

SKRIPSI

**PENERAPAN ALGORITMA *SPIDER-MONKEY OPTIMIZATION*
UNTUK PENYELESAIAN *SOFTWARE PROJECT SCHEDULING*
*PROBLEM***

***APPLICATION OF THE SPIDER-MONKEY OPTIMIZATION
ALGORITHM TO SOLVE SOFTWARE PROJECT SCHEDULING
PROBLEMS***



NOVIAR IRIANSYA

D0219013

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SULAWESI BARAT
MAJENE
2023**

HALAMAN JUDUL

**PENERAPAN ALGORITMA *SPIDER MONKEY OPTIMIZATION*
UNTUK PENYELESAIAN *SOFTWARE PROJECT SCHEDULING*
*PROBLEM***

***APPLICATION OF THE SPIDER MONKEY OPTIMIZATION
ALGORITHM TO SOLVE SOFTWARE PROJECT SCHEDULING
PROBLEMS***

Diajukan untuk melengkapi persyaratan
menyelesaikan tugas akhir mahasiswa



NOVIAR IRIANSYA

D0219013

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SULAWESI BARAT
MAJENE
2023**

HALAMAN PERSETUJUAN
SKRIPSI

PENERAPAN ALGORITMA *SPIDER MONKEY OPTIMIZATION*
UNTUK PENYELESAIAN *SOFTWARE PROJECT SCHEDULING*
PROBLEM

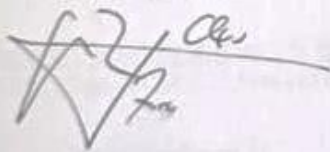
Diusulkan oleh

NOVIAR IRIANSYA
D0219013

Telah disetujui

Pada tanggal 30 Maret 2023

Pembimbing I



Ir. Sugiarto Cokrowibowo, S.Si., M.T.
NIP.198605242015041004

Pembimbing II



Armita Irianti, S.Si., M.Si
NIP.1987080602018032001

HALAMAN PENGESAHAN

SKRIPSI

**PENERAPAN ALGORITMA *SPIDER MONKEY OPTIMIZATION*
UNTUK PENYELESAIAN *SOFTWARE PROJECT SCHEDULING*
*PROBLEM***

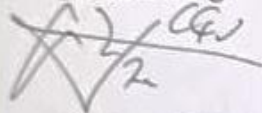
Telah disiapkan dan disusun oleh

**NOVIAR IRIANSYA
D0219013**

Telah dipertahankan di depan Tim Penguji
Pada Tanggal 27 Juli 2023

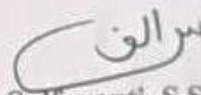
Susunan Tim Penguji

Pembimbing I



Ir. Sugiarto Cokrowibowo, S.Si., M.T.
NIP.198605242015041004

Penguji I



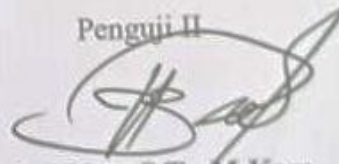
Dr. Eng. Sulfavanti, S.Si., M.T
NIP.198903172020122011

Pembimbing II



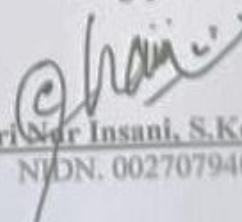
Arnita Irianti, S.Si., M.Si
NIP.1987080602018032001

Penguji II



Nahya Nur, S.T., M.Kom
NIP. 199111052019032024

Penguji III



Chairi Nur Insani, S.Kom., M.T
NIDN. 0027079404

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjaan disuatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Majene, 27 Juli 2023



Noviar Iriansya
NIM. D0219013

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Bismillahirrahmanirrahim

Puji syukur atas kehadiran Allah SWT Yang maha pengasih lagi Maha Penyayang, karena dengan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini dengan judul **“PENERAPAN ALGORITMA SPIDER MONKEY OPTIMIZATION UNTUK PENYELESAIAN SOFTWARE PROJECT SCHEDULING PROBLEM”** yang merupakan tugas akhir dalam melengkapi persyaratan menyelesaikan ujian akhir mahasiswa pada Program Studi Informatika Fakultas Teknik Universitas Sulawesi Barat.

Perjalanan dalam meraih ilmu pengetahuan selama ini merupakan pengalaman yang sangat berharga. Penulis tidak dapat memungkiri bahwa apa yang diperoleh selama ini adalah perjuangan bersama. Sejatinya keberhasilan dan kesuksesan ini tidak lepas dari berbagai dukungan dan peran dari berbagai elemen yang terlibat didalamnya, baik itu pembimbing, orang-tua, para dosen dan teman-teman.

