BAB II

LANDASAN TEORI

2.1 SISTEM PENDUKUNG KEPUTUSAN (SPK)

Menurut Raul Verde (2011), Sebuah sistem pendukung/penunjang keputusan dapat digambarkan sebagai sistem, interaktif berbasis komputer yang dirancang untuk membantu para pengambil keputusan untuk memecahkan masalah tak terstruktur. Menggunakan kombinasi dari model, teknik analisis, dan pengambilan informasi, sistem seperti membantu mengembangkan dan mengevaluasi atau menjadi sistem alternatif yang benar-benar sesuai.

Pada dasarnya sistem penunjang keputusan merupakan pengembangan lebih lanjut dari Sistem Informasi Manajemen Terkomputerisasi (*Computerized Management Information System),* yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya.

Tujuan dari dibuatnya sistem penunjang keputusan adalah membantu menyelesaikan masalah semi-terstruktur dan mendukung dalam mengambil keputusan. Model perhitungan yang akan dijadikan dasar Analisa Aplikasi Pendukung Keputusan Penentu Rangking adalah Algoritma C.4.5.

2.2 *DATA MINING*

Data Mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Kata *mining* sendiri berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar. Karena itu Data Mining

sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik dan database. Data mining adalah proses menerapkan metode ini untuk data dengan maksud untuk mengungkap pola-pola tersembunyi.

Alasan utama untuk menggunakan *data mining* adalah untuk membantu dalam analisis koleksi pengamatan perilaku. Data tersebut rentan terhadap *collinearity* karena diketahui keterkaitan. Fakta yang tak terelakkan data mining adalah bahwa subset/set data yang dianalisis mungkin tidak mewakili seluruh domain, dan karenanya tidak boleh berisi contoh-contoh hubungan kritis tertentu dan perilaku yang ada di bagian lain dari domain (Susanto Sani & Suryadi Dedi, 2010).

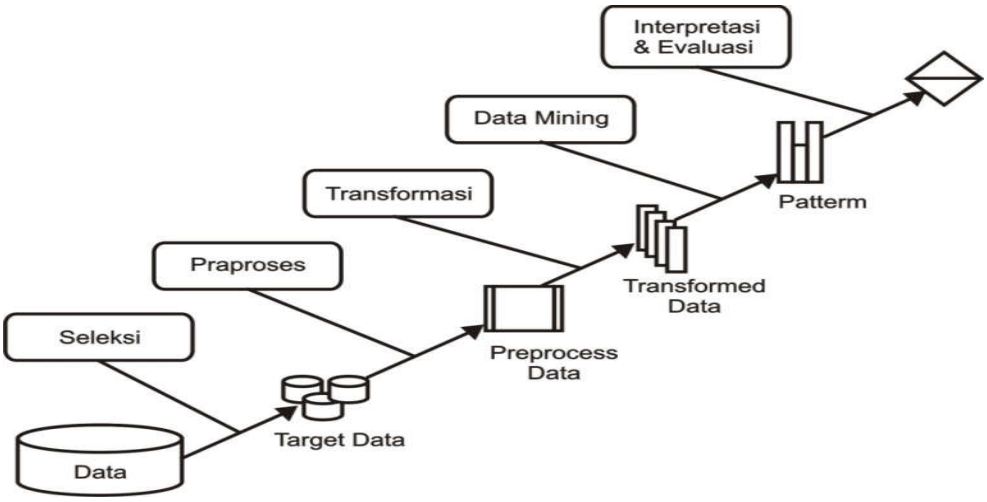
Dari beberapa definisi yang tersebut diatas ada beberapa hal penting yang terkait dengan *data mining* :

a. *Data mining* merupakan proses otomatis dari data yang sudah ada
b. Data yang diproses merupakan data yang besar
c. Tujuan dari *data mining* adalah menemukan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Dalam otomatisasi data mining ini bukan berarti tidak menggunakan campur tangan manusia. Manusia harus ikut aktif dalam setiap fase dalam proses *data mining.* Kemampuan algoritma *data mining* yang ada di dalam perangkat lunak analisis yang terdapat saat ini memungkinkan terjadinya kesalahan penggunaan yang berakibat fatal.

