Available online at: http://e-jurnal.lppmunsera.org/index.php/JSMI

Jurnal Sistem dan Manajemen Industri

ISSN (Print) 2580-2887 ISSN (Online) 2580-2895

Algoritma *Simulated Annealing* untuk Optimasi Rute Kendaraan dan Pemindahan Lokasi Sepeda pada Sistem *Public Bike Sharing*

A.A.N. Perwira Redi[1*], Anak Agung Ngurah Agung Redioka[2]

[1]Department of Logistics Engineering, Universitas Pertamina, Jl. Teuku Nyak Arief, RT.7/RW.8, Simprug, Kec. Kby. Lama, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12220, Indonesia

[2]Jurusan Sistem Informasi Akuntansi, STMIK Primakara, Bali, Jl. Tukad Badung No.135, Renon, Kec. Denpasar Sel., Kota Denpasar, Bali 80226, Indonesia

ARTICLE INFORMATION

Article history:

Received: June 17, 2019
Revised: July 24, 2019
Accepted: July 30, 2019

Kata Kunci

Bike-Sharing
Simulated Annealing
Static Repositioning Problem

Keywords:

Bike-Sharing
Simulated Annealing
Static Repositioning Problem

***Corresponding Author**

A.A.N. Perwira Redi
E-mail: wira.redi@gmail.com

A B S T R A K

Sistem *public bike sharing* memiliki permasalahan dimana jumlah sepeda pada *docking station* (tempat parkir sepeda) perlu diseimbangkan untuk menjamin kepuasan pengguna sistem. Untuk itu solusi yang sering dilakukan adalah distribusi sepeda agar pengguna sistem tetap dapat parkir untuk lokasi yang biasanya penuh dengan sepeda ataupun mengambil sepeda pada lokasi yang biasanya kekurangan sepeda. Tujuan dari pemecahan masalah ini adalah untuk mendapatkan rute kendaraan dengan total biaya operasi dari kendaraan tersebut. Total biaya operasi kendaraan diasosiasikan dengan total waktu yang dibutuhkan kendaraan untuk melakukan distribusi sepeda. Selain itu terdapat juga biaya penalti yang berasosiasi dengan kekurangan jumlah sepeda ataupun *slot* parkir pada waktu operasi dari fasilitas *public bike-sharing*. Dalam penelitian ini, dua variasi algoritma *simulated annealing (SA)* dikembangkan untuk menyelesaikan permasalahan SBRP disebut dengan SA_BF dan SA_CF. Data yang digunakan berasal dari studi kasus milik *Velib bike sharing system* di Paris, Prancis. Hasil dari eksperimen menunjukkan bahwa kedua algoritma SA_BF dan SA_CF berhasil dengan baik menghasilkan solusi untuk SBRP. Algoritma ini memiliki selisih rata-rata 2.21% dan 0.36% terhadap algoritma Arc-Indexed (AI) dari penelitian sebelumnya pada *dataset* pertama. Sedangkan untuk rata-*dataset* kedua, algoritma Tabu Search, SA_BF dan SA_CF memperoleh selisih rata-rata 0.65%, 1.08% dan 0.38% terhadap hasil optimal.

A B S T R A C T

The public bike-sharing system has a problem where the number of bicycles at the docking station needs to be balanced to ensure system user satisfaction. The usual solution is to distribute bicycles so that system users can still park for locations that are usually full of bicycles or pick up bicycles at locations that normally lack bicycles. The purpose of solving this problem is to get a vehicle route with the total operating costs of the vehicle. The full vehicle operating costs are associated with the full time taken by the vehicle to distribute the bicycle. Besides, there are also penalty fees related to the lack of bikes or parking slots at the time of operation of the public bike-sharing facility. In this study, two variations of the simulated annealing (SA) algorithm were developed to solve the SBRP problem called SA_BF and SA_CF. The data used comes from a Velib bike-sharing system case study in Paris, France. The results of the experiment show that both the SA_BF and SA_CF algorithms succeeded in solving SBRP. This algorithm has an average difference of 2.21% and 0.36% of the Arc-Indexed algorithm (AI) from previous studies in the first dataset. As for the second dataset, Tabu Search algorithm, SA_BF and SA_CF obtained an average difference of 0.65%, 1.08% and 0.38% of the optimal results.

PENDAHULUAN

Public bike-sharing system merupakan sistem yang menyediakan penyewaan sepeda kepada publik, dimana masyarakat yang ingin menggunakan sepeda dapat melakukan transaksi pada pos parkir sepeda yang biasanya tersebar di area publik di dalam kota. Secara umum *public bike-sharing* digunakan di daerah perkotaan untuk transportasi dengan jarak pendek, seperti misalnya penyambung moda transportasi dari stasiun kereta menuju moda transportasi lainnya ataupun menuju lokasi lainnya dengan jarak dekat [1], [2]. Salah satu faktor penting yang mendukung kesuksesan sistem *public bike sharing* adalah kemampuan untuk memenuhi kebutuhan sepeda dan menyediakan parkir sepeda di lokasi tujuan [3], [4]

Tempat peminjaman sepeda biasanya berupa parkir dengan *electronic docking / station* (tempat parkir khusus sepeda) dengan beberapa sepeda terparkir dan terkunci pada *docking*. *Docking* ini digunakan untuk memantau posisi dan jumlah sepeda yang ada pada sistem. Saat ada pengguna yang menyewa sepeda, sistem pusat akan mengirimkan sinyal untuk membuka kunci sepeda pada *docking*. Pengguna dapat mengembalikan sepeda dengan memarkir pada *docking* yang kosong. Semua transaksi peminjaman dan pengembalian tercatat secara *real-time* pada sistem. Sistem menginformasikan kepada pengguna posisi sepeda dan jumlah parkir yang kosong melalui website atau media lainnya [5].