Dalam kesempatan ini pula, penulis menyadari bahwa penyusunan skripsi ini tidak akan terselesaikan tanpa dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Orang-tua dan keluarga yang telah memberikan bantuan materil, moril, doa dan motivasi serta dukungan yang tiada hentinya kepada penulis.
2. Ibu Dr. Ir. Hafsah Nirwana, M.T., selaku Dekan Fakultas Teknik Universitas Sulawesi Barat (UNSULBAR).
3. Bapak Ir. Sugiarto Cokrowibowo, S.Si., M.T., selaku Wakil Dekan Fakultas Teknik Sulawesi Barat dan selaku pembimbing I penulis yang telah meluangkan waktu dan

pemikirannya untuk memberikan bimbingan, arahan, dan petunjuk bagi penulis dalam penyusunan skripsi ini.

4. Ibu Arnita Irianti, S.Si., M.Si selaku pembimbing II penulis yang telah meluangkan waktu dan pemikirannya untuk memberikan bimbingan, arahan, dan petunjuk bagi penulis dalam penyusunan skripsi ini.
5. Bapak Muh. Fahmi Rustan, S.Kom., M.T., selaku Ketua Program Studi Informatika dan selaku pembimbing akademik penulis.
6. Segenap Dosen dan seluruh Staf akademik akademik yang selalu membantu dalam memberikan fasilitas, ilmu, serta pendidikan pada penulis hingga dapat menunjang dalam penyelesaian skripsi ini.
7. Segenap teman-teman dan saudara seperjuangan Informatika A19 yang telah memberikan banyak motivasi dan dukungan bagi penulis.

Majene, 27 Juli 2023

Noviar Iriansya
NIM. D0219013

ABSTRAK

Manajemen proyek adalah sebuah aktivitas kunci perusahaan atau suatu organisasi dalam pengembangan sebuah proyek. Kesulitan utama dalam aktivitas ini adalah sulitnya membagi alokasi pekerja terhadap pekerjaan yang ada sehingga akan berakibat pada durasi dan biaya pengerjaan proyek yang dikeluarkan nantinya. Masalah inilah yang disebut dengan *Software Project Scheduling Problem* (SPSP) atau masalah penjadwalan proyek perangkat lunak. Tujuan dari penelitian ini adalah untuk mengimplementasikan sebuah algoritma optimasi untuk mencoba menyelesaikan SPSP ini. Algoritma optimasi yang digunakan dalam penelitian ini adalah algoritma *Spider Monkey Optimization* (SMO). Berdasarkan hasil pengujian yang telah dilakukan ditarik kesimpulan bahwa nilai parameter *perturbation rate* (pr), jumlah populasi sangat berpengaruh pada hasil *fitness* dan *cost* dan *duration* yang dihasilkan.

Kata kunci : *Software Project Scheduling Problem, SPSP, Spider Monkey Optimization, SMO* Algoritma Optimasi.

BAB I

PENDAHULUAN

A. Latar Belakang

Manajemen proyek adalah sebuah aktivitas kunci dalam perusahaan atau sebuah organisasi pengembangan proyek (Vega-velázquez et al., 2018). Kesalahan dalam aktivitas ini adalah seringkali memiliki biaya yang terlampaui tinggi, dan terkadang, dalam memutuskan pembagian tugas terhadap aktivitas yang berbeda-beda tergolong salah satu tugas paling penting selama perencanaan sebuah proyek. Beberapa dari aktivitas yang dilakukan sebagai bagian dari fase perencanaan proyek adalah salah satunya menentukan waktu tugas yang diberikan ke sumber daya (pekerja) dan kemudian memperkirakan tanggal penghentian dan biaya yang terkait dengan proyek (Jalote, 2003). Salah satu dari jenis proyek ini adalah *Software project scheduling problem* atau proyek pengembangan perangkat lunak.

Salah satu tujuan paling penting ketika membuat perencanaan suatu proyek tentu saja adalah meminimalkan biaya dan durasi proyek, karena seringkali anggaran terbatas dan ada tenggat waktu untuk menyelesaikan proyek tersebut. Durasi dan biaya, bagaimanapun, adalah dua tujuan yang biasanya saling bertentangan, karena ketika satu dikurangi, yang lain biasanya meningkat (Chicano et al., 2011). Masalah inilah yang disebut dalam literatur sebagai *Software project scheduling problem (SPSP)* atau Masalah Penjadwalan Proyek Perangkat Lunak.

