

SECURITY CONSTRAINED UNIT COMMITMENT MENGGUNAKAN METODE BINARY SELF ADAPTIVE MODIFIED FIREFLY ALGORITHM

Totok Dewantoro, Adi Soeprijanto, Rony Seto Wibowo
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: totok.dewantoro12@mhs.ee.its.ac.id

Abstrak— Hasil unit comitment (UC) memberikan penjadwalan pembangkit harian untuk operator sistem tenaga, Namun, jadwal dari UC tidak menjamin bahwa daya akan mengalir dengan sukses dari generator ke beban melalui sistem jaringan transmisi. Oleh karena itu, istilah security constrained unit commitment (SCUC) diperkenalkan untuk mendefinisikan UC. Tujuan utama dari makalah ini adalah untuk menentukan penjadwalan unit pembangkit dengan total biaya minimal. Batasan batasan pembangkitan, seperti batasan permintaan beban, cadangan berputar, batasan ramp up/down, batasan minimum On/Off, batasan keamanan tegangan masing masing bus, dan batasan aliran tiap tiap cabang ikut diperhitungkan pada makalah ini. Untuk menyelesaikan permasalahan SCUC, sebuah binary firefly algorithm (BFA) digunakan untuk menyelesaikan masalah utama unit commitment dan real coded firefly algorithm (RCFA) digunakan untuk menyelesaikan permasalahan economic dispatch dan optimal power flow. Usulan Algoritma dites pada sistem IEEE 30 bus dengan 6 generator thermal. Hasil menunjukkan bahwa total biaya dari SCUC lebih tinggi sebesar \$195,65 dibandingkan dengan UC. Akan tetapi hasil penjadwalan dengan SCUC ini memastikan bahwa sistem dapat melayani permintaan beban secara layak dan aman.

Kata Kunci— **Security Constraint Unit Commitment (SCUC), Firefly Algorithm (FA)**

I. PENDAHULUAN

Unit commitment (UC) didefinisikan sebagai penjadwalan produksi daya listrik yang dihasilkan unit pembangkit pada periode harian atau mingguan yang akan datang (pada periode tertentu di masa datang) dengan tujuan mendapatkan biaya ekonomis pembangkitan. Banyak penelitian yang dilakukan mengenai masalah UC. Hal ini disebabkan UC merupakan masalah yang penting dalam perencanaan operasi jangka-pendek dari sistem tenaga listrik.

Di tahun 1940-an sampai dengan 1970-an, suatu metode digunakan untuk menyelesaikan masalah UC. Metode ini disebut sebagai metode pemrograman integer campuran, dan berdasar atas daftar skala prioritas. Metode ini memiliki kelebihan pada kesederhanaan dalam penyelesaiannya

Pada tahun 1980-an metode *Branch dan Bound Algorithm* dan metode *Dynamic Programming* diaplikasikan untuk menyelesaikan masalah UC. Kedua metode ini dapat

menemukan solusi yang optimal, untuk jumlah unit pembangkit yang kecil. Untuk jumlah unit pembangkit yang besar bisa dilakukan dengan metode pendekatan *Lagrangian Relaxation*. Metode ini membutuhkan waktu penyelesaian yang lebih pendek bila dibandingkan dengan *Dynamic Programming*, akan tetapi metode ini sangat sensitif terhadap pengali Lagrange, sehingga dengan bertambahnya pengali lagrange sulit dicari solusinya.

Sebuah pendekatan baru yang disebut *Sequential Unit Commitment* dikembangkan untuk menyempurnakan metode *Lagrangian Relaxation* dan 2 metode *Dynamic Programming*. Metode ini membutuhkan daftar prioritas heuristic untuk mendapatkan harga awal. Selain prioritas tradisional, metode ini menyeleksi unit yang paling menguntungkan untuk komit pada basis operasi ekonomis dan permintaan sistem selama proses iterasi.

Sebuah algoritma optimisasi unit-tunggal baru diajukan untuk menggantikan metode *Dynamic programming*. Metode ini menggunakan batasan *minimum up and down time* unit dan batasan cadangan dari sistem yang diteliti untuk menangani jadwal secara efisien, sehingga ruang penelitian bisa direduksi secara signifikan, dan perhitungan menjadi lebih cepat. Batasan keseimbangan daya, batasan cadangan sistem, batasan minimum up and down time, batasan level minimum dan maksimum pembangkitan dan batasan ramp rate dipertimbangkan.

