BAB II

TINJAUAN PUSTAKA

II.1. Konsep Sistem

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatau tujuan. Suatu organisasi seperti perusahaan atau suatu bidang fungsional cocok dengan definisi ini, organisasi terdiri dari sejumlah sumber daya dan sumber daya tersebut bekerja menuju tercapainya suatu tujuan tertentu yang ditentukan oleh pemilik atau manajemen.Sistem adalah suatu intergrasi elemen-elemen yang semua nya bekerja menuju satu tujuan. Semua sistem meliputi tiga elemen utama yakni *input*, transformasi, dan *output*. Sebagian sistem dapat mengendalikan operasi merreka sendiri, dan disebut lingkaran tertutup (closed-loop system).Sistem lingkaran tertutup mencakup suatau mekanisme pengendalian, tujuan, dan lingkaran umpan balik (feedback-loop). (Mcleod,2001 : 11).

Atau sistem juga dapat diartikan sebagai proses sekumpulan elemen yang berhubungan satu dengan yang lainnya secara fungsional. Agar suatu sistem terlaksana diperlukan data yang relevan, akurat, tepat guna dan tepat waktu yang memungkinkan pihak manajemen dapat mengambil suatu keputusan yang tepat.

II.1.1 Konsep Informasi

Informasi adalah data yang telah diproses, atau data yang memiliki arti.Data terdiri dari fakta-fakta dan angka-angka yang relative tidak berarti bagi pemakai, perubahan data menjadi informasi dilakukan oleh pengolah informasi (information processor), pengolah informasi adalah salah satu elemen kunci dalam sistem konseptual.Pengolah informasi dapat meliputi elemen-elemen computer, elemen-elemen

non computer, atau kombinasi keduanya. Informasi adalah salah satu dari lima jenis utama sumber daya yang dapat dipakai oleh manajer. Pengelolaan informasi semakin penting saat bisnis menjadi lebih rumit dan kemapuan computer berkembang. (Mcleod, 2001 : 15). Sistem informasi adalah suatau sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Mcleod, 2001 : 15).

II.1.2 Konsep Sistem Informasi

Sistem Informasi adalah sebagai suatu kumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambilan keputusan untuk mengendalikan informasi.

Pengertian Sistem Informasi secara umum adalah himpunan komponen yang terpadu dengan sistematis maupun formal yang mendukung atau melakukan fungsi informasi, manajemen dan menyajikan informasi untuk pengambilan keputusan dalam suatu organisasi guna menunjang fungsi manajemen sebagaimana yang diinginkan.

Sistem Informasi mempunyai beberapa komponen sebagai berikut :

1. Data/Input
2. Proses/Pengolahan
3. Informasi/Output

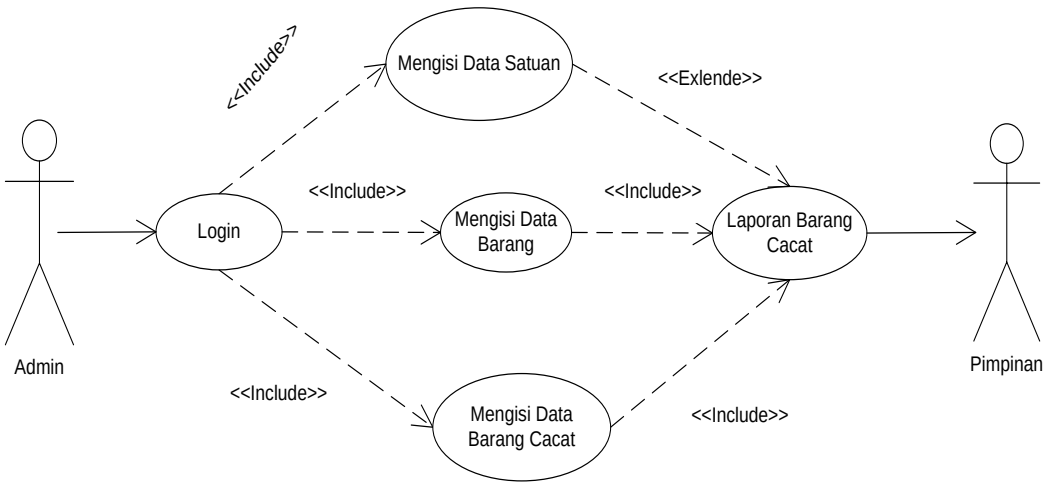
II.2. Alat Bantu yang Digunakan dalam Peracangan Sistem