Tujuan dalam menyelesaikan SPSP biasanya adalah mencari solusi dalam menentukan pembagian karyawan untuk tugas tertentu sehingga biaya dan durasi proyek bisa diminimalkan. Karena hal ini seringkali bertentangan, yaitu ketika meminimalisasi salah satunya maka yang lain akan sebaliknya (merosot), dari proses

optimasi inilah kemudian akan menghasilkan serangkaian solusi yang bisa mewakili tujuan yang ingin dicapai (Vega-velázquez et al., 2018).

SPSP merupakan masalah yang digolongkan kedalam kategori *NP-Hard Problem*, yang mana ini merupakan masalah yang sama tingkatannya dengan masalah *NP-Complete* atau lebih sulit lagi (Vega-velázquez et al., 2018), bahkan beberapa literatur mengatakan bahwa *NP-Hard* merupakan masalah dengan tingkatan paling sulit atau bahkan tidak ada solusinya (Shapiro & Delgado-eckert, 2012). Terdapat beberapa metode yang sudah digunakan dalam mencoba menyelesaikan masalah ini, salah satunya dari penelitian yang telah dilakukan oleh Jing Shiao dan kawannya (Xiao et al., 2013) yang mencoba menerapkan algoritma *Ant-Colony Optimization (ACO)*. Dalam penelitiannya, mereka membandingkan algoritma *Ant-Colony Optimization* dengan algoritma Genetika (GA) dalam menyelesaikan SPSP selama 30 kali percobaan. Hasilnya percobaan menunjukkan bahwa algoritma ACO memperoleh hit rate dan akurasi yang lebih tinggi dibandingkan dengan solusi dari algoritma genetika, namun masih jauh dari target yang ingin dicapai (Vega-velázquez et al., 2018). Sudah banyak solusi yang coba ditawarkan oleh para peneliti dalam menyelesaikan SPSP ini, namun belum ada solusi yang benar-benar optimal untuk menyelesaikan masalah ini (Nigar, 2020). Alasan inilah yang mendorong penelitian ini, untuk mencoba sebuah algoritma optimisasi yang masih belum lama digunakan didalam dunia penelitian, yaitu algoritma yang disebut *Spider Monkey Optimization (SMO)* atau algoritma monyet laba-laba yang mana merupakan sebuah algoritma *Swarm Intelligence (SI)* yang telah diadaptasi dari perilaku alam.

Alam merupakan inspirasi besar bagi perkembangan umat manusia, sejak zaman dahulu. Ketika umat manusia belajar untuk melihat, mengamati, dan menganalisis aktivitas alam, mereka menyadari bahwa alam adalah sistem dengan kompleksitas yang

sangat besar dan selalu menghasilkan solusi yang mendekati optimal (Agrawal, 2017). Bidang komputasi yang terinspirasi dari alam telah mendapatkan popularitas selama 50 tahun terakhir. Algoritma yang dihasilkan dalam bidang ini seringkali dapat diuraikan dalam penggunaan simulasi komputer untuk mempelajari alam. Algoritma *evolusioner*, algoritma berbasis kecerdasan segerombolan (*Swarm Intelligence*) (SI) dan algoritma berbasis jaringan syaraf tiruan adalah tiga kategori dasar dari algoritma pengoptimalan yang terinspirasi dari alam. *Swarm Intelligence* atau kecerdasan segerombolan adalah metodologi meta-heuristik dibidang Kecerdasan Buatan yang digunakan untuk memecahkan masalah optimisasi kompleks dunia nyata (Vega-velázquez et al., 2018).

Algoritme SI yang dikembangkan ditemukan efektif dalam menyelesaikan berbagai tugas pengoptimalan (Akhand et al., 2019). Dalam rangkaian algoritma berdasarkan kecerdasan segerombolan (*Swarm Intelligence*), salah satu diantaranya adalah metode SMO. SMO adalah metode optimasi numerik yang terinspirasi dari perilaku alami monyet laba-laba (Chand et al., 2013).

Monyet laba- laba adalah spesies yang berasal dari Amerika Selatan, yang hidup berkelompok dan memiliki kecerdasan sosial serta mencari makan di suatu daerah. Monyet laba-laba mencari makanan secara berkelompok yang berada di bawah pemimpin kelompok dan untuk kelompok kecil di bawah subkelompok pemimpin. Mereka mencari makanan dimulai dengan kelompok besar, kemudian dibagi menjadi kelompok yang lebih kecil (subkelompok) ketika mereka gagal menemukan makanan. Pembagian kelompok dimaksudkan untuk melakukan pencarian yang lebih luas tersebar ke berbagai arah untuk menjelajahi suatu daerah. Subkelompok pemimpin dapat membagi kelompok lagi ketika masih gagal mendapatkan makanan. Ketika jumlah kelompok mencapai maksimum, ketua kelompok utama menggabungkan semua subkelompok menjadi kelompok besar kemudian membagi mereka menjadi beberapa

subkelompok (Chand et al., 2013). SMO dimulai dengan populasi awal yang dilakukan secara acak pada posisi populasi monyet. Selanjutnya akan selalu diupdate setiap iterasi melalui interaksi antar monyet. Dalam SMO posisi terbaik dari monyet Laba-laba disebut *Global Leader* dan solusi terbaik dari setiap kelompok kecil disebut *Local Leader* (subkelompok) (Yiqiu et al., 2019).

