

OPTIMASI BIAYA PONDASI *BORED PILE* DENGAN METODE METAHEURISTIK

Leonard Chandra¹, Christian Yoga Pramono², Doddy Prayogo³, dan Y. T. Teddy Susanto⁴

^{1,2}Mahasiswa Program Studi Teknik Sipil, Universitas Kristen Petra, Surabaya

^{3,4}Dosen Program Studi Teknik Sipil, Universitas Kristen Petra, Surabaya

¹b11170112@john.petra.ac.id, ²b11170169@john.petra.ac.id, ³prayogo@petra.ac.id,

⁴tsusanto53@gmail.com

ABSTRAK: Pondasi adalah bagian dari struktur bangunan yang berfungsi meneruskan beban struktur atas ke lapisan tanah dengan aman. Sementara pondasi dalam digunakan apabila lapisan tanah keras terletak jauh di bawah permukaan tanah. Untuk mendapatkan desain pondasi dalam yang optimal, terdapat tiga kriteria penting yang harus diperhatikan yaitu *ultimate limit state (ULS)*, *serviceability limit state (SLS)*, dan ekonomis. Penggunaan metode optimasi yang baik dapat membantu menghasilkan dimensi pondasi yang optimal dan ekonomis namun tetap memenuhi syarat aman. Penelitian-penelitian sebelumnya mengindikasikan bahwa metode metaheuristik dapat digunakan sebagai alternatif yang mampu menyelesaikan permasalahan optimasi. Oleh karena itu, penelitian ini menggunakan metode metaheuristik *particle swarm optimization (PSO)* dan *symbiotic organisms search (SOS)* untuk menyelesaikan permasalahan optimasi pondasi dalam. Pada penelitian ini, optimasi pondasi dalam dilakukan terhadap pondasi *bored pile* untuk studi kasus bangunan empat lantai dan bangunan sembilan lantai. Hasil penelitian menunjukkan metode metaheuristik mampu menemukan dimensi pondasi *bored pile* yang optimal.

Kata kunci: optimasi, pondasi dalam, metaheuristik, ekonomis

ABSTRACT: Foundation is part of the building structure to transfer the load from upper structure to a soil layer. A deep foundation is used when the hard soil layer is located far below the soil surface. To design an optimal and economical deep foundation that still meet the safety requirements, *ultimate limit state (ULS)*, *serviceability limit state (SLS)*, and economic are required. Previous studies indicate that metaheuristic methods can be used as an alternative to solve existing optimization problems. Therefore, this study uses metaheuristic methods, that is, *particle swarm optimization (PSO)* and *symbiotic organism search (SOS)* to solve the optimization of deep foundation. In this study, the optimization was carried out for case studies of four- and nine-story buildings which used *bored pile* foundation. The results showed that the metaheuristic methods were able to find the optimal deep foundation dimensions.

Keywords: optimization, deep foundation, metaheuristic, economic

1. PENDAHULUAN

Pondasi adalah bagian dari struktur bangunan yang diharapkan dapat meneruskan beban struktur atas ke lapisan tanah atau batu yang dengan aman (Das, 2010). Pondasi bangunan dapat dibedakan menjadi dua jenis, yaitu pondasi dalam (*deep foundation*) dan pondasi dangkal (*shallow foundation*) (Das, 2015). Pondasi dalam menggunakan *piles* yang dapat dibagi menjadi beberapa jenis, yaitu *conventional steel, concrete, timber, dan composite piles*. Pada umumnya jenis *piles* yang banyak digunakan adalah *concrete piles*. *Concrete piles* dapat dibagi menjadi dua kategori dasar, yaitu *precast piles* dan *cast-in-situ piles*. Pada kategori *cast-in-situ piles*, *piles* dibuat dengan membuat lubang di tanah dan memasukkan campuran beton kedalamnya (Das, 2015). Pada kategori *precast piles*, *piles* dibuat dengan cetakan ukuran tertentu terlebih dahulu dari pabrik sebelum dibawa ke lokasi konstruksi. Sedangkan, pondasi *bored pile* merupakan *cast-in-situ piles* di mana memiliki keuntungan yaitu dapat dibuat dalam diameter yang cukup besar, dapat dibuat kedalaman yang dalam, dapat menembus bebatuan jika menggunakan alat yang sesuai, tidak menyebabkan getaran yang signifikan, dapat menambahkan *bearing capacity* dengan memberikan *grouting* pada bagian bawah *bored pile*, dan dapat terlihat jika mencapai lapisan keras yang diinginkan dengan melihat tanah yang dikeluarkan selama proses pengeboran berlangsung (Toprak et al., 2018).

Terdapat tiga kriteria yang harus dipenuhi dalam mendesain suatu pondasi, yaitu *ultimate limit state* (ULS), *serviceability limit state* (SLS), dan ekonomi. Namun, kriteria ketiga inilah yang menjadi suatu tantangan tersendiri bagi seorang *engineer*. Untuk mendapatkan desain pondasi *bored pile* yang optimal dengan biaya konstruksi yang ekonomis, dibutuhkan adanya suatu optimasi pondasi *bored pile*. Metode metaheuristik merupakan salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan optimasi rumit untuk mendapatkan hasil yang optimal.

Penelitian ini akan menggunakan dua algoritma metaheuristik, yaitu algoritma *particle swarm optimization* (PSO) dan algoritma *symbiotic organisms search* (SOS). Kedua algoritma akan bekerja dengan *penalty function method* untuk mengontrol batasan-batasan yang ada saat melakukan suatu desain sehingga diperoleh hasil yang optimal, ekonomis, dan tetap memenuhi syarat aman.

