

memaksimalkan total cost di PT. Sarana Samudera Pasifik Bitung, sehingga total cost tersebut dapat dikurangi dari tahun 2011.

- 4 Oleh Noer Novijanto (2010), meneliti tentang “Penentuan Jumlah Persediaan Bahan Baku Produk Tempe dengan Menggunakan Metode *Economic Order Quantity (EOQ)*”. Hasil dari penelitian ini yaitu pengolahan data yang dilakukan yaitu dengan menggunakan metode ramalan penjualan, perputasan persediaan, kebutuhan bahan baku, pembelian bahan baku yang ekonomis, dan frekuensi pengadaan.
- 5 Oleh Cahya Kurniawan (2015), judul penelitian tentang “Optimalisasi Sistem persediaan Bahan Baku Kain Menggunakan Metode *Economic Order Quantity (EOQ)*”. Hasil dari penelitian ini yaitu menghasilkan suatu perancangan *Supply Chain Manajement System* dalam pengadaan bahan baku yang dapat memudahkan dalam proses persediaannya dengan menggunakan model *Economic Order Quantity (EOQ)*.

## **2.1 Pengertian Dasar**

### **2.1.1 Pengembangan Perangkat Lunak**

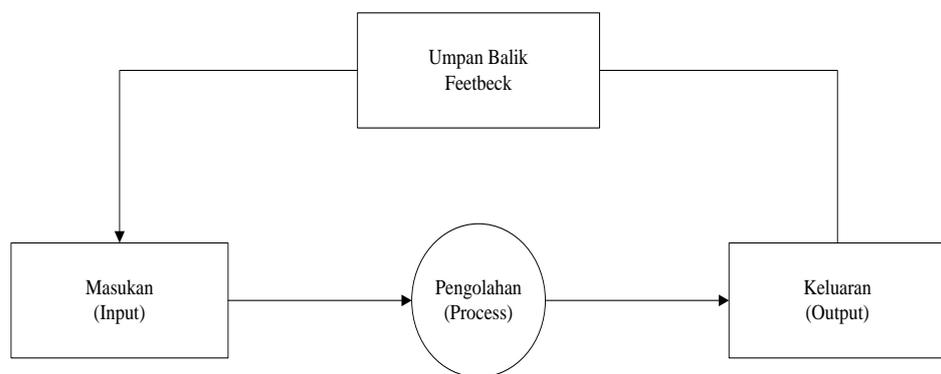
Pengembangan perangkat lunak adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan, mengubah, atau menghasilkan perangkat lunak. Hal-hal yang harus diperhatikan dalam proses pengembangan perangkat lunak yaitu sebagai berikut:

- 1) Pengetahuan lingkungan terhadap teknologi informasi dan komputer.
- 2) Pengetahuan mengenai budaya *local* di lingkungan yang dikembangkan, apakah memungkinkan dilakukan pengembangan sistem.
- 3) Pengetahuan tentang apa saja yang bisa dibatasi dan yang tidak, sehingga saat pengembangan perangkat lunak dapat mendefinisikan aturan main dari perangkat lunak. (Rosa A.S & M. Salahuddin, 2015)

### 2.1.2 Sistem

Pada umumnya definisi sistem mengandung dua konotasi yaitu benda atau entitas, dan proses atau metode. Namun secara umum definisi sistem itu sendiri adalah suatu himpunan dari berbagai bagian atau elemen yang saling berhubungan secara terorganisasi berdasarkan fungsi menjadi suatu kesatuan. (DR. Bambang Hartono, 2013)

Adapun model umum dari sistem itu sendiri dapat dilihat pada gambar berikut ini:



Gambar 2.1 Model Umum Sebuah Sistem

#### 1. Elemen Sistem

Elemen Sistem adalah suatu sistem yang terdiri dari sejumlah elemen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan. Pendekatan suatu sistem yang merupakan suatu jaringan prosedur lebih menekankan pada urutan-urutan operasi didalam sistem, sedangkan pendekatan yang menekankan pada elemen-elemen atau komponen merupakan interaksi antar elemen atau komponen atau mencapai sasaran atau tujuan sistem.

#### 2. Karakteristik Sistem

Sistem memiliki minimal 10 karakteristik yaitu sebagai berikut:

- 1) Komponen (*Components*)

Bagian-bagian atau dapat berupa benda atau manusia berbentuk nyata atau abstrak yang disebut sub sistem.

