran untuk penelitian selanjutnya adalah sebaiknya lakukan perbandingan teknik klasifikasi lain dengan metode seleksi fitur GBC setelah itu kombinasikan metode seleksi fitur lainnya dengan *classifier* ANN-CGP untuk melihat kelebihan dari masing-masing metode, lalu lakukan beberapa perubahan pada parameter yang digunakan untuk GBC dan ANN-CGP untuk melihat pengaruhnya terhadap perubahan akurasi sistem yang dibangun.

Daftar Pustaka

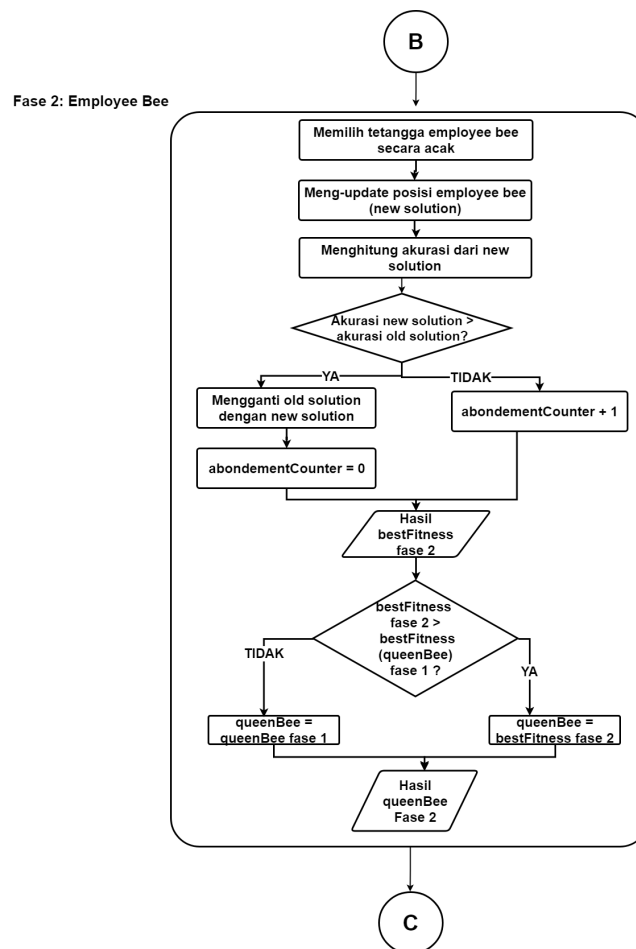
- [1] Adiwijaya. *Aplikasi Matriks dan Ruang Vektor*. Yogyakarta: Graha Ilmu, 2014.
- [2] Adiwijaya. *Matematika Diskrit dan Aplikasinya*. Bandung: Alfabeta, 2016.
- [3] Adiwijaya, U. N. Wisesty, E. Lisnawati, A. Aditsania, and D. S. Kusumo. Dimensionality reduction using principal component analysis for cancer detection based on microarray data classification. *JCS*, 2018.

