

**Pengembangan Model RUASP – GRASP Metaheuristics
Dengan Mempertimbangkan Time Window:
Studi Kasus Bencana Letusan Gunung Semeru di Jawa Timur**

Riky Adi Putra S.

Jurusan Teknik Industri Fakultas Teknik
riky.a7@gmail.com

Abstrak - Indonesia sangat rawan terhadap bencana gunung meletus. Mengingat hal tersebut, Indonesia perlu untuk melakukan manajemen bencana yang baik. Salah satu fase yang cukup krusial adalah fase tanggap darurat, khususnya ketika kegiatan evakuasi. Pengalokasian dan penjadwalan tim penyelamat yang tepat diperlukan untuk meminimalisir waktu evakuasi dengan harapan bisa menyelamatkan korban jiwa sebanyak mungkin. Untuk itu, digunakan model optimasi RUASP (*Rescue Unit Assignment and Scheduling Problem*) sebagai model awal. Model ini memiliki fungsi tujuan untuk meminimumkan *completion time*. Namun, model ini belum mempertimbangkan kurun waktu yang aman bagi tim penyelamat saat sedang bertugas. Faktor alam, keterbatasan fisik, dan keterbatasan alat menjadi penyebabnya. Oleh karena itu, dalam pengembangan model ditambahkan batasan terkait *time window* dan penyesuaian terhadap langkah-langkah GRASP *Metaheuristics*. Model dikerjakan dengan menggunakan software MATLAB R2009a sebanyak 300 iterasi. Hasil pengujian menunjukkan perbedaan *completion time* sebesar 6,25% lebih cepat dibandingkan dengan model awal dengan waktu komputasi selama 12,56 detik. Sedangkan, hasil analisis sensitivitas menunjukkan bahwa fungsi tujuan tidak sensitif terhadap perubahan waktu operasional tim, namun sensitif terhadap perubahan kapabilitas tim. Selain itu, didapati temuan lain pula yang menunjukkan bahwa nilai kapabilitas tim ternyata sangat mempengaruhi keseimbangan beban kerja tim.

Kata kunci: *disaster, humanitarian logistics, penjadwalan dan pengalokasian tim penyelamat, GRASP Metaheuristics, time window.*

Abstract - Indonesia is very vulnerable to the volcano eruption. Given this, Indonesia needs to do a good disaster management. One of the most crucial phases is the emergency response period, especially evacuation activity. The proper allocation and scheduling of rescue teams is needed to minimize evacuation time with expectation of saving as many lives as possible. For that purpose, RUASP (*Rescue Unit Assignment and Scheduling Problem*) model is used as the initial model. This model has an objective function to minimize completion time. However, this model has not considered a safe period for rescue teams while on duty. Natural factors, physical limitations, and device limitations are the causes. Therefore, in the model development, time window constraint is added and there are few adjustments to its GRASP *Metaheuristics* steps. The model is done by using MATLAB R2009a with 300 iterations. The test results show completion time 6.25% faster than the initial model with computation time of 12.56 seconds. Meanwhile, sensitivity analysis results indicate that objective function is not sensitive to changes in team operational time, but sensitive to changes in team capability. In

addition, other findings also show that team's capability was greatly affect the balance of workload team.

Keywords: *disaster, humanitarian logistics, scheduling dan allocating rescue team, GRASP Metaheuristics, time window*

PENDAHULUAN

Indonesia merupakan negara kepulauan yang rawan terjadi bencana alam, khususnya gunung meletus. Salah satu penyebabnya adalah karena Indonesia terletak di jalur Cincin Api Pasifik. Penyebab lainnya adalah karena Indonesia terletak pada pertemuan 4 lempeng tektonik, yaitu lempeng Benua Asia, lempeng Benua Australia, lempeng Samudera Hindia, dan lempeng Samudera Pasifik. Pada bagian Selatan dan Timur Indonesia terdapat juga sabuk vulkanik (*volcanic arc*) yang memanjang dari Pulau Sumatera, Jawa, Nusa Tenggara, Sulawesi, yang sisinya berupa pegunungan vulkanik tua dan dataran rendah yang sebagian didominasi oleh rawa-rawa (BNPB, 2016; www.bnpb.go.id). Menurut BNPB (2016; www.bnpb.go.id), gunung meletus merupakan akibat dari adanya aktivitas vulkanik (erupsi) yang sangat berbahaya karena mengakibatkan terjadinya awan panas, lontaran material (pijar), hujan abu lebat, lava, gas racun, banjir lahar, bahkan tsunami. Selain menimbulkan banyak korban jiwa, gunung meletus juga menyebabkan kerugian ekonomi yang besar karena banyaknya infrastruktur yang rusak dan memerlukan biaya pemulihan yang tinggi.

Dalam Altay & Green (2006), terdapat 4 fase dalam manajemen bencana. Salah satu fase tersebut dan yang menjadi sorotan dalam penelitian ini adalah fase tanggapan (masa tanggap darurat). Menurut UU No. 24 Tahun 2007 tentang Penanggulangan Bencana, pada masa ini dilakukan upaya-upaya tertentu untuk menangani dampak buruk yang ditimbulkan oleh bencana tersebut seperti kegiatan penyelamatan dan evakuasi korban, harta benda, pemenuhan kebutuhan dasar, perlindungan, pengurusan pengungsi, penyelamatan, serta pemulihan prasarana dan sarana. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa dalam fase ini perlu mendapat perhatian lebih dan penanganan yang cepat, tepat, dan tanggap, khususnya pada saat evakuasi korban. Evakuasi korban menjadi hal yang tentu tidak bisa dihindari pada saat tanggap darurat. Pengalokasian dan penjadwalan tim penyelamat yang tepat diperlukan untuk meminimalisir waktu evakuasi dengan harapan bisa menyelamatkan korban jiwa sebanyak mungkin.

