BAB II
STUDI LITERATUR

2.1 Uang Kuliah Tunggal

Sejak tahun 2013 pemerintahan indonesia menerapkan kebijakan baru mengenai sistem pembayaran biaya kuliah bagi seluruh perguruan tinggi (PTN) di Indonesia, yaitu dari sistem Sumbangan Pengembangan Manajemen Pendidikan (SPMA) dengan uang pangkalnya menjadi sistem pembayaran tanpa uang pangkal yang disebut dengan Uang Kuliah Tunggal (UKT).

Dalam Peraturan Menteri Pendidikan dan Kebudayaan RI (PERMENDIKBUD) nomor 55 tahun 2013 tentang Biaya Kuliah Tunggan dan Uang Kuliah Tunggal pada Perguruan Tinggi Negeri di Lingkungan Kementerian Pendidikan dan Kebudayaan RI menimbang dan menetapkan beberapa pasal.

Pasal 1 berisikan tentang:

1. Biaya kuliah tunggal meruapakan kesuluran biaya operasional per mahasiswa per semester pada program studi di perguruan tinggi negeri.
2. Biaya kuliah tunggal digunakan sebagai dasar penetapan biaya yang dibebankan kepada mahasiswa masyarakat dan pemerintahan.
3. Uang kuliah tunggal merupakan sebagaian biaya kuliah tunggal yang ditanggung setiap mahasiswa berdasarkan kemampuan ekonominya.
4. Uang kuliah tunggal sebagaimana dimaksud pada ayat (3) ditetapkan berdasarkan biaya kuliah tunggal dikurangi biaya yang ditanggung oleh Pemerintah.

Pasal 2 berisikan tentang:

Uang kuliah tunggal sebagaiman yang diamksud dalam Pasal 1 ayat (2) terdiri atas beberapa kelompok yang ditentukan berdasarkan kelompok kemampuan ekonomi masyarakat.

Pasal 3 berisikan tentang:

Biaya kuliah tunggal dan uang kuliah tunggal sebagaimana dimaksud dalam Pasal 1 dan Pasal 2 tercantum dalam Lampiran yang merupakan bagian tidak terpisahkan dengan Peraturan Menteri ini.

Pasal 4 berisikan tentang:

1. Uang kuliah tunggal kelompok 1 sebagaimana dimaksud dalam Lampiran diterapkan paling sedikit 5 (lima) persen dari jumlah mahasiswa yang diterima disetiap perguruan tinggi negeri.
2. Uang kuliah tunggal kelompok II sebagaimana dimaksud dalam lampiran diterapkan paling sedikit 5 (lima) persen dari jumlah mahasiswa yang diterima di setiap perguruan tinggi negeri.

Pasal 5 berisikan tentang:

Perguruan tinggi negeri tidak boleh memungut uang pangkal dan pungutan lain selain uang kuliah tunggal dari mahasiswa baru program Sarjana (S1) dan program diploma mulai tahun akademik 2013 – 2014.

Pasal 6 berisikan tentang:

Peruguran tinggi negeri dapat memungut diluar ketentuan uang kuliah tunggal dari mahasiswa baru program Sarjana (S1) dan program diploma *nonreguler* paling banyak 20 (dua puluh) persen dari jumlah mahasiswa baru mulai tahun akademik 2013 – 2014.

Pasal 7 berisikan tentang:

Biaya kuliah tunggal dan uang kuliah tunggal sebagaimana dimaksud dalama Pasal 3 berlaku mulai tahun akademik 2013 – 2014.

Pasal 8 berisikan tentang:

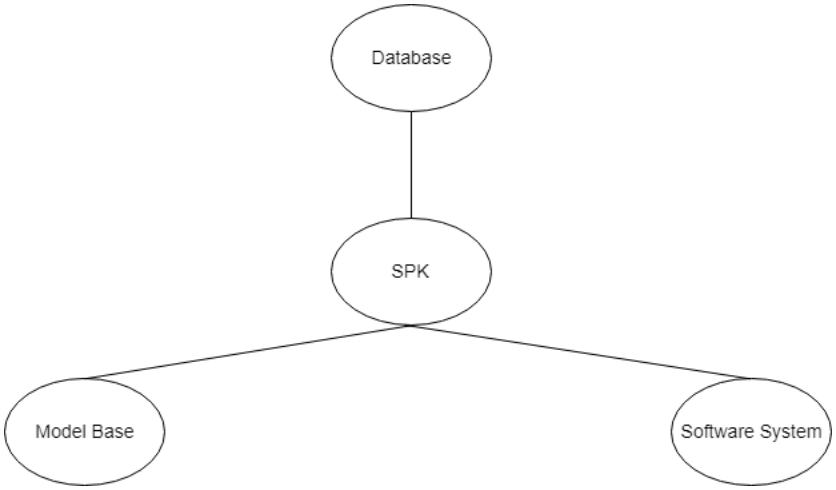
Peraturan Menteri ini mulai beralaku pada tanggal diundangkan.

2.2 Pengertian Sistem Pendukung Keputusan

Pada dasarnya Sistem Pendukung Keputusan (SPK) adalah pengembangan dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Interaktif dengan tujuan untuk memudahkan integrasi antar komponen dalam proses pengambilan keputusan manajerial sehingga kredibilitas instansi tersebut semakin lebih baik.

SPK merupakan salah satu sistem aplikasi yang sangat terkenal di kalangan manajemen organisasi. Sistem Pendukung Keputusan dirancang untuk membantu manajemen dalam proses pengambilan keputusan. SPK memadukan data dan pengetahuan untuk meningkatkan efektifitas dan efisiensi dalam proses pengambilan keputusan tersebut [5]. SPK merupakan sistem informasi interaktif yang menyediakan informasi, permodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambil keputusan dalam situasi semi terstruktur maupun tidak terstruktur [6].
Jika pada pemrosesan tradisional, pengambilan keputusan dilakukan melalui perhitungan iterasi secara manual, SPK menawarkan informasi pendukung keputusan dengan melakukan perhitungan yang cepat. Dalam mendukung keputusan tersebut, SPK mempresentasikan permasalahan manajemen dalam bentuk kuantitatif [7].

Secara garis besar, SPK dibangun atas tiga komponen utama, yaitu *database, model base,* dan *software system*. Sistem *database* berisi kumpulan dari semua data bisnis yang dimiliki oleh perusahaan, baik yang berasal dari transaksi sehari-hari, maupun data dasar (*master file*). Basis model (*model base*) merupakan komponen *software* yang terdiri dari model-model yang digunakan dalam rutinitas komputasional. Penggabungan antara dua komponen sebelumnya yaitu *database*t dan *model base* yang disatukan dalam komponen ketiga *software system*[8]. Skema tiga komponen SPK tersebut dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Skema komponen SPK

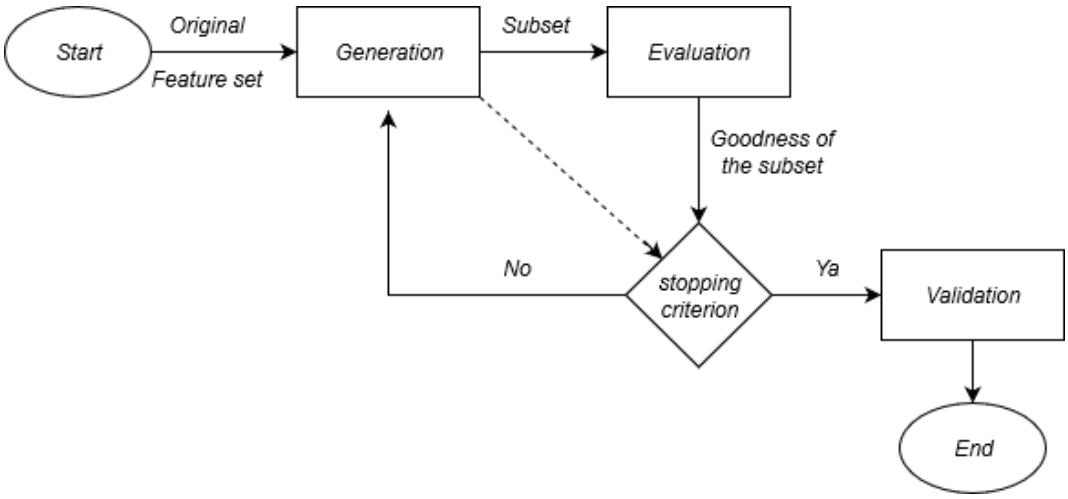
Bagaimanapun juga SPK tidak ditekankan untuk membuat suatu keputusan. SPK hanya berfungsi sebagai alat bantu manajemen dalam proses pengambilan keputusan. Jadi, SPK tidak dimaksudkan untuk menggantikan fungsi pengambil keputusan dalam proses pengambilan keputusan. Sistem ini dirancang hanyalah untuk membantu pengambil keputusan dalam melaksanakan tugasnya.

