BAB II

LANDASAN TEORI

Metode yang dilakukan setiap daerah dalam pengambilan keputusan penerimaan beras bagi keluarga miskin masih menggunakan cara manual dan data base yang digunakan dalam bentuk kertas, sehingga membutuhkan waktu yang lama untuk pengolahan, dan kendala terbesar adalah kesulitan dalam penyimpanan atau pencarian arsip jika ingin dicocokan dengan informasi atau pedoman baru. Disamping itu, pengambilan keputusan untuk menentukan kriteria penerima beras yang sudah terjadi biasanya tidak mengacu pada kriteria-kriteria yang telah ditentukan. Memperhatikan permasalahan ini, maka diperlukan sebuah sistem informasi yang baik untuk mencegah kesalahan dan kecurangan, yaitu dengan Sistem Pendukung Keputusan (SPK), sistem ini merupakan bagian dari Sistem Informasi berbasis komputer. Sistem ini diharapkan dapat memberikan kemudahan dalam pemetaan pemberian bantuan beras miskin yang tepat dan akurat.

Beberapa teori yang dapat menunjang pemecahan masalah yang ada kaitannya dengan sistem pemetaan pemberian batuan beras raskin. Sebagaimana telah diuraikan pada bab sebelumnya, dalam penelitian ini penulis akan menggunakan metode Algoritma K-Means. Terlebih dahulu penulis ingin memberikan beberapa pengertian-pengertian digunakan sebagai landasan dasar dalam penulisan ini, sebagai berikut :

2.1 K-MEANS

K-Means merupakan metode klasterisasi yang paling terkenal dan banyak digunakan di berbagai bidang karena sederhana, mudah diimplementasikan, memiliki kemampuan untuk mengklaster data yang benar, mampu menangani data outlier, dan kompleksitas waktunya linwar O (nKT) dengan n adalah jumlah dokumen, k adalah jumlah klaster, dan T adalah jumlah iterasi. K-Means merupakan metode pengklasteran secara partitioning yang memisahkan data ke dalam kelompok yang berbeda. Dengan partitioning secara iteratif, K-Means mampu meminimalkan rata-rata jarak setiap data ke klasternya. Metode K-Means dikembangkan oleh Mac Queen pada tahun 1967.

Sejalan dengn hal tersebut, disebutkan juga bahwa K-means merupakan suatu algoritma pengklasteran yang cukup sederhana yang mempartisi dataset ke dalam beberapa klaster k. Algoritmanya cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan (Wu & Kumar, 2009). Prinsip utama dari teknik ini adalah menyusun k buah partisi / pusat massa (*centroid*) / ratarata (*mean*) dari sekumpulan data. Algoritma K-means dimulai dengan pembentukan partisi klaster di awal kemudian secara iteratif partisi klaster ini diperbaiki hingga tidak terjadi perubahan yang signifikan pada partisi klaster (Witten, Eibe, & Hall, 2011).

Karakteritik dari Algoritma K-Means salah satunya adalah sangat sensitif dalam penentuan titik pusat awal kluster karena K-Means membangkitkan titik pusat klaster awal random tersebut memdekati solusi akhir pusat klaster, K-Means mempunyai posibilitas yang tinggi untuk menemukan titik pusat klaster yang tepat. sebaliknya jika awal titik pusat tersebut jauh dari solusi akhir pusat klaster, maka besar kemungkinan dapat.menyebabkan hasil pengklasteran yang tidak tepat. Akibatnya K-Means tidak menjamin hasil pengklasteran yang unik. Hal inilah yang menyebabkan K-Means sulit untuk memcapai optimum global, akan tetapi hanya optimum lokal. Selain itu, Algoritma K-Means hanya bisa digunakan untuk data yang atributnya bernilai numerik.