- [4] Adiwijaya, U. N. Wisesty, T. A. B. Wirayuda, Z. K. A. Baizal, and U. Haryoko. An improvement of backpropagation performance by using conjugate gradient on forecasting of air temperature and humidity in indonesia. *Far East J. Math. Sci*, 1:57–67, 2013.
- [5] H. Alshamlan, G. Badr, and Y. Alohal. *mRMR-ABC: A Hybrid Gene Selection Algorithm for Cancer Classification Using Microarray Gene Expression Profiling*. Hindawi Publishing Corporation, 2015.
- [6] H. M. Alshamlan, G. H. Badr, and Y. Alohal. Genetic bee colony (gbc) algorithm: A new gene selection method for microarray cancer classification. *Computational Biology and Chemistry*, 56:49–60, 2015.
- [7] R. Aziz, C. Verma, and M. Jha. *Artificial Neural Network Classification of Microarray Data using New Hybrid Gene Selection Method*, volume 17. 2017.
- [8] N. Bacanin and M. Tuba. Artificial bee colony (abc) algorithm for constrained optimization improved genetic operators. *Studies in Informatics and Control*, 21:137–146, 2012.
- [9] C. de Martel, J. Ferlay, and S. Franceschi. *Global Burden of Cancers Attributable to Infections in 2008: A Review and Synthetic Analysis*. 2nd International Symposium on Big Data and Cloud Computing (ISBCC 2015), VIT University, Chennai Campus, 2012.
- [10] B. A. Garro, K. Rodriguez, and R. A. Vazquez. *Classification of DNA microarrays using Artificial Neural Networks and ABC Algorithm*. Elsevier: Applied Soft Computing, 2015.
- [11] A. H and Adiwijaya. A clustering approach for feature selection in microarray data classification using random forest. *Journal of Information Processing Systems*, 14, 2018.
- [12] S. E. J and N. Chandrakar. *Artificial Neural Networks as Classification and Diagnostic Tools for Lymph Node-negative Breast Cancers*, volume 33. The Korean Institute of Chemical Engineers, 2016.
- [13] A. Jain, J. Mao, and K. Mohiuddin. *Artificial Neural Networks: A Tutorial*, volume 29. IEEE, 1996.
- [14] S. Y. Kung and M. W. Mak. *Machine Learning in Bioinformatics, Chapter 1: Feature Selection for Genomic and Proteomic Data Mining*. g, John Wiley Sons, Hoboken, NJ, USA, 2009.
- [15] C. Man-Chung, W. Chi-Cheong, and L. Chi-Chung. Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization. *Computing in Economics and Finance*, 61, 2000.
- [16] A. Nurfalah, Adiwijaya, and A. A. Suryani. Cancer detection based on microarray data classification using pca and modified back propagation. *Far East Journal of Electronics and Communications*, 2015.
- [17] W. H. Organization. Cancer. <http://www.who.int/en/news-room/fact-sheets/detail/cancer>, 2018. Online; Accessed 14 Juli 2018.
- [18] P. T. Ramadhani, U. N. Wisesty, and A. Aditsania. Deteksi kanker berdasarkan klasifikasi data microarray menggunakan functional link neural network dengan seleksi fitur genetic algorithm. *Indonesian Journal on Computing (Indo-JC)*, 2017.
- [19] M. Sarkar and T.-Y. Leong. *Application of KNearest Neighbors Algorithm on Breast Cancer Diagnosis Problem*. AMIA Annual Symposium Proceedings Archive, 2000.
- [20] T. C. Siang and T. W. Soon. A review of cancer classification software for gene expression data. *International Journal of Bio-Science and Bio-Technology*, 7:89–108, 2015.
- [21] Suyanto. *Artificial Intelligence: Searching, Reasoning, Planning, Learning (Revisi Kedua)*. Penerbit INFORMATIKA, 2014.
- [22] N. N. Toolbox. Conjugate gradient algorithms. <http://radio.feld.cvut.cz/matlab/toolbox/nnet/backpr58.html>.
- [23] T. P. S. University. Cross-validation. <https://onlinecourses.science.psu.edu/stat555/node/118/>.
- [24] W.-K. Yip, S. B. Amin, and C. Li. *A Survey of Classification Techniques for Microarray Data Analysis*. Springer Handbook of Statistical Bioinformatics. Heidelberg, Berlin, 2011.

