

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Perancangan

Perancangan didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Langkah-langkah yang dilakukan pada tahap perancangan sistem sebagai berikut :

1. Menyiapkan rancangan sistem yang terinci.
2. Mengidentifikasi berbagai alternatif konfigurasi sistem.
3. Mengevaluasi berbagai alternatif konfigurasi sistem.
4. Memilih konfigurasi yang terbaik.
5. Menyiapkan usulan penerapan.
6. Menyetujui atau menolak penerapan sistem.

Beberapa tujuan dari perancangan sistem menurut Tata Sutabri, S.Kom, MM (2003: 88) sebagai berikut :

1. Melakukan evaluasi serta merumuskan pelayanan sistem yang baru secara rinci dan menyeluruh dari informasi yang dihasilkan.
2. Mempelajari dan mengumpulkan data untuk disusun menjadi sebuah struktur data yang teratur sesuai dengan sistem yang akan dirancang.
3. Penyusunan perangkat lunak sistem yang berfungsi sebagai sarana pengolahan data dan penyaji informasi yang dibutuhkan.

4. Menyusun kriteria tampilan informasi yang akan dihasilkan secara keseluruhan.
5. Penyusunan buku pedoman (manual) tentang pengoperasian perangkat lunak sistem yang akan dirancang.

II.2. Konsep Sistem Informasi

Sistem dapat didefinisikan sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Hal pertama yang perlu diperhatikan dalam suatu sistem adalah elemen-elemennya yang kombinasi setiap sistemnya berbeda. Susunan suatu sistem pada dasarnya terdiri atas unit *input*, unit pengolahan dan unit *output*. *Input* atau masukan masuk ke dalam sistem melalui unit input. Selanjutnya, *input* diproses oleh unit pemrosesan dan hasilnya ditampilkan ataupun dicetak keluar melalui unit *output*. Sistem dapat dibedakan sebagai sistem terbuka dan sistem tertutup. Sistem dikatakan terbuka jika terjadi arus sumber daya antara sistem dan lingkungannya. Jika tidak ada interaksi dengan lingkungannya, sistem disebut sistem tertutup (Eko Nugroho, 2008:17).

Informasi adalah suatu pengetahuan yang berguna untuk pengambilan keputusan. Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Informasi yang disampaikan kepada pemakai mungkin merupakan hasil dari data yang dimasukkan ke dalam pengolahan. Akan tetapi, dalam pengambilan keputusan yang kompleks, informasi hanya dapat menambah kemungkinan kepastian atau mengurangi bermacam-macam pilihan. Informasi dapat dikelompokkan menjadi 3 bagian

yaitu informasi strategis, informasi taktis dan informasi teknis. Informasi strategis yaitu informasi yang digunakan untuk mengambil keputusan jangka panjang, mencakup informasi eksternal, rencana perluasan perusahaan dan sebagainya. Informasi taktis adalah informasi yang dibutuhkan untuk mengambil keputusan jangka menengah seperti informasi *trend* penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan. Informasi teknis adalah informasi yang dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan barang, retur penjualan dan laporan kas harian (Tata Sutabri, 2003:17).

Sistem informasi merupakan sistem konseptual yang memakai sumber daya konseptual, data dan informasi untuk mewakili sistem fisik yang dalam hal ini berupa perusahaan atau organisasi. Komputer merupakan suatu sistem fisik, tetapi data dan informasi yang tersimpan di dalamnya dapat dipandang sebagai suatu sistem konseptual. Sistem fisik penting karena keberadaannya, sedangkan sistem konseptual penting karena penggambaran atas sistem fisik. Oleh karena itu, sistem informasi membantu para manajer dan pimpinan perusahaan untuk mendapatkan gambaran mengenai perusahaan. Informasi yang didapatkan merupakan bahan masukan penting bagi manajer dalam pengambilan keputusan (Eko Nugroho, 2008:18).

Sistem informasi adalah sekelompok elemen yang saling bekerja sama dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi organisasi untuk dapat menyediakan kepada pihak luar tertentu informasi yang diperlukan dalam pengambilan keputusan untuk mencapai suatu tujuan.