Algoritma SMO diketahui dapat mengatasi masalah optimisasi yang rumit dengan efisien. Algoritma ini juga memiliki skalabilitas yang baik dan dapat digunakan pada data yang besar serta memiliki akurasi yang akurat dan stabil (Vishwanathan et al., n.d.). beberapa penelitian juga menunjukkan bahwa algoritma SMO terbukti tepat dan efektif untuk digunakan dalam menyelesaikan permasalahan dunia nyata, seperti beberapa hasil penelitian berikut ; SMO lebih baik dari GA dalam menyelesaikan TSP (Firgiawan, 2021). Analisis kinerja SMO menunjukkan bahwa SMO melampaui algoritma *Artificial Bee Colony* (ABC), *Differential Evolution* (DE) dan *Particle Swarm Optimization* (PSO), dalam hal ketergantungan, efektivitas dan presisi (Sharma et al., n.d.). SMO telah diuji dan dibandingkan dengan beberapa algoritma yang lain pada serangkaian TSP dan dinyatakan sebagai metode yang paling cocok untuk memecahkan TSP. Algoritma SMO terbukti cocok untuk sistem layanan-produk (PSS), serta menunjukkan bahwa algoritma ini lebih efektif dibandingkan dengan algoritma optimasi lainnya (Mumtaz et al., 2019)

B. Rumusan Masalah.

Berdasarkan rumusan masalah di atas, maka diperoleh rumusan masalah: Bagaimana hasil implementasi algoritma *Spider Monkey Optimization* (SMO) dalam menyelesaikan Masalah penjadwalan proyek perangkat lunak (SPSP)?

C. Tujuan Penelitian.

Mengetahui hasil implementasi *Spider Monkey Optimization* untuk menyelesaikan masalah penjadwalan proyek perangkat lunak.

D. Manfaat Penelitian

Menjadi rekomendasi bagi pimpinan proyek untuk menyusun jadwal proyek perangkat lunak, dalam hal ini akan memudahkan pimpinan proyek untuk menentukan siapa akan mengerjakan tugas apa dan kapan tugas tersebut akan dikerjakan.

E. Batasan Masalah

Adapun batasan masalah pada penelitian ini, yaitu:

1. Dalam penyusunan solusi, seorang *employee* bisa bergabung untuk mengerjakan sebuah *task* apabila *employee* tersebut memenuhi syarat kebutuhan pada *task*. *Employee* tersebut harus memiliki *skill* yang dibutuhkan untuk menyelesaikan *task*.
2. Tidak boleh ada *employee* yang lembur atau melebihi *max dedication* dalam bekerja, yang direpresentasikan dalam angka $0 \leq \text{max dedication} \leq 1.0$ (maksimal nilai 1).
3. Dataset yang digunakan sebanyak 18 data.
4. *Task Precedence Graph* (TPG) menyatakan *tasks* prayarat yang harus diselesaikan sebelum *task* berikutnya dimulai.

BAB IV

HASIL DAN PEMBAHASAN

A. Hasil dan Pembahasan

Penjadwalan dilakukan untuk mencari biaya dan durasi terpendek dari setiap solusi dengan representasi sebagai nilai *fitness* terbaik atau solusi terbaik. Nilai *fitness* didapatkan setelah melakukan perhitungan biaya dan durasi dari *dedication matrix*, kemudian dimodelkan ke dalam bentuk *ganttt diagram*. Berikut adalah contoh dataset untuk *software project scheduling problem* (SPSP) dapat dilihat pada halaman lampiran.

1. Skenario Pengujian

Pada penelitian ini dilakukan pengujian algoritma merubah nilai parameter algoritma. Parameter yang digunakan yakni: *perturbation rate* (pr), populasi dan iterasi. Nilai pr yang digunakan diantaranya 0.3, 0.5 dan 0.7, ketiga nilai ini diambil untuk mewakili banyaknya kemungkinan nilai pr dari rentang 0.0 – 1.0.

Pengujian dilakukan dengan dataset dt01

(*tracer.lcc.uma.es_problems_psp_instances_inst10-5-10-5.conf*) dengan beberapa skenario sebagai berikut:

a. Skenario 1

Pada skenario pertama, algoritma SMO diuji pada dataset dt01 dengan menggunakan parameter pr 0,3 dengan populasi 100, 200, 300, 400, 500, 600, 700, 800, 900 dan 1000 sebanyak 50.