Dari berbagai penelitian tersebut, hasil penjadwalan UC tidak menjamin bahwa daya yang dibangkitkan akan berhasil mengalir dari generator ke beban melalui jaringan. Oleh karena itu istilah *Security Constrained Unit Commitment (SCUC)* diperkenalkan untuk mentukan UC.

SCUC merupakan masalah optimisasi pada sistem tenaga yang besar. Dalam penyelesaiannya diperlukan suatu metode metaheuristik. *Firefly Algorithm (FA)* merupakan salah satu metode optimisasi metaheuristik yang terinspirasi oleh kecerahan dari sinar yang dipancarkan oleh kunang - kunang. Dr. Xin She Yang dari Cambridge University berhasil mengembangkan metode ini. FA memiliki tingkat kehandalan dan efisiensi yang tinggi dalam menyelesaikan masalah nonlinear. Pada makalah ini FA digunakan sebagai metode utama untuk menyelesaikan masalah SCUC.

II. SECURITY CONSTRAINED UNIT COMMITMENT DAN SELF ADAPTIVE MODIFIED FIREFLY ALGORITHM

2.1 Security Constrained Unit Commitment (SCUC)

Secara umum, fungsi objektif dari *security constrained unit commitment* (SCUC) dapat dirumuskan sebagai berikut:

$$TCP = \min \sum_{t=1}^T \sum_{i=1}^N \left\{ \begin{array}{l} F_i(P_{i,t}) \cdot I_{i,t} \\ + SU_i^t \cdot I_{i,t} [1 - I_{i,t-1}] \\ + SD_i^t \cdot I_{i,t-1} [1 - I_{i,t}] \end{array} \right\} \quad (1)$$

TCP pada persamaan (1) menyatakan biaya total operasional untuk N unit pembangkit selama T dari jam beroperasi, Adanya persyaratan operasional, minimalisasi fungsi objektif diatas harus memenuhi batasan batasan berikut:

- Keseimbangan Daya

Total daya yang dihasilkan pada setiap jam harus sama dengan jumlah permintaan beban tiap jam.

$$\sum_{i=1}^N P_{i,t} I_{i,t} = Load_t \quad (2)$$

- Minimum On/Off

Minimum on/off menunjukkan bahwa unit harus on/off untuk sejumlah jam tertentu, sebelum dapat dimatikan atau dinyalakan kembali.

$$X_i^{on}(t) \geq MU_i \text{ dan } X_i^{off}(t) \geq MD_i \quad (3)$$

- Batasan Pembangkitan dan Ramp/up/down

Secara praktis output pembangkit dari generator tidak dapat ditentukan seketika tanpa batasan. Jarak pengoperasian untuk semua unit secara online dibatasi batasan ramp rate selama tiap jam.

$$P_{i,t(r\min)} = \{P_{i(\min)}, P_{i,t-1} - RD_i\} \quad (4)$$

$$P_{i,t(r\max)} = \{P_{i(\max)}, P_{i,t-1} - RU_i\} \quad (5)$$

$$P_{i,t(r\min)} \leq P_{i,t} \leq P_{i,t(r\max)} \quad (6)$$

- Cadangan Berputar

Untuk operasi yang handal, sistem listrik harus mempertahankan kapasitas Megawatt tertentu sebagai cadangan berputar.

$$\sum_{i=1}^N P_{i(\max)} I_{i,t} \geq Load_t + SR_t \quad (7)$$

- Batasan keamanan

$$V_{i(\min)} \leq V_{i,t} \leq V_{i(\max)} \text{ dan } |S_{i,t}| \leq S_{i(\max)} \quad (8)$$

- Inisial status unit pada awal periode penjadwalan harus diperhitungkan

Keterangan :

TCP Total Biaya Produksi

$F_i(P_{i,t})$ Fungsi biaya pembangkitan unit i, dengan $P_{i,t}$ pada jam ke T yang dirumuskan sebagai berikut :

$$F_i(P_{i,t}) = a + b * P_{i,t} + c * P_{i,t}^2 \quad (9)$$

N Jumlah generator

T Interval waktu dalam Jam

$P_{i,t}$ Output generator i pada jam t

$I_{i,t}$ Status On/Off unit i saat jam ke t, $I_{i,t} = 0$ ketika Off, $I_{i,t} = 1$ ketika On

