

Peningkatan Akurasi Estimasi Usaha dan Biaya COCOMO II Berdasarkan Gaussian dan BCO

Rahmi Rizkiana Putri¹, Daniel Oranova Siahaan², Sarwosri³

Abstract— An accurate effort and cost estimation provides good management for software projects. Less accurate estimation will affect the management of the software project and cause the ineffectiveness of the project development process. The addition of cost driver, introduced by Barry Boehm in 2000, is used in this paper to provide better accuracy, because it has covered the entire section in the estimation. However, in this paper, the accuracy of effort and cost estimation by COCOMO II Fuzzy Gaussian method is still far from actual effort. Therefore, the accuracy can still be increased using Bee Colony Optimization (BCO), as seen in the MMRE loyal results. The value of parameter A and B on COCOMO II is also changed with the initial gradual of 0.01 to give optimal value on a certain gradual. Based on the result of the implementation, the error accuracy of effort estimation and software project cost is reduced by 38%, compared to previous research. In conclusion, the proposed method can increase the accuracy of effort and cost estimation.

Intisari— Perkiraan usaha dan biaya yang akurat memberikan manajemen yang baik untuk proyek perangkat lunak. Perkiraan usaha dan biaya yang kurang akurat akan memengaruhi manajemen proyek perangkat lunak dan menyebabkan kurang efektifnya proses pengembangan proyek tersebut. Penambahan *cost driver* yang diperkenalkan Barry Boehm pada tahun 2000 digunakan dalam makalah ini guna memberikan akurasi yang lebih baik, karena telah mencakup keseluruhan bagian yang diestimasi. Namun, dalam makalah ini akurasi hasil perkiraan usaha dan biaya dengan metode COCOMO II Fuzzy Gaussian masih jauh dari *Actual Effort*. Oleh sebab itu, peningkatan akurasi hasil COCOMO II Fuzzy Gaussian masih dapat dilakukan dengan menggunakan metode *Bee Colony Optimization* (BCO) yang meliputi hasil optimasi yang lebih baik, terlihat dari hasil *MMRE loyal*. Tidak hanya menggunakan metode BCO saja, tetapi juga dilakukan perubahan nilai parameter A dan B pada COCOMO II dengan gradual awal adalah 0,01 untuk mencapai nilai optimal pada gradual tertentu. Berdasarkan hasil implementasi metode yang diusulkan, kesalahan akurasi perkiraan usaha dan biaya proyek perangkat lunak dapat turun 38%, dibandingkan dengan penelitian sebelumnya. Jadi, metode yang digunakan berhasil meningkatkan akurasi perkiraan usaha dan biaya.

Kata Kunci— Usaha, perkiraan biaya proyek perangkat lunak, *Constructive Cost Model II*, akurasi, Fuzzy Gaussian, *Bee Colony Optimization*.

¹ Mahasiswa, Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo Surabaya 60111 INDONESIA (tlp: 031-5939214; fax: 031-5939363; e-mail: rahmi.rizkiana.putri15@mhs.if.its.ac.id)

^{2,3} Dosen, Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo Surabaya 60111 INDONESIA (tlp: 031-5939214; fax: 031-5939363; e-mail: daniel@if.its.ac.id, sarwosri@if.its.ac.id)

I. PENDAHULUAN

Pengembangan perangkat lunak merupakan proses yang tidak sepenuhnya sempurna. Di dalamnya sering terdapat keterlambatan, kegagalan, hingga ditolaknya suatu proyek perangkat lunak. Sebagai contoh perkiraan, manajemen, efisiensi biaya, dan usaha seluruh proyek, hal tersebut adalah faktor-faktor penting dalam pengukuran perkiraan proyek perangkat lunak. Manajemen yang baik dari pengembangan perangkat lunak dapat memperkirakan biaya dan sumber daya perangkat lunak yang tepat. Perkiraan biaya proyek adalah tugas memprediksi biaya, usaha, atau produktifitas yang dibutuhkan untuk menyelesaikan proyek [1].

Sebelumnya telah diusulkan penelitian usaha dan biaya proyek perangkat lunak menggunakan metode COCOMO-ABC, dengan *data set* NASA 93 dan *Mean Magnitude Relative Error* (MMRE) yang dihasilkan adalah 33,22% [2]. Adapun penelitian selanjutnya mengusulkan metode COCOMO II-TLABC, dengan *data set* NASA 93 dan MMRE yang dihasilkan adalah 21,57% [3]. Penelitian berikut mengusulkan metode COCOMO II-algoritme genetika, *data set* NASA 60, dengan MMRE yang dihasilkan adalah 21,53% [4]. Sedangkan penelitian berikutnya mengusulkan metode Fuzzy Gaussian (FG) dengan *data set* COCOMO [5]. Hasil MMRE yang didapatkan adalah 17,02%. Dari beberapa penelitian tersebut belum diperoleh hasil estimasi yang akurat. Hal tersebut dikarenakan masih besar nilai *error* yang dihasilkan dan masih jauh dari nilai *Actual Effort* di tiap-tiap *data set* yang gunakan sebelumnya. Menurut analisis, beberapa penelitian sebelumnya hanya fokus melakukan perhitungan estimasi biaya dan usaha dengan memasukkan *data set* dan mengimplementasikan ke metode yang diusulkan. Karena masih terdapat kelemahan pada metode-metode usulan penelitian sebelumnya serta kurang akuratnya hasil estimasi usaha dan biaya yang diperoleh, maka diusulkan metode *Bee Colony Optimization* (BCO) dengan melakukan perubahan nilai pada parameter A dan parameter B. makalah ini bertujuan untuk fokus terhadap masalah perkiraan biaya dalam proyek perangkat lunak dan meningkatkan akurasi. Terdapat beberapa metode yang terkenal dalam hal perkiraan biaya perangkat lunak, salah satunya adalah COCOMO [5]. Metode ini bertujuan untuk membantu membuat perkiraan biaya sebuah proyek perangkat lunak yang sederhana, mengurai langkah-langkah dasar, istilah, dan perlengkapan yang digunakan. COCOMO II memperjelas, tidak hanya di bagian biaya proyek yang diharapkan dan durasinya, tetapi juga memverifikasi semua sisi dasar proyek perangkat lunak dengan menyediakan metodologi yang diuji pada berbagai proyek kehidupan yang nyata, dan dengan demikian mengurangi resiko dan dapat membantu memberikan alasan yang masuk akal ketika berkomunikasi dengan *stakeholder*