2.2.1 Tahapan Proses *Data Mining*

Ada beberapa tahapan proses dalam *data mining*. Diagram di bawah ini akan menunjukan beberapa tahapan proses dalam data mining. Fase awal dimulai dari data sumber dan berakhir dengan adanya informasi yang dihasilkan dari beberapa tahapan, yaitu :



Gambar 2.1. *Data Mining* sebagai Tahapan dalam Penemuan Pengetahuan

1. *Data cleaning*, untuk membersihkan data dari *noise* data dan data yang tidak konsisten.
2. Data integrasi, mengkombinasikan/mengintegrasikan beberapa sumber data.
3. Seleksi data, mengambil data-data yang relevan dari database untuk dianalisis.
4. Transformasi data, pemetaan data dari semua elemen dengan mengunakan aturan transformasi satu kebanyak dan banyak ke satu. Pembangkitan kode yang menciptakan program transformasi yang sesungguhnya agar dapat menghasilkan progam yang dapat dieksekusi pada komputer.

5. *Data mining*, merupakan proses yang esensial dimana metode digunakan untuk mengekstrak pola data yang tersembunyi.
6. Evaluasi pola, untuk mengidentifikasi pola sehingga merepresentasikan pengetahuan berdasarkan nilai-nilai yang menarik.
7. Pengetahuan presentasi, dimana teknik representasi dan evaluasi data digunakan untuk mempresentasikan pengetahuan yang didapat kepada user.

2.2.2 Teknik *Data Mining*

Terdapat beberapa teknik *data mining* yang sering disebut-sebut dalam literatur. Namun ada 3 teknik data mining yang popular, yaitu :

a. Association Rule Mining

Association Rule Mining adalah teknik mining untuk menemukan asosiatif antara kombinasi atribut. Contoh dari aturan asosiatif dari analisa pembelian di suatu pasar swalayan dapat mengatur penempatan barangnya atau merancang strategi pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu.

b. Clustering

Berbeda dengan *Association rule mining* dan klasifikasi dimana kelas *data* telah ditentukan sebelumnya, *clutering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui. Karena itu *clustering* sering digolongkan sebagai metode unsupervised learning. Prinsip *clustering* adalah memaksimalkan kesamaan antar cluter. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai multidemensi.

c. Klasifikasi

Dalam klasifikasi, terdapat target variable kategori, Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, pendapatan rendah.

2.3 Pohon Keputusan

Pohon keputusan adalah pemetaan mengenai alternatif-alternatif pemecahan masalah yang dapat diambil dari masalah tersebut. Pohon tersebut juga memperlihatkan faktor-faktor kemungkinan/probablitas yang akan mempengaruhi alternatif-alternatif keputusan tersebut, disertai dengan estimasi hasil akhir yang akan didapat bila kita mengambil alternatif keputusan tersebut.

Pohon keputusan adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari pohon keputusan adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan. Manfaat utama dari penggunaan pohon keputusan adalah kemampuannya untuk memangkas proses pengambilan keputusan yang kompleks menjadi lebih simpel sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan. Pohon Keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi karena pohon keputusan memadukan antara eksplorasi data dan pemodelan, dia sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain (Adiana, Julie, 2012).

Sebuah pohon keputusan mungkin dibangun dengan seksama secara manual atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algoritma pohon keputusan untuk memodelkan himpunan data yang belum terklarifikasi.

Variabel tujuan biasanya dikelompokkan dengan pasti dan model pohon keputusan lebih mengarah pada perhitungan probabilitas dari tiap-tiap *record* terhadap kategori-kategori tersebut atau untuk mengklarifikasi *record* dengan mengelompokkannya dalam satu kelas.

Pohon keputusan juga dapat digunakan untuk mengestimasi nilai dari variabel *continue* meskipun ada beberapa teknik yang lebih sesuai untuk kasus ini.

Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan, antara lain ID3, CART, dan C 4.5. Algoritma C 4.5 merupakan pengembangan dari algoritma ID3.

Data dalam pohon keputusan biasanya dinyatakan dalam bentuk tabel dengan atribut dan *record*. Atribut menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan *tree*. Misalkan untuk menentukan rangking, kriteria yang digunakan adalah data absensi, tingkah laku dan nilai raport. Salah satu atribut merupakan atribut yang menyatakan solusi per-item data disebut dengan target atribut. Atribut memiliki nilai-nilai yang dinamakan dengan *instance* (nilai atribut). Misalkan atribut berupa data absensi, tingkah laku dan nilai raport.