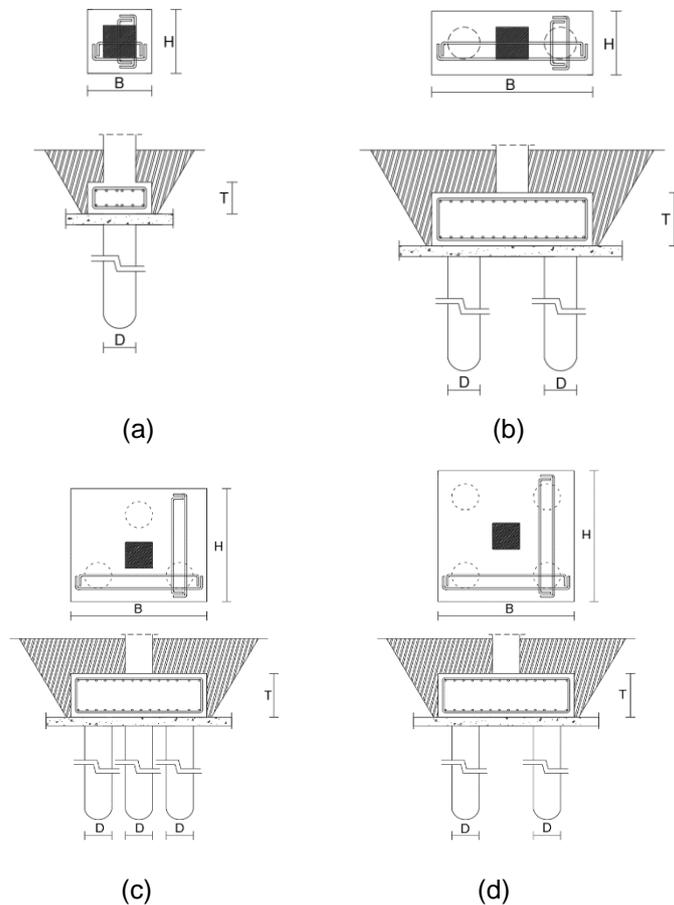
2. TINJAUAN PUSTAKA

2.1 Formulasi *Problem* Optimasi Desain untuk *Bored Pile*

Pada proyek konstruksi, terdapat dua jenis pondasi, yakni pondasi dalam dan pondasi dangkal. Dalam penelitian ini, hal yang diteliti adalah pondasi *bored pile*. Hal yang dipertimbangkan dalam pembuatan pondasi *bored pile* adalah diameter *pile*, dimensi *pile cap*, dan kedalamannya. Semakin besar diameter *pile* dan dimensi *pile cap*, maka biaya yang dikeluarkan semakin besar pula dan begitu pula sebaliknya. Sementara dimensi pondasi *bored pile* tersebut belum tentu efisien untuk menahan beban bangunan di atasnya, sehingga dibutuhkan adanya proses optimasi. Untuk melakukan proses optimasi sehingga tercapainya hasil yang optimal, dari desain suatu struktur dibutuhkan variabel yang akan dioptimasi, *constraint*, dan tujuan atau *objective* yang ingin dicapai dari proses optimasi.

2.2 Variabel Desain Pondasi *Bored Pile*

Variabel-variabel yang digunakan dalam proses optimasi berhubungan dengan diameter *pile* (d), jumlah *pile* (n), *layout*, serta dimensi *pile cap* yang akan digunakan. Variabel desain yang digunakan dapat dilihat pada Gambar 1.



Gambar 1. Pemodelan Pondasi *Bored Pile* dengan Jumlah *Pile*: (a) 1, (b) 2, (c) 3, dan (d) 4.

di mana:

- D = diameter *pile*
- N = jumlah *pile*
- T = ketebalan *pile cap*
- B = panjang *pile cap*
- H = lebar *pile cap*

Variabel D , N , dan T merupakan variabel yang berhubungan dengan dimensi dari pondasi *bored pile*. Parameter desain lain yang digunakan adalah *properties* tanah, beban bangunan, panjang *pile*, mutu beton, mutu tulangan, tebal selimut beton, harga satuan pekerjaan galian, harga satuan pekerjaan bekisting, harga satuan pekerjaan penulangan, harga satuan pekerjaan beton, harga satuan pekerjaan pemadatan kembali. *Properties* tanah yang menjadi parameter desain menggunakan data *standard penetration test* (SPT). Adanya *upper bound* dan *lower bound* bertujuan untuk membatasi variabel desain. *Upper bound* atau batas atas adalah pembatasan nilai maksimum yang dimiliki oleh variabel, sedangkan *lower bound* atau batas bawah merupakan nilai minimum yang dimiliki oleh variabel. *Upper bound* dan *lower bound* yang digunakan dapat dilihat pada Tabel 1.

Tabel 1. *Upper Bound* dan *Lower Bound* untuk Variabel Desain

Parameter	<i>Lower Bound</i> (m)	<i>Upper Bound</i> (m)
D	0.3	0.8
N	1	4
T	0.3	1

2.3 Constraints

Dalam optimasi pondasi *bored pile*, dibutuhkan suatu *constraint*. *Constraint* tersebut digunakan untuk memastikan bahwa desain yang dihasilkan dari proses optimasi tidak mengalami kegagalan. *Constraint* yang digunakan adalah *bearing capacity limit state* dan *shear force of concrete limit state*. Syarat-syarat untuk batasan yang digunakan dalam penelitian ini mengacu pada SNI 8460:2017, SNI 2847:2013, dan *Bearing Capacity Theory*. Dalam SNI 8460:2017, disajikan faktor keamanan minimum untuk pondasi dalam. Pada SNI 2847:2013, diatur syarat pemasangan tulangan *horizontal* dan *vertical* serta perhitungan untuk desain gaya geser dan desain beban lentur. *Bearing capacity theory* digunakan untuk melakukan perhitungan desain struktur berkaitan dengan perhitungan data tanah.

2.4 Objective Function

Objective function merupakan rumusan fungsi yang menggambarkan suatu nilai yang merupakan tujuan dari proses optimasi. Tujuan yang diinginkan dalam penelitian ini adalah pengoptimasian biaya. *Objective function* yang digunakan dirumuskan dalam Persamaan 1.

$$f(X) = C_e V_e + C_{ff} A_{ff} + C_r W_r + C_c V_c + C_b V_b + C_p D + C_{pp} D \quad (1)$$

di mana:

$f(X)$ = *objective function* (Rp)

C_e = biaya penggalian tanah per satuan volume (Rp/m³)

V_e = volume penggalian tanah (m³)

C_{ff} = biaya bekisting *pile cap* per satuan luas (Rp/m²)

A_{ff} = luas bekisting *pile cap* (m²)

C_r = biaya penulangan per satuan massa (Rp/kg)

W_r = massa tulangan baja (kg)

C_c = biaya pengecoran beton per satuan volume (Rp/m³)

V_c = volume pengecoran beton (m³)

C_b = biaya pengurugan dan pemadatan tanah kembali per satuan volume (Rp/m³)

V_b = volume pengurugan dan pemadatan tanah kembali (m³)

C_p = biaya pengeboran *pile* (Rp/m)

D = panjang *pile* (m)

C_{pp} = biaya pembuangan tanah hasil pengeboran (Rp/m)

Selanjutnya, dalam *final objective function* dilakukan eliminasi hasil desain yang tidak memenuhi *constraint* yang ada dengan menggunakan penalti seperti pada Persamaan 2.

$$F(X) = f(X) + r \sum_{i=1}^n \max\{0, g_i(X)\}^l \quad (2)$$

di mana:

$F(X)$ = *final objective function*

r = *penalty factor*

l = *power of penalty function*

$g_i(X)$ = *relative violation function of the constraint*

$f(X)$ = *objective function*

n = *jumlah constraint*

Proses optimasi yang digunakan pada penelitian ini berupa dua algoritma metaheuristik, yaitu PSO dan SOS. PSO adalah metode optimasi global yang diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 berdasarkan penelitian terhadap simulasi perilaku sosial kawanan burung atau sekumpulan ikan di mana terdiri dari tindakan individu dan pengaruh individu lain dalam suatu kelompok (Eberhart & Kennedy, 1995). Inisiasi awal dari PSO adalah penempatan partikel secara acak. Setiap partikel memiliki kecepatan masing-masing. Pergerakan setiap partikel untuk menentukan lokasi selanjutnya dipengaruhi oleh kecepatan partikel saat ini, lokasi terbaik yang pernah ditempati partikel tersebut, lokasi terbaik dari populasi, dan lokasi dari partikel tersebut. Lokasi terbaik dari seluruh partikel akan dipilih menjadi *global solution* yang paling optimal.

SOS diusulkan oleh Cheng dan Prayogo pada tahun 2014 (Cheng & Prayogo, 2014). Algoritma ini mensimulasikan interaksi simbiosis antara makhluk hidup agar dapat bertahan hidup di suatu ekosistem. Simbiosis yang digunakan dalam algoritma SOS adalah simbiosis mutualisme, komensalisme, dan parasitisme. Cara kerja dari algoritma SOS adalah melalui semua fase dengan proses yang terus berulang sampai kriteria terpenuhi. Penerapan simbiosis mutualisme adalah menggantikan organisme lama jika hasil organisme baru lebih baik untuk kemudian digunakan dalam fase komensalisme lalu fase parasitisme.

2.5 Metaheuristik

Metaheuristik adalah pendekatan komputasi untuk penyelesaian suatu kasus optimasi dengan mencari solusi *random* dari skala global kemudian fokus pada solusi terbaik dari iterasi sebelumnya hingga ditemukan solusi yang optimal. Dengan adanya teknik metaheuristik ini, berbagai macam persoalan dapat diselesaikan dengan langkah pengerjaan yang relatif sama. Metaheuristik dapat dimodifikasi agar dapat menyelesaikan masalah khusus yang memiliki banyak *constraint* yaitu dengan menambahkan suatu teknik *constraint handling* di dalamnya. Seiring berkembangnya zaman, semakin banyak metode metaheuristik yang ada seperti *Genetic Algorithm (GA)*, *Ant Colony Optimization (ACO)*, *Particle Swarm Optimization (PSO)*, dan *Symbiotic Organisms Search (SOS)*. Pada penelitian ini, metode metaheuristik yang digunakan adalah PSO dan SOS.

2.5.1 Particle Swarm Optimization

Algoritma PSO diperkenalkan pertama kali oleh Kennedy dan Eberhart pada tahun 1995 (Kennedy & Eberhart, 1995). Algoritma optimasi ini mensimulasikan perilaku sosial kawanan burung atau sekumpulan ikan di mana terdiri dari tindakan individu dan pengaruh individu lain dalam suatu kelompok. Setiap individu berperilaku dipengaruhi oleh perilaku kelompok dan

juga menggunakan kecerdasan sendiri. Contohnya, apabila seekor burung menemukan jalan tercepat menuju sumber makanan, sisa kelompok yang lain juga akan mengikuti meskipun lokasinya jauh.

Pada algoritma PSO, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel (Shi, 2001). Partikel diletakkan secara acak dan memiliki kecepatan awal masing-masing. Untuk menentukan kecepatan partikel selanjutnya dapat digunakan Persamaan 4. Kecepatan partikel dipengaruhi oleh kecepatan partikel saat ini ($V_i(t)$), lokasi terbaik yang pernah ditempati oleh partikel tersebut ($P_i(t)$), lokasi terbaik dari populasi ($P_g(t)$), dan lokasi dari partikel tersebut ($X_i(t)$).

$$v_i(t + 1) = wv_i(t) + r_1c_1(X_{pbest}(t) - X_i(t)) + r_2c_2(X_{gbest}(t) - X_i(t)) \quad (3)$$

Setelah didapatkan kecepatan setiap partikel, maka posisi partikel selanjutnya dapat dihitung dengan:

$$X_i(t + 1) = X_i(t) + v_i(t + 1) \quad (4)$$

Keterangan:

$v_i(t + 1)$	= kecepatan partikel
w	= berat partikel
$v_i(t)$	= kecepatan partikel saat ini
r_1, r_2	= angka acak antara 0-1
c_1, c_2	= konstanta yang telah ditentukan sebelumnya (umumnya 2)
$X_{pbest}(t)$	= posisi <i>personal best</i> (lokasi terbaik dari suatu partikel individu)
$X_i(t)$	= posisi partikel saat ini
$X_{gbest}(t)$	= posisi <i>global best</i> (lokasi terbaik dari seluruh partikel)
$X_i(t + 1)$	= posisi partikel

2.5.2 Symbiotic Organisms Search

Algoritma SOS diusulkan oleh Cheng dan Prayogo pada 2014 (Cheng & Prayogo, 2014). Algoritma ini mensimulasikan interaksi simbiotik dalam hubungan organisme berpasangan yang digunakan untuk mencari organisme yang paling sesuai. Terdapat 3 fase yang digunakan, seperti dalam interaksi biologi antara dua organisme dalam ekosistem yaitu fase mutualisme, fase komensalisme, fase parasitisme.

Fase mutualisme adalah fase di mana hubungan antara kedua makhluk hidup yang saling menguntungkan. Contohnya hubungan antara lebah dan bunga. Lebah terbang mendekati bunga dan hal ini juga memberikan manfaat pada bunga karena membantu proses penyerbukan. Dalam SOS, penerapannya adalah iterasi selanjutnya dapat menggantikan iterasi sebelumnya apabila hasil yang didapatkan lebih baik sehingga saling menguntungkan.

