

## STUDI ALGORITMA IGVNS, ALGORITMA GVNS, DAN ALGORITMA ABC PADA MULTIPLE TRIP VEHICLE ROUTING PROBLEM (MTVRP)

Nurul Faridhatul Aini<sup>1</sup>, Vita Kusumasari\*<sup>1</sup>, Desi Rahmadani<sup>1</sup>

<sup>1</sup>Jurusan Matematika, FMIPA, Universitas Negeri Malang

Email : nurul.faridhatul.1703126@students.um.ac.id<sup>1</sup> (NF. Aini),

vita.kusumasari.fmipa@um.ac.id<sup>2</sup> (V. Kusumasari), desi.rahmadani.fmipa@um.ac.id<sup>3</sup> (D. Rahmadani)

\*Corresponding Author

### Abstrak

The Multiple Trip Vehicle Routing Problem (MTVRP) is one of the VRP variants with vehicle capacity constraints, and the limited number of vehicles allows each vehicle to distribute more than one route so as to minimize the number of vehicles used. The algorithm used in this study is the Improved General Variable Neighborhood Search (IGVNS) algorithm. The IGVNS algorithm is a hybrid between the GVNS algorithm and the VNS algorithm. The calculation results of the IGVNS algorithm will be compared with the GVNS algorithm and the ABC algorithm. The main stages of the IGVNS and GVNS algorithms are initial solution formation, perturbation, and solution improvement. The main stages of the ABC algorithm are initialization, solution improvement and optimization. Based on manual calculations using 8 points, the ABC algorithm produces a distance of 86 km and a service time of 1.47 hours, the GVNS algorithm produces a distance of 80 km and a service time of 1.37 hours, the IGVNS algorithm produces a distance of 79 km and service time is 1.35 hours. Based on one example of the ABC algorithm, the calculation solution using the IGVNS algorithm shows more optimal results. Based on the results of parameter testing, the  $v_{max}$  parameter affects the calculation results, that is the greater the  $v_{max}$  value, the more optimal the resulting solution. While the  $\alpha_{max}$  parameter does not affect the calculation results because it shows constant results in two successive iterations carried out.

**Keywords:** Improved General Variable Neighborhood Search, Multiple Trip Vehicle Routing Problem (MTVRP)

Submitted: 18 Juni 2021; Revised: 20 Juli 2021; Accepted Publication: 29 Juli 2021;

Published Online: July 2021

DOI:10.17977/um055v2i2p26-31

### PENDAHULUAN

Dalam dunia bisnis, kegiatan distribusi memiliki peran yang sangat penting. Pola pendistribusian yang tidak dilakukan secara tepat, dapat menghabiskan biaya yang tinggi dan mengakibatkan pemborosan dalam segi waktu, jarak dan tenaga. Permasalahan distribusi dapat diselesaikan dengan salah satu cabang ilmu matematika yaitu teori *graph* khususnya pada materi *Vehicle Routing Problem* (VRP).

Permasalahan VRP didefinisikan sebagai upaya pencarian solusi yang meliputi penentuan sejumlah rute, di mana setiap rute dilayani oleh satu kendaraan yang berawal dan berakhir di tempat yang sama, sehingga permintaan atau kebutuhan semua *customer* dapat terpenuhi dengan syarat tidak melanggar kendala-kendala yang ada dan untuk meminimalkan total biaya transportasi (Laporte, 1992). Salah satu varian VRP yaitu *Multiple Trip Vehicle Routing Problem* (MTVRP) di mana setiap kendaraan memiliki beberapa rute dalam suatu periode waktu atau dengan kata lain kendaraan dapat melakukan beberapa kali perjalanan selama jam kerja kendaraan berlangsung (Cheikh et al., 2015), sehingga berdasarkan batasan tersebut dapat meminimalkan jumlah kendaraan yang digunakan. Menurut Cattaruzza, dkk. (2014), tujuan MTVRP adalah menentukan rute dan penugasan setiap kendaraan untuk pelayanan masing-masing rute sedemikian sehingga menghasilkan total waktu pelayanan minimal dan memenuhi kondisi bahwa setiap rute berawal dan berakhir pada satu depot, setiap pelanggan dalam rute dikunjungi tepat satu kali, jumlah permintaan *customer* dari satu rute

tidak melebihi kapasitas maksimum kendaraan, dan total waktu pelayanan untuk setiap rute tidak melebihi batasan waktu kerja kendaraan.

