

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan. Selain itu, sistem juga dapat di definisikan sebagai sekumpulan objek – objek yang saling berelasi dan berinteraksi, serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan yang telah ditetapkan.

Sistem yang baik adalah sistem yang selalu menyesuaikan dengan perubahan lingkungan disekitarnya. Sistem tersebut harus dinamis menuju pada keadaan yang lebih baik secara berkelanjutan. Tahap awal yaitu perencanaan, tahapan ini menyangkut studi kebutuhan user, studi kelayakan baik secara teknis maupun teknologi serta penjadwalan pengembangan suatu proyek sistem informasi. Tahap kedua adalah analisis, yaitu tahap saat kita berusaha mengenali segenap permasalahan yang muncul pada pengguna, mengenali komponen – komponen sistem, obyek – obyek, hubungan antar obyek dan sebagainya. Tahap ketiga adalah perancangan, yaitu tahap saat kita mencoba mencari solusi permasalahan dari tahap analisis. Tahap keempat adalah implementasi, yaitu tahap saat kita mengimplementasikan perancangan sistem ke situasi yang nyata dengan pemilihan

perangkat keras, kemudian masuk dalam penyusunan perangkat lunak dan aplikasi. Tahap terakhir adalah pemeliharaan atau perawatan, pada tahap ini kita bisa memulai melakukan pengoperasian sistem dan jika diperlukan dapat melakukan perbaikan, kemudian jika waktu pengguna sistem habis, maka kita masuk lagi pada tahap perencanaan (Hamim Tohari ; 2014: 2 - 6).

II.2. Keputusan

Menurut Nadwa dalam Jurnal Pendidikan Islam (2014) Keputusan (*decision*) secara harfiah berarti pilihan (*choice*). Pilihan yang dimaksud disini adalah pilihan dari dua atau lebih kemungkinan, atau dapat dikatakan pula sebagai keputusan dicapai setelah dilakukan pertimbangan dengan memilih satu kemungkinan pilihan.

Definisi diatas mengandung pengertian, dalam keputusan yaitu :

1. Ada pilihan atas dasar logika atau pertimbangan.
2. Ada beberapa alternatif yang harus dipilih salah satu yang terbaik.
3. Ada tujuan yang ingin dicapai dan keputusan itu makin mendekatkan pada tujuan tersebut.

II.3. Sistem Pendukung Keputusan (*Decision Support System*)

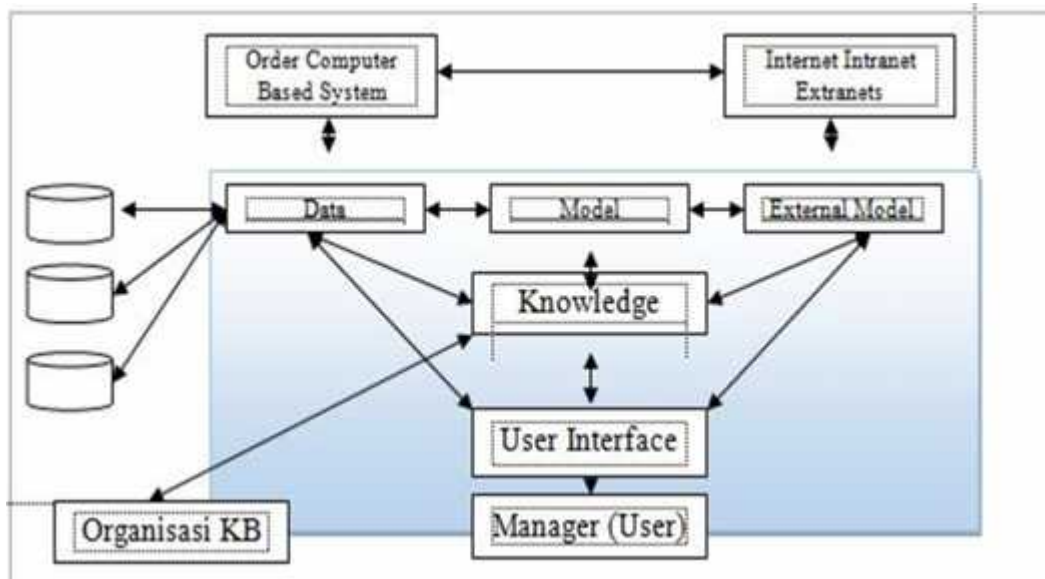
Sistem adalah kumpulan dari obyek-obyek seperti manusia, *resources*, konsep, dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau memenuhi suatu tujuan. Ludwig Von Bertalanfy berpendapat bahwa sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur

