

BAB II

TINJAUAN PUSTAKA

II.1. Definisi Sistem

Teknologi informasi memainkan peranan penting dan makin luas dalam bisnis. Teknologi informasi dapat membantu segala jenis bisnis meningkatkan efisiensi dan efektivitas proses bisnis mereka, pengambilan keputusan manajerial, dan kerja sama kelompok kerja, hingga dapat memperkuat posisi kompetitif mereka dalam pasar yang cepat sekali berubah. Hal ini berlaku ketika teknologi informasi digunakan untuk mendukung tim pengembangan produk, proses dukungan untuk pelanggan, transaksi *e-commerce*, atau dalam aktivitas bisnis lainnya. Teknologi dan sistem informasi berbasis Internet dalam waktu singkat, menjadi bahan yang dibutuhkan untuk keberhasilan bisnis di lingkungan global yang dinamis saat ini.

Menurut Jogiyanto (2009 : 34), Sistem dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu.

Menurut Bonnie Soeherman dan Marion Pinontoan (2008 : 3), Sistem dapat diartikan sebagai serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu.

Dari definisi sistem di atas, dapat disimpulkan sistem adalah suatu jaringan kerja yang saling memiliki keterkaitan antar bagian dan prosedur-prosedur yang ada.

II.2. Informasi

Menurut Jogiyanto (2009 : 36), Informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya.

Menurut Bonnie Soeherman dan Marion Pinontoan (2008 : 4), Informasi merupakan hasil pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan keputusan.

Menurut Edhy Sutanta (2011 : 13), Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang. Untuk memperoleh informasi, diperlukan data yang akan diolah dan unit pengolah. Informasi diperlukan oleh pemakai (manajemen) pada seluruh level manajemen dalam seluruh fungsi *organisatoris*. Informasi tersebut dapat mempunyai beberapa fungsi, antara lain :

1. Menambah Pengetahuan

Adanya informasi akan menambah pengetahuan bagi penerimanya yang dapat digunakan sebagai bahan pertimbangan yang mendukung proses pengambilan keputusan.

2. Mengurangi Ketidakpastian

Adanya informasi akan mengurangi ketidakpastian karena apa yang akan terjadi dapat diantisipasi dengan baik sehingga menghindari keraguan pada saat pengambilan keputusan.

3. Mengurangi Risiko Kegagalan

Adanya informasi akan mengurangi risiko kegagalan karena apa yang akan terjadi dapat diantisipasi dengan baik sehingga kemungkinan terjadinya kegagalan akan dapat dikurangi dengan pengambilan keputusan yang tepat.

4. Mengurangi Keanekaragaman Yang Tidak Diperlukan

Adanya informasi akan mengurangi keanekaragaman yang tidak diperlukan karena keputusan yang diambil lebih terarah.

5. Memberi standar, aturan, ukuran, dan keputusan yang menentukan pencapaian sasaran dan tujuan.

II.3. Sistem Informasi

Menurut Jogiyanto (2009 : 33), Sistem Informasi merupakan suatu sistem yang tujuannya menghasilkan informasi. Sebagai suatu sistem, untuk dapat memahami sistem informasi, akan lebih baik jika konsep dari sistem itu dipahami terlebih dahulu. Demikian juga sebagai sistem penghasil informasi, maka konsep informasi perlu dipahami terlebih dahulu.

Menurut Bonnie Soeherman dan Marion Pinontoan (2008 : 5), Sistem informasi merupakan serangkaian komponen berupa manusia, prosedur, data, dan

teknologi (seperti komputer) yang digunakan untuk melakukan sebuah proses untuk menghasilkan informasi yang bernilai untuk pengambilan keputusan.

II.4. Basis Data

Menurut Edhy Sutanta (2011 : 29), Basis data sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol (*controlled redundancy*), data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali, data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya, data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

Menurut Janner Simarmata (2007 : 2), Suatu basis data adalah koleksi data yang bisa mencari secara menyeluruh dan secara sistematis memelihara dan me-retrieve informasi. Suatu basis data yang terkomputerisasi menawarkan keuntungan yang lebih banyak dari pada basis data yang tidak terkomputerisasi, yang mencakup:

1. Terpusat dan Berbagi Data, Memasukkan dan menyimpan semua data didalam computer. Tindakan itu mengurangi penggunaan kertas, file, folder, dan juga kemungkinan kehilangan atau kesalahan saat menempatkan data.
2. Current Data, Pengguna bisa secara cepat memperbarui data dan data yang tersedia siap untuk digunakan.