Permasalahan MTVRP telah banyak diselesaikan dengan berbagai macam algoritma, salah satunya algoritma *Large Neighborhood Search* yang diteliti oleh François et al (2016). Penelitian yang dilakukan oleh Ren (2020) menjelaskan bahwa algoritma IGVNS merupakan algoritma metaheuristik yang didasarkan pada penggabungan strategi antara *Variable Neighborhood Search* (VNS) dan *General Variable Neighborhood Search* (GVNS). Penelitian yang berkaitan dengan algoritma GVNS yang dilakukan oleh Bezerra et al (2018), dapat dipahami bahwa algoritma GVNS merupakan salah satu varian dari algoritma VNS di mana tahap *local search* menggunakan algoritma heuristik yaitu *Variabel Neighborhood Descent* (VND). Algoritma VNS memiliki dua tahapan utama yakni terdiri dari tahap *shaking* dan tahap perbaikan. Berdasarkan penelitian oleh Ranjbar & Saber (2021), tahap perbaikan dalam VNS ini menggunakan *Variable Neighborhood Descent* (VND) yang bertujuan untuk memperbaiki solusi awal dengan cara mengeksplorasi beberapa struktur lingkungan.

Berdasarkan uraian di atas, penelitian ini akan menerapkan algoritma *general variable neighborhood search* dan *improved general variable neighborhood search* pada permasalahan yang berbeda yaitu *Multiple Trip Vehicle Routing Problem* (MTVRP). Data yang digunakan pada penelitian ini diperoleh berdasarkan penelitian yang dilakukan oleh Ernanti (2017) dengan menggunakan algoritma ABC, solusi berdasarkan penelitian tersebut akan dibandingkan dengan algoritma *general variable neighborhood search* dan *improved general variable neighborhood search*.

## METODE

Algoritma yang digunakan pada penelitian ini yaitu algoritma *Improved General Variable Neighborhood Search* (IGVNS), data pada penelitian ini diperoleh berdasarkan penelitian terdahulu oleh Ernanti (2017). Hasil perhitungan dengan algoritma IGVNS selanjutnya akan dibandingkan dengan solusi hasil perhitungan menggunakan algoritma ABC dan algoritma GVNS.

## HASIL DAN PEMBAHASAN

### *Improved General Variable Neighborhood Search* (IGVNS)

GVNS adalah salah satu varian dari algoritma VNS di mana tahap *local search* menggunakan algoritma heuristik yaitu *Variabel Neighborhood Descent* (VND) (Venkatesh et al., 2018). IGVNS dikembangkan dengan mengintegrasikan *Basic VNS* (BVNS) dan *General VNS* (GVNS), di mana strategi hibrid menggabungkan strategi peningkatan pertama dari BVNS dan strategi tetangga terbaik dari GVNS untuk mengontrol proses pencarian lokal. Tahapan-tahapan utama pada algoritma IGVNS untuk menyelesaikan permasalahan MTVRP adalah sebagai berikut:

#### 1. Pembentukan Solusi Awal

Pembentukan solusi awal menggunakan alat bantu *Microsoft Excel* dalam melakukan pengacakan urutan pelanggan yang akan dilayani, kemudian menerapkan metode orientasi pelanggan. Solusi tersebut didefinisikan dengan  $x = \{x_1, x_2\}$ , dengan  $x_1 = \{x_1^1, \dots, x_1^i, \dots, x_1^n\}$  merupakan urutan pelanggan,  $x_2 = \{x_2^1, \dots, x_2^i, \dots, x_2^n\}$  merupakan array yang terdiri dari  $n$  bilangan biner, dengan  $n$  menyatakan jumlah *customer*. Jika  $x_1^i$  merupakan *customer* pertama dalam suatu rute kendaraan, maka  $x_2^i = 1$  dan  $x_2^i = 0$  untuk lainnya. Untuk mengingat *customer* yang tersedia yang dapat dikunjungi oleh kendaraan dinotasikan dengan himpunan  $E$ . Langkah-langkah pembentukan solusi awal dengan menggunakan alat bantu *Microsoft Excel* dan metode orientasi *customer* adalah sebagai berikut.

