

# Real-Time Location Recommendation for Field Data Collection

Aris Prawisudatama

School of Electrical Engineering and Informatics  
Institut Teknologi Bandung  
Bandung, Indonesia  
Email: soedomoto@gmail.com

I Gusti Bagus Baskara Nugraha

School of Electrical Engineering and Informatics  
Institut Teknologi Bandung  
Bandung, Indonesia  
Email: baskara@stei.itb.ac.id

**Abstract**—Field data collection is one of the main activities performed by the statistical agencies of a country. Data collection activities have a common workflow with Multi Depot Vehicle Routing Problem (MDVRP). The use MDVRP to generate pre-calculated routes resulted in a total route costs with high standard deviation. Real-time mechanism by utilizing the paradigm of publish / subscribe combined with COEs MDVRP algorithm is proposed to reduce the inequality (large variation) the time of completion. The test results show that the mechanisms pub / sub combined with COEs produce a total route times more prevalent among enumerators compared with pre-calculated COEs algorithm.

## I. INTRODUCTION

Pengumpulan data lapangan merupakan salah satu tugas dan wewenang dari lembaga statistik suatu negara. Terdapat dua macam metode pengumpulan data yang digunakan, yaitu: pengumpulan data primer dan pengumpulan data sekunder. Pada pengumpulan data primer, petugas pengumpulan data secara langsung melakukan interview dengan responden, sementara pada pengumpulan data sekunder, instansi melakukan kompilasi data yang telah dilakukan oleh pihak lain. Pada proses pengumpulan data (primer), petugas pengumpulan data dialokasikan pada sejumlah lokasi pencacahan yang telah ditentukan. Mekanisme pengalokasian yang biasanya digunakan adalah alokasi dengan jumlah lokasi yang sama antar petugas.

Alur kerja pada proses pengumpulan adalah, setiap petugas pengumpulan data memulai pengumpulan data dari sebuah titik tertentu, baik itu kantor maupun tempat tinggal masing-masing petugas. Kemudian petugas melakukan pengumpulan data pada sebuah lokasi sampai selesai, kemudian dilanjutkan dengan lokasi berikutnya, dan seterusnya sampai seluruh lokasi pencacahan dikunjungi. Alur kerja pengumpulan data ini sangat mirip dengan *Vehicle Routing Problem* (VRP), atau lebih spesifik lagi *Multi Depot VRP*.

Akan tetapi, meskipun masalah pengumpulan data memiliki kesamaan alur kerja dengan MDVRP, akan tetapi implementasi algoritma MDVRP secara mentah tidak dapat menjadi solusi atas permasalahan alokasi petugas pengumpulan data, karena salah satu komponen yang penting, yaitu lamanya waktu pelayanan, tidak tersedia sampai dengan lokasi dikunjungi. This research aims to propose an integration antara MDVRP dengan mekanisme pub/sub untuk menciptakan sistem rekomendasi lokasi pencacahan secara real-time.

## II. LITERATURE REVIEW

### A. Multi-Depot Vehicle Routing Problem

*Multi-Depot Vehicle Routing Problem* (MDVRP) merupakan salah satu varian dari VRP klasik dimana terdapat lebih dari satu *depot* yang digunakan [1]. Gambar 1 menunjukkan contoh solusi dari permasalahan VRP yang menggunakan dua depot dan dua buah rute kendaraan yang berasosiasi dengan masing-masing depot. Basically, a solution to this problem is a set of vehicle routes such that: (i) each vehicle route starts and ends at the same depot, (ii) each customer is served exactly once by one vehicle, (iii) the total demand on each route does not exceed vehicle capacity (iv) the maximum route time is satisfied and (v) the total cost is minimized.