tersebut dengan lingkungan. Sedangkan menurut Jogiyanto sistem adalah komponen-komponen yang saling berhubungan untuk mencapai satu tujuan tertentu. Suatu sistem dapat dirumuskan sebagai setiap kumpulan bagian-bagian atau subsistem-subsistem yang disatukan, yang dirancang untuk mencapai suatu tujuan. Kata keputusan (*decision*) berarti pilihan (*choice*), yaitu pilihan dari dua atau lebih kemungkinan. Keputusan juga dapat berarti kegiatan memilih suatu strategi atau tindakan dalam pemecahan suatu masalah. Tindakan memilih strategi atau aksi yang diyakini manajer akan memberikan solusi terbaik atas sesuatu itu disebut pengambilan keputusan (ma'ruf, 2016).

Sistem Pendukung Keputusan (SPK) didefinisikan sebagai sebuah sistem yang bertujuan untuk mendukung seorang manajer mengambil sebuah keputusan dalam kondisi permasalahan yang semi terstruktur. SPK difungsikan sebagai fasilitas yang dapat memperkuat kapabilitas sang pengambil keputusan, namun tidak sepenuhnya menggantikan peran pengambil keputusan tersebut. SPK digunakan pada pengambilan keputusan yang melibatkan pertimbangan dari manajer, atau pada pengambilan keputusan yang tidak sepenuhnya dapat diselesaikan dengan perhitungan. Komponen Sistem Pendukung Keputusan (SPK) terdiri dari: (ma'ruf, 2016).

1. Subsistem Manajemen Data (*Data-management Subsystem*).
2. Subsistem Manajemen Model (*Model Management Subsystem*).
3. Subsistem Antarmuka Pengguna (*User Interface Subsystem*).

4. Subsystem Manajemen Berbasis Pengetahuan (*Knowledge-based Management Subsystem*) (ma'ruf, 2016).



Gambar II.1 Skema Sistem Pengambil Keputusan
Sumber : Ma'ruf, 2016

II.4. Mobil Bekas

Mobil bekas adalah kendaraan yang telah dipakai oleh pemilik mobil dalam jangka waktu tertentu sehingga menimbulkan penyusutan. Nilai penyusutan mobil bekas yang membuat harganya turun. Jual beli mobil bekas telah berlangsung lama dan menjadikan ceruk bisnis tersendiri. Sekarang ini sudah banyak sekali *showroom* mobil bekas yang menawarkan harga dan kualitas yang sepadan. Bisnis mobil bekas yang semakin pesat merupakan pesaing baru bagi dealer-dealer mobil baru yang dapat merebut pasarnya. Pasar mobil bekas yang besar juga membuat para

perusahaan pembiayaan berlomba-lomba membiayai konsumen secara kredit dengan bunga yang kompetitif yang tidak kalah dengan mobil baru.

Kualitas produk mobil bekas sangat mempengaruhi minat konsumen untuk membeli. Kualitas produk mobil bekas antara lain dapat diukur dari tahun perakitan, kilometer, kondisi fisik interior dan eksterior, kondisi mesin, kondisi ban, warna serta atribut-atribut lainnya yang membuat harga mobil tersebut dapat berubah ubah. Kualitas produk sebuah mobil bekas menjadi tolak ukur konsumen dalam menilai kelayakan mobil untuk dibeli. (Indra Sasmita et al : 2013)

II.4.1. Showroom

Pengertian *showroom* adalah ruang pameran biasa didefinisikan sebagai tempat untuk memamerkan produk tertentu, seperti otomotif, furniture, dll yang berfungsi untuk meningkatkan pemasaran (Arvin Dovan Sulaksono : 2016).

