

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pemetaan**

Pemetaan adalah pengelompokan suatu kumpulan wilayah yang berkaitan dengan beberapa letak geografis wilayah yang meliputi dataran tinggi, pegunungan, sumber daya dan potensi penduduk yang berpengaruh terhadap sosial kultural yang memiliki ciri khas khusus dalam penggunaan skala yang tepat (Munir, 2012).

Peta adalah penggambaran dua dimensi pada bidang datar keseluruhan atau sebagian dari permukaan bumi yang diproyeksikan dengan perbandingan atau skala tertentu (Nasution, 2016).

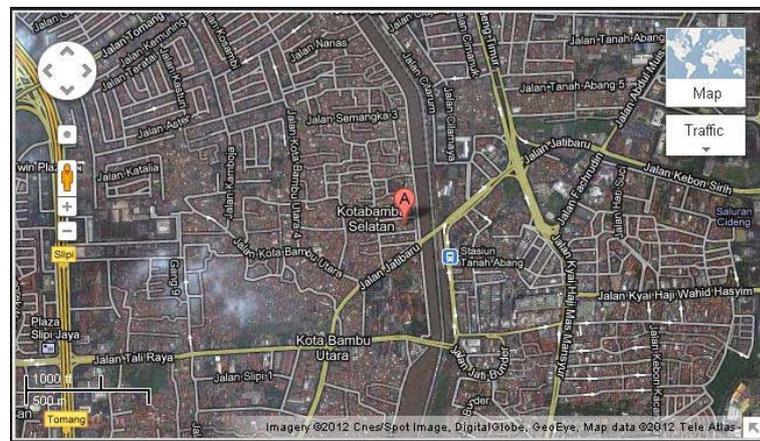
Jadi, dari dua definisi diatas dan disesuaikan dengan penelitian ini maka pemetaan merupakan proses pengumpulan data untuk dijadikan sebagai langkah awal dalam pembuatan peta, dengan menggambarkan penyebaran kondisi alamiah tertentu secara meruang, memindahkan keadaan sesungguhnya kedalam peta dasar, yang dinyatakan dengan penggunaan skala peta.

Dalam proses pembuatan peta harus mengikuti pedoman dan prosedur tertentu agar dapat dihasilkan peta yang baik, benar, serta memiliki unsur seni dan keindahan. Secara umum proses pembuatan peta meliputi beberapa tahapan dari pencarian dan pengumpulan data hingga sebuah peta dapat digunakan. Proses pemetaan tersebut harus dilakukan dengan urut dan runtut, karena jika tidak dilakukan secara urut dan runtut,

tidak akan diperoleh peta yang baik dan benar. Jenis peta berdasar bentuknya dapat dibedakan menjadi:

### 2.1.1 Peta Digital

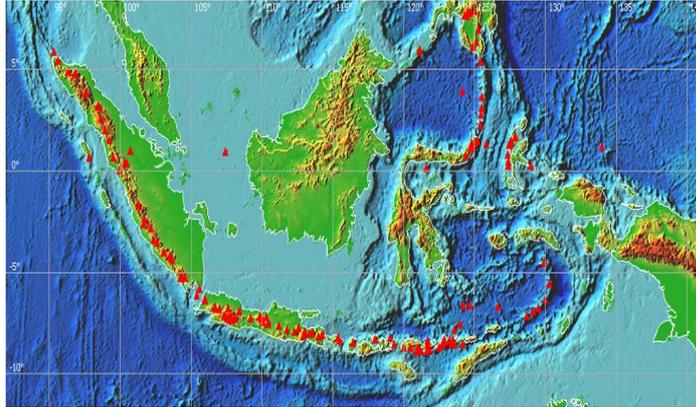
Peta yang digambarkan pada sebuah aplikasi komputer, biasanya menggunakan Sistem Informasi Geografis (SIG). Sebagai contoh : 10 objek berupa lokasi sebuah titik akan disimpan sebagai sebuah koordinat, sedangkan objek berupa wilayah akan disimpan sebagai sekumpulan koordinat.



Gambar 2.1 Peta Digital

### 2.1.2 Peta Timbul (*relief*)

Peta timbul atau *relief* adalah peta yang menggambarkan bentuk sebenarnya dari permukaan bumi. Contoh peta timbul bisa dilihat pada gambar 2.2.



Gambar 2.2 Peta Timbul

### 2.1.3 Peta Datar

Peta datar adalah peta yang digambarkan dalam bidang datar berbentuk dua dimensi. Misalnya kertas, kain atau kanvas.



Gambar 2.3 Peta Datar

Untuk Peta Digital yang menggunakan Sistem Informasi Geografis (SIG). Ada dua macam data dasar geografi, yaitu data spasial dan data atribut.

### 2.1.4 Data spasial (keruangan)

Data spasial (keruangan) adalah sebuah data yang mengacu pada posisi, obyek dan hubungan di antaranyadalam ruang bumi.

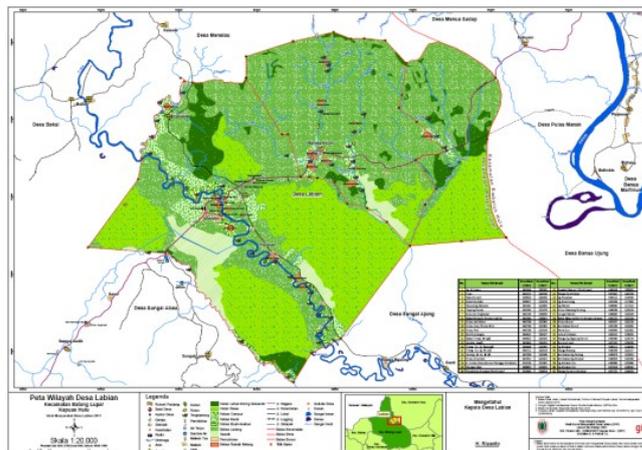
Data spasial merupakan salah satu sistem dari informasi dimana di dalamnya terdapat informasi mengenai bumi termasuk permukaan bumi, di bawah permukaan bumi, perairan, kelautan dan bawah atmosfer (Irwansyah, 2013). Sumber data spasial diantaranya adalah :