Untuk menyelesaikan masalah seperti uraian di atas, digunakan model optimasi RUASP Rizqi (2016) sebagai model awal. Model RUASP ini memiliki fungsi tujuan untuk meminimumkan *completion time*. Model ini dapat dikatakan sudah cukup kompleks mengingat banyak faktor penting yang sudah dipertimbangkan, seperti faktor tingkat keparahan dan kemampuan tim penyelamat. Meskipun begitu, ternyata masih ada hal yang cukup krusial yang belum dipertimbangkan dalam model ini. Hal tersebut adalah belum dipertimbangkannya batasan waktu (*time window*) bagi tim penyelamat untuk dapat melakukan tugasnya dengan aman. Keterbatasan fisik dan keterbatasan alat adalah faktor penghambat dilakukannya kegiatan evakuasi pada saat keadaan kurang cahaya. Jadi, apabila model ini tidak mempertimbangkan penggunaan *time window* di dalamnya, maka akan dapat membahayakan keselamatan dari tim penyelamat itu sendiri saat sedang bertugas nantinya. Oleh karena itu, perlu dipertimbangkan adanya penggunaan *time window* untuk mengantisipasi hal ini dan menyempurnakan model yang sudah ada.

Berdasarkan latar belakang tersebut, maka dapat disimpulkan bahwa yang menjadi permasalahan dalam penelitian ini adalah bagaimana menentukan pengalokasian dan penjadwalan tim penyelamat yang tepat sehingga dapat menghindari jatuhnya korban jiwa selama kurun waktu tertentu, sekaligus tidak membahayakan keselamatan dari tim penyelamat saat sedang bertugas. Adapun tujuan yang ingin dicapai adalah untuk mendapatkan pengalokasian dan penjadwalan tim penyelamat yang tepat dengan nilai *completion time* paling minimum yang sudah memperhatikan *time window* melalui metode GRASP *Metaheuristics*. Dengan demikian, tim penyelamat tetap dapat melakukan tugasnya dengan baik tanpa harus membahayakan keselamatannya sendiri sehingga bisa menyelamatkan para korban letusan gunungapi. Selain itu, akan dilakukan juga analisis sensitivitas terhadap model yang sudah dikembangkan nantinya untuk mengetahui pengaruh perubahan parameter terhadap solusi optimal yang didapat. Parameter tersebut adalah waktu operasional dan kapabilitas tim penyelamat.

METODE PENELITIAN

Untuk mencapai tujuan penelitian ini, diperlukan langkah-langkah yang sistematis dan terstruktur agar tidak menyimpang dari jalur. Langkah yang pertama

adalah mengidentifikasi masalah yang ada. Masalah tersebut adalah berupa model RUASP yang belum mempertimbangkan *time window*. Langkah kedua adalah melakukan studi kepustakaan yang diperlukan. Langkah ketiga adalah pendefinisian rumusan dan tujuan masalah yang ingin dicapai. Langkah keempat adalah menyusun skenario yang terdiri atas beberapa data yang diperoleh di lapangan (seperti lokasi insiden) dan beberapa data asumsi (seperti waktu operasional). Langkah kelima adalah menyusun *programming* MATLAB untuk metode awal GRASP *Metaheuristics*. Setelah itu, dilakukan uji validasi dan analisis hasilnya. Melalui analisis hasil tersebut, maka model dapat dikembangkan. Lalu, dilakukan penyusunan *programming* MATLAB untuk metode usulan GRASP *Metaheuristics*. Uji validasi dan analisis hasil juga dilakukan pada metode usulan ini. Selanjutnya adalah melakukan analisa sensitivitas. Langkah terakhir yang dilakukan adalah berupa penyusunan kesimpulan dan saran.

DATA YANG DIPERLUKAN

Secara garis besar, data yang diterapkan pada kasus ini didapat melalui Rencana Kontinjensi (Renkon) Gunung Semeru Tahun 2015 beserta tambahan data hasil asumsi. Data yang didapat dari Renkon adalah data terkait jumlah lokasi insiden sebanyak 17 lokasi dan 2 lokasi semu yaitu posko yang sama yang terletak di Kota Lumajang. Sedangkan, data hasil asumsi adalah data yang tidak didapatkan di Renkon, tetapi diperlukan untuk menyelesaikan model yang ada. Data tersebut sengaja dibuat dengan menggunakan logika, pendekatan distribusi tertentu, maupun *fuzzy logic* untuk dapat merepresentasikan kondisi yang sebenarnya di lapangan. Data tersebut antara lain tingkat keparahan lokasi, jumlah tim yang tersedia, waktu perjalanan tim, waktu operasional tim, kapabilitas tim, dan kapabilitas yang dibutuhkan di tiap lokasi.

GRASP METAHEURISTICS

GRASP (*Greedy Randomized Adaptive Search Procedure*) *Metaheuristics* adalah sebuah algoritma metaheuristik yang memberikan variasi solusi dari heuristik konstruksi yang ada. Penggunaan GRASP *Metaheuristics* ini bertujuan

untuk mengatasi kekurangan dari heuristik-heuristik yang dibangun sebelumnya yang mana hanya memberikan solusi yang sama secara berulang-ulang (Wex *et al.*, 2014). Dengan penggunaan GRASP *Metaheuristics*, maka akan didapatkan solusi yang paling optimal. Di dalam GRASP *Metaheuristics* sendiri terdapat 2 fase yang harus dilakukan. Fase yang pertama adalah fase *construction* yang mana pada fase ini heuristik yang digunakan akan menghasilkan solusi yang *feasible* (layak). Lalu, fase yang kedua adalah fase *local search* yang mana pada fase ini akan dilakukan pencarian solusi yang paling optimal dari heuristik yang telah dikembangkan tersebut.