Formula yang digunakan adalah:

$$X_{i\ new} = X_i + rand(0,1) * (X_{best} - Mutual_Vector * BF_1) \quad (5)$$

$$X_{j\ new} = X_j + rand(0,1) * (X_{best} - Mutual_Vector * BF_2) \quad (6)$$

$$Mutual_Vector = \frac{x_i + x_j}{2} \quad (7)$$

Dimana:

X_i = organisme yang sesuai dengan anggota ke-i dalam ekosistem

- X_j = organisme yang dipilih secara acak dari ekosistem
 $X_{i\ new}$ = kandidat baru dari X_i
 $X_{j\ new}$ = kandidat baru X_j
 BF_1 = angka acak antara 1 atau 2
 BF_2 = angka acak antara 1 atau 2
 X_{best} = solusi global atau tingkat adaptasi terbaik

Fase komensalisme adalah hubungan antara organisme di mana satunya memperoleh keuntungan, sementara yang lain tidak memperoleh keuntungan maupun kerugian. Contohnya, hubungan antara ikan remora dengan ikan hiu. Untuk formula yang digunakan sebagai berikut:

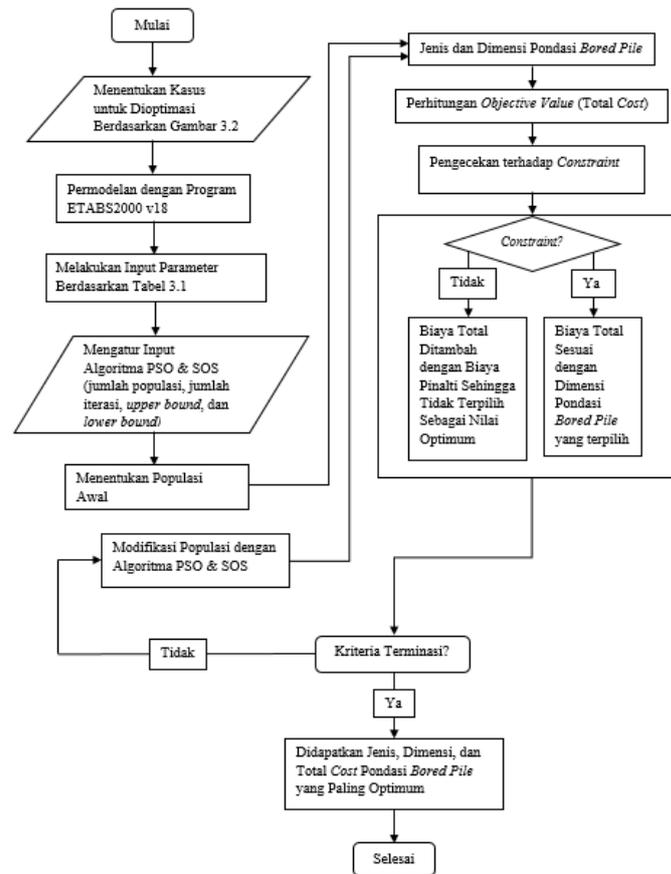
$$X_{i\ new} = X_i + rand(-1,1)(X_{best} - X_j) \quad (8)$$

Formula $(X_{best} - X_j)$ dimaksudkan untuk sejauh mana X_j dapat menambah keuntungan X_i dalam ekosistem untuk memperoleh hasil yang lebih baik dalam suatu organisme.

Fase parasitisme adalah hubungan antara organisme di mana satunya memperoleh keuntungan, sementara yang lain mengalami kerugian. Contohnya, hubungan antara nyamuk dengan manusia. Dalam SOS, digunakan perumpamaan serupa yaitu nyamuk anopheles sebagai parasit yang disebut *Parasite_Vector*. *Parasite_Vector* ini akan berusaha menggantikan X_j dalam ekosistem. Namun apabila X_j lebih dapat menghasilkan hasil yang lebih baik, *Parasite_Vector* akan tidak berlaku dalam ekosistem. Namun berlaku sebaliknya, apabila *Parasite_Vector* dapat menghasilkan hasil yang lebih baik, maka X_j akan tergantikan oleh *Parasite_Vector*.

3. METODOLOGI PENELITIAN

Optimasi dimensi pondasi *bored pile* dilakukan dengan menggunakan metode metaheuristik. Dalam optimasi ini, beban yang digunakan untuk pondasi berasal dari reaksi perletakkan terhadap studi kasus pada bangunan bertingkat. Pemodelan untuk struktur bangunan menggunakan ETABS2000 v18. Selain beban bangunan, juga terdapat parameter lain yang digunakan di dalam optimasi seperti karakteristik tanah. Lokasi tanah didasarkan pada data tanah yang dimiliki oleh peneliti. Parameter-parameter tersebut akan menjadi *input* dalam *coding* dengan MATLAB R2017b. Proses optimasi menggunakan algoritma PSO dan SOS, di mana masing-masing algoritma akan bekerja untuk pondasi *bored pile*. *Constraint* yang digunakan di dalam optimasi ini akan mengacu pada SNI 8460:2017, SNI 2847:2013, dan *Bearing Capacity Theory*. Jika *output* yang dihasilkan melanggar *constraint* maka algoritma akan memberikan *penalty function* berupa penambahan biaya konstruksi sehingga solusi yang melanggar *constraint* tidak terpilih sebagai solusi yang terbaik. Proses iterasi akan berhenti hingga kriteria terminasi tercapai dengan kata lain tercapai iterasi maksimum. Kemudian hasil dari PSO dan SOS juga akan dibandingkan untuk menentukan algoritma yang dapat menghasilkan biaya konstruksi yang paling minimum sebagai *final output*. Untuk lebih jelas, diagram alir proses optimasi dapat dilihat pada Gambar 2.



Gambar 2. Diagram Alir Optimasi Struktur Pondasi *Bored Pile*

4. HASIL DAN DISKUSI

4.1 Studi Kasus Penelitian

Pada penelitian ini, tipe pondasi yang digunakan adalah *bored pile* yang akan dioptimasi untuk masing-masing studi kasus berupa bangunan empat lantai dan bangunan sembilan lantai. Pembebanan yang digunakan berupa beban gravitasi dan beban gempa berdasarkan peta gempa di kota Surabaya. Kemudian beban tersebut dimasukkan pada ETABS2000 v18. Proses optimasi ini dijalankan sebanyak 30 kali percobaan untuk pondasi *bored pile*. Hal ini untuk mengetahui persebaran hasil yang diperoleh dari masing-masing algoritma. Dalam satu kali percobaan algoritma SOS terdiri dari 500 iterasi, sedangkan untuk satu kali percobaan algoritma PSO terdiri dari 2000 iterasi untuk pondasi *bored pile* pada bangunan empat lantai dan sembilan lantai. Terjadi perbedaan antara algoritma PSO dan SOS pada jumlah iterasi karena dalam satu iterasi algoritma SOS melakukan perhitungan *objective function* sebanyak empat kali dimana terdapat beberapa fase pada optimasinya. Hal ini yang menyebabkan satu iterasi SOS setara dengan empat iterasi PSO. Sementara populasi dalam masing-masing algoritma sebanyak 200 populasi. Setelah dilakukan 30 kali percobaan, dilakukan analisis terhadap biaya keseluruhan pondasi *bored pile* terbaik, terburuk, median, rata-rata, dan standar deviasi dari setiap algoritma yang digunakan. Untuk parameter *input* pada program optimasi dapat dilihat pada Tabel 2.