- a) Citra Satelit, data ini menggunakan satelit sebagai wahananya. Satelit tersebut menggunakan sensor untuk dapat merekam kondisi atau gambaran dari permukaan bumi. Umumnya diaplikasikan dalam kegiatan yang berhubungan dengan pemantauan sumber daya alam di permukaan bumi (bahkan ada beberapa satelit yang sanggup merekam hingga dibawah permukaan bumi), studi perubahan lahan dan lingkungan, dan aplikasi lain yang melibatkan aktifitas manusia di permukaan bumi. Kelebihan dari teknologi terutama dalam dekade ini adalah dalam kemampuan merakam cakupan wilayah yang luas dan tingkat resolusi dalam merekam obyek yang sangat tinggi. Data yang dihasilkan dari citra satelit kemudian diturunkan menjadi data tematik dan disimpan dalam bentuk basis data untuk digunakan dalam berbagai macam aplikasi. Mengenai spesifikasi detail dari data citra satelit dan teknologi yang digunakan akan dibahas dalam bab tersendiri.



Gambar 2.4 Citra Satelit

- b) Peta Analog, sebenarnya jenis data ini merupakan versi awal dari data spasial, dimana yang membedakannya adalah hanya dalam bentuk penyimpanannya saja. Peta analog merupakan bentuk tradisional dari data spasial, dimana data ditampilkan dalam bentuk kertas atau film. Oleh karena itu dengan perkembangantechnologi saat ini peta analog tersebut dapat di *scan* menjadi format digital untuk kemudian disimpan dalam basis data. Contoh peta analog bisa dilihat dibawah ini.



Gambar 2.5 Peta Analog

c) Foto Udara (*Aerial Photographs*), merupakan salah satu sumber data yang banyak digunakan untuk menghasilkan data spasial selain dari citra satelit. Perbedaannya dengan citra satelit adalah hanya pada wahana dan cakupan wilayahnya. Biasanya foto udara menggunakan pesawat udara. Secara teknis proses pengambilan atau perekaman datanya hampir sama dengan citra satelit. Sebelum berkembang teknologi kamera digital, kamera yang digunakan adalah menggunakan kamera konvensional menggunakan negatif film, saat ini sudah menggunakan kamera digital, dimana data hasil perekaman dapat langsung disimpan dalam basis data. Sedangkan untuk data lama (format foto film) agar dapat disimpan dalam basis data harus dilakukan konversi dahulu dengan menggunakan *scanner*, sehinggadihasilkan foto udara dalam format digital. Lebih lanjut mengenai spesifikasi foto udara akan dibahas dalam bab tersendiri.



Gambar 2.6 Foto Udara

- d) Data Tabular, data ini berfungsi sebagai atribut bagi data spasial. Data ini umumnya berbentuk tabel. Salah satu contoh data ini yang umumnya digunakan adalah data sensus penduduk, data sosial, data ekonomi, dll. Data tabular ini kemudian di relasikan dengan data spasial untuk menghasilkan tema data tertentu.

ID	Outlook	Temp	Humidity	Windy	Play Golf
1	Rainy	85	92	False	No
2	Rainy	80	88	True	No
3	Overcast	83	86	False	Yes
4	Sunny	70	80	False	Yes
5	Sunny	68	?	False	Yes
6	Sunny	65	58	True	No
7	Overcast	64	62	True	Yes
8	Rainy	72	95	?	No
9	Rainy	?	70	False	Yes
10	Sunny	75	72	False	Yes
11	Rainy	75	74	True	Yes
12	?	72	78	True	Yes
13	Overcast	81	66	False	Yes
14	Sunny	71	79	True	No

Gambar 2.7 Data Tabular

- e) Data Survei (Pengamatan atau pengukuran dilapangan), data ini dihasilkan dari hasil survei atau pengamatan dilapangan. Contohnya adalah pengukuran persil lahan dengan menggunakan metode survei terestris.

Tahun	Jumlah Penduduk Miskin (Ribuan)			Persentase Penduduk Miskin		
	Kota	Desa	Kota+Desa	Kota	Desa	Kota+Desa
2002	305,8	480,9	786,7	6,47	12,64	9,22
2003	309,4	546,4	855,8	6,62	12,76	9,56
2004	279,9	499,3	779,2	5,69	11,99	8,58
2005	370,2	460,3	830,5	6,56	12,34	8,86
2006	417,1	487,3	904,3	7,47	13,34	9,79
2007	399,4	486,8	886,2	6,79	12,52	9,07
2008	371,0	445,7	816,7	6,15	11,18	8,15
2009	348,7	439,3	788,1	5,62	10,70	7,64
2010	318,3	439,9	758,2	4,99	10,44	7,16

Gambar 2.8 Data Survei

### 2.1.5 Data atribut (deskripsi)

Data atribut (deskripsi) adalah data yang terdapat pada ruang atau tempat yang menerangkan suatu informasi. Data atribut diperoleh dari statistik, sensus, catatan lapangan, dan tabular (data yang disimpan dalam bentuk tabel) lainnya. Data atribut dapat dilihat dari segi kualitas, seperti kekuatan pohon, dan dapat dilihat dari segi kuantitas, seperti jumlah pohon.