- 2) Penghubung Antar Bagian (*Interface*)

Sesuatu yang bertugas menjembatani suatu bagian dengan bagian lain dan memungkinkan terjadinya interaksi atau komunikasi antar bagian.

3) Batasan (*Boundary*)

Sesuatu yang membedakan antara suatu sistem dengan sistem atau sistem-sistem lain.

4) Lingkungan (*Environment*)

Segala sesuatu yang ada diluar sistem dan dapat bersifat menguntungkan atau merugikan sistem yang bersangkutan.

5) Masukkan (*Input*)

Sesuatu yang merupakan bahan untuk diolah atau diproses oleh sistem.

6) Keluaran Sistem (*Output*)

Berbagai macam bentuk hasil atau produk yang dikeluarkan oleh pengolahan.

7) Mekanisme Pengolahan (*Processing*)

Perangkat atau prosedur untuk mengubah masukan menjadi keluaran dan menampilkannya.

8) Tujuan(*Goal*)

Sesuatu atau keadaan yang ingin dicapai oleh sistem, baik dalam jangka panjang atau jangka pendek.

9) Sensor dan Kendali (*Sensor&Control*)

Sesuatu yang bertugas memantau dan menginformasikan perubahan-perubahan dalam lingkungan dan dalam diri sistem kepada sistem.

10) Umpan Balik (*Feedback*)

Informasi tentang perubahan-perubahan lingkungan dan perubahan-perubahan penyimpanan dalam sistem. (Bambang Hartono, (2013)

### 3. Klasifikasi Sistem

Sistem dapat diklasifikasikan berbagai sudut pandang, antara lain :

1. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical Sistem*)

Sistem Abstrak merupakan sistem yang berbentuk pemikiran / gagasan tidak tampak secara fisik, sebagai contoh: Sistem Teologia, sedangkan sistem fisik merupakan sistem yang ada secara fisik dan dapat ditangkap oleh inderawi manusia secara langsung sebagai contoh: Sistem Komputer.

2. Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Human Made System*)

Sistem Alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat manusia, sebagai contoh: Sistem Perputaran Bumi, sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dengan mesin, sebagai contoh: Sistem Informasi.

3. Sistem Tertentu (*Deterministic System*) dan Sistem Tak Tentu (*Probabilistic System*)

Sistem Tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan, sebagai contoh: Sistem Pemrograman Dalam Komputer, sedangkan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsure probabilitas.

#### 2.1.3 Informasi

Informasi pada dasarnya adalah sehimpunan data yang telah diolah menjadi sesuatu yang memiliki arti dan kegunaan lebih luas. *Lippeveld*, *Sauerborn*, dan *Bodart* mendefinisikan informasi sebagai himpunan fakta atau data yang memiliki makna.

Sedangkan data sendiri adalah hasil pengukuran atau pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan atau kejadian. Transformasi data menjadi suatu informasi dilakukan di dalam sebuah sistem, dimana data yang merupakan sebuah masukan kemudian diolah atau diproses oleh sistem menjadi suatu keluaran yang berupa informasi.

Kualitas informasi bergantung beberapa kriteria yaitu sebagai berikut:

1. Relevansi

Informasi yang bernilai tinggi adalah informasi yang dapat disajikan sesuai dengan kebutuhan.

2. Kelengkapan dan Keluasan

Informasi yang bernilai tinggi adalah informasi yang disajikan dengan lengkap tanpa dipotong-potong dan memiliki cakupan lebih luas.

3. Kebenaran

Informasi yang bernilai tinggi adalah informasi yang dihasilkan dari fakta dan bukan dari opini atau ilusi.

4. Terukur

Informasi yang bernilai tinggi adalah informasi yang jika dilacak kembali kepada datanya, informasi tersebut dapat diukur dari faktanya.

5. Keakuratan

Kecermatan dalam mengukur dan mencatat fakta akan menentukan keakuratan data dan nilai dari informasi yang dihasilkan.

6. Kejelasan

Informasi yang dihasilkan harus mudah dipahami oleh para penerimanya walaupun informasi tersebut disajikan dalam bentuk teks, grafik, tabel, chart, dan lain-lain.

7. Keluwesan

Informasi yang baik adalah informasi yang dapat diubah ubah bentuk penyajiannya sesuai kebutuhan penggunaannya.

