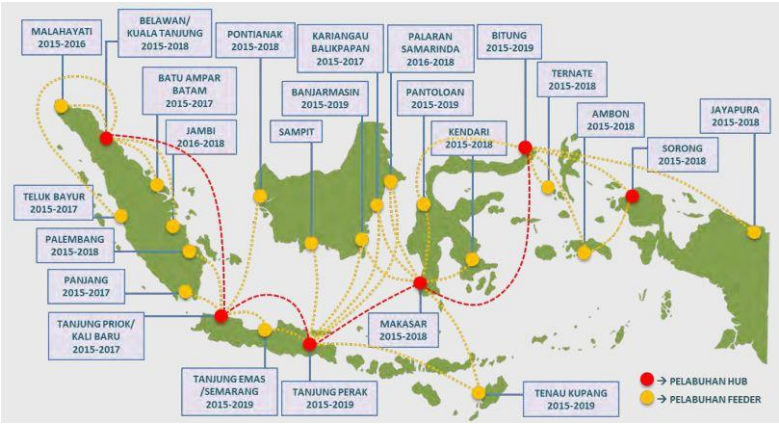
PERENCANAAN PERBAIKAN RUTE TOL LAUT MENGGUNAKAN METODE PARALLEL INSERTION DAN EXHAUSTIVE SEARCH

Hastawati Chrisna Suroso, Irfan Subiantoro, Lukmandono

PENDAHULUAN

Mayoritas wilayah Indonesia adalah lautan sebesar 63% dengan panjang garis pantai adalah 80.791 km2. Sedangkan 37% merupakan daratan dengan jumlah pulau 17.499 [1] . Dengan kondisi Indonesia yang memiliki banyak pulau dan dikelilingi oleh perairan, tol laut dirasa menjadi sebuah solusi dalam pemerataan logistik secara merata serta mampu menekan harga logistik yang beredar di masyarakat. Pada tahun 2019 ini terdapat 18 rute perjalanan tol laut dengan target frekuensi dan hari pelayaran yang berbeda pada masing-masing rute. Namun demikian dengan semakin bertambahnya rute tol laut jumlah bahan bakar yang dikonsumsi, proses loading dan unloading material, serta jumlah sumber daya manusia yang digunakan juga akan semakin meningkat dan akan berpengaruh pada harga logistik. Ditambah lagi dengan kesenjangan antara wilayah barat dan timur Indonesia dalam mendapatkan supply logistik membuat sangat diperlukannya perbaikan rute tol laut guna meratakan *supply* logistik. *Vehicle Routing Problem* (VRP) adalah sebuah masalah pada distribusi guna menentukan rute kendaraan dengan kapasitas yang ada dari satu atau dua pangkalan untuk memenuhi kebutuhan konsumen [2]. Untuk menyelesaikan VRP diperlukan kebijakan strategis dengan penataan rute yang tepat pada 24 pelabuhan strategis yang tersebar di wilayah Indonesia.



Gambar 1. Persebaran Pelabuhan di Indonesia

Beberapa penelitian yang mengangkat topik tol laut sudah diterapkan akhir akhir ini, seperti penelitian yang digagas oleh [3] mengenai pelaksanaan program tol laut di Indonesia yang membahas konsistensi rute dan frekuensi selama tiga tahun terakhir. Penelitian ini memberikan rekomendasi untuk meninjau dan meningkatkan tiga aspek yaitu rute, frekuensi, dan volume agar meningkatkan efektivitas dari tol laut ini. Efektifitas dan efisiensi tol laut juga telah dibahas oleh [4] dengan menggunakan metode *Analytical Hierarchy Process* (AHP) untuk menganalisa faktor dan subfaktor yang menjadi prioritas utama dalam pelaksanaan tol laut. Sementara itu analisa mengenai perbaikan rute tol laut belum pernah dilakukan secara rinci pada 24 pelabuhan yang menjadi titik persebaran logistik. Dalam mendukung penelitian ini, maka diperlukan sebuah metode untuk menyelesaikan VRP seperti penelitian [2] dengan menggunakan beberapa metode untuk memperbaiki aliran rute sebelumnya.

Dasar Permasalahan yang diteliti adalah *existing route* selama 2015-2019 untuk dievaluasi dan diberikan solusi yang lebih baik dalam penataan rute untuk tahun mendatang. Tujuan khusus pada penelitian ini adalah peminimalan biaya transportasi yang digunakan pada tol laut dengan mempertimbangkan jarak sebagai variabel utama. Dengan pemilihan jarak yang lebih pendek maka akan

mengurangi biaya transportasi secara umum yang digunakan dalam penyaluran logistik antar pelabuhan. Optimasi rute pelayaran tol laut dirasa sangat perlu dilakukan untuk memberikan ongkos yang minimum, jumlah armada yang minimum, keseimbangan rute, serta meminimalkan keluhan konsumen [2]. Selain itu optimasi rute tol laut ini juga berfungsi sebagai usaha penghematan bahan bakar sehingga dapat menunjang usaha pemerintah untuk meminimalkan penggunakan bahan bakar minyak (BBM).

Penggunaan tol laut ini sudah dilakukan selama 4 tahun dan telah dirasakan manfaatnya khususnya untuk Indonesia Timur dengan adanya penurunan harga barang, akan tetapi yang seringkali juga dikeluhkan adalah kembalinya kapal tol laut dari Indonesia Timur yang tidak cukup membawa barang angkutan. Hal ini juga perlu dievaluasi sehingga optimasi biaya operasional dapat diterapkan dan segera dirasakan manfaatnya untuk rakyat indonesia. Dari berbagai uraian diatas, telah diketahui permasalahan yang hadir dalam penggunaan tol laut selama empat tahu ini. Oleh karena itu analisa perbaikan rute tol dirasa perlu dilakukan guna meminimalkan biaya transportasi serta memberikan alternatif rute agar dapat digunakan pada tahun mendatang

ISI

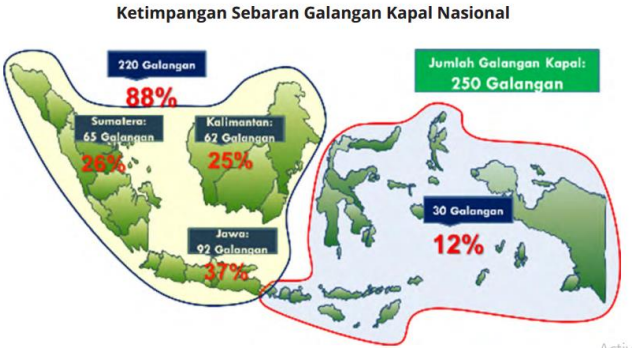
Tol Laut

Tol laut yang diperkenalkan oleh Presiden Joko Widodo adalah sebuah konsep dalam memperkuat jalur pelayaran yang di titik beratkan pada Indonesia bagian timur. Konsep dalam menyambungkan antara wilayah barat dan timur juga diharapkan akan mempermudah akses negara-negara pasifik bagian selatan ke negara asia bagian timur. Ide terhadap tol laut akan membuka akses regional, oleh sebab itu Indonesia menjadi memiliki peran yang signifikan dalam mendukung distribusi logistik secara Internasional. Terbukanya akses regional melalui implementasi konsep tol laut dapat memberikan peluang industri kargo/logistik nasional untuk berperan dalam distribusi internasional, dimana saat ini 40% melalui wilayah Indonesia [5], [6][7].