Alat-alat yang digunakan dalam perancangan sistem informasi adalah diagram aliran data (*data flow diagram*/DFD), bagan alir (*flowchart)*, entry relation diagram (ERD), kamus data, normalisasi dan bagan tertrusktur, dan sebagainya. Namun dalam penulisan skripsi ini hanya digunakan DFD dan ERD.

II.2.1 UML (*Unified Modelling Language*)

1**.** *Use Case Diagram*

Use case adalah merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang aakan dibuat. Dengan pengertian yang cepat, diagram use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan *use case* yaitu *use case*, *actor* dan relasi.



***Use Case Diagram* Sistem Informasi Barang cacat**

Use case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi

pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Diagram use case menunjukkan 3 aspek dari sistem yaitu *: actor*, *use case* dan *sistem/ sub sistem boundary*. Actor mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.

2. *Activity Diagram* (Aktivitas)

Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas mendukung prilaku paralel.

Activity diagrams menggambarkan berbagai alir aktivitas dalam system yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa seksekusi.

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 digram* tidak menggambarkan *behaviour internal* sebuah system (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

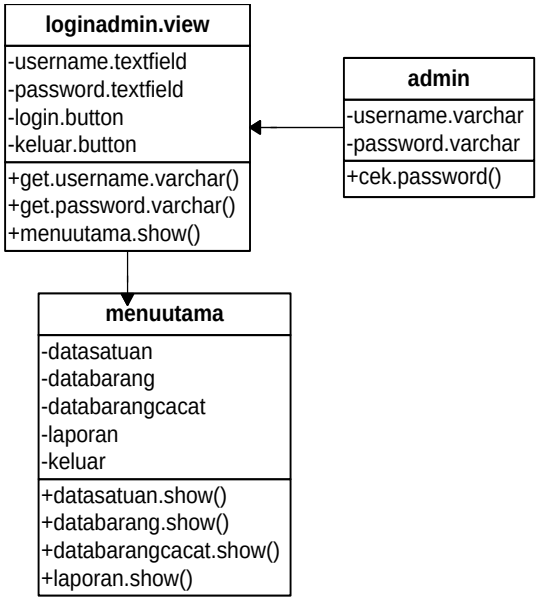
Diagram aktivitas juga banyak digunakan untuk mendefenisikan hal-hal berikut:

- Rancang proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefenisikan.

- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

3. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antara kelas seperti pada gambar berikut :

