

---

**USULAN TUGAS AKHIR**

**1. IDENTITAS PENGUSUL**

Nama : **Timotius Ichan Hariadi**  
NRP : **5109 100 199**  
Dosen Wali : **Ir. Mohammad Husni, M.Kom**

**2. JUDUL TUGAS AKHIR**

**“Optimasi Penyelesaian Permasalahan *Workover Rig Routing Problem* (WRRP) Menggunakan Algoritma *Adaptive Large Neighborhood Search* (ALNS)”**

***“The Completion Optimization of Workover Rig Routing Problem (WRRP) Using An Adaptive large Neighborhood Search Algorithm (ALNS)”***

**3. ABSTRAKSI TUGAS AKHIR**

Saat ini operasi produksi minyak *onshore* membutuhkan biaya mahal. Ketika peralatan rusak maka diperlukan layanan perbaikan untuk mengembalikan operasi produksi sumur minyak *onshore* sebagaimana semestinya. Layanan ini dilakukan oleh sejumlah *workover rig* mahal berakomodasikan truk. Untuk di Indonesia sendiri biaya *workover rig* mencapai US\$. 20.00 - US\$. 25.00 per hari per HP, berdasarkan keputusan THO APMI. Dengan mahalnya biaya operasi *workover rig* maka dibutuhkan optimalisasi proses perbaikan sumur minyak dengan cara melakukan minimalisasi waktu perjalanan *workover rig* menuju sumur minyak. *Workover Rig Routing Problem* (WRRP) adalah varian dari permasalahan *Vehicle Routing Problem with Time Windows* (VRPTW). WRRP adalah permasalahan dimana sekumpulan truk *workover rig* terletak di beberapa lokasi yang berbeda dari lokasi sumur minyak yang membutuhkan perbaikan. Ketika proses perbaikan dilakukan, produksi minyak pada sumur minyak akan dihentikan dengan alasan keselamatan. Oleh karena itu, batasan waktu harus diperhatikan untuk meminimalkan kehilangan produksi minyak.

Dalam proposal tugas akhir ini, permasalahan WRRP diselesaikan algoritma *Adaptive Large Neighborhood Search* (ALNS) yang telah terbukti lebih efisien dari algoritma *Clustering Search* maupun *Iterated Local Search*. Hal ini terbukti dikarenakan *Adaptive Large Neighborhood Search* merupakan metode metaheuristik yang cukup handal untuk menyelesaikan permasalahan WRRP dengan kompleksitas *NP-Hard*.

## 4. PENDAHULUAN

Pendahuluan ini berisikan latar belakang, rumusan masalah, batasan masalah, serta tujuan dan manfaat dari proposal tugas akhir yang diajukan.

### 4.1 LATAR BELAKANG

Saat ini operasi produksi minyak *onshore* membutuhkan biaya mahal. Ketika peralatan rusak maka diperlukan layanan perbaikan untuk mengembalikan operasi produksi sumur minyak *onshore* sebagaimana semestinya. Layanan ini dilakukan oleh sejumlah *workover rig* mahal berakomodasikan truk. Untuk di Indonesia sendiri biaya *workover rig* mencapai US\$. 20.00 - US\$. 25.00 per hari per HP, berdasarkan keputusan THO APMI [1]. Ketika sumur minyak memerlukan perbaikan maka kegiatan operasi produksi minyak akan dihentikan untuk alasan keamanan. Selanjutnya *workover rig* harus dikirim ke sumur minyak yang membutuhkan perbaikan dalam rangka memulihkan lagi produksi minyak yang terhenti. Dalam operasi produksi minyak terdapat permasalahan yaitu satu set *workover rig* terletak pada posisi yang berbeda dengan sumur minyak dan membutuhkan waktu yang lama untuk menuju sumur minyak. Oleh karena itu diperlukan routing untuk *workover rig* menuju sumur minyak yang akan diperbaiki. Selanjutnya permasalahan ini disebut *Workover Rig Routing Problem* (WRRP).

WRRP adalah varian dari permasalahan *Vehicle Routing Problem with Time Windows* (VRPTW) [2]. Pada permasalahan klasik VRPTW bertujuan untuk meminimalkan panjang rute yang ditempuh sedangkan permasalahan WRRP bertujuan untuk menemukan routing terbaik ke sumur minyak sedemikian rupa sehingga semua permintaan dipenuhi, tenggat waktu dipertimbangkan dan hilangnya produksi minyak diminimalkan. WRRP disamakan juga dengan permasalahan *Delivery Man Problem with Time Windows* (DMPTW) [3]. DMPTW adalah permasalahan bagaimana sebuah *depot*, pusat distribusi barang, dengan sejumlah kendaraan berkapasitas tertentu melayani sejumlah customer pada titik-titik lokasi terpisah, dengan permintaan dan batasan *time window* tertentu, dengan tujuan meminimalkan total biaya perjalanan, tanpa mengabaikan batasan kapasitas kendaraan dan *time window depot*. DMPTW diketahui mempunyai kompleksitas NP-hard dikarenakan permasalahan ini mempunyai banyak rute *open vehicle* dan *depot*. Hal ini yang membuat WRRP dan DMPTW mempunyai kesamaan yaitu keduanya menerapkan aturan *open vehicle* dan *multiple depot* (dengan asumsi inisiasi posisi *workover rig* ditempatkan di lokasi-lokasi yang berbeda).

