

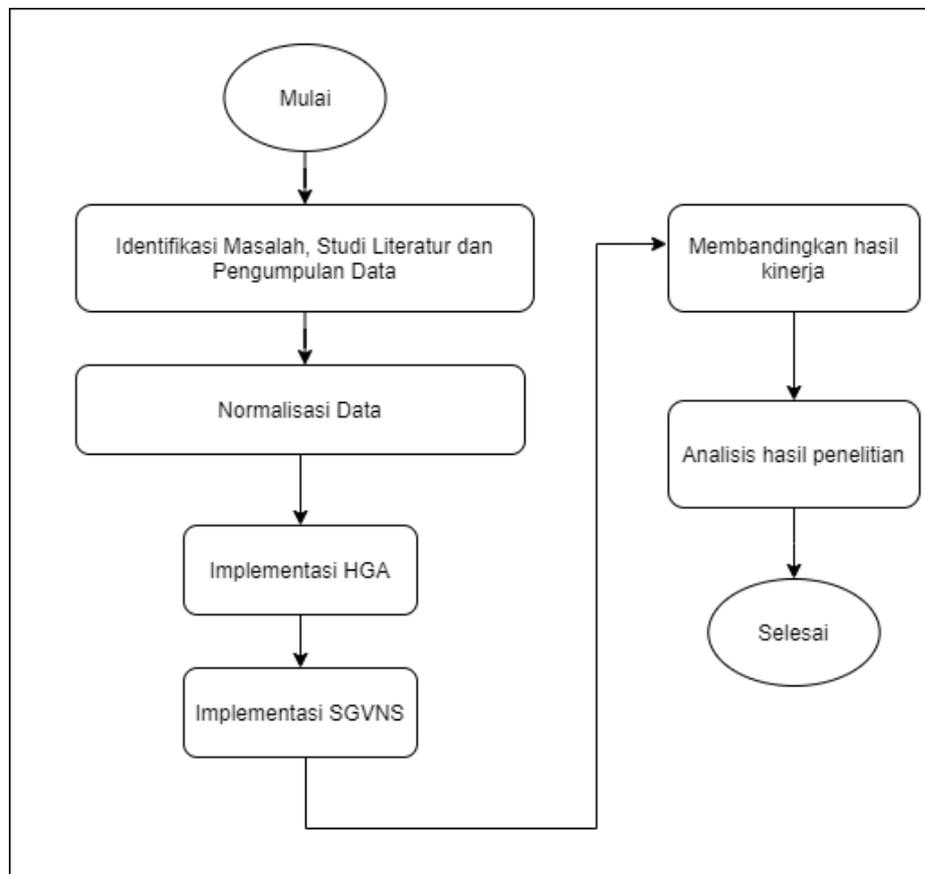
## BAB III METODOLOGI

### 3.1 Kategori Penelitian

Penelitian ini menggunakan metodologi eksperimental dimana akan dilakukan perbandingan dua buah metode atau algoritma terhadap objek yang diteliti. Hasil dari dua buah metode tersebut akan diukur secara kualitatif dan dibandingkan performanya.

### 3.2 Tahapan Penelitian

Berikut adalah langkah-langkah yang dilakukan pada penelitian ini seperti pada Gambar 3.1 yaitu Identifikasi Masalah, Studi Literatur, Pengumpulan Data, Normalisasi Data, Implementasi SGVNS dan HGA, Perbandingan Hasil Kinerja SGVNS dan HGA, dan Analisis Hasil Penelitian.



Gambar 3.1 Tahapan Penelitian

### 3.2.1 Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan data mahasiswa calon penghuni kamar asrama. Pada penelitian ini, data diperoleh dari divisi TIK ITERA dengan spesifikasi seperti pada Tabel 3.1 dan contoh pada Tabel 3.2. Beberapa atribut dari data mahasiswa memiliki kode sebagai penomoran. Atribut yang memiliki kode adalah atribut agama dan wilayah yang contohnya dapat dilihat pada Tabel 3.3 dan Tabel 3.4. Data yang terkumpul berbentuk file teks yang nantinya dijadikan sebagai inputan program. File teks tersebut memuat lima buah data yang dipisahkan oleh titik koma. Berikut adalah urutan penulisan data yaitu dimulai dari NIM, daerah asal mahasiswa, kode program studi, kode golongan UKT, dan terakhir kode agama.

Tabel 3.1 Spesifikasi Dataset

No	Spesifikasi	Keterangan
1	Jumlah data	305 dan 460
2	Tahun data	2018
3	Jumlah atribut data	4
4	Sumber dataset	Divisi TIK ITERA

Tabel 3.2 Contoh Data Mahasiswa

NIM	Nama	Kode Wilayah	Kode Prodi	Gol. UKT	Kode Agama
118220078	Nurintan	022200	PWK	5	1
118190068	Suci Dewi	020500	TI	2	1
118160090	Derlina Sinaga	016300	MA	6	2
118160068	Kadek Melinda Sunari	016400	MA	5	4
118270043	Clara Uli Cartika Simamora	026600	KI	3	3
118110036	Herlina	020600	FI	0	1

Tabel 3.3 Keterangan Kode Agama

ID Agama	Nama Agama
1	Islam
2	Kristen
3	Katholik
4	Hindu
5	Buddha
6	Konghucu
98	Tidak diisi
99	Lainnya

Tabel 3.4 Keterangan Kode Wilayah

Kode Wilayah	Nama Wilayah
000000	Indonesia
010000	Prop. D.K.I. Jakarta
010100	Kab. Kepulauan Seribu
010101	Kec. Kepulauan Seribu Selatan
010102	Kec. Kepulauan Seribu Utara