II.3. Konsep Pemasaran

Pemasaran adalah serangkaian proses untuk menciptakan, mengkomunikasikan dan memberi nilai kepada pelanggan. Sistem informasi pemasaran adalah sistem yang memberikan dukungan penyediaan informasi untuk pemasaran (Eko Nugroho, 2008:136). Ada lima kelompok informasi penting dalam bidang pemasaran sebagai berikut :

1. Informasi produk yaitu informasi mengenai produk yang dijual dan produk kompetitor.
2. Informasi promosi yaitu informasi mengenai cara, media dan waktu promosi yang dilakukan.
3. Informasi harga yaitu informasi mengenai harga produk termasuk harga dari kompetitor.
4. Informasi tempat yaitu informasi mengenai wilayah dan strategi tempat pemasaran produk.
5. Informasi gabungan yaitu informasi mengenai strategi terintegrasi yang merupakan kesatuan dari produk, promosi, harga dan tempat.

II.4. Konsep Material

Material merupakan bahan yg akan dipakai untuk membuat barang lain atau dapat diartikan sebagai bahan mentah untuk bangunan seperti pasir, kayu, kapur. Ada beberapa cirri-ciri yang dimiliki oleh material yaitu fisik, mekanik, termal, listrik, magnetik, produksi, estetika dan ekonomi. Ada beberapa jenis material sebagai berikut :

1. Material menurut sifat-sifat atom penyusunnya yaitu material logam dan material non-logam.
2. Material menurut pergerakan atom penyusunnya yaitu cair, gas dan padat.
3. Material menurut sifat konduktivitasnya yaitu isolator, semikonduktor konduktor dan super konduktor.

II.5. Alat Bantu yang Digunakan dalam Perancangan Sistem



Alat-alat yang digunakan dalam perancangan sistem informasi adalah bagan alir (*flowchart*), *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*, kamus data, normalisasi, bagan terstruktur dan sebagainya.

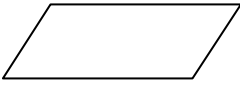
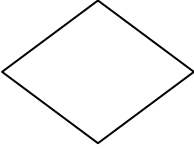
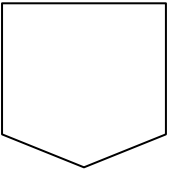
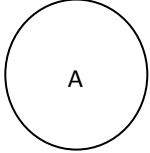
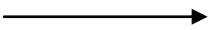

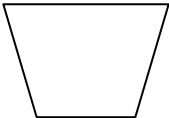
II.5.1. Diagram Alir (*Flowchart*)

Diagram alir (*flowchart*) adalah peralatan yang sesuai untuk menggambarkan proses sistem informasi secara rinci. (Dr. Ir. Eko Nugroho, M.Si, Sistem Informasi Manajemen, 2008 : 115).

Di bawah ini penulis menjelaskan arti dari lambang *flowchart* atau diagram alir dapat dilihat pada Tabel II.1.

Table II.1. Simbol Bagan Alir (*Flowchart*)

| No | Simbol | Nama | Keterangan |
|----|---|------------|---|
| 1 |  | Proses | Proses merupakan suatu simbol yang menunjukkan setiap pengolahan yang dilakukan oleh komputer |
| 2 |  | Terminator | Terminator berfungsi untuk memulai atau mengakhiri program. |

| | | | |
|---|---|----------------------------------|--|
| 3 |  | Input – output | Input - output berfungsi untuk memasukkan data atau menunjukkan hasil dari proses. |
| 4 |  | Keputusan | Keputusan merupakan suatu kondisi yang menghasilkan beberapa kemungkinan jawaban atau pilihan |
| 5 |  | Penyambung pada Halaman Berbeda | Penyambungan pada halaman berbeda merupakan simbol masuk atau keluarnya suatu prosedur pada lembar kertas lain. |
| 6 |  | Sambungan pada halaman yang sama | Penyambungan pada halaman yang sama merupakan simbol masuk atau keluarnya suatu prosedur pada lembar kertas yang sama. |
| 7 |  | Konektor | Arus / <i>flow</i> yang menghubungkan setiap bagian dalam sistem. |
| 8 |  | Dokumen | Dokumen adalah data <i>input</i> atau <i>output</i> dalam format yang akan dicetak. |
| 9 |  | Operasi manual | Operasi manual adalah kegiatan memproses data secara manual. |