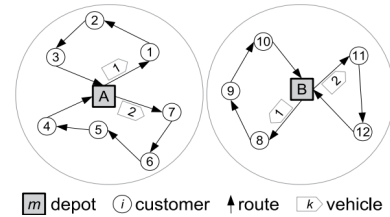


Fig. 1. Ilustrasi *Multi Depot VRP*

The MDVRP can be formalized as follows. Let  $G = (V, A)$  be a complete graph, where  $V$  is the set of nodes and  $A$  is the set of arcs. The nodes are partitioned into two subsets: the customers to be served,  $V_C = \{1, \dots, N\}$ , and the multiple depots  $V_D = \{N+1, \dots, N+M\}$ , with  $V_C \cup V_D = V$  and  $V_C \cap V_D = \emptyset$ . There is a non-negative cost  $c_{ij}$  associated with each arc  $(i, j) \in A$ . The demand of each customer is  $d_i$  (there is no demand at the depot nodes). There is also a fleet of  $K$  identical vehicles, each with capacity  $Q$ . The service time at each customer  $i$  is  $t_i$  while the maximum route duration time is set to  $T$ . A conversion factor  $w_{ij}$  might be needed to transform the cost  $c_{ij}$  into time units. In this work, however, the cost is the same as the time and distance units, so  $w_{ij} = 1$ .

In the mathematical formulation that follows, binary variables  $x_{ijk}$  are equal to 1 when vehicle  $k$  visits node  $j$

immediately after node  $i$ . Auxiliary variables  $y_i$  are also used in the subtour elimination constraints.

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^K c_{ij} x_{ijk}; \quad (1)$$

$$\sum_{i=1}^{N+M} \sum_{k=1}^K x_{ijk} = 1 (j = 1, \dots, N); \quad (2)$$

$$\sum_{j=1}^{N+M} \sum_{k=1}^K x_{ijk} = 1 (j = 1, \dots, N); \quad (3)$$

$$\sum_{i=1}^{N+M} x_{ihk} - \sum_{j=1}^{N+M} x_{hjk} = 0 \quad (k = 1, \dots, K; h = 1, \dots, N + M); \quad (4)$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} d_i x_{ijk} \leq Q (k = 1, \dots, K); \quad (5)$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} (c_{ij} w_{ij} + t_i) x_{ijk} \leq T (k = 1, \dots, K); \quad (6)$$

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^N x_{ijk} \leq 1 (k = 1, \dots, K); \quad (7)$$

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^N x_{ijk} \leq 1 (k = 1, \dots, K); \quad (8)$$

$$y_i - y_j + (M + N) x_{ijk} \leq N + M - 1; \quad \text{for } 1 \leq i \neq j \leq N \text{ and } 1 \leq k \leq K; \quad (9)$$

$$x_{ijk} \in \{0, 1\} \forall i, j, k; \quad (10)$$

$$y_i \in \{0, 1\} \forall i; \quad (11)$$

The objective 1 minimizes the total cost. Constraints 2 and 3 guarantee that each customer is served by exactly one vehicle. Flow conservation is guaranteed through constraint 4. Vehicle capacity and route duration constraints are found in 5 and 6, respectively. Constraints 7 and 8 check vehicle availability. Subtour elimination constraints are in 9. Finally, 10 and 11 define  $x$  and  $y$  as binary variables.

## B. Evolution Algorithms

*Evolution Algorithms (EAs)* merupakan salah satu *family* dari algoritma optimasi yang terinspirasi dari alam. Pada algoritma evolusi, terdapat proses seleksi dan reproduksi untuk memurnikasi populasi dari kandidat solusi [2]. Secara umum *lifecycle* dari *Evolution Algorithms* diawali dengan konfigurasi populasi awal secara random. Kemudian, setiap iterasi dimulai dengan evaluasi atas fungsi obyektif atas individu pada populasi. Berdasarkan hasil evaluasi, kemudian *fitness value* disematkan pada setiap kandidat solusi pada populasi. Kandidat solusi tersebut kemudian mengalami reproduksi dan menghasilkan generasi berikutnya. Proses kemudian berulang dengan menggunakan individu generasi yang baru.