Dalam sistem *public bike sharing*, permasalahan timbul karena adanya fluktuasi permintaan pada waktu-waktu tertentu. Sistem *public bike sharing* harus mampu menyediakan parkir kosong bagi pengguna [6]. Salah satu keluhan utama dari pengguna tentang *public bike sharing* adalah ketidakmampuan sistem menyediakan parkir di beberapa tempat pada waktu tertentu [7]. Hal ini sering kali terjadi karena pada waktu-waktu tertentu suatu lokasi menjadi tujuan utama dari pengguna dan lokasi lain menjadi lokasi asal sepeda sehingga terjadi kekurangan sepeda pada lokasi tertentu, di lain pihak terjadi penumpukan sepeda yang mengakibatkan tidak adanya tempat parkir pada *electronic dock*. Untuk mengatasi permasalahan ini, biasanya operator sistem *public bike sharing* menggunakan mekanisme pendistribusian untuk mereposisi beberapa sepeda dari parkir yang penuh ke tempat parkir yang kosong secara berkala. Aktifitas ini sering disebut *bike repositioning*. Tujuan utama dari operator melakukan reposisi ini adalah meminimalisir *shortage*,*overcapacity,* dan biaya untuk melakukan pendistribusian ulang sepeda. Dalam studi literatur, problem optimasi yang mempertimbangkan permasalahan pendistribusian ulang sepeda pada *public bike sharing* dikenal dalam dua model yaitu disebut dengan the *static bicycle repositioning problem* (SBRP) [8], [9] dan *the dynamic bicycle repositioning problem* (DBRP) [10], [11].

Penelitian ini berfokus pada SBRP. Istilah *static repositioning* merujuk kepada pengaturan posisi sepeda pada sistem *public bike sharing* untuk periode operasi selanjutnya. Secara umum, SBRP merupakan kondisi sistem pada malam hari dimana peramalan permintaan sepeda dari satu tempat ke tempat lain sudah dilakukan. Dalam permasalahan ini, diasumsikan terdapat satu depo sepeda yang mengontrol keseluruhan posisi sepeda pada sistem. Setiap *docking* atau station memiliki jumlah persediaan awal / jumlah sepeda yang terdapat pada akhir periode operasi, kapasitas *docking,* dan fungsi penalti yang didefinisikan berdasarkan peramalan permintaan di periode operasi berikutnya. Waktu yang dibutuhkan untuk satu truk yang mengangkut sepeda tersebut dari *station* ke *station* lain juga telah diketahui. Solusi yang dicari adalah mendefinisikan rute untuk setiap truk dan jumlah sepeda yang perlu dipindahkan dari satu station ke station lain. Selain itu, terdapat pula batasan waktu dalam satu periode operasi pendistribusian sepeda, yakni dari awal periode hingga sebelum periode berikutnya dimulai, contohnya dari malam hari hingga pagi sebelum sistem beroperasi. Model SBRP di penelitian ini mengacu pada penelitian Raviv et al. [4].

SBRP tergolong permasalahan yang memiliki kompleksitas *NP-hard*. Dimana untuk mendapatkan solusi optimal dari permasalahan dengan kategori ini, dibutuhkan metode-metode khusus yang biasanya tergolong metode metaheuristik. Metode metaheuristik adalah metode yang menggunakan pendekatan terhadap solusi optimal [12], [13], dan [14]. Berbeda dengan metode pendekatan eksak, metode metaheuristik tidak menjamin diperolehnya solusi optimal. Meskipun demikian, penyelesaian problem yang kompleks dengan metode metaheuristik banyak digunakan karena mampu memberikan solusi yang cukup baik, yaitu dekat dengan solusi optimal pada waktu yang relatif cepat. Berdasarkan studi literatur yang dilakukan sebelumnya, metode yang diimplementasikan untuk penyelesaian permasalahan model SBRP dari Raviv et al. [4] memperlihatkan bahwa belum ada penelitian yang menggunakan algoritma *simulated annealing*

untuk penyelesaian model tersebut.

Tujuan dari penelitian ini adalah untuk mengisi gap penelitian, yaitu melakukan eksperimen penggunaan algoritma SA untuk permasalahan SBRP. Adapun kontribusi utama dari penelitian ini adalah pengembangan algoritma *simulated annealing* (SA) untuk pemecahan SBRP. Algoritma *simulated annealing* tergolong kedalam algoritma metaheuristik yang telah terbukti efektif dalam menyelesaikan permasalahan yang tergolong dalam *NP-hard problem*. Algoritma SA yang dikembangkan terdiri dari dua versi yakni: SA_BF dan SA_CF. SA_BF merupakan algoritma SA dengan fungsi yang digunakan untuk menentukan penggunaan solusi yang lebih buruk berdasarkan fungsi exponential yang disebut *Boltzman Function*. Sedangkan, SA_CF menggunakan fungsi kuadratik terhadap temperatur yang disebut *Cauchy Function*.