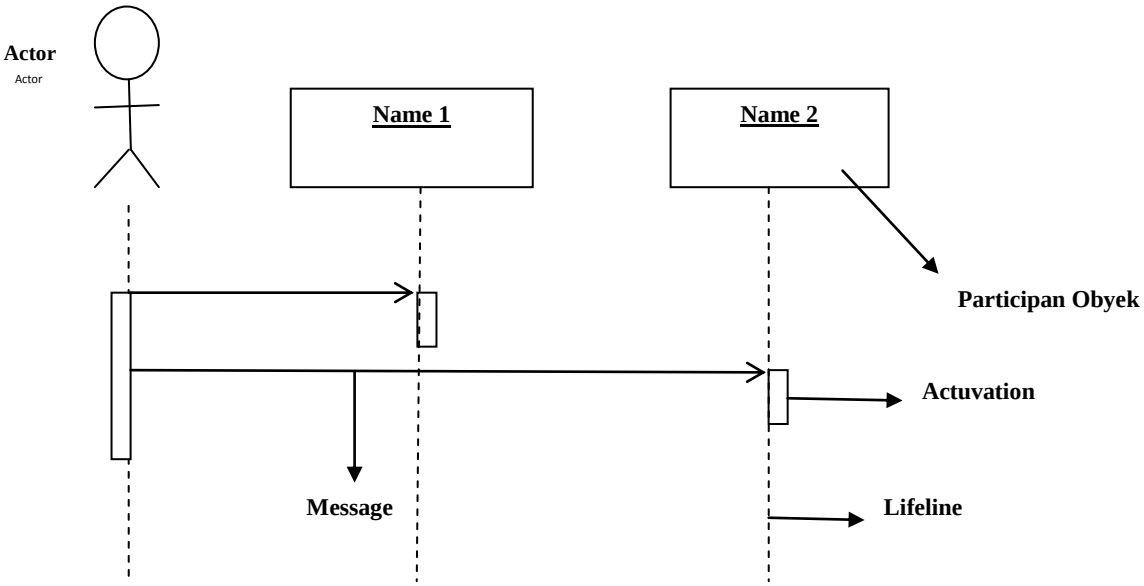


***Class Diagram Login* Admin**

Menurut Prabowo Pudjo Widodo dalam bukunya menggunakan UML (2011 : 10) class diagram bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodalan sistem berorientasi objek. Meskipun bersifat statis , sering pula diagram kelas memuat kelas-kelas aktif.

4. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku ada sebuah scenario. Diagram ini menunjukan sejumlah contoh objek dan message (pesan) yang diletakan diantara objek – objek ini di dalam *use case*.Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukan dengan progress vertical. Sequence diagram menambahkan dimensi waktu pada interaksi diantara objek. Pada diagram ini participant diletakan diatas dan waktu ditunjukan dari atas ke bawah



Simbol-simbol yang ada pada sequence diagram

II.2.3 Entity Relationship Diagram (ERD)

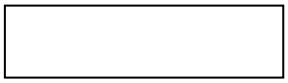
Entity Relationship Diagram atau ERD berfungsi untuk mendokomentasikan data perusahaan dengan mengidentifikasikan jenis entitas dan hubunganya.

Menurut Kuswayatno (2007 : 109) : "Diagram ER digunakan untuk menyatakan hubungan antar tabel. Hubungan dau tabel atau lebih ini biasanya dilakukan dengan mencocokkan data dalam field-field kunci (primary key).

Hubungan yang terjadi antara dua tabel atau lebih ini dapat terjadi hubungan satu ke satu, satu ke banyak, dan banyak ke banyak".

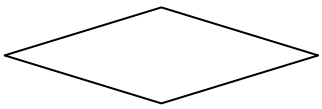
Komponen-komponen yang digunakan dalam entity relationship Diagram adalah :

1. Entity atau entitas, adalah sesuatu yang dapat dibedakan dalam suatu sistem, dari kumpulan data, atau elemen.



Gambar II.2 : Simbol Entity atau Entitas

2. Attributes adalah sesuatu yang merupakan sifat dan karakteristik dari entitas dan relasi.
3. Relationship, disebut juga hubungan entitas, biasanya terjadi antara sebuah entitas dengan satu atau lebih entitas lainnya.

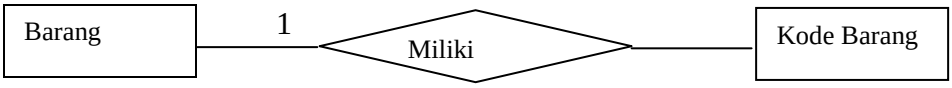


Gambar II.3 : Simbol Relationship

Didalam ERD terdapat tingkatan hubungan antar entitas dilihat dari segi banyak atau tidaknya hubungan antar entitas tersebut.Hal ini disebut mapping cardinalities.Ada tingkatan dari hubungan yang mungkin terjadi :

1. **One to One (1 : 1)**

Mempunyai hubungan satu berbanding satu,yaitu suatu entitas hanya mempunyai satu hunbungan dengan entitas lain, begitu pula sebaliknya. Sebagai contoh, satu barnag hanya dapat memiliki satu kodebarang. Relasi one to one dapat dilihat pada gambar 2.4



Gambar II.4 : One to One Relationship

2. **One to Many (1 : M)**

Atu entitas dapat mempunyai lebih dari satu hubungan dengan entitas lain. Sebagai contoh, satu pelanggan dapat memiliki beberapa faktur, tetapi satu faktur hanya dapat dimiliki oleh satu pelanggan. Relasi one to many dapat dilihat pada gambar 2.5.

