

## BAB 3

### METODE PENELITIAN

Pada bab ini dibahas metode yang akan digunakan dalam menganalisa performa dari masing-masing metode *clustering* UE yang terdeteksi dalam lingkup layanan suatu BTS. dibahas juga alat dan bahan yang di gunakan untuk melaksanakan penelitian.

#### 3.1 ALAT YANG DIGUNAKAN

Penelitian ini menggunakan suatu pemodelan algoritma pengelompokan pada *Direct Virtual Small Cell* untuk menganalisis performa algoritma *clustering* GAPC dan algoritma K-means yang diimplementasikan pada VSC.

##### 3.1.1 *Hardware*

Perangkat keras yang digunakan pada penelitian ini menggunakan 1 laptop dengan spesifikasi terdapat pada tabel 3.1 sebagai berikut :

**Tabel 3. 1 Spesifikasi Perangkat**

Spesifikasi Laptop	OS	Windows 10
	Processor	AMD Athlon 300U with Radeon Vega Mobile Gfx 2.40 GHz
	RAM	8,00 GB (5,88 GB usable)
	SSD	512 GB

##### 3.1.2 *Software*

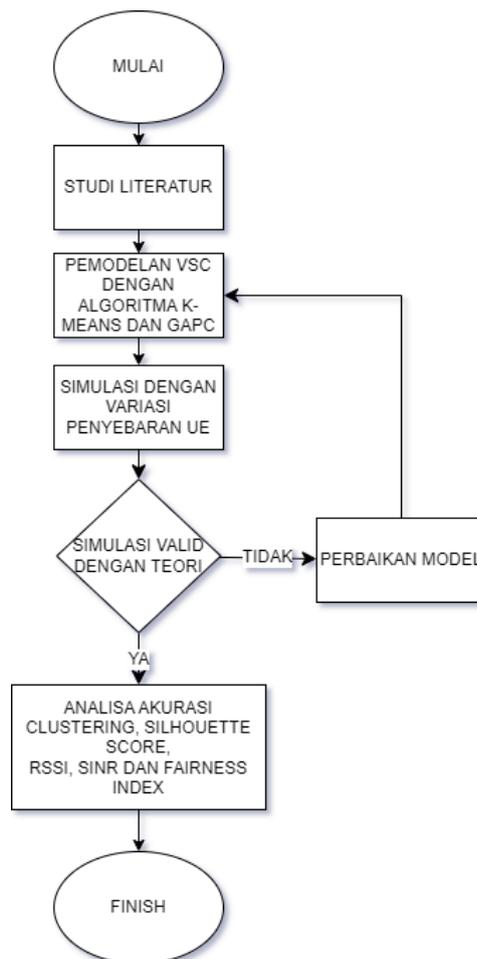
Perangkat lunak yang digunakan untuk merancang dan menjalankan pemodelan yang telah dirancang dalam penelitian ini ditunjukkan pada tabel 3.2.

**Tabel 3. 2 Software yang Digunakan**

Software	Versi	Fungsi
MATLAB	2020 a	Simulasi pengelompokan UE menggunakan algoritma K-Means dan GAPC

### 3.2 ALUR PENELITIAN

Pada penelitian ini terdapat tahapan – tahapan yang perlu di laksanakan untuk dapat merancang VSC dengan algoritma *clustering* K-means dan GAPC, tahapan penelitian dibuat sebagai pedoman penuln untuk melaksanakan penelitian, sehingga penelitan yang dilaksanakan tetap pada prosedur pelaksanaan dan mendapatkan hasil yang telah ditentukan.



**Gambar 3. 1 Alur Penelitian**

Pada gambar 3.1 dapat dipahami bahwa untuk melaksanakan penelitian perbandingan performansi VSC dengan algoritma K-Means dan GAPC-SNR, penulis perlu melaksanakan tahapan penelitian untuk dapat mengetahui performa masing masing algoritma yang digunakan. Pada bagian pertama terdapat studi literatur untuk menambah wawasan dalam menyusun penelitian ini, selanjutnya terdapat perancangan VSC yaitu membuat konsep sistem kerja dan setiap parameter yang dapat mempengaruhi VSC, selanjutnya perancangan model algoritma K-Means dan GAPC yang digunakan dalam VSC sehingga dapat membantu memilih sekumpulan UE mana yang membutuhkan pancaran VSC, setelah perancangan selesai selanjutnya tahap pembuatan model VSC yang menggunakan algoritma K-means dan GAPC sehingga model tersebut dapat disimulasikan, untuk menguji model yang telah dibuat maka penulis perlu membuat skenario pengujian model, setelah model VSC diuji maka penulis dapat menentukan apakah model tersebut telah berfungsi sebagaimana mestinya, jika ternyata belum sesuai maka akan dilaksanakan perbaikan, namun jika telah sesuai maka dilakukan analisis akurasi *clustering*, *silhouette score*, RSSI, SINR dan *fairness index* dari masing masing algoritma.