No	Nama	Data Absensi	Tingkah Laku	Nilai Raport	Keputusan
1	Ardiyanto	>75% (Baik)	Baik (Baik)	80 (Baik)	YA
2	Risky	<75% (Kurang Baik)	Kurang Baik (Kurang Baik)	75 (Baik)	TIDAK
3	Cahaya	>75% (Baik)	Kurang Baik (Kurang Baik)	70 (Baik)	YA
4	Anang	<75% (Kurang Baik)	Baik (Baik)	65 (Kurang Baik)	TIDAK

Sample Atribut Target/ Keputusan

Gambar 2.2 Keputusan Perolehan Rangking

2.4 Algoritma C4.5

Algoritma c4.5 dikenal dengan algoritma yang membentuk keputusan berdasarkan pohon keputusan (*Decision tree*) pada proses ini mampu menghasilkan keputusan yang kompleks menjadi lebih sederhana.

Dalam algoritma C4.5 untuk membangun pohon keputusan hal pertama yang dilakukan yaitu memilih atribut sebagai akar. Kemudian dibuat cabang untuk tiap-tiap nilai didalam akar tersebut. Langkah berikutnya yaitu membagi kasus dalam cabang. Kemudian ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama (Drs. Suarga, 2010).

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut.

a. Pilih atribut sebagai akar.
b. Buat cabang untuk tiap-tiap nilai.
c. Bagi kasus dalam cabang.
d. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain*.

rumus (2.1) seperti tertera sebagai berikut :

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} * Entropy(Si)$$

Keterangan :

S : himpunan kasus

A : Atribut

n : jumlah partisi atribut

|Si| : jumlah kasus pada partisi ke –i

|S| : jumlah kasus dalam S

Setelah mendapatkan nilai *gain*, ada satu hal lagi yang perlu dilakukan perhitungan yaitu mencari nilai *Entropy. Entropy* digunakan untuk menentukan seberapa *informative* sebuah *input atribut* untuk menghasilkan *output atribut*.

Rumus (2.2) dasar dari *entropy* tersebut adalah sebagai berikut :

$$Entropy(S) = \sum_{i-1}^{n} - pi * \log 2\, pi$$

Keterangan :

S : Himpunan kasus

n : Jumlah partisi S

pi : Proporsi dari Si terhadap S

2.5 CODEGEAR RAD STUDIO DELPHI 2009

2.5.1 RAD Studio Delphi 2009

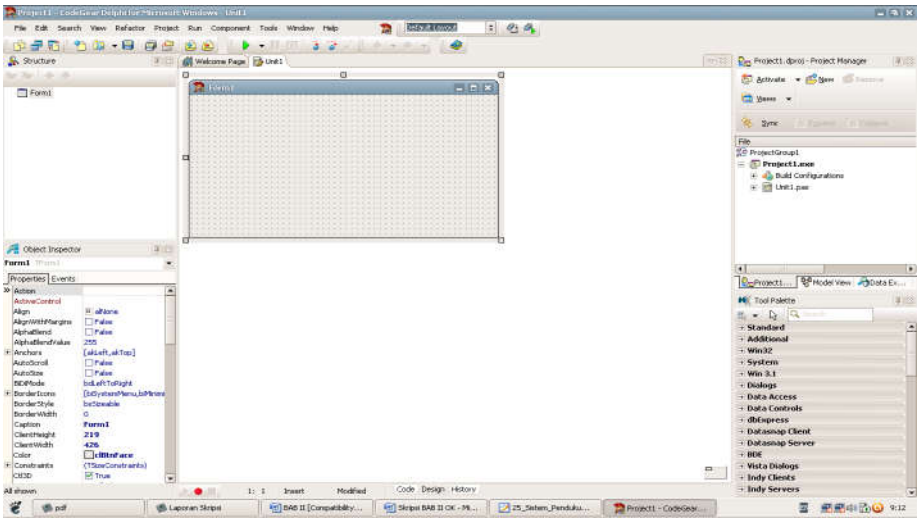
Codegear RAD Studio Delphi 2009 adalah Suatu bahasa pemrograman yang menggunakan visualisasi sama seperti bahasa pemrograman Visual Basic (VB) . Namun Delphi menggunakan bahasa yang hampir sama dengan pascal (sering disebut objeck pascal) sehingga lebih mudah untuk digunakan . Bahasa pemrograman Delphi dikembangkan oleh Code Gear sebagai divisi pengembangan perangkat lunak milik embarcadero. Divisi tersebut awalnya milik borland , sehingga bahasa ini memiliki versi Borland Delphi.