Tabel 2. Parameter *Input* pada Studi Kasus Pondasi *Bored Pile*

Jenis Parameter	Input Parameter	Input Value	Input Value
		Bored Pile 4 lantai	Bored Pile 9 lantai
Parameter Tanah	Rata-rata nilai SPT dibawah alas (MPa)	32,18	29,85
	Rata-rata nilai SPT sepanjang <i>pile</i> yang terhambat (Mpa)	18,44	23,85
	Berat jenis tanah (kN/m ³)	18	18
	Penambahan galian tanah (m)	0,5	0,5
	Kedalaman tanah (m)	34	54
Parameter Beton	<i>Required safety factor bored pile</i>	2,5	2,5
	Kekuatan tekan beton (MPa)	25	30
	Berat jenis beton (kN/m ³)	24	24
	Selimut beton <i>pile cap</i> (mm)	75	75
	Selimut beton <i>pile</i> (mm)	75	75
Parameter Baja Tulangan	Diameter tulangan utama <i>pile cap</i> (mm)	19	25
	Diameter tulangan utama <i>pile</i> (mm)	16	16
	Diameter tulangan sengkang <i>pile</i> (mm)	16	16
	Kekuatan leleh baja tulangan utama <i>pile cap</i> (MPa)	400	400
	Kekuatan leleh baja tulangan utama <i>pile</i> (MPa)	400	400
Parameter Struktur Bangunan	Kekuatan leleh baja tulangan sengkang <i>pile</i> (MPa)	400	400
	Panjang kolom (mm)	600	1000
	Lebar kolom (mm)	600	1000
	Konstanta letak kolom	40	40
	Sudut beban terhadap horizontal (°)	90	90

Parameter algoritma PSO yang digunakan pada studi kasus ini berbeda dengan parameter optimasi PSO pada umumnya. Perbedaan ini digunakan untuk memaksimalkan algoritma PSO dalam studi kasus ini. Parameter algoritma PSO terdiri dari w , c_1 , dan c_2 . Untuk parameter algoritma PSO yang digunakan merupakan hasil dari *trial and error* pada kasus bangunan empat lantai dan sembilan lantai. Percobaan *trial and error* dilakukan dengan menggunakan beberapa angka pada parameter w , c_1 , dan c_2 sampai ditemukan hasil yang optimal dimana memiliki hasil minimum. Nilai parameter algoritma yang dipakai pada penelitian ini $w = 0,05$, $c_1 = 1,8$ dan $c_2 = 1,8$. Sedangkan nilai parameter algoritma umum $w = 0,4-0,9$, $c_1 = 2$ dan $c_2 = 2$.

Berdasarkan hasil dari pemodelan bangunan menggunakan ETABS2000 v18, maka didapatkan reaksi perletakan yang merupakan *input* parameter pada studi kasus. Untuk parameter *input* beban pada pondasi *bored pile* merupakan reaksi perletakan yang dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. *Input* Parameter Beban Pondasi *Bored Pile* untuk Studi Kasus Bangunan Empat Lantai

Tipe Pondasi Ke-	Jenis Beban				
	Fz (kN)	Fx (kN)	Fy (kN)	Mx (kNm)	My (kNm)
1	1020,9174	176,8973	181,0252	-269,3478	470,9962
2	1372,9436	204,7348	199,4423	-313,2969	525,689
3	626,5834	201,2773	-179,3113	-514,2955	270,1376

Tabel 3. *Input* Parameter Beban Pondasi *Bored Pile* untuk Studi Kasus Bangunan Empat Lantai (Lanjutan)

Titik Pondasi Ke-	Jenis Beban				
	Fz (kN)	Fx (kN)	Fy (kN)	Mx (kNm)	My (kNm)
4	625,2847	-175,6892	178,4272	511,6194	-264,7512
5	1372,8838	-204,7368	-198,3635	272,3669	-525,6913
6	1036,6702	-202,4833	-180,1412	231,6116	-546,7637

Tabel 4. *Input* Parameter Beban Pondasi *Bored Pile* untuk Studi Kasus Bangunan Sembilan Lantai

Titik Pondasi Ke-	Jenis Beban				
	Fz (kN)	Fx (kN)	Fy (kN)	Mx (kNm)	My (kNm)
1	3665,0755	521,2752	521,3198	-2685,4234	1639,8774
2	4051,6427	552,6013	-566,3931	-1687,2688	2781,6275
3	4076,6651	566,5699	566,39	-1687,4806	2851,0077
4	1478,2874	559,2466	-519,1735	1621,5094	-2943,2475
5	4064,448	-564,43	552,3438	-2773,5613	1699,5026
6	4429,7286	-590,8619	-594,3611	2858,3486	-1635,2616
7	4417,5504	-606,0884	594,3426	1624,0982	-2933,1573
8	4059,2609	-609,5984	-552,5055	2773,7318	-1485,4338
9	4086,0954	564,4229	566,3512	-2843,7369	1699,8143
10	4428,9198	590,845	-609,8633	2930,1019	-1634,4938
11	4429,889	606,1056	609,8715	-2930,1108	1563,5742
12	4091,0203	609,6054	-566,1945	2843,5713	-1485,1278
13	2882,3859	-518,8197	560,1404	-2520,5286	218,9664
14	4051,0508	-552,9413	-612,1181	1475,8482	-2781,9864
15	4077,2675	-566,2303	612,1315	1475,6366	-2850,6491
16	3666,366	-561,702	-562,2816	2894,8305	-1430,5442

Keterangan :

- Fz = beban vertikal kolom
- Fx = beban horizontal arah X kolom
- Fy = beban horizontal arah Y kolom
- Mx = momen tegak lurus arah X kolom
- My = momen tegak lurus arah Y kolom

Objective function merupakan suatu rumusan fungsi yang digunakan untuk menggambarkan suatu nilai yang merupakan tujuan dalam suatu proses optimasi. *Objective function* yang digunakan dalam penelitian ini dapat dilihat dalam Persamaan 1. Dalam penelitian ini, perhitungan *objective function* tidak hanya pada biaya material, melainkan juga biaya tenaga kerja. Analisa biaya yang digunakan dalam penelitian ini dapat dilihat pada Tabel 5.