Beberapa elemen utama yang diperlukan dalam mendukung

konsep tol laut adalah sebagai berikut :

1. Pelabuhan yang handal
2. Kecukupan muatan dari barat-timur
3. Pelayaran rutin dan berjadwal
4. Shipping Industri
5. Rute Akses yang efektif



Gambar 2. Ketimpangan Sebaran Galangan Kapal Nasional

Dari aspek logistik untuk angkutan laut, terdapat permasalahan tidak efisiennya pengangkutan barang yang diangkut terutama untuk angkutan laut ke Indonesia bagian timur. Pada saat ini angkutan laut dari Pulau Jawa ke Papua terisi penuh, namun kembali dalam keadaan kosong. Salah satu penyebabnya adalah karena wilayah di timur Indonesia masih memiliki konektivitas yang rendah. Hal ini menyebabkan biaya logistik yang dibebankan kepada komoditi menjadi tinggi, sehingga diperlukan keberpihakan dalam penyelenggaraan layanan angkutan laut dari Barat ke Timur[8][9].

Untuk merealisasikan rute/jaringan pelayaran tersebut, diperlukan kebijakan strategis yaitu[9], [10]:

1. Penataan jaringan trayek angkutan laut (revisi SK Trayek).
2. Perluasan jaringan trayek, peningkatan frekuensi layanan, serta peningkatan
   keandalan kapal untuk angkutan laut dan keperintisan.

3. Optimalisasi penyelenggaraan PSO angkutan laut penumpang maupun barang, mengingat jumlah muatan barang dari wilayah Indonesia Timur yang masih rendah.

Vehicle Routing Problem

Permasalahan *Vehicle Routing Problem* (VRP) berhubungan dengan distribusi produk antara depot dengan konsumen. *Capacitated Vehicle Routing Problem* (CVRP) merupakan bentuk paling dasar dari VRP yaitu optimasi untuk menemukan rute dengan biaya yang minimal untuk sejumlah kendaraan dengan kapasitas tertentu yang homogen untuk melayani permintaan sejumlah pelanggan yang kuantitas permintaannya telah diketahui sebelum proses pengiriman berlangsung [2] [3] [4]. Terdapat beberapa karakteristik dalam VRP seperti pelanggan, depot, kendaraan, pengemudi, dan rute yang ditempuh. Masing-masing karakteristik memiliki komponen yang berkaitan dalam VRP seperti dibawah ini [11]:

1. Pelanggan
   a. Lokasi pelanggan
   b. Permintaan pelanggan
   c. Rentang waktu dalam menangani pelanggan
   d. Loading unloading time
   e. Jenis kendaraan
2. Depot
   a. Lokasi depot
   b. Jam operasional
   c. Kendaraan yang tersedia
3. Kendaraan
   a. Jumlah dan kapasitas tiap kendaraan
   b. Setiap rute hanya dilayani oleh satu kendaraan
   c. Total permintaan pelanggan
   d. Biaya yang terkait dengan kendaraan
4. Pengemudi

   Pekerjaan pengemudi harus sesuai dengan syarat yang berlaku seperti jam kerja, jam lembur, serta jam istirahat.
5. Rute Perjalanan

Rute perjalanan sebaiknya menggambarkan kondisi asli seperti kelayakan jalan agar mengurangi resiko barang dan kendaraan selama proses pengiriman berlangsung.

Vehicle Routing Problem (VRP) adalah masalah dalam distribusi yang memiliki tujuan untuk menciptakan rute yang optimal terutama untuk kendaraan yang telah diketahui kapasitasnya untuk memenuhi harapan konsumen di lokasi tertentu dan permintaan tertentu [9]. VRP juga dapat didefinisikan sebagai masalah untuk merancang rute pengiriman yang optimal dari beberapa depo ke beberapa pemasok dengan memenuhi batasan-batasan tertentu [10]. Kedua definisi tersebut didukung oleh [6] bahwa VRP adalah masalah yang berfokus pada distribusi logistik dari depo ke konsumen.

Solusi VRP adalah beberapa rute untuk mengirimkan logistik dengan kendaraan tertentu. Setiap rute ditempuh oleh satu kendaraan dan kendaraan harus kembali ke pelabuhan awal. Ada beberapa tujuan umum VRP seperti meminimalkan biaya transportasi, meminimalkan kendaraan, menyeimbangkan rute dan meminimalkan penalti karena layanan pelanggan yang tidak menyenangkan.

Karakteristik VRP dapat dilihat sebagai berikut [10] [11]:

1. Rute perjalanan kendaraan harus dimulai dan selesai di depot awal
2. Ada beberapa tempat yang harus dipenuhi satu kali
3. Jika kapasitas kendaraan telah digunakan dan tidak dapat melayani konsumen berikutnya, kendaraan harus kembali ke depo utama dan memenuhi kapasitas untuk melayani konsumen berikutnya
4. Tujuan utama dari masalah ini adalah untuk meminimalkan jarak total kendaraan dengan mengatur urutan konsumen dan ketika kendaraan kembali ke depot utama untuk memenuhi kapasitas

Berbagai metode dalam pengoptimalan rute transportasi telah

banyak dikembangkan dan yang paling populer dengan metode heuristik klasik dalam penyelesaian VRP. Heuristik merupakan sebuah pendekatan yang digunakan untuk menyelesaikan sebuah permasalahan untuk mendapatkan solusi yang optimal. Tahapan penentuan rute pada penelitian ini adalah dengan *tour construction* dan *tour improvement*. Pada tahap *tour construction*, penentuan solusi rute awal akan dibuat dan selanjutnya akan disempurnakan pada tahap *tour improvement* [11]

Clark And Wright Saving Matriks

Clark and Wright Savings merupakan metode yang tepat digunakan dalam tahap *tour construction* dengan prinsip penghematan penggabungan dua rute menjadi satu. Rute dengan penghematan terbesar akan memiliki kesempatan untuk dieksekusi [12] [13], [14]. Pada langkah ini solusi awal rute tol laut akan dibuat. Metode yang digunakan untuk membuat rute awal adalah metode konstruksi terutama *Clarke dan Wright Saving Algorithm*. Algoritma ini dikategorikan sebagai metode konstruksi karena berfungsi dengan secara bertahap memasukkan konsumen ke dalam rute. Penghematan berarti beberapa jarak dapat dikurangi dengan menggabungkan dua konsumen menjadi satu [3]. Dua rute yang memiliki penghematan terbesar yang memiliki peluang untuk dibuat sebagai rute awal [6] [12] [13].