Skenario 2

Pada skenario pertama, algoritma SMO diuji pada dataset dt01 dengan menggunakan parameter pr 0,5 dengan populasi 100, 200, 300, 400, 500, 600, 700, 800, 900 dan 1000 sebanyak 50.

b. Skenario 3

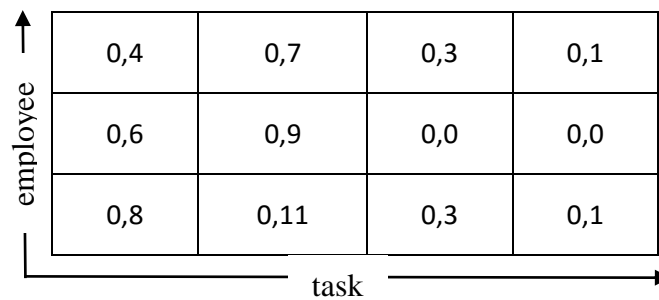
Pada skenario pertama, algoritma SMO diuji pada dataset dt01 dengan menggunakan parameter pr 0,7 dengan populasi 100, 200, 300, 400, 500, 600, 700, 800, 900 dan 1000 sebanyak 50.

2. Model Representasi Solusi

Pada penelitian ini, dalam melakukan pencarian jadwal terbaik dibutuhkan pemodelan *Spider Monkey* yang baik dan memudahkan dalam pembacaannya ke dalam *gantt diagram*. Pemodelan dilakukan dalam bentuk *array* 2 dimensi. *Array* tersebut memuat *task-task* tiap proses. Dapat dilihat pada gambar berikut:

a. Solusi 1:

b. Tabel 4.1 Contoh Pemodelan solusi *Spider Monkey*



| | | | |
|-----|------|-----|-----|
| 0,4 | 0,7 | 0,3 | 0,1 |
| 0,6 | 0,9 | 0,0 | 0,0 |
| 0,8 | 0,11 | 0,3 | 0,1 |

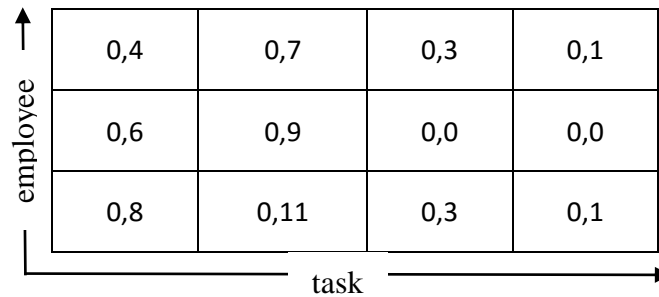
pada tabel 4.1 digambarkan sebuah contoh solusi/*Spider Monkey* dengan membangkitkan nilai *dedication* (nilai elemen matrix) secara acak yang dimodelkan dalam bentuk matrix/array 2 dimensi dimana jumlah baris sama dengan jumlah *employee* dan jumlah kolom sama dengan jumlah *task*.

c. Solusi 2 (*Interaction Operation*):

Untuk solusi 2, diambil sebuah contoh matrix (*Spider Monkey* ke-i) kemudian dilakukan *interaction operation* untuk mendapatkan hasil solusi (*Spider Monkey*) yang baru.

Sebelum *Interaction*:

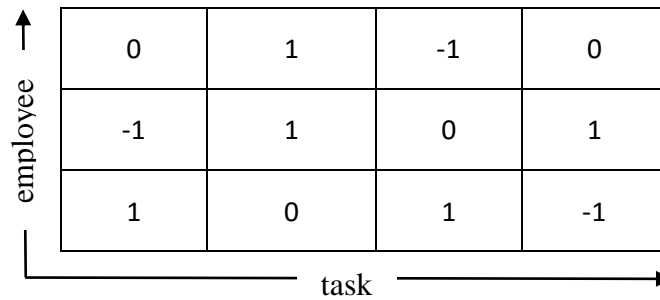
Tabel 4.2 Contoh *Dedication matrix* sebelum *interaction*



| | | | | |
|--|-----|------|-----|-----|
| | 0,4 | 0,7 | 0,3 | 0,1 |
| | 0,6 | 0,9 | 0,0 | 0,0 |
| | 0,8 | 0,11 | 0,3 | 0,1 |

Tabel 4.2 di atas merupakan contoh matrix dengan nilai dedication yang dibangkitkan secara random. Matrix dengan kondisi sebelum diterapkan *interaction operation*.