### 3.3 PARAMETER INPUT

Penelitian ini dilaksanakan dengan langkah-langkah sesuai pada gambar 3.2.1, masukkan yang digunakan dalam penelitian ini sebagai berikut :

**Tabel 3. 3 Parameter *Input* Algoritma K-Means**

Lokasi BS	500 500 (tengah area)
Luas Area <i>Clustering</i>	1000 m <sup>2</sup>
Min Member <i>Cluster</i>	5
Radius <i>Centroid</i>	100 m
Max <i>Centroid</i>	5
Jumlah UE	200
Pembagian Daya	50%, 60%, 70%, 80%, 90%

**Tabel 3. 4 Parameter *Input* Algoritma GAPC**

Luas Area <i>Clustering</i>	1000 m
Lokasi BS	500 500 (tengah area)
Jumlah UE	200
Min Member <i>Cluster</i>	5
Radius <i>Eligible</i> UE	100 m
Max Member <i>Cluster</i>	12
<i>Grid Size</i>	100
Max Radius <i>Cluster</i>	100
Pembagian Daya	50%, 60%, 70%, 80%, 90%

Dengan parameter pada tabel 3.3 dan tabel 3.4 akan dilakukan masing masing 100 kali iterasi dengan perbedaan posisi UE. Persebaran UE yang berbeda akan mengakibatkan terjadinya perhitungan ulang untuk membuat *clustering* baru, sehingga keluaran dari parameter *input* pada tabel 3.1 dan tabel 3.2 akan digunakan sebagai pertimbangan dalam menentukan algoritma mana yang memiliki performa paling baik.

### **3.4 PARAMETER *OUTPUT***

Untuk menentukan algoritma mana yang memiliki performa yang paling baik maka penulis akan menganalisa parameter – parameter yang ada pada tabel 3.5

**Tabel 3. 5 Parameter *Output* Masing-Masing Algoritma**

parameter	keterangan
Akurasi <i>Clustering</i>	Berapa kelompok UE yang seharusnya dapat menjadi <i>cluster</i> namun tidak menjadi <i>cluster</i>
<i>Silhouette Score</i>	Suatu parameter untuk mengevaluasi perbedaan dan persamaan dari suatu objek dalam dua algoritma yang digunakan

**Tabel 3.5 Parameter *Output* Masing-Masing Algoritma (lanjutan)**

parameter	keterangan
<i>Received Signal Strength Indicator</i> (RSSI)	Parameter untuk menjelaskan kekuatan sinyal yang diterima oleh <i>Receiver</i> dari <i>Transmitter</i>
<i>Signal Interference Noise Ratio</i> (SINR)	Parameter yang digunakan untuk mengukur kualitas sinyal
<i>Fairness Index</i>	Parameter yang digunakan untuk mengukur keadilan <i>throughput</i> diantara <i>makrocell</i> dan DVSC

Nilai parameter yang didapat merupakan hasil dari simulasi dari model yang dibuat pada *software* MATLAB.

### 3.5 SKENARIO PENGUJIAN

Pada penelitian ini menggunakan algoritma yang dipengaruhi oleh posisi UE, sehingga diperlukan metode distribusi persebaran UE yang mengacu pada kehidupan nyata.