### 3.2.2 Normalisasi Data

Normalisasi data dilakukan agar nilai atribut berada pada rentang nilai yang sama. Rentang nilai yang digunakan pada penelitian ini adalah 0 sampai 100 untuk semua atribut. Pada studi kasus pengelompokan kamar asrama Institut Teknologi Sumatera mahasiswa akan dibagi kedalam kamar yang ditempati oleh 4 orang mahasiswa. Pengelompokan dilakukan dengan membagi mahasiswa berdasarkan 4 buah atribut mahasiswa yaitu asal daerah, program studi, kemampuan ekonomi, dan agama. Mahasiswa yang berasal dari asal daerah yang dekat pada satu kamar akan menghasilkan jarak atribut yang kecil, semakin jauh jarak antara dua kota asal maka semakin besar jarak atribut nya. Hal ini dibuat karena kota yang dekat biasanya masih mempunyai adat yang mirip, sebaliknya kota yang jauh biasanya mempunyai adat istiadat yang berbeda. Nilai jarak antar wilayah mahasiswa ITERA terbesar adalah 2552 dan akan digunakan sebagai nilai maksimal untuk proses normalisasi.

Nilai atribut program studi dan agama juga di normalisasi. Apabila program studi antar kedua mahasiswa sama maka jarak atribut bernilai 0, sebaliknya jika berbeda maka jarak atribut mendapat nilai maksimal yaitu 100. Begitu pula dengan atribut agama, jarak atribut bernilai 0 apabila mahasiswa memiliki agama yang sama dan bernilai 100 apabila berbeda agama. Untuk menentukan Uang Kuliah Tunggal (UKT) setiap mahasiswa ITERA dibagi menjadi 8 kelompok berdasarkan kemampuan ekonominya. Kelompok 1 sampai dengan 8 memiliki beban pembiayaan yang berbeda. Selain itu ada juga kelompok 0 untuk mahasiswa yang mendapatkan beasiswa tidak mampu dan dikelompokkan diluar kelompok UKT tersebut karena mahasiswa yang penerima beasiswa tidak membayar UKT. Kelompok UKT dan kelompok penerima beasiswa ini dapat digunakan sebagai atribut kemampuan ekonomi pada studi kasus ini.

Normalisasi atribut dapat dilihat pada Tabel 3.5, normalisasi atribut agama dapat dilihat pada Tabel 3.6, untuk atribut wilayah dapat dilihat pada Tabel 3.7, dan untuk UKT dapat dilihat pada Tabel 3.8. Berikut adalah contoh normalisasi data untuk dua mahasiswa dengan data yang ada pada Tabel.3.2 yaitu mahasiswa dengan NIM 118220078 dan 118190068.

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A \quad (24)$$

a. Untuk variabel Program Studi (Prodi)

Aturan normalisasi: apabila prodi berbeda maka jarak yang dihasilkan adalah sama dengan nilai maksimal (100) selain itu maka nilai jarak adalah nol (0).

Sebelum normalisasi:

118220078\_prodi = PWK

118190068\_prodi = TI

Setelah normalisasi maka jarak\_prodi = 100

b. Untuk variabel Agama

Aturan normalisasi: apabila agama berbeda maka jarak yang dihasilkan adalah sama dengan nilai maksimal (100) selain itu maka nilai jarak adalah nol (0).

Sebelum normalisasi:

$$118220078\_agama = 1 \text{ dan } 118190068\_agama = 1$$

Setelah normalisasi maka jarak\_agama = 0

c. Untuk variabel UKT

Sebelum normalisasi:

$$118220078\_UKT = 5$$

$$118190068\_UKT = 2$$

$$\text{Jarak} = 5 - 2 = 3$$

Setelah normalisasi maka jarak\_UKT = 37,5 dengan melakukan perhitungan menggunakan Persamaan (24) seperti di bawah ini.

$$v' = \frac{3 - 0}{8 - 0} (100 - 0) + 0$$

$$v' = 37,5$$

d. Untuk variabel Asal Daerah

Sebelum normalisasi, berdasarkan tabel 3.7 maka jarak\_wilayah = 64

Setelah normalisasi maka jarak\_wilayah = 2,51 dengan melakukan perhitungan menggunakan Persamaan (24) seperti dibawah ini.

$$v' = \frac{64 - 0}{2552 - 0} (100 - 0) + 0$$

$$v' = 2,51$$

e. Total jarak antara mahasiswa dengan NIM 118220078 dan 118190068 adalah

$$d_{12} = \text{jarak\_prodi} + \text{jarak\_agama} + \text{jarak\_UKT} + \text{jarak\_wilayah}$$

$$= \sqrt{(100)^2 + (0)^2 + 37,5^2 + (2,51)^2}$$

$$= 106,829$$

Tabel 3.5 Normalisasi Atribut

<b>Atribut</b>	<b>Nilai Minimal</b>	<b>Nilai Maksimal</b>	<b>Nilai Setelah Normalisasi Minimal</b>	<b>Nilai Setelah Normalisasi Maksimal</b>
Asal Daerah	0	2.552	0	100
Program Studi	0	100	0	100
Kemampuan Ekonomi	0	8	0	100
Agama	0	100	0	100

Tabel 3.6 Jarak Agama di Normalisasi

<b>Agama1</b>	<b>Agama2</b>	<b>Jarak Normalisasi</b>
1	2	100
1	3	100
1	4	100
1	5	100
1	6	100

Tabel 3.7 Jarak Wilayah di Normalisasi (Asal Daerah)