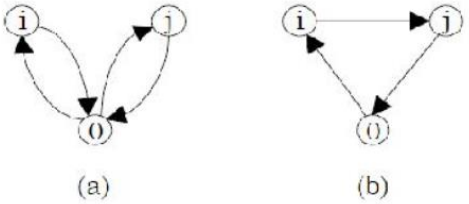
Algoritma penghematan matriks dikategorikan sebagai algoritma heuristik yang tidak diberi solusi optimal dalam masalah tertentu. Konsep ilustrasi matriks tabungan dapat digambarkan pada diagram di bawah ini [9]. Berikut merupakan langkah dalam melakukan saving matriks [1]:

1. Mengidentifikasi matrik jarak. Langkah awal metode ini adalah mencatat jarak antara gudang ke masing - masing lokasi pelanggan dan jarak antar lokasi. Dengan mengetahui koordinat dari masing – masing lokasi, maka jarak antar dua lokasi dapat dihitung dengan menggunakan rumus standar. Misalkan dua lokasi diketahui koorinat (x1,y1) dan (x2,y2),

maka dapat dihitung jarak antara dua lokasi tersebut dengan rumus:

$$J(1,2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2. Selanjutnya mengidentifikasi matriks penghematan. Pada awal Langkah ini diasumsikan bahwa setiap toko akan dikunjungi oleh satu truk secara eksklusif. Maka akan ada penghematan yang akan diperoleh jika dua atau lebih rute bila digabungkan menjadi satu rute. *Savings matrix* merepresentasikan penghematan yang bisa direalisasikan dengan menggabungkan dua toko maupun pelanggan ke dalam satu rute. Apabila masing-masing toko i dan toko j dikunjungi secara terpisah maka jarak yang dilalui adalah jarak dari gudang ke toko i dan dari toko i balik ke gudang. ditambah dengan jarak dari gudang ke toko j dan kemudian balik ke gudang o.



Gambar 3. *Saving Matrix Illustration*

I, j = consumer i, consumer j

O = depot

$$Da = Coi + Cio + Coj + Cja \quad (1)$$

$$Db = Coi + Cij + Cjo \quad (2)$$

$$Sij = Da - Db = (Coi + Cio + Coj + Cja) - Coi + Cij = Cio + Coj - Cij \quad (3)$$

Da = cost or total distance of two route (consumer route i plus consumer route j)
Db = cost or total one route (consumer route i and j)

Coi = cost or distance from depot to consumer i
Cio = cost or distance from i to depot
Coj = cost ot distance from depot to consumer j
Cjo = cost or distance from consumer j to depot
Cij = cost or distance from consumer i to consumer j
Sij = the saving cost or distance between consumer i and j

Parallel Insertion Dan Sequential Search

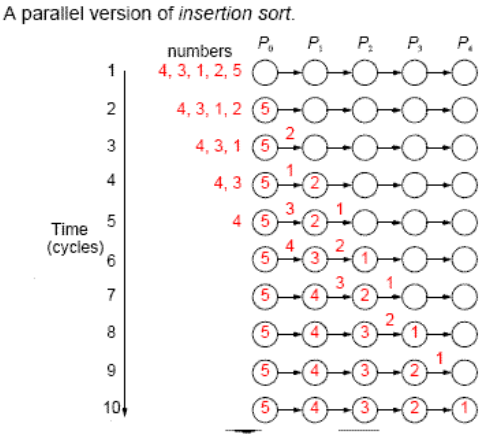
Setelah dilakukan matriks penghematan maka terbentuklah sebuah rute baru dimana dalam membuatnya dapat dilakukan dengan dua cara, yaitu *parallel insertion* dan *sequential version*. Pada *sequential version* pembentukan rute dilakukan secara bertahap dan nilai savings yang terdapat pada *saving list* akan dibaca serta perlanggan-pelanggan yang terkait dengan nilai tersebut dicoba dimasukkan dalam rute. Prioritas pembentukan rute kendaraan terkonsentrasi pada pembentukan satu rute terlebih dahulu berdasarkan nilai *savings* terbesar. Sedangkan pada *parallel version*, rute yang terbentuk bisa lebih dari satu pada setiap kali pembacaan nilai *savings* dan prioritas dalam metode ini terkonsentrasi pada nilai *savings* terbesar. *Parallel insertion* terbukti mengeluarkan hasil yang lebih baik dibandingkan dengan sequential version [11][14][15][16]sehingga selanjutnya penelitian ini akan di eksekusi menggunakan metode tersebut.

Pendekatan ini menghasilkan banyak rute secara paralel masing-masing pergi ke tujuan akhir. Dua strategi pelengkap dapat dipertimbangkan:

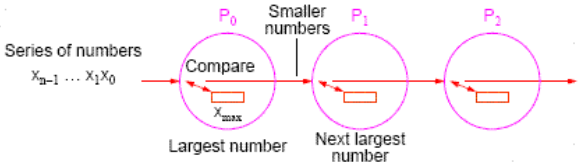
1. Diberikan busur, tentukan rute yang ada ke dalam yang harus disisipkan untuk meminimalkan jalan memutar yang terjadi.
2. Diberikan rute, tentukan busur yang tersisa harus disisipkan selanjutnya.

Prosedurnya adalah berpindah dari satu strategi ke strategi lain sampai rute dibuat. Prosedur seperti itu dirangkum dalam lampiran dan implementasi FeRTRAN. Urutan fase lainnya dapat diterapkan untuk menghasilkan prosedur lain. Contoh kasusnya adalah yang pertama, batas bawah pada jumlah rute yang dibutuhkan adalah

diperoleh dengan membagi jumlah siswa dengan kapasitas bus. Rute ini diinisialisasi menggunakan strategi 1 (langkah 1). Kemudian strategi 1 digunakan untuk mengisi rute (langkah 2). Prosedur pertukaran seperti yang digunakan [12] .



Gambar 4. *Parallel Insertion Illustration*



Gambar 5. *Parallel Insertion Illustration*

Penerapan skema ini pada banyak masalah menunjukkan bahwa langkah 1 menghasilkan himpunan minimal rute, mengambil sekitar 60% hingga 70% dari siswa. Hal ini juga menentukan 'benih' bagi yang lain rute. Langkah 2 kemudian menyeimbangkan rute dengan mengisi rute awal yang paling kosong. Di langkah ini, sekitar 85%

hingga 90% siswa dijemput terlebih dulu. Kemudian rute baru ditambahkan untuk mengambil sisanya (biasanya jumlah rute meningkat 10% ke 20%). Langkah terakhir dari algoritma biasanya menghapus satu atau dua rute 'terburuk', dengan mengurangi jarak total yang ditempuh sekitar 5%.

Frederickson dkk. [5] telah menggunakan hal yang serupa dengan pendekatan *parallel insertion* untuk masalah *routing node*, tapi mereka tidak menerapkannya pada masalah perutean busur. Sebaliknya mereka mengusulkan teknik untuk memisahkan file tur awal yang terdiri dari semua area di jaringan karena perilaku kasus terburuknya adalah perbaikan pada metode *insertion*. Dalam konteks kami, ini pendekatan tidak menghasilkan rute yang sesuai sejak itu ini tidak berorientasi pada sekolah.