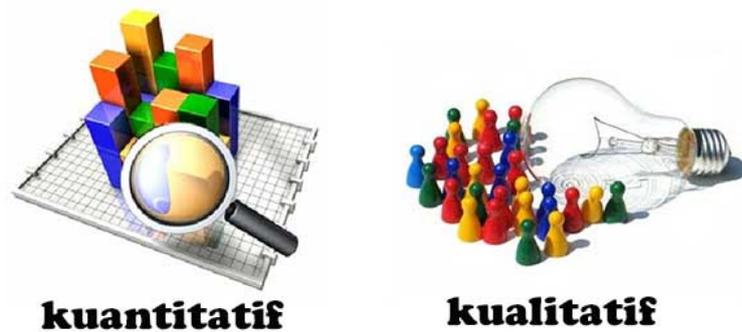
- a) Data kuantitatif, yaitu data yang berkaitan dengan sebuah penilaian angka. Data angka tersebut selanjutnya bisa di hitung secara matematis dan di proses pada sebuah sistem. Data tersebut juga bisa digunakan sebagai statistik informasi pada data pelengkap sistem informasi geografis. Salah satu contoh data kuantitatif ini msialnya: kepadatan penduduk, jumlah lokasi wisata, jarak lokasi wisata dan berbagai hal yang memberikan penafsiran berupa data angka pada objek lokasi.

No	PARTAI	KORUPTOR	SUARA 2009	INDEKS KORUPSI
1	PAN	36	6.254.580	5,76
2	PDIP	84	14.600.091	5,75
3	Partai Golkar	60	15.037.757	3,99
4	PPP	13	5.533.214	2,35
5	PKB	12	5.146.122	2,33
6	Partai Hanura	6	3.922.870	1,53
7	Partai Demokrat	30	21.703.137	1,38
8	PKPI	1	934.892	1,07
9	PBB	2	1.864.752	1,07
10	Partai Gerindra	3	4.646.406	0,65
11	PKS	2	8.206.955	0,24

Gambar 2.9 Data Kuantitatif

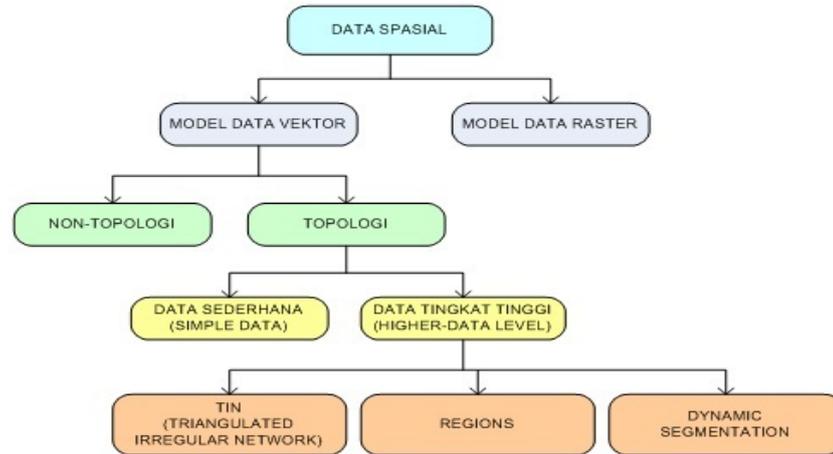
- b) Data kualitatif, yaitu data yang bersifat subjektif. Data ini adalah berupa data penafsiran yang bersifat kualitas. Contoh

data kualitatif ini misalnya, jauh, dekat, dan data-data yang bersifat subjektif.



Gambar 2.10 Data Kualitatif

Data spasial dan data atribut tersimpan dalam bentuk titik (*dot*), garis (*vektor*), *poligon* (area), dan piksel (*grid*). Data dalam bentuk titik (*dot*), meliputi ketinggian tempat, curah hujan, lokasi, dan *topografi*. Data dalam bentuk garis (*vektor*), meliputi jaringan jalan, pipa air minum, pola aliran sungai, dan garis kontur. Data dalam bentuk *poligon* (area), meliputi daerah administrasi, *geologi*, *geomorfologi*, jenis tanah, dan penggunaan tanah. Data dalam bentuk piksel (*grid*), meliputi citra satelit dan foto udara. Data dasar yang dimasukkan dalam SIG diperoleh dari tiga sumber, yaitu data lapangan (terestris), data peta, dan data penginderaan jauh (Irwansyah, 2013)



Gambar 2.11 Data Spasial

## 2.2 Clustering

*Clustering* merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan *cluster*. Objek yang di dalam *cluster* memiliki kemiripan karakteristik antar satu sama lainnya dan berbeda dengan *cluster* yang lain. Partisi tidak dilakukan secara manual melainkan dengan suatu algoritma *clustering*. Oleh karena itu, *clustering* sangat berguna dan bisa menemukan grup atau kelompok yang tidak dikenal dalam data. *Clustering* banyak digunakan dalam berbagai aplikasi seperti pada *business intelligence*, pengenalan pola citra, *web search*, bidang ilmu biologi dan untuk keamanan (*security*).

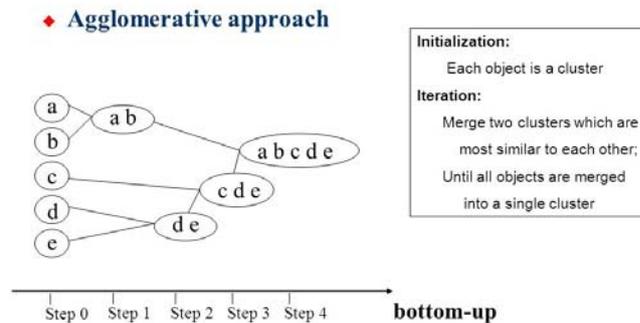
Metode *clustering* secara umum dapat dibagi menjadi dua yaitu *hierarchical clustering* dan *partitional clustering*.

### 2.2.1 Hierarchical Clustering

Pada *hierarchical clustering* data dikelompokkan melalui suatu bagan yang berupa hirarki, dimana terdapat penggabungan

dua *group* yang terdekat disetiap iterasinya atau pun pembagian dari seluruh set data kedalam *cluster*.