3. Kecepatan dan Produktivitas, Bisa mencari, mengurutkan, me-retrieve, membuat perubahan, dan mencetak data.
4. Akurasi dan Konsistensi, Bisa merancang basis data untuk memvalidasi masukan data, serta untuk memastikan bahwa data yang dimasukkan adalah valid dan konsisten.
5. Analisis, Basis data bisa menyimpan, menjejaki, dan memproses isi data yang besar dari sumber yang berbeda.
6. Keamanan, Bisa memproteksi basis data dengan penetapan daftar kata sandi (*password*) dan identifikasi pengguna yang diotorisasi.
7. Pemulihan, Kegagalan sistem tidak bisa diremehkan/diacuhkan. Penggunaan basis data memungkinkan integritas data mengalami kegagalan. Sistem manajemen basis data menggunakan suatu log transaksi untuk memastikan bahwa data akan menjadi baik/pulih saat menjalankannya kembali setelah terjadi *crash*.
8. Transaksi, Konsep transaksi menyediakan suatu mekanisme pemulihan kesalahan yang disamaratakan untuk memberikan konsekuensi atas kesalahan yang tidak terduga.

II.5. Normalisasi

Menurut Janner Simarmata (2007 : 77), Normalisasi adalah proses mengubah perancangan basis data untuk menghasilkan skema tabel dalam bentuk normal.

Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal, yaitu:

1. Bentuk Normal Pertama (1 NF)

Bentuk normal pertama sangat sederhana. Skema tabel disebut dalam bentuk normal pertama jika nilai atribut tidak terpisahkan.

2. Ketergantungan Fungsional

Berarti bahwa atribut di sisi kiri sepenuhnya menentukan atribut di sisi kanan dan penambahan data bisa ditambahkan dalam basis data. Sama halnya dengan konsep dari keterkaitan kunci, setiap skema tabel memiliki sekumpulan ketergantungan fungsional yang berhubungan di mana dasarnya adalah pada pengertian atribut.

3. Bentuk Normal Kedua (2 NF)

Berdasarkan skema, tabel T memiliki bentuk normal kedua jika semua atribut informasi (atribut yang tidak memiliki kunci manapun) adalah atribut dari entitas lain di dalam skema tabel dan bukan dari kelas entitas lainnya.

4. Bentuk Normal Ketiga (3 NF)

Jika skema tabel berbentuk normal kedua, maka bukan informasinya yang kuat, atribut tergantung pada *subset* yang sesuai dengan kunci.

5. Bentuk Normal Boyce Codd

BCNF merupakan bentuk normal sebagai perbaikan terhadap 3NF. Suatu relasi BCNF selalu memenuhi 3NF, tetapi tidak sebaliknya. Suatu relasi yang memenuhi 3NF belum tentu memenuhi BCNF. BCNF adalah perbaikan dari 3NF karena bentuk normal ketiga pun masih

memungkinkan mengandung anomali sehingga masih perlu dinormalisasikan lebih lanjut.

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

Menurut Munawar (2005 : 17), UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Karena ini merupakan sebuah bahasa, UML mempunyai sejumlah aturan untuk menggabungkan/mengkombinasikan elemen-elemen tersebut. Untuk lebih jelasnya akan lebih baik bila kita melihat diagram apa saja yang ada di UML, yaitu:

1. *Use Case Diagram*
2. *State Machine Diagram*
3. *Sequence Diagram*
4. *Collaboration Diagram*
5. *Activity Diagram*
6. *Component Diagram*
7. *Deployment Diagram*
8. *Package Diagram*

9. *Object Diagram*

10. *Communication Diagram*

11. *Composite Structure Diagram*

12. *Interaction Overview Diagram*

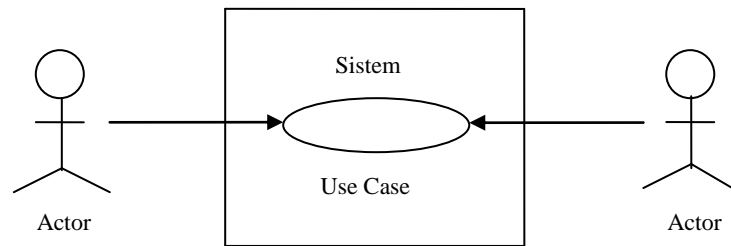
13. *Timing Diagram*

Berikut ini adalah keterangan dari masing-masing diagram pada *Unified Modelling Language (UML)*.