- a. Melakukan pengacakan pada himpunan  $x_1$  dengan menggunakan alat bantu *Microsoft Excel* dan menerapkan metode orientasi *customer*.
- b. Inisialisasi nilai  $i = 1$  dan tentukan kendaraan pertama yang akan digunakan ( $k = 1$ ).

- c. Bentuk rute kendaraan berdasarkan jumlah permintaan *customer* dengan syarat tidak melanggar kapasitas kendaraan dan total waktu pelayanan pada setiap rute ( $t_r$ ) tidak diperbolehkan melebihi waktu kerja kendaraan. Jumlah total waktu pelayanan pada setiap rute diperoleh berdasarkan waktu tempuh dan waktu pelayanan pada setiap *customer*.
- Jika total permintaan *customer* pada suatu rute tidak melebihi kapasitas kendaraan dan jumlah total waktu pelayanan pada setiap rute ( $t_r$ ) tidak melanggar waktu kerja kendaraan, maka *customer* tersebut akan dilayani pada rute yang sama dan himpunan  $E$  akan berkurang satu. Himpunan  $E$  akan terus berkurang apabila setiap *customer* masuk dalam suatu rute.
  - Jika total waktu pelayanan pada setiap rute ( $T_{total}$ ) tidak melanggar waktu kerja kendaraan tetapi total permintaan *customer* melebihi kapasitas kendaraan, maka *customer* tersebut akan dilayani pada rute selanjutnya. Himpunan  $E$  tetap.
  - Jika total permintaan *customer* tidak melebihi kapasitas kendaraan tetapi total waktu pelayanan pada setiap rute ( $T_{total}$ ) melanggar waktu kerja kendaraan, maka harus menghapus *customer* dari rute tersebut dimulai dari *customer* terakhir pada rute tersebut dan *customer* yang dihapus akan dilayani pada rute selanjutnya. Himpunan  $E$  tetap.
- d. Pilih *customer* pada  $x_1^i$  dan cek apakah *customer* tersebut merupakan *customer* pertama pada rute kendaraan. Jika *customer* pertama, maka tetapkan  $x_2^i = 1$ , jika tidak maka tetapkan  $x_2^i = 0$ . Lakukan proses ini hingga mencapai  $i \leq n$  dan berakhir hingga himpunan  $E$  kosong.
- e. Jika semua *customer* telah masuk dalam setiap rute dan setiap rute telah dilayani oleh setiap kendaraan. Maka algoritma selesai.

Dalam tahap ini, akan menghasilkan jumlah rute yang diperoleh dan jumlah kendaraan yang akan digunakan.

## 2. Tahap Perturbasi (*Shaking*)

Proses perturbasi pada solusi awal  $x$  dilakukan untuk memperoleh solusi acak. Jika solusi hasil perturbasi menghasilkan solusi yang lebih baik daripada solusi awal, maka lakukan perturbasi kembali dengan *neighborhood* yang sama dan  $a$  yang sama. Tahap perturbasi dilakukan dengan menggunakan 3 operator *neighborhood list* untuk perturbasi yaitu  $N_p(x) : \{N_1: exchange, N_2: forward insertion, N_3: backward insertion\}$ . Di mana  $p = \{1, 2, \dots, \sigma_{max}\}$ , jika  $p = 1$  maka gunakan  $N_1$  yaitu *exchange*. Menurut Ren (2020), pengertian mengenai operator *neighborhood* pada tahap perturbasi algoritma IGVNS adalah sebagai berikut:

- a. *Exchange* : Pilih secara acak dua *customer* di  $x_1$ , dan tukarkan keduanya.
- b. *Forward insertion* : Pilih dua *customer* secara acak dari  $x_1$  dan masukkan *customer* I setelah posisi *customer* II.
- c. *Backward insertion* : Pilih dua *customer* secara acak dari  $x_1$  dan masukkan *customer* I sebelum posisi *customer* II.