proyek [6]. *Cost driver* proyek perangkat lunak yang dikembangkan secara karakteristik jelas dan pasti pada tahap awal dari siklus hidupnya, sehingga sulit menghasilkan upaya perkiraan yang akurat. Biasanya muncul bias ketika pengukuran *cost driver* perangkat lunak didasarkan pada penilaian manusia. Pendekatan ini tidak mempertimbangkan fitur yang jelas dan pasti yang ditempati dalam *cost driver*. Ketidakjelasan *cost driver* secara signifikan memengaruhi akurasi perkiraan usaha dan biaya yang berasal dari model estimasi perangkat lunak, serta terkadang kurangnya informasi tentang model proyek menyebabkan metode COCOMO II ini gagal bekerja di pekerjaan yang tepat [7]. Untuk mengatasi hal tersebut, makalah ini mengusulkan dilakukannya pendekatan *fuzzy* [8]. Perkiraan biaya berdasarkan logika *fuzzy* akan lebih tepat digunakan ketika informasi yang diperoleh tidak jelas dan tidak tepat [5]. Agar mendapatkan akurasi yang lebih tepat dan mengurangi kesalahan dari COCOMO II, makalah ini menggunakan FG. Kinerja FG lebih baik dari anggota *fuzzy* lainnya, karena menunjukkan transisi interval yang halus serta hasil yang dicapai lebih dekat dengan upaya atau usaha yang sebenarnya [9].

Sejumlah besar model perkiraan biaya telah tersedia, tetapi hanya sedikit yang dapat mencapai tingkat kepuasan. Perlu adanya metode optimasi, yaitu BCO untuk mengoptimalkan COCOMO II dari sisi model parameter konstan dan evaluasi MMRE untuk memeriksa kinerja yang lebih baik dan harus lebih minimum [10]. Metode yang digunakan pada makalah ini adalah *fuzzy* dan optimasi. Gaussian *Membership Function* akan digunakan untuk mencari nilai baru pada *Effort Multiplier* (EM) untuk meningkatkan kesesuaian nilai pada setiap data EM, sedangkan BCO digunakan untuk mencari parameter baru yang cocok pada perhitungan *estimate effort* untuk mendekati nilai *Actual Effort* serta mengoptimalkan hasil yang dicapai oleh COCOMO II.

II. PENINGKATAN AKURASI ESTIMASI USAHA

A. Constructive Cost Model II

COCOMO berlaku untuk tiga kelas proyek perangkat lunak, yaitu organik-tim kecil, semi terpisah-tim menengah, dan terintegrasi yang adalah kombinasi antara organik dan semi terpisah (perangkat keras, perangkat lunak, operasional) [11]. Dengan mengikuti sistem klasifikasi Boehm ini, metode diringkas dalam tiga kategori, yaitu penilaian ahli, estimasi algoritmik, dan estimasi berdasarkan analogi [5]. Metode algoritme didasarkan pada model matematika yang menghasilkan perkiraan biaya sebagai fungsi dari sejumlah variabel, yang dianggap sebagai faktor biaya utama [12]. Metode ini menggunakan perkiraan biaya perangkat lunak metrik untuk mengidentifikasi teknik estimasi biaya, seperti SLOC (sumber baris kode). Algoritma COCOMO II memiliki 17 EM dan lima *Scale Factors* (SF) [13]. Dalam COCOMO upaya dinyatakan sebagai *Person Month* (PM). Hal ini menentukan upaya yang diperlukan untuk proyek berdasarkan ukuran proyek perangkat lunak di sumber baris ribuan kode (KSLOC) serta faktor biaya lainnya yang dikenal sebagai faktor skala dan usaha pengganda [9]. Kemampuan pemodelan baru yang utama pada COCOMO II adalah model *software sizing*, yang melibatkan *Object Points*, *Function*

Points, LOC, model *nonlinear* untuk digunakan kembali pada perangkat lunak dan rekayasa ulang, pendekatan eksponen untuk pemodelan disekonomi *software* skala *relative*, dan beberapa penambahan, penghapusan, serta *update cost driver* dari COCOMO sebelumnya. Adapun beberapa penelitian sebelumnya yang menggunakan metode ini menggunakan *fuzzy* trapesium untuk merancang ulang EM dari COCOMO II agar hasil EM lebih akurat [14]. Akan tetapi, penelitian berikut memperbaiki hasil akurasi COCOMO II menggunakan FG, karena jika menggunakan *fuzzy* trapesium, transisinya tidak sebaik Gaussian, sehingga hasil yang dicapai masih jauh dari *Actual Effort* [5]. COCOMO II juga dapat diperbaiki menggunakan BCO untuk mencapai hasil yang optimal [10]. Beberapa penelitian yang menggunakan COCOMO II hanya sebatas meningkatkan akurasi dengan metode lainnya. Namun, ada faktor penting lainnya di dalam COCOMO II, yaitu nilai parameter A dan B, yang dapat diubah sehingga memengaruhi akurasi estimasi biaya dan usaha [3]. Parameter A adalah konstanta multiplikatif di dalam COCOMO II. Konstanta multiplikatif adalah suatu nilai tetap yang dapat diperbanyak. Sedangkan parameter B adalah nilai konstanta di dalam COCOMO II [3]. Parameter A dan B di dalam COCOMO II berperan penting karena digunakan untuk estimasi usaha dan biaya perangkat lunak. Nilai parameter A yang diberikan oleh COCOMO II adalah 2,94, sedangkan nilai parameter B yang diberikan COCOMO II adalah 0,91. Parameter A digunakan ketika menghitung PM. Sedangkan parameter B digunakan ketika menghitung *Effort* (E). Adapun persamaan-persamaan yang terdapat pada COCOMO II adalah sebagai berikut [15].

Konstanta *Effort*:

$$A = 2,94; B = 0,91 \quad (1)$$

$$PM_{NS} = AxSize^E \times \prod_{i=1}^n EM_i \quad (2)$$

$$\text{dengan } E = B + 0,01 \times \sum_{j=1}^5 SF_j \quad (3)$$

Konstanta *Schedule*:

$$C=3,67; D=0,28 \quad (4)$$

$$TDEV_{NS} = Cx(PM_{NS})^F \quad (5)$$

$$\text{dengan } F = D + 0,2 \times 0,01 \times \sum_{j=1}^5 SF_j \\ = D + 0,2 \times (E - B) \quad (6)$$

Deskripsi dari tiap variabel yang terdapat dalam (1) hingga (6) yaitu nilai n adalah jumlah dari EM, EM_i adalah 16 EM dari *Post-Architecture* model, EM_i, dan enam dari rancangan awal model. SF_j berfungsi untuk eksponensial SF. Nilai A, B, EM_i,...EM₁₆, SF₁,...dan SF₅ untuk COCOMO II.2000 model *Post-Architecture* adalah kalibrasi ke parameter sebenarnya dan nilai usaha untuk 161 proyek di dalam basis data COCOMO II. Nilai C dan D untuk penjadwalan di dalam COCOMO II.2000 yang dikalibrasi ke nilai jadwal sebenarnya.