1. *Use Case Diagram*

Menurut Munawar (2005 : 63), *Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras, atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna. Model *use case* adalah bagian dari model requirement (Jacobson et al, 1992). Termasuk disini adalah problem domain object model dan penjelasan tentang user interface. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user.

Use Case Diagram 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*.



Gambar II.1: Contoh *Use Case Model*

(Sumber : Munawar ; 2005 : 64)

2. *State Machine Diagram*

Menurut Munawar (2005 : 73), Interaction diagram dan state chart menampilkan dua pandangan yang saling melengkapi tentang perilaku dinamis sebuah sistem. Interaction diagram menunjukkan pesan-pesan yang dilewatkan diantara obyek-obyek di dalam sistem selama periode waktu yang pendek. Sedangkan state chart diagram menelusuri individu-individu obyek melalui keseluruhan daur hidupnya, menspesifikasikan semua urutan yang mungkin dari pesan-pesan yang akan diterima obyek tersebut, bersama-sama dengan tanggapan atas pesan-pesan tersebut.

3. *Sequence Diagram*

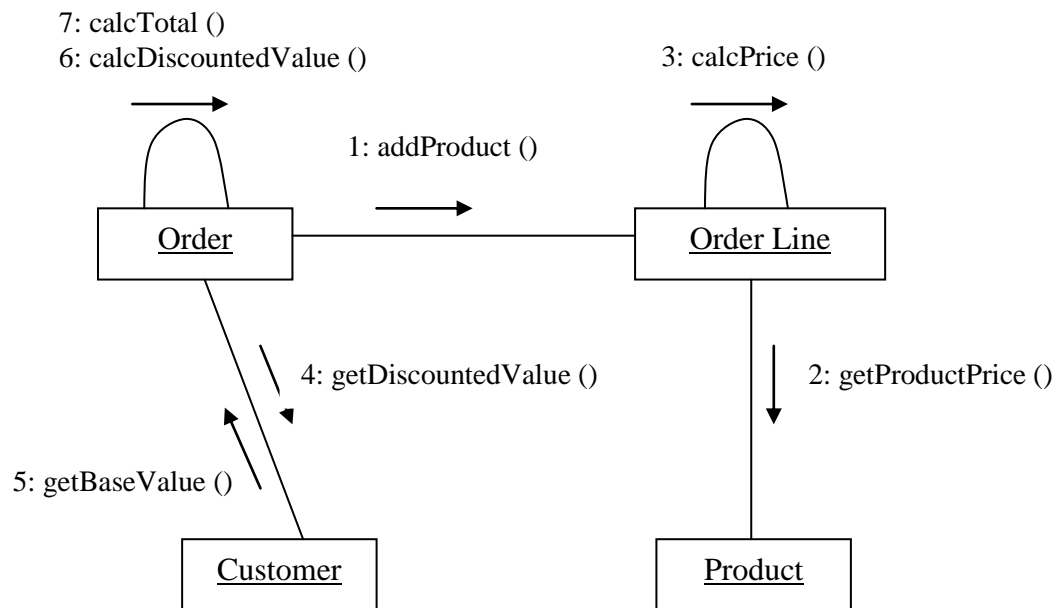
Menurut Munawar (2005 : 87), *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan

sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

4. *Collaboration Diagram*

Menurut Munawar (2005 : 101), *Collaboration Diagram* adalah perluasan dari obyek diagram. (Obyek diagram menunjukkan obyek-obyek dan hubungannya satu dengan yang lain). *Collaboration diagram* menunjukkan *message-message* obyek yang dikirimkan satu sama lain. Untuk menunjukkan sebuah pesan, buatlah tanda panah didekat garis asosiasi diantara 2 obyek. Arah panah menunjukkan obyek yang menerima pesan. Label didekat panah menunjukkan pesannya apa. Tipikal message meminta kepada obyek yang menerimanya untuk menjalankan salah satu operationnya. Sepasang tanda kurung digunakan untuk mengakhiri *message*. Jika ada parameter bisa diletakkan diantara tanda kurung.