Masyarakat umum menyebut demikian karena secara global dan sudah menjadi kebiasaan berasumsi bahwa tempat untuk memajang bernama *showroom*. Dimana *showroom* terdiri dari dua kategori yaitu *showroom* mobil baru atau *showroom* mobil bekas yang kerap kali disebut *showroom* mobkas. Dimana *showroom* mobil baru atau ruang pameran mobil adalah menjual dengan satu merk yang sudah mendunia dan berada dalam satu negara terdiri dari beberapa cabang dalam satu kota dan hanya ada satu pusat dalam satu negara. *Showroom* mobil bekas disimpulkan sebagai suatu ruang pameran mobil yang menjual berbagai jenis merk kendaraan bermotor yang tergolong sebagai *used car*. Kata-kata *showroom* pun

menjadi sama pengertiannya yaitu tempat untuk memajang mobil. Selain itu *showroom* juga dapat diartikan sebagai tempat *display* untuk *furniture* ataupun barang yang memang untuk dipamerkan. Tujuan *showroom* dibuat adalah untuk memberikan fasilitas akan kebutuhan, dimana kebutuhan kendaraan saat ini sangat kompleks.

II.5. TOPSIS (*Technique For Others Reference by Similarity to Ideal Solution*)

TOPSIS adalah metode teknik pembobotan linear dalam pengambilan keputusan multi-kriteria dengan asumsi dasarnya adalah solusi terbaik harus sedekat mungkin dengan solusi ideal positif dan sejauh mungkin dengan solusi ideal negatif.

Secara umum, prosedur TOPSIS mengikuti langkah-langkah sebagai berikut:

1. Membuat matriks keputusan yang ternormalisasi
2. Membuat matriks keputusan yang ternormalisasi terbobot
3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif
5. Menentukan nilai preferensi untuk setiap alternatif.

TOPSIS membutuhkan rating kerja setiap alternatif A_i pada setiap kriteria C_j yang ternormalisasi (Fahmiyadi ZA et al, 2015).

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} \dots \dots \dots (1)$$

$$r = \frac{X}{\sqrt{\sum_{i=1}^m X}} \dots \dots \dots (2)$$

Penghitungan bobot setiap kriteria perbandingan. Penghitungan dari bobot setiap kriteria perbandingan didasarkan pada perhitungan dari nilai entropi dan kemudian mengubahnya menjadi bobot yang dijelaskan pada dua langkah

Pertama hitung nilai entropi setiap kriteria C_1, C_2, \dots, C_n . Bobot dari setiap kriteria dihitung dengan memperkenalkan konsep entropi.

II.6. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) Menurut Pender di dalam jurnal Nandik Sesnika (2016) *Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, memspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *Object Oriented*. UML terdiri atas pengelompokan diagram-diagram sistem. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Salah satu cara untuk mengatur diagram UML adalah *View*. *View* adalah kumpulan diagram yang menggambarkan aspek yang sama dari proyek yang terdiri dari *Static View*, *Dynamic View*, dan *Functional View*.

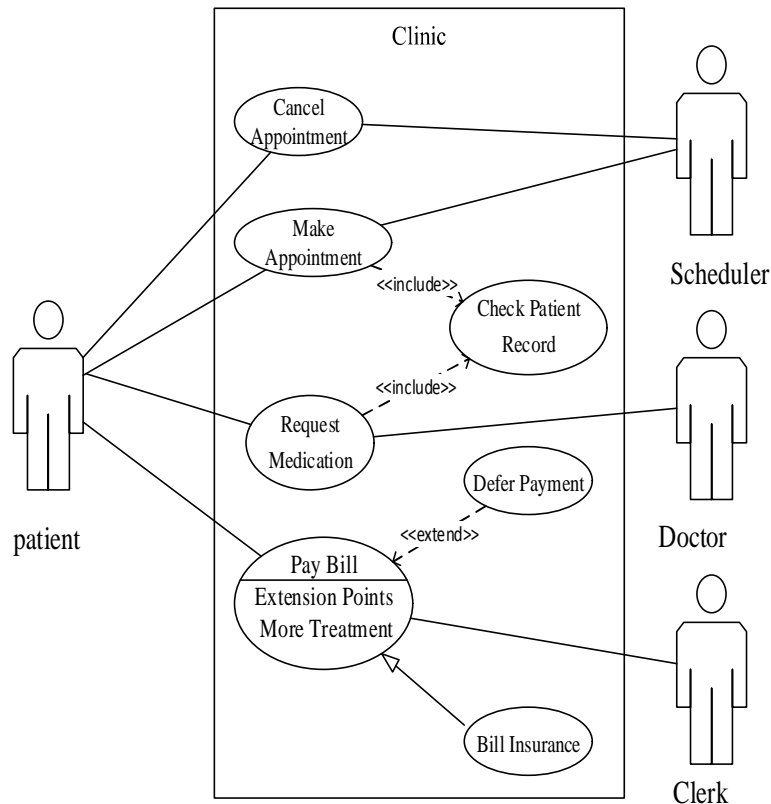
II.6.1. *Use case diagram*

Use case diagram adalah rangkaian atau uraian kelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

Use case digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.