B. Fuzzy Gaussian

Sebuah sistem yang berdasarkan pada logika *fuzzy* memiliki hubungan dengan konsep *fuzzy*, seperti kumpulan *fuzzy*, variabel linguistik [15]. *Fuzzy* dapat menangani masalah dengan data yang tidak tepat dan tidak lengkap yang didasarkan pada teori himpunan *fuzzy* yang diresmikan oleh

Prof. Lofti Zadeh pada tahun 1965 [8]. Hal ini digunakan untuk mewakili nilai-nilai linguistik, seperti rendah, sedang, dan kompleks. Sebuah himpunan *fuzzy* ditandai dengan fungsi keanggotaan yang berasosiasi dengan setiap titik dalam himpunan *fuzzy* sejumlah nyata dalam interval [0,1] yang disebut derajat atau kelas keanggotaan. Interval [0,1] artinya adalah logika *fuzzy* bernilai salah atau benar [16]. Fungsi keanggotaan terdiri atas segitiga, trapesium, Gaussian, dan lain-lain [9]. Fungsi keanggotaan Gaussian memberikan akurasi yang lebih baik daripada fungsi keanggotaan lainnya. Gaussian memiliki transisi yang lebih halus antar interval. Metode ini mengurangi penyimpangan antara estimasi dan biaya atau usaha sebenarnya [8]. Seperangkat aturan sudah ada batasnya untuk pemisah *fuzzy*, yang menghasilkan upaya *software* sesuai dengan atribut nilai *fuzzy*. Upaya *software* ini terkelompok ke aturan yang beragam berdasarkan parameter dan teknik yang memanfaatkan prosedur optimasi dengan maksud untuk mengoptimalkan aturan *fuzzy* [17]. Karena FG memiliki interval yang baik dan dapat berpengaruh terhadap hasil akurasi estimasi usaha dan biaya perangkat lunak, maka makalah ini berusaha memperbaiki hasil akurasi COCOMO II menggunakan FG [5]. Menurut penelitian tersebut, transisi yang dihasilkan *fuzzy* Trapesium tidak sebaik FG, sehingga hasil yang dicapai masih jauh dari *Actual Effort*. *Rating* rendah, nominal, dan tinggi yang terdapat pada *cost driver* dikalibrasi dan diperoleh dari sistem pembelajaran data proyek industri. *Rating* tersebut digunakan pada penelitian untuk merepresentasikan salah satu faktor kontribusi untuk mengembangkan usaha perangkat lunak, seperti pengalaman menggunakan aplikasi dan kompleksitas produk [3].

C. Bee Colony Optimization (BCO)

Sejumlah besar metode estimasi telah tersedia, tetapi hanya sedikit yang dapat mencapai tingkat kepuasan hasil. Karena kompleksitas dan tingginya persyaratan perangkat lunak menuntut hasil akurasi yang lebih baik, makalah ini mengusulkan penggunaan BCO untuk memperbaiki akurasi dan mengoptimalkan biaya perangkat lunak [10]. BCO adalah contoh pendekatan sempurna untuk pemodelan jenis kecerdasan. BCO melakukan pendekatan *bottom up* yang diciptakan dari analogi lebah. BCO digunakan sebagai solusi pencarian terbaik untuk masalah optimasi. Berdasarkan penelitian yang menjelaskan inspirasi untuk menciptakan sistem multi-agen seperti BCO, metode ini berasal dari perilaku mencari makan dari lebah, yaitu madu [18]. Perilaku ini cocok untuk pemodelan, karena praktik pengumpulan dan pengolahan nektar sangat terorganisir. Algoritme BCO yang dikembangkan dapat membuat lebah membangun solusi dari nol. Beberapa penyebab kegagalan proyek, termasuk di dalamnya yaitu rendahnya akurasi, dapat berpengaruh terhadap hasil yang buruk untuk manajemen proyek. Oleh karena itu, diusulkan metode COCOMO-ABC dengan jumlah iterasi dan tingkat populasi merupakan parameter paling penting dan efektif dalam meningkatkan akurasi [2]. Untuk mendapatkan solusi terbaik dari metode BCO, umumnya digunakan loyalti yang akan melakukan pengecekan untuk tiap solusi lebah di satu kali iterasi. Kemudian diambil nilai yang memenuhi syarat loyalti BCO.

III. METODOLOGI

Dalam usaha untuk meningkatkan akurasi COCOMO II, terdapat dua tahapan, yaitu menghitung kedua *data set* menggunakan metode COCOMO II FG dan menghitung kedua *data set* menggunakan metode BCO.

TABEL I
DATA DESCRIPTOR

DATA Descriptor		Testing DB Bytes/Pgm SLOC<10	10< D/P <100	100 < D/P < 1000	D/P ≥ 10000	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
EM	n/a	0,90	1,00	1,14	1,28	n/a

A. Metode COCOMO II Fuzzy Gaussian

Tahapan pertama adalah menghitung dua jenis *data set* menggunakan COCOMO II FG, yaitu *Cocomo_sdr* yang terdiri atas 12 titik data yang terdiri atas 25 atribut. Sedangkan *data set* kedua yaitu NASA 93-dem yang terdiri atas 93 titik data. Masing-masing *data set* tersebut memiliki 17 *cost driver* COCOMO II, lima SF, ID project, *actual effort*, dan SLOC. Hasilnya berupa desain baru dari nilai EM. Adapun implementasi COCOMO II FG dilakukan dengan cara mengubah EM yang berupa *crisp* data ke dalam *fuzzy set* dengan Gaussian *membership function*. Untuk membuat estimasi biaya perangkat lunak, dibutuhkan masukan, keluaran, serta aturan yang ditetapkan di dalam *fuzzy*. Gaussian *membership function* memerlukan tiga titik. Tiap *cost driver* dan tingkatannya pada COCOMO II dimasukkan ke dalam implementasi *fuzzy* guna menentukan rentang nilai *very low* hingga *extra high* pada FG. Jika seluruh nilai EM pada COCOMO II telah dimasukkan ke dalam FG sesuai *crisp* data pada *fuzzy*, akan ditemukan perubahan nilai EM yang akan memengaruhi nilai estimasi biaya. Sebagai contoh, Tabel I memperlihatkan rentang nilai untuk DATA. Rentang nilai yang dimiliki DATA dimulai dari nilai yang paling rendah 0,90 hingga nilai paling tinggi 1,28. Setiap puncak pada keluaran disesuaikan dengan setiap nilai EM.