$SU_{i,t}$ Biaya start up unit ke i saat t

$SD_{i,t}$ Biaya start down unit ke i saat t

$Load_t$ Permintaan beban saat jam ke t

X_i^{on} Lama durasi unit i menyala

X_i^{off} Lama durasi unit i padam

MU_i Minimum nyala unit i

MD_i Minimum padam unit i

$P_{i(\min)}$ Minimum pembangkitan unit i

$P_{i(\max)}$ Maksimum pembangkitan unit i

RD_i Ramp up unit i

RU_i Ramp Down unit i

$P_{i,t(r\min)}$ Min. pembangkitan unit i - RD

$P_{i,t(r\max)}$ Maks. pembangkitan unit i + RU

SR_t Cadangan berputar saat t

$V_{i,t}$ Tegangan bus i saat t

$S_{i,t}$ Lineflow i saat t

2.2 Self Adaptive Modified Firefly Algorithm

Security Firefly Algorithm (FA), pertama kali dibentuk dan dikembangkan oleh dr. Xin-She Yang pada tahun 2007 di cambridge university. FA adalah suatu *meta-heuristic* algoritma optimisasi berbasis pada populasi yang terinspirasi oleh pergerakan kunang-kunang. Sama halnya dengan algoritma PSO yang memperbarui posisi dengan cara memperbarui kecepatan, pada FA kunang-kunang akan bergerak dari kunang-kunang yang satu menuju ke kunang-kunang lain yang memiliki intensitas cahaya lebih terang.

FA terdiri dari sejumlah kunang-kunang dan berbagai macam tingkat kecerahan cahaya. Pada penyelesaian unit commitment jumlah kunang-kunang ini akan merepresentasikan jumlah pembangkit yang ada pada suatu sistem, sedangkan tingkat kecerahannya merepresentasikan fungsi biaya pembangkitan. Pada setiap iterasi akan dipilih fungsi fitness terbaik sebagai $best^k$. Struktur dari masing-masing kunang-kunang dari suatu populasi dapat didefinisikan pada persamaan (10)

$$P_{G,m}^k = [P_1^k, P_2^k, \dots, P_m^k] \quad m = 1, \dots, N_{firefly} \quad (10)$$

Setiap elemen dari kunang-kunang ke- m memodifikasi posisinya ke posisi saat ini, jarak *Cartesian* atau *Euclidean* dari P_m^k untuk masing Modifikasi posisi dihitung berdasarkan persamaan (11).

$$P_m^k = \begin{cases} \text{if } G(P_n^k) < G(P_m^k) \\ (P_m^k + \beta^k (P_n^k - P_m^k) \\ + \alpha^k |P_{\max} - P_{\min}| (rand - \frac{1}{2})) \\ \text{else} \\ P_m^k \end{cases} \quad (11)$$

$$m = 1, \dots, N_{firefly} \quad n = 1, \dots, N_{firefly}$$

Fungsi atraktif β^k dirumuskan melalui persamaan

$$\beta^k = (\beta_{\max}) e^{-\gamma (r_{mn}^k)^2} \quad (12)$$

Untuk menghasilkan perbaikan nilai β^k dengan konvergensi algoritma yang paling cepat dan keluar dari *local optima* β^k dapat ditulis dalam persamaan (13).

$$\beta^k = (\beta_{\max} - \beta_{\min}) e^{-\gamma (r_{mn}^k)^2} + \beta_{\min} \quad (13)$$

$$r_{mn}^k = \|P_n^k - P_m^k\|$$

$$r_{mn}^k = \sqrt{\sum_{i=1}^{NG} (P_{n,i}^k - P_{m,i}^k)^2}$$

Besarnya β_{\max} dan γ diset dengan nilai 1. Sedangkan β_{\min} mempunyai nilai konstan sebesar 0,2 [10][11][12].