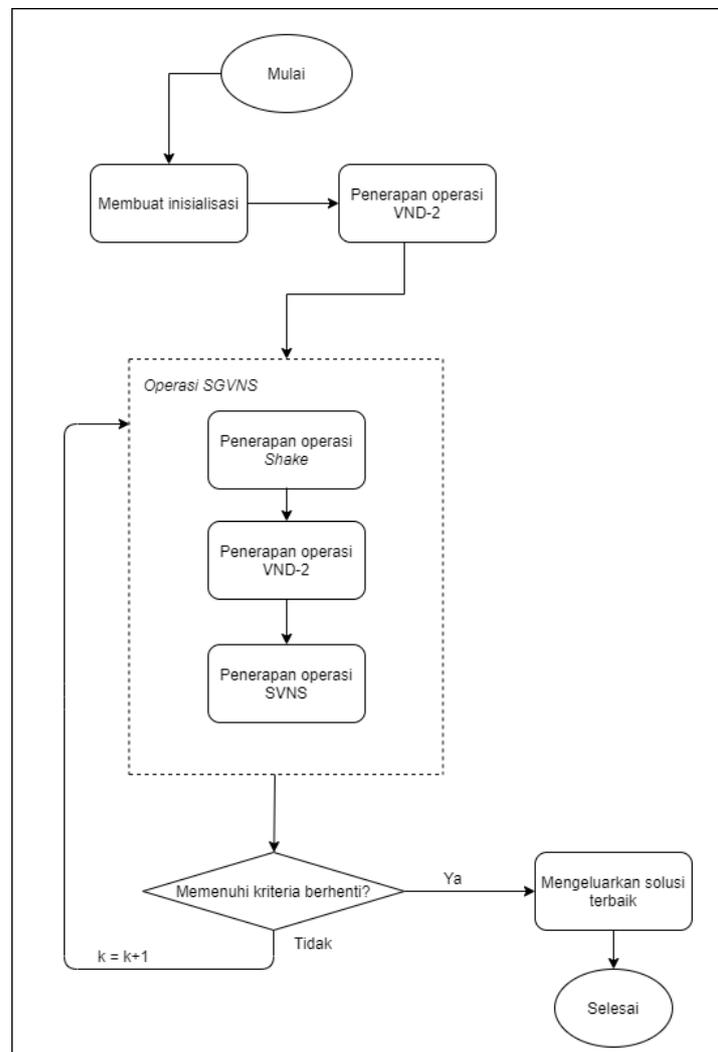
<b>Kota 1</b>	<b>Nama Kota 1</b>	<b>Kota 2</b>	<b>Nama Kota 2</b>	<b>Jarak</b>	<b>Jarak normalisasi</b>
022200	Kabupaten Bekasi	020500	Kabupaten Bogor	64	2.51
022200	Kabupaten Bekasi	016300	Kota Jakarta Selatan	42	1.65
022200	Kabupaten Bekasi	016400	Kota Jakarta Timur	35	1.37
022200	Kabupaten Bekasi	026100	Kota Bogor	49	1.92
022200	Kabupaten Bekasi	026500	Kota Bekasi	27	1.06
022200	Kabupaten Bekasi	026600	Kota Depok	43	1.68

Tabel 3.8 Jarak UKT di Normalisasi

UKT 1	UKT 2	Jarak	Jarak normalisasi
0	1	1	12.50
0	2	2	25.00
0	3	3	37.50
0	4	4	50.00
0	5	5	62.50
0	6	6	75.00
0	7	7	87.50
0	8	8	100.00

### 3.2.3 Implementasi SGVNS

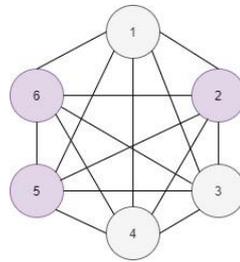
Implementasi SGVNS terdiri dari beberapa langkah yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Arsitektur SGVNS

### 1. Inisialisasi

Kegiatan pada tahap ini adalah melakukan penempatan elemen secara acak ke dalam grup dengan memperhatikan nilai maksimum dan minimum ukuran grup. Sebagai contoh ada 6 buah data mahasiswa seperti pada Tabel 3.2 yang akan ditempatkan pada 2 kamar asrama dengan nilai maksimum dan minimum yang sama yaitu 3 mahasiswa per kamar. Elemen data tersebut dipilih secara acak untuk menempati sebuah grup yang juga terpilih secara acak. Di misalkan dari prosedur acak tersebut dihasilkan  $g_1 = \{ 2, 5, 6 \}$  dan  $g_2 = \{ 4, 1, 3 \}$  yang dapat dilihat pada Tabel 3.10 dan diilustrasikan di Gambar 3.3 yang dibuat dengan mengacu kepada literatur [1]. Setelah semua grup terisi sesuai dengan nilai maksimum dan minimum nya maka segera dilakukan perhitungan nilai fungsi objektif dengan menjumlahkan nilai diversitas antar anggota di dalam grup sesuai dengan Persamaan (11).



Gambar 3.3 Ilustrasi dari Solusi Awal

Berikut adalah contoh perhitungan nilai fungsi objektif.

$$\begin{aligned}
 f &= d_{25} + d_{26} + d_{56} + d_{41} + d_{43} + d_{13} \\
 &= 213.48 + 126.96 + 239.66 + 201.37 + 112.93 + 214.15 \\
 &= 1,108.55
 \end{aligned}$$

Nilai diversitas tersebut dihasilkan dengan menjumlahkan jarak antar atribut dari setiap elemen. Nilai jarak antar elemen dapat dilihat pada Tabel 3.9. Selain itu juga dilakukan perhitungan matriks  $sd^c[i][g]$  yang akan digunakan untuk kunjungan yang lebih efisien dan perbaruan solusi serta dapat digunakan untuk menghitung nilai fungsi objektif. Contoh nilai matriks  $sd^c[i][g]$  dapat dilihat pada Tabel 3.11 yang dihitung menggunakan Persamaan (10).