Gambar II.5 : One to Many Relationship

3. Many to Many (M : M)

satu entitas dapat mempunyai lebih dari satu hubungan dengan entitas begitu pula sebaliknya. Sebagainya. Sebagai contoh, satu barang dapat tercantum dalam beberapa faktur, dan satu faktur dapat tercantum beberapa jenis barang Relasi many to many dapat dilihat pada gambar 2.6.

Gambar II.6 : Many to Many Relationship

II.3. Manajemen Basis Data

Basis data dapat diindentifikasikan dalam sejumlah sudut pandang seperti sebuah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.Kumpulan data yang saling berhubungan dan disimpan secara bersamaan sedemikian rupa tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

Prinsip utamanya adalah pengaturan data atau arsip dan tujuan utamanya adalah kemudahan dan kecepatan pengambilan data atau arsip. Sedangkan basis data (database) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

Satu database management Sistem (DBMS) berisi satu koleksi yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi DBMS terdiri dari

database dan set program pengelola untuk menambah data, mengambil dan membaca data.

Menurut Kuswayatno (2007 : 104) : “Database adalah representasi kumpulan data yang saling berhubungan yang disimpan secara bersama-sama dan tanpa pengulangan yang tidak perlu untuk memenuhi berbagai kebutuhan”.

Data perlu disimpan, diolah, dan diorganisasikan di dalam database sedemikian rupa sehingga informasi yang dihasilkan berkualitas dan efesien dalam penyimpanan data.Pengorganisasian data seperti ini dinamakan database management system (DBMS). Dalam satu DBMS terdiri dari beberapa file yang saling berhubungan dan terintegrasi.

Sedangkan dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk merupakan kumpulan yang seragam. Satu record terdiri dari field-field yang saling berhubungan untuk menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. Untuk menyebut dari isi dari field maka digunakan atribut atau merupakan judul dari satu kelompok entity tertentu, misalnya atribut alamat menunjukkan entity alamat dari siswa. Entity adalah suatu objek yang nyata dan akan direkam.

Kemudian set program pengelola merupakan satu paket program yang dibuat agar memudahkan dan mengefesiensikan pemasukan dan perekaman informasi dan pengambilan atau pembacaan informasi kedalam database.

Adapun syarat-syarat database dalam penyusunan data, yaitu :

1. Redudansi dan Inkonsistensi
    a. Penyusutan suatu database digunakan untuk mengatasi jika file dalam programaplikasi diciptakan oleh programmer yang berbeda pada waktu yang

berselang cukup panjang, Maka ada beberapa bagian data mengalami penggangaan (duplikasi data) pada file-file yang berbeda.

b. Kesulitan dalam pengaksesan data

Kesulitan yang timbul dalam pengaksesan data, penyelesaian untuk itu adalah kearah DBMS yang mampu mengambil data secara langsung dengan bahasa yang *familiar* dan mudah digunakan (*user friendly)*.

c. Isolasi data untuk standarisasi

jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka haruslah data dalam satu database dibuat satu format sehingga mudah dibuat program aplikasinya.

d. Banyak pemakai (multiple user)

Dalam rangka mempercepat semua daya guna sistem dan medapat response waktu yang cepat, beberapa sistem mengijinkan banyak pemakai untuk meng”update” data secara simultan. Salah satu alasan mengapa database dibangun karena nantinya data tersebut digunakan oleh banyak orang dalam waktu yang berbeda, diakses oleh program yang sama tetapi berbeda orang dan waktu. Semua ini memungkinkan terjadi karena data yang diolah tidaklah tergantung dan menyatu dalam program tapi terlepas dalam satu kelompok data.