### Hierarchical Clustering



Gambar 2.12 *Hierarchical Clustering*

Langkah melakukan *hierarchical clustering*:

- a) Identifikasi *item* dengan jarak terdekat
- b) Gabungan *item* itu kedalam satu cluster
- c) Hitung jarak antar *cluster*
- d) Ulangi dari awal sampai semua terhubung

#### 2.2.2 *Partitional Clustering*

*Partitional clustering* yaitu data dikelompokkan ke dalam sejumlah *cluster* tanpa adanya struktur hirarki antara satu dengan yang lainnya. Pada metode *partitional clustering* setiap *cluster* memiliki titik pusat *cluster* (*centroid*) dan secara umum metode ini memiliki fungsi tujuan yaitu meminimumkan jarak (*dissimilarity*) dari seluruh data ke pusat cluster masing–masing.

Metode *K-means* merupakan metode *clustering* yang paling sederhana dan umum. Hal ini dikarenakan *K-means* mempunyai

kemampuan mengelompokkan data dalam jumlah yang cukup besar dengan waktu komputasi yang cepat dan efisien. *K-means* merupakan salah satu algoritma klustering dengan metode pastisi yang berbasis titik pusat (*centroid*) selain algoritma *k-medoids* yang berbasis objek. Algoritma *K-means* penerapannya memerlukan tiga parameter yang seluruhnya ditentukan pengguna yaitu jumlah *cluster*  $k$ , inisialisasi *cluster*, dan jarak *system*. Biasanya *K-means* dijalankan secara independen dengan inisialisasi yang berbeda menghasilkan cluster akhir yang berbeda karena algoritma ini secara prinsip hanya mengelompokkan data menuju *local minimal*. Salah satu cara untuk mengatasi *local minimal* adalah dengan mengimplementasikan algoritma *K-means*, untuk  $K$  yang diberikan, dengan beberapa nilai initial partisi yang berbeda dan selanjutnya dipilih partisi dengan kesalahan kuadrat terkecil.

*K-Means* adalah teknik yang cukup sederhana dan cepat dalam proses *clustering* obyek (*clustering*). Algoritma *K-Means* mendefinisikan *centroid* atau pusat *cluster* dari *cluster* menjadi rata-rata *point* dari *cluster* tersebut. Dalam penerapan algoritma *K-means*, jika diberikan sekumpulan data  $X = (X_1, X_2, \dots, X_n)$  dimana  $x_i = (X_{i1}, X_{i2}, \dots, X_{in})$  adalah *system* dalam ruang *real*  $R_n$ , maka algoritma *K-means* akan menyusun partisi  $X$  dalam sejumlah  $k$  *cluster* (*a priori*). Setiap *cluster* memiliki titik tengah (*centroid*) yang merupakan nilai rata-rata (*mean*) dari data-data dalam *cluster*

tersebut. Tahapan awal, algoritma *K-means* adalah memiliki secara acak  $k$  buah obyek sebagai *centroid* dalam data. Kemudian, jarak antara obyek dan *centroid* dihitung menggunakan *Euclidian distance*. Algoritma *K-means* secara *iterative* meningkatkan variasi nilai dalam tiap *cluster* dimana obyek selanjutnya ditempatkan dalam kelompok yang terdekat, dihitung dari titik tengah *cluster*. Titik tengah baru ditentukan bila semua data telah ditempatkan dalam cluster terdekat. Proses penentuan titik tengah dan penempatan data dalam *cluster* diulangi sampai nilai titik tengah dari semua *cluster* yang terbentuk tidak berubah lagi (Han dkk, 2012).

Algoritma *K-means*:

- Langkah 1 : Tentukan berapa banyak cluster  $k$  dari dataset yang akan dibagi.
- Langkah 2 : Tetapkan secara acak data  $k$  menjadi pusat awal lokasi klaster.
- Langkah 3 : Untuk masing-masing data, temukan pusat cluster terdekat. Dengan demikian berarti masing-masing pusat cluster memiliki sebuah subset dari dataset, sehingga mewakili bagian dari dataset. Oleh karena itu, telah terbentuk cluster  $k$ :  $C_1, C_2, C_3, \dots, C_k$ .
- Langkah 4 : Untuk masing-masing cluster  $k$ , temukan pusat luasan klaster, dan perbarui lokasi dari masing-masing pusat cluster ke nilai baru dari pusat luasan.

Langkah 5 : Ulangi langkah ke-3 dan ke-5 hingga data-data pada tiap cluster menjadi terpusat atau selesai.

Untuk implementasinya, kasus sederhananya seperti ini andaikan suatu jurusan di suatu universitas pada gelombang pertama memiliki lulusan sebanyak 10 mahasiswa, dan kita ingin membagi ke-10 mahasiswa tersebut menjadi empat kelompok sehingga anggota satu kelompok memiliki karakteristik yang jauh berbeda dengan kelompok yang lain (untuk kasus yang ini karakteristik yang dimaksud adalah berdasarkan IPK dan lama studi).

Berikut adalah data mahasiswa lulusan gelombang pertama:

Tabel 2.1 Data Utama

<b>Mahasiswa ke-</b>	<b>IPK</b>	<b>Lama Studi</b>
1	3,15	3,9
2	2,93	4,1
3	3,19	5,1
4	3,63	3,9
5	2,95	4,1
6	3,03	4,1
7	2,97	4,1
8	2,62	4,9
9	3,16	3,9
10	3,26	3,9

Sekarang kita akan melakukan langkah pertama di metode ini. Pada iterasi pertama kita akan menentukan berapa banyak kelompok (*cluster*) yang akan dibentuk. Dan seperti yang dijelaskan sebelumnya kita menetapkan data tersebut akan dibagi menjadi 4 kelompok (*cluster*), lalu kita akan menentukan centroid awal pada masing-masing kelompok. Sebenarnya dalam algoritma

ini centroid awal bisa ditentukan nilainya dengan acak, namun kita akan menetapkan centroid masing-masing cluster awalnya sebagai berikut:

Tabel 2.2 Centriod Awal

Centroid Cluster ke-	IPK	Lama Studi
1	3,30	3,95
2	3,30	4,05
3	3,05	3,95
4	3,05	4,05

Setelah itu jarak data setiap mahasiswa dengan centroid di setiap cluster akan kita hitung dengan rumus *Euclidian* sebagai berikut :

$$c_x(x,y) = \sum_{i=1}^m \sqrt{x_i - y_i}^2 \quad (2.2)$$

Dimana :

x = Data Mahasiswa

y = Data *Cluster*

m = Jumlah Atribut (atribut dikasus ini berjumlah dua yaitu IPK dan lama studi)

Berikut ini merupakan contoh proses perhitungan jarak centroid masing-masing *cluster* dengan data mahasiswa pertama menggunakan rumus diatas

$$c_1(x, y) = \sqrt{|3,15 - 3,3|^2 + |3,9 - 3,95|^2} = 0,158113883$$

$$c_2(x, y) = \sqrt{|3,15 - 3,3|^2 + |3,9 - 4,05|^2} = 0,21232$$

$$c_3(x, y) = \sqrt{|3,05 - 3,05|^2 + |3,9 - 4,05|^2} = 0,111803399$$

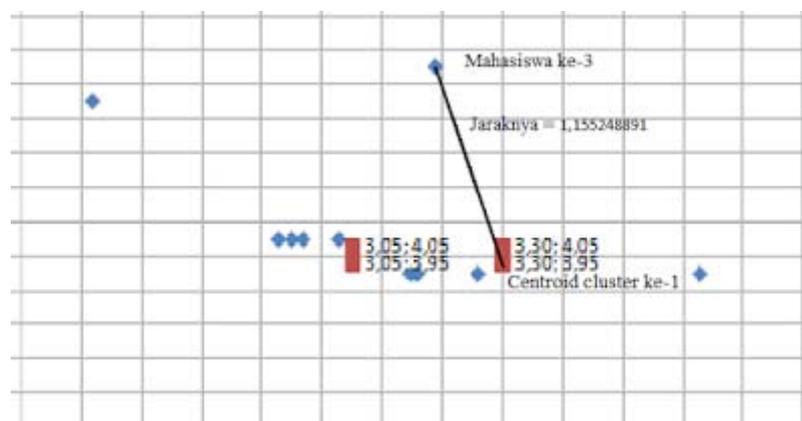
$$c_4(x, y) = \sqrt{|3,05 - 3,05|^2 + |3,9 - 3,95|^2} = 0,180278$$

Hasil dari perhitungan data tersebut diperoleh data seperti yang dapat dilihat pada Tabel dibawah :

Tabel 2.3 Perhitungan Centroid

Mahasiswa ke-	IPK	Lama Studi	Jarak dari Mahasiswa ke-n dengan centroid cluster ke-c			
			Centroid Cluster ke-1	Centroid Cluster ke-2	Centroid Cluster ke-3	Centroid Cluster ke-4
1	3,15	3,9	0,1581139	0,212132	0,1118034	0,1802776

Setelah semua jarak mahasiswa pertama dengan masing-masing cluster telah dihitung, lakukan juga terhadap mahasiswa kedua, ketiga sampai terakhir. Untuk visualisasi kita bisa melihat hasil perhitungan jarak antara centroid di cluster pertama dengan mahasiswa ketiga pada gambar dibawah



Gambar 2.13 Perhitungan Centroid

Berikut adalah hasil perhitungan jarak setiap mahasiswa dengan masing-masing cluster

Tabel 2.4 Hasil Perhitungan Centroid

Mahasiswa ke-	IPK	Lama Studi	Jarak dari Mahasiswa ke-n dengan centroid cluster ke-c			
			Centroid Cluster ke-1	Centroid Cluster ke-2	Centroid Cluster ke-3	Centroid Cluster ke-4
1	3,15	3,9	0,158113883	0,212132034	0,111803399	0,180277564
2	2,93	4,1	0,399249296	0,373363094	0,192093727	0,13
3	3,19	5,1	1,155248891	1,055746182	1,158490397	1,059292217
4	3,63	3,9	0,333766385	0,362491379	0,582151183	0,599082632
5	2,95	4,1	0,380788655	0,353553391	0,180277564	0,111803399
6	3,03	4,1	0,308868904	0,274590604	0,15132746	0,053851648
7	2,97	4,1	0,362491379	0,333766385	0,17	0,094339811
8	2,62	4,9	1,168289348	1,08853112	1,042784733	0,952575456
9	3,16	3,9	0,148660687	0,205182845	0,12083046	0,186010752
10	3,26	3,9	0,064031242	0,155241747	0,215870331	0,258069758

Mari kita lihat jarak mahasiswa pertama dengan masing-masing centroid di setiap cluster, logikanya disini adalah jika jarak antara dua titik semakin dekat, maka semakin dekatlah pula kesamaan antara kedua titik tersebut. Mahasiswa pertama mempunyai jarak paling dekat dengan centroid di kelompok 3 dibandingkan dengan centroid dari kelompok lain, maka kita bisa menyimpulkan bahwa mahasiswa pertama mempunyai karakteristik paling dekat dengan kelompok satu dibanding dengan kelompok lain sehingga mahasiswa pertama dimasukan kedalam kelompok satu. Lakukan pengelompokkan mahasiswa lainnya dengan cara yang sama juga, sehingga semua mahasiswa telah masuk ke kelompoknya masing masing

Tabel 2.5 Pengelompokan Mahasiswa

Anggota Cluster 1		
Mahasiswa ke-	IPK	Lama Studi
4	3,63	3,9
10	3,26	3,9

Anggota Cluster 2		
Mahasiswa ke-	IPK	Lama Studi
3	3,19	5,1
6	3,03	4,1

Anggota Cluster 3		
Mahasiswa ke-	IPK	Lama Studi
1	3,15	3,9
9	3,16	3,9

Anggota Cluster 4		
Mahasiswa ke-	IPK	Lama Studi
2	2,93	4,1
5	2,95	4,1
7	2,97	4,1
8	2,62	4,9