Proposal tugas akhir ini mengusulkan algoritma *Adaptive Large Search Neighborhood Search* (ALNS) [4] untuk menyelesaikan permasalahan WRRP. Hasil komputasional yang diperoleh akan berbeda-beda antara algoritma *Iterated Local Search* (ILS)[5], algoritma *Clustering Search* (CS) [5], dan algoritma ALNS. Hasil eksperimen komputasional menunjukkan bahwa algoritma yang diusulkan lebih efisien daripada algoritma ILS maupun CS untuk menyelesaikan permasalahan WRRP [6].

### 4.2 RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara mentransformasikan permasalahan VRPTW menjadi permasalahan WRRP ?
2. Bagaimana cara memahami konsep algoritma ALNS untuk penyelesaian permasalahan WRRP ?
3. Bagaimana cara mengimplementasikan algoritma ALNS untuk menyelesaikan permasalahan WRRP ?

4. Bagaimana cara menyusun uji coba yang sesuai dengan algoritma ALNS ?
5. Bagaimana cara merancang perbandingan komputasi algoritma ALNS dan algoritma *Parallel Tabu Search* ?

#### 4.3 BATASAN MASALAH

Permasalahan yang dibahas pada Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut :

1. Pembuatan tugas akhir ini menggunakan ketentuan *open vehicle router* yang berarti kendaraan *workover rig* tidak perlu kembali ke *depot* setelah melayani pelanggan terakhir.
2. Pada permasalahan WRRP, kecepatan kendaraan *workover rig* konstan.
3. Data permintaan perbaikan sumur minyak telah ditentukan sebelumnya.
4. Penyelesaian permasalahan WRRP menggunakan algoritma ALNS.
5. Simulasi eksperimen dilakukan menggunakan *solver* TOMLAB.

#### 4.4 TUJUAN DAN MANFAAT TUGAS AKHIR

Tugas akhir pada Tugas Akhir ini adalah, sebagai berikut :

1. Memahami cara mentransformasikan permasalahan VRPTW menjadi permasalahan WRRP.
2. Memahami konsep algoritma ALNS untuk penyelesaian permasalahan WRRP.
3. Mengimplementasikan algoritma ALNS yang mampu melakukan komputasi untuk menyelesaikan permasalahan WRRP.
4. Mengevaluasi kinerja dari algoritma ALNS dengan melakukan uji coba.
5. Mengetahui perbandingan komputasi algoritma ALNS dan algoritma *Parallel Tabu Search*.

Manfaat tugas akhir ini dikerjakan dengan tujuan untuk mengoptimalkan routing *workover rig* ketika terjadi kerusakan sumur minyak dengan algoritma ALNS pada operasi minyak *onshore* guna meminimalkan biaya produksi. Selain itu sedikitnya penelitian terkait permasalahan *Vehicle Routing Problem* dan variasinya yang lain. Dan memberikan alternatif metode dalam menyelesaikan WRRP menggunakan algoritma ALNS untuk mendapatkan solusi yang mendekati optimal.

## 5. TINJAUAN PUSTAKA

### ***RIG dan WORKOVER RIG***

*Rig* pengeboran adalah suatu bangunan dengan peralatan untuk melakukan pengeboran ke dalam reservoir bawah tanah untuk memperoleh air, minyak, atau gas bumi, atau deposit mineral bawah tanah. *Rig* pengeboran bisa berada di atas tanah (*on shore*) atau di atas laut/lepas pantai (*off shore*) tergantung kebutuhan pemakaiannya. Oleh karena itu, istilah "*rig*" mengacu pada kumpulan peralatan yang digunakan untuk melakukan pengeboran pada permukaan kerak bumi untuk mengambil contoh minyak, air, atau mineral. [7]

*Rig* pengeboran minyak dan gas bumi dapat digunakan tidak hanya untuk mengidentifikasi sifat geologis dari reservoir tetapi juga untuk membuat lubang yang memungkinkan pengambilan kandungan minyak atau gas bumi dari *reservoir* tersebut.

Umumnya, *rig* dikategorikan menjadi dua macam menurut tempat beroperasinya :

1. *Rig* darat (*land-rig*) : beroperasi di darat.
2. *Rig* laut (*offshore-rig*) : beroperasi di atas permukaan air (laut, sungai, rawa-rawa, danau atau delta sungai).