8. Ketepatan Waktu

Informasi yang baik adalah informasi yang disajikan tepat pada waktu saat dibutuhkan. (DR. Bambang Hartono, 2013)

#### **2.1.4 Sistem Informasi**

Sistem informasi adalah seperangkat komponen yang saling berhubungan, yang bekerja untuk mengumpulkan dan menyimpan data serta mengolahnya menjadi informasi untuk digunakan. (DR. Bambang Hartono, 2013)

#### **2.1.5 Sistem Informasi Manajemen**

Sistem informasi manajemen adalah sebuah sistem, yaitu rangkaian terorganisasi dari sejumlah bagian/komponen yang secara bersama-sama berfungsi atau bergerak menghasilkan informasi untuk digunakan dalam manajemen perusahaan.

Lingkungan dari sistem informasi manajemen terdiri dari dua lapisan sebagai berikut:

##### **1. Lingkungan Internal**

Yaitu segala sesuatu yang ada di luar sistem informasi manajemen, tetapi berada di dalam perusahaan.

##### **2. Lingkungan Eksternal**

Merupakan segala sesuatu yang berada diluar perusahaan. Seperti pelanggan/konsumen, pemilik, pemerintah, dan lain-lain. (DR. Bambang Hartono, 2013)

#### **2.1.6 Metode *Economic Order Of Quantity* (EOQ)**

Pengendalian Persediaan dengan Metode *Economic Order Quantity* (EOQ) berdasarkan "Heizer, Jay dan Barry Render. 2011. *Operation Management*" merupakan sumber daya yang harus disimpan oleh organisasi dalam antisipasinya terhadap pemenuhan permintaan, sumber daya yang dimaksud ini dapat berupa Material (bahan), Mesin, Uang maupun Tenaga kerja. Dua keputusan utama yang berkaitan dengan pengendalian persediaan tersebut adalah berapa banyak sumber daya yang harus dipesan (dibeli atau diproduksi)

dan kapan waktunya untuk melakukan pemesanan (pembelian atau produksi) untuk mengurangi biaya-biaya persediaan tersebut. (Budi, 2016)

Terdapat dua biaya yang harus dipertimbangkan pada saat melakukan keputusan terhadap “jumlah yang harus dipesan”, yaitu biaya penyimpanan (*carrying cost/holding cost*) dan biaya pemesanan (*ordering cost/acquisition cost*). Jika jumlah kuantitas yang dipesan meningkat maka biaya penyimpanan akan meningkat sedangkan biaya pemesanan akan menurun. Oleh karena itu, diperlukan suatu perhitungan yang berfungsi untuk menyeimbangkan kedua biaya tersebut. Salah satu metode yang paling sering digunakan dalam menentukan jumlah kuantitas pesanan pada Manajemen Persediaan adalah Metode Economic Order Quantity (EOQ) atau dalam bahasa Indonesia disebut juga dengan Jumlah Pemesanan Ekonomis.

$$EOQ = \frac{2 \cdot P \cdot R}{U}$$

R = Penggunaan yang diperkirakan per periode

P = Biaya Pemesanan (Persiapan pesanan / Penyimpanan mesin) per pesanan

SS = Safety stock

U = Harga per bahan baku

LT = Lead time

PSLT = Penggunaan selama leadtime

Model EOQ ini dapat diterapkan apabila anggapan-anggapan berikut ini dipenuhi :

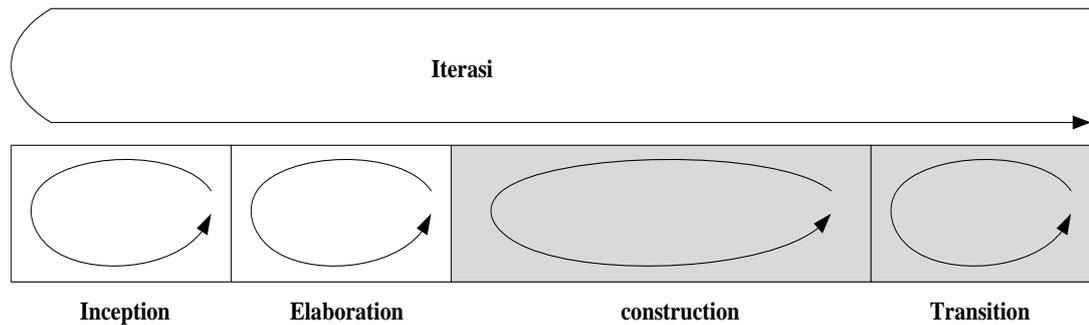
1. Permintaan akan produk adalah konstan, seragam dan diketahui (deterministik).
2. Harga per unit produk adalah konstan.
3. Biaya Pemesanan per pesanan (p) adalah konstan.
4. Waktu antara pesanan dilakukan dan barang-barang diterima (*Lead Time*) adalah Konstan.
5. Tidak terjadi kekurangan barang atau “*Back Orders*”.