Codegear RAD Studio Delphi 2009 merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual yang digunakan untuk merancang suatu aplikasi program. Untuk memulai pemrograman Delphi, yang pertama kali adalah membuka program Delphi bila belum dijalankan. Umumnya cara untuk menjalankan program Delphi adalah dengan melalui menu Start > Programs > Embarcadero RAD Studio 2009 > Delphi 2009 (Kusnassriyanto, 2012).

2.5.2 Integrated Development Environment (IDE)

Langkah awal dalam belajar Delphi adalah mengenal *Integrated Development Environment (IDE)* Delphi yang merupakan lingkungan pengembangan terpadu bagi programmer dalam mengembangkan aplikasinya.

IDE delphi terdiri dari beberapa bagian utama yaitu menu, *component pallet, toolbar,* object *tree view, code explorer, object inspector, from designer* dan *code editor* seperti ditunjukkan pada Gambar 2.6, dalam aplikasi program ini komponen-komponen yang penting diketahui, yakni :



Gambar 2.3 IDE Delphi

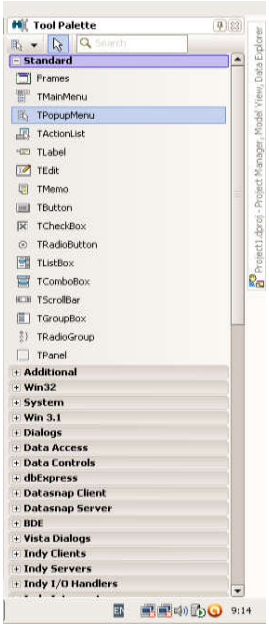
1) Object Treeview

Adalah merupakan sebuah diagram pohon yang menggambarkan hubungan logis antara komponen *visual* dan *nonvisual* yang terletak pada form, data modul ataupun frame. Semua objek yang diperlukan pada form, data ataupun frame akan muncul pada *objek tree view*, secara otomatis fokus pada akan berpindahan.

2) Component Pallete

Adalah merupakan bagian yang digunakan untuk meletakkan bebagai komponen yang sesuai dengan kategori. Misalnya komponen yang digunakan untuk aplikasi yang menggunakan

Borland Database Engine (BDE) akan diletakkan pada page BDE atau komponen yang digunakan untuk aplikasi yang membutuhkan *Interbase* akan diletakkan pada *Interbase* dan *Interbase Admin*.



Gambar 2.4 Component Pallete

Adapun secara garis besar fungsi dari masing-masing kontrol tersebut adalah :

a) *Button / Bitbtn* biasa digunakan sebagai tombol kendali. Perbedaan antara *Bitbtn* dengan *button* : pada *bitbtn* kita dapat menyisipkan warna pada tombol dan icon tertentu, lain halnya bila kita menggunakan *button*.

b) *Panel* berfungsi untuk mengelompokkan komponen-komponen di dalamnya.

c) *Label* berfungsi memberi keterangan pada program.

d) *Edit* berfungsi sebagai masukkan data dalam bentuk string, dari bentuk string ini kita dapat mengolahnya menjadi bentuk integer

atau bentuk lainnya, yang kemudian dapat digunakan untuk operasi selanjutnya.

e) *Chart*, data-data yang kita analisa dapat kita tampilkan kedalam grafik, sehingga memudahkan kita untuk menganalisanya.

f) *Stringgrit* berguna untuk menaruh data kedalam bentuk kolom tabel seperti pada excel.

g) *Popup* Menu berfungsi sebagai perintah yang aktif bila kita mengklik kanan mouse, untuk mengaktifkannya kita harus mengaktifkan popup menu pada komponen yang diinginkan, caranya dengan mengubah pada *object inspector*.

h) *Main Menu* adalah option pada tiap aplikasi program, dengan komponen ini, kita bisa menaruh fungsi-fungsi program seperti pada aplikasi umumnya.

i) *ComboBox* berfungsi sebagai petunjuk berbagai masukkan.