Tabel 3.9 Jarak Antar Elemen

$d_{ij}$	1	2	3	4	5	6
1	0	140.01	214.15	210.67	226.68	164.89
2	140.01	0	251.49	239.38	213.48	126.96
3	214.15	251.49	0	112.93	238.13	277.7
4	201.37	239.38	112.93	0	225.9	265.32
5	226.68	213.48	238.13	225.9	0	239.66
6	164.89	126.96	277.7	265.32	239.66	0

Tabel 3.10 Array  $x^c$  Sebagai Solusi Awal

i	1	2	3	4	5	6
$x^c [i]$	2	1	2	2	1	1

Tabel 3.11 Matriks  $sd^c$  dari Solusi Awal

i	1	2	3	4	5	6
$sd^c[i][1]$	531.58	340.44	767.32	730.6	453.14	366.62
$sd^c[i][2]$	415.52	630.88	327.08	314.3	690.71	707.91

## 2. Shake

Proses ini menghasilkan solusi acak dari  $k$  *neighborhood* dari solusi saat ini dengan melakukan  $k$  *random moves*. Di setiap gerakan memilih secara acak 2 elemen yang menjadi milik 2 kelompok berbeda lalu ditukar. Melanjutkan contoh sebelumnya, telah dipilih  $k = 2$  sehingga akan dilakukan 2 kali *random moves*. Elemen terpilih secara acak untuk bertukar adalah 4 dengan 2 di iterasi pertama dan 6 dengan 3 pada terasi kedua. Maka kondisi saat ini menghasilkan grup  $g_1 = \{4, 5, 3\}$  dan  $g_2 = \{1, 2, 6\}$ . Setelah proses penukaran selesai, maka nilai fungsi objektif dan nilai matriks  $sd^c$  harus diperbarui dengan cara yang sama dengan sebelumnya. Berikut adalah contoh nilai fungsi objektif dan matriks  $sd^c$  setelah dilakukan operasi *Shake* yang dapat dilihat pada Tabel 3.12.

$$\begin{aligned}
 f &= d_{34} + d_{35} + d_{45} + d_{12} + d_{16} + d_{26} \\
 &= 112.93 + 238.13 + 225.9 + 140.01 + 164.89 + 126.96 \\
 &= 1008.82
 \end{aligned}$$

Tabel 3.12 Matriks  $sd^c$  dari Proses *Shake*

i	1	2	3	4	5	6
$sd^c[i][1]$	642.2	704.35	351.06	338.83	464.03	782.68
$sd^c[i][2]$	304.9	266.97	743.34	706.07	679.82	291.85

### 3. VND-2

Proses pencarian lokal ini menggunakan 2 tahap yaitu *Insertion* dan *Swap*.

#### a. *Insertion*

Prosedur ini memindahkan tepat hanya satu buah elemen dari kelompok saat ini ke kelompok lain. Akan tetapi sebelum pemindahan elemen, perlu dihitung selisih nilai fungsi objektif setelah dan sebelum pemindahan. Apabila nilai selisih lebih besar dari nol maka elemen tersebut dimasukkan ke dalam kelompok yang baru. Untuk menghitung selisih nilai fungsi objektif digunakan Persamaan (14).

Dimisalkan elemen 1 yang saat ini berada pada grup  $g_2$  hendak disisipkan ke dalam grup  $g_1$ . Maka dilakukan perhitungan fungsi objektif dengan mengurangi nilai keberagaman elemen 1 di  $g_1$  ( $f(x^n)$ ) dengan nilai keberagaman elemen 1 di  $g_2$  ( $f(x^c)$ ).

$$\Delta f = f(x^n) - f(x^c)$$

$$= sd^c[i][g_1] - sd^c[i][g_2]$$

$$= 642.2 - 304.9$$

$$= 337.3$$

$$= 337.3 > 0, \text{ maka elemen 1 berpindah dari } g_2 \text{ ke } g_1$$

Karena adanya perpindahan elemen maka jumlah keanekaragaman dari masing-masing kelompok perlu diperbarui dengan menggunakan Persamaan (15) dan Persamaan (16). Berikut adalah contoh pembaruan nilai keberagaman disetiap kelompok.

a. Nilai keberagaman di kelompok 1 dengan anggota kelompok {4 5 3}

$$sd^c[j][g_1] = sd^c[j][g_1] + d_{ji}$$

$$- \quad sd^c[4][g_1] = sd^c[4][g_1] + d_{41}$$

$$- \quad sd^c[5][g_1] = sd^c[5][g_1] + d_{51}$$

$$- \quad sd^c[3][g_1] = sd^c[3][g_1] + d_{31}$$

b. Nilai keberagaman di kelompok 2 dengan anggota kelompok {2 6}

$$sd^c[j][g_2] = sd^c[j][g_2] - d_{ji}$$

$$- \quad sd^c[2][g_2] = sd^c[2][g_2] - d_{21}$$

$$- \quad sd^c[6][g_2] = sd^c[6][g_2] - d_{61}$$

Proses penyisipan elemen ini dapat mengakibatkan ukuran kelompok menjadi tidak sesuai dengan nilai maksimum dan minimum yang didefinisikan. Untuk menjaga konsistensi ukuran kelompok maka dilakukan proses pengeluaran elemen yang ditentukan berdasarkan nilai *fitness* yang dihasilkan. Elemen yang memberikan nilai *fitness* terkecil akan akan dikeluarkan dari kelompok dan disisipkan ke kelompok lain. Apabila elemen 1 gagal disisipkan pada kelompok 2 karena menghasilkan nilai *fitness* terkecil maka elemen 1 akan dicoba untuk disisipkan kedalam kelompok lainnya. Hal ini terus dilakukan sampai ke elemen terakhir.

b. *Swap*

Pada tahap ini, solusi baru dihasilkan dengan menukar sepasang elemen dari kelompok yang berbeda. Untuk menentukan apakah elemen tersebut ditukar atau tidak maka digunakan selisih antara fungsi objektif saat ini dengan fungsi objektif yang dihasilkan dari gerakan *swap*. Persamaan (17) digunakan untuk menghitung selisih nilai fungsi objektif. Apabila nilai selisih tersebut lebih besar dari nol maka pasangan elemen tersebut saling bertukar kelompok. Iterasi ini terus berulang sampai setiap elemen dilakukan percobaan pertukaran. Dimisalkan elemen 5 pada kelompok  $g_1$  hendak bertukar dengan elemen 6 di kelompok  $g_2$ , maka dilakukan perhitungan selisih fungsi objektifnya.