UJI COBA MODEL AWAL

Running model pada penelitian ini menggunakan laptop dengan prosesor Intel Core i3 @1.80Ghz, RAM 4GB, HDD 500GB, OS Windows 10 Education, dan software MATLAB R2009a. Dengan iterasi sebanyak 100 kali, didapatkanlah *completion time* selama 634 menit (10 jam 34 menit) dengan waktu komputasi selama 4 detik. Dari uji validasinya, model ini dinyatakan valid.

Akan tetapi, hasil yang didapat pada percobaan di atas tidak adil untuk nantinya dibandingkan dengan hasil yang didapat pada model usulan. Bagian ini memang sengaja dibuat setelah uji coba pada model usulan sudah dilakukan dengan maksud untuk mendapatkan perbandingan yang adil. Pada uji coba model usulan, iterasi yang dibutuhkan adalah sebanyak 300 iterasi. Uji coba model usulan dengan 100 iterasi maupun 200 iterasi tidak menghasilkan adanya solusi karena adanya pengaruh dari batasan *time window*. Sedangkan, pada uji coba model awal seperti di atas, hanya dibutuhkan sebanyak 100 iterasi saja untuk memperoleh solusi. Dengan perbedaan jumlah iterasi tersebut, maka hasil yang diperoleh juga tidak adil untuk dibandingkan. Hal ini dikarenakan uji coba pada model awal dihentikan lebih cepat sebelum model mencapai solusi yang paling optimum. Oleh karena itu, dilakukan juga iterasi sebanyak 300 kali pada uji coba model awal. Hasil yang didapat adalah *completion time* selama 528 menit (8 jam 48 menit) dengan waktu komputasi selama 12,4 detik dengan perincian sebagai berikut.

Tim 1: Kota Lumajang - Desa Sumbermujur - Desa Pasrujambe - Desa Gesang -
Desa Nguter - Kota Lumajang (Total waktu = 466 menit)

Tim 2: Kota Lumajang - Desa Kloposawit - Kota Lumajang (Total waktu = 200 menit)

Tim 3: Kota Lumajang - Desa Kertosari - Desa Sememu - Kota Lumajang (Total waktu = 183 menit)

Tim 4: Kota Lumajang - Desa Penanggal - Desa Jogosari - Kota Lumajang (Total waktu = 349 menit)

Tim 5: Kota Lumajang - Desa Jatisari - Kota Lumajang (Total waktu = 116 menit)

Tim 6: Kota Lumajang - Desa Supit Urang - Desa Pronojiwo - Desa Oro-Oro Ombo - Desa Gondoruso - Kota Lumajang (Total waktu = 528 menit)

Tim 7: Kota Lumajang - Desa Sumber Urip - Desa Sumberwuluh - Desa Sumberrejo - Kota Lumajang (Total waktu = 503 menit)

Tabel 1. Perbandingan Hasil Uji Coba Model Awal 100 Iterasi Dengan 300 Iterasi

	<i>Completion Time</i>	Selisih	% Perbedaan
100 iterasi	634 menit	106 menit	$\frac{106}{634} \times 100\% = 16,72\%$
300 iterasi	528 menit		

Batasan *time window* memaksa model memerlukan iterasi lebih banyak agar solusi yang didapatkan adalah solusi yang paling optimal. Pada Tabel 1 dapat dilihat bahwa perbedaan solusi (*completion time*) dari jumlah iterasi yang berbeda adalah cukup signifikan, yaitu sebesar 16,72%. *Completion time* yang dihasilkan melalui 300 iterasi jauh lebih optimal dibandingkan dengan *completion time* yang dihasilkan melalui 100 iterasi.

Tabel 2. Perbandingan Kapabilitas Tiap Tim Terhadap Kapabilitas yang Dibutuhkan di Lokasi

Kapabilitas Tim	Keperluan Kapabilitas di Tiap Lokasi
Tim 1 (0,74)	Desa Sumbermujur (0,61) - Desa Pasrujambe (0,57) - Desa Gesang (0,34) - Desa Nguter (0,53)
Tim 2 (0,54)	Desa Kloposawit (0,45)
Tim 3 (0,83)	Desa Kertosari (0,44) - Desa Sememu (0,49)
Tim 4 (0,62)	Desa Penanggal (0,55) - Desa Jogosari (0,51)
Tim 5 (0,43)	Desa Jatisari (0,29)
Tim 6 (0,91)	Desa Supit Urang (0,65) - Desa Pronojiwo (0,72) - Desa Oro-Oro Ombo (0,81) - Desa Gondoruso (0,55)
Tim 7 (0,85)	Desa Sumber Urip (0,69) - Desa Sumberwuluh (0,64) - Desa Sumberrejo (0,66)

Selain itu, dari hasil yang didapat di atas juga, dapat dilihat bahwa beban kerja tiap tim tidak merata. Ada tim yang menangani banyak insiden sampai dengan 4 lokasi, seperti tim 1 dan 6. Ada juga tim yang mengerjakan sedikit insiden sehingga waktu penyelesaiannya menjadi sebentar saja, seperti tim 2 dan 5. Kapabilitas tim

yang rendah mengakibatkan tim tersebut tidak mampu menangani banyak insiden. Hal inilah yang mengganggu *load balancing heuristics* yang digunakan karena insiden terakhir tidak dapat dipindahkan ke antrian tim lain yang kapabilitasnya tidak mencukupi kapabilitas yang diperlukan oleh insiden tersebut.