#### 3.5.1. Pembagian Daya

Pengambilan data divariasikan terhadap pembagian alokasi daya pada masing-masing sistem baik pada VSC ataupun pada sistem *Macrocell*. Sistem VSC digunakan untuk melayani sekumpulan UE yang membentuk hotspot sedangkan sistem *Macrocell* digunakan untuk melayani UE yang tidak termasuk kedalam anggota ataupun area *hotspot*. Untuk dapat membagi daya pada kedua sistem digunakan persamaan sebagai berikut :

$$Z = M + V \tag{3.1}$$

Daya total yang tersedia pada sistem antenna dimisalkan dengan  $Z$  dimana daya tersebut berasal dari penjumlahan daya yang digunakan untuk

mengoperasikan *Macrocell* "M" dan daya yang digunakan untuk mengoperasikan VSC "V" sehingga didapatkan persamaan untuk pembagian daya :

$$\alpha_V = \frac{V_{\max}}{V_{\max}+1} \quad (3.2)$$

$$\alpha_M = 1 - \alpha_V \quad (3.3)$$

Persamaan (3.2) digunakan untuk menentukan seberapa besar daya yang akan digunakan dalam menjalankan pemrosesan VSC. Daya total VSC yang digunakan " $\alpha_V$ " berasal dari alokasi daya maksimal yang dapat digunakan " $V_{\max}$ " dibagi dengan alokasi daya maksimal yang ditambahkan dengan alokasi daya yang diharapkan, sehingga pada persamaan (3.3) alokasi daya VSC " $\alpha_V$ " digunakan sebagai pengurang dari 1 untuk mendapatkan persentase daya yang akan digunakan oleh *Macrocell*. Dengan demikian hasil dari  $\alpha_V$  dan  $\alpha_M$  digunakan sebagai pembagi daya total pada tower BTS. Pembagian daya dilakukan dengan variasi  $\alpha_V$  sebesar 50%, 60%, 70%, 80%, 90%. Dengan demikian  $\alpha_M$  yang tersedia merupakan sisa dari penggunaan daya  $\alpha_V$ , pembagian ini dilakukan untuk menguji seberapa efektif operasi VSC pada setiap pembagian daya. Selanjutnya *Jain's Fairness Index* digunakan untuk menguji seberapa adil *throughput* dibagikan kepada setiap UE pada sistem VSC maupun pada sistem MC [25].

### 3.5.2. Model Simulasi Algoritma K-Means dan GAPC

#### A. Algoritma K-Means

Metode pengelompokan K-Means merupakan metode pengelompokan yang digunakan untuk menemukan kluster data terkait pada himpunan data. Objek data dalam konteks ini ialah UE, dan kumpulan data berisi koordinat dimensi  $d$  dan  $u$  untuk koordinat UE. K-means merupakan salah satu metode pengelompokan pertama dan paling mudah diakses, karena memungkinkan pencapaian kluster yang tidak tumpang tindih.

Seperti yang ditunjukkan dalam Algoritma 3.1 dapat disimpulkan bahwa titik pusat *cluster* di K-Means tidak diambil dari koordinat UE mana pun, biasanya ada penyesuaian tergantung pada kebijakan pembentukan VSC untuk menunjuk CH

untuk langkah selanjutnya, misalnya, itu bisa menjadi UE SINR terbaik di dalam *cluster* atau UE terdekat ke pusat *cluster*.

Algoritma 3.1

Pengelompokkan UE menggunakan  $K$ -Means

BS meminta koordinat dimensi  $d$  dari UE  $U$  dan  $\mathcal{K}$  adalah jumlah maksimum *cluster* untuk algoritma  $K$ -Means,  $\mathcal{P}$  adalah persentase jumlah UE minimum dalam populasi,  $d_{\max}$  adalah radius *cluster* maksimum

```

1      Input:  $\mathbf{u} \in \mathbb{R}^d$ ;  $\mathbf{U} = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_U]^T \in \mathbb{R}^{U \times d}$ ;  $\mathcal{K}$ 
2      for  $k = 1: \mathcal{K}$  do
3           $l = 0$ 
4           $\mathbf{K}_{k(l-1)} = \mathbf{0}$ ,  $\mathbf{K}_{kl} = [\mathbf{k}_{1l} \ \cdots \ \mathbf{k}_{kl}]^T \in \mathbb{R}^{k \times d}$  mengacak
           posisi centroid
5          while  $\mathbf{K}_{kl} \neq \mathbf{K}_{k(l-1)}$  do
6           $\mathbf{K}_{k(l-1)} = \mathbf{K}_{kl}$ ;  $l = l + 1$ ;  $J_o = \emptyset$ 

