

BAB II

LANDASAN TEORI

2.1. Penilaian Sikap

Penilaian sikap adalah kegiatan untuk mengetahui kecenderungan perilaku spiritual dan sosial siswa dalam kehidupan sehari-hari. Sebagai hasil pendidikan baik di dalam maupun di luar kelas. Penilaian sikap memiliki karakteristik yang berbeda dengan penilaian pengetahuan dan ketrampilan, sehingga teknik penilaian yang digunakan berbeda. Dalam hal ini, penilaian sikap diujikan untuk mengetahui capaian dan membina perilaku sesuai butir-butir nilai sikap dalam Kompetensi Dasar (KD) dari Kompetensi Inti (KI). Kompetensi Inti (KI) tersebut terdiri dari kompetensi sikap spiritual (KI-1) dan kompetensi sikap sosial (KI-2) yang terintegrasi pada setiap pembelajaran Kompetensi Dasar (KD) dari kompetensi pengetahuan (KI-3) dan kompetensi ketrampilan (KI-4). (Panduan Kurikulum 2013).

Penilaian sikap yang utama dilakukan dengan menggunakan teknik observasi selama periode satu semester. Teknik observasi dilakukan oleh Guru mata pelajaran (selama proses pembelajaran pada jam pelajaran), guru bimbingan konseling (BK), dan wali kelas (selama siswa di luar jam pelajaran). Hasil observasi ditulis dalam buku jurnal (yang selanjutnya disebut jurnal), yang mencakup catatan anekdot (*anekdot record*), catatan kejadian tertentu (*incidental record*), dan informasi lain yang valid dan relevan. Jurnal tidak hanya didasarkan pada apa yang dilihat langsung penilai tetapi juga informasi lain yang relevan dan valid yang diterima dari berbagai sumber.

Dalam pelaksanaan penelitian sikap diasumsikan setiap siswa memiliki perilaku yang baik. Jika tidak dijumpai perilaku yang sangat baik atau kurang baik, maka nilai sikap siswa tersebut adalah baik dan sesuai dengan indikator yang

diharapkan. Perilaku sangat baik atau kurang baik yang dijumpai selama proses pembelajaran dicatat dan dasumsikan ke dalam jurnal guru.

Penilaian kompetensi sikap oleh guru dapat diperkuat dengan penilaian diri dan penilaian antar teman. Teknik ini dapat dilakukan dalam rangka pembinaan dan pembentukan karakter siswa, yang hasilnya dapat dijadikan sebagai salah satu data konfirmasi dari hasil penilaian sikap oleh pendidik.

Selanjutnya, wali kelas mengumpulkan data/informasi dari hasil penilaian sikap yang dilakukan oleh guru mata pelajaran, guru BK, dan/atau penilaian diri dan antar teman kemudian merangkumnya menjadi deskripsi (bukan angka atau predikat) yang menggambarkan perilaku siswa.

2.2. K-Means

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

Beberapa penelitian yang telah dilakukan mengenai masalah sensitifitas inisialisasi jumlah *cluster* (k), dan algoritma yang digunakan. Penelitian yang dilakukan oleh (Deeler & 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 intereluster sebesar mungkin dan jarak intrakluster sekecil mungkin. Sel dipartisi atau 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 menunjukkan 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 kluster yaitu algoritma awal pusat kluster 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 pusat dataset sehingga menghapus titik pertama dan tetangganya. Proses ini berlanjut sampai set M awal berisi k poin. Hasil percobaan menunjukkan bahwa algoritma yang diusulkan sengaja meningkatkan kualitas dan stabilitas algoritma K-means. (Zhang & Fang, 2013) melakukan penelitian dalam perbaikan algoritma K-means untuk mengoptimalkan inisialisasi pusat kluster. Dengan menemukan satu set data yang mencerminkan karakteristik distribusi data sebagai pusat awal kluster 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 menunjukkan bahwa hasil setiap kluster lebih kompak.

2.3. Algoritma Cluster

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

Dalam penelitian penentuan sampel merupakan kebutuhan utama untuk memperoleh data yang menggambarkan realitas. Data tersebut akan dijadikan fokus dalam penelitian. Dalam kasus ini data menggunakan siswa 2 kelas sejumlah kurang lebih 45 siswa

Beberapa metode yang digunakan untuk menghitung selisih jarak adalah *Manhattan/City Block distance space*. Adapun persamaanya adalah sebagai berikut:

$$D_{L_1}(x_2, x_1) = \|x_2 - x_1\|_1 = \sum_{j=1}^p |x_{2j} - x_{1j}|$$

Ketrangan :

$D(x_2, x_1)$ = Jarak objek antara x_2 dan x_1

$| \ |$ = Nilai absolute

p = Dimensi data

x_1 = Koordinat objek x_1

x_2 = Koordinat objek x_2

2.4. MySQL

MySQL diciptakan di negara Swedia oleh perusahaan MySQL AB. MySQL adalah suatu jenis *database server* yang sangat terkenal. Sangat populer karena bersifat *free* (tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali pada windows, yang bersifat software atau anda perlu membayar setelah melakukan evaluasi dan memutuskan digunakan untuk keperluan produksi). Perangkat lunak MySQL sendiri bisa di *download* dari <http://www.mysql.org> atau <http://www.mysql.com> (Andre Adelheid dan Khairil, 2012).