Sumber (Dr.Ir. Eko Nugroho, M.Si, Sistem Informasi Manajemen, 2008 : 116)

II.5.2. Use Case Diagram

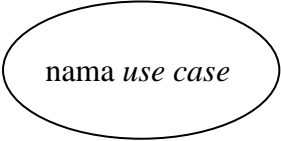
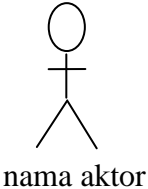

Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem

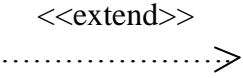
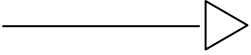
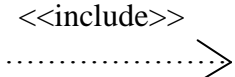
informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada dua hal utama pada *use case* yaitu :

1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat. Jadi, walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel II.2.

Table II.2. Simbol *Use Case Diagram*

| No | Simbol | Nama | Keterangan |
|----|---|-----------------|---|
| 1 |  | <i>Use Case</i> | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja. |
| 2 |  | Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, biasanya dinyatakan dengan kata benda. |
| 3 |  | Asosiasi | Komunikasi antara aktor dan <i>use case</i> yang saling berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |

| | | | |
|---|---|--------------|--|
| 4 |  | <<extend>> | Kelakuan yang hanya berjalan dibawah kondisi tertentu. |
| 5 |  | Generalisasi | Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 6 |  | <<include>> | <i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. |

Sumber (Rosa A.S dan M.Shalahuddin, Rekayasa Perangkat Lunak, 2011 : 131)


II.5.3. Class Diagram

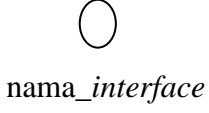

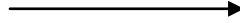
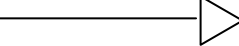
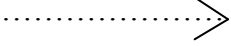

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Ada dua hal yang dimiliki oleh *class* diagram yaitu :

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Simbol-simbol yang digunakan dalam *Class* Diagram dapat dilihat pada Tabel II.3.

Table II.3. Simbol *Class* Diagram

| No | Simbol | Nama | Keterangan |
|----|---|-------|-----------------------------|
| 1 |  | Kelas | Kelas pada struktur sistem. |

| | | | |
|---|---|--------------------------------------|---|
| 2 |  | <i>Interface</i> | Tampilan antarmuka sistem. |
| 3 |  | Asosiasi | Relasi antar kelas dengan makna umum. |
| 4 |  | Asosiasi berarah | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
| 5 |  | Generalisasi | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus). |
| 6 |  | Ketergantungan/ <i>dependency</i> | Relasi antar kelas dengan makna ketergantungan antar kelas. |
| 7 |  | Agregasi | Relasi antar kelas dengan makna semua bagian. |

Sumber (Rosa A.S dan M.Shalahuddin, Rekayasa Perangkat Lunak, 2011 : 122)

II.5.4. Activity Diagram






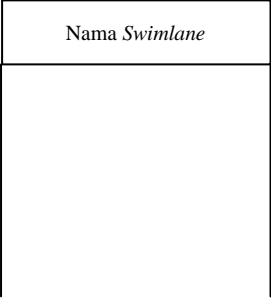
Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. *Activity* diagram banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel II.4.

Table II.4. Simbol *Activity Diagram*

| No | Simbol | Nama | Keterangan |
|----|---|------------------------------|--|
| 1 |  | Status awal | Status awal aktivitas sistem. |
| 2 |  | Aktivitas | Aktivitas yang dilakukan system, biasanya diawali dengan kata kerja. |
| 3 |  | Percabangan/ <i>decision</i> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | Penggabungan/ <i>joint</i> | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5 |  | Status akhir | Status akhir yang dilakukan sistem. |
| 6 |  | <i>Swimlane</i> | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