```

Menentukan posisi tengah dari  $k$  *centroid* yang diacak  $\mathbf{K}_{kl}$

```

7      for  $u = 1: U$  do
            $n = \arg \min_{i=1, \dots, k} \|\mathbf{u}_u - \mathbf{k}_{il}\|$  mencari UE yang memiliki
8      jarak terdekat dengan  $\mathbf{k}_{il}$ 
9       $J_{oi} = J_{oi} \cup \mathbf{u}_n$  mengelompokkan UE yang memiliki jarak
           terdekat dengan  $\mathbf{k}_{il}$ 
10     end for

```

$i = 1, \dots, k$ ;  $J_{oi} = [\mathbf{u}_{1oi}, \mathbf{u}_{2oi} \ \cdots \ \mathbf{u}_{oi|J_{oi}|}]^T$ ;  $J_{oi} \in \mathbb{R}^{|J_{oi}| \times d}$

```

11     for  $m = 1: k$  do
12          $\mathbf{k}_{ml} = \frac{\text{sum}(J_{om})}{|J_{om}|}$ ;  $\mathbf{K}_{kl} = [\mathbf{k}_{1l} \ \cdots \ \mathbf{k}_{kl}]^T$  mengupdate
           posisi setiap centroid  $\mathbf{K}_{kl}$ 
13     end for
14     end while

```

Menghitung interferensi yang terjadi pada setiap *cluster*

```

15     for  $i = 1: k$  do

```

16  $J_i = \sum_{u=1}^{|J_{oi}|} \|\mathbf{u}_{uoi} - \mathbf{k}_{il}\|$   
 17 end for  
 18  $J_k = \sum_{i=1}^k J_i$  merupakan total interferensi pada setiap  
*cluster*

Menghitung  $\alpha_k$  menggunakan 2.9 dan, Selanjutnya menghitung  $f(k)$  menggunakan persamaan 2.10,

19  $\alpha_k; f(k)$   
 20 end for

Memilih *cluster* yang optimal berdasarkan :

21  $K = \arg \min_{k=1, \dots, K} f(k)$

*Cluster* yang optimal  $K$  dengan koordinat :

22  $\mathbf{K}_K = [\mathbf{k}_1 \ \dots \ \mathbf{k}_K]^T$   
 23 for  $k = 1:K$  do  
 24  $\bar{J}_k = d_{max} + 1$  nilai yang ditentukan untuk memulai  
 pengulangan  
 25 while  $\sqrt{\bar{J}_k} > d_{max}$  do

Menghitung nilai rata – rata jarak antara UE dengan *centroid*

$$27 \ \bar{d}_k = \frac{1}{|J_{ok}|} \sum_{i=1}^{|J_{ok}|} \|\mathbf{u}_{iok} - \mathbf{k}_k\|$$

Menghitung standar deviasi

$$28 \ \sigma_k = \sqrt{\frac{1}{|J_{ok}|} \sum_{i=1}^{|J_{ok}|} (\|\mathbf{u}_{iok} - \mathbf{k}_k\| - \bar{d}_k)^2}$$

Menghilangkan UE(s) diluar jangkauan *centroid* dan membandingkan dengan fungsi  $\bar{d}_k$  &  $\sigma_k$

29 for  $u = 1:|J_{ok}|$  do  
 30  $d_{ku} = \|\mathbf{u}_{uok} - \mathbf{k}_k\|$   
 31 if  $d_{ku} > (\bar{d}_k + \sigma_k)$  then  
 32  $J_{ok} = J_{ok} \setminus \mathbf{u}_{uok}$  membuang UE  $u$  dari  $J_{ok}$   
 33 end if

34           end for

Perhitungan ulang interferensi antara masing masing *cluster*

$$35 \quad \bar{J}_k = \frac{1}{|J_{ok}|} \sum_{u=1}^{|J_{ok}|} \|\mathbf{u}_{uok} - \mathbf{k}_k\|$$

36           end while

Menentukan jarak UE terjauh dengan pusat *cluster*

$$37 \quad r_k = \max(d_{ku})$$

Menghilangkan *cluster* dengan anggota sedikit (*s*)

38           if  $|J_{ok}| < \mathcal{P}U$  then

39            $J_o = J_o \setminus J_{ok}$  menghapus *cluster*  $J_{ok}$  dari  $J_o$

40           end if

41           end for

## B. Algoritma GAPC-SNR

Algoritma ini menggunakan matriks tanggung jawab (*responsibility matrix*) dan matriks ketersediaan (*availability matrix*) untuk menentukan UE VBS yang layak. Proses iteratif dilakukan untuk menemukan UE VBS yang memiliki nilai positif pada matriks  $E$  setelah iterasi terakhir. Selain itu, algoritma ini juga mempertimbangkan konvergensi waktu dan jumlah UE-VBS aktif yang dihasilkan, parameter yang digunakan dalam menentukan UE VBS dapat dimodifikasi berdasarkan kebutuhan.