$$\Delta f = f(x^n) - f(x^c)$$

$$= sd^c[i][g_j] - sd^c[j][g_i] + sd^c[i][g_i] - sd^c[j][g_j] - 2d_{ij}$$

$$= sd^c[5][g_2] - sd^c[6][g_1] + sd^c[5][g_1] - sd^c[6][g_2] - 2d_{56}$$

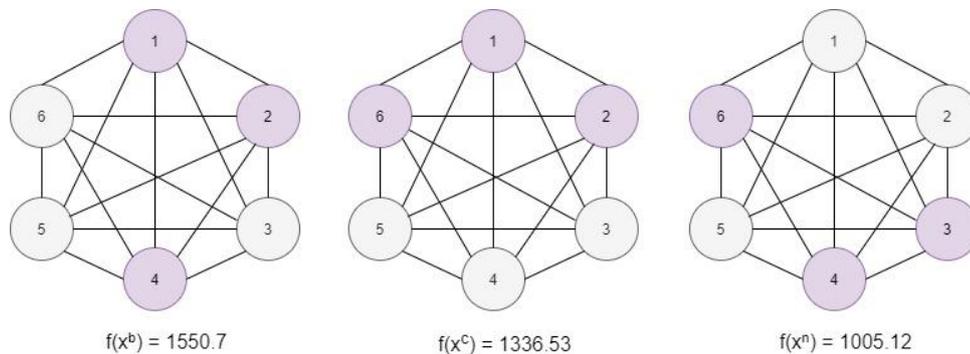
$$= x$$

$$x = \begin{cases} x < 0, & \text{elemen 5 dan 6 tidak bertukar posisi} \\ x > 0, & \text{elemen 5 dan 6 bertukar posisi} \end{cases}$$

Jika peningkatan solusi tidak ditemukan pada proses *Insertion* maka pencarian dilanjutkan ke tahap *Swap*. Apabila peningkatan tidak ditemukan setelah selesai melakukan *Insertion* dan *Swap* maka pencarian lokal dihentikan.

#### 4. SVNS

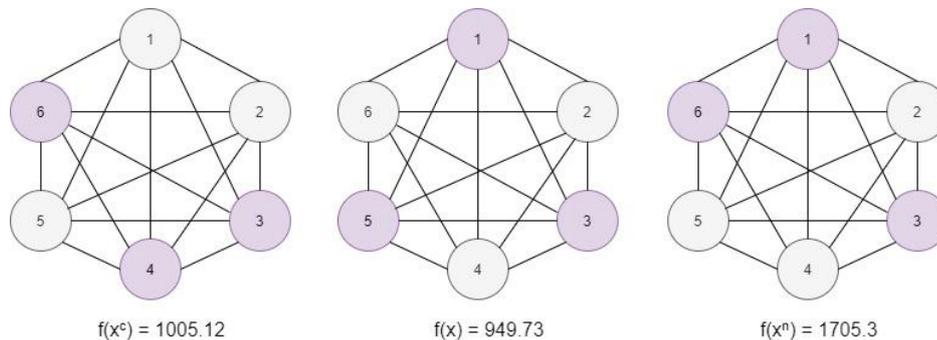
Solusi yang dihasilkan mungkin bukanlah solusi terbaik ( $x^b$ ), oleh karena itu solusi terbaik tidak akan digantikan dengan solusi saat ini ( $x^c$ ) ini apabila fungsi objektif pada solusi saat ini tidak lebih besar dari solusi terbaik. Akan tetapi, sangat memungkinkan untuk mengganti solusi saat ini dengan solusi baru apabila Persamaan (22) terpenuhi meskipun nilai fungsi objektif dari solusi baru tidak lebih besar dari solusi saat ini.



Gambar 3.4 Ilustrasi solusi terbaik, solusi saat ini dan solusi baru

Dimisalkan seperti pada Gambar 3.4 yang dibuat dengan mengacu pada literatur [1], solusi terbaik ( $f^b$ ) bernilai 1550,7 dan solusi saat ini ( $f^c$ ) bernilai 1336.53. Setelah dilakukan pencarian lokal ternyata solusi baru ( $f^n$ ) tidak lebih baik dari solusi saat ini, dengan nilai 1005.12. Untuk mengetahui apakah  $x^c$  akan berpindah ke  $x^n$  maka perlu dilakukan perhitungan dengan Persamaan (22). Untuk mengukur jarak antara solusi  $x_1$  dengan  $x_2$  digunakan Persamaan (23). Apabila syarat terpenuhi maka  $x^c$  akan digantikan dengan  $x^n$  meskipun nilainya jauh lebih rendah dari  $x^c$ . Dalam contoh ini, dianggap bahwa  $x^n$  memenuhi Persamaan (22) sehingga solusi saat ini berpindah ke solusi baru. Pada Gambar 3.5 diketahui bahwa setelah melakukan prosedur *shake* didapatkan nilai  $f^n$  sebesar 949.73 dan kemudian meningkat setelah dilakukan pencarian lokal dengan hasil  $f^n$  senilai 1705.3. Solusi

terbaru menunjukkan hasil yang lebih baik dari solusi terbaik sebelumnya sehingga nilai solusi terbaik berganti menjadi nilai solusi baru.