Beberapa penelitian yang telah dilakukan mengenai masalah sensitifitas inisialisasi jumlah cluster (k), danalgoritma yang digunakan. Penelitian yang dilakukan oleh (Deelers & Auwatanamongkol, 2007) mengusulkan sebuah algoritma partisi data untuk menghitung awal pusat kluster. Partisi data mencoba membagi ruang data kedalam sel kecil atau kelompok, mana yang jarak intercluster sebesar mungkin dan jarak intracluster sekecil mungkin. Sel dipartisi satu persatu sampai jumlah sel sama dengan jumlah kluster (k) yang telah ditetapkan, dan pusat-pusat sel k menjadi awal pusat kluster untuk K-means. Hasil percobaan menunjukan bahwa algoritma partisi data bekerja lebih baik dibandingkan dengan inisialisasi pusat kluster secara acak dari sebagian kasus eksperimental dan dapat mengurangi waktu running algoritma K-means untuk dataset yang

besar. (Yi et al., 2010), mengusulkan sebuah algoritma partisi data untuk memperbaiki awal pusat cluster yaitu Algoritma awal pusat cluster berbasis kepadatan (density). Algoritma ini menggunakan fungsi gaussian untuk memenuhi konsistensi global fitur clustering.

Algoritma yang diusulkan memilih titik kepadatan terbesar sebagai titik pusat awal pertama dari dataset, kemudian menentukan pusat awal kedua menggunakan metode yang sama dari dataset sehingga menghapus titik pertama dan tetangganya. Proses ini berlanjut sampai set M awal berisi k poin. Hasil percobaan menunjukan bahwa algoritma yang diusulkan sangat meningkatkan kualitas dan stabilitas

Algoritma K-means. (Zhang & Fang, 2013) melakukan penelitian dalam perbaikan algoritma K-means untuk mengoptimalkan inisialisasi pusat cluster. Dengan menemukan satu set data yang mencermikan karakteristik distribusi data sebagai pusat awal cluster untuk mendukung pembagian data ke batas yang terbaik. Hasil percobaan didapatkan hasil akurasi algoritma perbaikan K-means meningkat secara signifikan dibandingkan dengan algoritma K-means tradisional, dan algoritma yang diusulkan menunjukan bahwa hasil setiap cluster lebih kompak.

Pada penelitian ini kami mengkombinasikan algoritma cluster dinamik dengan algoritma K-Means untuk menghasilkan kualitas cluster yang optimal dalam sistem pemetaan pemberian bantuan beras miskin.

2.2 ALGORTITMA CLUSTER

Algoritma cluster dinamik pada algoritma K-means dinamik dapat dilihat pada Gambar 1. Algoritma yang diusulkan mencari jumlah cluster yang dijalankan berdasarkan kualitas cluster keluaran. Diawal cara kerja sama dengan algoritma k-means, diakhir akan dilakukan perhitungan intra dan inter cluster, jika jarak intra lebih kecil dan jika jarak intra lebih besar, maka algoritma menghitung cluster baru dengan menambahkan counter k dengan satu atau k=k+1 disetiap iterasi sampai memenuhi batas validitas kualitas cluster yang berkualitas (M & Hareesha, 2012). Berikut tahapan Algoritma K-means:

1. Membuat partisi sejumlah k dari segmentasi yang akan dibentuk.
2. Pilih secara acak k point untuk dijadikan pusat cluster
3. Menghitung jarak data yang lain dengan pusat cluster
4. Mengisi setiap obyek dalam dataset kedalam segmen terdekat.
5. Kalkulasi ulang setiap segmentasi yang terbentuk
6. Ulangi langkah hingga data di dalam segmentasi tidak berubah

Istilah inter adalah minimum jarak antar pusat cluster, inter digunakan untuk mengukur pemisahan antar cluster, yang didefinisikan sebagai :

$$inter = min\{\|m_k - m_{kk}\|\} \quad \forall\, k = 1,2,\ldots,K-1 \; dan \; k = k+1,\ldots,K \quad \text{..... (1)}$$

Istilah intra digunakan untuk mengukur kekompakan dari suatu kelompok. Standar deviasi digunakan untuk memeriksa kedekatan titik data setiap cluster, dan dihitung sebagai :

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - X_m)^2} \quad \text{..... (2)}$$

2.3. METODE *WEIGHTED PRODUCT* (WP)

Weighted Product (WP) adalah metode menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatan dulu dengan bobot atribut yang bersangkutan. Proses ini sama halnya dengan proses normalisasi.