Coevolutionary (CoE) algorithms (CoEA), merupakan pengembangan dari EAs. In standard EAs, evolution is usually viewed as if the population attempts to adapt in a fixed physical environment. In contrast, coevolutionary (CoE) algorithms (CoEA) realize that in natural evolution the physical environment is influenced by other independently-acting biological populations [3]. CoEA dapat dibagi menjadi dua, tergantung interaksi dari masing-masing spesies: *competitive* dan *cooperative*. Pada *competitive coevolution* setiap individu berkompetisi dengan satu kelompoknya. Sementara pada *cooperative coevolution*, interaksi antar spesies bersifat saling menguntungkan, atau paling tidak, tidak saling membahayakan [3].

## C. Publish/Subscribe Paradigm

*Publish/subscribe interaction* merupakan salah satu metode komunikasi antara *client* dan *server*. Paradigma interaksi pada *publish/subscribe* adalah adanya *subscriber* yang memiliki ketertarikan pada suatu *event* atau *pattern of event*, agar dapat dikirimkan notifikasi tentang sebuah *event* oleh *publisher* yang sesuai dengan *interest*-nya [2].

Model dasar dari sistem *publish/subscribe*, seperti Gambar 2, bergantung pada *event notification service* yang menyediakan penyimpanan dan *management of subscription*. *Event service* tersebut berperan sebagai mediator antara *publisher* yang berperan sebagai *producer* dari *event* dan *subscriber* yang berperan sebagai *consumer* dari *event*.

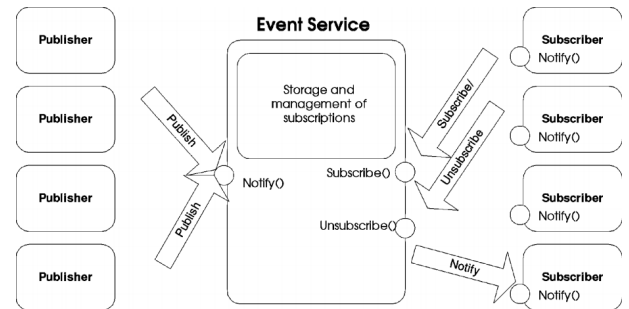


Fig. 2. Arsitektur dasar pada Pub/Sub [2]

Salah satu keuntungan dari mekanisme *publish/subscribe* adalah bersifat *loose coupling* [2] antara *publisher* dengan *subscriber*. Pemisahan informasi (*decoupling*) yang terjadi

antara *subscriber* dan *publisher* dapat dipisahkan dalam 3 (tiga) dimensi, yaitu:

1) *Space decoupling*.

Interaksi antara *publisher* dan *subscriber* tidak perlu saling mengetahui satu dengan yang lain. *Publisher* mengirimkan *event*, dan *subscriber* menerima *event* secara tidak langsung melalui *event service*. *Publisher* biasanya tidak memegang referensi terhadap *subscriber*. Begitu juga sebaliknya, *subscriber* biasanya tidak memegang referensi terhadap *publisher*.

2) *Time decoupling*.

Pihak-pihak yang berinteraksi, tidak harus berinteraksi pada waktu yang bersamaan. *Publisher* dapat mengirimkan *event* pada saat *subscriber* dalam kondisi terputus. Begitu juga sebaliknya, *subscriber* tetap dapat menerima *event*, meskipun *original publisher* dalam kondisi terputus.

3) *Synchronizing decoupling*.

*Publisher* tidak diblok ketika memproduksi *event*, dan juga *subscriber* tetap dapat memperoleh informasi meskipun sedang mengerjakan tugas yang lain. *Publisher* dan *subscriber* tidak berada dalam *main flow*, sehingga tidak berinteraksi secara *synchronous*.

### III. PROPOSED SOLUTION

Kelemahan MDVRP ketika digunakan dalam rekomendasi lokasi pencacahan adalah ketiadaan data service time, padahal data itu sangat penting dalam kalkulasi rekomendasi. Untuk itu, MDVRP perlu diintegrasikan dengan mekanisme real-time. Terdapat beberapa mekanisme yang dapat diadopsi pada real-time system, diantaranya: Web service, RPC, message passing, dan publish/subscribe [2].