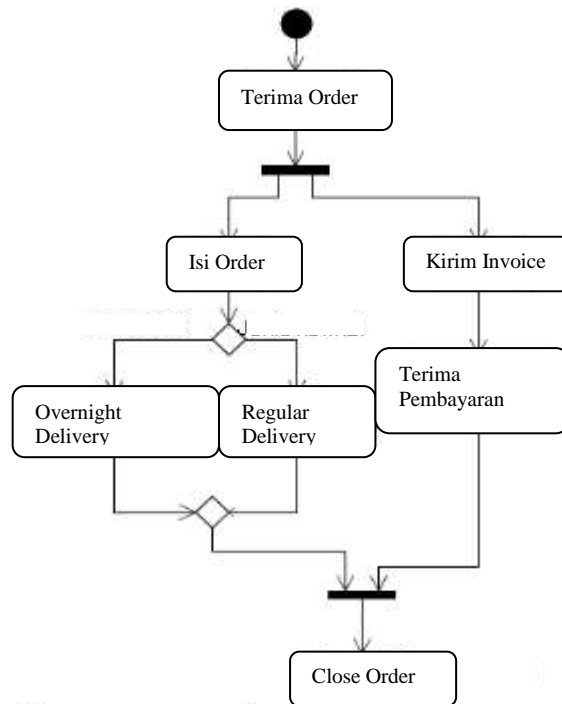


Gambar II.2 : Contoh *Collaboration Diagram* untuk order

(Sumber : Munawar ; 2005 : 102)

5. Activity Diagram

Menurut Munawar (2005 : 109), Activity Diagram adalah teknik untuk mendiskripsikan logika procedural, proses bisnis, dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah *activity diagram* bisa mendukung perilaku paralel sedangkan flowchart tidak bisa.



Gambar II.3 : Contoh *Activity Diagram* Sederhana

(Sumber : Munawar ; 2005 : 111)

6. *Component Diagram*

Menurut Munawar (2005 : 124), *Component Diagram* merepresentasikan dunia riil item yaitu component software. Component Software menetap di computer bukan dibenak para analis. Component bisa diakses melalui interfacenya yaitu koleksi operasi-operasi. Relasi antara component dan interfacenya disebut *realization*. Suatu component bisa mengakses service-service yang ada di component lain dengan cara import interface. Sedangkan component yang menyediakan service menggunakan *export interface*.

7. *Deployment Diagram*

Menurut Munawar (2005 : 125), *Deployment Diagram* menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware. Bagian utama hardware/perangkat keras adalah node, yaitu nama umum untuk semua jenis sumber komputasi. Ada 2 tipe node yang mungkin. *Processor* adalah node yang bisa mengeksekusi sebuah component, sedangkan device tidak. *Device* adalah perangkat keras (seperti printer atau monitor) tipikalnya menjadi interface dengan dunia luar.

8. *Package Diagram*

Menurut Munawar (2005 : 134), *Package Diagram* adalah pengelompokkan konstruksi ke level yang lebih tinggi. Sebuah package bisa menjadi anggota package yang lain. Bila dibuat hierarki, package yang paling tinggi akan mengandung package-package yang lain. Demikian seterusnya hingga yang paling bawah adalah class.

Manfaat utama menggunakan package adalah penerapannya pada sistem skala besar untuk mendapatkan gambaran saling ketergantungan diantara component-component utama pada sistem.

9. *Object Diagram*

Menurut Munawar (2005 : 135), *Object Diagram* adalah gambaran obyek-obyek secara ringkas disebuah sistem pada suatu waktu. *Object Diagram* sering disebut sebagai instance diagram karena menunjukkan

instance-instance dari class. Obyek diagram bisa digunakan untuk menunjukkan contoh konfigurasi dari obyek-obyek. Hal ini sangat berguna untuk menunjukkan hubungan yang mungkin ada diantara obyek-obyek yang sangat kompleks.

Object Diagram bisa digunakan untuk memodelkan pandangan dari rancangan atau proses yang statis dari suatu sistem. Termasuk disini adalah pemodelan sistem secara ringkas pada suatu waktu dan membuat serangkaian obyek, state, dan relasinya.

10. *Communication Diagram*

Menurut Munawar (2005 : 141), *Communication Diagram* adalah sejenis dengan diagram interaksi, yang lebih menekankan pada link data diantara bermacam-macam participant pada interaksi tersebut. Pada *Communication Diagram* kita bebas menempatkan participant, boleh menggambarkan link untuk menunjukkan bagaimana participant berhubungan serta boleh juga memberikan penomoran untuk menunjukkan urutan *message*.

11. *Composite Structure Diagram*

Menurut Munawar (2005 : 148), *Composite Structure Diagram* adalah diagram untuk menunjukkan dekomposisi secara hirarkis sebuah class ke sebuah struktur internal. Hal ini memungkinkan untuk memecah obyek yang kompleks menjadi bagian-bagian kecil. Perbedaan antara package dengan *composite structure* adalah package digunakan untuk pengelompokkan saat

kompilasi (*compiletime*), sedangkan *composite structure* digunakan untuk pengelompokan saat dijalankan (*runtime*).