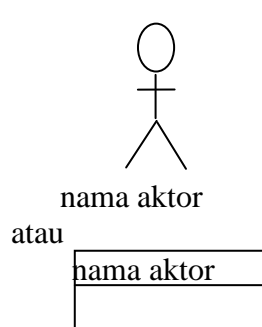

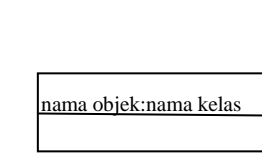
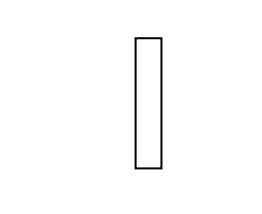
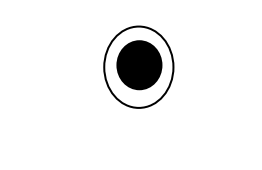
Sumber (Rosa A.S dan M.Shalahuddin, *Rekayasa Perangkat Lunak*, 2011 : 134)

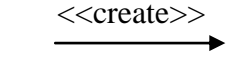
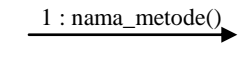
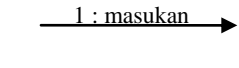
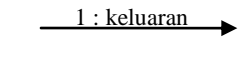
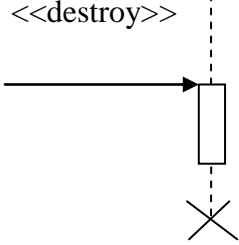
II.5.5. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antarobjek. Banyaknya diagram *sequence* yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri.

Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel II.5.

Table II.5. Simbol *Sequence Diagram*

| No | Simbol | Nama | Keterangan |
|----|--|------------------------------|--|
| 1 |  nama aktor atau nama aktor | Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat. |
| 2 |  | Garis hidup/ <i>lifeline</i> | Menyatakan kehidupan suatu objek. |
| 3 |  | Objek | Menyatakan objek yang berinteraksi pesan. |
| 4 |  | Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. |
| 5 |  | Status akhir | Status akhir yang dilakukan sistem. |

| | | | |
|----|---|---------------------------|--|
| 6 |  | Pesan tipe <i>create</i> | Objek yang lain, arah panah mengarah pada objek yang dibuat. |
| 7 |  | Pesan tipe <i>call</i> | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. |
| 8 |  | Pesan tipe <i>send</i> | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 9 |  | Pesan tipe <i>return</i> | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi/metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 10 |  | Pesan tipe <i>destroy</i> | 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> . |

Sumber (Rosa A.S dan M.Shalahuddin, Rekayasa Perangkat Lunak, 2011 : 137)

II.6. Manajemen Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Kebutuhan basis data dalam sistem informasi meliputi :

1. Memasukkan, menyimpan dan mengambil data.
2. Membuat laporan berdasarkan data yang telah disimpan.

Database Management System (DBMS) atau biasa disebut sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Suatu sistem aplikasi dapat disebut DBMS, jika memenuhi persyaratan sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan secara bebas.
4. Mampu menangani *backup* data.

Pentingnya data bagi suatu organisasi atau perusahaan, maka hampir sebagian perusahaan memanfaatkan DBMS dalam mengelola data. Pengelolaan DBMS biasanya ditangani oleh tenaga ahli yang disebut *Database Administrator* (DBA).

Beberapa keuntungan yang diperoleh dengan menerapkan DBMS sebagai berikut :

1. Mengurangi duplikasi data.
2. Memperbaiki konsistensi data. Apabila sebuah data diubah, maka semua data yang terkait akan ikut berubah secara otomatis.
3. Memperbaiki keseragaman data.
4. Mengembangkan integrasi data. Semua file disatukan menjadi sebuah sistem yang terintegrasi, saling berhubungan satu dengan yang lain.
5. Peningkatan kecepatan akses data.
6. Meningkatkan keamanan data.

II.7. Microsoft Visual Basic 2008

Visual Basic 2008 merupakan salah satu paket pemrograman dari *Visual Studio* 2008. Software ini digunakan untuk membuat aplikasi pada *Windows* seperti aplikasi *database*, *inventory*, *penjualan* dan lain-lain. *Visual Basic* 2008 merupakan aplikasi pemrograman yang menggunakan teknologi .NET Framework. Teknologi .NET Framework merupakan komponen *Windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi dan halaman web. Teknologi .NET Framework mempunyai 2 (dua) komponen utama yaitu :

1. Common Language Runtime (CLR) yang digunakan untuk menjalankan aplikasi yang berbasis .NET.
2. Library adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi.

Visual Basic 2008 dapat diinstal, apabila komputer memenuhi beberapa persyaratan yang dapat dilihat pada Tabel II.6.