Proses ini Ai diberikansebagaiberikut :

$$S_i = \prod_{j=1}^{n} X_{ij}^{w_j} \quad \text{..... (2)}$$

dimana :

S : *Preferensi* alternatif dianologikan sebagai vektor S

X : Nilai kriteria

W : Bobot kriteria/subkriteria

i : Alternatif

j : Kriteria

n : Banyaknya kriteria

Dimana ∑wj = 1.. wj adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

Preferensi relatif dari setiap alternatif, diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; \qquad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

dimana :

V : Preferensi alternatif dianalogikan sebagai vektor V

X : Nilai Kriteria

W : Bobot kriteria/subkriteria

i : Alternatif

j : Kriteria

n : Banyaknya kriteria

* : Banyaknya kriteria yang telah dinilai pada vektor S

Dimana ∑wj= 1... wj adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negative untuk atribut biaya. (Kusumadewi, Hartati, Harjoko, danWardoyo, 2006)

2.4. Pemrograman Berorientasi Objek

2.4.1. *Unified Modeling Language (UML)*

UML (*Unified Modeling Language)* merupakan pengganti dari metode analisa berorientasi *objek* dan *design* berorientasi objek (O) OA&D) yang dimunculkan sekitar akhir tahun 80-an dan awal 90-anpakan gabungan dari metode *Booch, Rumbaugh* (OMT) dan Jacobson. Teapi UML ini akan mencangkup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan

standarisasi proses dengan OMG(*Object Managemant Group)* dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara tepat. Bahasa pemodelan merupakan bagian terpenting dari metode, ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain. UML merupakan bahasa standar untuk penulisan blueprint software yang digunakan untuk visualisasi, spesifikasi, pembentukan dari pendokumentasian alat-alat dari sistem perangkat lunak.

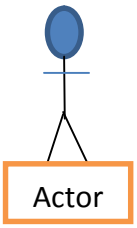
Secara mendasar UML mendefinisikan diagram sebagai berikut :

2.4.2. Use Case Diagram

Menggambarkan sejumlah *external actor* dan hubungannya ke *Use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use Case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan

bagaimana fungsi yang ada di dalam sistem. Notasi *Use Case Diagram* ditunjuk pada tabel 2.1

Tabel 2.1 Notasi *Use CAse Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*
2	_____	Association	Apa yang menghubungkan antara objek satu dengan objek lain
3	---------->	Despendency	Hubungan dimana perubahan terjadi pada suatu elemen mandiri (*independent)* akan mempengaruhi elemen yang bergantun padannya elemen yang tidak mandiri
4	<----------	Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada diatasnya

			objek induk (ancestor)
5	----------> <<Include>>	Include	Memspesifikasi bahwa *Use case* sumber secara *explicit.*
6	<---------- << Extend>>	Extend	Memspsifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
7		System	Memspesifikasikan paket yang menampilakan sistem secara terbatas
8		Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghsilkan suatu hasil yang terukur bagi suatu aktor.

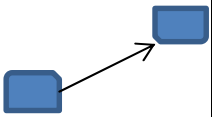
2.4.3. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi

sehingga dapat juga digunakan untuk aktifitas lainta seperti *use case* atau interaksi. Notasi *Activity Diagram* ditunjuk pada Tabel 2.2

Tabel 2.2 Notasi *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Meperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	*State* dari sistem yang mencerminan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali
4		*Actifity Final Node*	Akhir dari banyak aliran dalam diagran aktivity
5		*Flow Final Node*	Menunjukkan akhir dari aliran tunggal dalam diagram aktivitas
6		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

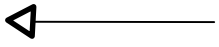
7		*Swimlane*	Partisi untuk mengisi aksi dan aktifitas
8		*Control Node*	Menunjukkan aliran kendali dari aksi ke aksi yang berikutnya
9		*Desision Node*	Digunakan untuk mempresentasikan keputusan dalam alur kendali, ini juga bisa digunakan untuk menggabungkan alur. Sebuah keputusan akan memiliki kondisi yang harus ada untuk menentukan jalan atau alur yang akan diambil

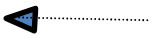
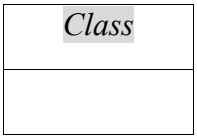
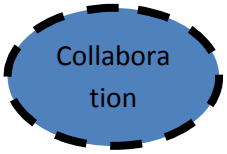
2.4.4. Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* mempresentasikan sesuatu yang ditangani oleh sistem. *Class*

dapat berhubungan dengan yang lain melalui berbagai cara : *associated (*terhubung satu sama Lain),*dependent* (satu *class* tergantung/menggunakan *classi yang lain), speciled* (satu *class* merupakan specialisasi dari *class* lainnya), atau *package* (group bersama sebagai satu unit). Notasi C*lass Diagram* ditunjuk pada Tabel 2.3.