Tabel II menyajikan nilai EM tanpa perubahan yang berasal dari COCOMO II. Saat menggunakan FG, nilai EM mengalami perubahan di beberapa nilai *cost driver* seperti pada Tabel III. Nilai tersebut digunakan untuk melakukan estimasi terhadap biaya perangkat lunak. Perubahan nilai EM yang diimplementasi menggunakan FG ditandai dengan warna hitam. Contoh penggunaan *cost driver* yang terdapat pada Tabel I dan Tabel II yaitu jika proyek perangkat lunak akan dikembangkan untuk mengatur penerbangan pesawat maka *Required Software Reliability* (RELY) harus memiliki nilai 1,26, karena pengaturan penerbangan pesawat berkaitan dengan kehidupan manusia. RELY berfungsi untuk mengukur kinerja perangkat lunak terhadap kegagalan. Adapun rentang nilai pada *rating* RELY berkisar dari paling rendah yaitu 0,82 hingga sangat tinggi yaitu 1,26 dan tidak memiliki nilai ekstra tinggi. Berdasarkan contoh kasus tersebut dengan nilai RELY yang dicapai, maka efek kegagalan perangkat lunak sangat rendah dan tidak berisiko terhadap kehidupan manusia. Berdasarkan rentang nilai yang ada, jika RELY memiliki nilai 0,85, maka nilai tersebut berada pada level sangat rendah.

TABEL II
NILAI EFFORT MULTIPLIER COCOMO II

No	Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
1	RELY	0,82	0,92	1	1,1	1,26	-
2	DATA	-	0,9	1	1,14	1,28	-
3	RESU	-	0,95	1	1,07	1,15	1,24
4	DOCU	0,81	0,91	1	1,11	1,23	-
5	TIME	-	-	1	1,11	1,29	1,63
6	STOR	-	-	1	1,05	1,17	1,46
7	PVOL	-	0,87	1	1,15	1,3	-
8	ACAP	1,42	1,19	1	0,85	0,71	-
9	PCAP	1,34	1,15	1	0,88	0,76	-
10	PCON	1,29	1,12	1	0,9	0,81	-
11	APEX	1,22	1,1	1	0,88	0,81	-
12	PLEX	1,19	1,09	1	0,91	0,85	-
13	LTEX	1,2	1,09	1	0,91	0,84	-
14	TOOL	1,17	1,09	1	0,9	0,78	-
15	SITE	1,22	1,09	1	0,93	0,86	0,8
16	SCED	1,43	1,14	1	1	1	-
17	CPLX	0,73	0,87	1	1,17	1,34	1,74

TABEL III
NILAI EFFORT MULTIPLIER COCOMO II FUZZY GAUSSIAN

No	Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
1	RELY	0,82	0,92	1	1,1	1,26	-
2	DATA	-	0,9	1	1,14	1,28	-
3	RESU	-	0,95	1	1,07	1,15	1,24
4	DOCU	0,81	0,91	1	1,11	1,23	-
5	TIME	-	-	1	1,11	1,29	1,59
6	STOR	-	-	1	1,05	1,16	1,45
7	PVOL	-	0,87	1	1,14	1,23	-
8	ACAP	1,41	1,18	1,01	0,85	0,71	-
9	PCAP	1,34	1,15	1	0,88	0,76	-
10	PCON	1,26	1,12	1	0,9	0,81	-
11	APEX	1,11	1,1	0,998	0,88	0,81	-
12	PLEX	1,13	1,06	1	0,91	0,857	-
13	LTEX	1,16	1,1	1	0,91	0,85	-
14	TOOL	1,17	1,09	1	0,9	0,78	-
15	SITE	1,22	1,09	1	0,93	0,86	0,8
16	SCED	1,31	1,14	1	1	1	-
17	CPLX	0,73	0,87	1	1,17	1,34	1,74

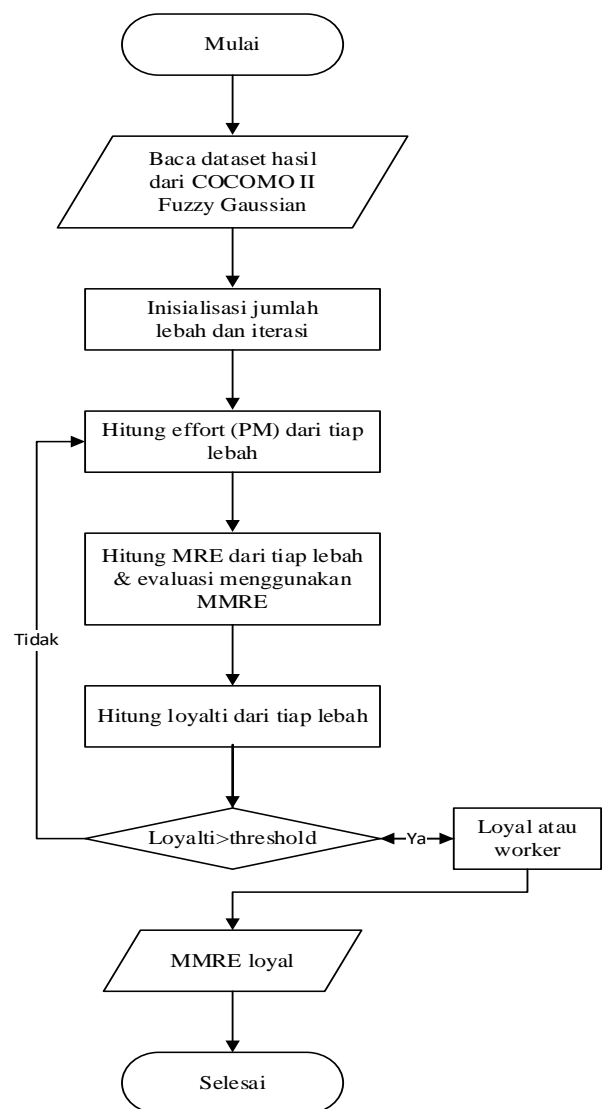
B. Metode Bee Colony Optimization

Tahapan kedua adalah mengolah *data set* dengan menggunakan metode BCO agar *error* yang dihasilkan oleh FG semakin kecil dan akurasi semakin meningkat, sehingga hasil lebih optimal. Nilai baru dari EM yang dihasilkan dari implementasi *fuzzy* digunakan untuk implementasi pada BCO. Hasil implementasi tersebut diperoleh melalui beberapa tahapan proses. Proses dilakukan dari mengolah *data set*, menghitung *data set* menggunakan COCOMO II FG, hingga tahapan BCO. Proses tersebut dijelaskan secara lebih rinci seperti berikut.