PENGEMBANGAN MODEL

Model awal (Rizqi, 2016) akan dikembangkan dengan menambahkan batasan *time window*. Batasan ini diperlukan mengingat kegiatan evakuasi harus memperhatikan juga kurun waktu yang aman bagi tim penyelamat ketika sedang bertugas. Keamanan tim penyelamat dapat dipengaruhi oleh faktor alam, khususnya potensi gunung yang diperkirakan bisa meletus lagi, faktor keterbatasan fisik, dan juga faktor keterbatasan alat saat keadaan di lapangan kurang cahaya (gelap). Berdasarkan keterangan dari pihak BPBD Lumajang, kegiatan evakuasi yang baik biasanya dimulai pada pukul 07.00 saat matahari sudah bersinar dan berakhir paling lambat pada pukul 17.00 sebelum matahari terbenam. Periode waktu tersebut itulah yang menjadi *time window* pada kasus ini yang mana tergolong ke dalam jenis hard time window seperti dalam VRP (*Vehicle Routing Problem*) dalam Cordeau *et al.* (2002). Jika dalam VRP kendaraan tidak diijinkan untuk melakukan pelayanan di luar kurun waktu yang ada, maka pada kasus ini tim penyelamat juga tidak diijinkan untuk melakukan kegiatan evakuasi di luar kurun waktu tersebut. Tidak ada penalti yang diberikan karena tim penyelamat hanya bisa melakukan tugasnya di dalam kurun waktu yang diberikan. Kegiatan evakuasi yang baik harusnya mampu menyelamatkan korban jiwa tanpa menimbulkan jatuhnya korban jiwa yang lain, apalagi dari pihak tim penyelamat itu sendiri. Oleh karena itu, penambahan batasan *time window* dalam model menjadi solusi yang terbaik untuk menyelesaikan permasalahan ini.

Berikut adalah keseluruhan model matematis yang baru setelah adanya penambahan batasan *time window*. Perubahan yang dibuat ditunjukkan melalui tulisan yang berwarna biru.

Parameter

n	Jumlah insiden, dengan $set I = (1, \dots, n)$
m	Jumlah tim penyelamat, dengan $set K = (1, \dots, m)$

$w_j \in R^{\geq 0}$	Tingkat keparahan insiden j
$p_j^k \in R^{\geq 0}$	Waktu yang dibutuhkan tim penyelamat k untuk memproses insiden j ; ∞ jika tim penyelamat k tidak mampu mengatasi insiden j .
$s_{ij}^k \in R^{\geq 0}$	Waktu perjalanan yang dibutuhkan tim penyelamat k untuk berpindah dari insiden i ke insiden j ; jika $i = 0$ maka tim penyelamat k berada di lokasi awal (posko) sebelum pergi ke lokasi insiden j .
$cap_{ki} \in \{0,1\}$	1 jika tim penyelamat k mampu mengatasi insiden i ; 0 jika sebaliknya.
t_{ik}	waktu kedatangan tim penyelamat k pada lokasi i
$[twa, twb]$	time window yang tersedia

Variabel keputusan

$X_{ij}^k \in \{0,1\}$	1 jika insiden i diproses oleh tim penyelamat k segera sebelum memproses insiden j ; 0 jika sebaliknya.
$Y_{ij}^k \in \{0,1\}$	1 jika insiden i diproses oleh tim penyelamat k kapanpun sebelum memproses insiden j ; 0 jika sebaliknya.

Fungsi tujuan

$$\min z = \sum_{j=1}^n \left(T \sum_{i=0}^n \sum_{k=1}^m \left[C_i^k p_i^k Y_{ij}^k + (C_j^k p_j^k + U) X_{ij}^k + Y_{ij}^k \left(\sum_{l=0}^n X_{li}^k V \right) \right] \right) \quad (O)$$

dengan:

$$T = 0,3lw_j + 0,4mw_j + 0,3uw_j \quad (Oa)$$

$$U = 0,3ls_{ij}^k + 0,5ms_{ij}^k + 0,2us_{ij}^k \quad (Ob)$$

$$V = 0,3ls_{li}^k + 0,5ms_{li}^k + 0,2us_{li}^k \quad (Oc)$$

Fungsi tujuan dalam model ini adalah untuk meminimumkan nilai *completion time*. Adapun komponen waktu dari *completion time* ini terdiri atas waktu operasional tim penyelamat dan waktu perjalanan antar lokasi tim penyelamat.

Batasan

$$\sum_{i=0}^n \sum_{k=1}^m X_{ij}^k = 1, \quad j = 1, \dots, n \quad (C1)$$

Batasan C1 menyatakan bahwa lokasi insiden j pasti berasal dari salah satu lokasi insiden i yang ditangani oleh salah satu tim k .

$$\sum_{j=1}^{n+1} \sum_{k=1}^m X_{ij}^k = 1, \quad i = 1, \dots, n \quad (C2)$$

Batasan C2 menyatakan bahwa lokasi insiden i pasti menuju ke salah satu lokasi insiden j yang ditangani oleh salah satu tim k .