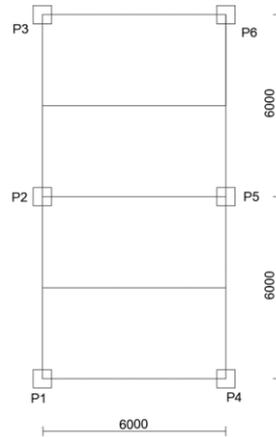
Tabel 5. Biaya Material dan Tenaga Kerja

Jenis Pekerjaan	Uraian	Jumlah Harga
Galian tanah	Tenaga kerja	Rp100.125,00
	Bahan	-
Pemasangan bekisting pondasi	Tenaga kerja	Rp108.550,00
	Bahan	Rp65.500,00
Pembesian	Tenaga kerja	Rp18.050,00
	Bahan	Rp92.250,00
Pengecoran dengan <i>ready mix fc'</i> 25	Tenaga kerja	Rp165.000,00
	Bahan	Rp877.200,00
Pengecoran dengan <i>ready mix fc'</i> 30	Tenaga kerja	Rp165.000,00
	Bahan	Rp943.500,00
Pengurugan dan pemadatan tanah kembali	Tenaga kerja	Rp61.250,00
	Bahan	-
Pengeboran dengan alat bor untuk diameter 30 cm	Tenaga kerja	Rp136.944,44
	Bahan	-
Pengeboran dengan alat bor untuk diameter 40 cm	Tenaga kerja	Rp170.000,00
	Bahan	-
Pengeboran dengan alat bor untuk diameter 50 cm	Tenaga kerja	Rp207.777,78
	Bahan	-
Pengeboran dengan alat bor untuk diameter 60 cm	Tenaga kerja	Rp245.555,56
	Bahan	-
Pengeboran dengan alat bor untuk diameter 70 cm	Tenaga kerja	Rp294.666,67
	Bahan	-
Pengeboran dengan alat bor untuk diameter 80 cm	Tenaga kerja	Rp353.600,00
	Bahan	-
Pembuangan tanah bor dengan jarak angkut 10 km	Tenaga kerja	Rp108.494,03
	Bahan	-

4.2 Hasil Penelitian

4.2.1 Studi Kasus Bangunan Empat Lantai

Pada Gambar 2 dapat terlihat P1, P2, P3, P4, P5, dan P6 yang merupakan pondasi dari bangunan empat lantai. Untuk masing-masing variabel P terdiri dari variabel N (jumlah *pile*), D (diameter *pile*), dan T (tebal *pile cap*) yang merupakan variabel yang akan dioptimalkan. Untuk variabel D dari ke-enam pondasi disamakan. Sebagai contoh, P1 terdiri dari N1, D, dan T1. Lalu, untuk P2 terdiri dari N2, D, dan T2. Hal ini berlaku juga untuk P3, P4, P5, dan P6.



Gambar 2. Denah Pondasi *Bored Pile* untuk Studi Kasus Bangunan Empat Lantai

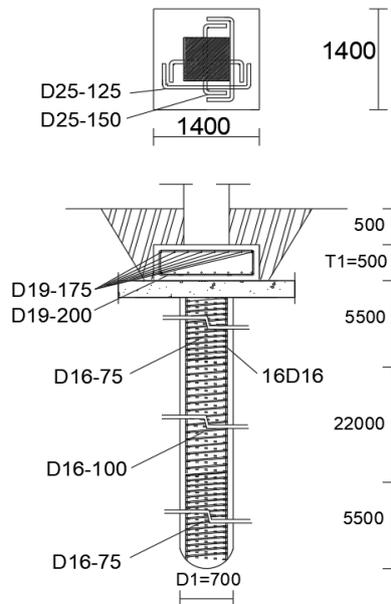
Tabel 6 menunjukkan perbandingan hasil optimasi pondasi *bored pile* untuk bangunan empat lantai dengan algoritma PSO dan SOS untuk menunjukkan hasil percobaan yang paling optimal. Untuk penamaan dimensi pondasi setempat mengacu pada Gambar 1.

Tabel 6. Hasil Percobaan Optimasi Pondasi *Bored Pile* untuk Studi Kasus Bangunan Empat Lantai

Variabel	SOS	PSO
D (mm)	700	600
N1 (buah)	1	1
T1 (mm)	500	500
N2 (buah)	1	2
T2 (mm)	500	800
N3 (buah)	1	1
T3 (mm)	500	500
N4 (buah)	1	1
T4 (mm)	500	500
N5 (buah)	1	2
T5 (mm)	500	800
N6 (buah)	1	1
T6 (mm)	500	500
Terbaik (Rp)	420.005.813,08	420.005.813,08
Terburuk (Rp)	420.005.813,08	436.839.411,92
Median (Rp)	420.005.813,08	436.839.411,92
Rata-rata (Rp)	420.005.813,08	436.278.291,96
Standar Deviasi (Rp)	0,00	3.073.380,60
Success Rate	30/30	30/30

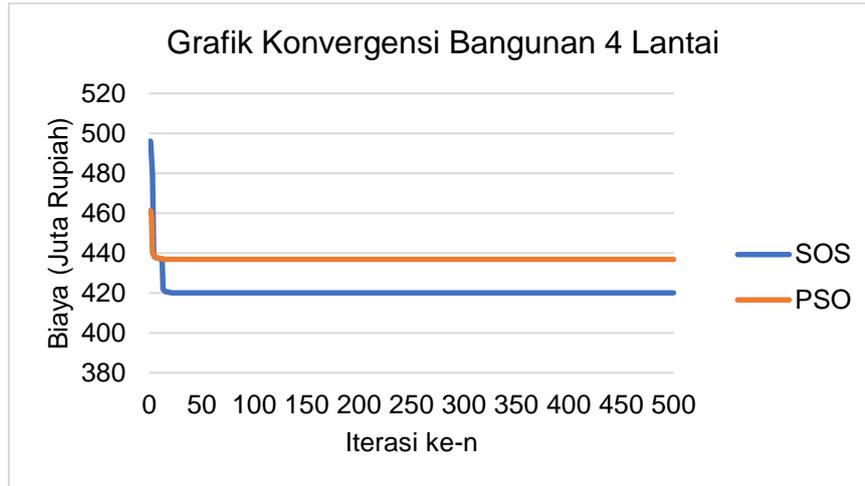
Berdasarkan pada Tabel 6, ditunjukkan perbandingan proses olah data dari kedua algoritma yaitu PSO dan SOS. Pengambilan data pada tabel tersebut hanya menggunakan data yang berhasil tidak melanggar *constraint*. Dapat dilihat bahwa terdapat perbedaan variabel D dari kedua algoritma. SOS memiliki hasil 700 mm sedangkan PSO memiliki hasil 600 mm yang menyebabkan perbedaan pada variabel N dan T antara algoritma SOS dan PSO. Dapat dilihat juga bahwa *success rate* algoritma SOS dan PSO memiliki hasil yang sama baiknya, yaitu 30/30 yang berarti semua proses optimasi berhasil mengeluarkan dimensi yang tidak melanggar *constraint*. Dari rata-rata 30 kali percobaan, diperoleh bahwa SOS memiliki nilai