2.2.1. Strategi Mutasi

Untuk meningkatkan performa dari FA yang asli dan agar keluar dari *local optima*, terciptalah strategi mutasi yang baru. Di setiap tahap dari masing-masing penyelesaian yang ada, 4 faktor q_1, q_2, q_3 , dan q_4 terpilih secara acak dari populasi yang ada untuk mencakup algoritma pada area tersebut dengan seksama. Akan tetapi untuk ke-empat vektor yang telah disebutkan tadi memiliki ketentuan $q_1 \neq q_2 \neq q_3 \neq q_4 \neq m$ membentuk suatu mutasi individu $P_{G,m,mut}^k$ yang dihasilkan dari persamaan (14)

$$P_{G,m,mut}^k = P_{G,q_1,firefly}^k + \begin{pmatrix} rand1(.) (1 - rand2(.)) \begin{pmatrix} P_{G,q_2,firefly}^k \\ -P_{G,q_3,firefly}^k \end{pmatrix} \\ + rand3(.) (1 - rand4(.)) (best^k - P_{G,q_3,firefly}^k) \end{pmatrix} \quad (14)$$

Vektor mutasi digabungkan dengan $P_{G,m,firefly}^k$ yang menghasilkan $P_{G,m,new}^k$ yang dirumuskan melalui persamaan (15) sampai (17).

$$P_{G,m,mut}^k = [P_{1,mut}^k, P_{2,mut}^k, \dots, P_{N_{firefly,mut}}^k] \quad (15)$$

$$P_{G,m,new}^k = [P_{1,new}^k, P_{2,new}^k, \dots, P_{N_{firefly,new}}^k] \quad (16)$$

$$P_{m,new}^k = [P_{1,new}^k, P_{2,new}^k, \dots, P_{NG,new}^k] \\ m = 1, \dots, N_{firefly} \\ P_{m,i,new}^k = \begin{cases} P_{m,i,new}^k & \text{if } (rand5(.) \leq rand6(.)) \\ P_{m,i,firefly}^k & \text{else} \end{cases} \quad (17)$$

$$m = 1, \dots, N_{firefly} \quad i = 1, \dots, NG$$

Penyelesaian baru akan menggantikan solusi yang asli berdasarkan fungsi fitnessnya melalui (18)

$$P_{m,firefly}^k = \begin{cases} P_{m,new}^k & \text{if } (G(P_{m,new}^k) \leq G(P_{m,firefly}^k)) \\ P_{m,firefly}^k & \text{else} \end{cases} \quad (18)$$

$$m = 1, \dots, N_{firefly}$$

Hasil akhir dari keluaran iterasi ke- k akan dijadikan masukan pada iterasi selanjutnya.

Dapat dikatakan bahwa kriteria pemilihan strategi mutasi ini memiliki perbedaan karakteristik yang mencakup kondisi yang berbeda. Hal ini akan memperbaiki penyelesaian dalam skala besar dan kenaikan perbedaan populasi, selain itu juga akan meningkatkan kemampuan pencarian secara global dan local. $best^k$ digunakan sebagai pemicu untuk memandu perpindahan informasi di antara kunang-kunang yang ada dengan cara yang lebih baik. Akan tetapi, penggabungan yang terlalu cepat mungkin dapat terjadi dalam penyelesaian *local optima* yang sangat besar. Dengan begitu, penggunaan tipe ini pada penghasil fungsi acak dan pemilihan empat kunang-kunang untuk operasi mutasi pada metode SAMFA akan menghasilkan kecepatan tinggi dan terhindar dari terjebak pada *local optima* pada permasalahan yang lebih kompleks [8]. 2012).

2.2.2. Tuning α dan γ Secara *Adaptive* dan *Self-Adaptive*

Terdapat tiga parameter yang digunakan untuk menghasilkan performa FA yang lebih baik. Faktor perpindahan acak (α) mengatur dampak gerakan acak dari penyelesaian yang ada dan biasanya dipilih *range* [0, 1]. Koefisien ini sangat efektif untuk meningkatkan performa FA. Nilai α yang besar akan membuat perpindahan untuk mengeksplorasi solusi yang ada melalui jarak dari tempat yang dicari. Sedangkan nilai α yang lebih kecil akan memudahkan pencarian local. Koefisien absorpsi (γ) mengatur pengurangan intensitas cahaya dan biasanya dimulai dari 1. Dengan begitu, perubahan parameter FA ini akan mengubah efektifitas dari algoritma dan pemilihan parameter-parameter ini akan berdampak besar pada performansi dari FA. Pemilihan parameter *adaptive* dan *self-adaptive* direpresentasikan untuk memodifikasi nilai parameter yang ada untuk performa algoritma yang lebih baik. Pada prosedur pengaturan *adaptive*, percobaan dalam sekian kali running algoritma menyebabkan sebuah fungsi heuristic yang dinamik berubah setiap iterasinya. Misal, inertia weight factor PSO yang berkurang sejalanannya waktu. Sedangkan dalam pengaturan *self-adaptive*, parameter-parameter algoritma dikombinasikan dengan demonstrasi dari masing-masing individu dan dipasang sesuai