METODOLOGI PENELITIAN

Data yang digunakan dalam penelitian ini adalah data yang berasal dari studi kasus milik Velib bike-sharing system di Paris, Prancis. Optimasi pemecahan dalam *static bike repositioning* menggunakan metode *simulated annealing*.

Static Bike Repositioning Problem (SBRP)

Model matematika yang digunakan sebagai referensi utama dalam SBRP pada penelitian ini merupakan model yang dikembangkan oleh Raviv et al. [4]. Secara singkat dapat kami jelaskan variabel keputusan, fungsi tujuan, dan batasan dari model SBRP adalah sebagai berikut:

1. Parameter dan Variabel Keputusan

N dan N_0	:	N adalah himpunan dari seluruh *station* dan N_0 adalah N ditambah dengan depot.
$f_i(s_i)$	:	Fungsi penalti dengan parameter diambil dari jumlah sepeda yang ditempatkan pada *station i*
α	:	*Weighting / scaling factor* dari fungsi tujuan dengan fungsi penalti
t_{ij}	:	Waktu yang dibutuhkan untuk menempuh perjalanan dari *station i* ke *station j*
c_i	:	Jumlah *locker* pada *station i* yang disebut juga sebagai kapasitas dari *station*.
s_i^0	:	Jumlah awal sepeda pada *station i* sebelum proses reposisi sepeda
x_{ijv}	:	Variabel biner yang bernilai satu jika kendaraan *v* menempuh perjalanan dari *i* ke *j*
s_i	:	Jumlah sepeda pada *station i* pada akhir proses reposisi sepeda

2. Fungsi tujuan

$$\text{Min z} = \sum_{i \in N} f_i(s_i) + \alpha \sum_{i \in N_0} \sum_{j \in N_0} \sum_{v \in N_0} t_{ij} x_{ijv} \quad (1)$$

Fungsi tujuan pada persamaan (1) mendeskripsikan bahwa dalam SBRP, fungsi tujuan yang dicari adalah minimasi biaya berdasarkan fungsi penalti dan biaya operasional kendaraan untuk mereposisi sepeda.

3. Batasan dan Asumsi
   a. Pada setiap *station* jumlah sepeda yang diletakkan sebagai inventori tidak boleh melebihi kapasitas *locker* yang ada.
   b. Untuk efisiensi distribusi sepeda, satu *station* hanya dikunjungi sekali pada rute kendaraan.
   c. Terdapat batasan total waktu untuk melakukan reposisi sepeda.
   d. Kendaraan untuk mengangkut sepeda mulai berangkat dari depot.

Simulated Annealing (SA)

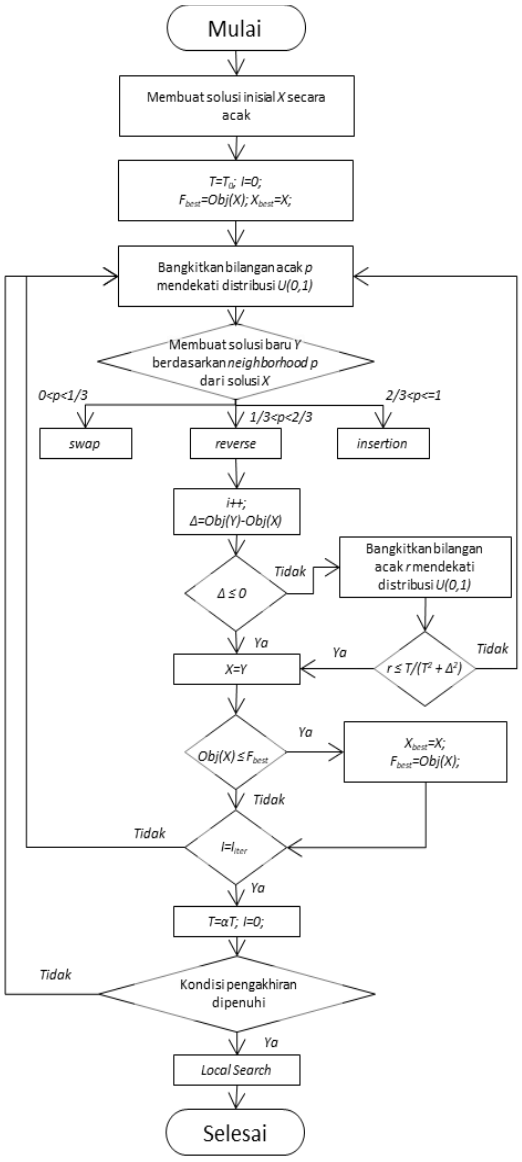
SA merupakan algoritma metaheuristik yang terinspirasi dari proses *annealing* yang dikenal dalam ilmu metalurgi sebagai proses pemanasan material. Pemanasan dilakukan sampai temperatur mencapai titik lebur dari material tersebut lalu pendinginannya dilakukan secara perlahan-lahan, di dalam sebuah tungku. Berdasarkan analogi ini, Metropolis et al. [15] memperkenalkan konsep dari SA,kemudian menjadi elemen penting yang digunakan pada algoritma ini selanjutnya dikenal dengan *metropolis criterion*. Algoritma SA melakukan pencarian dengan memberikan solusi yang lebih buruk dipertimbangkan dalam pencarian dan perlahan-lahan mengurangi peluang solusi yang lebih buruk dari kondisi sekarang diterima dalam proses pencarian. Hal ini merupakan mekanisme yang membuat SA mampu memperoleh solusi yang baik dan keluar dari perangkap lokal optimal [16].