Mekanisme publish/subscribe digunakan dalam penelitian ini karena memiliki sifat *loose coupling*. Selain itu mekanisme publish/subscribe juga sesuai digunakan pada sistem yang bersifat information driven [4]. Karena karakteristiknya ini, maka publish/subscribe dapat bekerja secara asynchronous, dimana *request* dan *reply* tidak harus diproses secara berurutan. Pada saat *subscriber* melakukan subscribe, *publisher* tidak harus dalam kondisi online, begitu juga sebaliknya, pada saat *publisher* melakukan publish, *subscriber* tidak harus dalam kondisi online.

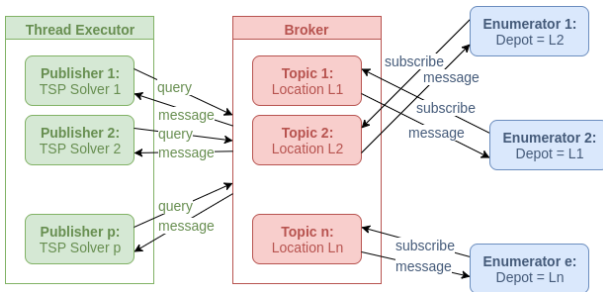


Fig. 3. Garis Besar Sistem

Gambar 3 menunjukkan garis besar sistem, dimana sistem tersusun atas 3 (tiga) komponen utama, yaitu: *publisher*,

*subscriber*, dan *message broker*. Komunikasi antara *publisher* dan *subscriber* didasari atas satu kesamaan, yaitu **event** atau **topic**. Topik yang digunakan sebagai dasar komunikasi antara *publisher* dan *subscriber* dalam penelitian ini adalah *current location* dari masing-masing *subscriber*.

#### A. Recommendation Publisher

VRP Solver diimplementasikan pada sisi *publisher*, yang akan melakukan pencarian solusi/rute dan kemudian dipublish kepada *subscriber* melalui *message broker*. Sebagaimana sebelumnya disebutkan, bahwasannya *current location* dari masing-masing *subscriber* digunakan sebagai topik. Sebuah thread disiapkan, yang secara periodik akan melakukan pengecekan apakah terdapat topik yang baru pada message broker.

Setiap kali terdapat topik yang baru, sebuah thread akan disiapkan, dan topik tersebut digunakan sebagai ID dari thread yang bersangkutan. Pada thread tersebut terdapat procedure VRP solver yang akan melakukan pencarian solusi/rute. Untuk itu, sebuah threadpool disiapkan untuk menampung thread dari masing-masing topik tersebut secara berurutan berdasarkan kedatangan. Karena topik digunakan sebagai thread ID, maka ukuran dari threadpool akan sama dengan jumlah seluruh lokasi pencacahan.

Setiap thread dalam threadpool kemudian dijalankan secara berurutan, satu session untuk masing-masing thread. Setiap thread yang dijalankan, yang menggunakan topik sebagai ID, akan melakukan pencarian solusi dengan melibatkan seluruh  $M$  vehicles dan  $(unassigned)N$  locations. Pencarian solusi dilakukan pada procedure VRP solver pada section III-B.

---

#### Algorithm 1 TopicWatcher

---

**Input:**  $C$

**Output:**  $T$

```

1: for  $i = 1$  to  $len(C)$  do
2:   for  $j = 1$  to  $N$  do
3:     if  $(C_i == E_j)$  then
4:        $T_j = Thread(C_i, V_1...V_M, (unassigned)E_1...E_N)$ 
5:     end if
6:   end for
7: end for
8: return  $T$ 

```

---

Proses dalam procedure VRP solver pada masing-masing thread akan menghasilkan sejumlah minimal 1 rute, dan maksimal  $M$  rute. Setiap rute  $R_i$  yang dihasilkan kemudian akan dipublish dengan topik  $C_i$ . Setelah rute dipublish, *message broker* akan menginformasikan kepada *publisher*, jumlah *subscriber* yang menerima pesan. Adapun identitas *subscriber* yang menerima pesan tetap tidak diketahui oleh *publisher*.