2.3 *Feature Selection*

Feature selection adalah suatu proses yang mencoba untuk menemukan sub-himpunan dari himpunan fitur yang tersedia untuk meningkatkan aplikasi dari suatu algoritma pembelajaran. *Feature selection* digunakan dibanyak area aplikasi

sebagai alat untuk menghilangkan fitur yang tidak relevan atau fitur berlebihan. Sebuah fitur dikatakan tidak relevan jika memberikan sedikit informasi, sedangkan sebuah fitur dikatakan berlebihan jika informasi yang diberikan adalah informasi yang terkandung dalam fitur lain (tidak memberikan informasi baru). Ada empat langkah yang dilakukan dalam *feature selection* [4] yaitu:

1. Prosedur generasi (pembangkitan), untuk menghasilkan calon subhimpunan berikutnya dapat dilakukan dengan beberapa cara yaitu : lengkap, heuristik dan acak.
2. Evaluasi fungsi, untuk mengevaluasi subhimpunan, dengan cara mengukur jarak, informasi, konsistensi, ketergantungan, dan mengukur tingkat kesalahan klasifikasi.
3. Kriteria penghentian, untuk memutuskan kapan harus berhenti, dengan cara melihat nilai ambang batas (*threshold*), diawali dengan sejumlah pengulangan dan sebuah ukuran subhimpunan fitur terbaik.
4. Prosedur validasi, untuk memeriksa apakah subhimpunan valid. (opsional). Proses dalam *feature selection* tersebut dapat dituangkan dalam skema berikut:



Gambar 2. 2 Proses *Feature Selection*

2.3.1 Algoritma *ReliefF*

ReleifF adalah perbaikan metode dari *relief*, yaitu metode estimasi bobot sebuah fitur. Semakin besar bobot sebuah fitur, maka dianggap semakin relevan fitur tersebut dengan *output*. Namun, *relief* sudah lama tidak digunakan lagi karena ketidakstabilan akurasi yang dihasilkan, hal ini disebabkan karena ketidakmampuannya untuk mengambil dan mengevaluasi sampel berulang kali dengan bobot fitur yang sama. Sedangkan *ReliefF* dapat mengevaluasi nilai fitur dengan berulang kali mengambil sampel *instance* dan mempertimbangkan nilai fitur yang diberikan untuk *instance* terdekat dari kelas yang sama dan yang berbeda [3]. Evaluasi atribut ini memberikan bobot untuk masing-masing fitur berdasarkan kemampuan fitur untuk membedakan antar kelas, dan kemudian memilih fitur-fitur yang nilainya melebihi *threshold* yang ditetapkan sebelumnya sebagai fitur yang relevan [9]. Pemilihan atribut dilakukan dengan menghitung perbedaan bobot untuk tiap fitur yang terpilih secara acak dengan fitur yang terpilih sebagai *near hit* (tetangga terdekat fitur terpilih pada kelas yang sama) dan *near miss* (tetangga terdekat fitur terpilih pada kelas yang berbeda) [11]. Perhitungan *threshold* dilakukan berdasarkan pada probabilitas *nearest neighbors* dari dua kelas berbeda dengan nilai yang berbeda untuk sebuah fitur dan probabilitas dua *nearest neighbors* dari kelas yang sama dan fitur dengan nilai yang sama. Semakin tinggi perbedaan nilai antara dua probabilitas, semakin signifikan pula fitur tersebut. Karakteristik *ReliefF* yaitu tidak hanya dapat menggunakan data diskrit saja dalam melakukan pemeringkatan fitur, tetapi juga *ReliefF* dapat menangani data numerik. Adapun algoritma *ReliefF* menurut Yang pada Muhamad adalah sebagai berikut [12]:

Input : sebuah *training set* D, jumlah iterasi m, jumlah k *nearest neighbors*, jumlah fitur n, predefine bobot fitur *threshold* δ.