Diagram *use case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Hal yang ditekankan pada diagram ini adalah “ apa “ yang diperbuat sistem, bukan “ bagaimana “. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* menyatakan sebuah aktivitas atas pekerjaan tertentu, misalnya *login* ke sistem, meng *-create* sebuah daftar belanja, dan sebagainya. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sebuah sistem untuk melakukan pekerjaan – pekerjaan tertentu.

Sebuah *use case* dapat meng *-include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum, di asumsikan bahwa *use case* yang di *-include* akan di panggil setiap kali *use case* yang meng *-include* di eksekusi secara normal. Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain (Hamim Tohari ; 2014 : 47-48).



Gambar II.2. : Contoh Diagram Use case
(Sumber : Hamim Tohari ; 2014 : 49)

II.6.2. 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 bentuk sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (Hamim Tohari ; 2014 : 83).

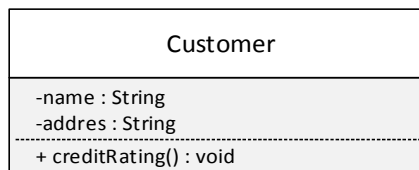
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (*stereotype*)
2. Atribut
3. Metode atau operasi

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.



Gambar II.3. *Class Diagram*
(Sumber : Hamim Tohari ; 2014 : 84)

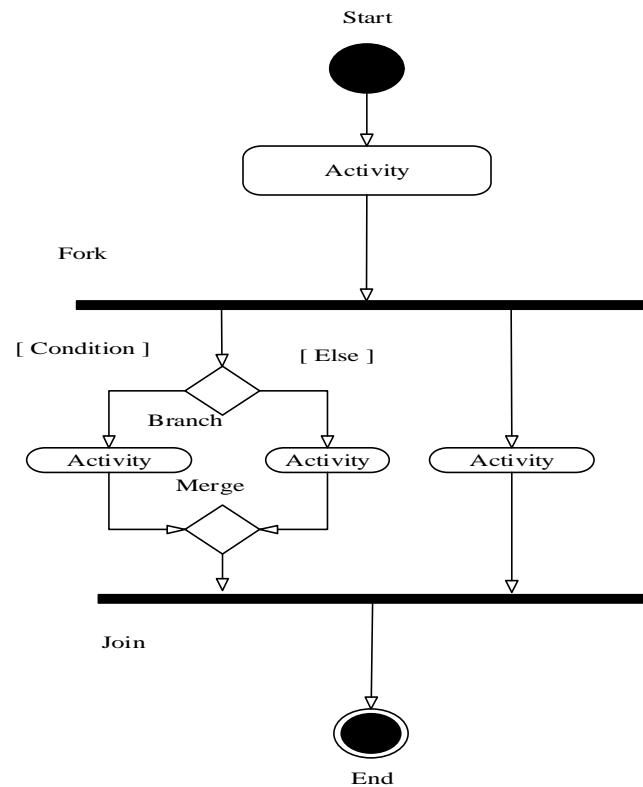
II.6.3. *Activity Diagram*

Menurut Hofmeister dkk di dalam jurnal Edgar Winata (2013) *Activity Diagram* merupakan teknik untuk menjelaskan business process, procedural logic, dan work flow. Bisa dipakai untuk menjelaskan teks *use case* dalam notasi grafis

dengan menggunakan notasi yang mirip flow chart, meskipun terdapat sedikit perbedaan notasi.

Di dalam *Activity Diagram* terdapat beberapa bagian :

- a. *Nodes*, menandakan *initial* dan *final node*, *final node* boleh lebih dari satu.
- b. *Activity*, aktivitas sistem dapat berupa aktifitas fisik juga bagi *user*.
- c. *Flow/edge*, arah sebuah proses.
- d. *Fork*, awal sebuah paralel.
- e. *Join*, akhir proses paralel.
- f. *Condition*, kondisi yang di tuliskan dalam bentuk teks.
- g. *Decision*, implementasi *if* dan *then*.
- h. *Merge*, penyatuan beberapa *flow*.
- i. *Partition*, siapa atau apa yang menjalankan aktivitas.