MySQL termasuk jenis RDMS (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel , baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : David Axmark, Allan Larsson, dan Michael “Monty” Widenius. Tersedia sebagai perangkat lunak gratis dibawah lisensi GNU, GPL (*General Public Licence*), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaanya tidak cocok menggunakan aturan GPL

Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu *command-line*. Juga dapat di download dari situs MySQL yaitu sebuah modul berbasis grafis (GUI): *MySQL Administrator* dan *MySQL Query Browser*. Selain itu terdapat juga sebuah perangkat lunak gratis untuk

administrasi basis data MySQL berbasis web yang sangat populer yaitu phpMyAdmin.

2.5. PHP

PHP atau *Hypertext Preprocessor* adalah bahasa yang berbentuk *script* yang ditempatkan dalam *server* dan dieksekusi didalam *server* untuk selanjutnya *ditransfer* dan dibaca oleh *client* (Andre Adelheid dan Khairil, 2012). Ada pun keuntungan dari bahasa pemrograman PHP adalah sebagai berikut :

- Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan *apache*, dengan konfigurasi yang relative mudah.
- Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
- Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak. PHP adalah bahasa *open source* yang dapat digunakan diberbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah system





2.6. UML

UML atau *Unified Modeling Language* adalah sebuah bahasa pemodelan untuk sistem atau perangkat lunak yang *berpradigma* “berorientasi objek”. Pemodelan sesungguhnya digunakan untuk penyederhanan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Adi Nugroho, 2010).

2.6.1. Use Case Diagram

Use case diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case diagram* dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu system (Indrajani, 2010).

.Tabel 2.1. Simbol *Use Case Diagram*.

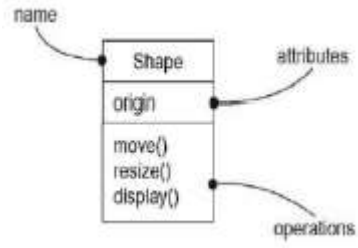


Simbol	Keterangan
	<i>System Bondary</i> menggambarkan batasan antara sistem dengan aktor
	Simbol ini menggambarkan interaksi antara aktor dengan sistem.
	Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem <i>terotomatisasi</i> lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
	Menggambarkan hubungan antar aktor dan <i>use case</i> .

(Indrajani, 2010)

2.6.2. Class Diagram



Class diagram biasanya digunakan untuk menggambarkan perbedaan yang mendasar antara *class-class*, hubungan antara *class*, dan di mana *sub-sistem class* tersebut (Indrajani, 2010).

Tabel 2.2. Simbol *Class Diagram*.

Simbol	Keterangan
 <p>The diagram shows a class box for 'Shape'. The top section contains the name 'Shape'. The middle section contains the attribute 'origin'. The bottom section contains the methods 'move()', 'resize()', and 'display()'. Labels with arrows point to these sections: 'name' points to the top section, 'atributes' points to the middle section, and 'operations' points to the bottom section.</p>	<p><i>Class</i> adalah blok – blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i>. Bagian tengah mendefinisikan <i>property /atribut class</i>. Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i>.</p>
 <p>The diagram shows a horizontal line representing an association between two classes. The left end is labeled '1..n' and the right end is labeled '1'. The text 'Owned by' is written above the line.</p>	<p>Sebuah <i>asosiasi</i> merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i>. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum <i>multiplisitas</i> pada sebuah <i>relationship</i>. (Contoh: <i>One-to-one</i>, <i>one-to-many</i>, <i>many-to-many</i>).</p>
 <p>The diagram shows a solid line representing composition. It starts with a filled diamond symbol at the left end and extends to the right.</p>	<p>Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang <i>berisi/solid</i>.</p>

(Indrajani, 2010)

Tabel 2.2. Simbol *Class Diagram*.

	<p>Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i>. Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik</p>
	<p><i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi</p>

(Indrajani, 2010)


2.6.3. *Sequence Diagram*

Sequence diagram merupakan suatu diagram interaksi yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi dan pesan yang ditukar dalam urutan waktu (Indrajani, 2010).

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 respon 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 2.3. Simbol *Sequence Diagram*.

Simbol	Keterangan
	<p><i>Object Lifeline</i>: menggambarkan <i>object</i> apa saja yang terlibat.</p>

(Indrajani, 2010)

Tabel 2.3. Simbol *Sequence Diagram*.

	<p><i>Actor</i>: menggambarkan hubungan <i>actor</i> yang terlibat.</p>
	<p><i>Activation</i>: menggambarkan hubungan antara <i>object</i> dengan <i>message</i>.</p>
	<p><i>Message(call)</i>: menggambarkan alur <i>message</i> yang merupakan kejadian objek pengirim <i>lifeline</i> ke objek penerima <i>lifeline</i>.</p>
	<p><i>Message(return)</i>: menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.</p>

(Indrajani, 2010)