Setiap rute  $R_i$  yang memiliki jumlah penerima pesan lebih dari satu, maka *thread* yang mempunyai ID  $C_{R_i}$  akan dicancel. Pembatalan *thread* ini dimaksudkan agar solusi/rute yang telah diterima oleh *subscriber* tidak dikalkulasi kembali. Terdapat sebuah kondisi, dimana pada saat thread menjalankan VRPSolver dengan ID  $C_i$ , ternyata rute yang dihasilkan tidak mengandung solusi untuk topik  $C_i$ . Pada kondisi tersebut,

maka topik  $C_i$  tersebut akan diantrikan kembali dengan hanya menyertakan dirinya sendiri sebagai pencacah, sehingga diperoleh kepastian topik  $C_i$  akan memperoleh solusi/rute.

---

**Algorithm 2** VRPWorker

---

**Input:**  $T$

**Output:**  $R$

```

1: for  $i = 1$  to  $\text{len}(T)$  do
2:    $R = \text{VRPSolver}(T_i)$ 
3:   for  $j = 1$  to  $\text{len}(R)$  do
4:      $r = \text{publish}(C_{R_j}, R_j)$ 
5:     if  $(r > 0)$  then
6:        $\text{cancelSolver}(T_j)$ 
7:     else if  $(C_{T_i} \notin C_{R_j})$  then
8:        $T_i = \text{Tthread}(C_{T_i}, V_m, (\text{unassigned})E_1 \dots E_N)$ 
9:     end if
10:  end for
11: end for
12: return  $R$ 

```

---

Salah satu keuntungan mekanisme publish/subscribe, yang membuat dia fleksible, adalah karakteristiknya yang loose coupling. Dalam kasus ini, keuntungan dari mekanisme ini sekaligus menjadi kelemahan. Ketidaktahuan *publisher* akan identitas *subscriber* menyebabkan *current location* dari keseluruhan *subscriber* tidak dapat diketahui. Untuk itu diperlukan mekanisme yang dapat mengatasi permasalahan pertukaran identitas, yaitu dengan menggunakan *shared memory*. Pada *shared memory* tersimpan data *current location* dari seluruh *subscriber*, yang mana *publisher* dapat mengetahui *current location* dari seluruh *subscriber* dari data yang tersimpan pada *shared memory*.

Proses tersebut diatas terus berulang, sampai dengan seluruh lokasi pencacahan semuanya ter-assign. Gambar 4 memberikan ilustrasi algoritma yang digunakan pada *recommendation publisher*.

### B. VRP Solver

VRP Solver merupakan module yang digunakan untuk melakukan pencarian solusi/rute yang akan dikunjungi. VRP Solver dipanggil pada setiap thread yang terdapat pada thread-pool 2. Terdapat beberapa algoritma yang dapat digunakan dalam pencarian solusi pada MDVRP, antara lain: tabu search [5], adaptive large neighborhood search [6], fuzzy logic guided genetic algorithm [7], paralel iterated tabu search [8], hybrid algorithm combining Iterated Local Search and Set Partitioning [9], hybrid genetic algorithm with adaptive diversity control [10], hybrid Granular Tabu Search [11], dan *Evolution Algorithms (EAs)* [12]. Pada kasus ini akan digunakan coevolutionary algoritma CoEAs, karena CoEAs menghasilkan *mean solution values* yang competitive dengan *CPU time* yang relatif lebih rendah dibandingkan algoritma yang lain [12].

Langkah-langkah yang dilakukan pada VRP Solver yang mengadopsi algoritma CoEAs adalah sebagai berikut