Implementasi SA untuk SBRP

Algoritma SA sudah teruji dalam penyelesaian problem kompleks, seperti permasalahan pembuatan rute kendaraan. Contohnya Lin et al. [17], menyelesaikan permasalahan pembuatan rute dengan pertimbangan penggunaan truk besar dan kecil yang dikenal dengan *truck and trailer problem* menggunakan algoritma SA. Yu et al. [16] menggunakan SA untuk permasalahan *hybrid vehicle routing problem*. Dalam penelitian tersebut, penggunaan *Cauchy Function* pada SA berhasil memperoleh hasil yang baik. Peneliti mengaplikasikan algoritma SA yang dimodifikasi representasi solusinya untuk memecahkan permasalahan SBRP. Kedua algoritma SA pada

 http://dx.doi.org/10.30656/jsmi.v3i1.1473

penelitian ini dimulai dari solusi inisial yang dibuat secara acak. Kemudian pada setiap iterasi, pemecahan masalah dibentuk berdasarkan pembentukan solusi baru yang sering disebut neighborhood move. Metode pembentukan solusi baru ini adalah metode sederhana yang terdiri dari menukar (*swap*), membalik (*reverse*) dan *insert*. Selanjutnya fungsi objektif dari solusi yang baru dibentuk dibandingkan dengan fungsi objektif dari solusi sebelumnya. Jika solusi baru memiliki fungsi objektif lebih baik, maka solusi baru menggantikan solusi lama. Jika tidak, maka kriteria *metropolis* digunakan untuk menentukan apakah suatu solusi baru akan menggantikan solusi lama. Jika solusi baru lebih baik dari solusi yang sudah pernah ditemukan maka solusi ini juga dicatat.



Gambar 1. Diagram alir algoritma SA_CF untuk permasalahan SBRP.

Pada akhir iterasi, strategi intensifikasi dilakukan dengan mencoba berkali-kali metode *neighborhood move* pada solusi terbaik yang sudah pernah ditemukan sepanjang pencarian solusi. Prosedur ini disebut *local search*. Gambar 1 memperlihatkan diagram alir algoritma SA untuk permasalahan SBRP. Adapun perbedaan algoritma SA_BF dan SA_CF terletak pada fungsi keputusan untuk penerimaan solusi yang memiliki nilai objektif lebih besar dari sebelumnya. Algoritma SA_BF menggunakan fungsi *exponential* yang dikenal dengan *Boltzman Function*, $\exp(-\Delta / KT)$. Di sisi lain SA_CF menggunakan *Cauchy Function* yaitu is $T/(T^2 + \Delta^2)$.

Representasi Solusi

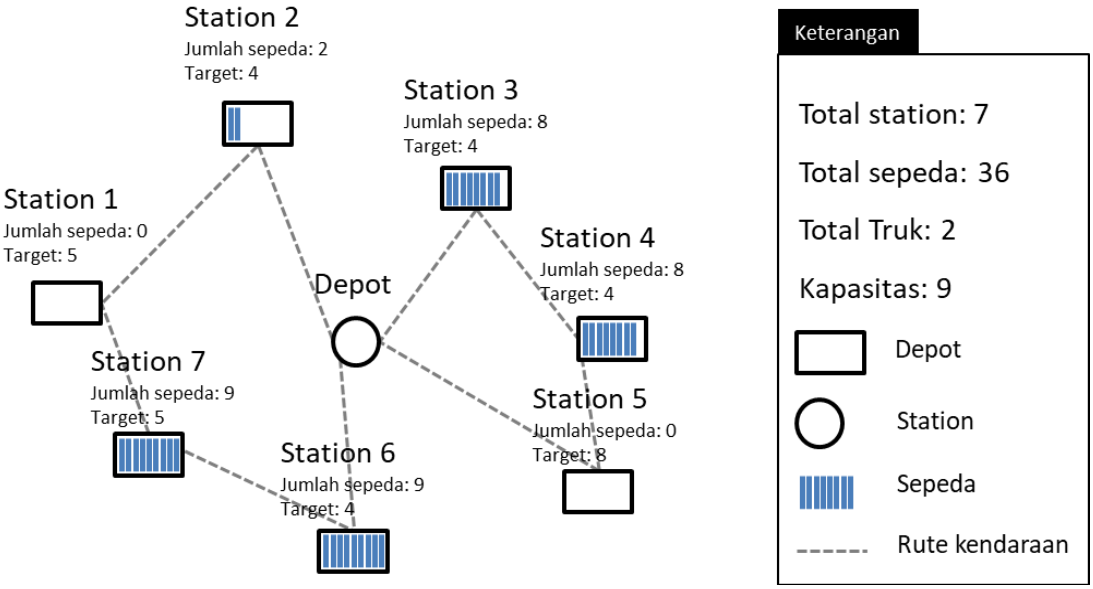
Penentuan representasi solusi merupakan salah satu proses penting dalam implementasi suatu algoritma metaheuristik untuk pemecahan masalah optimasi. Representasi solusi harus dapat merepresentasikan variabel keputusan dengan baik. Variabel keputusan pada SBRP terdiri atas jumlah sepeda yang didistribusikan dari satu *docking station* ke *docking* lainnya dan rute yang dipilih oleh kendaraan/truk yang digunakan untuk mendistribusikan sepeda-sepeda tersebut. Seperti yang digunakan pada Ho dan Szeto [18] nilai tujuan jumlah sepeda di setiap *station* sudah ditentukan menggunakan nilai prediksi yang berasal dari pendekatan data historis. Kemudian, representasi solusi digunakan sebagai wadah yang memperlihatkan suatu *station* perlu ditambahkan sepeda atau dikurangi sepedanya. Selain itu, fungsi penalti digunakan jika waktu yang dibutuhkan untuk operasi distribusi sepeda yang direncanakan melebihi limitasi waktu yang diberikan.

