

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem**

Menurut Abdul Kadir (2017), Sistem adalah kumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama maka elemen tersebut dapat dipastikan bukan bagian dari sistem. Sehingga dapat disimpulkan bahwa suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

#### **2.2 Informasi**

Menurut Abdul Kadir (2017) Informasi berarti data yang telah diproses sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Sebaliknya data merupakan sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau lingkungan fisik sebelum diolah dalam suatu format yang dapat dipahami dan digunakan orang.

#### **2.3 E-Recruitment**

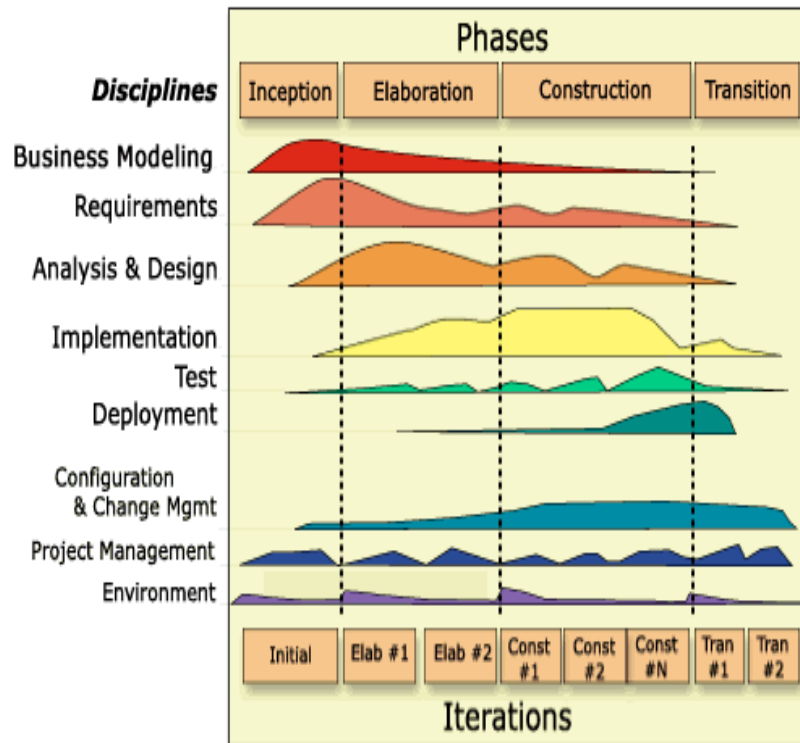
Menurut Kapse dkk (2017) mengemukakan bahwa: *e-Recruitment* adalah penggunaan teknologi untuk membantu proses perekrutan. Mereka mengiklankan lowongan pekerjaan melalui *web* di seluruh dunia. Pencari kerja mengirimkan aplikasi mereka atau *curriculum vitae* yaitu CV melalui *e-mail* menggunakan *Internet*. Atau pencari kerja menempatkan CV mereka di *web* di seluruh dunia, yang dapat ditarik oleh calon karyawan tergantung pada kebutuhan mereka.

## 2.4 Metode Pengembangan Sistem

Alat Pengembangan Sistem adalah Menyusun sesuatu yang baru untuk menggantikan atau mengembangkan sistem yang lama secara keseluruhan.

Menurut Sukamto dan Shalahuddin (2018), *RUP (Rational Unified Process)* adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*).

*Rational Unified Process (RUP)* adalah kerangka proses yang menyediakan simulasi sistem pada industri untuk sistem, software, implementasi, dan manajemen proyek yang efektif. RUP adalah salah satu dari sekian banyak proses yang terdapat di dalam *Rational Process Library*, yang memberikan simulasi terbaik untuk pengembangan atau kebutuhan proyek (Rosadi, 2017) RUP mempunyai beberapa tahapan, yaitu :



**Gambar 2. 1 Metode RUP(Rasional Unified Process)**  
(Rosadi, 2017)

RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur, lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik dan penstrukturan yang baik. RUP memiliki empat buah tahap fase sebagai berikut (Rosadi, 2017):

### **1. Inception**

*Inception* merupakan tahap untuk mengidentifikasi sistem yang akan dikembangkan. Aktivitas yang dilakukan pada tahap ini antara lain

- a. Analisis sistem *existing* (sistem yang berjalan)
- b. Perumusan sistem target
- c. Identifikasi kebutuhan dan perumusan persyaratan (fungsional, performansi, keamanan, *GUI*, dll),
- d. Perumusan kebutuhan pengujian (level unit, integrasi, sistem, performansi, fungsionalitas, keamanan, dll), UML diagram, dan pembuatan dokumentasi.