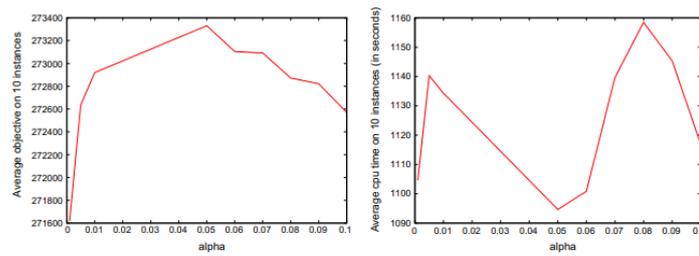
Gambar 3.5 Ilustrasi solusi saat ini, solusi setelah prosedur *shake*, dan solusi baru

### 5. Kriteria Berhenti

Pencarian solusi akan berhenti apabila waktu (*running time CPU*) yang telah di definisikan di awal telah habis atau telah mencapai batas nya. Selama waktu masih tersedia maka pencarian solusi terbaik akan terus dilakukan. Pada penelitian sebelumnya parameter berhenti yang digunakan adalah  $t_{\max}$  dengan nilai yaitu 120s dan 600s. Nilai 120s digunakan pada penelitian yang dilakukan Jack Brimberg, dkk untuk permasalahan MDGP jenis EGS dengan jumlah data 480 [1]. Sementara nilai 600s digunakan pada penelitian yang dilakukan oleh Arief, dkk untuk memproses data dengan jumlah 460 [2]. Apabila  $t_{\max}$  telah mencapai batas nya sementara proses *Insertion* dan *Swap* belum selesai maka program akan tetap memproses *Insertion* dan *Swap* hingga selesai dengan tidak melakukan perulangan pencarian lokal.

### 6. Parameter VNS

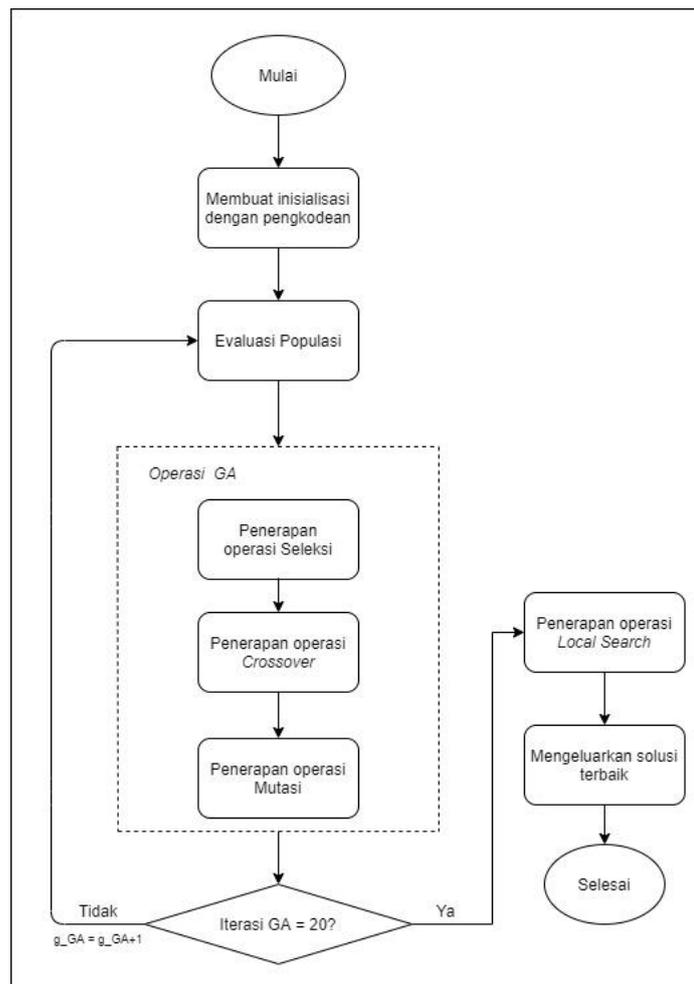
Ada tiga buah parameter yang harus didefinisikan diawal sebelum menjalankan program. Ketiganya yaitu batas waktu CPU ( $t_{\max}$ ), jumlah maksimum *neighborhood* ( $k_{\max}$ ) dan parameter  $\alpha$  untuk proses *skewed moves*. Pada penelitian sebelumnya telah didapatkan nilai terbaik untuk parameter  $\alpha$  adalah 0.05 dengan jumlah  $k_{\max}$  adalah 60 [1]. Untuk mendapatkan nilai  $\alpha$  tersebut dilakukan uji coba terhadap 10 buah data dengan rentang nilai  $\alpha = 0.01 - 0.1$  dan dipatkan nilai fungsi objektif maksimum serta waktu CPU terendah adalah saat  $\alpha$  bernilai 0.05 seperti yang ditunjukkan pada Gambar 3.6.



Gambar 3.6 Pencarian nilai alpha terbaik [1]

### 3.2.4 Implementasi HGA

Implementasi HGA terdiri dari beberapa langkah yang dapat dilihat pada Gambar 3.7.



Gambar 3.7 Arsitektur *Hybrid Genetic Algorithm*

#### 1. Inisialisasi

Pada tahap inisialisasi ini dilakukan penempatan elemen  $N$  secara acak ke dalam kelompok  $G$  dengan menyesuaikan ukuran maksimal dan minimal masing-masing

kelompok. Sebagai contoh, di tentukan bahwa ada 12 mahasiswa yang akan di tempat kan ke dalam 3 kamar asrama. Masing-masing kamar memiliki nilai maksimum dan minimum yang sama yaitu 4 mahasiswa per kamar asrama. Diawal perlu ditentukan jumlah populasi, misalnya jumlah populasi adalah 5 maka perlu di bangkitkan 5 buah kromosom dengan contoh sebagai berikut.