rata-rata yang lebih baik sebesar Rp 420.005.813,08 dengan standar deviasi sebesar 0 sementara PSO sebesar Rp 436.278.291,96 dengan standar deviasi sebesar Rp 3.073.380,60. Selain itu, hasil terbaik yang dihasilkan oleh algoritma SOS dan PSO sebesar Rp 420.005.813,08. Sedangkan hasil terburuk yang dihasilkan oleh algoritma SOS sebesar Rp 420.005.813,08 sementara PSO adalah sebesar Rp 436.839.411,92. Walaupun hasil terbaik yang dihasilkan sama, akan tetapi jika melihat hasil rata-rata dan standar deviasi akan terlihat bahwa algoritma SOS memiliki hasil yang lebih konsisten. Untuk gambar struktur pondasi dengan hasil paling optimal dapat dilihat pada Gambar 3.



Gambar 3. Struktur Pondasi *Bored Pile* Studi Kasus Bangunan Empat Lantai

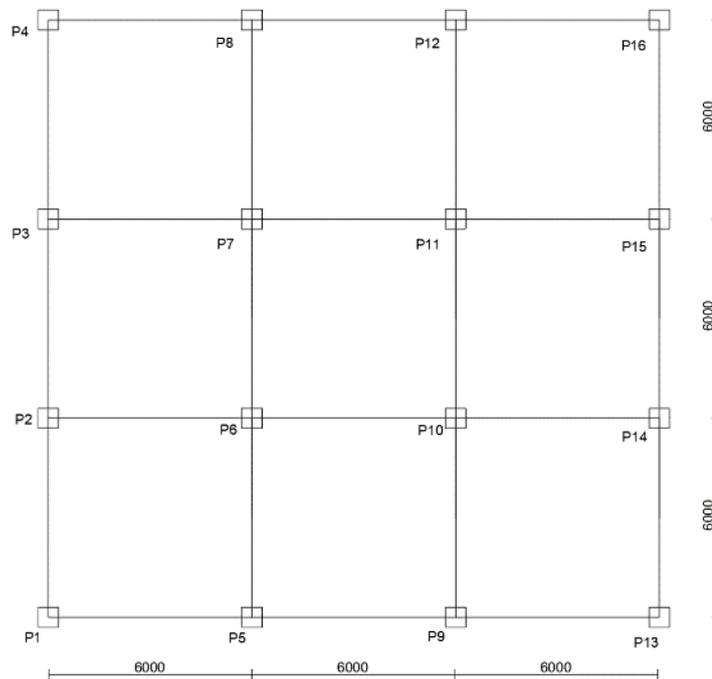
Gambar 4 menunjukkan grafik konvergensi dari hasil percobaan median pada optimasi pondasi *bored pile* bangunan empat lantai dengan algoritma SOS dan PSO. Grafik konvergensi menggunakan percobaan median karena median berada di antara performa paling baik dan paling buruk, sehingga representatif dalam menunjukkan performa kedua algoritma. Pada awal iterasi, algoritma PSO mampu menemukan biaya konstruksi yang lebih rendah dari algoritma SOS, yaitu sebesar Rp 461.109.370,49 sementara SOS menghasilkan biaya konstruksi yang lebih tinggi sebesar Rp 496.137.585,55. Sebelum iterasi ke-50, algoritma PSO dan SOS sudah mencapai hasil yang optimal, yaitu algoritma PSO sebesar Rp 436.839.411,92, sedangkan algoritma SOS sebesar Rp 420.046.061,71. Dapat terlihat pada Gambar 4.4 algoritma PSO mencapai hasil optimal sebelum algoritma SOS. Akan tetapi, dari grafik konvergensi dapat dilihat bahwa algoritma SOS mampu memberikan hasil yang paling optimal pada akhir iterasi seperti yang dapat dilihat pada Gambar 4.4 sebelumnya, sedangkan PSO belum mampu memberikan biaya optimal pada akhir iterasinya. Dikarenakan algoritma PSO terlihat terjebak dalam suatu kondisi dimana populasi yang digunakan dalam optimasi mengalami konvergen yang terlalu cepat sehingga mendapatkan biaya yang kurang optimal yang menyebabkan terjebak dalam biaya yang *suboptimal* meskipun iterasi ditambahkan biaya yang optimal tidak bisa ditemukan. Sehingga, performa algoritma SOS dapat dikatakan lebih baik dalam menemukan dimensi dan biaya konstruksi pondasi *bored pile* untuk studi kasus bangunan empat lantai.



Gambar 4. Konvergensi Pondasi *Bored Pile* untuk Studi Kasus Bangunan Empat Lantai

4.2.2 Studi Kasus Bangunan Sembilan Lantai

Pada Gambar 5 dapat terlihat P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, dan P16 yang merupakan pondasi dari bangunan sembilan lantai. Untuk masing-masing variabel P terdiri dari variabel N (jumlah *pile*), D (diameter *pile*), dan T (tebal *pile cap*) yang merupakan variabel yang akan dioptimalkan. Untuk variabel D dari ke-enambelas pondasi disamakan. Sebagai contoh, P1 terdiri dari N1, D, dan T1. Lalu, untuk P2 terdiri dari N2, D, dan T2. Hal ini berlaku juga untuk P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, dan P16.