Tabel 4.3 Contoh Matrix *Interaction Operation*



| | | | | |
|--|----|---|----|----|
| | 0 | 1 | -1 | 0 |
| | -1 | 1 | 0 | 1 |
| | 1 | 0 | 1 | -1 |

Tabel 4.3 di atas merupakan contoh matrix hasil interaksi antar 2 *Spider Monkey*, yang kemudian dinamakan matrix *interaction operation* (IOM). IOM ini berisi nilai dari -1, 0 dan 1. Nilai interaksi = -1 berarti bahwa pada *Spider Monkey* baru nilai X_{ij} -nya akan mengalami penurunan, nilai interaksi = 1 berarti bahwa pada *Spider Monkey* baru nilai X_{ij} -nya akan mengalami peningkatan, sedangkan nilai interaksi = 0 berarti bahwa pada *Spider Monkey* baru nilai X_{ij} -nya tidak mengalami perubahan.

Setelah *Interaction* :

Tabel 4.4 Contoh Solusi baru hasil *Interaction Operation*

| | | | | |
|----------|--------|------|-----|-----|
| ↑ | 0,4 | 0,9 | 0,1 | 0,1 |
| employee | 0,5 | 1,0 | 0,0 | 0,8 |
| ↓ | 1 | 0,11 | 0,7 | 0,0 |
| | task → | | | |

Setelah proses *interaction operation* diterapkan antara *Spider Monkey* (ke-i) dengan IOM, maka didapatkan solusi/*spider monkey* kedua yang digambarkan dalam bentuk contoh matrix pada tabel 4.4 di atas. Tiap solusi kemudian digambarkan dalam matrix *dedication*, *fitness* dan model *gantt diagram*.

Pada setiap solusi/*spider monkey* menghasilkan *dedication matrix* melalui proses solusi 1 atau solusi 2. *Dedication matrix* ini kemudian dihitung untuk mencari *task duration*, *task duration* untuk menghitung *project duration* dan *project cost*, kemudian digambarkan dalam model *gantt diagram*, juga masing-masing solusi yang dihasilkan dari setiap *Spider Monkey* memiliki nilai *fitness*. Nilai *fitness* dari setiap *spider monkey* inilah yang kemudian dijadikan perbandingan dengan *spider monkey* yang lain, *spider monkey* mana yang paling bagus, jika nilai *fitness spider monkey* A lebih besar dari nilai *fitness spider monkey* B, maka *spider monkey* A yang menjadi solusi terbaik.

3. Analisis *Fitness*

Pada pengujian dengan skenario 1, 2, dan skenario 3 dilakukan pengujian dengan membandingkan perubahan parameter populasi dan *perturbation rate* pada algoritma SMO dengan dataset dt01. Hasil dari rata-rata pengujian dipaparkan pada tabel 4.8.

Tabel 4.5 Rata-rata nilai *fitness* pada $pr = 0.3$

| Populasi | <i>fitness</i> |
|-----------------|-----------------------|
| 100 | 1.78859351E-06 |
| 200 | 1.79402678E-06 |
| 300 | 1.79786623E-06 |
| 400 | 1.79828991E-06 |
| 500 | 1.79909974E-06 |
| 600 | 1.80140000E-06 |
| 700 | 1.80000000E-06 |
| 800 | 1.80360000E-06 |
| 900 | 1.80640000E-06 |
| 1000 | 1.80340000E-06 |

Tabel 4.6 Rata-rata nilai *fitness* pada $pr = 0.5$

| Populasi | <i>fitness</i> |
|-----------------|-----------------------|
| 100 | 1.79340000E-06 |
| 200 | 1.79985126E-06 |
| 300 | 1.80262075E-06 |
| 400 | 1.80740000E-06 |
| 500 | 1.80756937E-06 |
| 600 | 1.80760000E-06 |
| 700 | 1.80820000E-06 |
| 800 | 1.80990150E-06 |

| | |
|-------------|----------------|
| 900 | 1.81226802E-06 |
| 1000 | 1.81200000E-06 |

Tabel 4.7 Rata-rata nilai *fitness* pada $pr = 0.7$

| Populasi | <i>fitness</i> |
|-----------------|----------------|
| 100 | 1.79720000E-06 |
| 200 | 1.80640000E-06 |
| 300 | 1.80500000E-06 |
| 400 | 1.81060000E-06 |
| 500 | 1.81420000E-06 |
| 600 | 1.81120000E-06 |
| 700 | 1.81720000E-06 |
| 800 | 1.81280000E-06 |
| 900 | 1.81780000E-06 |
| 1000 | 1.81720000E-06 |