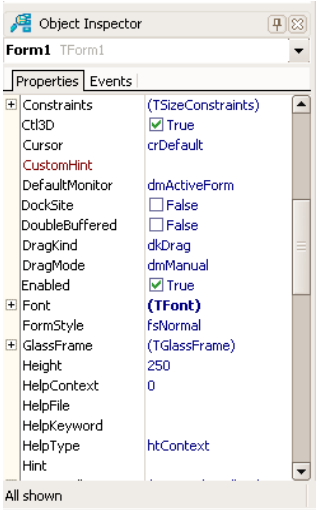
j) *CheckBox*, bila komponen ini di check maka ada aplikasi yang bisa disetting untuk bekerja di bawahnya dan memiliki lebih dari satu pilihan.

k) *RadioButton* dengan prinsip kerjanya hampir sama dengan *CheckBox*, cuma tampilannya saja yang berbeda dan hanya memiliki satu pilihan.

Masih banyak komponen / perintah lain yang terdapat pada *Component* Pallete. Untuk mempelajarinya lebih lanjut dapat dibaca pada *Help Menu*.

3) Object Inspector

Object *inspector* terdiri dari 2 bagian yaitu : *properties* dan *events*. Pada bagian *properties* dapat diatur berbagai *properti* dari *object* atau *komponen* yang tentunya mempunyai *properti* sendiri-sendiri. Misalnya jika ditempatkan komponen *button* pada *from* tentu dapat diganti propertinya. Saat membuka Delphi pertama kali, nampak sebuah jendela *Object Inspector*. Jika tidak muncul pilih menu *View* | *ObjectInspector* atau tekan tombol F11.



Gambar 2.5 Jendela Object Inspector untuk Mengganti Properti Caption

Semua properti diurutkan berdasarkan *alpabetik*, dan dapat juga diurutkan berdasarkan kategori. *Caption* digunakan untuk menyimpan keterangan yang dimunculkan pada *form*, sedangkan *Name* digunakan sebagai Nama dari objek tersebut.

4) Code Explorer

Pada jendela *code explorer* ini akan ditampilkan semua tipe *variabel* dan routine yang didefinisikan pada unit. Selain itu juga ditampilkan semua unit yang digunakan dan terletak pada *clausa uses*. Untuk tipe yang kompleks seperti kelas pada explorer akan menampilkan semua informasi termasuk daftar *field*, *properties* dari *method*. Jika dilakukan klik dua kali pada salah satu item *code explorer*, secara otomatis *code editor* akan menampilkan deklarasi dari item yang di klik.

5) Code Editor

Dalam bagian ini dapat dituliskan semua *code* dan memasukkan semua unit yang dipakai untuk program yang dibuat. Jika dibuka lebih dari satu file unit *code editor* ini, pada bagian atas akan muncul tab atau page yang berisi nama file yang sedang dibuka. Susunan atau urutan page dapat dibuat judul yang lebih menarik pada bagian posisi yang diinginkan.

2.5.3 Menu dan Perintah pada Codegear RAD Studio Delphi 2009

Ada empat cara untuk memberi perintah pada lingkungan Codegear RAD Studio Delphi 2009 (Delphi environment) :

a) Gunakan *menu*.

b) Gunakan *Short Cut* (misal F9, F12, dll).

c) Gunakan SpeedBar (atau toolbar).

d) Gunakan*SpeedMenu* (lokal menu yang diaktifkan dengan tombol mouse kanan).

2.6 DATABASE

Sebuah database adalah sebuah struktur yang umumnya dikategorikan dalam 2 hal: Sebuah database flat dan sebuah database relasional. Database relasional lebih disukai karena lebih masuk akal dibandingkan database flat, MySQL adalah sebuah database relasional (Aswan,M.Si., 2012).

Pada database yang memiliki struktur relasional.Ada tabel-tabel yang menyimpan data.Setiap tabel terdiri dari kolom dan baris. Sebuah kolom mendefinisikan jenis informasi apa yang akan disimpan. Diperlukan kolom khusus untuk setiap jenis informasi yang ingin di simpan (misalnya umur, tinggi, berat, alamat).