station	6	7	1	0	2	3	4	5
#sepeda	+5	+4	-5	0	-4	+4	+4	-8

Gambar 2. Representasi solusi untuk SBRP

Detail dari representasi solusi untuk SBRP dijelaskan sebagai berikut. Representasi solusi menggunakan *array*/larik yang berbentuk permutasi dari lokasi n station yang dikunjungi, di mana didefinisikan dalam himpunan station *N = {1,2,.., n}* seperti diilustrasikan pada Gambar 2. Hal ini menunjukkan rute yang dikunjungi oleh kendaraan. Terkait dengan berapa jumlah sepeda yang diangkut dari dan menuju setiap *station*, ditentukan oleh selisih jumlah sepeda pada kondisi terakhir dengan prediksi jumlah sepeda yang diinginkan. Jika ada suatu *station* yang tidak dapat dilayani oleh kendaraan, maka *station*

http://dx.doi.org/10.30656/jsmi.v3i1.1473



Gambar 3. Ilustrasi solusi dari *static bicycles routing problem*

tersebut tidak dikunjungi (Gambar 3). Angka pada representasi solusi menunjukkan bahwa kendaraan dapat kembali.

Struktur *Neighborhood*

Seperti yang dijelaskan sebelumnya kedua versi SA yang digunakan pada penelitian ini menggunakan prosedur *neighborhood standar* yang terdiri atas: *swap, reverse* dan *insert*. Ketiga prosedur tersebut dipilih secara acak dalam setiap iterasi untuk menghasilkan solusi baru. Notasi *N(X)* merupakan notasi yang digunakan untuk merepresentasikan bahwa prosedur yang didefinisikan dalam *neighborhood* digunakan untuk menghasilkan solusi baru dari solusi awal *X*. Dengan ini dapat dihasilkan solusi baru *Y* yang merupakan hasil dari *N(X)* yakni prosedur dari salah satu diantara *swap, reverse, dan insert*. Metode *neighborhood* ini merupakan adaptasi dari metode *neighborhood* yang pernah diterapkan pada algoritma SA pada penelitian terdahulu [14], [19], dan [20].

Dalam prosedur *swap* dilakukan penukaran nilai dari dua posisi the *i* dan *j* pada solusi *X* dimana *i* dan *j* dipilih secara acak. Prosedur lainnya yaitu *reverse*, dua posisi *i* dan *j* dipilih secara acak kemudian urutan nilai diantara dua posisi tersebut dibalik. Terakhir, *insertion* adalah mengambil nilai dari suatu posisi yang dipilih secara acak yaitu *i* kemudian menyisipkan nilai tersebut setelah posisi yang dipilih secara acak lainnya yaitu *j*. Peluang terpilihnya salah satu diantara *neighborhood* tersebut diatur sama berdasarkan distribusi *uniform random* dimana masing-masing mendapatkan proporsi sebanyak 1/3. Setelah mendapatkan solusi baru berdasarkan *neighborhood* yang terpilih, maka solusi rute yang dihasilkan harus dihitung dahulu nilai objektifnya sebelum dibandingkan dengan solusi sebelumnya.

Swap

Sebelum	6	7	1	2	3	4	5
Sesudah	6	1	7	2	3	4	5

Insert

Sebelum	6	7	1	2	3	4	5
Sesudah	6	1	2	3	4	7	5

Reverse

Sebelum	6	7	1	2	3	4	5
Sesudah	6	3	2	1	7	4	5

Gambar 4. Ilustrasi *neighbourhood swap, insert*, dan *reverse*

Untuk lebih memperjelas prosedur masing-masing *neighborhood*, dapat dilihat pada Gambar 4. Pada prosedur *swap* posisi 2 dan 3 dipilih *station* 7 dan 1. Kemudian *station* pada kedua posisi ini ditukar posisinya yang mengakibatkan solusi awal dengan urutan 6,7,1,2,3,4,5 menjadi solusi baru dengan urutan 6,1,7,2,3,4,5. Penjelasan mengenai *insert* yakni pertama-tama

 http://dx.doi.org/10.30656/jsmi.v3i1.1473

dua posisi dipilih, contohnya posisi 2 dan 6 yakni station 7 dan 4. Kemudian, station di posisi 2 dipindahkan ke posisi setelah 6 menghasilkan solusi baru dengan urutan 6,1,2,3,4,7,5. Langkah awal prosedur *reverse* sama dengan prosedur *swap* dan *insert* yaitu memilih dua posisi, contohnya posisi 2 dan 5. Urutan *station* diantara dan kedua posisi tersebut dibalik sehingga menghasilkan solusi baru dengan urutan 6,3,2,1,7,4,5.