$$\sum_{j=1}^{n+1} X_{0j}^k = 1, \quad k = 1, \dots, m \quad (C3)$$

$$\sum_{i=0}^n X_{i,n+1}^k = 1, \quad k = 1, \dots, m \quad (C4)$$

Batasan C3 dan C4 digunakan untuk memastikan bahwa setiap tim penyelamat akan memulai pencarian dari posko dan akan kembali ke posko juga setelah menyelesaikan tugasnya.

$$Y_{il}^k + Y_{ij}^k - 1 \leq Y_{ij}^k, \quad i = 0, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m; l = 1, \dots, n \quad (C5)$$

Batasan C5 menunjukkan urutan bahwa i pasti mendahului j .

$$\sum_{i=0}^n X_{il}^k = \sum_{j=1}^{n+1} X_{ij}^k, \quad l = 1, \dots, n; k = 1, \dots, m \quad (C6)$$

Batasan C6 digunakan untuk memastikan bahwa tim penyelamat k akan pergi ke suatu titik lokasi yang lain atau kembali.

$$X_{ij}^k \leq Y_{ij}^k, \quad i = 0, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m \quad (C7)$$

Batasan C7 digunakan untuk memastikan bahwa apabila suatu titik lokasi langsung ditangani oleh tim penyelamat k , maka titik tersebut berada pasti ada di sebelumnya.

$$Y_{ii}^k = 0, \quad i = 0, \dots, n+1; k = 1, \dots, m \quad (C8)$$

Batasan C8 digunakan untuk memastikan agar tim penyelamat tidak *stuck* di suatu lokasi dan dapat berpindah ke lokasi yang lain.

$$Y_{ij}^k = cap_{ki}, \quad i = 1, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m \quad (C9)$$

Batasan C9 digunakan untuk memastikan apabila tim penyelamat k memiliki kapabilitas yang cukup untuk memproses insiden i , maka tim penyelamat k tersebut dapat memprosesnya.

$$\sum_{l=1}^{n+1} X_{il}^k \geq Y_{ij}^k, \quad i = 0, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m \quad (C10)$$

$$\sum_{l=0}^n X_{lj}^k \geq Y_{ij}^k, \quad i = 0, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m \quad (C11)$$

Batasan C10 dan C11 digunakan untuk memastikan Y_{ijk} akan bernilai 0 jika tim penyelamat k tidak memproses insiden i sebelum insiden j .

$$X_{ij}^k, Y_{ij}^k \in \{0, 1\}, \quad i = 0, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m \quad (C12)$$

Batasan C12 digunakan untuk memastikan bahwa nilai X_{ij}^k dan Y_{ij}^k adalah biner.

$$t_{jk} \geq t_{ik} + X_{ij}^k p_i^k + X_{ij}^k (0,3ls_{ij}^k + 0,5ms_{ij}^k + 0,2us_{ij}^k), \\ i = 0, \dots, n; j = 1, \dots, n+1; k = 1, \dots, m \quad (C13)$$

Batasan C13 ini menyatakan bahwa waktu tim penyelamat datang pada lokasi selanjutnya adalah setelah dari waktu tim penyelamat menangani lokasi sebelumnya yang terdiri atas waktu tim penyelamat datang pada lokasi sebelumnya ditambah dengan waktu operasional tim penyelamat pada lokasi sebelumnya dan waktu perjalanan tim penyelamat dari lokasi sebelumnya ke selanjutnya.

$$twa \leq t_{n+1,k} \leq twb, \quad k = 1, \dots, m \quad (C14)$$

Batasan C14 ini digunakan untuk menjamin agar waktu tim penyelamat kembali ke posko tidak berada di luar kurun waktu yang sudah ditentukan. Dengan demikian, keselamatan tim penyelamat tetap dapat terjamin.

Perubahan pada model matematis mengakibatkan perlu adanya penyesuaian pada langkah-langkah GRASP Metaheuristics-nya. Berikut ini adalah *pseudocode* secara keseluruhan metode usulan GRASP *Metaheuristics* yang baru.

```

METODE USULAN GRASP METAHEURISTICS
Input parameter & inialisasi variabel
while iter < maxiter
Update  $S \leftarrow \theta$ 
Hitung completion time ( $C$ )
while  $i > 0$ 
Hitung  $C_{min} \leftarrow \min \{c|c \in C\}$  dan  $C_{max} \leftarrow \max \{c|c \in C\}$ 
Cek RCL  $\leftarrow \{c \in C | c \leq C_{min} + \alpha(C_{max} - C_{min})\}$ 
Pilih random nilai  $c \in$  RCL beserta indeks lokasi dan timnya
Lakukan construction heuristics
Update  $C \leftarrow C \setminus \{i^*\}$ 
end while
Simpan insiden terpilih pada jadwal
Lakukan improvement heuristics dari jadwal yang terbentuk
Update jadwal & hitung completion time-nya
if  $twa \leq completiontime \leq twb$ 
Simpan completion time beserta jadwalnya
end if
end while
Pilih completion time terkecil beserta jadwalnya
    
```

Gambar 1. *Pseudocode* Metode Usulan

UJI COBA MODEL USULAN

Berdasarkan skenario yang sudah dibuat ditambah dengan data baru berupa *time window* mulai dari pukul 07.00-17.00, maka dilakukanlah uji coba model usulan. Berikut ini adalah hasil uji coba model dalam bentuk penjadwalan tiap tim.