Table II.6. Persyaratan *Visual Basic* 2008

| Sistem | Syarat Minimal | Syarat yang Direkomendasikan |
|----------------|--|---|
| Arsitektur | X86 dan X64 (WOW) | - |
| Sistem Operasi | Microsoft Windows XP Service Pack 2, Microsoft Windows Server 2003, Windows Vista. | - |
| Prosesor | CPU 1,6 GHz | Windows XP dan Windows Server 2003: CPU 2,2 GHz Windows Vista : CPU 2,4 GHz. |
| RAM | Windows XP dan Windows Server 2003 : 384 MB. Windows Vista : 768 MB | RAM 1024 MB/ 1 GB |

| | | |
|------------------|---|-----------------------------|
| Harddisk | Tanpa MSDN Ruang kosong harddisk 2 GB. Sisa ruang harddisk kosong 1 GB | Kecepatan harddisk 7200 RPM |
| | Dengan MSDN Ruang kosong harddisk 3,8 GB. Sisa ruang kosong 2,8 GB. | Kecepatan harddisk 5400 RPM |
| Display Layar | 1024x768 display | 1280x1024 display |

Sumber (Wahana Komputer, Membuat Aplikasi *Client-Server* dengan *Visual Basic* 2008, 2010 : 2).

Untuk memenuhi kebutuhan programmer, *Visual Basic* diedarkan dalam berbagai versi sebagai berikut :

1. *Standart Edition Learning Edition*

Versi ini adalah versi standar yang sudah mencakup berbagai sarana dasar dari *Visual Basic* 6,0 untuk mengembangkan aplikasi.

2. *Professional Edition*

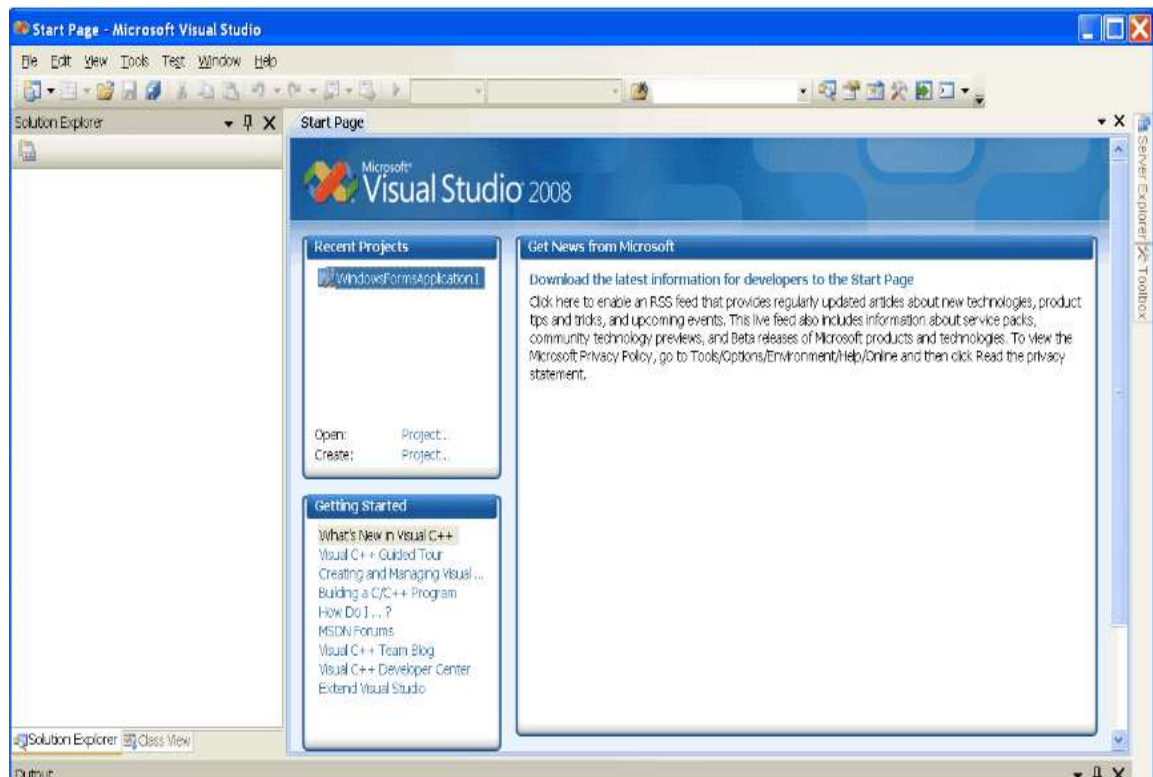
Versi ini memberikan berbagai sarana ekstra yang dibutuhkan oleh para programmer, misalnya control tambahan, dukungan programmer, internet dan lain-lain.