Dari fungsinya, *rig* dapat digolongkan menjadi dua macam :

1. *Drilling rig* : *rig* yang dipakai untuk membor sumur, baik sumur baru, cabang sumur baru maupun memperdalam sumur lama.
2. *Workover rig* : fungsinya untuk melakukan sesuatu terhadap sumur yang telah ada, misalnya untuk perawatan, perbaikan, penutupan, dan sebagainya.

### ***VEHICLE ROUTING PROBLEM***

*Vehicle Routing Problem* (VRP) diperkenalkan pertama kali oleh Dantziq dan Ramser pada tahun 1959 [2] dan semenjak itu telah dipelajari secara luas. Oleh Fisher [8], VRP didefinisikan sebagai berikut sebuah pencarian atas cara penggunaan yang efisien dari sejumlah *vehicle* yang harus melakukan perjalanan untuk mengunjungi sejumlah tempat untuk mengantar dan/atau menjemput orang/barang. Istilah *customer* digunakan untuk menunjukkan pemberhentian untuk mengantar dan/atau menjemput orang/barang. Setiap *customer* harus dilayani oleh satu *vehicle* saja. Penentuan pasangan *vehicle-customer* ini dilakukan dengan mempertimbangkan kapasitas *vehicle* dalam satu kali angkut, untuk meminimalkan biaya yang diperlukan. Biasanya, penentuan biaya minimal erat kaitannya dengan jarak yang minimal.

VRP adalah sebuah problem pemrograman integer yang masuk kategori *NP-Hard Problem*, yang berarti usaha komputasi yang digunakan akan semakin sulit dan banyak seiring dengan meningkatnya ruang lingkup masalah. Untuk masalah-masalah seperti ini, biasanya yang dicari adalah aproksimasi solusi yang terdekat, karena solusi tersebut dapat dicari dengan cepat dan cukup akurat. Biasanya masalah ini terselesaikan dengan menggunakan berbagai variasidari metode heuristik yang memerlukan sedikit pengamatan pada ruang lingkup masalah.

### ***VEHICLE ROUTING PROBLEM WITH TIME WINDOWS***

*Vehicle Routing Problem with Time Windows* adalah perluasan dari permasalahan dari VRP [2]. Jika pada VRP ditambahkan time window pada masing-masing konsumen, maka permasalahan tersebut menjadi VRPTW. Untuk VRPTW, selain

adanya kendala kapasitas kendaraan, terdapat tambahan kendala yang mengharuskan kendaraan untuk melayani tiap konsumen pada time frame tertentu.

Kendaraan boleh datang sebelum *time window* “open”, tetapi konsumen tersebut tidak dilayani sampai *time window* “open”. Kendaraan tidak diperbolehkan untuk datang setelah *time window* “closed”.

Definisi VRPT yang lain adalah permasalahan untuk menjadwalkan sekumpulan kendaraan, dengan kapasitas, dan *travel time* terbatas, dari *central depot* ke sekumpulan konsumen yang tersebar secara geografis, dengan *demand* diketahui, dalam *time windows* tertentu. *Time Windows* adalah *two sided*, yang berarti bahwa tiap konsumen harus dilayani saat atau setelah *earliest time*, dan sebelum *latest time* dari konsumen tersebut. Jika kendaraan datang ke konsumen sebelum *earliest time* dari konsumen tersebut, maka akan menghasilkan *idle* atau waktu tunggu. Kendaraan yang datang ke konsumen setelah *latest time* adalah *tardy*. Terdapat pula waktu *service* yang diperlukan untuk melayani tiap konsumen. Biaya rute dari suatu kendaraan adalah total dari waktu *travel* (proposional dengan jarak), waktu tunggu, dan waktu *service*, yang diperlukan untuk mengunjungi sekumpulan konsumen.

### **TOMLAB**

Tomlab adalah *software add-on* pada Matlab yang digunakan untuk memberikan solusi pada permasalahan yang bersifat yang memang digunakan untuk melakukan optimasi pada permasalahan dengan fungsi tujuan, batasan, variabel keputusan *integer*, atau *binary* yang berbentuk *linear programming* [9].

## 6. METODOLOGI

### 1. MEMBENTUK MODEL MATEMATIKA WRRP

Untuk lebih mempermudah penyelesaian WRRP maka permasalahan ini dimodelkan secara matematis. Misalkan  $K$  adalah himpunan sumur minyak dan saling terkait membentuk sebuah graph berarah  $G^k = (V^k, A^k)$  dimana  $V^k$  adalah himpunan node dan  $A^k$  adalah himpunan arc dari sumur minyak  $k \in K$ . Selanjutnya,  $V^k = W^k \cup \{o(k), d(k)\}$ , dimana  $o(k)$  dan  $d(k)$  adalah notasi untuk origin dan depot menunjukkan posisi awal dan posisi akhir dari sumur minyak  $k$  dan  $W^k$  adalah himpunan yang berisikan sumur minyak yang terbatas.  $W = \bigcup_{k \in K} W^k$  yang merepresentasikan semua sumur minyak harus dikunjungi sekali pada sumur minyak  $k$ .