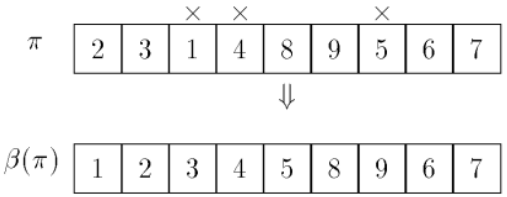
Parallel Insertion Model

Pada *parallel insertion* model, data yang disortir disajikan dengan sebuah permutasi π dari (1,2,...,n) dimana n adalah jumlah dari *processor*. Nilai data π [i] disimpan pada *processor* i; π dipertimbangkan untuk disortir jika π [i] = i untuk 1 ≤ i ≤ n. Untuk membuat simpel definisinya, kita asumsikan bahwa ada *dummy processors* 0 dan n + 1 dengan π [0] = -∞ dan π [n+1] = ∞. Langkah *insertion* kiri β didefinisikan oleh sebuah set dari *processors* yang ditandai Mβ ⊂ {0,1,2,...,n} dimana 0 adalah selalu sebuah anggota dari Mβ. Permutasi β(π) didefinisikan untuk menjadi pemutasi yang hasilnya dari perpindahan nilai data π[i+1], ... , π [j01] ke *processors* i + 2, i+3, ..., j, masing-masing, dan nilai π[j] ke *processors* i + 1 untuk setiap pasang i, j dari *procesor* berturut turut. Untuk lebih tepatnya Mβ = {i0,i1,i2,...,ik} dengan 0 = i0 < i1 < i2 < ...< Ik. Hasil dari *insertion* kiri β diaplikasikan ke π adalah permutasu β(π), dimana

$$\beta(\pi)[i] = \begin{Bmatrix} \pi[i_j]\ if\ i - i_{j-1} + 1\ and\ 1 \le j \le k \\ \pi[i-1]\ if\ i - 1\ \text{№} \notin M\beta\ and\ i - 1 < ik \\ \pi[i]\ if\ i \notin M\beta\ and\ i > ik \end{Bmatrix}$$

Gambar di bawah ini mengilustrasikan langkah insertion kiri dengan processor yang ditandai 0,3,4, dan 7. Perhatikan bahwa, karena *prosessor* berurutan 3 dan 4 ditandai, nilai 4 tidak bergerak.

Di sebuah simetris, kita dapat menentukan langkah-langkah penyisipan yang benar. Untuk insertion kanan β, Mβ memiliki Mβ⊂ {1,2,...,n, n+1}, dimana n+1 adalah selalu sebuah angota dari Mβ. Sebuah strategi sorting untuk π adalah urutan dari langkah insertion β1, β2, ..., βT dimana π0 = π, untuk 1 ≤ i ≤ T, π1 = β1 (πi-1) dan πT disorting. Kita mengatakan bahwa sorting strategi β1, β2, ..., βT sorting π dalam langkah T [17].



Gambar 6. *Parallel Insertion Model*

Strategi pengurutan hanya kiri untuk π adalah strategi pengurutan untuk π yang seluruhnya terdiri dari langkah *insertion* kiri. Demikian pula, sebuah strategi *sorting* hanya-kanan menggunakan langkah-langkah *insertion* kanan. Sebuah strategi *sorting* satu arah adalah salah satu yang hanya kiri atau strategi *sorting* hanya-kanan. Strategi *sorting* yang lebih umum adalah strategi pengurutan dua arah, yang memungkinkan keduanya kiri dan kanan langkah *insertion*. Kasus khusus dari strategi *sorting* dua arah adalah strategi *sorting* bergantian, di mana langkah *insertion* kiri bergantian dengan langkah *insertion* kanan, dimulai dengan langkah *insertion* kiri dan diakhiri dengan langkah *insertion* kanan.

```
LEFTINSERTIONSTEP(pi, marked)
  data ← ∞
  if marked then data ← broadcast_left(pi)
  left_marked ← true
  left_marked ← shift_right(marked)
  left ← shift_right(pi)
  if data ≠ ∞ then
     if not left_marked then pi ← left
     if not marked and left_marked then pi ← data
LEFTGREEDYSTEP(pi)
  right_min ← SUFFIXMINIMUM(pi)
  marked ← right_min ≥ pi
  LEFTINSERTIONSTEP(pi, marked)
```

Gambar 7. Algoritma Pemrograman *Parallel Insertion*

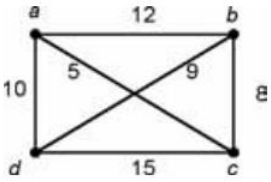
Seperti yang ditunjukkan pada kode subbus untuk LEFTINSERTIONSTEP pada gambar diatas, pergerakan data dalam langkah *parallel insertion* bisa diimplementasikan dalam jumlah operasi subbus yang konstan. Diberikan data dalam nilai jamak π dan nilai Boolean jamak ditandai yang menunjukkan apakah prosesor ditandai atau tidak, sebuah broadcast_left mengirimkan data yang ditandai ke kiri untuk disisipkan, dan dua operasi shift_right menggeser yang sesuai data di sebelah kanan. Hal penting yang perlu diperhatikan tentang model *parallel insertion* adalah bahwa langkah penyisipan ditentukan sepenuhnya oleh prosesor yang ditandai dan arah langkah penyisipan. Model tidak membatasi cara penandaan set dan arah langkah penyisipan ditentukan, sehingga hanya ukuran kompleksitas yang kami pertimbangkan dalam model ini adalah jumlah langkah penyisipan yang digunakan oleh strategi penyortiran — pada dasarnya, kompleksitas komunikasi dari strategi tersebut. Untuk menilai sepenuhnya efisiensi strategi penyortiran, kami menyediakan kode subbus untuk setiap strategi kami dan pertimbangkan jumlah operasi subbus yang dibutuhkan oleh masing-masing.

Exhaustive Search

Setelah mendapatkan rute sebelumnya dari konstruksi tur, langkah selanjutnya adalah mengoptimalkan rute dengan peningkatan wisata. Perbaikan dilakukan dengan mengganti node yang tetap pada solusi (perbaikan rute tunggal). Lokasi simpul sekuensial bergerak menurut aturan tertentu. Selama perpindahan jika ada solusi terbaik sehingga solusi sebelumnya akan digantikan oleh solusi baru. Untuk mempermudah dilakukan langkah-langkah sebagai berikut [18] [19]:

1. Membuat daftar keseluruhan sirkuit Hamilton dari graf yang lengkap dengan jumlah n buah simpul
2. Mengevaluasi skor dari setiap sirkuit Hamilton yang ditemukan pada langkah pertama
3. Memilih sirkuit Hamilton dengan skor paling kecil

Contoh n = 4



Gambar 8. Hamilton Sirkuit

Hasilnya adalah sebagai berikut:

Rute perjalanan (tour)	Bobot
aÆbÆcÆdÆa	12+8+15+10 = 45
aÆbÆdÆcÆa	12+9+15+5 = 41
aÆcÆbÆdÆa	5+8+9+10 = 32
aÆcÆdÆbÆa	5+15+9+12 = 41
aÆdÆbÆcÆa	10+9+8+5 = 32
aÆdÆcÆbÆa	10+15+8+12 = 45

Gambar 9. Contoh Proses Enumerasi

Apabila permasalahan diselesaikan menggunakan metode exhaustive search, maka jumlah enumerasi yang dilakukan adalah sebanyak (n-1) ! buah sirkuit hamilton, kemudian melakukan ketiga langkah seperti diatas. Jumlah kompurtasi dapat dihemat dengan mengamati bahwa setengah dari rute perjalanan merupakan pencerminan dari setengah rute lainnya, yakni dengan cara mengubah arah dari rute perjalanan tersebut.