3. *Enterprise Edition*

Versi ini dikhususkan untuk para programmer yang ingin mengembangkan aplikasi *remote computing* atau *client-server*. Program ini digunakan untuk membuat aplikasi jaringan.

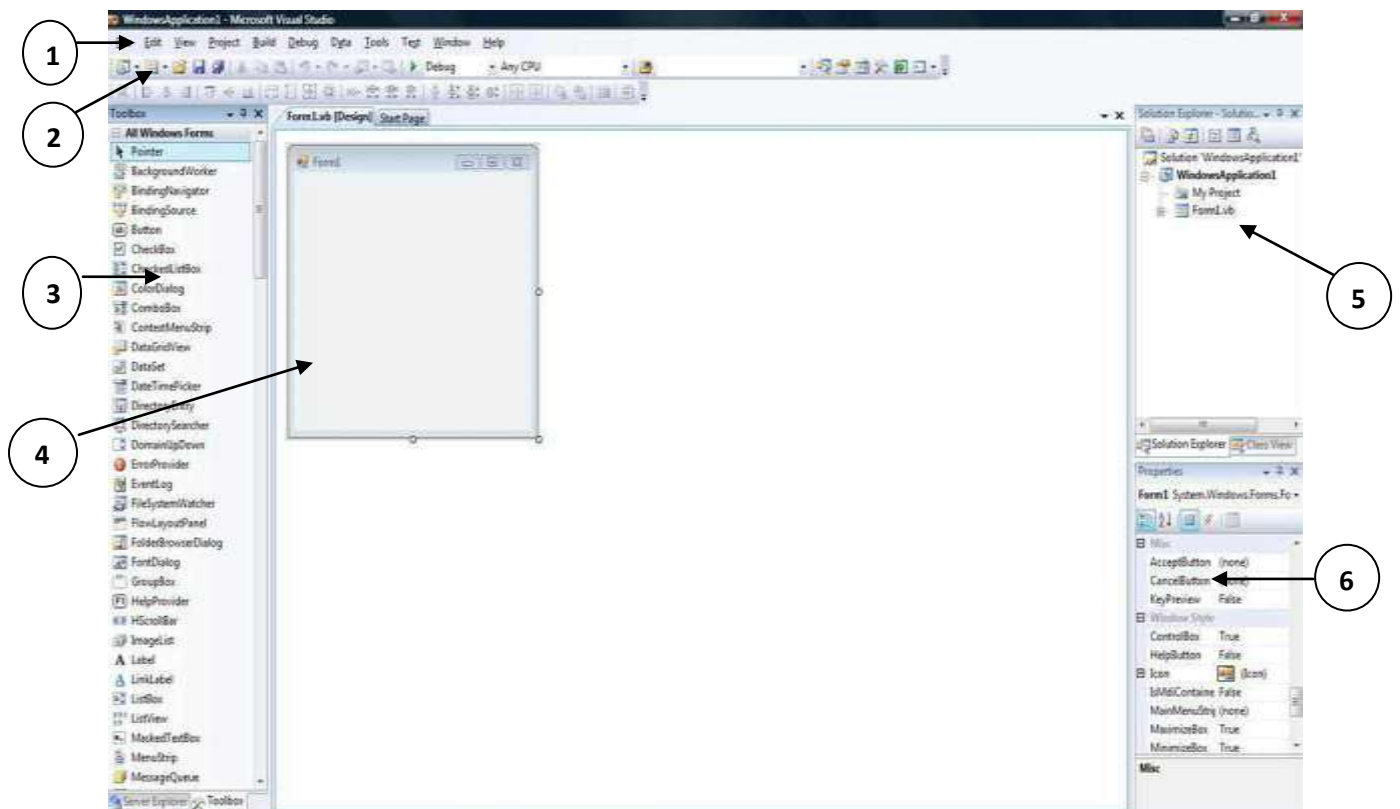
II.7.1. Mengenal Layar Kerja *Visual Basic* 2008

Saat menjalankan *Visual Basic* 2008 pertama sekali yang muncul adalah kotak dialog, dapat dilihat pada Gambar II.1.