Algoritma 3.2

Pengelompokkan UE dengan Modified-Affinity  
Propagation Clustering

*Input:*  $U$  nomor dari  $d$ -dimensi koordinat UE  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_U]^T$  dengan  $\mathbf{U} \in \mathbb{R}^{U \times d}$ ;  $L$  nomor dari  $d$ -dimensi koordinat *Eligible* UE,  $\mathbf{L} = [\mathbf{l}_1 \ \mathbf{l}_2 \ \dots \ \mathbf{l}_L]^T$  dengan  $\mathbf{L} \in \mathbb{R}^{L \times d}$ ;  $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_U\}$  merupakan nilai SINR dari  $u$  berdasar transmisi dari BS;  $n_{min}$  dan  $n_{max}$  yang masing-masing merupakan jumlah UE minimum dan maksimum yang dapat dilayani oleh satu CH



```

17       $A = A \setminus \mathbf{a}_i$            $\mathbf{a}_i$  menghapus UE yang tersedia pada
list  $A$ 
18       $J = J \cup \mathbf{a}_i$           menambahkan  $\mathbf{a}_i$  dengan  $J$ 
19       $L = L \cup \mathbf{a}_i$           memperbaiki eligible UE pada  $L$ 
20      else if ( $|J_{ai}| \geq n_{max}$ )

```

Memindahkan ke-atas  $n_{max}$   $\mathbf{u}_{ai}$  dari  $J_{ai}$  , memindahkan selebihnya ke-  $J$

```

21   $J_{ai} = [\mathbf{u}_{1ai} \ \mathbf{u}_{2ai} \ \dots \ \mathbf{u}_{n_{max}ai}]^T$ 
22   $W = W \cup \mathbf{a}_i$ 
23   $J = J \cup [\mathbf{u}_{(n_{max}+1)ai} \ \mathbf{u}_{(n_{max}+2)ai} \ \dots \ \mathbf{u}_{|J_{ai}|ai}]^T$ 
24      else
25   $W = W \cup \mathbf{a}_i$ 
26      end if
27      end for
28      if ( $J \neq \emptyset$ ) then
29      UE pada  $J$  dilayani oleh MC
30      end if
31      end while

```

### 3.5.3. MaP Persebaran UE

Dalam membuat map persebaran UE penulis menggunakan lebar 1000 x 1000 m<sup>2</sup>, persebaran UE dilaksanakan sedemikian rupa sehingga setiap persebaran yang terjadi dapat menyebabkan terbentuknya *cluster*. Selain mempertimbangkan terjadinya sebuah *cluster* dalam menggunakan metode persebaran UE, perlu mempertimbangkan kepadatan yang berbeda dalam masing masing *cluster*, selain itu terdapat persebaran yang berbeda pada setiap waktu yang berlalu. Dengan demikian metode persebaran yang digunakan mengacu pada kondisi di dunia nyata.

### 3.5.4. Poisson Cluster Process

*Poisson Cluster Process* (PCP) melibatkan pengelompokan titik-titik acak dalam ruang, dimana masing-masing kelompok diatur menurut proses Poisson.

Dalam konteks distribusi pengguna, PCP digunakan untuk memodelkan lokasi pengguna yang terkelompok di sekitar pusat-pusat klaster, sementara pusat-pusat klaster tersebut mengikuti proses Poisson. Hal ini memungkinkan untuk memodelkan korelasi antara lokasi pengguna dan BS dalam jaringan seluler heterogen, yang tidak dapat ditangani oleh proses Poisson biasa.