Contoh :

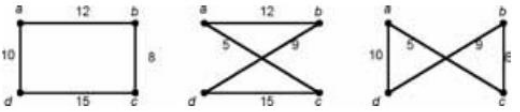
1 dan 6

2 dan 4

3 dan 5

Maka dapat dihilangkan setengahnya dari jumlah permutasi dari 6

menjadi 3. Ketiga sirkuit hamilton yang didapatkan adalah sebagai berikut :



Gambar 10. Hamilton Sirkuit *Illustration*

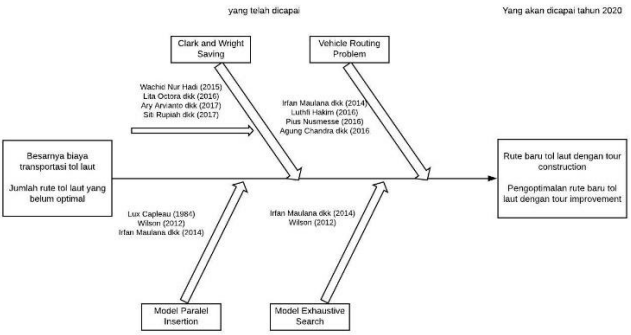
Pencarian lengkap adalah pendekatan untuk pemecahan masalah dengan enumerasi brute-force [15]. Metode ini sederhana berdasarkan pada argumen matematis. Pencarian lengkap memeriksa setiap titik pencarian di dalam wilayah pencarian dan memberikan kemungkinan kecocokan. Prinsip dasar dari metode ini adalah membagi proses pencarian menjadi langkah sekuensial dan memilih arah pencarian langkah selanjutnya berdasarkan hasil langkah saat ini [16]. Namun, metode ini dianggap tidak elok, membosankan, rawan kesalahan, dan melelahkan. Ketika opsi terlalu banyak maka metode ini tidak direkomendasikan untuk diterapkan [6].

Metode *exhaustive search* merupakan teknik pencarian elemen dengan sifat yang khusus diantara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan [20], [21]. Algoritma *exhaustive search* memiliki keunggulan dengan pemeriksaan secara sistematis pada setiap kemungkinan solusi. Tahapan dari metode *exhaustive search* adalah enumerasi setiap solusi yang mungkin dengan cara yang sistematis, evaluasi setiap kemungkinan solusi satu per satu, mengeluarkan beberapa solusi yang tidak layak dan menyimpan solusi yang terbaik, dan pencarian terakhir merupakan solusi terbaik[11], [20]. Akan tetapi metode ini kurang efektif apabila dilakukan pada rangkaian rute yang sangat banyak dikarenakan hasil permutasi dan kombinasi yang besar akan menghabiskan waktu yang dalam penyusunan rute akan sangat lama.

Metode Pemecahan Permasalahan

Penggunakan metode secara kuantitatif masih dirasa perlu dilakukan dalam penghematan rute. Pada penelitian kali ini, secara umum terdapat dua tahapan dalam menentukan rute yang optimal yaitu dengan *tour construction* khususnya dengan metode *Clark and Wright Savings* serta *paralell insertion*. Tahapan kedua yang dilakukan adalah dengan *tour improvement* yang bertujuan untuk memperbaiki hasil rute yang telah ada sehingga didapatkan hasil yang lebih baik. Pada *tour improvement* metode yang dipakai adalah *exhaustive search [5]*.

Data diperoleh dari basis data kementerian kelautan yang telah dibagikan ke dalam dokumen. Ada 22 port yang menjadi konsumsi dan ada port yang menjadi depo utama. Depot utama terletak di Belawan bagian barat Indonesia. Perangkat lunak utama yang digunakan dalam penelitian ini adalah Ms. Excel 2013 terutama pada Excel Solver untuk menghitung matriks penghematan dan membuat rute yang optimal. Netpas Distance juga telah digunakan untuk menentukan jarak antara depo ke konsumen dan konsumen ke konsumen. Metrik jarak diilustrasikan dalam mil Nautical (Nm).



Gambar 11. *Framework*

Pembahasan Permasalahan

Matriks jarak dan demand adalah data yang dikumpulkan sebagai

pedoman dalam pembuatan rute tol laut dimana optimasi rute akan dilakukan menggunakan jarak terpendek dari depot menuju ke tiap pelabuhan. Penentuan jarak dilakukan dengan penarikan titik dari satu pelabuhan ke pelabuhan lain menggunakan software Netpass Distance dengan satuan Nm (Nautical Mile), sedangkan penentuan titik pelabuhan menyesuaikan pelabuhan yang telah mampu beroperasi sesuai dengan buku pedoman kemaritiman tol laut dari Bappenas. Jumlah kapasitas masing-masing pelabuhan didapatkan dari data heuristik kementrian kemaritiman. Berikut merupakan tabel jarak masing-masing pelabuhan.

Tabel 1. Destination Matrix

		DESTINATION (Nm – Nautical Mile)									
		1	2	3	4	5	6	7	8	9	10
ORIGIN		Belawan	Tj. Priok	Tj. Perak	Makassar	Sorong	Batam	Malahayati	Jambi	Telukbayur	Palembang	Panjang
Belawan	0	0	1021	1119	1427	2243	372	251	630	784	673	1102
Tj. Priok	1	1021	0	358	750	1492	468	977	334	495	225	100
Tj. Perak	2	1119	358	0	431	1226	725	1325	640	1005	540	455
Makassar	3	1427	750	431	0	754	999	1569	967	1167	887	842
Sorong	4	2243	1492	1226	754	0	1644	2176	1656	849	1595	1578
Batam	5	372	468	725	999	1644	0	573	155	245	246	398
Malahayati	6	251	977	1325	1569	2176	573	0	649	490	756	886
Jambi	7	630	334	640	967	1656	155	649	0	200	108	255
Telukbayur	8	784	495	1005	1167	849	245	490	200	0	289	400
Palembang	9	673	225	540	887	1595	246	756	108	289	0	153
Panjang	10	1102	100	455	842	1578	398	886	255	400	153	0
Tj. Emas	11	1033	217	195	547	1298	617	1168	518	700	32	317
Pontianak	12	696	393	503	670	1316	330	893	352	541	326	406
Sampit	13	1104	422	281	416	1101	583	1154	561	761	491	490
Banjarmasin	14	1151	488	257	309	1011	689	1260	662	862	586	567
Balikpapan	15	1421	660	461	279	867	779	1340	789	986	728	732
Samarinda	16	1481	700	510	308	846	792	1347	812	1008	757	769
Pantoloan	17	1555	841	578	266	683	955	1507	973	1169	915	916
Kendari	18	1793	949	636	202	552	1152	1720	1143	1343	1068	1036
Tenau Kupang	19	1806	1018	665	390	724	1347	1924	1293	1488	1198	1119
Ternate	20	2048	1295	1107	595	252	1400	1931	1431	1625	1376	1379
Ambon	21	1994	1281	960	531	250	1476	2035	1475	1675	1403	1371
Jayapura	22	2819	2036	1696	1282	555	2190	2732	2203	2422	2134	2106

Tabel 1. Destination Matrix (Cont.)