Output : subset fitur S yang didasari oleh fitur yang bobotnya lebih besar dari *threshold* δ.

Langkah 1.

Diasumsikan S=Ø, set semua bobot fitur W(F_t)=0, t=1,2,…,n.

Langkah 2.

(1) Pilih *instance* R dari D secara acak.

(2) Cari k *nearest neighbors* H_i (i=1,2,…,k) dari kelas yang sama dan k *nearest neighbors* M_i (C)(i=1,2,…,k) dari setiap kelas C yang berbeda.

(3) For t = 1 to n do

$$W(F_t) = W(F_t) - \sum_{i=1}^{k} \frac{\text{diff(Ft,R,Hi)}}{(mk)}$$

$$+ \sum_{c=\notin ClassR} \frac{\left(\frac{\text{P(C)}}{1-P(Class(R))} \sum_{i=1}^{k} \text{diff(Ft,R,Mi(C))}\right)}{(mk)} \quad \text{..... (2.1)}$$

Langkah 3. For *t* = 1 to n do

If *W(Ft) > δ, then add fitur Ft to S.*

W(Ft) merupakan bobot tiap fitur, *P(C)* adalah distribusi probabilitas class C, Class(R) adalah kelas yang masuk ke dalam kategori *R*, sedangkan *Mi(C)* menunjukkan *i near miss* dari *R* dalam kelas *C*, *diff(Ft,R1,R2)* menunjukkan perbedaan *R1* dan *R*2 pada Ft.

jika Ft adalah data diskrit:

$$diff(Ft, R1, R2) = \begin{cases} 0; \text{R1[Ft]} = \text{R2[Ft]} \\ 1; \text{R1[Ft]} \neq \text{R2[Ft]} \end{cases} \quad \text{..... (2.2)}$$

jika Ft adalah data kontinu:

$$diff(Ft, R1, R2) = \frac{|\text{R1[Ft]} - \text{R2[Ft]}|}{\max(\text{Ft}) - \min(\text{Ft})} \quad \text{..... (2.3)}$$

$R1$ dan $R2$ adalah dua sampel, $R1[Ft]$ dan $R2[Ft]$ adalah nilai fitur dari masing-masing $R1$ dan $R2$ pada Ft.

Threshold (ambang batas) merupakan nilai batas relevan untuk pemilihan fitur optimal. Nilai *threshold* berada pada interval 0 sampai 1 dan penggunaannya bersifat independen (tergantung pada pengguna). Dalam algoritma *Relief, threshold* akan dibandingkan dengan nilai *weight* (bobot) dari suatu fitur. Apabila suatu fitur memiliki nilai bobot lebih besar dari *threshold* yang digunakan maka fitur tersebut merupakan fitur optimal sedangkan jika nilai bobot fitur lebih kecil sama dengan dari *threshold* maka fitur tersebut tidak akan dipilih karena tidak termasuk dalam kategori fitur optimal [13].

2.4 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar[10].

Istilah data mining sebenarnya mulai dikenal sejak tahun 1990, pada saat itu pemanfaatan data sangatlah penting dalam berbagai bidang seperti akademik, bisnis, hingga medis. Data mining dapat diterapkan pada berbagai bidang yang memiliki sejumlah data, namun karena wilayah penelitian beserta sejarah yang belum lama, maka data mining masih mengalami perdebatan mengenai posisinya dibidang pengetahuan. Data mining merupakan perpaduan antara statistik, kecerdasan buatan, dan riset basis data [17].

1. Statistik

   Akar yang paling tua diantara lainnya, mungkin tanpa adanya statistik data mining juga tidak akan pernah ada. Dengan menggunakan statistik klasik data dapat diringkas kedalam *exploratory data analysis* (EDA). EDA berguna untuk mengidentifikasi hubungan sistematis antar variable ketika tidak ada cukup informasi alami yang dibawanya.

2. Kecerdasan Buatan atau *artificial intelligence* (AI)

Bidang ilmu ini berbeda dari bidang sebelumnya. Teorinya dibangun berdasarkan teknik heuristik sehingga AI berkontribusi terhadap teknik pengolahan informasi berdasarkan pada model penalaran manusia.