12. *Interaction Overview Diagram*

Menurut Munawar (2005 : 149), *Interaction Overview Diagram* adalah pencangkakan secara bersama antara activity diagram dengan sequence diagram. *Interaction Overview Diagram* bisa dianggap sebagai activity diagram dimana semua aktivitas diganti dengan sedikit sequence diagram, atau bisa juga dianggap sebagai sequence diagram yang dirincikan dengan notasi activity diagram yang digunakan untuk menunjukkan aliran pengawasan. Dengan kata lain interaction diagram adalah gabungan diantara keduanya yang membuat campuran sedikit ganjil.

13. *Timing Diagram*

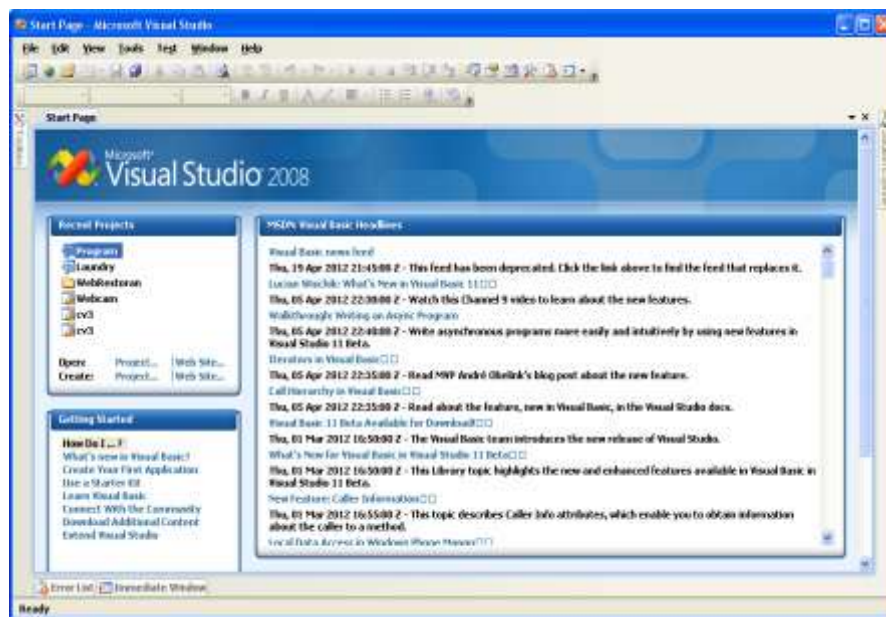
Menurut Munawar (2005 :151), *Timing Diagram* adalah bentuk lain dari interaction diagram, dimana focus utamanya lebih ke waktu. *Timing diagram* ini bisa dipakai untuk obyek tunggal maupun sekelompok obyek.

II.7. *Microsoft Visual Basic.Net 2008*

Menurut Muhammad Sadeli (2009 : 2), *Visual Studio 2008* merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi *desktop* (*windows form, command line (console)*), aplikasi *web*, *windows mobile (pocket pc)*.

Untuk membuka program *Visual Studio 2008* dapat dilakukan dengan langkah-langkah di bawah ini:

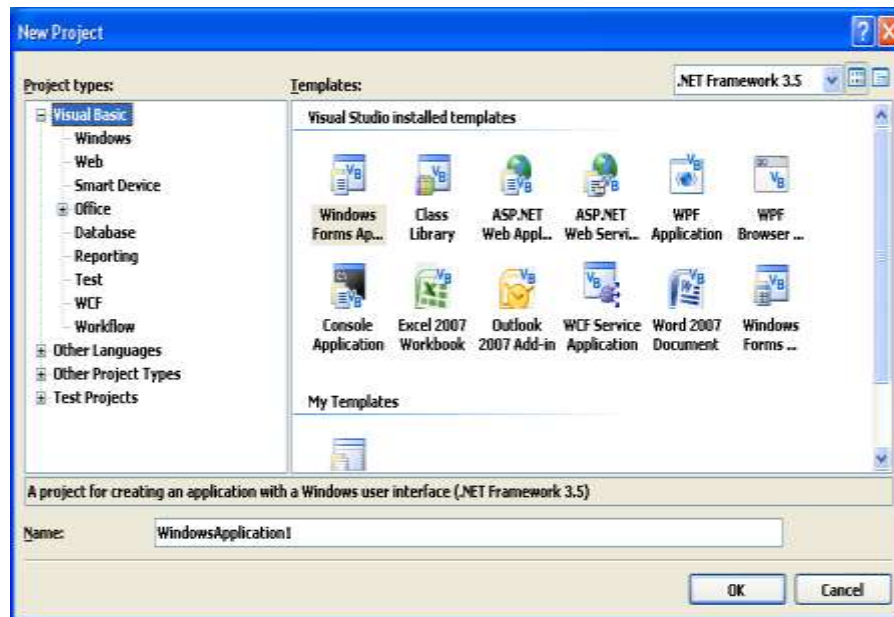
1. Klik tombol *start* pada *windows taskbar*.
2. Klik menu *All Program > Microsoft Visual Studio 2008 > Microsoft Visual Studio 2008*.