	DESTINATION (Nm – Nautical Mile)												
	11	12	13	14	15	16	17	18	19	20	21	22	
ORIGIN	Tj. Emas	Pontianak	Sampit	Banjarmasin	Balikpapan	Samarinda	Pantoloan	Kendari	Tenau Kupang	Ternate	Ambon	Jayapura	
Belawan	0	1033	696	1104	1151	1421	1481	1555	1793	1806	2048	1994	2819
Tj. Priok	1	217	393	422	488	660	700	841	949	1018	1295	1281	2036
Tj. Perak	2	195	503	281	257	461	510	578	636	665	1107	960	1696
Makassar	3	547	670	416	309	279	308	266	202	390	595	531	1282
Sorong	4	1298	1316	1101	1011	867	846	683	552	724	252	250	555
Batam	5	617	330	583	689	779	792	955	1152	1347	1400	1476	2190
Malahayati	6	1168	893	1154	1260	1340	1347	1507	1720	1924	1931	2035	2732
Jambi	7	518	352	561	662	789	812	973	1143	1293	1431	1475	2203
Telukbayur	8	700	541	761	862	986	1008	1169	1343	1488	1625	1675	2422
Palembang	9	32	326	491	586	728	757	915	1068	1198	1376	1403	2134
Panjang	10	317	406	490	567	732	769	916	1036	1119	1376	1371	2106
Tj. Emas	11	0	421	305	329	511	558	677	748	801	1118	1078	1811
Pontianak	12	421	0	265	371	454	470	633	829	1044	1085	1151	1887
Sampit	13	305	265	0	106	242	279	427	582	779	888	914	1643
Banjarmasin	14	329	371	106	0	182	230	354	482	674	809	816	1548
Balikpapan	15	511	454	242	182	0	50	186	382	669	647	697	1415
Samarinda	16	558	470	279	230	50	0	162	386	695	619	688	1398
Pantoloan	17	677	633	427	354	186	162	0	255	610	461	530	1235
Kendari	18	748	829	582	482	382	386	255	0	377	407	335	1070
Tenau Kupang	19	801	1044	779	674	669	695	610	377	0	699	478	1106
Ternate	20	1118	1085	888	809	647	619	461	407	699	0	274	803
Ambon	21	1078	1151	914	816	697	688	530	335	478	274	0	734
Jayapura	22	1811	1887	1643	1548	1415	1398	1235	1070	1106	803	734	0

Permintaan setiap lokasi atau konsumen juga perlu diuraikan sebagai salah satu kendala dalam penelitian ini. Permintaan akan diilustrasikan dalam TEUS (Unit Setara Dua Puluh kaki) yang biasanya digunakan untuk metrik kontainer atau logistik. Data permintaan telah dikumpulkan dan dapat dilihat di bawah:

Table 2. Demand of each location/consumer

Location/ Consumer	Demand (TEUS)	Location/ Consumer	Demand (TEUS)
Tj. Priok	132510	Pontianak	159303
Tj. Perak	225514	Sampit	159303
Makassar	67547	Banjarmasin	229438
Sorong	47850	Balikpapan	159303
Batam	253369	Samarinda	159303

Malahayati	159303	Pantoloan	159303
Jambi	159303	Kendari	159303
Telukbayur	159303	Tenau Kupang	159303

Table 2. Demand of each location/consumer (lanjutan)

Location/ Consumer	Demand (TEUS)	Location/ Consumer	Demand (TEUS)
Palembang	159303	Ternate	159303
Panjang	159303	Ambon	159303
Tj. Emas	159303	Jayapura	159303

Setelah semua data didapatkan, langkah selanjutnya yang dilakukan adalah membuat saving list. Saving list merupakan hasil optimasi jarak menggunakan metode *Clark and Wreight Saving Matriks* dimana menerapkan algoritma yang telah ada, dan selanjutnya digunakan sebagai acuan dalam pembuatan rute dengan jarak terkecil sebagai prioritas utama. Saving list dapat dihitung dengan menggunakan rumus seperti dibawah dan tabel 3 merupakan hasil dari saving list ke-22 lokasi Pelabuhan.

$$Sij = Cio + Coj - Cij$$

$$S\,sorong, Jayapura\ (S\,4{,}22) = C4, o + Co, 22 - C4{,}22$$

$$= 2243 + 2819 - 555$$

$$= 4507\ Nm$$

Dimana,

S = Saving dan C = Consumer

Table 3. Sample of Saving List

Consumer i	Consumer j	Saving (Nm)
Sorong	Jayapura	4507
Ambon	Jayapura	4079
Ternate	Jayapura	4064
Sorong	Ternate	4039
Sorong	Ambon	3987
Ternate	Ambon	3768
Kendari	Jayapura	3542
Tenau Kupang	Jayapura	3519
Sorong	Kendari	3484
Kendari	Ambon	3452
Kendari	Ternate	3434
Sorong	Tenau Kupang	3325
Tenau Kupang	Ambon	3322
Kendari	Tenau Kupang	3222

Setelah mendapatkan hasil saving list, rute awal dibuat dengan menggunakan metode *parallel insertion* yang memilih daftar simpanan tertinggi yang akan dieksekusi terlebih dahulu. Parallel insertion lebih dipertimbangkan daripada *sequential insertion* dalam penelitian ini karena pada penelitian sebelumnya *parallel insertion* telah mampu dibuktikan lebih baik dalam pembuatan rute dengan jarak lebih pendek dibandingkan dengan *sequential insertion*. Rute baru telah dipecahkan oleh Ms.Excel dan dapat diuraikan di bawah ini:

Table 4. New Sea Highway Route

Number of Route	Sequence Route
First Route	Depot – Jambi – Panjang – Tj.Priok – Tj. Emas – Palembang - Depot
Second Route	Depot – Pontianak – Samarinda – Balikpapan – Sampit – Depot
Third Route	Depot – Banjarmasin – Pentoloan – Kendari – Tenau Kupang – Tj. Perak - Depot
Fourth route	Depot – Malahayati – Batam - Telukbayur – Depot
Fifth route	Depot – Makassar – Ambon – Jayapura – Sorong – Ternate - Depot

Kelima rute diatas masih dirasa belum optimal, sehingga diperlukan peningkatan rute agar mendapatkan jarak yang lebih pendek. Tahap ini selanjutnya disebut dengan *tour improvement* dimana proses yang dilakukan adalah menggunakan algoritma *exhaustive search*. Langkah yang dilakukan adalah sebagai berikut :

1. Pertama daftar enumerasi harus dicatat secara sistematis
2. dievaluasi setiap kemungkinan solusi satu per satu
3. ketika semua daftar enumerasi telah diproses kemudian umumkan skor terbaik dimana rute lebih pendek yang akan dipilih.