- Membaca data dari kedua *data set* (COCOMO-sdr dan NASA 93-dem).
- Memastikan jumlah secara lebih spesifik dan pembagian kelompok lebah menjadi tiga bagian.

- Inialisasi parameter dari metode COCOMO II (LOC, EM, SF) dari FG.
- Menyiapkan modul BCO agar dapat dipanggil.
- Menghitung nilai *effort* (E) dari tiap lebah.
- Menghitung MRE di tiap lebah yang sudah mencari nilai E.
- Menghitung MMRE dari keseluruhan MRE. Jika hasil MMRE mendekati *Actual Effort* (AE), maka akan dipilih.
- Menghitung nilai loyalti dari setiap lebah yang nantinya akan digunakan sebagai faktor proses seleksi.
- Langkah 5 hingga langkah 8 diulang kembali untuk setiap lebah.
- Solusi yang tidak loyal (optimal) tidak digunakan.
- Mengatur nilai solusi yang paling mendekati nilai MMRE.
- BCO model berhenti.

Adapun diagram alir berdasarkan proses tersebut ditunjukkan pada Gbr. 1.



Gbr. 1 Diagram alir implementasi BCO.

IV. UJI COBA DAN ANALISIS HASIL

Setelah mengimplementasikan metode yang diusulkan, tahap selanjutnya adalah melakukan uji coba. Uji coba dilakukan untuk melihat kinerja yang dihasilkan oleh metode yang diusulkan. Data masukan pada uji coba ini adalah Cocomo-sdr dan NASA 93-dem. Data yang digunakan diperoleh dari *Turkish Software Industry* dan NASA. *Data set* Turkish terdiri atas 12 titik data sedangkan NASA 93-dem terdiri atas 93 titik data. Masing-masing terdiri atas 25 atribut, 17 *cost driver*, lima SF, SLOC, dan *Actual Effort*. Terdapat dua skenario uji coba yang dilakukan pada penelitian ini, yaitu sebagai berikut.

- Membandingkan hasil akurasi estimasi usaha dan biaya menggunakan COCOMO II FG pada *data set* Cocomo-sdr dan NASA 93-dem [14].
- Membandingkan hasil akurasi estimasi usaha dan biaya menggunakan BCO pada *data set* Cocomo-sdr dan NASA 93-dem [3].

Evaluasi uji coba dilakukan dengan membandingkan akurasi hasil dari mengubah nilai parameter A dan B dengan akurasi hasil tanpa mengubah nilai parameter A dan B. Perubahan nilai parameter A dan B berfungsi sebagai solusi pada tiap parameter untuk setiap kemungkinan yang ada. Pembuatan solusi pada setiap perulangan dilakukan dengan menaikkan dan menurunkan nilai tiap parameter A dan B sebesar 0,01. Pencarian dilakukan hingga memenuhi keadaan berhenti. Batasan pencarian juga dapat ditentukan berdasarkan jumlah perulangan yang akan dilakukan menggunakan BCO. Setiap kemungkinan dicari untuk mendapatkan nilai terbaik dari parameter perhitungan. Penentuan solusi awal merupakan langkah pertama, kemudian dilakukan pencarian untuk kemungkinan solusi baru, lalu dilakukan pengecekan mengenai solusi baru, memenuhi kriteria atau tidak. Jika memenuhi kriteria, maka kondisi akan berhenti dan nilai solusi baru tersebut menjadi nilai solusi akhir yang dipilih. Evaluasi dilakukan dengan dua cara, yaitu metode COCOMO II FG menggunakan *data set* Cocomo_sdr dan COCOMO II FG menggunakan *data set* NASA 93-dem. Evaluasi kedua metode BCO menggunakan *data set* Cocomo_sdr dan BCO menggunakan *data set* NASA 93-dem.

A. Hasil Uji Coba Skenario 1

Uji coba pertama adalah melakukan implementasi *fuzzy* pada COCOMO II menggunakan metode FG pada *data set* NASA 93-dem dan membandingkan dengan penelitian yang menggunakan *data set* Cocomo_sdr [14]. Beberapa EM diubah nilainya ketika menggunakan FG untuk meminimalkan *error*. Uji coba pertama ini dimaksudkan untuk mengetahui hasil dari kinerja metode yang menggunakan COCOMO II FG. Tabel IV menunjukkan hasil implementasi pada skenario uji coba pertama. Jika hasil MMRE dari *data set* Cocomo-sdr tersebut diimplementasikan ke dalam bentuk biaya estimasi usaha perangkat lunak, maka uang berjumlah Rp1.000.000,00 yang dimiliki akan ditambahkan dengan hasil MMRE yang ada, yaitu 716,85%. Hasilnya adalah Rp8.168.500,00, yang artinya *actual* 1 juta rupiah dan estimasinya adalah Rp8.168.500,00.