e. Masalah Keamanan (Security)

Tidak setiap pemakai sistem database diperbolehkan untuk mengakses semua data.Keamanan ini dapat diatur lewat program yang dibuat atau fasilitas keamanan dari sistem operasi, seperti Novvel Network untuk LAN.

f. Masalah Kesatuan (Integrity)

Database berisi file-file yang saling berkaitan, masalah utama adalah bagaimana kaitan antara file tersebut terjadi. Meskipun kita mengetahui bahwa file A berkaitan dengan file B, namun secara teknis maka ada field kunci yang mengaitkan kedua file tersebut.

g. Masalah Kebebasan Data (Data Indenfence)

Paket bahasa yang diciptakan dari DBMS, apapun yang terjadi pada struktur file, setiap kali kita hendak melihat data cukuplah dengan *Utilty* LIST, hendak menambah data cukuplah dengan APPEND. Ini berarti perintah-perintah dalam paket DBMS bebas terhadap *database*, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah. Data independence dapat dibagi menjadi dua bagian, yaitu :

1. Physical Data Independence

   Kebolehan untuk mengubah pola fisik *database* tanpa mengakibatkan suatu aplikasi program ditulis kembali. Modofikasi pada level konseptual teristimewa saat struktur logika database berubah, ditambah atau dikurangi.

2. Logika Data Independence

   Kebolehan untuk mengubah pola konseptual tanpa mengakibatkan suatu aplikasi program ditulis kembali. Modifikasi pada level konseptual teristimewa saat struktur logika database berubah, ditambahkan atau dikurangi.

h. Abstraksi Data (Data Abtraction)

Kegunaan utama database adalah agar pemakai mampu menyusun pandangan abstraksi dari data. Bayangan mengenai data tidak lagi memperhatikan kondisi sesungguhnya bagaimana satu data masuk ke database, disimpan dalam disk disektor mana, tetapi menyangkut menyeluruh bagaimana data tersebut dapat diabstraksikan atau digambarkan menyerupai kondisi yang dihadapi oleh pemakai sehari-hari. Sistem yang sesungguhnya tentang teknis bagaimana data disimpan dan dipelihara seakan-akan disembunyikannya kerumitannya dan kemudian diungkapkan dalam bahasa dan gambar yang mudah dimengerti orang awam. Pemakai dapat dikelompokkan menjadi tiga tingkatan abstraksi saat memandang suatu *database*, yaitu :

1. Level Fisik

   Level abstraksi paling rendah, menggambarkan bagaimana data disimpan dalam kondisi sebenarnya. Level ini tentu paling kompleks, struktur data level terendah digambarkan pada level ini.