Berikut model matematika pada permasalahan WRRP

(WRRP) minimize

$$\sum_{k \in K} \sum_{i \in W^k} P_i t_i^k \quad (1)$$

$$\sum_{k \in K} \sum_{j: (i,j) \in A^k} x_{ij}^k = 1 \quad i \in W \quad (2)$$

$$\sum_{j: (o(k),j) \in A^k} X_{o(k),j}^k = \sum_{i: (i,d(k)) \in A^k} X_{i,d(k)}^k = 1 \quad k \in K \quad (3)$$

$$\sum_{j: (i,j) \in A^k} X_{ij}^k - \sum_{j: (j,i) \in A^k} X_{ji}^k = 0 \quad k \in K, i \in W^k \quad (4)$$

$$X_{ij}^k (t_i^k + C_{ij} + t_j^k) \leq 0 \quad k \in K, (i,j) \in A^k \quad (5)$$

$$e_i \sum_{j: (i,j) \in A^k} X_{ij}^k \leq t_i^k \leq l_i \sum_{j: (i,j) \in A^k} X_{ij}^k \quad k \in K, i \in W^k \quad (6)$$

$$t_{o(k)}^k = 0 \quad k \in K \quad (7)$$

$$X_{ij}^k \in \{0,1\} \quad k \in K, (i,j) \in A^k \quad (8)$$

Persamaan (1) adalah persamaan total production yang harus diminimalisasi (fungsi tujuan). Berikutnya batasan (2) merupakan batasan yang bertujuan untuk membatasi setiap sumur minyak tepat dikunjungi satu kali. Batasan (2) dan (3) untuk mendefinisikan batasan mengenai *network flow* yang artinya jumlah flow yang keluar dari *node*  $i$   $\sum_{j: (i,j) \in A^k} X_{ij}^k$  harus sama dengan *flow* yang masuk ke *node*  $j$   $\sum_{j: (j,i) \in A^k} X_{ji}^k$  dan memastikan bahwa ketika *flow* itu berasal dari *origin node*  $o(k)$  maka akan berakhir pada *depot node*  $d(k)$ . Batasan (5) untuk mendefinisikan hubungan antara batasan waktu dan *arc flow*. Batasan (6) menunjukkan bahwa batasan *deadline* diperoleh dari batasan *time window* dimana *node*  $i$  dikunjungi oleh *workover rig*  $k$  sehingga nilai  $t_i^k$  (waktu untuk menuju sumur minyak ke  $k$ ) berada diantara nilai  $e_i$  (waktu tercepat menuju sumur minyak) dan  $l_i$  (tenggat waktu), selain itu jika

$t_i^k = 0$  maka tidak akan ada dampak pada persamaan (1) . Batasan (7) ini untuk inisiasi waktu pelayanan *rig* dimulai dari 0. Dan terakhir, untuk batasan (8) memberikan nilai berupa nilai biner untuk *variabel flow*.

## 2. ALGORITMA ADAPTIVE LARGE NEIGHBORHOOD SEARCH

Pada bagian ini akan dibahas penggunaan algoritma ALNS dalam permasalahan WRRP. Kata *neighborhood* mempunyai peranan penting dalam algoritma ALNS. Pencarian dengan berdasarkan *neighborhood* berarti mengambil solusi potensial untuk masalah dan memeriksa tetangganya (yaitu, solusi yang sama kecuali untuk satu atau dua detail kecil) dengan harapan menemukan solusi perbaikan.

Algoritma ALNS merupakan pengembangan dari algoritma *Large Neighborhood Search* (LNS) dengan berdasarkan prinsip membangun dan menciptakan untuk menemukan solusi perbaikan. Pada setiap iterasi, algoritma ini akan menghapus bagian dari solusi ( $s$ ) yang didapat dan selanjutnya memperbaiki solusi tersebut dan dijadikan solusi baru ( $s'$ ). Solusi baru ini selanjutnya diterima berdasarkan syarat jika  $s'$  lebih baik daripada  $s$  maka pencarian solusi dimulai dari  $s'$  jika sebaliknya maka pencarian solusi dimulai dari  $s$  dengan beberapa probabilitas.

Dalam ALNS, prosedur menghapus dan memperbaiki dipilih berdasarkan mekanisme pemilihan peluang yang adaptif. Di setiap iterasi, prosedur pemilihan peluang bergantung pada sebagaimana baik performa solusi sebelumnya.