**Gambar 3. 2 Diagram Alur *Poisson Cluster Process***

Pada gambar 3.2 merupakan alur kerja PCP dalam mendistribusikan data sedemikian rupa. Distribusi data tersebut dilakukan supaya setiap data yang telah terdistribusi dapat membentuk *cluster* berdasarkan posisi data yang lain. Terdapat dua distribusi persebaran UE proses *cluster poisson*, yang pertama *Relevant distance distributions* dan yang kedua *association policies*. Untuk *Relevant distance distributions* mengacu pada distribusi jarak dari SBS ke UE yang dapat disederhanakan dengan fungsi berikut :

$$\begin{aligned}
 & \textit{Probability Density Function(PDF)} \quad : \\
 & f_{R_m}(r_m) = 2\pi\lambda_m r_m \exp(-\pi\lambda_m r_m^2) \quad (3.11)
 \end{aligned}$$

$$\begin{aligned}
 & \textit{Cumulative Distribution Function(CDF)} \quad : \\
 & F_{R_s}(r_m) = 1 - \exp(-\pi\lambda_m r_m^2) \quad (3.12)
 \end{aligned}$$

Dimana  $R_m$  dan  $R_s$  merupakan karakterisasi dari jarak pengguna biasa ke BS terdekat. Dan untuk *association policies* yang mengacu pada bagaimana menentukan distribusi UE dengan pusat SBS yang dapat disederhanakan menjadi fungsi berikut:

$$\mathcal{A}_m^{P1}(v_0) = \int_0^\infty [1 - F_{R_s}(\xi_{sm} r_m | v_0)] f_{R_m}(r_m) d_{r_m} \quad (3.13)$$

Dimana  $F_{Rs}$  dan  $f_{Rm}$  mengacu pada fungsi 3.11 dan fungsi 3.12. fungsi 3.13 ditujukan untuk mendistribusikan persebaran UE berdasarkan pada SBS terdekat dengan mempertimbangkan UE disekitarnya[26].

### 3.5.5. *Silhouette Score*

Dalam mengelompokkan UE kedalam suatu *cluster* yang didasarkan pada *centroid/eligible* UE yang memiliki radius dalam penentuan *cluster*, tentunya ada UE yang berada pada garis radius atau dekat dengan garis radius. Adanya UE tersebut terkadang mengakibatkan kesalahan *clustering* sehingga diperlukan model untuk mengetahui penyimpangan *cluster* yang terjadi. Model tersebut nantinya akan membandingkan suatu UE dengan *cluster* yang telah terbentuk. Seberapa cocok suatu UE dikelompokkan dengan *cluster*-nya dibandingkan dengan *cluster* lain[27]. *Silhouette score* dapat dihitung menggunakan perhitungan sebagai berikut:

Pertama menghitung kecocokan UE dengan *cluster*-nya :

$$a(i) = \frac{1}{|C_I|-1} \sum_{j \in C_I, i \neq j} d(i, j) \quad (3.14)$$

Kedua menghitung kecocokan UE dengan *cluster* lain :

$$b(i) = \min_{j \neq I} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j) \quad (3.15)$$

Selanjutnya menghitung *silhouette* dari  $a(i)$  dan  $b(i)$  sebagai berikut :

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, \text{ if } |C_I| > 1 \quad (3.16)$$

Pada persamaan 3.14 dimana  $i$  merupakan UE, sedangkan  $|C_I|$  merupakan semua UE yang berada didalam *cluster* dan  $d(i, j)$  merupakan jarak antara UE dengan UE lainnya didalam *cluster* yang sama. Setelah mendapatkan nilai kecocokan suatu UE dengan *clusternya*. Pada persamaan 3.15 menghitung kecocokan suatu UE dengan *cluster* yang lain, dimana  $j$  merupakan UE yang berada

pada *cluster* yang berbeda dengan  $i$ , sedangkan  $|C_j|$  merupakan variabel yang mewakili semua UE dalam *cluster* lain. Persamaan 3.16 dimana persamaan 3.14 dan 3.15 dioperasikan untuk mendapatkan nilai silhouette pada semua *cluster* yang terbentuk. Biasanya *silhouette score* didasarkan pada rentang nilai  $-1 \leq s(i) \leq 1$  namun jika mendekati nilai 1 maka pengelompokkan akan dianggap semakin bagus, karena  $a(i) \ll b(i)$  yang mana nilai  $a(i)$  menunjukkan seberapa tidak cocok suatu UE dengan *cluster*-nya sendiri dan  $b(i)$  menunjukkan seberapa cocok suatu UE dengan *cluster*-nya sendiri[28].