dengan penyelesaian proses optimisasi. Dalam penelitian ini, penambahan pemilihan *adaptive* α^k , koefisien absorpsi γ diatur secara *self-adaptive* untuk meningkatkan efektifitas dari pendekatan metode. α^k dipasang secara dinamis pada setiap iterasi menggunakan persamaan (19)

$$\alpha^{k+1} = \left(\frac{1}{2^{k_{max}}}\right)^{1/k_{max}} \alpha^k \quad (19)$$

Dalam penelitian ini, γ tidak diset dengan nilai konstan seperti original FA. sebagai gantinya, γ digabungkan dalam masalah minimalisasi seperti penjelasan di bawah ini.

Setiap posisi kunang-kunang akan diterima untuk dipasangkan dengan parameternya sendiri melalui prosedur *self-adaptive* dengan menerapkan proses yang sama dengan yang digunakan dalam FA, seperti yang ditunjukkan oleh persamaan (19). Variabel baru ini dihubungkan dengan vektor unit pembangkit dengan posisi P_m^k . Vektor posisi yang baru untuk elemen ini akan dirumuskan pada persamaan (20)

$$P_m^{k, self-adaptive} = [P_{m,1}^k \quad P_{m,2}^k \quad \dots \quad P_{m,NG}^k \quad \gamma_{m,NG+1}^k] \quad (20)$$

$m = 1, \dots, N_{firefly}$

Hal ini seharusnya ditunjukkan bahwa variabel yang baru tidak dimasukkan ke dalam penjumlahan tingkat kecerahan pada masing-masing kunang-kunang, tetapi dipasang secara *self-adaptive* untuk meningkatkan perubahan karakteristik dari kunang-kunang dalam masalah pencarian posisi dan mengurangi resiko terjebak dalam *local minima*. Selain itu, dalam persamaan (20) dirumuskan berdasarkan peningkatan dimensi masing-masing elemen seperti ditunjukkan dalam persamaan (21).

$$P_{m,firefly}^{k, self-adaptive} = \begin{cases} \text{if } G(P_{n,firefly}^k) < G(P_{m,firefly}^k) \\ P_{m,firefly}^{k, self-adaptive} \\ + \beta^k (P_{n,firefly}^{k, self-adaptive} - P_{m,firefly}^{k, self-adaptive}) + \\ \alpha^k |P_{max, self-adaptive} - P_{min, self-adaptive}| \\ \left(rand_{1 \times NG+1}(\cdot) - \frac{1}{2} \right) \\ \text{else} \\ P_{m,firefly}^{k, self-adaptive} \end{cases} \quad (21)$$

$m = 1, \dots, N_{firefly} \quad n = 1, \dots, N_{firefly}$

Dari penjelasan pada sub bab 2.2.1 dan 2.2.2 di atas, FA akan dimodifikasi sedemikian rupa hingga membentuk algoritma baru dengan nama SAMFA.

III. IMPLEMENTASI SAMFA PADA SCUC

Pada permasalahan UC, angka 0 dan 1 digunakan untuk mengidikasi bahwa status unit ON/OFF. Sedangkan algoritma FA yang digunakan pada dasarnya adalah sebuah algoritma bilangan real. Oleh karena itu, beberapa modifikasi diperlukan untuk menangani masalah optimasi kode-biner.