Selanjutnya dalam implementasi ALNS untuk permasalahan WRRP di deskripsikan dalam beberapa langkah utama antara lain :

1. *Large neighborhood*: Solusi  $s$  pada setiap iterasi,  $q$  sumur minyak dihapus dan disisipkan lagi ke dalam set solusi. Proses ini menggunakan metode penghapusan dan penyisipan secara heuristik. Pada setiap iterasi, bilangan integer akan dirandom dan yang memenuhi  $2 \leq q \leq \min \{150, \xi |W|\}$  akan dipilih. Simbol  $\xi$  adalah parameter untuk mengontrol banyaknya sumur minyak yang akan dihapus.
2. *Adaptive search engine*: Pemilihan dalam metode heuristik penghapusan dan perbaikan menggunakan mekanisme *roulette-wheel* dimana setiap nilai heuristik akan ada pembobotan sesuai dengan tingkah laku sebelumnya. Lebih detailnya, misalkan  $W_i$  adalah ukuran heuristik bagaimana performa sumur minyak  $i$  pada iterasi sebelumnya. Kemudian, diketahui nilai heuristik  $h$  dengan bobot  $w_i$ , nilai heuristik  $j$  dipilih dengan rumus probabilitas  $w_j / \sum_{i=1}^h w_i$ . Mekanisme heuristik untuk memperbaiki dipilih secara independen dengan mekanisme heuristik untuk menghapus.
3. *Adaptive weight adjustment*: Pencarian solusi dibagi dalam beberapa segmen yang banyaknya segmen telah diinputkan sebelumnya. Pada segmen pertama, semua heuristik mempunyai bobot yang sama yaitu 1. Pada iterasi ke- $\phi$ , bobot tersebut untuk memilih perbaikan dan penghapusan heuristik yang terus di update berdasarkan skor yang didapat selama beberapa segmen. Skor tersebut menunjukkan seberapa baik performa yang ditunjukkan perbaikan dan penghapusan hingga segmen yang terakhir. Ketika ditemukan solusi terbaru, peningkatan skor yang diberikan berdasarkan parameter  $\sigma_1$ ,  $\sigma_2$ , atau  $\sigma_3$ . Untuk kasus ini, jika pasangan heuristik perbaikan dan penghapusan menemukan solusi terbaik yang baru, maka skornya ditingkatkan menjadi  $\sigma_1 = 50$ , jika ditemukan solusi yang lebih baik dari sebelumnya, maka skornya ditingkatkan menjadi  $\sigma_2 = 10$ , dan jika solusi yang ditemukan tidak mengalami peningkatan dan itu diterima maka skornya ditingkatkan menjadi  $\sigma_3 = 6$ . Ketika segmennya berakhir, bobot yang baru akan dihitung menggunakan rumus penghitungan bobot, dan semua skor akan dikembalikan menjadi 0 untuk segmen berikutnya.

Rumus penghitungan bobot

$$W_{ij} = \begin{cases} W_{ij} & \text{if } O_{ij} = 0 \\ (1 - \eta)W_{ij} + \frac{\eta \pi_i}{O_{ij}} & \text{if } O_{ij} \neq 0 \end{cases} \quad (9)$$

$\pi_i$  adalah nilai skor dari heuristik  $i$

$O_{ij}$  adalah nilai yang menunjukkan berapa banyak heuristic  $I$  telah dipilih pada segmen  $j$

$\eta \in [0,1]$  adalah factor reaksi, parameter ini digunakan kontrol seberapa cepat reaksi algoritma penambahan bobot untuk keefektifitasan heuristic. Dalam proposal ini menggunakan nilai  $\eta = 0.01$ .

4. *Penalized objective function*: Setiap solusi yang dihasilkan akan dievaluasi dengan fungsi tujuan (persamaan (1)) . Jadi setiap solusi yang melebihi tenggat waktu akan dipertimbangkan. Berikut adalah fungsi tujuan untuk tenggat waktu.

$$v(s) = \sum_{k \in K} \sum_{i \in W^k} (p_i t_i^k + \lambda \max\{0, t_i^k - l_i\}) \quad (10)$$

$\lambda$  adalah nilai yang diinputkan oleh user dan  $s$  adalah solusi yang akan divalusi. Sedangkan  $p_i t_i^k$  adalah persamaan total production (fungsi tujuan),  $t_i^k$  adalah waktu untuk menuju sumur minyak ke  $k$ , dan  $l_i$  adalah tenggat waktu.

5. *Acceptance and stopping criteria*: Kriteria penerimaan menggunakan metode SA, contoh: diberikan solusi saat ini  $s$ , solusi tetangga  $s'$  diterima jika  $v(s') < v(s)$ , dan dengan peluang  $e^{-(v(s') - v(s))/T}$ . Dimana  $T > 0$  adalah temperature saat ini dan  $v(s)$  adalah penalized solution cost. Temperatur di set mulai dari  $T_{\text{start}} = 2500$  dan mengalami penurunan pada setiap iterasi  $cT$ , dimana jangkauan nilainya  $0 < c < 1$ . Nilai  $c$  di set 0.99975 pada eksperimen komputasional ini.