Daftar enumerasi dibuat dengan menggunakan Ms. Excel sesuai dengan jumlah yang telah dihitung pada tabel 5, sedangkan dalam menentukan jumlah enumerasi dapat dibuat dengan menggunakan rumus di bawah ini:

$$Number\ of\ enumeration = (n-1)!$$

Table 5. Enumeration

Number of Route	Calculation	Result
First Route	(6-1) ! = 5!	120
Second Route	(5-1) ! = 4!	24
Third Route	(6-1) ! = 5!	120
Fourth route	(4-1) ! = 3!	6
Fifth route	(6-1) ! = 5!	120

Dari kelima rute tersebut akan dilakukan perhitungan pada setiap kemungkinan-kemungkinan dari rute untuk dihitung ulang score pada setiap kemungkinan rute sehingga memungkinkan adanya perbaikan rute. Pada tabel 5 diketahui bahwa rute pertama, ketiga, dan kelima akan terjadi 120 enumerasi, rute kedua 24 enumerasi, dan rute keempat 6 enumerasi. Hal ini berarti pada rute pertama, ketiga, dan kelima akan dilakukan 120 kali trial, pada rute kedua akan dilakukan 24 kali trial, dan pada rute keempat akan dilakukan 6 kali trial. Pada setiap trial skor akan dicatat dan dipilih yang paling rendah untuk ditetapkan sebagai rute pengganti dari rute hasil *tour construction*.

Table 6. Enumeration fourth route

Tour route	Score
Depot – Malahayati – Telukbayur – Batam – Depot	251 + 490 + 245 + 372 = 1358
Depot – Malahayati – Batam- Telukbayur – Depot	251 + 573 + 245 + 784 = 1853

Depot –Telukbayur - Malahayati – Batam – Depot	784 + 490 + 573 + 372 = 2219
Depot – Telukbayur – Batam – Malahayati –Depot	784 + 245 + 573 + 251 = 1853
Depot – Batam – Malahayati – Telukbayur – Depot	372 + 573 + 490 + 784 = 2219
Depot – Batam – Telukbayur – Malahayati - Depot	372 + 245 + 490 + 251 = 1358

Table 7. Enumeration second route

Tour Route	Score
Depot – Pontianak – Samarinda - Balikpapan - Sampit – Depot	2562
Depot – Pontianak – Samarinda – Sampit – Balikpapan – Depot	3108
Depot – Pontianak – Sampit – Samarinda – Balikpapan – Depot	2711
Depot – Pontianak – Sampit – Balikpapan – Samarinda – Depot	2734
Depot – Pontianak – Balikpapan – Sampit – Samarinda – Depot	3152
Depot – Pontianak – Balikpapan – Samarinda – Sampit – Depot	2583

Depot – Samarinda – Pontianak – Balikpapan – Sampit – Depot	3751
Depot – Samarinda – Balikpapan – Pontianak – Sampit – Depot	3354
Depot – Samarinda – Balikpapan – Sampit – Pontianak – Depot	2734
Depot – Samarinda – Pontianak – Sampit – Balikpapan – Depot	3879

Table 7. Enumeration second route

Tour Route	**Score**
Depot – Samarinda – Sampit – Pontianak – Balikpapan – Depot	3900
Depot – Samarinda – Sampit – Balikpapan – Pontianak – Depot	3152
Depot – Balikpapan – Pontianak – Samarinda – Sampit – Depot	3728
Depot – Balikpapan – Samrinda – Pontianak – Sampit – Depot	3310
Depot – Balikpapan – Sampit – Samarinda – Pontianak – Depot	3108
Depot – Balikpapan – Sampit – Pontianak – Samarinda – Depot	3879

Depot – Balikpapan – Samarinda – Sampit – Pontianak – Depot	2711
Depot – Balikpapan – Pontianak – Sampit – Samarinda – Depot	3900
Depot – Sampit – Pontianak – Balikpapan – Samarinda – Depot	3354
Depot – Sampit – Balikpapan – Samarinda – Pontianak – Depot	2562
Depot – Sampit – Balikpapan – Pontianak – Samarinda – Depot	2966
Depot – Sampit – Balikpapan – Pontianak – Samarinda – Depot	3751
Depot – Sampit – Samarinda – Pontianak – Balikpapan – Depot	3728
Depot – Sampit –Samarinda – Balikpapan – Pontianak - Depot	2583

Setelah dilakukan tahap kedua yaitu *tour improvement* dengan menggunakan metode *exhaustive search* didapatkan hasil enumerasi rute keempat seperti pada tabel 6 dan dapat disimpulkan solusi terbaik dari rute tol laut dengan skor terendah 1358 adalah rute *Depot-Malahayati - Telukbayur - Batam - Depot* dan rute sebaliknya yaitu *Depot - Batam - Telukbayur - Malahayati – Depot*. Dimana solusi pertama dari hasil *tour construction* dengan perolehan skor sebanyak 1853 yang yaitu *Depot -Malahayati – Batam - Telukbayur – Depot* dan rute sebaliknya *Depot – Telukbayur – Batam – Malahayati – Depot* . Perbedaan hasil kedua rute ini mungkin tidak terlalu signifikan yaitu sebesar 495 Nm dari sebelum

dan sesudah menerapkan pencarian lengkap. Akan tetapi metode ini masih layak untuk dicoba untuk menemukan rute yang lebih pendek.

Proses enumerasi rute kedua menghasilkan beberapa alternatif rute lainnya seperti yang dapat dilihat pada tabel 7. Pada tabel 7 menunjukkan skor terendah adalah 2.562 dengan rute *Depot - Pontianak - Samarinda - Balikpapan - Sampit - Depot* dan rute sebaliknya *Depot - Sampit - Balikpapan - Samarinda - Pontianak - Pontianak - Depot*. Dimana rute ini masih sama dengan rute pada tahap *tour counstruction*. Namun, dengan menggunakan exhaustive search juga dapat dilihat kemungkinan rute lain yang dapat dipilih oleh peneliti. Sebagai contoh, pada tabel 7 terdapat skor dengan selisih skor yang kecil dengan rute awal yaitu sebesar 2583 dengan detail rute *Depot – Sampit – Samarinda – Balikpapan – Pontianak – Depot* dan rute sebaliknya *Depot – Pontianak – Balikpapan – Samarinda – Sampit - Depot* yang hanya memiliki 18 skor yang berbeda sehingga dapat dipilih sebagai alternatif rute. Dengan adanya banyak kemungkinan yang muncul pada metode *exhaustive search,* pemilihan alternatif rute dengan selisih skor kecil juga dapat ditambahkan sebagai pertimbangan lain. Selain mempertimbangkan rute terpendek, alternatif rute juga dapat digunakan sebagai rute cadangan apabila kondisi cuaca laut tidak baik, jadwal setiap pelabuhan yang berganti dan sebagainya. Namun untuk beberapa kemungkinan permutasi seperti rute pertama, kedua, dan kelima, perhitungan *exhaustive search* akan menghabiskan banyak waktu untuk menghitung setiap enumerasi. Dengan begini, membuktikan bahwa *exhaustive search* tidak direkomendasikan untuk penelitian tingkat tinggi yang melibatkan banyak kemungkinan dan variabel.