Dalam *Binary Self Adaptive Modified Firefly Algorithm* (BSAMFA) yang diusulkan, nilai $P_{G,m}^k$ pada persamaan (10) hanya diperbolehkan dengan nilai 0 atau 1. Untuk menentukan status unit bernilai 0 atau 1, sebuah ambang batas harus ditentukan dengan nilai tetap. Apabila $P_{G,m}^k >$ ambang maka status unit adalah “1”, jika $P_{G,m}^k <$ ambang maka status unit adalah “0”. Ambang batas dapat dibuat dengan range antara 0 hingga 1. Untuk mencapai ini, umumnya menggunakan sebuah fungsi sigmoid

$$f(P_{G,m}^k) = \frac{1}{1 + \exp(-P_{G,m}^k)} \quad (22)$$

Algoritma BFA merupakan metode yang akan digunakan pada program utama dalam menyelesaikan penjadwalan UC. BFA akan menentukan kombinasi nyala-mati unit pembangkit dalam beberapa rentang waktu tertentu untuk memenuhi permintaan beban. Sedangkan *real coded firefly algorithm* (RCFA) digunakan untuk menyelesaikan permasalahan security constrained optimal power flow (SCOPF). Langkah-langkah yang disajikan dalam makalah ini terdapat pada sub bab 3.1

3.1 Strategi perbaikan pada Binary FA

Setiap kali status komitmen untuk setiap interval waktu dibuat secara acak atau jika posisi partikel dimodifikasi, pelanggaran batasan minimum up/down, dan batasan cadangan berputar harus diperiksa sebagai berikut.

Langkah 1: Jika cadangan berputar terpenuhi. Lanjutkan ke langkah 3. Jika tidak, Lanjutkan ke langkah berikutnya.

Langkah 2: Lakukan Pinalti Faktor, dengan memberikan nilai biaya yang tinggi.

Langkah 3: Cek batasan minimum up/down pada masing masing unit. Jika ada pelanggaran pada batasan minimum up atau minimum down, maka skema perbaikan dilakukan untuk mengatasi pelanggaran ini. Cara yang digunakan cukup sederhana, hanya dengan memaksa nilai logika setiap kali batasan MU atau MD dilanggar [7].

Langkah 4: Strategi perbaikan pada langkah 3 akan mempengaruhi batasan cadangan berputar pada sistem. Jika batasan terpenuhi, maka solusi dianggap layak. Jika tidak ke langkah 2.

3.2 Strategi perbaikan pada Real Coded FA

Setiap kali posisi partikel dimodifikasi oleh algoritma FA, pelanggaran batasan kapasitas unit pembangkit dan batasan line harus diperiksa sebagai berikut:

Langkah 1: Jika daya yang dibangkitkan dan batasan line terpenuhi, lanjutkan ke langkah 3. Jika tidak, lanjutkan ke langkah berikutnya.

Langkah 2: Lakukan pinalti faktor, dengan memberikan nilai biaya yang tinggi.

Langkah 3: Selesai.

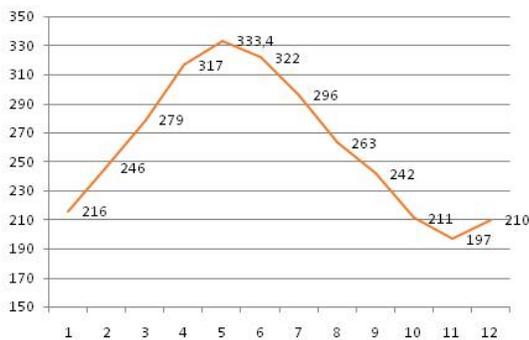
IV. SIMULASI DAN ANALISA

Simulasi dilakukan dengan program MATLAB R2010a menggunakan komputer dengan spesifikasi Windows 7 64-bit, Intel Core i5 2,4 GHz, 4GB RAM. Sistem yang digunakan adalah sistem IEEE 30 bus dengan 6 generator thermal. Fungsi biaya generator, biaya start up, karakteristik generator dapat dilihat pada Tabel 4.1, data pembebanan tiap jam terdapat pada Gambar 4.1. Sedangkan setting dari firefly adalah sebagai berikut:

- Populasi Firefly : 30
- Maksimum Iterasi : 50
- Dimensi = N (Maks. Unit) : 6
- Alpha : 0,1
- Gamma, Betamin : 0,02

Tabel 4.1 Data Karakteristik Generator

	Unit					
	1	2	3	4	5	6
a (\$/h)	0,00375	0,01750	0,06250	0,00834	0,02500	0,02500
b (\$/MWh)	2,0	1,75	1,0	3,25	3	3
c (\$/MWh ²)	0	0	0	0	0	0
Pmin (MW)	50	20	15	10	10	12
Pmax (MW)	200	80	50	35	30	40
Qmin (MVAR)	-	-20	-15	-15	-10	-15
Qmax (MVAR)	-	100	80	60	50	60
MU (h)	4	2	2	2	2	2
MD (h)	4	3	2	2	2	2
RU (MW/ h)	15	10	6	4	4	5
RD (MW/ h)	20	15	10	8	8	10
Shut Down Cost (\$)	50	60	30	85	52	30
Hot start Cost (\$)	70	74	110	50	72	40
Cold start Cost (\$)	176	187	113	267	180	113
Initial state (h)	-4	-3	-3	-3	-2	-2