Kromosom 1 = 132132131322|123

$$g_1 = \{1, 4, 7, 9\} \quad g_2 = \{3, 6, 11, 12\} \quad g_3 = \{2, 5, 8, 10\}$$

Kromosom 2 = 321123213312|123

$$g_1 = \{3, 4, 8, 11\} \quad g_2 = \{2, 5, 7, 12\} \quad g_3 = \{1, 6, 9, 10\}$$

Kromosom 3 = 231212331231|123

$$g_1 = \{3, 5, 9, 12\} \quad g_2 = \{1, 4, 6, 10\} \quad g_3 = \{2, 7, 8, 11\}$$

Kromosom 4 = 113321322123|123

$$g_1 = \{1, 2, 6, 10\} \quad g_2 = \{5, 8, 9, 11\} \quad g_3 = \{3, 4, 7, 12\}$$

Kromosom 5 = 312213123213|123

$$g_1 = \{2, 5, 7, 11\} \quad g_2 = \{3, 4, 8, 10\} \quad g_3 = \{1, 6, 9, 12\}$$

## 2. Evaluasi Populasi

Pada tahap ini dilakukan perhitungan nilai fungsi objektif dengan menjumlahkan nilai keberagaman pada semua kelompok. Nilai keberagaman pada suatu kelompok didapatkan dari penjumlahan nilai keberagaman masing-masing anggotanya. Sementara perhitungan nilai keberagaman antar 2 anggota/mahasiswa dilakukan dengan formula *Euclidean distance*. Pada permasalahan asrama ini, dilakukan perhitungan jarak antar mahasiswa dengan menghitung jarak antar atributnya. Sebagai contoh, dilakukan perhitungan nilai *fitness* dari masing-masing kromosom sehingga menghasilkan nilai seperti dibawah ini,

Kromosom 1

$$\text{nilai fitness} = dg_1 + dg_2 + dg_3 = 100 + 120 + 180 = 400$$

$$g_1 = d_{14} + d_{17} + d_{19} + d_{47} + d_{49} + d_{79} = 100$$

$$g_2 = d_{36} + d_{311} + d_{312} + d_{611} + d_{1112} = 120$$

$$g_3 = d_{35} + d_{28} + d_{210} + d_{58} + d_{810} = 180$$

$$\text{Kromosom 2} = 250$$

$$\text{Kromosom 3} = 100$$

$$\text{Kromosom 4} = 100$$

$$\text{Kromosom 5} = 150$$

### 3. Seleksi

Proses seleksi dilakukan untuk memilih *parent* dengan menggunakan *roulette wheel* dan menempatkan kromosom sesuai dengan besaran peluangnya. Semakin besar nilai *fitness* maka semakin besar pula peluang individu terpilih sebagai *parent*. Nilai peluang masing-masing kromosom untuk terpilih sebagai *parent* dapat dihitung dengan menggunakan Persamaan (9). Setelah semua kromosom menempati bagian dari roda *roulette* maka roda tersebut diputar. Individu yang berada pada segmen di mana roda *roulette* berhenti akan terpilih sebagai *parent*. Proses ini diulang sampai menghasilkan jumlah individu yang diinginkan. Dalam persoalan pembagaian kamar asrama ini dibutuhkan 2 individu yang akan bertindak sebagai *parent*. Berikut adalah contoh proses terpilihnya individu sebagai *parent* untuk proses *crossover*.

$$\text{Kromosom 1} = \text{nilai fitness} = 400 = 40\%$$

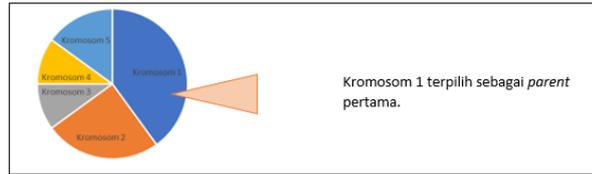
$$\text{Kromosom 2} = \text{nilai fitness} = 250 = 25\%$$

$$\text{Kromosom 3} = \text{nilai fitness} = 100 = 10\%$$

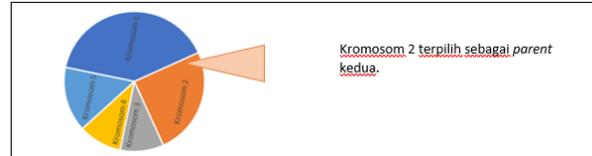
$$\text{Kromosom 4} = \text{nilai fitness} = 100 = 10\%$$

$$\text{Kromosom 5} = \text{nilai fitness} = 150 = 15\%$$

Berdasarkan Gambar 3.8 terlihat bahwa roda berhenti pada segmen kromosom 1 dan pada Gambar 3.9 terlihat bahwa roda berhenti pada kromosom 2 sehingga individu 1 dan 2 terpilih sebagai *parent*.



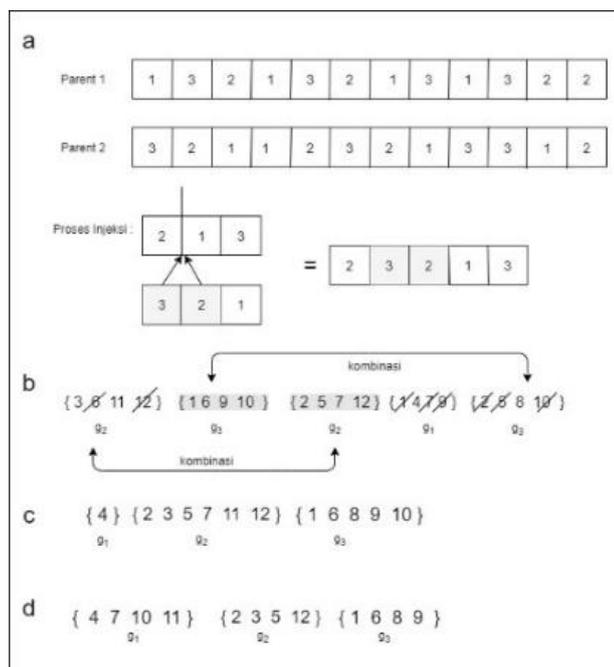
Gambar 3.8 Pemilihan Individu 1 dengan *Roulette Wheel*