Gambar II.1. Kotak Dialog *Visual Basic* 2008

Pada kotak dialog tersebut bisa dipilih aplikasi apa yang ingin dibuka atau ingin membuat suatu bentuk aplikasi yang baru dengan mengklik tombol *create*. Setelah *Visual Basic* dijalankan, akan muncul *form* kerja yang dapat dilihat pada Gambar II.2.



Gambar II.2. Lembar Kerja *Visual Basic 2008*

Keterangan gambar II.2. :

1. Menu

Menu adalah tempat menu standar yang ada pada *Visual Basic 2008*.

2. *Toolbar* Menu

Toolbar menu adalah baris menu yang berisi *icon-icon* pendukung pada *Visual Basic 2008*.

3. *Toolbox*

Toolbox adalah tempat yang berisi komponen-komponen yang digunakan untuk perancangan pada *Visual Basic 2008*.

4. *Form Window*

Form window merupakan lembar kerja tempat untuk merancang/ mendesain sebuah aplikasi.

5. *Solution Explorer*

Solution Explorer adalah properties yang menginformasikan tentang struktur dari satu *project* program yang sedang dibuat.

6. *Properties*

Properties adalah karakteristik sebuah komponen *visual basic* dan mungkin berbeda-beda antara satu komponen dengan komponen lainnya seperti nama komponen, *text* yang ditampilkan, ukuran, posisi dan sebagainya.

II.8. My SQL

My SQL adalah salah satu software sistem manajemen database (DBMS) *Structured Query Language* (SQL) yang bersifat *open source*. SQL adalah bahasa standar untuk mengakses database dan didefinisikan dengan standar ANSI/ISO SQL. My SQL merupakan aplikasi Relational Database Management System (RDMS) yang dapat digunakan sebagai aplikasi *client-server* atau sistem *embedded*.

My SQL memiliki sifat-sifat sebagai berikut :

1. My SQL merupakan DBMS (Database Management System).
2. Database adalah kumpulan data yang terstruktur.
3. My SQL merupakan RDBMS (Relational Database Management System).

4. Database relasional menyimpan data pada tabel-tabel yang terpisah, bukan menyimpan data dalam ruang penyimpanan yang besar. Hal ini menambah kecepatan dan fleksibilitas.
5. My SQL merupakan *software open source*.
6. *Open source* berarti setiap orang dapat menggunakan dan mengubah *software* yang bersangkutan.
7. My SQL mempunyai performa yang sangat cepat, dapat dipercaya dan mudah digunakan.
8. Software My SQL server adalah sistem *client-server* yang terdiri atas multi-threaded SQL server yang mendukung *backend* berbeda, beberapa program *client* dan pustaka (*libraries*) berbeda, peralatan administrasi dan jangkauan luas API (*Application Programming Interfaces*).

Tipe data yang ada dalam My SQL sebagai berikut :

1. Char : Data alfanumerik dengan panjang antara 1 sampai 255 karakter.
2. Date : Data termasuk juga tahun, bulan, hari, jam, menit dan detik.
3. Long : Data alfanumerik berupa teks yang disediakan sampai 2
Gigabyte.
4. Long Row : Data biner untuk menyimpan gambar, suara dan video sampai 2
Gigabyte.
5. Number : Data berupa bilangan positif, negative dan desimal.
6. Row : Data biner sampai 255 *Byte*.
7. Varchar : Data string yang menyimpan data sampai 225 *Byte*.

Hubungan antar tabel satu dengan tabel lainnya melalui *field* yang sama disebut relasi. Relasi antar tabel ini dibuat dari *Microsoft SQL Server* dengan menggunakan kunci. Ada 2 (dua) macam kunci yaitu :

1. *Primary Key*

Primary key adalah *field* unik mendefinisikan satu *record* sehingga dapat dihubungkan dengan tabel lain yang terkait.