Gambar 5. Denah Pondasi *Bored Pile* untuk Studi Kasus Bangunan Sembilan Lantai

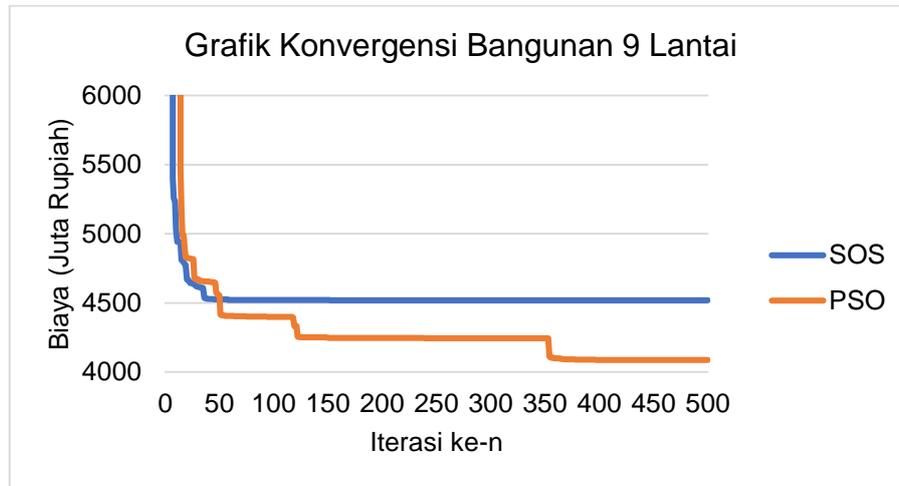
Tabel 7 menunjukkan perbandingan hasil optimasi pondasi *bored pile* untuk bangunan sembilan lantai dengan algoritma PSO dan SOS untuk menunjukkan hasil percobaan yang paling optimal. Untuk penamaan dimensi pondasi *bored pile* mengacu pada Gambar 1.

Tabel 7. Hasil Percobaan Optimasi Pondasi *Bored Pile* untuk Studi Kasus Bangunan Sembilan Lantai

Variabel	SOS	PSO
D (mm)	800	700
N1 (buah)	2	4
T1 (mm)	1000	1000
N2 (buah)	4	4
T2 (mm)	1000	1000
N3 (buah)	4	4
T3 (mm)	1000	950
N4 (buah)	2	2
T4 (mm)	950	1000
N5 (buah)	2	4
T5 (mm)	950	1000
N6 (buah)	2	4
T6 (mm)	1000	1000
N7 (buah)	3	3
T7 (mm)	750	900
N8 (buah)	2	4
T8 (mm)	950	1000
N9 (buah)	2	4
T9 (mm)	950	1000
N10 (buah)	2	4
T10 (mm)	1000	1000
N11 (buah)	2	4
T11 (mm)	1000	900
N12 (buah)	2	4
T12 (mm)	950	1000
N13 (buah)	2	2
T13 (mm)	950	750
N14 (buah)	3	4
T14 (mm)	700	1000
N15 (buah)	3	4
T15 (mm)	700	950
N16 (buah)	2	4
T16 (mm)	1000	950
Terbaik (Rp)	4.084.400.461,24	4.084.787.459,88
Terburuk (Rp)	4.519.134.831,44	4.524.435.307,70
Median (Rp)	4.519.134.831,44	4.088.455.331,25
Rata-rata (Rp)	4.330.828.161,62	4.199.994.131,84
Standar Deviasi (Rp)	219.018.218,92	171.858.789,38
<i>Success Rate</i>	30/30	30/30

Berdasarkan pada Tabel 7, ditunjukkan perbandingan proses olah data dari kedua algoritma yaitu PSO dan SOS. Pengambilan data pada tabel tersebut hanya menggunakan data yang berhasil tidak melanggar *constraint*. Dapat dilihat bahwa terdapat perbedaan variabel D dari kedua algoritma. SOS memiliki hasil 800 mm sedangkan PSO memiliki hasil 700 mm yang menyebabkan perbedaan pada variabel N dan T antara algoritma SOS dan PSO. Dapat dilihat juga bahwa dalam optimasi pondasi *bored pile* untuk studi kasus bangunan sembilan lantai, algoritma PSO memiliki performa yang lebih baik daripada algoritma SOS. *Success rate* algoritma SOS dan PSO memiliki hasil yang sama baiknya, yaitu 30/30 yang berarti semua

menemukan dimensi dan biaya konstruksi pondasi *bored pile* untuk studi kasus bangunan sembilan lantai.



Gambar 7. Konvergensi Pondasi *Bored Pile* untuk Studi Kasus Bangunan Sembilan Lantai

5. KESIMPULAN

Berdasarkan hasil penelitian dan diskusi yang telah dilakukan, dapat diambil kesimpulan pada penelitian ini, peneliti sudah membandingkan performa optimasi dari dua buah algoritma metaheuristik, yaitu PSO dan SOS dengan meninjau studi kasus bangunan empat dan sembilan lantai menggunakan batasan-batasan menurut SNI 8460:2017, SNI 2847:2013, dan *bearing capacity theory*. Pada bangunan empat lantai dapat dilihat bahwa algoritma SOS memiliki performa yang lebih baik dibandingkan dengan algoritma PSO. Hal ini disebabkan karena pada kasus ini algoritma SOS menghasilkan biaya konstruksi yang lebih rendah dari pada menggunakan algoritma PSO. Selain itu algoritma SOS memiliki nilai yang lebih konsisten jika dibandingkan dengan PSO karena jumlah dari rata-rata SOS dan standar deviasi SOS lebih rendah apabila dibandingkan jumlah rata-rata PSO dan standar deviasi PSO. Algoritma SOS dan PSO berhasil menghasilkan biaya konstruksi yang sama besarnya sehingga dapat dikatakan kedua algoritma optimasi memiliki performa yang sama baik. Sedangkan pada bangunan sembilan lantai, SOS memiliki performa yang lebih baik dibandingkan dengan PSO. Hal ini disebabkan karena pada kasus ini algoritma SOS menghasilkan biaya konstruksi yang lebih rendah dari pada menggunakan algoritma PSO. Tetapi algoritma PSO memiliki hasil yang lebih konsisten karena algoritma PSO memiliki nilai yang lebih konsisten jika dibandingkan dengan SOS karena jumlah dari rata-rata PSO dan standar deviasi PSO lebih rendah apabila dibandingkan jumlah rata-rata SOS dan standar deviasi SOS.

6. DAFTAR REFERENSI

- Cheng, M. Y., & Prayogo, D. (2014). Symbiotic Organisms Search: a New Metaheuristic Optimization Algorithm. *Computers & Structures*, 139, 98-112.
- Das, B. M. (2010). *Geotechnical Engineering Handbook*. USA: J. Ross Publishing.
- Das, B. M. (2015). *Principles of Foundation Engineering*. USA: Cengage learning.

- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of ICNN'95-International Conference on Neural Networks 4*, 1942-1948
- SNI, 2847:2013. (2013). *Persyaratan Beton Struktural untuk Bangunan Gedung*. Jakarta: Badan Standardisasi Nasional.
- SNI, 8460:2017. (2017). *Persyaratan Perancangan Geoteknik*. Jakarta: Badan Standardisasi Nasional.
- Toprak, B., Sevim, O., Totic, E., & Kalkan, I. (2018). *The Functions of Pile Types and Piles Used In Construction*. September, 34–36.