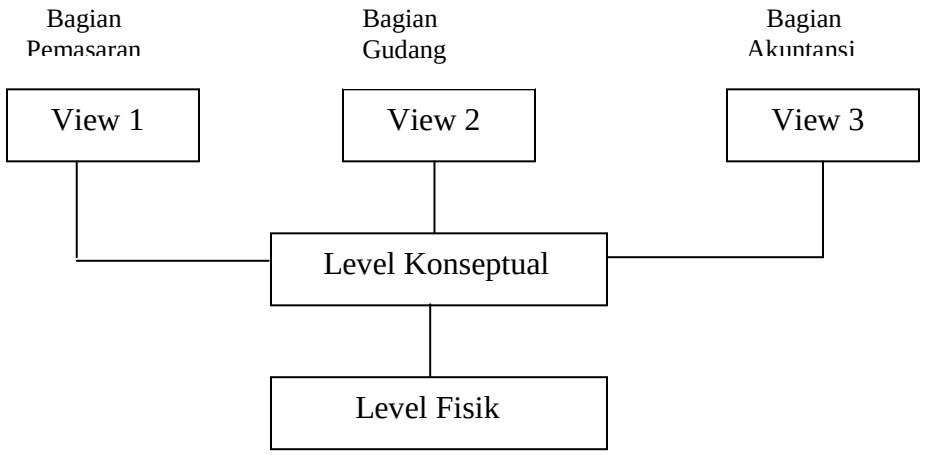
2. Level Konseptual

   Level abtraksi data level lebih tinggi yang menggambarkan data apa yang disimpan dalam *database*, dan hubungan relasi yang terjadi antar data. Level ini menggambarkan keseluruhan *database*. Pemakai tidak mempedulikan kerumitan dalam struktur level fisik lagi, penggambaran cukup dengan memakai kotak, garis dan keterangan secukupnya. Level konseptual ini digunakan oleh *databaseadministrator*, yang memutuskan informasi apa yang akan dipelihara dalam *database*.

3. Level Pandangan Pemakai (View Level)

Level abstraksi tertinggi yang menggambarkan hanya satu bagian dari keseluruhan *database*. Pada level ini hanya sebagian saja yang dapat dilihat dan dipakai. Hal ini disebabkan beberapa pemakai *database* tidak membutuhkan semua isi *database*.Level ini sangat dekat dengan pemakai.Setiap pemakai butuh sebagian dari *database*.Ada beberapa kelompok pemakai dengan pandangan berbeda butuh data dalam *database*.

Contoh hubungan antar level tersebut dapat digambarkan sebagai berikut :



Gambar II.7 : Level Pandangan Pemakai

II.4. Microsoft Visual Basic 2008

Visual Basic adalah bahasa pemrograman tingkat tinggi yang sudah sangat terkenal, di mulai dengan BASIC yang terdapat pada computer “angkatan tua”, seperti AT286.Pada saat itu, bahasa BASIC merupakan sebuah bahasa yang sangat diandalkan dalam pembuatan beberapa aplikasi penting.

Visual Basic 2008 merupakan software yang sering digunakan dalam pengembangan sistem aplikasi dan sistem informasi, yang memanfaatkan Visual Basic

2008 sebagai front-end atau user interface yang dapat digunakan pada aplikasi yang berskala kecil sampai dengan yang berskala besar.

Seiring dengan berkembangnya sistem operasi ke sistem operasi berbasis grafik, para pengguna DOS beralih ke WINDOWS. Keberadaan Windows mengilhami para programmer untuk menciptakan program dangan tampilan grafik yang mirip dengan windows, demikian juga dengan BASIC yang juga dikembangkan dengan tambahan beberapa fungsi untuk menciptakan tampilan grafik seperti Windows. Akan tetapi, pada saat itu para programmer masih harus menentukan sendiri koordinat sebuah jendela atau tombol dengan memasukkan koordinatnya.Hal ini berubah ketika Visual Basic muncul untuk pertama kalinya.Walaupun masih kurang sempurna, Visual Basic persi pertama sudah memikat hati para programmer.