Gambar II.4 : Tampilan Microsoft Visual Basic 2008

(Sumber : Muhammad Sadeli ; 2009 : 4)

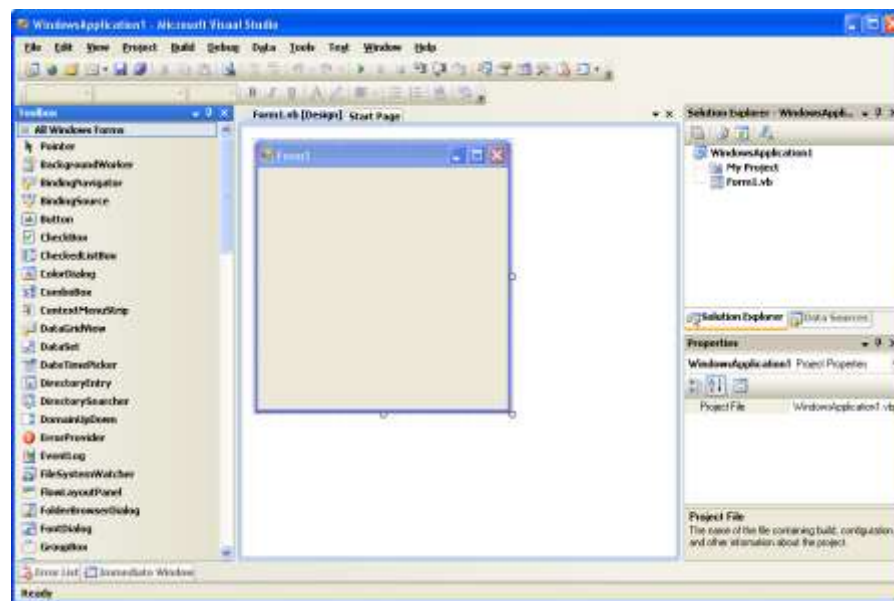
3. Untuk membuat project baru klik menu *Creat Project* atau Anda klik menu *file > New Project* seperti pada gambar berikut ini :



Gambar II.5: Jendela New Project Visual Basic 2008

(Sumber : Muhammad Sadeli ; 2009 : 5)

Pada *Microsoft Visual Basic 2008* terdapat banyak tipe project yang bisa dibuat antara lain *Windows (Windows Form, Command line)*, *Web (ASP.Net 3.5)*, *Smart Device (Pocet PC)*, *Office (2003, 2007)*, *Database (Access, SQL Server)*, *Report (Crystal Report)*, *Test*, *WCF*, dan *Workflow*. Pilih *Windows (Windows Form Application)* seperti gambar di bawah ini.



Gambar II.6 : *Integrated Development Environment (IDE)*

(Sumber : Muhammad Sadeli ; 2009 : 7)

II.8. *SQL Server*

Menurut Ema Utami dan Sukrisno (2008 : 1), Menyatakan *SQL (Structured Query Language)* pada dasarnya adalah bahasa komputer standar yang ditetapkan untuk mengakses dan memanipulasi sistem database. Sebuah *database* terdiri dari satu tabel atau lebih dan memiliki nama yang berbeda untuk masing-masing tabel. Masing-masing tabel memiliki satu kolom (*field*) atau lebih dan memiliki baris (*record*). *Query* digunakan untuk mengakses dan mengolah database. *SQL* terdiri dari 5 bagian utama, yaitu:

1. *Retrieving data*: perintah untuk menampilkan data dari database (SELECT).
2. *Data Definition Language (DDL)*: bahasa yang digunakan untuk membuat dan menghapus tabel atau database itu sendiri (CREATE, DROP, ALTER)