#### 1) Create problem

MDVRP problem yang akan disusun terdiri dari sejumlah

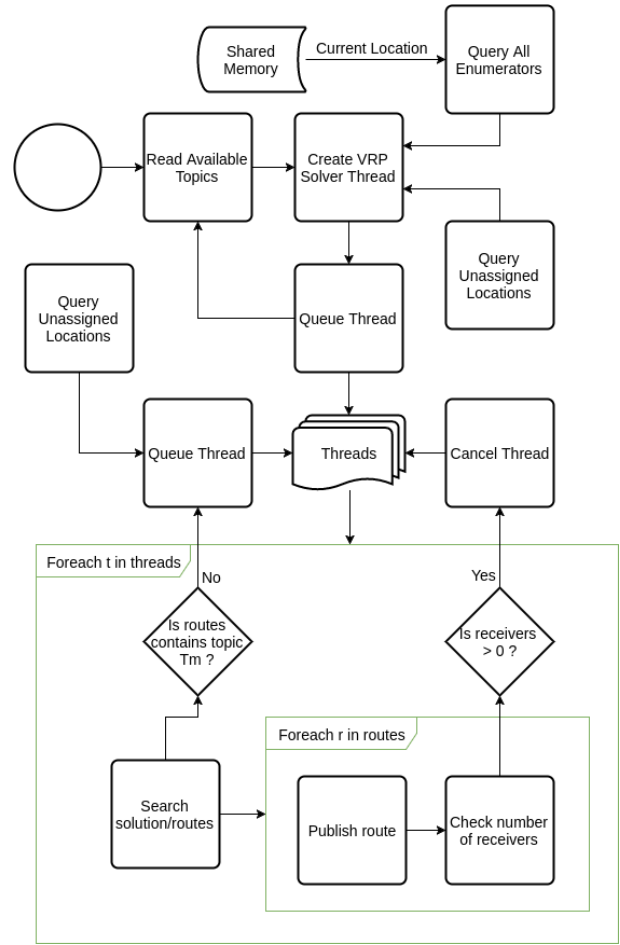


Fig. 4. Publisher Workflow

vehicle  $V$  dan sejumlah edge  $E$  yang telah diassign pada masing-masing thread.

#### 2) Decompose problem.

MDVRP problem didecompose menjadi beberapa sub-problem.

#### 3) Evolve.

Masing-masing individu pada setiap subproblem kemudian berevolusi. Setiap evolusi berakhir, individu baru akan dievaluasi. Jika individu baru yang terbentuk lebih baik, maka solusi diterima, dan individu-individu terbaik diupdate. Evolusi akan terus berlangsung sampai stop kriteria terpenuhi. Stop kriteria yang digunakan adalah waktu. Evolusi akan berhenti ketika total waktu mencapai 60 detik, atau 40 detik tanpa perubahan individu terbaik.

### C. Message Broker

*Message broker* adalah sebuah komponen yang bertanggung jawab dalam menyalurkan (*routing*) pesan dari *publisher* ke *subscriber* sesuai dengan topik yang *unsubscribe* [13]. Suatu sistem *publish/subscribe* dapat memiliki *single broker* maupun *multi broker*. Pada arsitektur *single broker*, seluruh *subscriber* dan *publisher* terkoneksi pada satu *broker*, sementara pada *multi broker*, setiap *subscriber* maupun *publisher*

**Algorithm 3** VRPWorker**Input:**  $T_i$ **Output:**  $R$ 

```

1:  $P \leftarrow \text{createProblem}(V_1 \dots V_{T_i}, E_1 \dots E_{T_i})$ 
   Decompose problem into S subproblem
2:  $SP \leftarrow \text{decomposeProblem}(P)$ 
3: while stopCriteriaUnmet do
4:   for  $i = 1$  to  $\text{len}(SP)$  do
5:      $I \leftarrow SP_i \rightarrow \text{individuals}$ 
6:     for  $j$  to  $\text{len}(I)$  do
7:        $O_j \leftarrow I_j \rightarrow \text{evolve}()$ 
8:     end for
9:     shrink( $I, O$ )
10:  end for
11:  updateBestIndividuals()
12: end while
13:  $B \leftarrow \text{getBestIndividuals}()$ 
14:  $R \leftarrow \text{convert}(B)$ 
15: return  $R$ 

```

dapat terkoneksi pada broker terdekat. Arsitektur *multi depot* ini juga disebut dengan *distributed pub/sub system* [4], seperti ilustrasi pada Gambar 5. Adapun rancangan yang diusulkan akan menerapkan arsitektur terdistribusi, dikarenakan lokasi pencacahan secara geografis tersebar.