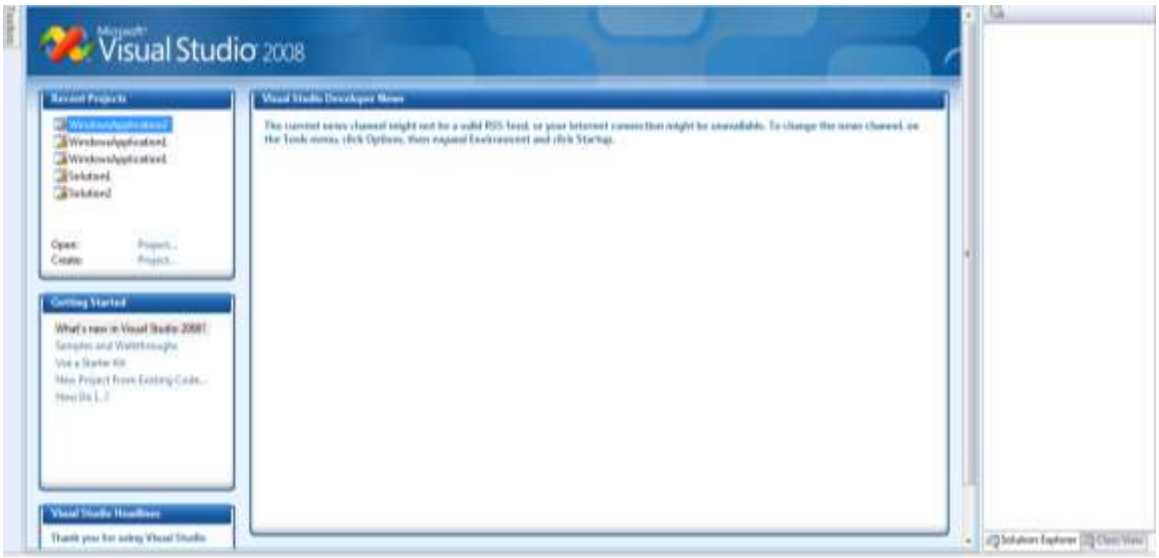
Kemudahan pemrograman dengan *drag and drop*ini kemudian lebih dikenal dengan Visual Programming, metode yang memungkinkan programmer tidak perlu memasukkan koordinat sebuah jendela lagi secara manual, termasuk pada perkembangannya, programmer hanya sedikit sekali melakukan *coding* (memasukkan kode program), demki terciptanya sebuah program aplikasi yang tangguh.

II.4.1 Mengenal Visual Basic 2008

Pada dasarnya, pembuatan program Visual Basic tidak tergantung dengan adanya Visual Basic .Net 2008, sehingga anda tidak memerlukan instalasi Visual Basic untuk dapat membuat program Visual Basic. Anda dapat menuliskan program Visual Basic pada sebuah notepad biasa karena kemampuan untuk menjalankan program terletak pada **framework .Net**yang diinstal saat pertama kali anda menginstal Visual Studio. Namun demikian, alangkah sengsaranya apabila kita mengembangkan sebuah aplikasi besar

menggunakan notepad.Oleh karena itu, Microsoft melengkapi produk nya dengan Integrated Depelovement Environment (IDE) yang canggih.

Lingkungan pemrograman atau disebut Integrated Development Environment (IDE) adalah tempat yang akan Anda gunakan untuk menghabiskan waktu dalam mengembangkan sebuah aplikasi Visual Basic dimana IDE ini telah dirancangf sedemikian canggih oleh Microsoft untuk membantu para pengembang aplikasi dalam pengembangan aplikasi agar semakin cepat dan otomatis. Lebih dari itu , Anda akan mampu membuat sebuah aplikasi Windows yang tampak profesional hanya dengan memasukkan sedikit kode. Untuk memahami lingkungan IDE Visual Basic, kita dapat mengaktifkan program Visual Basic yang telah diinstal sebelumnya.



Gambar II.8 : Start page Visual Basic 2008

Adapun jendela-jendela yang ada pada Visual Basic adalah sebagai berikut :

1. *Menu Bar* digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project dan lain-lain.

2. *Main Toolbar* digunakan untuk melakukan tugas-tugas tertentu.
3. Jendela *Project* ini berisi gambaran dari semua modul yang terdapat dalam aplikasi. Kita dapat menggunakan *icon Toggle Folder* untuk menampilkan modul-modul dalam jendela tersebut secara group atau berurut berdasarkan nama.
4. Jendela *form Designer* ini merupakan tempat untuk merancang *user interface* dari aplikasi.
5. Jendela *Toolbox* in berisi komponen-komponen yang dapat digunakan untuk mengembangkan *user interface*.
6. Jendela *Code* merupakan tempat untuk menulis kode program.
7. Jendela *Properties* merupakan daftar properti-properti object yang sedang terpilih.
8. Jendela *Color Palette* adalah fasilitas cepat untuk mengubah warna suatu object.
9. Jendela *Form Layout* akan menunjukkan bagaimana form bersangkutan ditampilkan ketika runtime.