## 2.2 Metodologi Pengembangan Sistem

Untuk mengembangkan suatu sistem informasi, kebanyakan perusahaan menggunakan metodologi yang sering disebut metodologi pengembangan sistem. Metodologi itu sendiri adalah suatu proses standar yang diikuti organisasi untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan melihat sistem informasi. (Abdul Kadir; 2014)

Seperti yang dilakukan pada kebanyakan proses, pengembangan sistem juga memiliki daur hidup yang sering disebut dengan Daur Hidup Pengembangan Sistem Informasi yang dalam hal ini adalah pengembangan sistem informasi Analisis Desain Berorientasi Objek (ADBO). Metodologi Berorientasi Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Alat yang digunakan dalam metodologi ini meliputi *use case*, *activity diagram*, *sequence diagram*, dan *class diagram*. (Rosa A.S & M.Shalahudin; 2013).

Di dalam metodologi pengembangan sistem berorientasi objek, terdapat metode tahapan dalam penyelesaian proses pengembangan sistem. Adapun metode yang akan digunakan dalam penelitian ini adalah metode *Rational Unified Process* (RUP). Metode RUP merupakan pendekatan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur, lebih diarahkan pada penggunaan kasus. RUP merupakan proses rekayasa perangkat lunak dengan mendefinisikan struktur yang baik untuk alur hidup proyek perangkat lunak. (Rosa As & M.Salahuddin; 2015)*Rational Unified Process* (RUP) memiliki 4 fase atau tahapan yang dapat dilakukan secara interaktif. Berikut ini gambar alur hidup RUP:



Gambar 2.2 Alur Hidup *Rational Unified Process* (RUP)

Berikut ini penjelasan untuk setiap fase pada RUP:

1. *Inception* (Permulaan)

Tahapan ini lebih pada pemodelan proses bisnis yang dibutuhkan dan mendefinisikan kebutuhan akan sistem yang akan dibuat seperti ruang lingkup dari proyek, dan membangun kasus bisnis yang dibutuhkan. Kriteria dari hasil tahapan ini adalah berikut:

- a. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, dan perkiraan jadwal.
- b. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, resiko dan proses pengembangan.
- c. Ruang lingkup prototype yang akan dikembangkan.
- d. Membangun garis dasar dengan membandingkan perencanaan actual dengan perencanaan yang direncanakan

2. *Elaboration* (Perluasan/Perencanaan)

Tahapan ini lebih fokus pada perencanaan arsitektur sistem. Perencanaan ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Tahapan ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada prototype sistem. Hasil yang diharapkan dari tahapan ini sebagai berikut:

- a. Model kasus yang digunakan (use-case) dimana kasus dan actor yang terlibat telah diidentifikasi dan sebagian besar kasus harus dikembangkan.
- b. Deskripsi dari arsitektur perangkat lunak dari proses pengembangan sistem yang telah dibuat.

- c. Rancangan arsitektur yang dibuat diimplementasikan dan mengimplementasikan *use-case*.
- d. Rencana pengembangan untuk seluruh proyek yang telah dibuat.
- e. Purwarupa yang dapat didemonstrasikan untuk mengurangi setiap resiko teknis yang diidentifikasi.

### 3. *Construction* (Konstruksi)

Tahapan ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahapan ini fokus pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program.

### 4. *Transition* (Transisi)

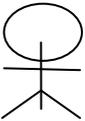
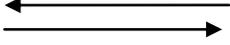
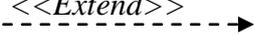
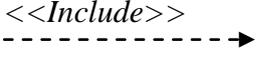
Tahapan ini lebih fokus pada *deployment* atau instalasi perangkat lunak agar dapat dipahami oleh user. Aktifitas ini lebih pada pelatihan pemeliharaan, dan pengujian sistem apakah sistem tersebut sudah sesuai harapan user itu sendiri.