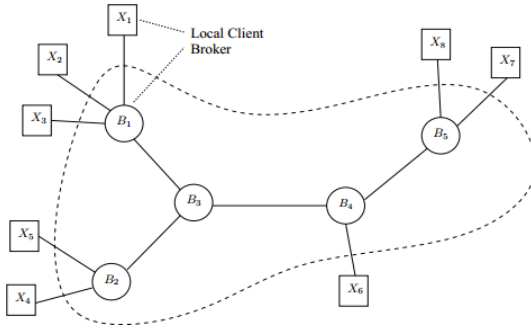


Fig. 5. Arsitektur Publish-Subscribe terdistribusi

## IV. TESTING

The performance of the proposed system was assessed through a number of experiments. Eksperimen ini menggunakan dua macam data: Cordeau and data field. Service time pada kedua macam data tersebut akan digenerate secara random dengan mengikuti distribusi normal.

- 1) Data yang diambil dari Cordeau [5], terdiri dari 50 *customers* dan 4 *vehicles*. Jarak antar lokasi dihitung dengan cara euclidean.
- 2) Data yang diambil dari kondisi real lapangan, yang terdiri dari 182 *customers* dan 15 *vehicles*. Jarak dan durasi antar lokasi diukur dengan bantuan Google Direction API.

Data yang akan digunakan, Cordeau type 2 dan data lapangan, akan diujicobakan dengan menggunakan program usulan, berbasis Pub/Sub dan CoES. Data kemudian diperbandingkan

TABLE I  
KOMPARASI COES DAN PUB/SUB + COES PADA DATA CORDEAU

| Ukuran           | CoES       | CoES + Pub/Sub |
|------------------|------------|----------------|
| Total time       | 1165513.28 | 1165706.84     |
| Mean of time     | 291378.32  | 291426.71      |
| Std. Dev of time | 98268.51   | 12997.91       |

TABLE II  
KOMPARASI COES DAN PUB/SUB + COES PADA DATA LAPANGAN

| Ukuran          | CoES       | CoES + Pub/Sub |
|-----------------|------------|----------------|
| Total Waktu     | 4448989.67 | 4559658.67     |
| Rata-rata       | 296599.31  | 303977.24      |
| Standar Deviasi | 119720.84  | 34472.12       |

dengan program pembandingan yang mengimplementasi algoritma CoES tanpa mekanisme Pub/Sub. Dari kedua program tersebut akan diperoleh output rute untuk masing-masing *vehicle* dengan ilustrasi sebagai berikut:

- *Vehicle A* = Loc1  $\rightarrow$  Loc5  $\rightarrow$  Loc15  $\rightarrow$  Loc6
- *Vehicle B* = Loc 6  $\rightarrow$  Loc2  $\rightarrow$  Loc16  $\rightarrow$  Loc3
- *Vehicle C* = Loc4  $\rightarrow$  Loc8  $\rightarrow$  Loc14  $\rightarrow$  Loc 7
- *Vehicle D* = Loc9  $\rightarrow$  Loc10  $\rightarrow$  Loc11  $\rightarrow$  Loc12

Masing-masing rute kemudian dihitung *total cost*-nya, yang merupakan penjumlahan dari seluruh *service time* pada masing-masing lokasi dan seluruh waktu tempuh dari masing-masing perpindahan. Metric yang akan dijadikan pembandingan adalah standar deviasi dari seluruh rute pada masing-masing program dan program pembandingan. Program yang lebih baik akan menghasilkan standar deviasi yang lebih kecil. Standar deviasi dipilih sebagai *metric* karena merepresentasikan kondisi sebenarnya di lapangan, dimana jika variasi waktu dari seluruh pencacah lebih kecil, maka penyelesaian pencacahan akan lebih merata.

Pada pengujian kondisi normal dengan menggunakan data Cordeau, diperoleh hasil bahwasannya pre-calculated routes dengan menggunakan algoritma CoES menghasilkan total time yang lebih kecil, yaitu 1165513.28 seconds berbanding 1165706.84 seconds untuk Pub/Sub + CoES, namun menghasilkan standar deviasi yang jauh lebih besar, yaitu 98268.51 seconds berbanding 12997.91 seconds (Table I). Sementara pengujian kondisi normal dengan menggunakan data lapangan, diperoleh hasil bahwasannya pre-calculated routes dengan menggunakan algoritma CoES menghasilkan total time yang lebih kecil, yaitu 4448989.67 seconds berbanding 4559658.67 seconds untuk Pub/Sub + CoES, namun menghasilkan standar deviasi yang jauh lebih besar, yaitu 119720.84 seconds berbanding 34472.12 seconds (Table II).