## 3. Perbaikan solusi dengan *hybrid strategy*

Dalam proses perbaikan solusi menggunakan algoritma VND terdapat 5 operator *neighborhood* yaitu  $N_u(x) = \{2-opt operator, 3-opt operator, forward insertion of subsequence, backward insertion of subsequence, exchange of subsequence\}$ . Langkah-langkah *hybrid strategy* adalah sebagai berikut :

- a Melakukan perbaikan solusi  $x_p$  dengan VND untuk mendapatkan solusi baru  $x_u'$ . Dimulai dengan menetapkan nilai  $v = 1$ , dengan  $v$  merupakan siklus yang terus diulang pada proses VND
- b Apabila  $v \leq v_{max}$ , lakukan proses *local search* dimulai dengan menerapkan operator pertama pada proses *local search* yaitu  $u = 1$  dan ganti  $x_p$  menjadi  $x_p'$ .

- c Proses *local search* diterapkan hingga mencapai  $u \leq \tau_{max}$ , ganti  $x_p'$  menjadi  $x'$ . Proses *local search* dengan operator tertentu ( $x'_u$ ) akan berhenti ketika menghasilkan solusi yang lebih baik, dan kemudian menerapkan operator selanjutnya. Jika solusi  $x'_u$  menghasilkan solusi yang lebih baik dibandingkan dengan solusi  $x'$ , maka ganti  $x'_u$  menjadi  $x'$  dan terapkan operator *neighborhood* selanjutnya hingga mencapai  $u \leq \tau_{max}$ . Jika solusi  $x'_u$  tidak menghasilkan solusi yang lebih baik dibandingkan dengan solusi  $x$ , maka terapkan operator *neighborhood* selanjutnya hingga mencapai  $u \leq \tau_{max}$ .
- d Ketika mencapai  $\tau_{max}$ , maka dilanjutkan pada siklus selanjutnya yaitu  $v = 2$  dan lakukan proses yang sama dari *point* b dan c. Jika telah mencapai  $v_{max}$  lanjutnya pada *point* e.
- e Jika solusi  $x'$  menghasilkan solusi yang lebih baik daripada solusi awal ( $x$ ) maka ganti  $x'$  menjadi  $x$ , tetapkan  $a = 0$  dan terapkan proses perturbasi dengan operator *neighborhood* yang sama yaitu  $p = 1$ . Jika solusi  $x'$  tidak menghasilkan solusi yang lebih baik daripada solusi awal ( $x$ ) maka  $x$  tetap, tetapkan  $a = 1$  dan terapkan proses perturbasi dengan operator *neighborhood* selanjutnya yaitu  $p = 2$ . Proses berlanjut hingga mencapai  $\sigma_{max}$ .
- f Ketika mencapai  $\sigma_{max}$ , cek apakah proses perbaikan solusi memenuhi kondisi  $a < a_{max}$ . Jika memenuhi, maka proses dilanjutkan pada iterasi selanjutnya yaitu  $g = 2$ . Jika tidak, maka algoritma IGVNS berhenti.
- g Algoritma IGVNS akan berhenti ketika mencapai  $a_{max}$  atau telah mencapai iterasi maksimal ( $g_{max}$ ).

**Implementasi Algoritma GVNS dan Algoritma IGVNS pada MTRVP**

Permasalahan *Multiple Trip Vehicle Routing Problem* (MTRVP) dengan algoritma *general variable neighborhood search* dan *improved general variable neighborhood search* diterapkan dalam contoh permasalahan yang diambil dari Skripsi Ernanti (2017). Suatu perusahaan A akan melakukan pemuatan dan pembongkaran barang ke *customer-customer* yang tersebar di berbagai daerah. Pelayanan dilakukan selama 1 jam dengan kapasitas tiap kendaraan sebesar 65 kardus, kecepatan rata-rata 60 km/jam, dan waktu pelayanan tiap kardusnya selama 0,87 detik/kardus atau 0,0002417 jam/kardus. Adapun daftar permintaan tiap-tiap *customer* diberikan pada Tabel 1 dan jarak antar depot dengan *customer* atau antar *customer* akan diberikan pada Tabel 2 sebagai berikut:

**Tabel 1 Permintaan customer**

Customer	1	2	3	4	5	6	7	8
Permintaan	15	20	10	25	15	10	25	20
Waktu	0,22	0,29	0,15	0,36	0,22	0,15	0,36	0,29