Gambar 3.9 Pemilihan Individu 2 dengan *Roulette Wheel*

4. *Crossover*

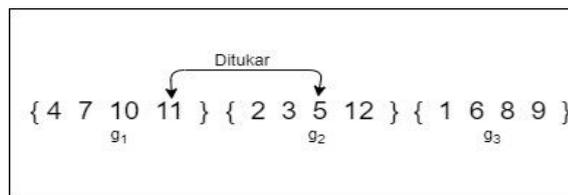
Pada tahap ini, keturunan dihasilkan dengan memasukkan isi antara titik tengah *parent* kedua ke titik tengah *parent* pertama. Pada proses seleksi sebelumnya telah terpilih individu 1 dan individu 2 sebagai *parent*. Oleh karena itu dilakukan injeksi dari titik tengah individu 2 ke titik tengah individu 1 seperti yang ditunjukkan pada Gambar 3.10a. Setelah proses injeksi dilakukan akan ada elemen yang muncul lebih dari sekali sehingga untuk mengatasinya perlu dilakukan eliminasi seperti pada gambar 3.10b. Setelah proses eliminasi selesai, selanjutnya elemen-elemen tersebut disesuaikan dengan ukuran kelompok nya yang dapat dilihat pada Gambar 3.10 c dan d.



Gambar 3.10 Contoh Proses *Crossover*

## 5. Mutasi

Proses mutasi dilakukan dengan menentukan nilai  $\beta$  yang merupakan nilai dari jumlah gen yang akan di mutasi. Sepasang kelompok secara acak akan terpilih untuk melakukan mutasi lalu terjadilah pertukaran  $\beta$  elemen dari kelompok satu ke kelompok lain yang terpilih. Sebagai contoh, proses mutasi didefinisikan  $\beta$  dengan nilai 1 dan kelompok yang terpilih adalah  $g_1$  dan  $g_3$ . Maka 1 buah elemen dari masing-masing kelompok terpilih secara acak untuk dilakukan pertukaran. Di misalkan elemen yang terpilih untuk bertukar adalah elemen 5 dari  $g_2$  dan elemen 11 dari  $g_1$  yang dapat dilihat pada Gambar 3.11. Maka hasil akhir dari mutasi tersebut adalah  $g_1 = \{ 4 \ 5 \ 7 \ 10 \}$  dan  $g_2 = \{ 2 \ 3 \ 11 \ 12 \}$ .



Gambar 3.11 Contoh Proses Mutasi

## 6. Local Search

Proses ini melakukan percobaan perpindahan antar elemen dari kelompok yang berbeda, apabila terjadi peningkatan nilai fungsi objektif maka elemen ditukar dan solusi saat ini digantikan dengan solusi baru yang didapat dari hasil pertukaran.

## 7. Kriteria Berhenti

Berdasarkan penelitian sebelumnya dinyatakan bahwa hasil maksimum yang stabil didapat saat jumlah generasi mencapai kurang lebih 15 generasi [9] sementara pada penelitian yang dilakukan oleh Arief dkk menggunakan 30 generasi sebagai kriteria berhentinya [2]. Oleh karena penelitian ini menggunakan permasalahan yang serupa dengan penelitian yang dilakukan Arief dkk maka digunakan 30 generasi sebagai parameter berhenti. Proses GA akan terus berulang hingga jumlah iterasi dari GA mencapai 30 dan melakukan sekali pencarian lokal. Setelah semua langkah dilakukan maka akan dihasilkan solusi pengelompokan dari algoritma HGA.

### 3.2.5 Membandingkan Hasil Kerja

Perbandingan hasil kinerja dari masing-masing algoritma diukur berdasarkan nilai fungsi objektif yang diperoleh beserta waktu (*running time CPU*) yang

digunakan untuk memproses. Besaran nilai fungsi objektif berbanding lurus dengan kualitas pengelompokkan, semakin tinggi nilai fungsi objektif atau mengarah pada nilai tak hingga maka semakin tinggi pula kualitas pengelompokkan yang dihasilkan. Cara menghitung fungsi objektif adalah dengan menerapkan Persamaan (2) yang secara sederhana dapat ditulis seperti Persamaan(11). Pada metode SGVNS dapat digunakan Persamaan (13) dimana ketiga persamaan tersebut menghasilkan nilai fungsi objektif yang sama. Berikut adalah contoh menghitung nilai fungsi objektif dengan  $g_1 = \{3\ 4\ 5\}$  dan  $g_2 = \{1\ 2\ 6\}$ .

$$\begin{aligned}
 \text{Maximize } \sum_{g=1}^G \sum_{i=1}^N \sum_{j>i}^N d_{ij} x_{ig} x_{jg} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{ij} \chi(x^c, i, j) \\
 &= d_{12}(1) + d_{13}(0) + d_{14}(0) + d_{15}(0) + d_{16}(1) + d_{23}(0) + d_{24}(0) + d_{25}(0) + d_{26}(1) \\
 &\quad + d_{34}(1) + d_{35}(1) + d_{36}(0) + d_{45}(1) + d_{46}(0) + d_{56}(0) \\
 &= d_{12} + d_{16} + d_{26} + d_{34} + d_{35} + d_{45}
 \end{aligned}$$

### 3.3 Spesifikasi Komputer

Untuk pengujian dua algoritma yang telah dipaparkan sebelumnya digunakan komputer yang sama. Komputer yang akan digunakan untuk melakukan pengujian tersebut adalah komputer dengan spesifikasi yang dapat dilihat pada Tabel 3.13.

Tabel 3.13 Spesifikasi Komputer

Sistem Manufaktur	DELL Inc
Sistem Operasi	Windows 7 Pro 64-bit
Prosesor	Intel® Xeon® Silver 4112
Memori	16384MB RAM