TABEL IV
MMRE DENGAN METODE COCOMO II FUZZY GAUSSIAN

<i>Data set</i>	MMRE (%)
Cocomo_sdr	716,85
NASA 93-dem	51,08

TABEL V
MMRE LOYAL DENGAN METODE BCO

<i>Data set</i>	MMRE loyal (%)
Cocomo_sdr	7,9743
NASA 93-dem	12,92

TABEL VI
COCOMO-SDR DENGAN GRADUAL 0,01

Solusi Parameter A	Solusi Parameter B	MMRE(%)
0,19	0,91	9,21

TABEL VII
COCOMO-SDR DENGAN GRADUAL 0,001

Solusi Parameter A	Solusi Parameter B	MMRE(%)
0,189	0,905	8,01

TABEL VIII
COCOMO-SDR DENGAN GRADUAL 0,0001

Solusi Parameter A	Solusi Parameter B	MMRE(%)
0,1894	0,9045	7,9743

TABEL IX
NASA 93-DEM DENGAN GRADUAL 0,01

Solusi Parameter A	Solusi Parameter B	MMRE(%)
2,95	0,91	28,01

TABEL X
BCO DENGAN GRADUAL 0,001

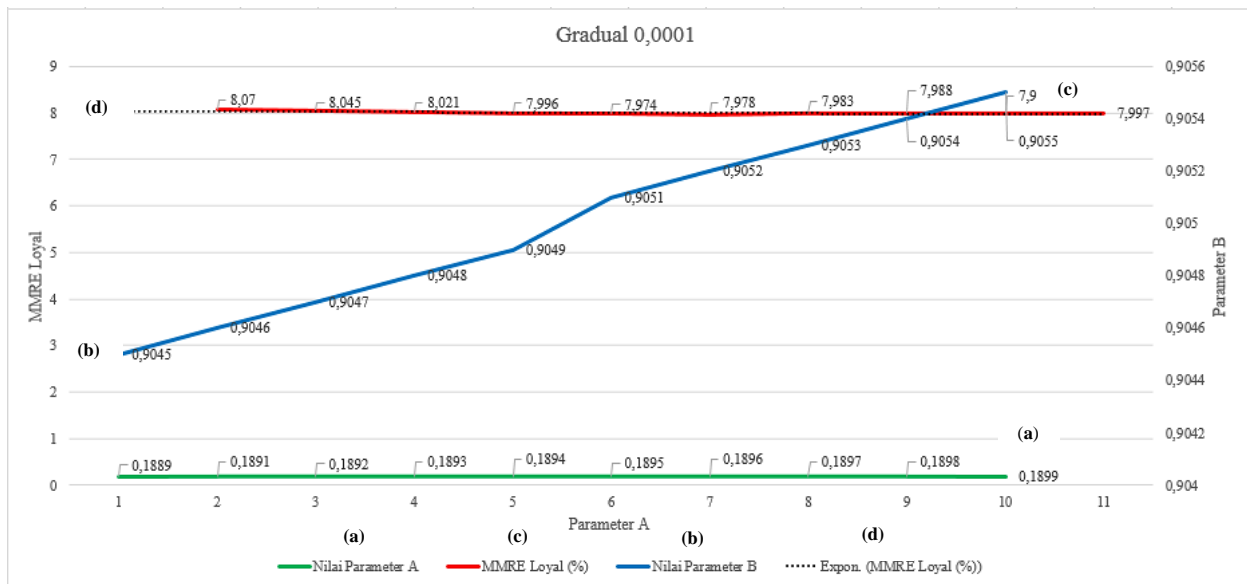
Solusi Parameter A	Solusi Parameter B	MMRE(%)
2,955	0,905	1,6927345

TABEL XI
BCO DENGAN GRADUAL 0,0001

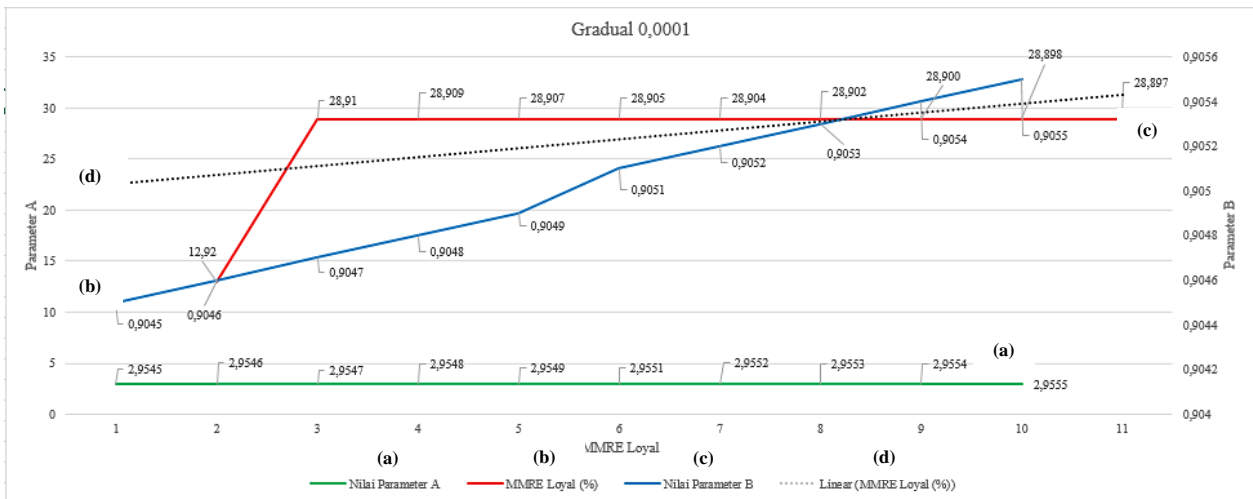
Solusi Parameter A	Solusi Parameter B	MMRE(%)
2,9545	0,9045	12,92

B. Hasil Uji Coba Skenario 2

Uji coba kedua yaitu melakukan implementasi BCO dengan perubahan nilai parameter A dan B dan gradual tertentu pada *data set* Cocomo-sdr dan NASA-93-dem serta membandingkannya dengan penelitian sebelumnya [3]. Beberapa EM diubah nilainya ketika menggunakan BCO untuk meminimalkan *error* dan mencapai akurasi yang optimal. Uji coba kedua ini dilakukan untuk mengetahui hasil dari unjuk kerja metode yang menggunakan BCO. Tabel V menunjukkan hasil implementasi skenario uji coba kedua. Jika hasil MMRE loyal dari *data set* NASA 93-dem tersebut diimplementasikan ke dalam bentuk biaya estimasi usaha perangkat lunak, maka uang berjumlah Rp1.000.000,00 yang dimiliki akan ditambahkan dengan hasil MMRE loyal yang ada, yaitu 12,92%. Maka hasilnya adalah Rp1.129.200,00, yang artinya *actual* 1 juta rupiah dan estimasinya adalah Rp1.129.200,00.



Gbr. 2 Grafik MMRE loyal data set Cocomo-sdr.



Gbr. 3 Grafik MMRE loyal data set NASA 93-dem.

C. Analisis Hasil Uji Coba

Pada makalah ini, data set yang digunakan adalah data set Cocomo-sdr yang diperoleh dari Turkish Software Industry, yang memiliki 12 proyek, dan data set NASA-93-dem yang diperoleh dari NASA. Pengujian awal dilakukan menggunakan COCOMO II FG. Metode tersebut digunakan dengan alasan hasil estimasi dari COCOMO II belum dapat meningkatkan akurasi karena hasil error yang masih besar, sehingga hasil yang diperoleh masih jauh dari nilai Actual Effort. Metode FG dapat menurunkan error lebih baik daripada metode COCOMO II, karena adanya perubahan EM yang berupa crisp data ke dalam fuzzy set dengan Gaussian membership function. Setelah hasil estimasi Gaussian diperoleh, ternyata hasil MMRE masih belum dapat mencapai akurasi yang mendekati Actual Effort.