Pada tabel 4.7 pengujian dengan parameter pr 0,3 dengan populasi 100 hingga 1000 memberikan hasil dengan nilai dari rentang 1.78859351E-06 sampai dengan 1.80640000E-06. Pada tabel 4.6 pengujian dengan pr 0,5 dengan populasi yang sama memberikan hasil yang sedikit lebih baik dari skenario pertama, dengan nilai dari rentang 1.79340000E-06 sampai dengan 1.81780000E-06. Sedangkan pada tabel 4.7 pengujian dengan parameter pr 0,7 dengan populasi yang sama memberikan hasil yang lebih baik dari parameter kedua pengujian sebelumnya, dengan nilai dari rentang 1.79720000E-06 sampai dengan 1.81780000E-06.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pada hasil penelitian yang dilakukan, dapat ditarik kesimpulan bahwa parameter *perturbation rate* (pr) pada algoritma SMO sangat berpengaruh terhadap hasil fitness dari solusi yang dihasilkan. Selain parameter pr, parameter yang berpengaruh lainnya adalah populasi, jika populasi ditingkatkan maka ruang pencarian solusi menjadi lebih besar dan kemungkinan untuk menemukan solusi terbaik menjadi lebih besar. Hasil dari penelitian ini juga menghasilkan kesimpulan yang sama dengan penelitian Alba Chicano bahwa semakin banyak *task* maka semakin tinggi *cost* dan *duration* yang dibutuhkan, sedangkan semakin banyak employee maka semakin tinggi cost, namun durasi yang dibutuhkan semakin berkurang untuk mengerjakan proyek. Selain itu, nilai fitness maksimal yang dihasilkan pada setiap skenario pengujian dari 18 buah data adalah 2.68837148E-06 (dt10), yang direpresentasikan sebagai solusi optimal yang dihasilkan dalam penelitian ini.

Sehingga kesimpulan dari semua pengujian yang dilakukan yaitu sistem sudah berjalan dengan baik sesuai dengan yang diharapkan. Namun kedepannya masih dibutuhkan berbagai penyempurnaan agar menghasilkan luaran aplikasi yang lebih baik lagi dan dapat digunakan untuk permasalahan yang lebih kompleks lagi.

B. Saran

Pada penelitian selanjutnya diharapkan dapat mengembangkan algoritma *Spider Monkey Optimization* pada kemampuan ruang pencarian yang lebih ditingkatkan. Untuk penelitian selanjutnya dapat melakukan kombinasi algoritma

Spider Monkey Optimization dengan algoritma genetika atau membandingkan dengan algoritma Swarm Intelligence lainnya dan menguji waktu komputasi untuk setiap algoritma tersebut.

DAFTAR PUSTAKA

- Agrawal, V. (2017). Spider Monkey Optimization : a survey. *International Journal of System Assurance Engineering and Management*. <https://doi.org/10.1007/s13198-017-0685-6>
- Akhand, M. A. H., Ayon, S. I., Shahriyar, S. A., Siddique, N., & Adeli, H. (2019). Journal. *Applied Soft Computing Journal*, 105887. <https://doi.org/10.1016/j.asoc.2019.105887>
- Alba, E., & Chicano, J. F. (2007). *Software project management with GAs*. 177, 2380–2401. <https://doi.org/10.1016/j.ins.2006.12.020>
- Chand, J., Harish, B., Singh, S., & Maurice, J. (2013). *Spider Monkey Optimization algorithm for numerical optimization*. <https://doi.org/10.1007/s12293-013-0128-0>
- Chicano, F., Luna, F., & Nebro, A. J. (2011). *Using Multi-objective Metaheuristics to Solve the Software*. 1915–1922.
- Cirua, A. A. A., & Nurtanio, I. (2022). *Scheduling Job Machines with Swap Sequence to Minimize Makespan Using Spider Monkey Optimization Algorithm*. 291–296.
- Crawford, B., Johnson, F., Soto, R., & Monfroy, E. (2014). A Max-Min Ant System algorithm to solve the Software Projects Scheduling Problem. *EXPERT SYSTEMS WITH APPLICATIONS*. <https://doi.org/10.1016/j.eswa.2014.05.003>
- Elena, M., Patrizia, B., Francesca, B., Erika, G., Bruni, M. E., Beraldi, P., & Guerriero, F. (2013). *A scheduling methodology for dealing with uncertainty in construction projects*. <https://doi.org/10.1108/02644401111179036>