Gambar II.4. Contoh Activity Diagram

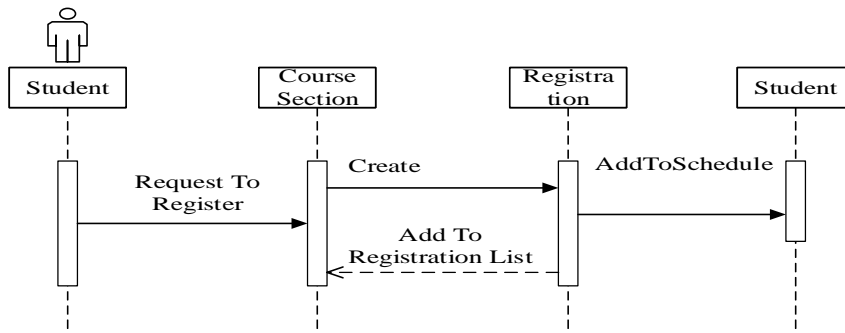
(Sumber : Edgar Winata, 2013)

II.6.4. Sequence Diagram

Menurut Hofmeister dkk di dalam jurnal Edgar Winata (2013) *sequence diagram* menjelaskan interaksi obyek – obyek yang paling berkolaborasi (berhubungan), mirip dengan *activity diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau *behaviour* yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah program.

Dalam sequence diagram terdapat beberapa bagian :

- Participant*, yaitu obyek yang terkait dengan sebuah urusan proses.
- Lifeline*, menggambarkan daur hidup sebuah obyek.
- Activation*, suatu titik waktu dimana sebuah obyek mulai berpartisipasi dalam sebuah sequence.
- Time*, elemen paling penting dalam sequence diagram yang konteksnya adalah urutan, bukan durasi.
- Return*, suatu hasil kembalian sebuah operasi. Operasi mengembalikan hasil tetapi boleh tidak di tulis jika tidak ada perbedaan dengan *Getter* – nya.



Gambar II.5. Contoh Sequence Diagram

(Sumber : Edgar Winata : 2013)

II.7. Database

Database adalah sekumpulan data yang tersimpan pada sebuah media penyimpanan komputer yang memiliki hubungan secara logika dengan tujuan untuk melakukan proses terhadap data-data yang menghasilkan informasi tertentu (Rizal Saiful Hamdhani, et al, 2015)

II.7.1 Normalisasi Database


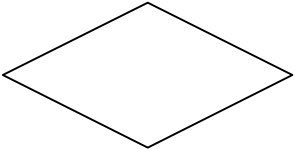
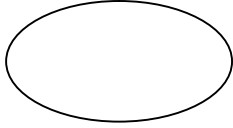

Menurut Eka Rindi Handayani (2014) tujuan normalisasi adalah untuk .menghilangkan dan mengurangi redudansi data dan memastikan depedensi data. Jika data dalam *database* tersebut belum di normalisasikan maka akan terjadi 3 kemungkinan yang akan merugikan sistem secara keseluruhan yaitu :

1. *INSERT* Anomali : Situasi dimana tidak memungkinkan memasukkan beberapa jenis data secara langsung di *database*.
2. *DELETE* Anomali: Penghapusan data yang tidak sesuai dengan yang diharapkan, artinya data yang harusnya tidak terhapus mungkin ikut terhapus.
3. *UPDATE* Anomali: Situasi dimana nilai yang diubah menyebabkan inkonsistensi *database*, dalam artian data yang diubah tidak sesuai dengan yang diperintahkan atau yang diinginkan.

II.7.2. *Entity Relationship Diagram*(ERD)

Entity Relational Diagram (ERD) merupakan diagram yang menunjukkan hubungan antara entitas-entitas yang ada dalam suatu sistem entitas tentang apa data itu berbicara. *Entity Relationship Diagram* berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut –atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R). Adapun simbol-simbol yang digunakan dalam ERD adalah :

Tabel II.7. Simbol *Entity Relationship Diagram*

Simbol	Nama
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

(Sumber : Alfian Helmi Muhbib : 2013)

II.8. SQL Server 2008

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun

dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional. (Herpendi : 2016)

II.9. Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, *Software Development Kit (SDK)*, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework). (Herpendi : 2016)