Perancangan database merupakan hal yang sangat penting, karena dapat memberikan informasi kepada pemakai tentang sistem yang dibuat dan dapat membantu memudahkan dalam proses pengolahan database itu sendiri. Tidaklah mengherankan bila perancangan database haruslah informatif dan memuaskan, sesuai kebutuhan yang ada, baik saat ini ataupun disaat yang akan datang.

Sifat-sifat database :

a. Internal : Kesatuan (integritas) dari file-file yang terlibat.
b. Terbagi/share : Elemen-elemen database dapat dibagikan pada para user baik secara sendiri-sendiri maupun secara serentak dan pada waktu yang sama (*Concurrent sharing*).

2.7 MYSQL

Dalam manajamen sistem *database* sebenarnya banyak software pilihanya seprti Oracle, PostgreSQL, MySQL, dan Microsoft Sql *server*. Namun perlu di perhatikan yang paling penting apakah performa, *support*, fitur-fitur SQL, keamanan dan harga. Dengan pertimbangan tersebut, *database* MySQL memiliki beberapa kelebihan dan keuntungan dibanding *database* lain, diantaranya adalah :

a. MySQL merupakan *server* yang cepat.
b. MySQL merupakan *system* manajemen *database* yang *Open Source*, yaitu *software* ini bersifat bebas digunakan oleh perorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
c. MySQL punya performa tinggi tapi simpel.
d. *Database* MySQL mengerti bahasa SQL (Structured Query Language).
e. MySQL dapat diakses melalui *protocol* ODBC (*Open Database Connectivity*) buatan *Microsoft*. Ini menyebabkan MySQL dapat diakses oleh banyak *software.*

Database MYSQL bersifat open source dan mampu menangani data yang sangat besar hingga ukuran Giga Byte . MySQL merupakan program *database server* dengan SQL sebagai bahasa yang digunakan di dalamnya. SQL dibagi menjadi dua bentuk *query*, yaitu:

1. DDL (*Data Definition Language*)

   DDL adalah sebuah Metode *Query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*.

2. DML (*Data Manipulation Language*)

DML adalah sebuah metode *Query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *Query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau yang telah dibuat sebelumnya. (Agus, 2011)

MySQL adalah sebuah *server database open source* yang termasuk populer keberadaannya. MySQL umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi *server* yang dinamis dan *powerful*.

2.8 UML (*Unified Model Language*)

UML adalah UML merupakan singkatan dari *"Unified Modelling Language"* yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software.

UML adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun (Widodo, Pujo Prabowo 2011).

Dalam sistem pendukung keputusan ini UML yang dibuat terdiri dari *Use case* diagram, *Activity* diagram, *Class* diagram dan *Sequence* diagram.

a. *Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

b. *Activity* Diagram

Activity diagrams 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.

c. *Class* Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment* , pewarisan, dan asoisasi.

d. *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 atar

dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

2.9 Sistem Penyusunan Rangking dan Raport di SMA N Kerjo

2.9.1 Prestasi Belajar

Prestasi belajar adalah penguasaan pengetahuan atau keterampilan yang dikembangkan oleh mata peljaran lazimnya dengan nilai atau angka yang diberikan oleh guru setelah melaksanakan tugas atau mengerjakan soal-soal tes ujian yang diberikan padanya. Prestasi belajar berkaitan erat dengan peringkat atau rangking yang di raih siswa, semakin tinggi jumlah nilai mata pelajaran yang ia dapat, semakin tinggi pula peringkat yang akan ia raih.

Dalam penentuan prestasi belajar meliputi beberapa nilai, diantaranya :

1. Nilai hasil belajar tiap-tiap mata pelajaran
2. Jumlah Nilai
3. Nilai Rata-rata
4. Rangking / Peringkat
5. Catatan Absensi siswa
6. Catatan kepribadian siswa meliputi : Kelakuan, Kerajinan, Kebersihan dan Kerapihan.