3. Pengenalan Pola

   Data mining juga meropakan turunan dari bidang pengenalan pola,tapi hanya pada pengolahan data dari basis data. Data yang diambil dari basis data kemudian diolah bukan dalam bentuk relasi, namun diolah dalam bentuk normal pertama biarpun begitu data mining memiliki ciri khas yaitu pencarian pola asosiasi dan pola sekuensial.

4. Sistem Basis Data

   Akar bidang ilmu yang terakhir dari data mining yang menyediakan informasi berupa data yang akan “digali” menggunakan metode-metode yang ada pada data mining.

2.4.1 *Clustering*

Pada dasarnya *cluster*ing terhadap data adalah suatu proses untuk mengelompokkan sekumpulan data tanpa suatu atribut kelas yang telah didefinisikan sebelumnya, berdasarkan pada prinsip konseptual *cluster*ing yaitu memaksimalkan dan juga meminimalkan kemiripan intra kelas. Misalnya, sekumpulan obyek-obyek komoditi pertama-tama dapat di *cluster*ing menjadi sebuah himpunan kelas-kelas dan lalu menjadi sebuah himpunan aturan-aturan yang dapat diturunkan berdasarkan suatu klasifikasi tertentu.

Proses untuk mengelompokkan secara fisik atau abstrak obyek-obyek ke dalam bentuk kelas-kelas atau obyek-obyek yang serupa, disebut dengan *clustering* atau *unsupervised classification*. Melakukan analisa dengan *cluster*ing, akan sangat membantu untuk membentuk partisi-partisi yang berguna terhadap sejumlah besar himpunan obyek dengan didasarkan pada prinsip "*divide and conquer*" yang mendekomposisikan suatu sistem skala besar, menjadi komponen-komponen yang

lebih kecil, untuk menyederhanakan proses desain dan implementasi. Perbedaan utama antara *Clustering Analysis* dan klasifikasi adalah bahwa *Clustering Analysis* digunakan untuk memprediksi kelas dalam format bilangan real dan pada format katagorikal atau Boolean.

2.5 Logika *Fuzzy*

Logika *fuzzy* diperkenalkan pertama kali pada tahun 1965 oleh Prof Lutfi A. Zadeh seorang peneliti di Universitas California di Barkley dalam bidang ilmu komputer [18]. Professor Zadeh beranggapan logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkanlah logika *fuzzy* yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegasdan logika *fuzzy* terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika *fuzzy*, keanggotaan elemen berada di interval [0,1]. Logika *fuzzy* menjadi alternatif dari berbagai sistem yang ada dalam pengambilan keputusan karena logika *fuzzy* mempunyai kelebihan sebagai berikut:

a. Logika *fuzzy* memiliki konsep yang sangat sederhana sehingga mudah untuk dimengerti.
b. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian.
c. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.
d. Logika *fuzzy* mampu mensistemkan fungsi-fungsi non-linier yang sangat kompleks.
e. Logika *fuzzy* dapat mengaplikasikan pengalaman atau pengetahuan dari para pakar.
f. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.

g. Logika *fuzzy* didasarkan pada bahasa sehari-hari sehingga mudah dimengerti.

Logika *fuzzy* memiliki beberapa komponen yang harus dipahami seperti himpunan *fuzzy*, fungsi keanggotaan, operator pada himpunan *fuzzy*, inferensi *fuzzy* dan defuzzifikasi [19].

2.5.1 *Fuzzy C-Means*

Metode *Fuzzy C-Means* (FCM) didasakan pada teori logika *fuzzy* yang diperkenalkan pertamakali oleh Lotfi Zadeh. FCM merupakan suatu teknik pengelompokan data yang keberadaan tiap-tiap titik data suatu *cluster* ditentukan oleh nilai keanggotaan. Nilai keanggotaan tersebut akan mencakup bilangan real pada interval 0-1. Tujuan dari algoritma FCM yaitu untuk mendapatkan pusat *cluster* yang nantinya akan digunakan untuk mengetahui data yang masuk kedalam *cluster*. Berikut adalah penjabaran dari algoritma FCM [14]:

1. Menentukan data yang akan *cluster X*, berupa matriks berukuran *n* x *m* (*n*=jumlah sampel data, *m*=atribut setiap data). X_{ij}=data sampel ke-*i* (*i*=1,2,...,*n*), atribut ke-j (*j*=1,2,...,*m*).