### 3. METODE HEURISTIK PENGHAPUSAN ( $H^-$ ) dan PENYISIPAN ( $H^+$ )

Bagian ini menjelaskan lima penghapusan dan tiga penyisipan heuristic. Empat penghapusan heuristic menghapus sumur ( $q$ ) dari solusi ( $s$ ) dan mengembalikan ( $a$ ) solusi ( $s'$ ). Salah satu penghapusan heuristic menghapus rute secara acak dan jumlah ( $q$ ) menghapus sumur yang bergantung pada rig yang sesuai. Ketika diterapkan,  $D$  menjadi satu set penghapusan sumur.

#### 1. Penghapusan heuristic Shaw 1 – PHS1

Ide umum penghapusan Shaw [10] heuristic adalah untuk menghapus sumur yang memiliki kemiripan, sehingga dapat memberikan solusi yang lebih baik. Tingkat kesamaan antara dua sumur  $i$  dan  $j$ , masing-masing dilayani oleh  $rig$   $k_1$  dan  $k_2$ , dihitung melalui ukuran keterkaitan  $R(i, j)$ , di mana nilai yang lebih rendah sesuai dengan sumur yang serupa. Penghapusan shaw heuristic pertama (PHS1) didasarkan pada perbedaan nilai mutlak antara kontribusi sumur  $i$  dan  $j$  ke fungsi tujuan. Lebih tepatnya, ukuran keterkaitan pertama ditunjukkan dengan:

$$R_{PHS1} = |p_i t_i^{k_1} - p_j t_j^{k_2}| \quad (11)$$

Keterkaitan ini digunakan sesuai dengan milik Shaw [10] untuk menghapus sumur ( $q$ ). Mengingat solusi ( $s$ ) dan satu set sumur  $D$  dihapus, algoritma acak memilih sumur ( $w$ )



dari  $D$ , menghitung ukuran keterkaitan antaranya dan semua sumur yang belum dihapus, dan kemudian memilih sebuah sumur baru yang akan dimasukkan ke dalam  $D$ . Proses ini diulang ketika  $|D| < q$ . Menurut Ropke dan Pisinger [4], parameter  $v \geq 1$  digunakan untuk menghindari determinisme ketika memilih sumur.

## 2. Penghapusan heuristik Shaw 2 – PHS2

Penghapusan shaw heuristik (PHS2) berdasarkan pada jarak. Kesamaan antara dua sumur  $i$  dan  $j$  diukur dengan mengukur keterkaitan  $PHS2(i, j) = d_{ij}$ , dimana  $d_{ij}$  adalah jarak antara  $i$  dan  $j$ . Terlepas dari ukuran keterkaitan, prosedur ini identik dengan yang sebelumnya.

## 3. Penghapusan acak heuristik– PAH

Penghapusan heuristik sederhana ini menghilangkan sumur ( $q$ ) secara acak untuk solusi ( $s$ ) saat itu. Kecenderungan heuristik untuk menghasilkan satu himpunan berisikan sumur kualitas rendah yang dimusnahkan, tetapi membantu diversifikasi pencarian.

## 4. Penghapusan rute acak – PRA

Route removal heuristic merupakan satu lagi penghapusan heuristik sederhana di mana beberapa rute ( $k$ ) yang dipilih secara acak dan sumur tersebut akan dihapus. Namun, seperti yang ditunjukkan pada algoritma ALNS bagian ke 3, parameter  $\phi \geq 1$  memperkenalkan beberapa keacakan di urutan sumur yang dihapus. Sebuah greedy heuristik yang sederhana menyisipkan sumur sesuai dengan urutan penghapusan. Ketidakteraturan membantu diversifikasi pencarian.

## 5. Penghapusan heuristik terburuk – PHT

Heuristik ini untuk menghilangkan sumur dengan biaya tinggi dalam solusi ( $s$ ) dengan upaya untuk memasukkan ke dalam posisi yang lebih baik.  $ProdLoss(i, s) = v(s) - v_{-i}(s)$  menjadi kerugian produksi yang disebabkan oleh sumur ( $i$ ) dalam solusi ( $s$ ), di mana  $v_{-i}(s)$  merupakan solusi biaya tanpa sumur ( $i$ ) kepada ( $s$ ). Urutan pertama PRA sesuai  $ProdLoss(i, s)$ , pilih satu untuk dihapus, memperhitungkan ulang  $ProdLoss(i, s)$  untuk sumur yang tersisa, dan proses ini diulang. Lihat Algoritma 4 untuk rincian dan perhatikan bahwa penghapusan juga acak, tetapi dikendalikan oleh parameter  $\rho \geq 1$ .