Tabel 3. Jadwal Tiap Tim Penyelamat

Tim	Mulai	Lokasi Tujuan	A menit	<i>Arrive</i>	B menit	<i>Depart</i>	Selesai	C menit
1	07.00	1. Pronojiwo	74	08.14	86	09.40	15.15	495
		2. Sumberrejo	36	10.16	110	12.06		
		3. Gesang	28	12.34	42	13.16		
		4. Nguter	8	13.24	69	14.33		
		5. Posko	42	15.15	-			
2	07.00	1. Jogosari	85	08.25	112	10.17	12.39	339
		2. Sememu	38	10.55	73	12.08		
		3. Posko	31	12.39	-			
3	07.00	1. Sumbermujur	67	08.07	75	09.22	14.15	435
		2. Sumber Urip	38	10.00	67	11.07		
		3. Kloposawit	46	11.53	90	13.23		
		4. Posko	52	14.15	-			
4	07.00	1. Kertosari	50	07.50	116	09.46	12.43	343
		2. Penanggal	22	10.08	100	11.48		
		3. Posko	55	12.43	-			
5	07.00	1. Jatisari	27	07.27	62	08.29	08.56	116

Tim	Mulai	Lokasi Tujuan	A menit	Arrive	B menit	Depart	Selesai	C menit
		2. Posko	27	08.56	-			
6	07.00	1. Supit Urang	85	08.25	68	09.33	14.18	438
		2. Oro-Oro Ombo	20	09.53	73	11.06		
		3. Pasrujambe	59	12.05	64	13.09		
		4. Posko	69	14.18	-			
7	07.00	1. Sumberwuluh	67	08.07	99	09.46	12.32	332
		2. Gondoruso	45	10.31	59	11.30		
		3. Posko	62	12.32	-			

A=Waktu Perjalanan, B=Waktu Operasional, C=Total Time

Dari tabel di atas, dapat diketahui bahwa *completion time* model usulan ini adalah 495 menit (8 jam 15 menit). Sedangkan, waktu komputasi untuk mendapatkan hasil tersebut adalah selama 12,56 detik. Pada uji coba model usulan ini, iterasi dilakukan sebanyak 300 kali untuk bisa mendapatkan hasil. Dari uji validasi, hasil model usulan ini juga valid. Batasan *time window* memungkinkan solusi yang didapat agar tetap berada di dalam kurun waktu yang ditentukan. Seperti yang diketahui bahwa setiap iterasi yang dilakukan akan mengeluarkan satu nilai hasil. Apabila nilai hasil tersebut berada di dalam kurun waktu yang sudah ditentukan, maka nilai hasil tersebut akan disimpan. Sedangkan, apabila nilai hasil tersebut berada di luar kurun waktu yang ditentukan, maka nilai hasil tersebut tidak akan disimpan dan program akan terus berjalan sesuai dengan iterasi yang diinginkan. Batasan *time window* tersebut membuat *range* solusi yang didapat menjadi lebih sempit lagi. Hal ini ternyata mempengaruhi jumlah iterasi yang harus digunakan. *Running* model usulan ini tidak menghasilkan solusi (*not feasible*) ketika model hanya dijalankan dengan iterasi sebanyak 100 kali. Penambahan jumlah iterasi menjadi diperlukan agar didapatkan solusi. Pada kasus ini, iterasi ditambah menjadi sebanyak 300 kali untuk mendapatkan solusi. Batasan *time window* mampu memaksa model memerlukan iterasi lebih banyak agar solusi yang didapatkan adalah solusi yang paling optimal. Oleh karena itu, pada uji coba model awal yang awalnya dilakukan sebanyak 100 iterasi saja diubah menjadi sebanyak 300 iterasi untuk perbandingan yang adil.

Tabel 4. Perbandingan Hasil Model Awal Dengan Model Usulan

	Completion Time	Selisih	% Perbedaan
Model awal	528 menit	33 menit	$\frac{33}{495} \times 100\% = 6,25\%$
Model usulan	495 menit		

Seperti yang dapat dilihat pada Tabel 4, hasil model awal yang dilakukan dengan 300 iterasi terhadap model usulan menghasilkan persentase perbedaan sebesar 6,25%. Model awal ternyata juga mampu menghasilkan solusi yang hampir optimal seperti model usulan, tetapi dengan catatan perlu dilakukan penambahan iterasi menjadi sebanyak 300 kali. Tanpa adanya batasan *time window*, maka akan sulit untuk mengetahui seberapa banyak jumlah iterasi yang diperlukan untuk menghasilkan solusi yang lebih optimal. Hal ini terbukti ketika model awal dilakukan dengan 100 iterasi, solusi yang didapat kurang optimal. Terdapat perbedaan hasil sebesar 16,72% dari uji coba model awal yang dilakukan sebanyak 100 iterasi dan 300 iterasi seperti yang sudah dijelaskan sebelumnya. Oleh karena itu, peran dari *time window* sangat penting untuk diterapkan dalam model ini. Pertama, adalah karena batasan *time window* dapat menjamin keselamatan dari tim penyelamat yang sedang bertugas. Kedua, adalah karena batasan *time window* mampu memaksa model memerlukan iterasi yang lebih banyak sehingga solusi yang dihasilkan menjadi lebih optimal.

Selain itu, dilihat dari segi *load balancing* tim, apabila dibandingkan dengan hasil uji coba model awal, hasilnya tidak jauh berbeda. Beban kerja tiap tim masih belum merata. Hal ini nampak dari jam kembali ke posko tiap tim yang jauh berbeda satu sama lain. Seperti yang sudah dijelaskan sebelumnya, kapabilitas tim yang rendah adalah penyebab dari hal ini. Nilai kapabilitas tim yang rendah mempengaruhi *load balancing heuristics* yang dipakai. Tim dengan kapabilitas rendah hanya dapat menangani sedikit insiden sehingga mereka akan kembali ke posko lebih cepat ketimbang tim lainnya dengan kapabilitas tinggi. Hal ini dapat diatasi apabila kapabilitas tiap tim bernilai sama-sama tinggi sehingga tiap tim dapat menangani lebih banyak insiden dan beban kerja dapat menjadi lebih seimbang.