2. *Foreign Key*

Foreign key adalah *field* yang berisi nilai indetifikasi dengan *record* yang berkaitan dalam tabel lainnya.

II.9. Client-Server

Client-server adalah salah satu model komunikasi 2 (dua) komputer atau lebih yang berfungsi melakukan pembagian tugas. *Client* bertugas untuk melakukan *input*, *update*, penghapusan dan menampilkan data sebuah *database*. *Server* bertugas menyediakan pelayanan untuk melakukan manajemen yaitu menyimpan dan mengolah *database* (Wahana Komputer, Membuat Aplikasi *Client-Server* dengan *Visual Basic* 2008, 2010 : 5).

Terdapat beberapa arsitektur yang digunakan untuk melakukan pemrograman *database* pada aplikasi berbasis *client-server* sebagai berikut :

1. Arsitektur Standalone (1-Tier)

Konsep 1 tier adalah sebuah komputer yang mengakses sebuah *database* dari komputer sendiri. Aplikasi antarmuka *user* dan aplikasi DBMS terdapat pada

satu komputer yang sama. Beberapa karakteristik arsitektur 1-tier sebagai berikut:

- a. Beban jaringan menjadi lebih tinggi karena yang diminta adalah *file database* keseluruhan pada komputer *server* dan *client* melalui jaringan.
- b. Setiap komputer pada jaringan harus mempunyai DBMS sendiri untuk menyimpan hasil salinan dari *server* sehingga dapat mengurangi pemakaian memori pada komputer *client*.
- c. Komputer *client* harus mempunyai kemampuan proses yang tinggi untuk mendapatkan waktu respon yang baik pada saat mengirim *file*.
- d. Cocok untuk bisnis kecil yang hanya membutuhkan sebuah komputer untuk memproses dan menyimpan data.

2. Arsitektur *Client-Server* (2-Tier)

Pada model ini komputer *client* bertugas menyediakan antarmuka untuk *user*, permintaan data ke DBMS *server* dan pemrosesan data. Komputer *client* hanya mengirimkan perintah untuk menambah data, menghapus, mengubah dan meminta data tertentu untuk ditampilkan melalui antarmuka yang telah dibuat. *Server* bertanggung jawab terhadap penyimpanan, pengelolaan, melayani permintaan akses data dan pemrosesan oleh *client*. Beberapa karakteristik arsitektur 2 tier sebagai berikut :

- a. 2 tier terjadi pada jaringan dan melakukan pemodelan pemrograman database dalam 2 tingkat yaitu *client* dan *server*.
- b. Komputer *client* hanya dapat mengirimkan perintah SQL untuk meminta data ke *server*.

- c. *Server* hanya memberikan data yang diminta melalui perintah *client*.
- d. Komputer *server* dituntut memiliki kemampuan pemrosesan yang tinggi.
- e. Beban jaringan menjadi lebih ringan karena data yang berjalan pada jaringan hanya data yang diminta *client*.
- f. Otentifikasi pemakai, pemeriksaan integritas dan pemeliharaan kamus data dilakukan pada sisi *server*.
- g. Sederhana dan mudah diterapkan, khususnya untuk bisnis kecil yang hanya berada pada satu gedung.

3. Arsitektur N-Tier

Arsitektur n-tier berarti membagi komponen menjadi n entitas yaitu 1 tier *client* dan n-1 tier *server*. Karakteristik model 3-tier sebagai berikut :

- a. Arsitektur ini dibagi menjadi 3 lapisan yaitu *client*, *middle* tier dan *database server*.
- b. *Client* bertugas menyediakan antarmuka bagi *user* untuk mengakses *database*.
- c. Lapisan *middle* tier menyediakan perintah untuk mengelola *database*.
- d. Lapisan *server* menyediakan ruang untuk menyimpan *database* yang dapat diakses melalui *middle* tier.
- e. Mudah untuk melakukan perubahan pada lapisan *middle* tier.
- f. Lebih mahal dibanding dengan model 2-tier.
- g. Memerlukan adaptasi yang luas apabila terjadi perubahan semua sistem.
- h. Dapat mengakses berbagai tipe DBMS berbeda dengan mudah walaupun berbeda *platform*.