### **2. Elaboration**

*Elaboration* merupakan tahap untuk melakukan desain secara lengkap berdasarkan hasil analisis pada tahap *inception*. Aktivitas yang dilakukan pada tahap ini antara lain

- a. Pembuatan desain arsitektur subsistem (*architecture pattern*),
- b. Desain komponen sistem,
- c. Desain *database*,
- d. Desain *user interface*,
- e. Pemodelan diagram *UML* (diagram *sequence*, *class*, *component*, *deployment*, dll.), dan
- f. Pembuatan dokumentasi

### **3 Construction**

*Construction* merupakan tahap untuk mengimplementasikan hasil desain dan melakukan pengujian hasil implementasi. Pada tahap awal construction, ada baiknya dilakukan pemeriksaan ulang hasil analisis dan desain, terutama desain pada *sequence* diagram, *class* diagram, *component* dan *deployment*. Apabila desain yang dibuat telah sesuai dengan analisis sistem, maka implementasi dengan bahasa pemrograman tertentu dapat dilakukan. Aktivitas yang dilakukan pada tahap ini antara lain

- a. Pengujian hasil analisis dan desain,
- b. Pendataan kebutuhan implementasi lengkap (berpedoman pada identifikasi kebutuhan di tahap analisis),
- c. Penentuan coding pattern yang digunakan,
- d. Pembuatan program,
- e. Pengujian dan optimasi program,
- f. Pendataan berbagai kemungkinan pengembangan atau perbaikan lebih lanjut, dan
- g. Pembuatan dokumentasi.

### **4. Transition**

*Transition* merupakan tahap untuk menyerahkan sistem aplikasi kepada *user* (*roll-out*), yang umumnya mencakup

- a. pelatihan dan
- b. beta *testing* aplikasi.

## **2.5 Alat Pengembangan Sistem**

*Unified Modelling Language* (UML) 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. Tujuan penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. Tipe-tipe dari diagram UML adalah sebagai berikut. (Achmad Sidik, Bronto Waluyo E.T, dan Siti Susilawati, 2018).




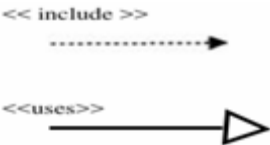
### 2.5.1 Use Case Diagram

*Use Case Diagram* adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal atau pengguna. Secara grafis *Use Case Diagram* menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan interaksi dengan sistem.

*Use case* sendiri menggambarkan interaksi antara sistem, sistem eksternal, dan *user*, dengan kata lain mendeskripsikan siapa yang menggunakan sistem dan dengan cara apa pengguna mengharapkan interaksi dengan sistem itu. Simbol *Use Case diagram* dapat dilihat pada table di bawah ini. (Muhamad dan Insan 2019).

**Tabel 2.1 Simbol Use Case Diagram**



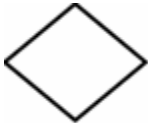


No	Simbol	Keterangan
1	<p><i>Use cases</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	<p>Aktor / <i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .

3	Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .
4	Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
5	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / <i>include / uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> .

### 2.5.2 Activity Diagram

*Activity Diagram* adalah diagram yang secara grafis menggambarkan aliran aktifitas baik proses bisnis atau *use case*. Diagram ini juga dapat digunakan untuk memodelkan aksi yang akan dilakukan saat sebuah operasi di eksekusi dan memodelkan hasil dari aksi tersebut. Simbol *activity diagram* dapat dilihat pada *table* di bawah ini. (Muhamad dan Insan 2019).

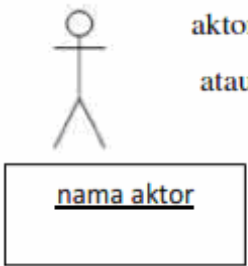

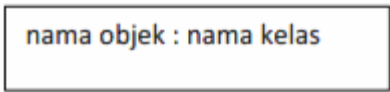


**Tabel 2.2 Simbol Activity Diagram**

No	Simbol	Keterangan
1	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	Keputusan / <i>decision</i> 	Asosiasi dimana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

### 2.5.3 Sequence diagram


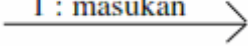
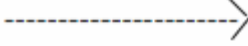
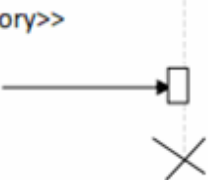
*Sequence diagram* menggambarkan bagaimana objek berinteraksi satu sama lain melalui pesan dalam *use case* atau suatu operasi. *Sequence diagram* menggambarkan bagaimana pesan dikirim dan diterima antara objek dan dalam satu urutan. Sebuah *sequence diagram* memiliki aktor beserta dengan pesan dan fungsi dimana pesan tersebut dikerjakan dengan berurutan. (Irani Yuni M.N dan Sri Muryani ,2018).