ANALISIS SENSITIVITAS

Menurut Hillier & Lieberman (2001), analisis sensitivitas dilakukan untuk menyelidiki seberapa besar efek atau pengaruh perubahan parameter tertentu terhadap nilai dari solusi optimal (fungsi tujuan) yang dihasilkan. Suatu fungsi tujuan dapat dikatakan sensitif terhadap suatu parameter apabila persentase

perubahan nilai fungsi tujuan lebih besar dari persentase perubahan parameter tersebut. Analisis sensitivitas ini menjadi hal yang sangat penting mengingat terkadang banyak sekali parameter yang merupakan hasil estimasi. Adanya perubahan terhadap parameter tertentu akan mempengaruhi seberapa jauh *optimality* maupun *feasibility* dari solusi optimal yang dihasilkan. Analisis sensitivitas dilakukan dengan cara mengubah nilai dari suatu parameter. Parameter pertama yang akan diuji cobakan adalah waktu operasional tim. Waktu operasional tim bisa berubah sewaktu-waktu karena berbagai hal sehingga waktu operasionalnya bisa menjadi lebih cepat atau lebih lambat. Lebih cepat apabila ternyata tim dapat menemukan korban kurang dari waktu yang diperkirakan. Lebih lambat apabila tim menemui kesulitan di medan tertentu yang ternyata di luar perkiraan sebelumnya. Parameter kedua yang dipilih adalah kapabilitas masing-masing tim. Parameter ini dipilih untuk melihat seberapa besar kapabilitas tim berpengaruh terhadap *completion time*, juga ingin membuktikan apakah benar kapabilitas tim juga berpengaruh terhadap keseimbangan beban kerja yang ada.

Tabel 5. Analisis Hasil Perubahan Fungsi Tujuan Terhadap Perubahan Parameter Waktu Operasional Tim

% Perubahan Parameter	% Perubahan Fungsi Tujuan	Total Waktu (menit)
20%	19,79%	593
10%	2,63%	508
0%	0%	495
-10%	-8,89%	451
-20%	-10,30%	444

Dari hasil yang diperoleh, didapati bahwa persentase perubahan fungsi tujuan tidak lebih besar ketimbang perubahan parameter waktu operasional tim. Oleh karena itu, dapat disimpulkan bahwa fungsi tujuan tidak sensitif terhadap perubahan parameter waktu operasional tim. Selain itu, dapat diketahui bahwa hubungan antara *completion time* terhadap perubahan waktu operasional tim adalah berbanding lurus. Semakin tinggi perubahan waktu operasional tim, maka semakin tinggi pula *completion time* yang dihasilkan. Begitu pula sebaliknya.

Tabel 6. Jam Kembali ke Posko Masing-Masing Tim Pada Setiap Perubahan Waktu Operasional Tim

Tim	Jam Kembali ke Posko				
	Turun 20%	Turun 10%	Normal	Naik 10%	Naik 20%
Tim 1	09.17	10.20	15.15	15.20	16.02
Tim 2	12.57	09.08	12.39	12.57	09.38
Tim 3	13.51	13.52	14.15	15.19	16.53

Tim	Jam Kembali ke Posko				
	Turun 20%	Turun 10%	Normal	Naik 10%	Naik 20%
Tim 4	10.19	14.31	12.43	10.48	12.23
Tim 5	08.57	08.49	08.56	09.04	09.10
Tim 6	14.24	13.41	14.18	14.15	15.13
Tim 7	13.50	14.16	12.32	15.28	15.59

Melalui Tabel 6, dapat dilihat sekali lagi bahwa terjadi ketidakseimbangan beban kerja setiap tim. Tim yang memiliki nilai kapabilitas rendah cenderung kembali ke posko lebih awal karena hanya menangani sedikit lokasi insiden. Sedangkan, tim dengan nilai kapabilitas tinggi cenderung kembali ke posko lebih lama karena harus menangani banyak lokasi insiden.

Tabel 7. Analisis Hasil Perubahan Fungsi Tujuan Terhadap Perubahan Parameter Kapabilitas Tim

% Perubahan Parameter	% Perubahan Fungsi Tujuan	Total Waktu (menit)
10%	-16,97%	411
5%	-15,35%	419
0%	0%	495
-5%	5,66%	523
-10%	31,72%	652

Berdasarkan Tabel 7, dapat diketahui bahwa persentase perubahan fungsi tujuan lebih besar terhadap persentase perubahan parameter kapabilitas tim. Angka negatif yang terjadi hanya menunjukkan bahwa fungsi tujuan mengalami penurunan. Oleh karena itu, dapat disimpulkan bahwa fungsi tujuan sensitif terhadap perubahan parameter kapabilitas tim. Selain itu, dapat diketahui bahwa hubungan antara *completion time* terhadap perubahan kapabilitas tim adalah berbanding terbalik. Semakin tinggi perubahan kapabilitas tim, maka semakin rendah *completion time* yang dihasilkan. Begitu pula sebaliknya.