Parameter SA

SA yang diajukan pada penelitian ini memerlukan lima parameter yaitu *Iiter, T0, TF, Nnon-improving* dan α.Iiter merupakan jumlah iterasi pada setiap temperatur. *T0* merupakan temperatur inisial. *TF* adalah temperatur akhir. *Nnon-improving* jumlah maksimum iterasi, dimana fungsi objektif dari solusi yang dicoba tidak mampu memperoleh objektif yang lebih baik dari solusi terbaik yang sudah pernah ditemukan. Terakhir, α adalah koefisien yang mengontrol kecepatan penurunan suhu.

HASIL DAN PEMBAHASAN

Eksperimen dilakukan pada komputer dengan spesifikasi CPU 3.4 Ghz, RAM 16GB dengan menggunakan sistem operasi Windows 10. Algoritma SA sendiri, diimplementasikan dengan bahasa pemrograman C++ dengan menggunakan Microsoft Visual Studio 2017. Penentuan parameter pada algoritma SA menggunakan pendekatan seting parameter dengan metode *One Factor At a Time (OFAT)*. Dalam metode ini setiap parameter diatur satu persatu dengan mengasumsikan parameter lain yang tidak sedang diatur bernilai tetap. Kemudian, parameter yang diatur dipilih berdasarkan pertimbangan nilai objektif terbaik dan waktu komputasi. Dari metode ini didapatkan parameter yang selanjutnya digunakan untuk eksperimen yaitu *Iiter=9000*, *T0=50*, *TF=0.001*, *Nnon-improving=5000*, dan *α=0.99.*

Data yang digunakan sebagai *benchmark* pada SBRP ini berasal dari data Velib bike sharing di Paris, berdasarkan penelitian Raviv et al. [4]. Hasil komputasi algoritma SA kemudian dibandingkan dengan hasil algoritma yang digunakan oleh Raviv et al. [4], dan Ho dan Szeto [18]. Jumlah *docking station* yang dipertimbangkan pada data tersebut adalah 30, 45, dan 60, dimana data dengan jumlah *station* lebih kecil adalah subset dari yang lebih besar jumlah *station*nya. Jumlah kendaraan yang tersedia untuk melakukan distribusi sepeda berjumlah satu kendaraan pada data set pertama dan dua pada data set kedua. Kapasitas setiap kendaraan adalah 20 buah sepeda. Dengan waktu total yang tersedia untuk distribusi sepeda diujikan pada 2.5 jam dan 5 jam. Asumsi waktu untuk melakukan *loading* ataupun *unloading* adalah 1 menit per sepeda.

Eksperimen pada data pertama

Tabel 1 memperlihatkan hasil eksperimen dari algoritma SA pada data *set* pertama. Pada data *set* ini dilakukan perbandingan dengan SBRP yang dicari solusinya berdasarkan formulasi *Arc-Indexed (AI)* oleh Raviv et al. [4]. AI di-implementasikan menggunakan *software* komersil bernama CPLEX yang merupakan *software* khusus untuk pemecahan permasalah-an optimasi, dengan parameter pada batasan waktu *solver* diatur maksimal selama 2 jam. Kolom pertama dan kedua menunjukkan nilai solusi *feasible* terbaik dan waktu komputasi dari solusi berdasarkan AI. *Best feasible solution,* dalam artian adalah solusi terbaik yang dapat ditemukan oleh *solver* selama batasan waktu tertentu. Kolom berikutnya berhubungan dengan algoritma menampilkan fungsi objektif yang diperoleh SA_BF dengan AI adalah 2.21%, sedangkan SA_CF dengan AI adalah 0.36%. Hasil tersebut diperoleh dengan waktu komputasi yang relatif cepat yakni rata-rata 0.51 detik dan 0.41 detik untuk SA_BF dan SA_CF.

Eksperimen pada data kedua

Eksperimen kedua pada Tabel 2 mem-bandingkan performa algoritma SA_BF dan SA_CF dengan algoritma *tabu search* yang dikembangkan oleh dan Ho dan Szeto [18]. Untuk data set ini jumlah kendaraan yang digunakan untuk melakukan reposisi sepeda adalah satu buah. Rata-rata waktu komputasional yang diperlukan oleh algoritma *tabu search, SA_BF* dan SA_CF adalah 0.31 detik, 0.44 detik, dan 0.38. Rata-rata persentase selisih antara algoritma tabu search, SA_BF, dan SA_CF dengan solusi optimal adalah 0.65% and 1.08%, dan 0.31%. Dalam dataset kedua ini algoritma SA_BF, masih tergolong kompetitif untuk digunakan menyelesaikan permasalahan SBRP. Sedangkan hasil dari algoritma SA_CF sedikit lebih baik daripada algoritma *tabu search*. Hasil tersebut dapat ditingkatkan lagi menjadi lebih baik dengan mengadopsi pendekatan-pendekatan yang terkini pada riset yang berhubungan dengan metaheuristik seperti halnya *hybrid* antara metaheuristik dan pendekatan eksak.