Gambar 4.1 Grafik permintaan beban daya aktif

Untuk mengetahui efektifitas dari penjadwalan menggunakan SCUC, maka program dijalankan dengan dua cara, pertama simulasi dilakukan tanpa batasan keamanan (UC

Biasa) dan kedua dengan batasan keamanan (SCUC). Dari simulasi yang dilakukan, hasil yang didapatkan untuk penjadwalan UC dan SCUC terlihat sama seperti ditunjukkan pada tabel 4.2.

Tabel 4.2 Hasil Penjadwalan UC dan SCUC

Jam Ke	Unit					
	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	1	1	1	1	1	1
6	1	1	1	1	1	1
7	1	1	1	1	1	1
8	1	1	1	1	1	1
9	1	1	1	1	1	1
10	1	1	1	1	1	1
11	1	1	1	1	1	1
12	1	1	1	1	1	1

Sedangkan daya pembangkitan masing masing unit untuk UC maupun SCUC adalah sebagai berikut:

Tabel 4.3 Daya Pembangkitan dan Biaya dari UC

Jam Ke	Unit (MW)						Biaya Total (\$)
	1	2	3	4	5	6	
1	120	48	30	21	18	24	593,0614
2	120	48	30	21	18	24	593,0438
3	120	57,9	34,6	22,4	22	28,9	649,6494
4	135	65,3	40,3	26,4	26	33,9	753,5584
5	134,6	67,9	41,6	30,4	30	38,1	789,0575
6	132,7	67,4	39,9	34,4	30	35,8	782,1244
7	120	61	37,5	26,2	23,6	34,2	687,0204
8	120	48	30	21	18	24,2	593,4145
9	120	48	30	21	18	24	593,0614
10	120	48	30	21	18	24	593,0614
11	120	48	30	21	18	24	593,0614
12	120	48	30	21	18	24	593,0614
Total Biaya							7813,1754

Tabel 4.4 Daya Pembangkitan dan Biaya dari SCUC Normal Condition

Jam Ke	Unit (MW)						Biaya Total (\$)
	1	2	3	4	5	6	
1	120,0	48,0	30,0	21,0	18,0	24,0	593,0614
2	120,0	48,0	30,0	21,0	18,0	24,0	593,0438
3	129,8	58,0	33,9	24,9	20,9	26,3	674,0547
4	133,0	68,0	39,9	28,9	24,9	31,3	750,4443
5	140,0	75,4	43,4	32,1	28,9	33,6	819,4391
6	143,1	74,3	40,8	35,0	29,6	35,0	831,9514
7	120,0	69,2	38,4	27,0	22,4	32,6	705,3109
8	120,0	54,2	30,0	21,0	18,0	24,0	607,8199
9	120,0	48,0	30,0	21,0	18,0	24,0	593,0614
10	120,0	48,0	30,0	21,0	18,0	24,0	593,0614
11	120,0	48,0	30,0	21,0	18,0	24,0	593,0614
12	120,0	48,0	30,0	21,0	18,0	24,0	593,0614
Total Biaya							7947,3711

Tabel 4.5 Daya Pembangkitan dan Biaya dari SCUC Under Contingency

Jam Ke	Unit (MW)						Biaya Total (\$)
	1	2	3	4	5	6	
1	120	48	30	21	18	24	593,0614
2	120	48	33,5	21	18	26,5	606,3836
3	135	54,5	43,1	24,5	21,4	29,1	708,5908
4	150	64,5	49,1	28,5	24,9	34,1	820,7299
5	154,4	65,2	49,7	32,5	28,8	37,4	863,0451
6	162,6	69,9	49,2	32,8	28,4	35,6	895,1595
7	142,6	54,9	48,8	24,8	20,4	30,6	748,1751
8	128,9	48	38,8	21	18	24	638,9575
9	120	48	30	21	18	24	593,0438
10	120	48	30	21	18	24	593,0438
11	120	48	30	21	18	24	593,0438
12	120	48	30	21	18	24	593,0438
Total Biaya							8246,2781

Pada simulasi SCUC Contingency, sistem diasumsikan mengalami kontingensi. Kontingensi dipilih seolah olah salah satu saluran lepas. Saluran yang dianggap lepas pada simulasi ini adalah saluran bus 2 ke bus 5.