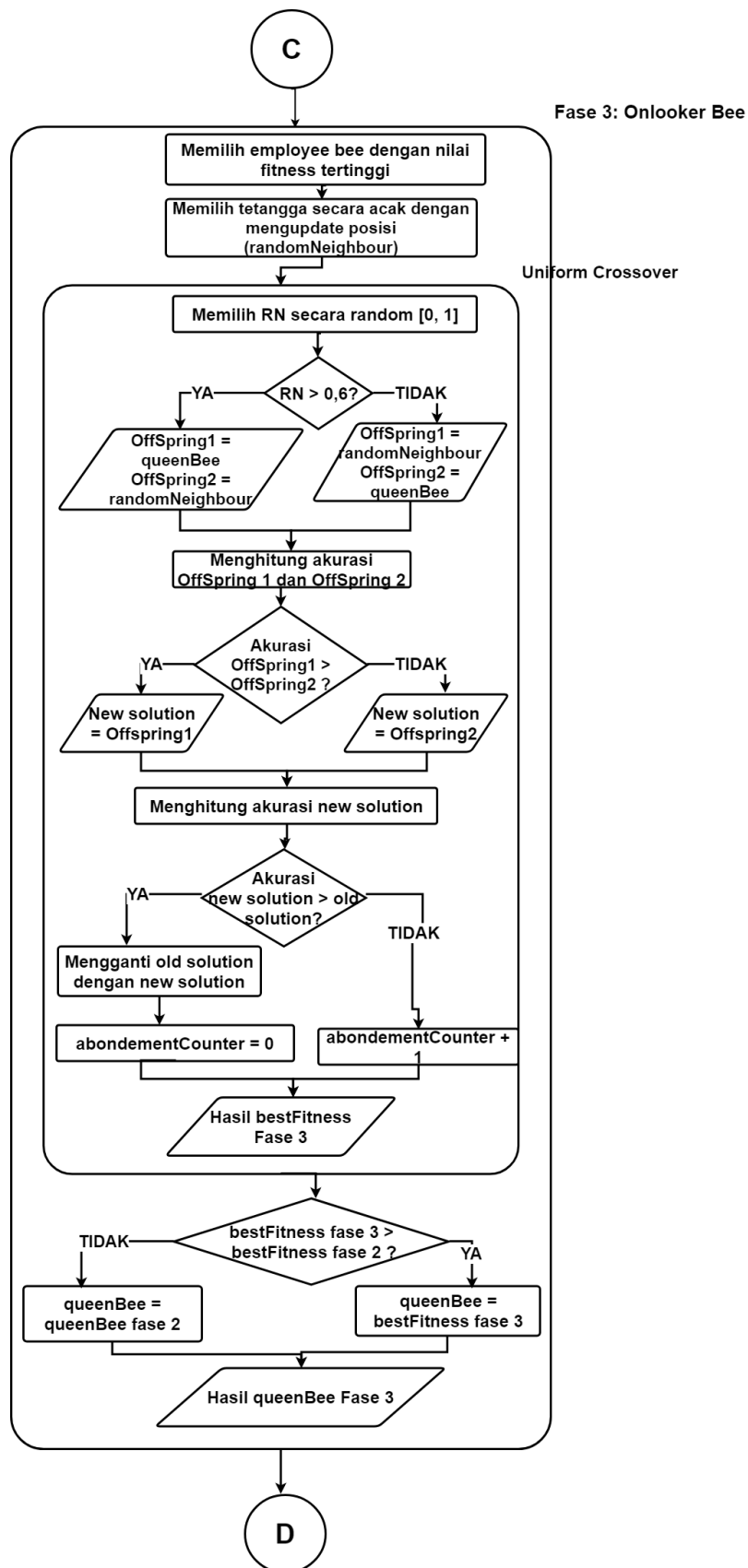
Lampiran



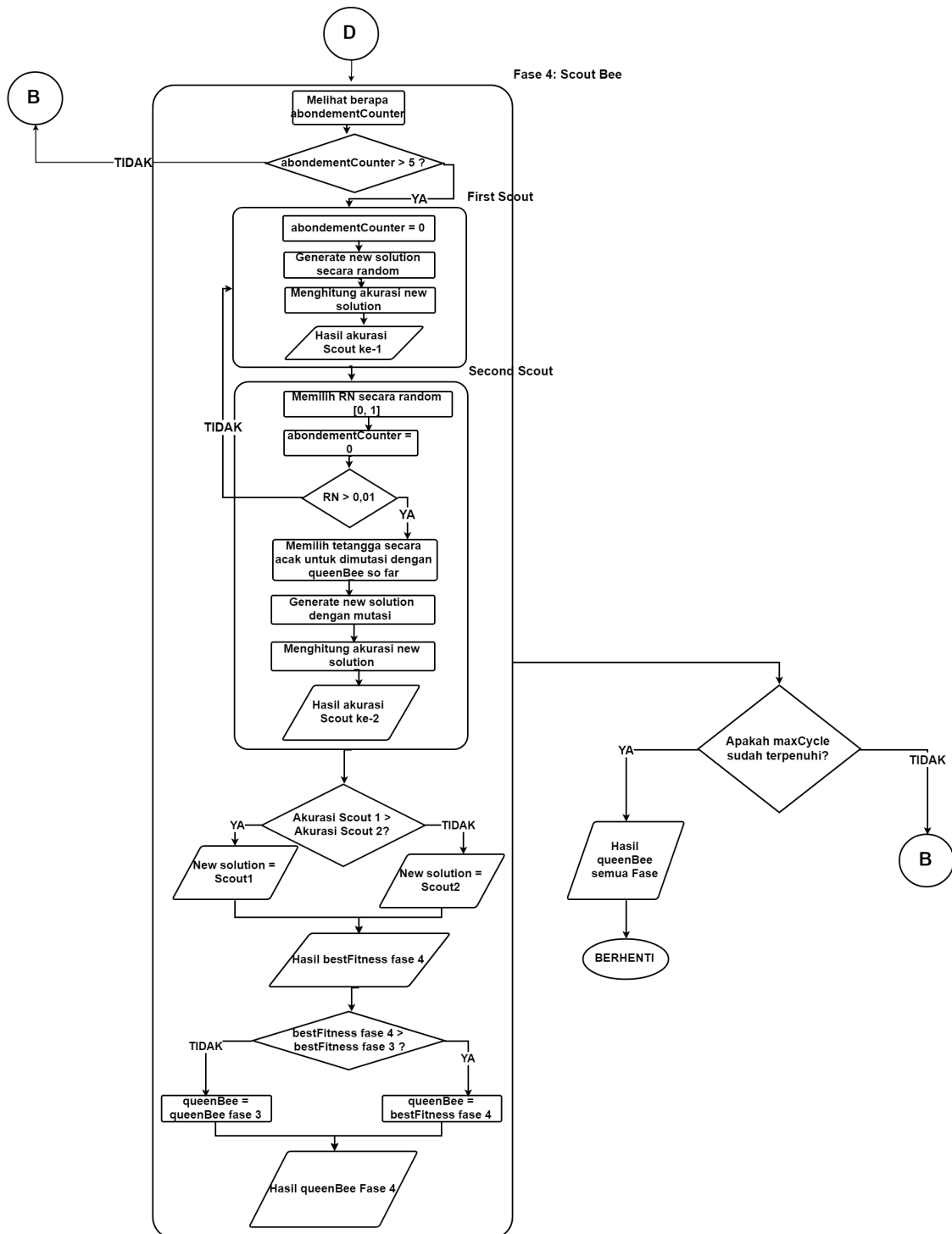
Gambar 5. Fase pertama dan kedua tahap seleksi fitur menggunakan GBC