## 6. Penyisipan heuristik *basic greedy*– PHBG

$\Delta_{vik}$  mengubah solusi biaya yang dikeluarkan dengan memasukkan sumur ( $i$ ) kedalam rute ( $k$ ) dalam posisi untuk meningkatkan kehilangan produksi pada solusi ( $s$ ) yang terjadi saat itu. Kemudian,  $ProdLoss^+(i, s) = \min_{k \in K} \{\Delta_{vik}\}$  menjadi kerugian produksi minimum yang ditemukan mewakili posisi biaya minimum. Jika  $D$  adalah seperangkat sumur yang dihapus, kita menghitung  $ProdLoss^+(D_1, s)$  dan sumur pertama ( $D_1$ ) akan dihapus dari  $D$  dan dimasukkan ke dalam rute yang sesuai dalam posisi yang terbaik. Perhatikan bahwa penyisipan *greedy* heuristik mematuhi urutan sumur di  $D$ . Setelah sumur pertama telah dimasukkan,  $ProdLoss^+(D_1, s)$  dihitung lagi dan proses ini diulang-ulang.

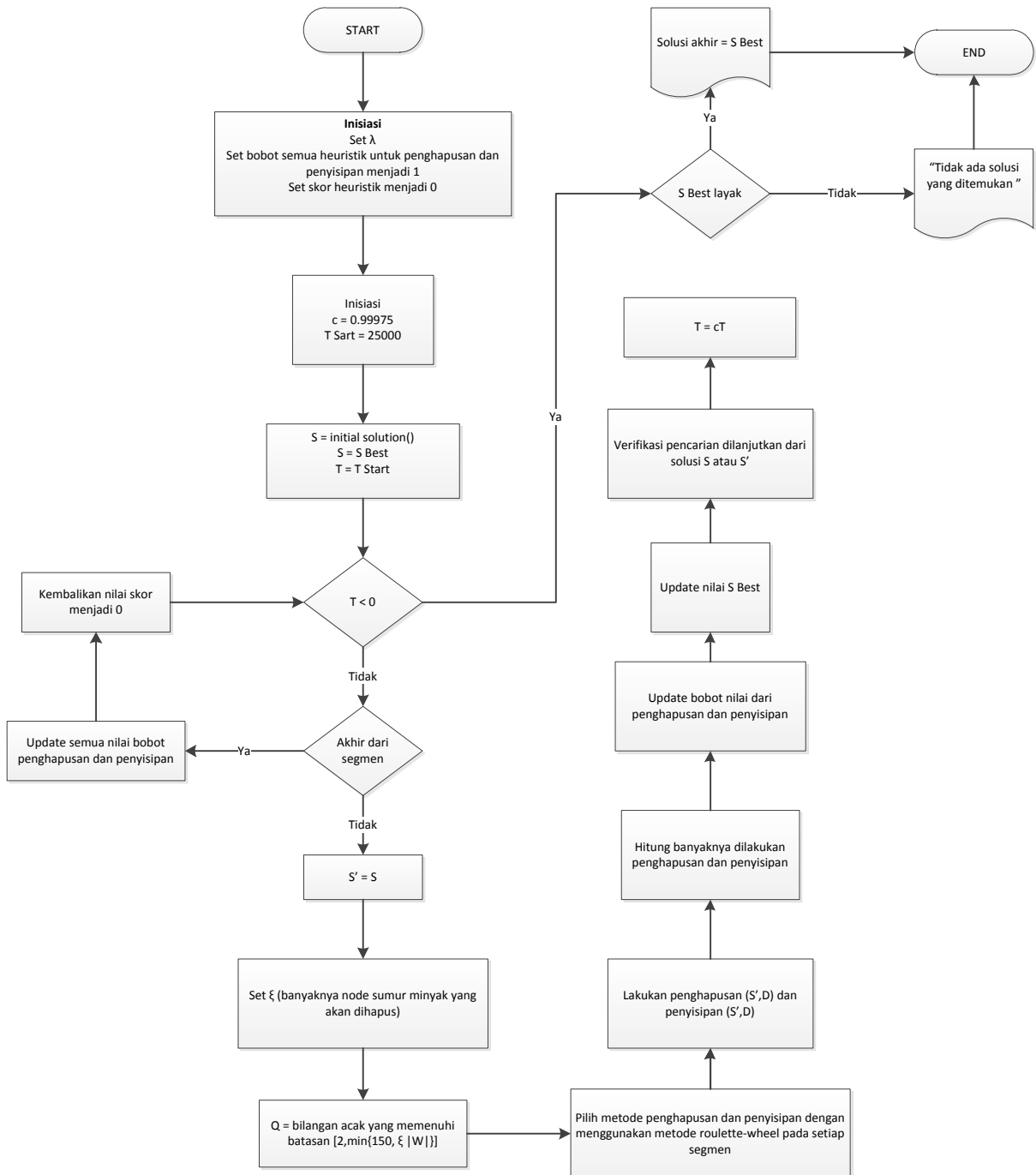
## 7. Penyisipan heuristik *deep greedy*– PHDG