http://dx.doi.org/10.30656/jsmi.v3i1.1473

Tabel 1. Perbandingan hasil komputasi antara SA dan AI formulation (Raviv et al. [4])

No	*Best feasible solution (unit cost)*	CPU1 *(detik)*	SA_BF *(unit cost)*	SA_CF *(unit cost)*	Gap1 (%)	Gap2 (%)	CPU1 (detik)	CPU2 (detik)
1	214.24	22.75	214.74	214.74	0.23	0.23	0.335	0.135
2	167.07	523.75	170.33	167.07	1.91	0.00	0.234	0.334
3	170.84	-	172.21	172.21	0.80	0.80	0.573	0.472
4	116.83	-	118.12	116.83	1.09	0.00	0.45	0.125
5	460.87	72000	467.23	460.87	1.36	0.00	0.375	0.323
6	382.6	638.1	389.9	389.9	1.87	1.87	0.886	0.752
7	404.48	-	418.82	404.48	3.42	0.00	0.57	0.19
8	310.14	-	333.42	310.14	6.98	0.00	0.688	0.921
	278.38		285.60	279.53	2.21	0.36	0.51	0.41

Tabel 2. Perbandingan hasil komputasi antara SA dan tabu search (Ho dan Szeto [18])

No	Opt *(unit cost)*	CPU *(detik)*	Tabu *(unit cost)*	SA_BF *(unit cost)*	SA_CF *(unit cost)*	Gap 1 (%)	Gap 2 (%)	Gap 3 (%)	CPU1 *(detik)*	CPU2 *(detik)*	CPU3 *(detik)*
1	214.16	22.75	215.38	214.74	214.74	0.57	0.27	-0.30	0.225	0.335	0.535
2	167.07	523.75	167.79	170.33	167.79	0.43	1.91	0.00	0.245	0.234	0.341
3	335.05	790.75	336.88	336.88	336.88	0.54	0.54	0.00	0.305	0.336	0.431
4	273.09	757.04	275.98	275.98	275.98	1.05	1.05	0.00	0.375	0.456	0.154
5	463.45	72000	463.23	467.23	465.23	-0.05	0.81	0.43	0.23	0.375	0.271
6	382.52	638.1	387.9	389.9	387.9	1.39	1.89	0.00	0.45	0.886	0.56
	305.89	12455.40	307.86	309.18	308.09	0.65	1.08	0.07	0.31	0.44	0.38

KESIMPULAN

Penelitian ini fokus pada pemecahan masalah optimasi posisi sepeda pada *public bike-sharing system* yang dikenal dengan *Static Bike Routing Problem* (SBRP). Untuk itu, dikembangkan dua variasi algoritma SA yang menggunakan 3 metode generator solusi baru / *neighborhood* sederhana yaitu *swap, reverse,* dan *insert*. Algoritma tersebut disebut dengan SA_BF dan SA_CF. Performa dan efektifitas dari algoritma SA telah diuji pada data set yang dirancang oleh Raviv et al. [4]. berdasarkan data set dari the Velib bike sharing system di Paris. Hasil kedua algoritma SA ini dibandingkan dengan hasil dari pendekatan eksak dan algoritma *tabu search*. Eksperimen yang dilakukan menunjukkan bahwa performa algoritma SA cukup kompetitif di-bandingkan pendekatan pada penelitian sebelumnya. Dimana pada *dataset* pertama persentase selisih nilai fungsi tujuan SA_BF dengan AI adalah 2.21%, sedangkan SA_CF dengan AI adalah 0.36%. Hasil tersebut diperoleh dengan waktu komputasi yang relatif cepat yakni rata-rata 0.51 detik dan 0.41 detik untuk SA_BF dan SA_CF. Sedangkan pada *dataset* kedua. Rata-rata waktu komputasional yang diperlukan oleh algoritma *tabu search, SA_BF* dan SA_CF adalah 0.31 detik, 0.44 detik, dan 0.38. Rata-rata persentase selisih antara algoritma tabu search, SA_BF, dan SA_CF dengan solusi optimal adalah 0.65% and 1.08%. Hal ini menunjukkan bahwa algoritma ini dapat digunakan untuk memecahkan masalah SBRP yang membutuhkan rute kendaraan dengan total biaya operasi dan biaya penalti yang minimal.

Arah pengembangan penelitian ini adalah mempertimbangkan lebih banyak karakteristik permasalahan yang terjadi pada sistem *bike sharing*. Kemudian data set dengan jumlah *station* ataupun sepeda yang lebih banyak perlu diperhatikan. Selain itu pengembangan algoritma dan eksperimen untuk penyelesaian masalah SBRP dengan metode metaheuristik lainnya perlu dilakukan.

DAFTAR PUSTAKA

[1] S. D. Parkes, G. Marsden, S. A. Shaheen, and A. P. Cohen, “Understanding the diffusion of public bikesharing systems: evidence from Europe and North America,” *J. Transp. Geogr.*, vol. 31, pp. 94–103, Jul. 2013, doi: 10.1016/j.jtrangeo.2013.06.003.

[2] C.-C. Hsu, J. J. H. Liou, H.-W. Lo, and Y.-

C. Wang, “Using a hybrid method for evaluating and improving the service quality of public bike-sharing systems,” *J. Clean. Prod.*, vol. 202, pp. 1131–1144, Nov. 2018, doi: 10.1016/j.jclepro.2018.08.193.