Dari semua simulasi yang dilakukan pada sistem IEEE 30 bus dengan 6 unit pembangkit, terlihat bahwa terdapat selisih biaya pembangkitan antara UC, SCUC Normal, dan SCUC Kontingensi. Perbedaan tersebut dapat dilihat pada tabel 4.6

Tabel 4.6 Hasil Perbandingan

Problem	TCP (\$)
UC	7813,1750
SCUC Normal	7947,3710
SCUC Kontingensi	8246,2780

V. KESIMPULAN

Makalah ini memperkenalkan sebuah metode firefly algorithm dalam menyelesaikan permasalahan security constrained unit commitment. Algoritma sudah dites pada sistem IEEE 30 Bus dengan 6 generator termal selama periode beban 12 jam. Hasil menunjukkan bahwa metode ini mampu digunakan untuk menyelesaikan permasalahan SCUC. Total biaya produksi yang lebih tinggi sebesar \$195,65 adalah sebagai konsekuensi dari permasalahan ini, namun nilai ini memastikan bahwa sistem dapat melayani permintaan beban secara layak dan aman. Untuk dapat mengetahui performa dari usulan algoritma ini sebaiknya dilakukan perbandingan dengan metode algoritma metaheuristik yang lain serta dapat menambahkan case lain semisal ketika terjadi kontingensi sistem. Penambahan metode optimasi dibagian sub problem (losses) mungkin diperlukan agar program lebih mendekati nilai yang sesungguhnya.

DAFTAR PUSTAKA

- [1] Allen J. Wood, Power Generation Operation and Control, second edisi tahun 1996.
- [2] C, Cristopher Columbus dan Sishaj P. Simon, "Hybrid Particle Swarm Approach for Security

- Constrained Unit Commitment", International Conference on Computing, Electronics and Electrical Technology [ICCEET]. 2012
- [3] K. Chandrasekaran dan Sishaj P. Simon, "Binary Real Coded Firefly Algorithm for Solving Unit Commitment Problem", Elsevier inc, pp 67-84. 2012
- [4] M.A Yoshimura, A.I Cohen, "Branch and Bound Algorithm For Unit Commitment", IEEE Trans. On Power App. Syst 102 pp 444-451, 1983
- [5] Nalan Laotumyingyong and Parnjit Damrongkulkamjorn, "Security Constrained Unit Commitment using Mixed-Integer Programming with Bender Decomposition", IEEE Conference Publication, pp 626-630 2010
- [6] Priyanto, Yun Tonce Kusuma. "Combine Active Reactive Dispatch Multiobjective Optimal Power Flow Using Firefly Algorithm", Master Theses of Electrical Engineering, RTE 621.319 Pri p, 2013
- [7] T.O Ting, M. V. C. Rao and C.K. Loo, "A Novel Approach for Unit Commitment Problem via an Effective hybrid Particle Swarm Optimization", IEEE Trans. On Power App. Syst Vol 21, No 1, 2006
- [8] Taher Niknam, Rasoul Azizpanah-Abarghoee, and Alireza Roosta, "Reserve Constrained Dynamic Economic Dispatch: A New Fast *Self-adaptive modified firefly algorithm*", IEEE Systems Journal, Vol. 6, No. 4, December 2012
- [9] V. S Pappala and I. Erlich, "A New Approach for Solving Unit Commitment Problem by Adaptive Particle Swarm Optimization", IEEE Conference Publications, pp 1-6, 2008
- [10] Xin-She Yang, Firefly algorithm, Levy flights and global optimization, in: Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, pp. 209-218, 2010
- [11] Xin-She Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimisation", Int. J. Bio-Inspired Computation, Vol. 2, No. 2, 2010
- [12] Xin-She Yang, "Firefly Algorithm for Multimodal Optimization, in: stochastic Algorithms: foundations and Applications", SAGA 2009, Lecture Notes in Computer sciences, Vol. 5792, pp. 169-178, 2009.