**Tabel 2 Jarak antar Depot ke customer dan antar customer**

$C_{ij}$	0	1	2	3	4	5	6	7	8
0	0	5	8	11	16	13	12	9	14
1	5	0	5	7	12	11	15	13	8
2	8	5	0	6	5	3	6	4	11
3	11	7	6	0	11	3	7	14	15
4	16	12	5	11	0	8	6	1	17
5	13	11	3	3	8	0	9	11	10
6	12	15	6	7	6	9	0	14	6
7	9	13	4	14	1	11	14	0	10
8	14	8	11	15	17	10	6	10	0

Penyelesaian permasalahan MTVRP dengan menggunakan algoritma GVNS dimulai dengan menginisialisasi parameter. Pada contoh permasalahan ini menggunakan parameter  $g_{max} = 1$  dan  $k_{max} = 3$ , nilai  $k_{max}$  yang digunakan berdasarkan rujukan artikel pada algoritma GVNS.

Penyelesaian permasalahan MTVRP dengan menggunakan algoritma IGVNS dimulai dengan menginisialisasi parameter. Pada contoh permasalahan ini menggunakan parameter  $g_{max} = 1$ ,  $\sigma_{max} = 3$  dan  $\tau_{max} = 5$  diperoleh berdasarkan rujukan artikel,  $v_{max} = 2$  diperoleh berdasarkan hasil percobaan (pada siklus  $v = 1$  diperoleh solusi yang lebih baik daripada solusi hasil perturbasi, siklus  $v = 2$  juga menghasilkan solusi yang lebih baik dan solusi tersebut menunjukkan total jarak dan total waktu pelayanan paling minimum dibandingkan algoritma ABC dan algoritma GVNS, sehingga jika dilanjutkan pada  $v = 3$  dan tidak menghasilkan solusi yang lebih baik maka algoritma IGVNS akan tetap menjadi algoritma yang lebih optimal diantara ketiga algoritma yang digunakan),  $a_{max} = 2$  juga diperoleh berdasarkan hasil percobaan yang dilakukan (tidak terjadi perbaikan solusi pada dua iterasi yang berurutan yaitu pada iterasi  $a = 1$  dan  $a = 2$  sehingga tidak dilakukan penambahan nilai parameter, selain itu dengan menggunakan parameter  $a_{max} = 3$  maka iterasi tersebut tidak akan melalui proses perturbasi karena jumlah *neighborhood* pada tahap perturbasi akan menunjukkan nilai  $p = 4$  sehingga dalam hal ini akan bertentangan dengan rujukan artikel yang digunakan yang hanya menerapkan 3 *neighborhood* dan tanpa melewati proses perturbasi ini memungkinkan solusi terjebak pada optimum lokal).

Contoh permasalahan di atas juga pernah diselesaikan menggunakan algoritma *Artificial Bee Colony* (ABC) pada penelitian yang dilakukan oleh Ernanti (2017). Perbandingan langkah-langkah dan hasil algoritma *Artificial Bee Colony*, algoritma *General Variable Neighborhood Search* dan algoritma *Improved General Variable Neighborhood* ditunjukkan pada Tabel 3 berikut.

**Tabel 3. Perbandingan Langkah-langkah dan Hasil Algoritma GVNS, ABC dan IGVNS**

<i>General Variable Neighborhood Search</i> (GVNS)	<i>Artificial Bee Colony</i> (ABC)	<i>Improved General Variable Neighborhood Search</i> (IGVNS)
<b>Langkah-langkah :</b> Inisialisasi (pembentukan solusi awal) – Tahap <i>Shaking</i> – Tahap Perbaikan Solusi (algoritma RVND)	<b>Langkah-langkah :</b> Inisialisasi (parameter dan pembentukan solusi awal) – Tahap Perbaikan solusi – Tahap Optimalisasi	<b>Langkah-langkah :</b> Inisialisasi (pembentukan solusi awal) – Tahap <i>Shaking</i> – Tahap Perbaikan Solusi (strategi hibrid)
<b>Kendaraan 1</b> $R_1 : 0 - 3 - 5 - 2 - 0$ $R_2 : 0 - 6 - 4 - 7 - 0$ Waktu pelayanan : 0,91 jam <b>Kendaraan 2 :</b> $R_3 : 0 - 1 - 8 - 0$ Total waktu pelayanan : 1,37 jam Total Rute Kendaraan : 80 km	<b>Kendaraan 1</b> $R_1 : 0 - 6 - 8 - 5 - 3 - 0$ Waktu pelayanan : 0,71 jam <b>Kendaraan 2 :</b> $R_2 : 0 - 7 - 4 - 0$ $R_3 : 0 - 2 - 1 - 0$ Total waktu pelayanan : 1,47 jam Total Rute Kendaraan : 86 km	<b>Kendaraan 1</b> $R_1 : 0 - 2 - 3 - 5 - 8 - 0$ Waktu pelayanan : 0,7 jam <b>Kendaraan 2 :</b> $R_2 : 0 - 7 - 4 - 6 - 0$ $R_3 : 0 - 1 - 0$ Total waktu pelayanan : 1,35 jam Total jarak rute kendaraan : 79 km