2. Tentukan jumlah *cluster* (c), pangkat (w), maksimum iterasi (MaxIter), error terkecil yang diharapkan (ζ), fungsi obyektif awal (P0=0), iterasi awal (t=1).

3. Bangkitkan bilangan random μ_{ik}, *i*=1,2,...,n; *k*=1,2,...,c; sebagai elemen elemen matriks partisi awal U. Matriks partisi (U) pada pengelompokan *fuzzy* memenuhi kondisi sebagai berikut [14]:

$$\mu_{ik} \varepsilon [0,1]; \qquad 1 \leq i \leq n; \qquad 1 \leq k \leq c$$

μ_{ik} adalah derajat keanggotaan yang merujuk pada seberapa besar kemungkinan suatu data bisa menjadi anggota ke dalam suatu *cluster*. Hitung jumlah setiap kolom (atribut):

$$Qi = \sum_{k=1}^{c} \mu_{ik} \quad \text{..... (2.4)}$$

$Q_i = \mu_{i1} + \mu_{i2} + \cdots + \mu_{ic}$

dengan $i = 1,2, \ldots, n$

4. Hitung pusat *cluster* ke-k: V_{kj}, dengan k=1,2,...,c; dan j=1,2,...,m

$$V_{kj} = \frac{\sum_{i=1}^{n}((\mu_{ik})^{w} x_{ij})}{\sum_{i=1}^{n}(\mu_{ik})^{w}} \quad \text{..... (2.5)}$$

5. Hitung fungsi obyektif pada iterasi ke-t, P_t:

   Fungsi obyektif digunakan sebagai syarat perulangan untuk mendapatkan pusat *cluster* yan tepat. Sehingga diperoleh kecenderungan data untuk masuk ke *cluster* mana pada step akhir. Untuk iterasi awal nilai t=1.

$$P_t = \sum_{i=1}^{n} \sum_{k=1}^{c} \left(\left[\sum_{j=1}^{m} \left(X_{ij} - V_{kj} \right)^2 \right] (\mu_{ik})^w \right) \quad \text{..... (2.6)}$$

6. Hitung perubahan matriks partisi:

$$\mu_{ik} = \frac{\left[\sum_{j=1}^{m} (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^{c} \left[\sum_{j=1}^{m} (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \quad \text{..... (2.7)}$$

7. Cek kondisi behenti:

   a. $|P_t - P_t\text{-}1| < \zeta$) atau (t>MaxIter) maka berhenti;

   b. Jika tidak, iterasi dinaikkan t=t+1, ulangi langkah ke-4.

2.6 Indeks Validitas

Pada konsep *fuzzy clustering*, suatu anggota dapat menjadi anggota beberapa *cluster* sekaligus menurut derajat keanggotaannya. Dalam proses *clustering* selalu mencari solusi terbaik untuk parameter yang didefinisikan. Akan tetapi, dalam beberapa hal terdapat *cluster* yang tidak sesuai dengan data. Untuk menentukan jumlah *cluster* yang optimal maka perlu adanya pengukuran indeks validitas.

2.6.1 *Partition Coefficient* (PC)

Partition Coefficient (PC) merupakan metode yang mengukur jumlah *cluster* yang mengalami *overlap*. Indeks PC mengukur validitas *cluster* menggunakan rumus :

$$PC(c) = \frac{1}{N}(\sum_{i=1}^{C}\sum_{j=1}^{N}(\mu_{ij}{}^2) \quad \text{..... (2.8)}$$

c : Jumlah *cluster*

N : Jumlah data

μ_{ij} : Nilai keanggotaan data ke-j pada *cluster* ke-i

$PC(c)$: Nilai indeks PC pada *cluster* ke-c

Pada pengukuran indeks validitas menggunakan indeks PC, *cluster* yang paling optimal ditentukan beradsarkan nilai PC yang paling besar [15]. indeks PC digunakan untuk mengukur jumlah *cluster* yang mengalami *overlapping* ditransformasikan menjadi fungsi penurunan yang dapat menyesuaikan nilai *fuzziness* secara otomatis.