Setelah semua mahasiswa masuk dalam kelompoknya, kita akan menghitung kembali centroid baru masing-masing cluster dengan cara menjumlahkan data-data yang ada di masing-masing *cluster* dan membaginya dengan jumlah data pada *cluster* itu. Contohnya, centroid baru pada kelompok satu untuk koordinat IPK akan bernilai 3,445 yang dihitung dengan cara IPK masing-masing anggota kelompok satu ditambahkan lalu dibagi dengan jumlah

anggota kelompok satu -->  $(3,6+3,26)/2=3,445$ , lakukan hal yang sama untuk mencari koordinat lama studi.

Centroid baru yang akan digunakan untuk iterasi kedua adalah

Tabel 2.6 Centroid Baru

Centroid Baru untuk iterasi yang kedua		
Centroid Cluster ke-	IPK	Lama Studi
1	3,445	3,9
2	3,11	4,6
3	3,155	3,9
4	2,8675	4,3

Pada iterasi kedua, lakukan penghitungan jarak setiap mahasiswa dengan masing-masing cluster dengan centroid masing-masing cluster yang baru tersebut.

### 2.3 Kepramukaan

Kepramukaan sebagaimana tercantum dalam Anggaran Rumah Tangga Gerakan Pramuka (bab II Pasal 7) adalah proses pendidikan di luar lingkungan sekolah dan di luar lingkungan keluarga dalam bentuk kegiatan menarik, menyenangkan, sehat, teratur, terarah, praktis yang dilakukan di alam terbuka dengan Prinsip Dasar Kepramukaan (PDK) dan Metode Kepramukaan (MK), yang sasaran akhirnya pembentukan watak, akhlak dan budi pekerti luhur.

Kepramukaan ini dicetuskan pertama kali oleh *Robert Stephenson Smith Boden Powell dan Willian Alexander Smith* pada tahun 1907 ketika mengadakan perkemahan kepanduan pertama (dikenal sebagai *jamboree*)

di Kepulauan Brownsea, Inggris. Kepramukaan kemudian berkembang ke seluruh penjuru dunia termasuk ke Indonesia (Rahmatia, 2015).

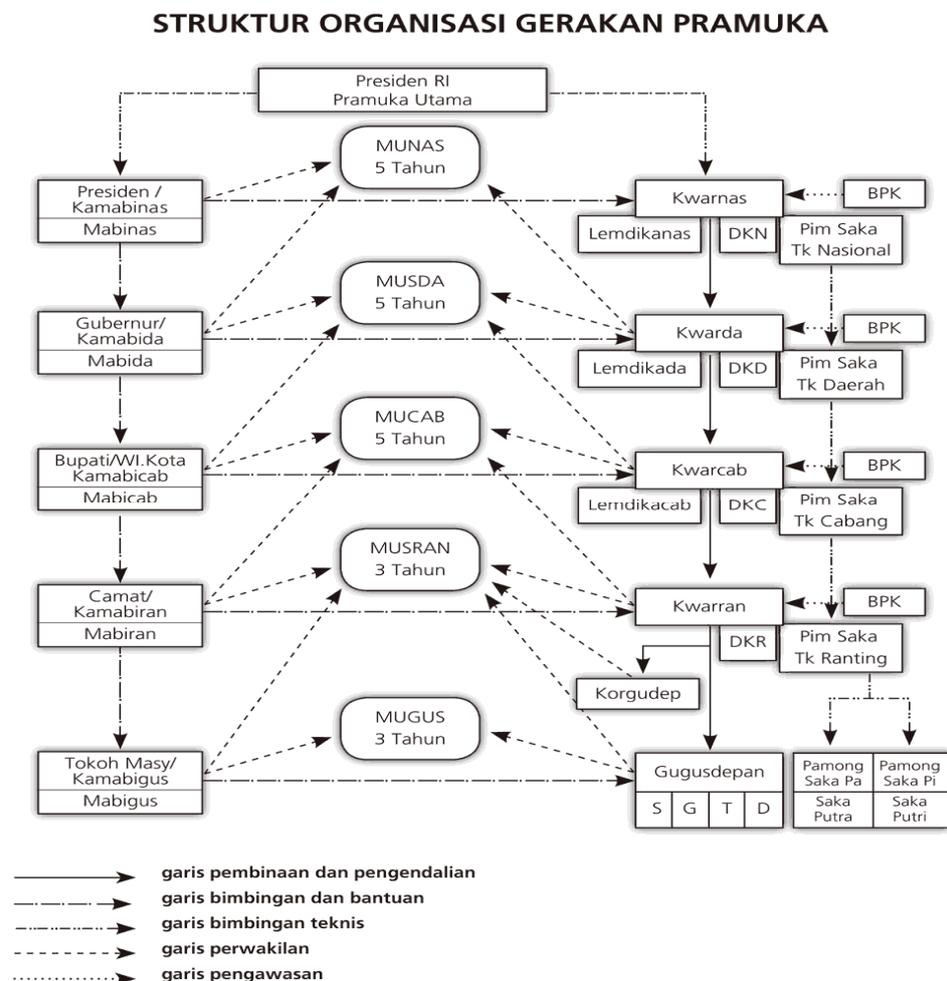
Pendidikan watak dan kepribadian diberikan kepadapeserta didik pada saat sedang asik melaksanakan kegiatan yang menarik, menyenangkan, kreatif dan menantang. Pembina Pramuka yang memiliki tugas dalam pembinaan watak atau karakter peserta didik harus dapat menciptakan kegiatan yang menarik, menyenangkan, kreatif dan menantang. Kegiatan Pramuka lebih mengutamakan kegiatan di alam terbuka sehingga setiap kegiatan kepramukaan mempunyai dua nilai yaitu nilai formal atau nilai pendidikan pembentukan watak (Character building dan nilai materiilnya yaitu kegunaan praktisnya).