IDE Visual Basic 2008 memiliki beberapa komponen yang akan membantu proses pengembangan program, yaitu:

1. **Toolbox**

Jendela kecil yang biasanya terletak disebelah kiri, berisi semua control yang akan digunakan dalam pengembangan aplikasi berbasis Windows form, seperti yang sedang kita buat saat ini. Toolbox memiliki beberapa kelompok kontrol yang disusun berdasarkan kegunaannya.

2. **Jendela Design**

Jendela ini terletak di tengah IDE dan menjadi pusat kegiatan perancangan aplikasi karena terdapat Windows form yang akan digunakan untuk menampung semua

control yang akan digunakan dalam pengembangan program; file Visual Basic 2008 berekstensi **.vb**.

3. **Jendela Solution Explorer**

Jendela Solution Explorer berfungsi untuk menampilkan semua file yang terdapat dalam proyek. Beberapa file yang ditampilkan pada jendela Solution Explorer adalah:

- Form dan semua kode program di dalamnya, berekstensi **.vb**.
- **Class** adalah file yang menampung semua *class* dalam apilkasi, berekstensi **.vb**.
- **Module** adalah file yang menampung kode program kecil yang digunakan dalam program aplikasi, berekstensi **.vb**.
- **Class Diagram,** berekstensi **.cd**.
- **XML,** berekstensi **.xml**.
- **Project File,** berekstensi **.vbproj.**
- File laporan dari Crystal Report, berekstensi **.rpt**.

Pada jendela Solution Explorer tersebut, Anda dapat menambahkan item baru pada proyek, menghapusnya, mengaktifkannya, serta menyuntingnya/editing.Anda juga dapat mengubah tampilan jendela Solution Explorer dengan mode Design View (untuk membuat antarmuka aplikasi) atau ke jendela kode untuk menambahkan kode ataupun melakukan editing.

4. **Jendela Data Source**

Pada bagian bawah jendela Solution Explorer terdapat tab **Data Source** yang berfungsi sebagai manajemen basisdata apabila aplikasi Anda menggunakan basisdata. Semua kegiatan basisdata dapat dilakukan pada jendela ini, mulai dari membuatkoneksi ke DBMS, membuat database, membuat table, dan lain-lain.

5. **Jendela Properties**

Jendela properties adalah jendela yang menampung emua property dari objek-objek aplikasi, seperti: *form, button, radio button, check box, text box*, dan lain-lain. Anda dapat mengganti nilai property setiap objek agar dapat ditampilkan sesuai dengan setting yang diharapkan, misalnya Properti Name atau Text dari masing-masing objek.Sebelumnya, Anda harus mengklik objek yang ingin dilihat propertinya agar semua property objek ditampilkan pada jendela ini.Namun demikian, Anda juga dapat mengganti property sebuah objek manggunakan kode pemrograman sewaktu program berjalan.

II.5. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database.SQL Server adalah sebuah DBMS (Database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle.SQL Server 2008 dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat.Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

SQL Server Management Studio membantu Anda mangatur database dengan mudah.Anda dapat melakukan pengaturan atas beberapa server pada sebuah komputer saja atau melakukan pengaturan server secara remote.Anda dapat juga membuat database, table, index, dan melakukan manipulasi data terhadap database dan table-tabelnya.

SQL Server Management Studio memiliki beberapa komponen penting yang mewakili kegunaannya dalam perancangan database, dan melakukan pengaturan sistem secara keseluruhan. Komponen-komponen tersebut adalah:

- Registered Server
- Object Explorer
- Query Editor

Pada saat membuka SQL Server Management Studio, Anda akan mendapatkan jendela kecil yang berjudul **Connect to Server** seperti tampak pada gambar.

Gambar II.9 : Jendela Connect to Server

Pada jendela tersebut terdapat beberapa komponen yang dibutuhkan untuk melakukan koneksi ke sebuah server database. Komponen tersebut adalah:

- Tipe dari server (Server Type)
- Nama server (Server Name)
- Tipe autentifikasi (Authentication)
- User name
- Password