PENUTUP

Tour Construction menghasilkan lima opsi rute jalan raya laut dengan menggunakan matriks hemat clark dan wright dan instruksi paralel yaitu rute pertama *Depot – Jambi – Panjang – Tj.Priok – Tj. Emas – Palembang - Depot* , rute kedua *Depot – Pontianak – Samarinda – Balikpapan – Sampit – Depot*, rute ketiga *Depot – Banjarmasin – Pentoloan – Kendari – Tenau Kupang – Tj. Perak – Depot,* rute keempat *Depot – Malahayati – Batam - Telukbayur –*

Depot, dan rute kelima *Depot – Makassar – Ambon – Jayapura – Sorong – Ternate – Depot*.

Kelima rute tol laut masih dapat dilakukan perbaikan dengan menggunakan metode exhaustive search sehingga beberapa rute awal diubah menggunakan metode tersebut. Setelah dilakukan penerapan metode exhaustive search diperoleh skor jarak yang lebih rendah sehingga terdapat rute yang dirubah yaitu rute keempat menjadi *Depot-Malahayati - Telukbayur - Batam – Depot.* Terdapat pula selisih rute yang sangat kecil yaitu sebesar 21 Nm pada rute kedua sehingga dapat digunakan sebagai pertimbangan rute cadangan apabila terjadi sesuatu yang diluar kendali manusia seperti pengalihan rute karena kecelakaan laut, atau cuaca buruk. Hal ini membuktikan bahwa exhaustive search dapat meningkatkan rute awal dengan melakukan langkah enumerasi.

Meskipun demikian, pencarian lengkap tidak direkomendasikan untuk kemungkinan permutasi yang besar karena memakan waktu dalam menggunakannya. Dalam pencarian enumerasi harus dilakukan trial satu per satu pada setiap kemungkinan yang ada, padahal belum tentu pada setiap perlakuan akan menghasilkan skor jarak yang lebih rendah. Sebaiknya dibentuk sebuah software atau solver sehingga mempercepat penemuan skor terendah dalam exhaustive search sehingga lebih efektif dalam menggunakannya.

DAFTAR PUSTAKA

[1] F. Ahmad and H. F. Muharram, “Penentuan Jalur Distribusi Dengan Metode Saving Matriks,” *Competitive*, vol. 13, no. 1, p. 45, 2018.

[2] C. Roch and S. Langer, “The Capacitated Vehicle Routing Problem,” *Digit. Welt*, vol. 3, no. 2, pp. 30–33, 2019.

[3] R. Fitriana, P. Moengin, and U. Kusumaningrum, “Improvement Route for Distribution Solutions MDVRP (Multi Depot Vehicle Routing Problem) using Genetic Algorithm,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 528, no. 1, 2019.

[4] A. Sampaio, J. Kinable, L. P. Veelenturf, and T. Van Woensel, “A Scenario-Based Approach for the Vehicle Routing Problem with

Roaming Delivery Locations under Stochastic Travel Times," *Optim. Online*, 2019.

[5] D. D. Andilas and L. A. Yanggana, "Pelaksanaan Program Tol Laut Pt Pelayaran Nasional Indonesia," *J. Manaj. Transp. Dan Logistik*, vol. 4, no. 1, p. 1, 2017.

[6] Kementerian PPN / Bappenas, "Laporan Implementasi Konsep Tol Laut 2015 Direktorat Transportasi," p. 55, 2015.

[7] M. F. Pradana, "Selayang Pandang Tol Laut Indonesia," no. July 2018, 2018.

[8] lutvia nur vitasari, "Analisis Evaluasi Implementasi Kebijakan Tol Laut," *Its*, p. 106, 2017.

[9] E. R Gultom, "Merefungsi Pengangkutan Laut Indonesia Melalui Tol Laut Untuk Pembangunan Ekonomi Indonesia Timur," *Develop*, vol. 1, no. 2, 2017.

[10] B. Prihartono, "Pengembangan Tol Laut Dalam Rpjmn 2015-2019 Dan Implementasi 2015," p. 110, 2015.

[11] I. Maulana and R. Arifati, "PERENCANAAN RUTE PENGIRIMAN MENGGUNAKAN METODE PARALLEL INSERTION DAN EXHAUSTIVE SEARCH PADA PT . STARMASS LOGISTICS UPN " VETERAN " JAKARTA UPN " VETERAN " JAKARTA."

[12] L. Octora, A. Imran, and S. Susanty, "Pembentukan Rute Distribusi Menggunakan Algoritma Clarke & Wright Savings dan Algoritma Sequential Insertion," *Reka Integr.*, vol. 2, no. 2, pp. 1–11, 2014.

[13] S. Suprayogi, "Pemecahan Masalah Rute Kendaraan Dengan Trip Majemuk, Jendela Waktu Dan Pengantaran-Penjemputan Simultan Menggunakan Algoritma Genetika," *J@ti Undip J. Tek. Ind.*, vol. 12, no. 2, p. 95, 2017.

[14] M. W. P. Savelsbergh, "A parallel insertion heuristic for vehicle routing with side constraints," *Stat. Neerl.*, vol. 44, no. 3, pp. 139–148, 1990.

[15] L. Chapleau, J. A. Ferland, G. Lapalme, and J. M. Rousseau, “A parallel insert method for the capacitated arc routing problem,” *Oper. Res. Lett.*, vol. 3, no. 2, pp. 95–99, 1984.

[16] M. A. Clark-wright, P. Studi, T. Industri, F. Teknik, and U. Diponegoro, “Optimasi Rute Angkutan Publik dengan Menggunakan,” pp. 8–9, 2017.

[17] J. D. Fix and R. E. Ladner, “Sorting by parallel insertion on a one-dimensional subbus array,” *IEEE Trans. Comput.*, vol. 47, no. 11, pp. 1267–1281, 1998.

[18] Wulandari et al., “Algoritma Exhaustive Search sebagai Pencari Solusi Terbaik,” pp. 1–3, 2018.

[19] T. Draganov Stojanovski, “Performance of exhaustive search with parallel agents,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 22, no. 5, pp. 1382–1394, 2014.

[20] N. I. Fajariyah *et al.*, “Unnes Journal of Mathematics Education,” *Ujme*, vol. 1, no. 2, pp. 153–167, 2012.

[21] G. Wilson, S. Harding, O. Hoeber, R. Devillers, and W. Banzhaf, “Parallel exhaustive search vs. evolutionary computation in a large real world network search space,” *2012 IEEE Congr. Evol. Comput. CEC 2012*, 2012.