Tabel 8. Jam Kembali ke Posko Masing-Masing Tim Pada Setiap Perubahan Kapabilitas Tim

Tim	Jam Kembali ke Posko				
	Turun 10%	Turun 5%	Normal	Naik 5%	Naik 10%
Tim 1	14.39	14.22	15.15	13.37	12.40
Tim 2	09.40	10.56	12.39	13.54	10.47
Tim 3	14.48	15.40	14.15	13.23	13.14
Tim 4	11.46	12.26	12.43	11.34	12.21
Tim 5	09.15	09.04	08.56	08.49	12.43
Tim 6	17.52	14.30	14.18	13.59	13.51
Tim 7	15.43	15.43	12.32	12.23	12.04

Tabel 8 sekali lagi membuktikan bahwa kapabilitas tim sangat mempengaruhi beban kerja tim yang dalam hal ini nampak dari jam kembali tiap tim ke posko. Semakin menurun kapabilitas suatu tim, maka tim tersebut tidak dapat menangani lokasi insiden lebih banyak lagi karena keterbatasan kemampuannya. Akibatnya,

tim tersebut hanya bisa menangani sedikit insiden dan pulang lebih awal ke posko. Begitu pula sebaliknya. Semakin meningkat kapabilitas suatu tim, maka tim tersebut bisa menangani lokasi insiden lebih banyak lagi karena ia mampu mengatasi insiden itu. Dengan demikian, beban kerja setiap tim dapat menjadi lebih seimbang satu sama lain dan *completion time* pun menjadi lebih cepat.

KESIMPULAN DAN SARAN

Kesimpulan pertama adalah model RUASP dikembangkan dengan menambahkan 2 batasan terkait *time window* dan penyesuaian pada langkah-langkah GRASP *Metaheuristics*-nya. *Running* model usulan dilakukan sebanyak 300 iterasi untuk mendapatkan hasil berupa *completion time* selama 8 jam 15 menit dengan waktu komputasi selama 12,56 detik. Batasan *time window* memaksa model memerlukan iterasi lebih banyak sehingga solusi yang didapatkan adalah solusi yang paling optimal. Oleh karena itu, pada uji coba model awal yang awalnya dilakukan sebanyak 100 iterasi diuji kembali dengan iterasi sebanyak 300 kali juga untuk perbandingan yang adil. Hasil yang didapatkan memiliki selisih 6,25% lebih lama dibandingkan dengan hasil model usulan. Model awal ternyata juga mampu menghasilkan solusi optimal yang sama dengan model usulan, dengan catatan perlu dilakukan penambahan iterasi menjadi sebanyak 300 kali. Tanpa adanya batasan *time window*, maka akan sulit untuk mengetahui seberapa banyak jumlah iterasi yang diperlukan untuk menghasilkan solusi yang lebih optimal. Hal ini terbukti ketika model awal dilakukan dengan 100 iterasi, solusi yang didapat kurang optimal. Terdapat perbedaan hasil sebesar 16,72% dari uji coba model awal yang dilakukan sebanyak 100 iterasi dan 300 iterasi. Oleh karena itu, peran dari batasan *time window* dapat dikatakan sangatlah penting karena membantu untuk memperoleh solusi yang paling optimal. Selain itu, dari uji validitas yang dilakukan dapat disimpulkan bahwa model ini juga valid.

Kesimpulan kedua didapat melalui analisis sensitivitas yang menunjukkan bahwa fungsi tujuan ternyata tidak sensitif terhadap perubahan waktu operasional tim, namun sensitif terhadap perubahan kapabilitas tim. Selain itu, didapati temuan lain pula yang menunjukkan bahwa nilai kapabilitas tim ternyata sangat mempengaruhi keseimbangan beban kerja tim. Dengan nilai kapabilitas tim yang

tinggi, maka tim tersebut dapat menangani banyak insiden sehingga beban kerja tiap tim dapat menjadi seimbang. Begitu pula sebaliknya.

Saran untuk penelitian ini lebih mengarah pada aplikasi/penerapan model. Pihak BPBD Lumajang maupun pihak-pihak terkait lainnya sebaiknya perlu untuk memperhatikan kemampuan dari tim penyelamat yang ada. Semakin tangguh tim yang dibentuk maka akan semakin baik. Semua tim penyelamat yang dipersiapkan harusnya memiliki *skill* yang sama-sama tinggi satu sama lain (*skill* yang tinggi ini diterjemahkan menjadi nilai kapabilitas tim yang tinggi juga dalam model). Dengan demikian, beban kerja tiap tim penyelamat dapat menjadi seimbang dan waktu evakuasi pun dapat menjadi lebih cepat.

DAFTAR PUSTAKA

- Altay, N. & Green, W. G., 2006. OR/MS research in disaster operations management. *European Journal of Operational Research*, Vol. 175, pp. 475-493.
- BNPB, 2017. *Definisi dan Potensi*. [Online]. Diperoleh 6 April 2017 dari: <https://www.bnpb.go.id/home>.
- BPBD Lumajang, 2015. *Rencana Kontinjensi Gunung Semeru*.
- Cordeau *et al.*, 2002. VRP with Time Windows dalam Paolo Toth & Daniele Vigo (Eds.). *The Vehicle Routing Problem*, pp.157-186.
- Hillier, F. & Lieberman, G., 2001. *Introduction to Operations Research Seventh Edition*. New York: McGraw-Hill.
- Rizqi, D., 2016. *Pengembangan Model Rescue Unit Assignment and Scheduling Problem Dalam Penanggulangan Terhadap Bencana Alam Dengan Metode GRASP Metaheuristics*. Surabaya: Universitas Surabaya.
- Undang-Undang Republik Indonesia Nomor 24 Tahun 2007 Tentang Penanggulangan Bencana.
- Wex, F. *et al.*, 2014. Emergency response in natural disaster management: Allocating and scheduling of rescue units. *European Journal of Operation Research*, 235, pp.697-708.