Struktur Organisasi Gerakan Pramuka adalah bagan atau skema yang menggambarkan tingkatan-tingkatan organisasi Gerakan Pramuka mulai dari tingkatan yang paling bawah sampai dengan yang paling atas beserta mekanisme kerjanya. Dengan struktur organisasi tersebut, Gerakan Pramuka sebagai organisasi kependuan di Indonesia dapat menyusun dan menata organisasi gerakan pramuka dari tingkat Nasional, Daerah, Cabang, Ranting, sampai ke Gugusdepan. Sehingga organisasi dapat berjalan secara efektif.

Struktur organisasi Gerakan Pramuka diatur dalam Keputusan Kwartir Nasional Gerakan Pramuka Nomor 220 Tahun 2007 tentang Petunjuk Penyelenggaraan Pokok-Pokok Organisasi Gerakan Pramuka. Dalam keputusan ini juga diatur tentang tugas pokok dan fungsi Gerakan

Pramuka, pembagian tugas dan tanggung jawab, musyawarah, dan garis hubungan dalam organisasi Gerakan Pramuka.

Struktur Organisasi Gerakan Pramuka disusun mulai dari tingkat Nasional, Daerah, Cabang, Ranting, hingga ke Gugusdepan. Struktur organisasi tersebut terdiri atas Majelis Pembimbing (Mabi), Kwartir, Badan Pemeriksa Keuangan (BPK), Kordinator Gugusdepan (Korgudep), Gugusdepan (Gudep) dan Satuan Karya Pramuka (Saka), dan Badan Kelengkapan Kwartir.



Gambar 2.14 Struktur Organisasi Gerakan Pramuka

## 2.4 Kwartir Ranting

Kwartir Ranting Gerakan Pramuka atau biasa disingkat Kwarran adalah satuan organisasi yang mengelola Gerakan Pramuka di tingkat kecamatan. Kwarran diketuai oleh seorang Ketua Kwartir Ranting (Ka Kwarran) yang dipilih melalui Musyawarah Ranting (Musran) untuk masa bhakti selama tiga tahun. (Yusup dan Rustini, 2016).

Pengertian Kwartir Ranting secara lebih luas adalah lembaga kepemimpinan kolektif di tingkat kecamatan yang diketuai seorang ketua yang dalam menjalankan tugas dan kewajibannya bertanggungjawab kepada Musyawarah Ranting Gerakan Pramuka. Organisasi inilah yang menjadi ujung tombak Gerakan Pramuka yang berhubungan langsung dengan pembinaan gugusdepan dan satuan karya pramuka.

Terkait organisasi dan tata kerja Kwartir Ranting, Gerakan Pramuka telah mengeluarkan sebuah petunjuk penyelenggaraan Kwarran Gerakan Pramuka melalui Keputusan Kwartir Nasional Gerakan Pramuka Nomor : 224 Tahun 2007. Jukran ini membahas tentang tugas pokok, fungsi, dan organisasi; tugas dan fungsi andalan ranting; organisasi pelaksana Kwarran; Badan Pemeriksa Keuangan Ranting; tata kerja; musyawarah; hubungan kerja; dan pemekaran Kwarran.

Dewan Kerja Ranting merupakan bagian integral dari Kwartir Ranting, bertanggung jawab atas pelaksanaan tugas pokok Dewan Kerja Ranting kepada Kwartir Ranting.

## 2.5 Unified Modeling Language (UML)

Bahasa pemodelan atau UML (*Unified Modeling Language*) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML dapat digunakan untuk membuat model semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Bahasa Pemodelan yang sering digunakan :

1. *Use case diagram* yaitu menggambarkan bagaimana sistem digunakan dan titik awal untuk pemodelan UML.
2. *Use case scenario* yaitu sebuah artikulasi verbal pengecualian terhadap perilaku utama yang dijelaskan oleh kasus penggunaan utama.
3. *Activity diagram* menggambarkan aliran keseluruhan kegiatan.
4. *Sequence diagrams* yaitu menampilkan urutan kegiatan dan hubungan kelas.
5. *Class diagrams* yaitu menampilkan kelas dan hubungan.
6. *Statechart diagrams* yaitu menampilkan transisi keadaan.

### 2.6.1. Use case

Diagram ini menggambarkan fungsi dari sebuah sistem, dan interaksi yang dilakukan oleh aktor dengan sistem. Diagram ini menekankan tentang apa yang dapat diperbuat oleh sistem dan bukan bagaimana sistem itu melakukannya (Moekasan, 2016).

Tabel 2. 7 Notasi *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

No	Gambar	Nama	Keterangan
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.



Gambar 2.15 Contoh Use Case

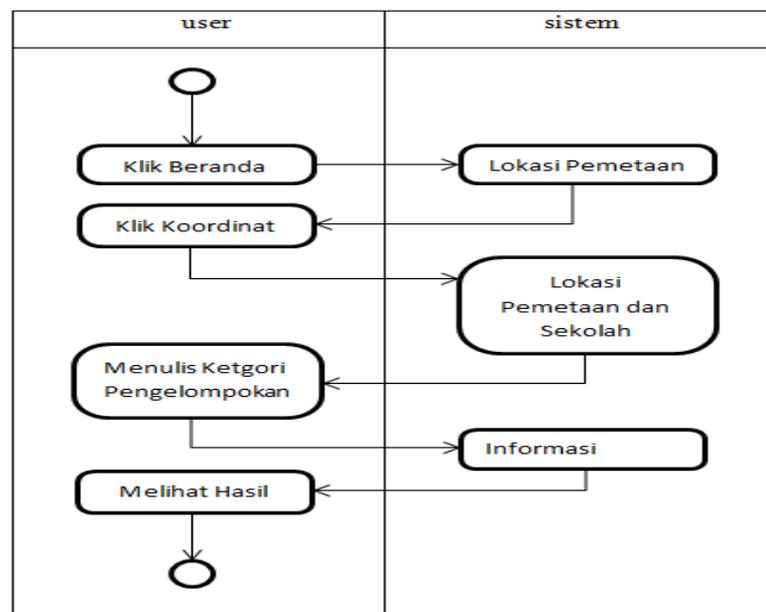
### 2.6.2. Activity diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak,

tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Tabel 2. 8 Notasi *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