2.9.2 Buku Raport

Buku rapor adalah suatu cara pengukuran kinerja siswa dan sebagai sarana komunikasi antara pihak sekolah dengan orang tua atau

wali murid yang dibagikan setiap akhir semester. Dalam buku rapot tersebut tertulis catatan-catatan hasil belajar siswa yang meliputi : Catatan Nilai Mata Pelajaran, Catatan Absensi dan Catatan Kepribadian siswa. Ada siswa yang mendapat nilai bagus, sedang dan rendah, adapula siswa yang menduduki peringkat satu, dua dan tiga, bahkan ada siswa yang tidak mendapatkan peringkat sama sekali. Masing-masing siswa memiliki catatan-catatan yang berbeda dalam buku rapotnya. Catatan-catatan tersebut adalah prestasi hasil belajar siswa dalam tiap-tiap semester. Nilai rapot didapat dari Nilai Ulangan Harian, Nilai Pekerjaan Rumah (PR), Nilai Ulangan Tengah Semester (UTS) dan Nilai Ulangan Semester, juga Nilai Kepribadian siswa yang meliputi 3 aspek yaitu : Kelakuan, Kerajinan, Kebersihan dan Kerapihan.

2.9.3 Pegertian Peringkat atau Rangking

Peringkat dalam kamus Bahasa Indonesia berarti : Tingkat atau Kedudukan. Peringkat atau Rangkng Kelas adalah : suatu tingkat atau kedudukan yang diraih oleh siswa dalam suatu pencapaian hasil belajar dikelasnya. Untuk mengetahui peringkat atau rangking kelas antara siswa yang satu dengan yang lain dapat dilihat dari jumlah nilai rapot dan nilai rata-rata rapot siswa tersebut. Lazimnya penulisan peringkat hanya ditulis dari peringkat 1 (Satu) sampai peringkat 3 (Tiga). Namun ada pula penulisan peringkat dari peringkat 1 (Satu) sampai 5 (Lima) disebut *THE BEST FIVE* (Lima Besar). Bahkan ada yang

menuliskan peringkat peringkat 1 (Satu) sampai 10 (Sepuluh) disebut *THE BEST TEN* (Sepuluh Besar).

2.9.4 Penentuan Peringkat atau Rangking

Untuk menentukan peringkat atau rangking guru dapat membandingkan Nilai Hasil Belajar siswa, tentunya nilai hasil belajar siswa tidak sama. Ada siswa yang mendapat nilai tinggi, sedang dan rendah, dengan membandingkan nilai hasil belajar siswa guru dapat mengetahui siswa siswinya yang layak mendapat peringkat tinggi, sedang dan rendah. Setelah guru membandingkan nilai tertinggi dan terendah dari tiap-tiap mata pelajaran kemudian guru menghitung rata-rata nilai masing-masing siswa dan selanjutnya menentukan peringkat atau rangking 1, 2, 3 dan seterusnya. Untuk mengetahui rata-rata nilai rapot dapat ditentukan dengan cara sebagai berikut :

$$\text{Nilai Rata-rata} = \frac{\text{Jumlah Seluruh Nilai Mata Pelajaran}}{\text{Jumlah Seluruh Mata Pelajaran}}$$

Dari contoh Format nilai rekap rapot diatas, dapat kita ketahui dengan jelas siapa yang mendapat peringkat Satu, Dua, Tiga dan seterusnya.

Apabila terdapat jumah nilai yang sama dalam nilai rata-rata antara siswa A dan siswa B, untuk menentukan siapa yang mendapat ranking / peringkat tertinggi apakah siswa A atau siswa B persamaan maka dilakukan dengan cara membandingkan nilai tertinggi pada mata pelajaran Eksak atau pelajaran utama seperti : Matematika, IPS dan B. Indonesia.

Antara siswa A dan B jumlah nilai rata-ratanya sama, setelah diperiksa dan diteliti siswa A lebih unggul nilai mata pelajaran Matematikanya disbanding siswa B, berarti siswa A menduduki peringkat 1 (Satu) dan siswa B menduduki peringkat 2 (Dua), walaupun kedua siswa tersebut memiliki jumlah nilai yang sama. Tetapi apabila terjadi kesamaan nilai ditiap mata pelajaran, baik mata pelajaran Agama, Eksak dan Muatan Lokal sama-sama kuat dan imbang, maka cara menentukan peringkat atau rangking dilihat dari kepribadian siswa tersebut, tentunya seorang guru dapat menentukan siswa mana yang mempunyai kepribadian lebih baik dan siswa tersebutlah yang seharusnya menduduki peringkat 1 (Satu).