Makalah ini mengusulkan metode BCO untuk mengoptimasi hasil estimasi usaha dan biaya perangkat lunak dari metode COCOMO II FG. BCO membutuhkan beberapa

masukan, seperti Actual Effort, SLOC, EM, dan SF. Perhitungan pada implementasi adalah dengan mencari solusi dari nilai parameter A dan B untuk setiap kemungkinan yang ada. Pembuatan solusi pada setiap perulangan dilakukan dengan menaikkan dan menurunkan kedua nilai parameter tersebut. Pencarian dilakukan hingga memenuhi keadaan berhenti. BCO dimulai dengan menentukan nilai solusi awal, kemudian dilakukan pencarian untuk kemungkinan nilai solusi baru. Selanjutnya, dilakukan pengecekan, solusi baru memenuhi kriteria atau tidak, dan jika telah memenuhi kondisi berhenti, maka nilai solusi baru tersebut menjadi nilai solusi akhir.

Kedua data set menggunakan cost driver serta parameter A dan B dari COCOMO II. Terdapat tabel dan gambar grafik yang menampilkan perubahan nilai pada masing-masing parameter serta dilengkapi dengan gradual nilai parameter pada kedua data set yang terdiri atas tiga tahapan perubahan gradual. Hal tersebut dilakukan guna mencapai hasil MMRE yang optimal. Perubahan nilai parameter yang pertama

dilakukan pada *data set* Cocomo-sdr. Tahap awal ditampilkan pada Tabel VI dengan gradual 0,01.

Berdasarkan Tabel VI, solusi parameter A terdapat pada iterasi ke-19, sedangkan solusi parameter B terdapat pada iterasi ke-91. Nilai tersebut didapatkan setelah dilakukan iterasi sebanyak 295 kali dan dicari nilai minimal di antara seluruh nilai di tiap iterasi. *Range* nilai pada tiap iterasi parameter A adalah antara nilai 0 hingga 2,94. *Range* nilai pada tiap iterasi parameter B adalah antara nilai 0 hingga 0,91.

Hasil tahap kedua ditunjukkan pada Tabel VII dengan gradual 0,001.

Berdasarkan Tabel VII, solusi parameter A terdapat pada iterasi ke-1, sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut diperoleh setelah dilakukan iterasi sebanyak 10 kali dan dicari nilai minimal di antara seluruh nilai di tiap iterasi. *Range* nilai pada tiap iterasi parameter A adalah antara nilai 0,185 hingga 0,195. *Range* nilai pada tiap iterasi parameter B adalah antara nilai 0,905 hingga 0,915. Hasil tahap ketiga ditunjukkan pada Tabel VIII dengan gradual 0,0001.

Berdasarkan Tabel VIII, solusi parameter A terdapat pada iterasi ke-5, sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut diperoleh setelah dilakukan iterasi sebanyak 10 kali dan dicari nilai minimal di antara seluruh nilai di tiap iterasi. *Range* nilai pada tiap iterasi parameter A adalah antara nilai 0,1889 hingga 0,1899. *Range* nilai pada tiap iterasi parameter B adalah antara nilai 0,9045 hingga 0,9055.

Perubahan nilai parameter yang pertama dilakukan pada *data set* NASA 93-dem. Hasil tahap awal ditunjukkan pada Tabel IX dengan gradual 0,01.

Berdasarkan Tabel IX, solusi parameter A terdapat pada iterasi ke-295, sedangkan solusi parameter B terdapat pada iterasi ke-91. Nilai tersebut didapatkan setelah dilakukan iterasi sebanyak 295 kali dan dicari nilai minimal di antara seluruh nilai di tiap iterasi. *Range* nilai pada tiap iterasi parameter A adalah antara nilai 0 hingga 2,94. *Range* nilai pada tiap iterasi parameter B adalah antara nilai 0 hingga 0,91.

Hasil tahap kedua ditunjukkan pada Tabel X dengan gradual 0,001.

Berdasarkan Tabel X, solusi parameter A terdapat pada iterasi ke-10, sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah dilakukan iterasi sebanyak 10 kali dan dicari nilai minimal di antara seluruh nilai di tiap iterasi. *Range* nilai pada tiap iterasi parameter A adalah antara nilai 2,945 hingga 2,955. *Range* nilai pada tiap iterasi parameter B adalah antara nilai 0,905 hingga 0,915. Hasil tahap ketiga ditunjukkan pada Tabel XI dengan gradual 0,0001.

Berdasarkan Tabel XI, solusi parameter A terdapat pada iterasi ke-1, sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah dilakukan iterasi sebanyak 10 kali dan dicari nilai minimal di antara seluruh nilai di tiap iterasi. *Range* nilai pada tiap iterasi parameter A adalah antara nilai 2,9545 hingga 2,9555. *Range* nilai pada tiap iterasi parameter B adalah antara nilai 0,9045 hingga 0,9055.

Hasil yang dianalisis pada bagian ini adalah hasil akurasi MMRE dari kedua *data set* dari pengujian tahap pertama hingga ketiga menggunakan metode BCO. Grafik yang menampilkan hasil solusi terbaik dari tiap uji coba diperlihatkan pada Gbr. 2 dan Gbr. 3. Hasil tersebut adalah hasil akhir dari MMRE loyal dengan *error* minimal.

Gbr. 2 memperlihatkan MMRE loyal dengan gradual 0,0001 dari *data set* Cocomo-sdr. Grafik menunjukkan *trendline* menurun, tetapi tidak secara signifikan. Nilai paling rendah berada pada titik 7,974. Hal tersebut dikarenakan *range* hasil nilai MMRE loyal tiap iterasi tidak terlalu besar.