- Firgiawan, W. (2021). *Performance Comparison of Spider Monkey Optimization and Genetic Algorithm for Traveling Salesman Problem*. 191–195.
- Jalote, P. (2003). *Optimal Resource Allocation for the Quality Control Process*.
- Kaur, A., & Chhabra, A. (2018). *An Efficient Deadline Constrained Job Scheduling Using Spider Monkey Optimization*. *13*(11), 10094–10104.
- Mauluddin, S., Indonesia, U. K., Sitanggang, A. S., & Indonesia, U. K. (2018). *Automation Lecture Scheduling Information Services through the Email Auto-Reply Application*. January 2019. <https://doi.org/10.14569/IJACSA.2018.091242>
- Monkeys, R. S., Source, C., & Url, S. (2014). *American Society of Mammalogists*. *16*(3), 171–180.
- Mumtaz, J., Stief, P., Dantan, J., Etienne, A., & Siadat, A. (2019). ScienceDirect
 ScienceDirect Application of an improved Spider Monkey Optimization algorithm for Application of an Monkey Optimization 28th improved CIRP Design Spider Nantes , France algorithm for component assignment problem in PCB assembly component assignment problem in PCB assembly c physical A new methodology to analyze the functional and *, Lei Yue b architecture of Yue b identification existing products for an assembly oriented product family. *Procedia CIRP*, *83*, 266–271.
<https://doi.org/10.1016/j.procir.2019.04.075>
- Nigar, N. (2020). *Multi-objective Dynamic Software Project Scheduling : An Evolutionary Approach for Uncertain Environments*. December.
- Pal, S. S., Kumar, S., & Kashyap, M. (2016). *Multi-level Thresholding Segmentation Approach Based on Spider Monkey Optimization Algorithm*.
<https://doi.org/10.1007/978-81-322-2523-2>
- Pinedo, M. L. (n.d.). *Michael L. Pinedo*.
- Rezende, A. V., Silva, L., Britto, A., & Amaral, R. (2019). *The Journal of Systems and*

- Software Software project scheduling problem in the context of search-based software engineering : A systematic review. *The Journal of Systems & Software*, 155, 43–56. <https://doi.org/10.1016/j.jss.2019.05.024>
- Roosmalen, M. G. M. Van. (n.d.). *Acta amazônica*. 15.
- Shapiro, M., & Delgado-eckert, E. (2012). Mathematical Biosciences Finding the probability of infection in an SIR network is NP -Hard. *Mathematical Biosciences*, 240(2), 77–84. <https://doi.org/10.1016/j.mbs.2012.07.002>
- Sharma, H., Hazrati, G., & Bansal, J. C. (n.d.). *Spider Monkey Optimization Algorithm*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-91341-4>
- Shen, Xiao-ning, Minku, L. L., Marturi, N., Guo, Y., & Han, Y. (2017). US CR. *Information Sciences*. <https://doi.org/10.1016/j.ins.2017.10.041>
- Shen, Xiaoning, Guo, Y., & Li, A. (2020). Cooperative coevolution with an improved resource allocation for large-scale multi-objective software project scheduling. *Applied Soft Computing Journal*, 88, 106059. <https://doi.org/10.1016/j.asoc.2019.106059>
- Singh, S., Sakshi, S., & Harish, S. (2020). Twitter sentiment analysis using hybrid Spider Monkey optimization method. *Evolutionary Intelligence*, 0123456789. <https://doi.org/10.1007/s12065-019-00334-2>
- Tomar, A. S., Dubey, H. M., & Pandit, M. (2019). Spider Monkey Optimization for Economic. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 1(6), 349–354.
- Vega-velázquez, M. Á., García-nájera, A., & Cervantes, H. (2018). AC. *International Journal of Production Economics*. <https://doi.org/10.1016/j.ijpe.2018.04.020>
- Vishwanathan, S. V. N., Sun, Z., & Theera-ampornpant, N. (n.d.). *Multiple Kernel Learning and the SMO Algorithm*. 1–9.

- Wu, X., Consoli, P., & Minku, L. (n.d.). *An Evolutionary Hyper-heuristic for the Software Project Scheduling Problem. 1*, 37–47. <https://doi.org/10.1007/978-3-319-45823-6>
- Xia, X., Liao, W., Zhang, Y., & Peng, X. (2021). A discrete spider monkey optimization for the vehicle routing problem with stochastic demands. *Applied Soft Computing*, *111*, 107676. <https://doi.org/10.1016/j.asoc.2021.107676>
- Xiao, J., Ao, X., & Tang, Y. (2013). Computers & Operations Research Solving software project scheduling problems with ant colony optimization. *Computers and Operation Research*, *40*(1), 33–46. <https://doi.org/10.1016/j.cor.2012.05.007>
- Yiqiu, F., Xia, X., & Junwei, G. (2019). Cloud Computing Task Scheduling Algorithm Based On Improved Genetic Algorithm. *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, *It nec*, 852–856. <https://doi.org/10.1109/ITNEC.2019.8728996>