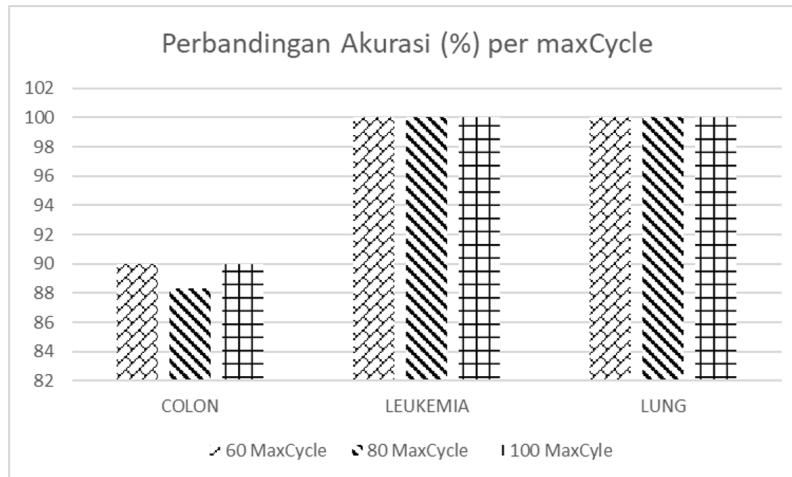
Gambar 6. Fase ketiga tahap seleksi fitur menggunakan GBC



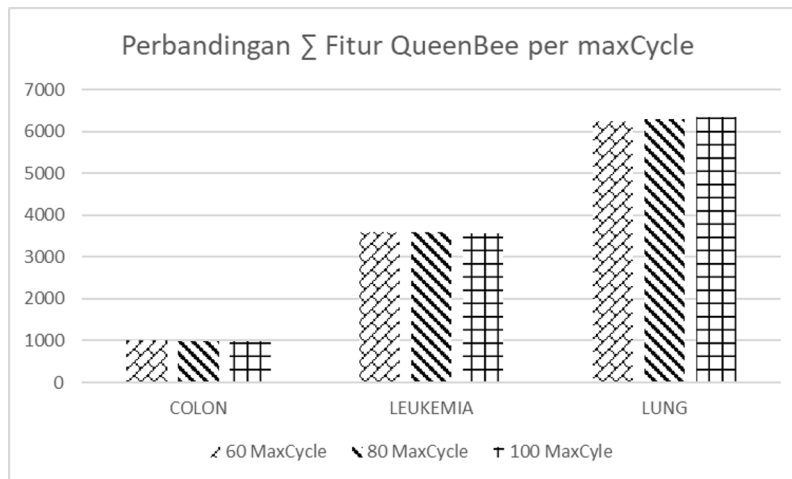
Gambar 7. Fase ke-empat tahap seleksi fitur menggunakan GBC



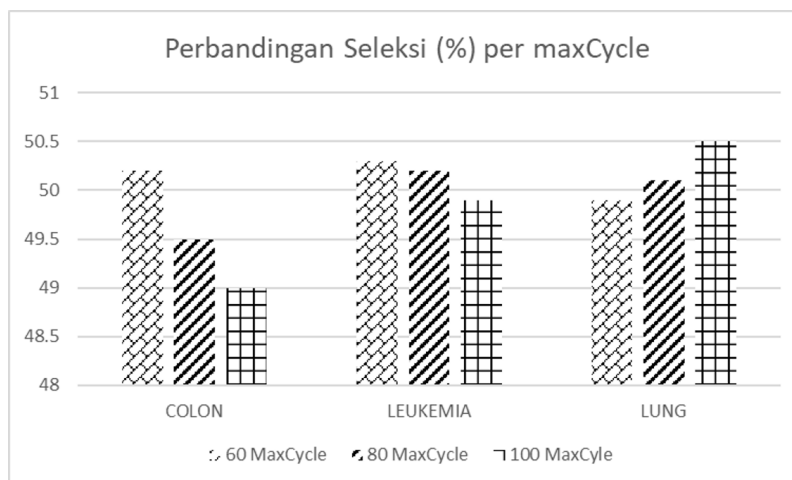
Gambar 8. Fase ke-lima tahap seleksi fitur menggunakan GBC



Gambar 9. Perbandingan akurasi per *maxCycle*



Gambar 10. Perbandingan Σ QueenBee per *maxCycle*



Gambar 11. Perbandingan seleksi per *maxCycle*