[3] R. Alvarez-Valdes *et al.*, “Optimizing the level of service quality of a bike-sharing system,” *Omega*, vol. 62, pp. 163–175, Jul. 2016, doi: 10.1016/j.omega.2015.09.007.

[4] T. Raviv, M. Tzur, and I. A. Forma, “Static repositioning in a bike-sharing system: models and solution approaches,” *EURO J. Transp. Logist.*, vol. 2, no. 3, pp. 187–229, Aug. 2013, doi: 10.1007/s13676-012-0017-6.

[5] J.-R. Lin and T.-H. Yang, “Strategic design of public bicycle sharing systems with service level constraints,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 47, no. 2, pp. 284–294, Mar. 2011, doi: 10.1016/j.tre.2010.09.004.

[6] T. Raviv and O. Kolka, “Optimal inventory management of a bike-sharing station,” *IIE Trans.*, vol. 45, no. 10, pp. 1077–1093, Oct. 2013, doi: 10.1080/0740817X.2013.770186.

[7] M. Rainer-Harbach, P. Papazek, B. Hu, and G. R. Raidl, “Balancing Bicycle Sharing Systems: A Variable Neighborhood Search Approach,” in *European conference on evolutionary computation in combinatorial optimization*, Springer, 2013, pp. 121–132, doi: 10.1007/978-3-642-37198-1_11.

[8] H. M. Espegren, J. Kristianslund, H. Andersson, and K. Fagerholt, “The Static Bicycle Repositioning Problem - Literature Survey and New Formulation,” in *International Conference on Computational Logistics*, Springer, 2016, pp. 337–351, doi: 10.1007/978-3-319-44896-1_22.

[9] I. A. Forma, T. Raviv, and M. Tzur, “A 3-step math heuristic for the static repositioning problem in bike-sharing systems,” *Transp. Res. Part B Methodol.*, vol. 71, pp. 230–247, Jan. 2015, doi: 10.1016/j.trb.2014.10.003.

[10] C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl, “Balancing Bicycle Sharing Systems: An Approach for the Dynamic Case,” in *European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, 2014, pp. 73–84, doi: 10.1007/978-3-662-44320-0_7.

[11] L. Caggiani and M. Ottomanelli, “A Dynamic Simulation based Model for Optimal Fleet Repositioning in Bike-sharing Systems,” *Procedia - Soc. Behav. Sci.*, vol. 87, pp. 203–210, Oct. 2013, doi: 10.1016/j.sbspro.2013.10.604.

[12] A. A. N. Perwira Redi, M. F. N. Maghfiroh, and V. F. Yu, “Discrete Particle Swarm Optimization with Path-Relinking for Solving the Open Vehicle Routing Problem with Time Windows,” in *Proceedings of the Institute of Industrial Engineers Asian Conference 2013*, Singapore: Springer Singapore, 2013, pp. 853–859, doi: 10.1007/978-981-4451-98-7_102.

[13] V. F. Yu, P. Jewpanya, S.-W. Lin, and A. A. N. P. Redi, “Team orienteering problem with time windows and time-dependent scores,” *Comput. Ind. Eng.*, vol. 127, pp. 213–224, Jan. 2019, doi: 10.1016/j.cie.2018.11.044.

[14] V. F. Yu, P. Jewpanya, and A. A. N. P. Redi, “Open vehicle routing problem with cross-docking,” *Comput. Ind. Eng.*, vol. 94, pp. 6–17, Apr. 2016, doi: 10.1016/j.cie.2016.01.018.

[15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953, doi: 10.1063/1.1699114.

[16] F. Y. Vincent, A. A. N. P. Redi, Y. A. Hidayat, and O. J. Wibowo, “A simulated annealing heuristic for the hybrid vehicle routing problem,” *Appl. Soft Comput.*, vol. 53, pp. 119–132, 2017, doi: 10.1016/j.asoc.2016.12.027.

[17] S.-W. Lin, V. F. Yu, and S.-Y. Chou, “Solving the truck and trailer routing problem based on a simulated annealing heuristic,” *Comput. Oper. Res.*, vol. 36, no. 5, pp. 1683–1692, May 2009, doi: 10.1016/j.cor.2008.04.005.

[18] S. C. Ho and W. Y. Szeto, “Solving a static repositioning problem in bike-sharing systems using iterated tabu search,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 69, pp. 180–198, Sep. 2014, doi: 10.1016/j.tre.2014.05.017.

[19] V. F. Yu, A. A. N. P. Redi, P. Jewpanya, A. Lathifah, M. F. N. Maghfiroh, and N. A. Masruroh, “A Simulated Annealing Heuristic for the Heterogeneous Fleet Pollution Routing Problem,” in *Environmental Sustainability in Asian Logistics and Supply Chains*, Singapore:

Springer Singapore, 2019, pp. 171–204, doi: 10.1080/0305215X.2018.1437153.

[20] V. F. Yu, S. S. Purwanti, A. A. N. P. Redi, C.-C. Lu, S. Suprayogi, and P. Jewpanya, “Simulated annealing heuristic for the general share-a-ride problem,” *Eng. Optim.*, vol. 50, no. 7, pp. 1178–1197, Jul. 2018, doi: 10.1080/0305215X.2018.1437153.