Agar dapat melakukan langkah-langkah sesuai dengan metodologi pengembangan analisis dan desain berorientasi objek, maka dibutuhkan alat dan teknik untuk melaksanakannya. Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar diagram atau grafik. Adapun alat pengembangan sistem yang digunakan yaitu :

#### a. *Use Case Diagram*

Use case atau diagram use case merupakan pemodelan untuk melakukan kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah iteraksi antara satu atau lebih actor dengan sistem informasi yang dibuat. Secara umum, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang hendak menggunakan fungsi-fungsi itu. (Rosa AS. & M. Salahuddin; 2015)  
Penjelasan simbol dapat dilihat pada tabel berikut:

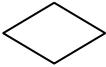
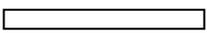
Tabel 2.1 Bagan *Use Case Diagram*

Simbol	Keterangan
<i>Actor</i> 	Mendefinisikan entitas diluar sistem yang memakai sistem.
<i>Use Case</i> 	Gambaran fungsionalitas dari sistem, sehingga pengguna mengerti kegunaan system.
<i>Relasi</i> 	Menceritakan hubungan antara aktor dan <i>use case</i> sehingga diagram dapat di pahami
<i>Ekstense / Extend</i> 	Relasi Use Case tambahan kesebuah Use Case yang di tambahkan dapat berdiri sendiri walau tanpa Use Case tambahan itu mirip dengan prinsip <i>inheritance</i> pada program berorientasi objek. Biasanya Use Case tambahan memiliki nama depan sama dengan use case yang ditambahkan
<i>Include</i> 	Digunakan untuk pengecekan apakah usecase yang ditambahkan telah dijalankan sebelum Use Case tambahan dijalankan.

#### b. *Activity Diagram*

Diagram aktifitas atau *activitydiagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis, atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa AS. & M. Salahuddin; 2015) Sebagai contoh fungsi matematika, pemanggilan perilaku, pemrosesan data. Aktivitas dapat mengakses atribut dan operasi dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan untuk model proses bisnis, informasi biasanya disebut *procces-relevant data*. Simbol *activitydiagram* dapat dilihat pada tabel berikut:

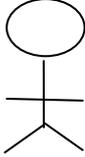
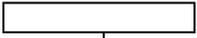
Tabel 2.2 Bagan *Activity Diagram*

Simbol	Keterangan
<i>Partition</i> 	Memperlihatkan dimana aliran berawal
<i>Relasi</i> 	Merupakan langkah atau aksi-aksi yang terjadi
<i>decision</i> 	Memperlihatkan dimana keputusan perlu diambil selama terjadi selama terjadi aliran kerja
<i>Fork</i> 	Merupakan percabangan dari sebuah aksi-aksi yang terjadi
<i>Final Node</i> 	Memperlihatkan dimana aliran itu berakhir

### c. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek. Oleh karena itu untuk menggambarkan diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. *Sequence diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara Khusus berasosiasi dengan *use case*. *Sequence Diagram* memperlihatkan tahap demi tahap apa yang sebenarnya terjadi untuk menghasilkan sesuatu didalam *use case*. (Rosa A.S & M.Shalahudin; 2015) Penjelasan *sequencediagram* dapat dilihat pada tabel 2.3.

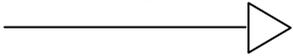
Tabel 2.3 Bagan *Sequence Diagram*

Simbol	Keterangan
<i>Actor</i> 	Prilaku didalam <i>sequence Diagram</i>
<i>Object</i> 	Menambahkan Objek baru pada diagram
<i>Object Messege</i> 	Menggambarkan pesan ( <i>message</i> ) antar dua objek
<i>Messege To Self</i> 	Menggambarkan pesan ( <i>message</i> ) yang menuju dirinya sendiri
<i>Return Messege</i> 	Menggambarkan pengembalian dari pemanggilan prosedur

#### d. *Class Diagram*

*Class Diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. (Rosa A.S & M. Shalahudin; 2013) Penjelasan simbol *class diagram* pada tabel berikut:

Tabel 2.4 Bagan *Class Diagram*

Simbol	Keterangan
<i>Class</i> 	Menambahkan kelas baru pada diagram
<i>Interface</i> 	Menambahkan kelas antarmuka( <i>interface</i> ) pada diagram
<i>Association</i> 	Menggambar relasi asosiasi
<i>Association class</i> 	Menghubungkan ( <i>association class</i> ) suatu relasiAsosiasi
<i>Generalization</i> 	Menggambarkan relasi generalisasi
<i>Aggregation</i> 	Menggambarkan relasi agregasi

### 2.3 Database

*Database* merupakan pangkalan data (tempat data) yang disusun sedemikian rupa dengan tujuan database tersebut efisien dan efektif, serta mengikuti aturan alat desain database terlepas dari perangkat lunak pemrograman yang digunakan. (Ir.Yuniar Supardi, 2011)

Adapun elemen-elemen penyusun basis data adalah sebagai berikut:

#### 1. Data Value (Nilai Data)

Data Value adalah data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat di mana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.

#### 2. Penyimpanan

1) *File/Tabel* : Kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

- 2) *Record/Tuple* merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.
- 3) *Entitas* adalah sekumpulan objek yang terdefiniskan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat, suatu kejadian atau konsep yang informasinya direkam.
- 4) *Atribut* adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas obyek, atau dengan kata lain, kumpulan atribut dari setiap entitas mengidentifikasi keunikan individu.
- 5) *Primarykey* adalah suatu kolom (*field*) yang menjadi titik acuan pada sebuah tabel, bersifat unik dalam artian tidak ada satu nilai pun yang sama atau kembar dalam tabel tersebut, dan dalam satu tabel hanya boleh ada satu primary key.
- 6) *Foreignkey* atau disebut juga kunci relasi adalah suatu kolom dalam tabel yang digunakan sebagai “kaitan” untuk melengkapi satu hubungan yang didapati dari tabel induk, dan biasanya hubungan yang terjalin antar tabel adalah satu ke banyak (*one to many*).
- 7) *Index* adalah struktur basis data secara fisik, yang digunakan untuk optimalisasi pemrosesan data dan mempercepat proses pencarian data.

### 2.3.1 Istilah Dalam Database

Dalam *database*, terdapat beberapa istilah, adapun definisi dari istilah-istilah itu adalah sebagai berikut:

#### 1. *Field*

Sekumpulan tanda-tanda yang membentuk kesatuan tersendiri. Merupakan bagian terkecil dari *record* dan yang bentuknya unik akan dijadikan *field* kunci yang dapat mewakili *recordnya*.

#### 2. *Candidate Key* (kunci kandidat)

*Candidate key* adalah satu atribut suatu sel minimal atribut yang mengidentifikasikan secara unik suatu kejadian spesifikasi dari *entity*.

### 3. *File* (berkas)

Kumpulan *record-record* sejenis dan saling berhubungan

### 4. Tambah Data

Menambah sejumlah data ke dalam *database* dengan cara memasukkan data.

### 5. *Editing* Data (perbaikan data)

Data yang tersimpan dalam *database* tidak terlepas dari kesalahan, hal ini dapat disebabkan oleh kesalahan pada waktu memasukkan data atau data yang sebenarnya telah berubah. Agar informasi yang disampaikan selalu *update*, maka hendaknya data yang mengandung kesalahan harus diperbaiki

### 6. *Delete* (hapus data)

Dalam suatu *database* kita sering menemukan data yang sudah tidak terpakai. Data yang tidak terpakai tersebut hendaknya dibuang. Selain memperkecil *database* juga mempengaruhi proses manipulasi data

Dalam *database* kita mengenal beberapa relasi antar tabel. Relasi antar tabel merupakan pengelompokan data menjadi tabel-tabel yang menunjukkan *entity* dan relasinya yang berfungsi untuk mengakses data *item* sedemikian rupa sehingga *database* tersebut mudah dimodifikasi. Pada dasarnya kunci (*key*) adalah satu atau gabungan dari beberapa *atribut* yang dapat membedakan semua baris data (*row*) dalam tabel secara *unik*. Ada empat kunci yang dapat digunakan dalam suatu tabel, yaitu :

#### 1. *Super key*

Merupakan kumpulan atau lebih *atribut* (kumpulan *atribut* ) yang dapat membedakan semua baris dalam sebuah tabel yang *unik*.