**Tabel 2.3 Simbol Sequence Diagram**

No	Simbol	Keterangan
1	<p>Aktor</p>  <p>aktor atau</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i></p>
2	<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi Pesan</p>
4	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>
5	<p>Pesan tipe create</p> <p>&lt;&lt;create&gt;&gt;</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>



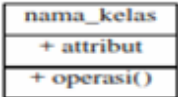






**Tabel 2.3 Simbol Sequence Diagram (Lanjutan)**

6	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7	Pesan tipe send 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah kepada objek yang dikirim
8	Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	Pesan tipe destroy <<destory>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

#### 2.5.4 Class diagram

*Class diagram* adalah gambar grafis mengenai struktur objek statis sebuah sistem menunjukkan kelas-kelas objek yang menyusun sebuah sistem dan juga hubungan antara kelas objek tersebut. Simbol *Class Diagram* dapat dilihat pada *table* di bawah ini. (Muhamad dan Insan 2019).

**Tabel 2.4 Simbol Class Diagram**

No	Simbol	Keterangan
1	<p style="text-align: center;">Kelas</p> 	Kelas pada struktur system
2	<p style="text-align: center;">Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	<p style="text-align: center;">nama_interface Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4	<p style="text-align: center;">Asosiasi berarah / <i>directed Association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5	<p style="text-align: center;">Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
6	<p style="text-align: center;">Kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas
7	<p style="text-align: center;">Agregasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semua bagian ( <i>whole-part</i> )

## **2.6 Software yang Digunakan**

### **2.6.1 *Adobe Dreamweaver***

*Dreamweaver* merupakan suatu perangkat lunak web editor yang digunakan untuk membangun dan mendesain suatu website dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya. Adobe Dreamweaver adalah salah satu program editor halaman web atau merupakan program penyunting-pengolah halaman web keluaran Adobe Systems yang dulu dikenal sebagai Macromedia Dreamweaver keluaran Macromedia. Program ini banyak digunakan oleh pengembang *web* karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 8. Versi terakhir Dreamweaver keluaran Adobe Systems adalah Adobe Dreamweaver CS6 yang ada dalam Adobe Creative Suite 5 (Sulhan, 2017).

### **2.6.2 PHP**

PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi berbasis web. PHP adalah kependekan dari PHP: *Hypertext Preprocessor*, bahasa *interpreter* yang mempunyai kemiripan dengan C dan Perl. PHP dapat digunakan bersama dengan HTML sehingga memudahkan dalam membuat aplikasi *web* dengan cepat. Dapat digunakan untuk menciptakan *dynamic website* baik itu yang memerlukan penggunaan *database* ataupun tidak.. PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain. (Achmad Sidik, Bronto Waluyo E.T, dan Siti Susilawati, 2018).

### **2.6.3. XAMPP**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang dirilis dengan bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan *web server* yang mudah digunakan yang apat melayai tampilan halaman web yang dinamis. XAMPP dikembangkan dari sebuah tim proyek bernama *Apache Friends*, yang terdiri dari Tim Inti (*Core Team*), Tim Pengembang (*Development Team*) & Tim Dukungan (*Support Team*) (Sulhan, 2017).

**XAMPP** adalah kepanjangan yang masing-masing hurufnya adalah:

- X** : Program ini dapat dijalankan dibanyak sistem operasi, seperti Windows, Linux, Mac OS, danjuga Solaris.
- A** : **Apache**, merupakan aplikasi web server. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat web, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman web yang dihasilkan.
- M** : **MySQL**, merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structure Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database besertaisinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database.
- P** : **PHP**, bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL.
- P** : **Perl** adalah bahasa pemrograman untuk segala keperluan, dikembangkan pertama kali oleh Larry Wall di mesin Unix. Perl dirilis pertama kali pada tanggal 18 Desember 1987 ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl tersedia pula untuk berbagai sistem operasi varian Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk

sistem operasi seperti DOS, Windows, Power PC, BeOS, VMS, EBCDIC, dan PocketPC.