Tabel 2.3 Notasi C*lass Diagram*

NO	*GAMBAR*	NAMA	KETERANGAN
1	_____	*Association*	Apa yang menghubungkan antara objek satu dengan yang lain
2	---------->	*Dependency*	Hubungan dimana perubahan terjadi pada suatu elemen mandiri (independent) akan menpengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		*Generalization*	Hubungan dimana abjek anak *(descendent)*

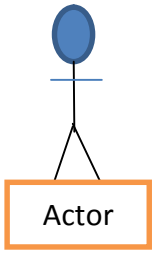
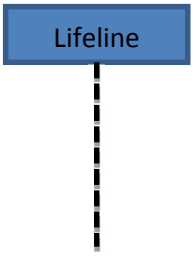
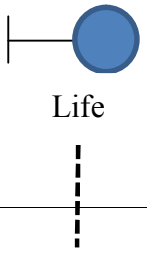
			berbagi perilaku dan strukturdata dari objek induk *(ancestor)*
4		*Realitation*	Operasi yang benar-benar dilakukan oleh suatu objek
5		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari dua objek
6		*Class*	Himpunan dari objek-objek berbagi atribut operasi yang sama
7		*Collaboration*	Deskripsi dari urutan aksi-aksi ditampilkan sistem yang menghasilkan actor

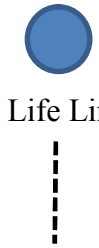
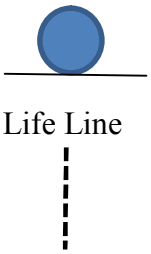
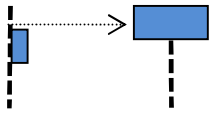
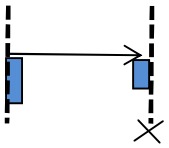
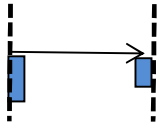
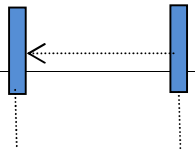
2.4.5. Sequence Diagram

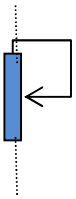
Menggambarkan kolaborasi dinamis antara sejumlah objek .kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara abject juga interaksi antara object, sesuatu yang terjadi pada

titik tertentu dalam eksekusi sisten . Notasi Squence Diagram ditunjuk pada Tabel 2.4.

Tabel 2.4. ***Sequence Diagram***

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Mempresentasikan entitas yang berada di luar sistem. Mereka bisa berupa manusia, perangkat keras atau sistem lain
2		*Life Line*	Mempresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak, entitas ini memiliki nama, stereotype atau berupa instance(menggunakan instant:class)
3		*Boundary* *Life Line*	Boundary biasanya berupa tepi sistem seperti user interface atau suatu alat yang

			berinteraksi dengan sistem lain
4		*Control* *Life Line*	Control elemen mengatur aliran dari informasi untuk sebuah skenario. Perilaku bisnis umumnya diatur oleh objek ini.
5		*Entity* *Life Line*	Entity biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau suatu objek.
6		*Create* *Message*	Digunakan untuk melakukan instalasi suatu objek
7		*Destroy* *Message*	Menghancurkan pesan yang mewakili permintaan dan siklus target garis hidup.
8		*Send* *message*	Mengirim pesan untuk mengawali eksekusi.
9		*Return* *message*	Pesan kembali ketika pesan yang lulus informasi kembali

a b	T		ke pemanggil dari mantan pesan berkorespodensi.
10 e l		Self message	Relasi ini menunjukkan bahwa suatu objek hendak memanggil dirinya sendiri
2 11 . 4 .		Alternate Combined Fragment	Memodelkan if then else blok
12 *S e*		Interaction Use	Memodelkan peryataan swift
13 *q u e n*		Loop Combined Fragment	Loop fragment

ce Dia