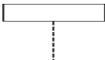


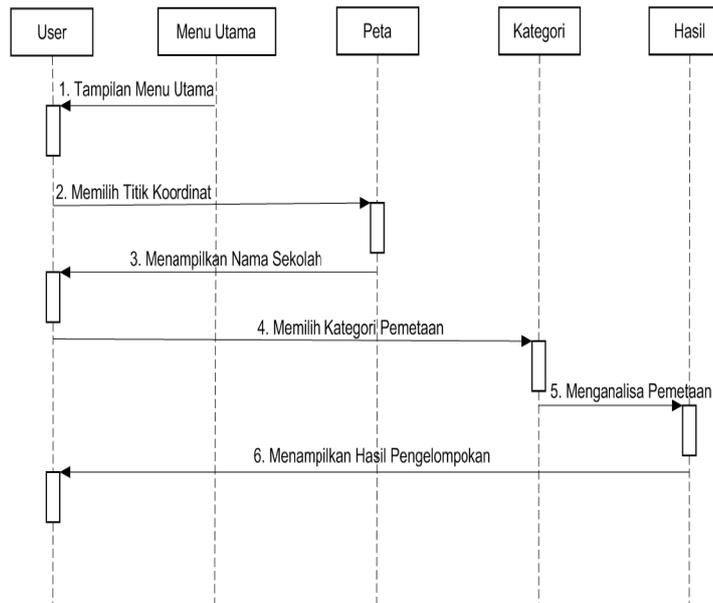
Gambar 2.16 *Activity diagram*

### 2.6.3. Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Tabel 2. 9 Notasi *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Gambar 2.17 Contoh *Sequence Diagram*

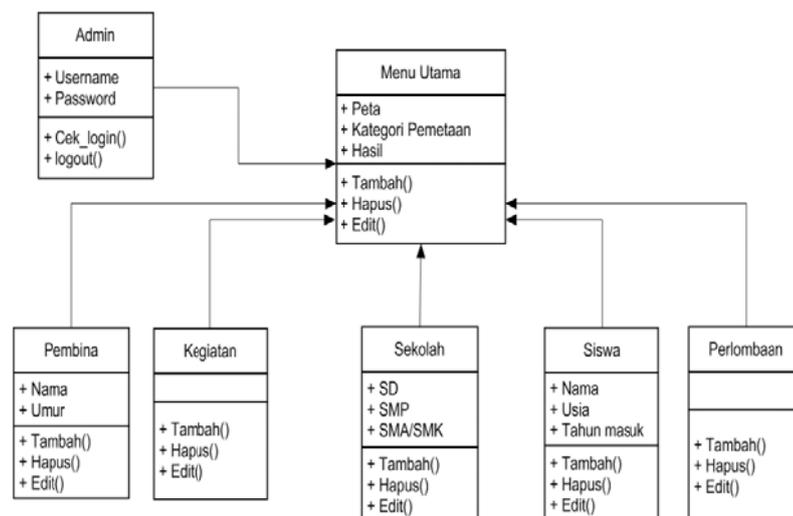
#### 2.6.4. Class Diagram

*Class Diagram* menggambarkan keadaan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem tersebut. *Class Diagram* bersifat statis di dalam *class diagram* digambarkan relasi dari masing - masing *class* tetapi tidak menggambarkan apa yang terjadi ketika *class* tersebut berrelasi.

Tabel 2.10 Notasi *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

No	Gambar	Nama	Keterangan
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2.18 Contoh *Class Diagram*

### Struktur *Class Diagram*

- *Class Main* kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- *Class Interface* kelas yang mendefinisikan pewarisan berganda.

- Kelas yang diambil dari pendefinisian *use case* kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.
- Kelas yang diambil dari pendefinisian data kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

### *Visibility* untuk Atribut dan Operasi *Class Diagram*

Dalam *class diagram* kita mengenal 3 *visibility*. Antara lain :

(+) untuk *public*, (-) untuk *private*, serta (#) untuk *protected*



Gambar 2. 19 *Visibility* untuk Atribut dan Operasi *Class Diagram*

- *Private*: tidak dapat di panggil dari luar kelas yang bersangkutan. Hanya dapat di pakai dalam kelas yang bersangkutan
- *Protected* :hanya dapat di panggil oleh kelas yang bersangkutan dan anak kelas yang diwarisinya.
- *Public* :dapat dipakai oleh kelas apa saja.

## 2.6 PHP

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan (Anhar, 2010).

PHP adalah bahasa *open source* yang dapat digunakan di berbagai sistem operasi seperti : *Linux, Unix, Macintosh, dan Windows*. PHP dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah-perintah *system*. *Open source* artinya *code-code* PHP terbuka untuk umum dan kita tidak harus membayar biaya pembelian atas keaslian *license* yang biasanya cukup mahal. Karena *source code* PHP tersedia gratis, maka hal tersebut memungkinkan komunitas dan *developer* untuk selalu melakukan perbaikan, pengembangan, dan menemukan *bug* dalam bahasa PHP.

*Web server* yang mendukung PHP dapat ditemukan dimana-mana, mulai dari *Apache, IIS, Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti *support* langsung ke berbagai macam database yang populer, misal: *Oracle, PostgreSQL*, dan lain-lain.