Heuristik ini bekerja secara berbeda dari yang sebelumnya. Memasukkan sumur pertama  $i$  dari  $D$  ke dalam solusi  $s$ , menyisipkan heuristik sumur  $i$  dengan posisi biaya minimum global. Secara formal, ia memasukkan sumur  $i$  yang meminimalkan  $\min_{i \in D} \{ProdLoss^+(i, s)\}$  pada posisi biaya minimum. Proses ini diulang sampai semua sumur telah dimasukkan. Akibatnya, heuristik ini lebih lambat dari PHBG.

8. Regret- $k$  insertion heuristic – RIH

Menurut Azi et al. (2010)[7], Potvin dan Rousseau (1993)[10] dan Ropke dan Pisinger (2006a)[10], *regret* heuristik mencoba untuk memperbaiki perilaku greedy heuristik. Diberi sekumpulan himpunan  $D$  untuk sumur yang dihapus, untuk masing-masing sumur  $i \in D$ , heuristik ini menghitung *regret value* dengan perbedaan biaya antara dua solusi di mana dimasukkan ke dalam rute terbaik atau kedua terbaik. Sumur  $i$  dengan *regret value* maksimum dipilih dan dimasukkan dalam solusi  $s$ . Secara formal,  $\omega_{ik} \in K$  menjadi variabel yang menunjukkan rute sumur  $i$  memiliki biaya terendah ke- $k$ , yaitu,  $\Delta v_{i\omega_{ik}} \leq \Delta v_{i\omega_{ik'}}$  untuk semua  $k \leq k'$ . Dengan demikian, *regret value* adalah  $RegretValue_i = \Delta v_{i\omega_{i2}} - \Delta v_{i\omega_{i1}}$  dan pada setiap iterasi, heuristik memilih sumur  $i$  menurut  $\max_{i \in D} \{RegretValue_i\}$ . Selanjutnya

#### 4. DIAGRAM ALIR ALGORITMA ALNS



Gambar 1. Flowchart Algoritma Adaptive Large Neighborhood Search

## 7. JADWAL KEGIATAN TUGAS AKHIR

No.	Tahapan	Bulan											
		Maret			April			Mei			Juni		
1.	Penyusunan Proposal												
2.	Studi Literatur												
3.	Implementasi												
4.	Pengujian dan Evaluasi												
5.	Penyusunan Buku Tugas Akhir												

## 8. DAFTAR PUSTAKA

- [1] APMI. (2012, November) Asosiasi Perusahaan Pemboran Minyak, Gas dan Panas Bumi Indonesia. [Online].  
[http://apmi-online.org/apmi/index.php?option=com\\_content&view=article&id=62:surat-keputusan-tho&catid=1:latest-news&Itemid=55](http://apmi-online.org/apmi/index.php?option=com_content&view=article&id=62:surat-keputusan-tho&catid=1:latest-news&Itemid=55)
- [2] Paolo Toth and Daniele Vigo, Eds., *The Vehicle Routing Problem*. Bologna, Italy: Siam Monographs on Discrete Mathematics and Applications, 2002.
- [3] Géraldine Heilporn, Jean-François Cordeau, and Gilbert Laporte, "The Delivery Man Problem with time windows," *Discrete Optimization*, vol. 7, July 2010.
- [4] David Pisinger and Stefan Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, pp. 2403 – 2435, October 2007.
- [5] Antonio Augusto Chaves and Luiz Antonio Lorena, "Clustering search algorithm for the capacitated centered clustering problem," *Computers & Operations Research*, vol. 37, pp. 552-558, October 2010.
- [6] Glaydston Mattos Ribeiro, Gilbert Laporte, and Geraldo Regis Mauri, "A comparison of three metaheuristics for the workover rig routing problem," *European Journal of Operational Research*, vol. 220, pp. 28–36, January 2012.
- [7] Noor Adinugroho. (2013, March) Noor Adinugroho. [Online].  
<http://nooradinugroho.wordpress.com/2007/09/27/macam-sumur-dan-rig-dalam-perminyakan/>
- [8] Marshall Fisher, *Handbooks in Operations Research and Management Science*. Philadelphia, United States of America, 1995.
- [9] (2013, March) Tomlab Optimization. [Online].  
<http://tomopt.com/tomlab/about/>
- [10] Shaw, Paul, "A new local search algorithm providing high quality," Department of Computer Science, University of Strathclyde, Glasgow, 1997.