## V. CONCLUSION AND FUTURE WORKS

Paper ini mengusulkan penggunaan Cooperative Coevolutionary Algorithm (CoES) yang dikombinasikan dengan mekanisme Publish/Subscribe untuk mengatasi permasalahan

rekomendasi lokasi pengumpulan data secara real-time. Dari hasil pengujian diperoleh hasil bahwasannya pre-calculated routes yang dihasilkan dengan algoritma CoES menghasilkan total route times lebih kecil daripada CoES Pub/Sub. Akan tetapi, algoritma CoES Pub/Sub menghasilkan standar deviasi yang lebih kecil dari pada CoES, yang artinya beban yang dialokasikan untuk setiap petugas lebih merata.

#### REFERENCES

- [1] J. R. Montoya-Torres, J. Lpez Franco, S. Nieto Isaza, H. Felizzola Jimnez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Computers & Industrial Engineering*, vol. 79, pp. 115–129, Jan. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S036083521400360X>
- [2] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM computing surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003. [Online]. Available: <http://dl.acm.org/citation.cfm?id=857078>
- [3] A. P. Engelbrecht, "Coevolution," in *Computational Intelligence*. John Wiley & Sons, Ltd, 2007, pp. 275–283, doi: 10.1002/9780470512517.ch15. [Online]. Available: <http://onlinelibrary.wiley.com.proxy.lib.odu.edu/doi/10.1002/9780470512517.ch15/summary>
- [4] G. Mhl, "Large-scale content-based publish-subscribe systems," Ph.D. dissertation, TU Darmstadt, 2002. [Online]. Available: <http://tuprints.ulb.tu-darmstadt.de/274>
- [5] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, no. 2, pp. 105–119, Sep. 1997. [Online]. Available: [http://onlinelibrary.wiley.com.proxy.lib.odu.edu/doi/10.1002/\(SICI\)1097-0037\(199709\)30:2<105::AID-NET5>3.0.CO;2-G/abstract](http://onlinelibrary.wiley.com.proxy.lib.odu.edu/doi/10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G/abstract)
- [6] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, Aug. 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054805003023>
- [7] H. C. W. Lau, T. M. Chan, W. T. Tsui, and W. K. Pang, "Application of Genetic Algorithms to Solve the Multidepot Vehicle Routing Problem," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 383–392, Apr. 2010.
- [8] J.-F. Cordeau and M. Maischberger, "A parallel iterated tabu search heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 39, no. 9, pp. 2033–2050, Sep. 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054811002826>
- [9] A. Subramanian, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for a class of vehicle routing problems," *Computers & Operations Research*, vol. 40, no. 10, pp. 2519–2531, Oct. 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S030505481300021X>
- [10] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Implicit depot assignments and rotations in vehicle routing heuristics," *European Journal of Operational Research*, vol. 237, no. 1, pp. 15–28, Aug. 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S037722171301028X>
- [11] J. W. Escobar, R. Linfati, P. Toth, and M. G. Baldoquin, "A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem," *Journal of Heuristics*, vol. 20, no. 5, pp. 483–509, Oct. 2014. [Online]. Available: <http://link.springer.com.proxy.lib.odu.edu/article/10.1007/s10732-014-9247-0>
- [12] F. B. de Oliveira, R. Enayatifar, H. J. Sadaei, F. G. Guimares, and J.-Y. Potvin, "A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem," *Expert Systems with Applications*, vol. 43, pp. 117–130, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417415005771>
- [13] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajaro, R. E. Strom, and D. C. Sturman, "An efficient multicast protocol for content-based publish-subscribe systems," in *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*. IEEE, 1999, pp. 262–272. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=776528](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=776528)