#### 2.6.4 MySQL

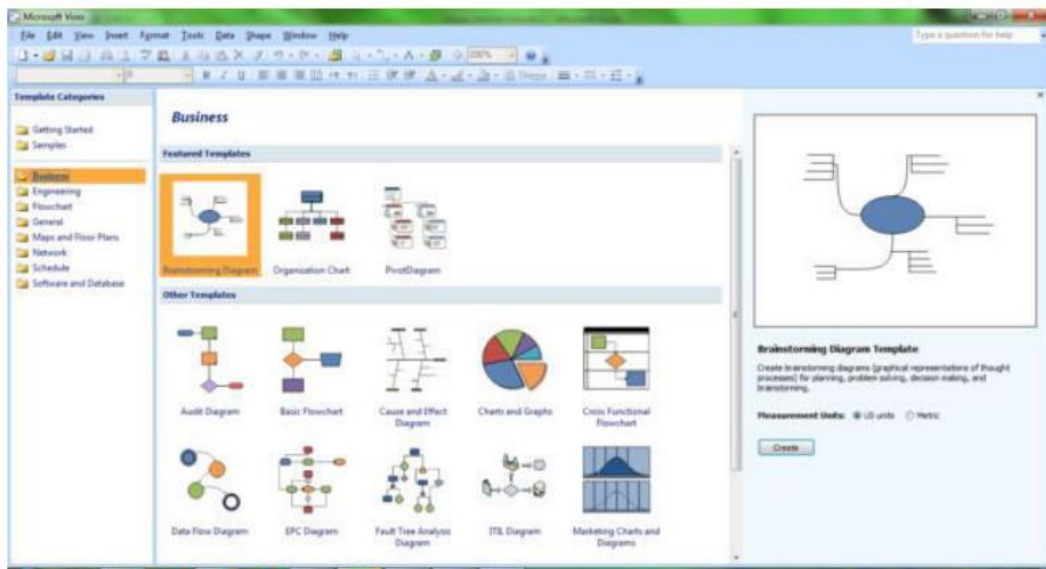
MySQL Merupakan database server yang paling sering digunakan dalam pemrograman PHP. MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan data dengan sangat cepat, MySQL memiliki dua bentuk lisensi yaitu *free software* dan *shareware*. MySQL yang bisa kita gunakan adalah MySQL *free software* yang berada di bawah lisensi GNU / QPL (*General Public License*).

Database adalah sistem penyimpanan beragam jenis data dalam sebuah entitas yang besar untuk diolah sedemikian rupa agar mudah dipergunakan lagi. Data yang disimpan bisa sangat variatif (angka, teks, gambar, suara, dan jenis data multi-media lainnya). *Web database* (basis data berbasis *web*) yaitu suatu sistem pengolahan dan penyimpanan data yang dapat diakses oleh bahasa pemrograman tertentu. *Web database* ini bersifat umum karena dapat diakses oleh aplikasi web yang dapat berjalan di berbagai macam platform, seperti aplikasi yang dikembangkan dengan tag HTML dan pemrograman yang bersifat *server-side* seperti PHP (Sulhan, 2017).

MySQL adalah sistem manajemen basisdata relasi yang bersifat terbuka atau *open source*. Sistem manajemen basisdata ini adalah hasil pemikiran dari Michael “Monty” Widenius, David Axmark, dan Allan Larson pada tahun 1995. Tujuan awal ditulisnya program MySQL adalah untuk mengembangkan aplikasi *web*. MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. Perintah SQL sering juga disebut *Query*. MySQL menawarkan berbagai keunggulan dibandingkan *database server* lain. (Achmad Sidik, Bronto Waluyo E.T, dan Siti Susilawati, 2018).

## 2.6.5 Microsoft Visio

Microsoft Visio (atau sering disebut Visio) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir, *brainstorm*, UML, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya (Sugianto,2018). Visio aslinya bukanlah buatan Microsoft Corporation, melainkan buatan Visio Corporation, yang diakuisisi oleh Microsoft pada tahun 2000. Versi yang telah menggunakan nama Microsoft Visio adalah Visio 2002, Visio 2003, Visio 2007, Visio 2013, dan Visio 2016 yang merupakan versi terbaru. Visio 2007 (atau lebih baru) Standard dan Professional menawarkan antarmuka pengguna yang sama, tetapi seri Professional menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (*layout*). Selain itu, edisi Professional juga memudahkan pengguna untuk mengoneksikan diagram-diagram buatan mereka terhadap beberapa sumber data dan juga menampilkan informasi secara visual dengan menggunakan grafik.



**Gambar 2.2 Tampilan Microsoft Visio**