Berdasarkan contoh permasalahan diberikan waktu kerja kendaraan selama 1 jam, karena setiap algoritma yang digunakan memerlukan waktu lebih dari 1 jam dan kurang dari 2 jam maka jumlah kendaraan minimal yang digunakan sejumlah dua kendaraan. Berdasarkan tabel di atas, dapat disimpulkan bahwa algoritma IGVNS menghasilkan solusi paling optimal karena menghasilkan total jarak dan waktu pelayanan paling minimum.

## PENUTUP

Algoritma *Improvement General Variable Neighborhood Search* yang diterapkan pada permasalahan MTRVP dengan menggunakan 8 titik menghasilkan total jarak tempuh sebesar 79 km dengan waktu pelayanan sebesar 1,35 jam. *Artificial Bee Colony* menghasilkan total jarak tempuh sebesar 86 km dengan waktu pelayanan sebesar 1,47 jam, sedangkan algoritma *General Variable Neighborhood Search* menghasilkan total jarak tempuh sebesar 80 km dengan waktu pelayanan sebesar 1,37 jam. Berdasarkan satu contoh dari algoritma ABC, dapat disimpulkan bahwa algoritma *Improvement General Variable Neighborhood Search* lebih efektif daripada algoritma *General Variable Neighborhood Search* dan algoritma *Artificial Bee Colony*.

## DAFTAR RUJUKAN

- Bezerra, S.N., de Souza, S.R., Souza, M.J.F., 2018. A GVNS Algorithm for Solving the Multi-Depot Vehicle Routing Problem. *Electronic Notes in Discrete Mathematics*, 66, 167–174. Dari <https://doi.org/10.1016/j.endm.2018.03.022>
- Cattaruzza, D., Absi, N., Feillet, D., Vidal, T., 2014. A Memetic Algorithm for The Multi Trip Vehicle Routing Problem. *European Journal of Operational Research*, 236, 833–848. Dari <https://doi.org/10.1016/j.ejor.2013.06.012>
- Cheikh, M., Ratli, M., Mkaouar, O., Jarboui, B., 2015. A Variable Neighborhood Search Algorithm for The Vehicle Routing Problem with Multiple Trips. *Electronic Notes in Discrete Mathematics*, 47, 277–284. Dari <https://doi.org/10.1016/j.endm.2014.11.036>
- Ernanti, K.A., 2017. *Penerapan algoritma Artificial Bee Colony (ABC) Pada Multiple Trip Vehicle Problem (MTRVP) dan Implementasi Programnya*. Skripsi tidak diterbitkan . Malang: FMIPA UM.
- François, V., Arda, Y., Crama, Y., Laporte, G., 2016. Large Neighborhood Search for Multi-Trip Vehicle Routing. *European Journal of Operational Research*, 255, 422–441. Dari <https://doi.org/10.1016/j.ejor.2016.04.065>.
- Laporte, G., 1992. The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *European Journal of Operational Research*, 59(3), 345–358. Dari [https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C).
- Ranjbar, M., Saber, R.G., 2021. A Variable Neighborhood Search Algorithm for Transshipment Scheduling of Multi Products at A Single Station. *Applied Soft Computing*, 98, 1-8. Dari <https://doi.org/10.1016/j.asoc.2020.106736>
- Ren, Y., 2020. An Improved General Variable Neighborhood Search for A Static Bike-Sharing Rebalancing Problem Considering The Depot Inventory. *Expert System with Applications*, 160. Dari <https://www.sciencedirect.com/science/article/abs/pii/S0957417420305765>