Gbr. 3 memperlihatkan MMRE loyal dengan gradual 0,0001 dari *data set* NASA 93-dem. Grafik menunjukkan *trendline* naik tetapi tidak secara signifikan. Hal tersebut dikarenakan *range* hasil nilai MMRE loyal tiap iterasi tidak terlalu besar. Nilai MMRE loyal paling rendah berada pada titik 12,92. Dari hasil tersebut, tampak bahwa solusi dengan gradual 0,0001 merupakan solusi terbaik dibandingkan solusi lain karena memiliki hasil MMRE loyal yang paling kecil serta akurasi estimasi yang optimal.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil uji coba dan analisis hasil pengujian dengan menggunakan *data set* Cocomo-sdr dan *data set* NASA 93-dem, nilai parameter A dan parameter B diubah naik dan turun dari gradual 0,01 guna menurunkan *error* dalam estimasi usaha. Sedangkan hasil optimal didapatkan pada gradual 0,0001. Evaluasi yang dilakukan melalui MMRE merupakan nilai rata-rata dari MRE seluruh data. MMRE loyal menggunakan BCO dapat menurunkan *error* lebih banyak daripada menggunakan COCOMO II FG. Hal tersebut terbukti dari MMRE loyal yang dihasilkan menggunakan *data set* Cocomo-sdr adalah 7,9743%, sedangkan menggunakan *data set* NASA 93-dem adalah 12,92%. Dapat disimpulkan bahwa metode yang diusulkan menghasilkan nilai MMRE loyal lebih optimal dan dapat menurunkan *error* 38% lebih baik dibandingkan penelitian sebelumnya. Hal tersebut dikarenakan adanya perubahan nilai parameter A dan parameter B yang dilakukan.

Untuk penelitian selanjutnya, disarankan memperbaiki hasil MMRE loyal pada makalah ini, karena hasil MMRE loyal pada *data set* NASA 93-dem lebih besar daripada hasil MMRE loyal pada *data set* Cocomo-sdr. Hal tersebut disebabkan oleh MMRE loyal yang dihasilkan pada *data set* NASA 93-dem dengan gradual awal 0,01 berada di iterasi ke-295, ketika nilai parameter A adalah 2,95. Akibatnya, nilai parameter A di setiap perubahan gradual dan di setiap iterasi mencapai 2,9555 pada gradual akhir yang optimal, yaitu 0,0001. Sementara itu, hasil MMRE loyal yang dihasilkan pada *data set* Cocomo-sdr dengan gradual awal 0,01 berada pada iterasi ke-19 ketika nilai parameter A adalah 0,19, sehingga nilai parameter A di setiap perubahan gradual dan di setiap iterasi mencapai 0,1899 pada gradual akhir yang optimal, yaitu 0,0001.

Penelitian selanjutnya dapat menggunakan *data set* atau jumlah *cost driver* yang berbeda. Dengan *data set* dan jumlah *cost driver* yang berbeda, maka akurasi estimasi akan semakin

meningkat karena jumlah data *training* dan *testing* akan semakin banyak, serta semakin banyak bagian dari proyek perangkat lunak yang diestimasi.

REFERENSI

- [1] A. C. Eberendu, "Software Project Cost Estimation : Issues , Problems and Possible Solutions," *Int. J. Eng. Sci. Invent.*, vol. 3, no. 6, pp. 38–43, 2014.
- [2] F. Soleimanian Gharehchopogh, I. Maleki, A. Kamalinia, and H. M. Zadeh, "Artificial bee colony based constructive cost model for software cost estimation," *J. Sci. Res. Dev.*, vol. 1, no. 2, pp. 44–51, 2014.
- [3] K. T. Le My, "Applying Teaching-Learning To Artificial Bee Colony for Parameter Optimization of Software Effort," *J. Eng. Sci. Technol.*, August, 2016.
- [4] F. S. Gharehchopogh and A. Pourali, "A new approach based on continuous genetic algorithm in software cost estimation," *J. Sci. Res. Dev.*, vol. 2, no. 4, pp. 87–94, 2015.
- [5] C. S. Reddy and K. Raju, "An Improved Fuzzy Approach for COCOMO ' s Effort Estimation using Gaussian Membership Function," *J. Softw.*, vol. 4, no. 5, pp. 452–459, 2009.
- [6] T. N. Sharma, "Analysis of Software Cost Estimation using COCOMO II," *Int. J. Sci. Eng. Res.*, vol. 2, no. 6, pp. 1–5, 2011.
- [7] C. H. S. Reddy and K. Raju, "Improving the accuracy of effort estimation through Fuzzy set combination of size and cost drivers," *WSEAS Trans. Comput.*, vol. 8, no. 6, pp. 926–936, 2009.
- [8] E. Stimulation, "A Fuzzy Approach For Software Effort Estimation," *Int. J. Cybern. Informatics*, vol. 2, no. 1, pp. 9–15, 2013.
- [9] C. S. Reddy and K. Raju, "Improving the accuracy of effort estimation through fuzzy set combination of size and cost drivers," *WSEAS Trans. Comput.*, vol. 8, no. 6, pp. 926–936, 2009.
- [10] S. Chalotra, S. K. Sehra, Y. S. Brar, and N. Kaur, "Tuning of COCOMO Model Parameters by using Bee Colony Optimization," *Indian J. Sci. Technol.*, vol. 8, no. July, 2015.
- [11] A. Malik, V. Pandey, and A. Kaushik, "An analysis of fuzzy approaches for COCOMO II," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 5, pp. 68–75, 2013.
- [12] J. G. Borade and V. R. Khalkar, "Software Project Effort and Cost Estimation Techniques," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 8, pp. 730–739, 2013.
- [13] S. Waghmode and K. Kolhe, "A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 4, pp. 3892–3899, 2014.
- [14] R. R. Putri, R. Sarno, D. Siahaan, A. S. Ahmadiyah, and S. Rochimah, "Accuracy Improvement of the Estimations Effort in Constructive Cost Model II Based on Logic Model of Fuzzy," *Adv. Sci. Lett.*, vol. 23, no. 3, pp. 2478–2480, 2017.
- [15] I. Attarzadeh and S. Ow, "Improving the accuracy of software cost estimation model based on a new fuzzy logic model," *World Appl. Sci. J.* 2010.
- [16] S. Adhimantoro, "Mengetahui Tingkat Kematangan Buah Dengan Ultrasonik Menggunakan Logika Fuzzy," *JNTETI*, vol. 3, no. 1, pp. 1–6, 2014.
- [17] T. K. G. Marappagounder, "An Efficient Software Cost Estimation Technique Using Fuzzy Logic With The Aid Of Optimization Algorithm," *Int. J. Innov. Comput. Inf. Control*, vol. 11, pp. 587 – 597, 2015.
- [18] T. Davidov, D. Teodorov, and M. Selim, "Bee Colony Optimization Part I: the Algorithm Overview," *Yugosl. J. Oper. Res.*, vol. 25, no. 1, pp. 33–56, 2015.