#### 2. *Candidate key*

Merupakan kumpulan *atribut* yang dapat membedakan setiap baris dalam sebuah tabel yang secara *unik*.

### 3. *Primary key*

Merupakan satu atribut yang tidak hanya mengidentifikasi secara unik untuk kejadian yang *spesifik*, tetapi dapat pula mewakili setiap kejadian dalam suatu *entitas*.

### 4. *Foreign key*

Merupakan satu *atribut* tamu dalam suatu *entitas*.

## 2.3.2 Relasi Antar Tabel

Relasi antar tabel adalah keterhubungan atau keterkaitan antara satu entitas dari satu atau lebih dengan entitas lain. Sedangkan entitas itu sendiri adalah sebuah nama lain dari *record* yaitu objek unik yang dapat dibedakan dengan dengan objek lain dalam sebuah file master.

Model data *relasional* menggunakan sekumpulan tabel berdimensi dua (yang bisa disebut relasi tabel) dengan masing-masing tabel tersusun atas sejumlah baris dan kolom.

Beberapa jenis relasi antar tabel, yaitu sebagai berikut :

#### 1. *1-1*

Hubungan antara lain *file* pertama dan *file* kedua adalah satu berbanding satu. Contoh tabel\_*probadi* dan tabel\_*pekerjaan*

#### 2. *1-M*

Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak lawan satu. Contoh tabel\_*pekerjaan* dan tabel\_*bagian*. (Abdul Kadir, 2014)

## 2.3.3 Kamus Data

Pada tahap analisis, Kamus Data (KD) dapat digunakan sebagai alat komunikasi antara analis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

Kamus data adalah kumpulan daftar elemen yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan Keluaran (output) dapat dipahami secara umum. (Rosa A.S & M. Salahuddin, 2015).

**Format Kamus Data**

Nama Database :  
 Nama Tabel :  
 Field Kunci :  
 Kunci Tamu :  
 Panjang Field :

Field	Data Type	Length	Keterangan

Gambar 2.3 *Format Kamus Data*

### 2.3.4 Pengkodean

Pengkodean atau kode adalah suatu pengkodean digunakan untuk tujuan mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Ada beberapa macam tipe dari kode yang digunakan didalam sistem informasi, diantaranya adalah.

1) Kode (*Memonic Code*)

Kode *Memonik* digunakan untuk tujuan supaya mudah diingat. Kode *memonik* dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang diwakili dengan kode ini. Umumnya kode *memonik* menggunakan huruf, tetapi juga dapat dengan gabungan huruf dan angka.

2) Kode Urut (*Sequential Code*)

Kode urut disebut juga dengan kode seri (*serial code*) merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

3) Kode Blok (*Block Code*)

Kode blok mengklasifikasikan item ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

#### 4) Kode Group (*Group Code*)

Kode group merupakan kode yang berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

#### 5) Kode Desimal (*Desimal Code*)

Kode desimal *mengklasifikasi* kode atas dasar 10 unit angka desimal dimulai angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

## 2.4 Teori Perangkat Lunak

### 2.4.1 MySQL

*MySQL* bukan termasuk bahasa pemrograman. *MySQL* merupakan salah satu database populer yang mendunia. Pada umumnya perintah yang sering digunakan dalam *MySQL* adalah *Select* (mengambil), *Insert* (menambah), *Update* (mengubah), dan *Delete* (menghapus). *MySQL* bekerja dengan menggunakan bahasa basis data atau bahasa yang sering disebut dengan *Database Management System* (DBMS). (Agus Saputra, 2013)

MySQL memiliki beberapa keistimewaan, antara lain :

#### 1. Portabilitas.

*MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.

#### 2. *Open Source*.

*MySQL* didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

#### 3. *Multiuser*.

*MySQL* dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

#### 4. *Performance tuning*'.

*MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

#### 5. Perintah dan Fungsi.

*MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

#### 6. Keamanan

*MySQL* memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

#### 7. Antar Muka.

*MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

### 2.5.2 *Microsoft visual studio 2012*

*Microsoft visual studio* adalah sebuah Integrated Development Environment buatan *Microsoft Corporation*. *Microsoft visual studio* dapat digunakan untuk mengembangkan aplikasi dalam native code ataupun managed code. (wahana, 2013)

*Microsoft visual studio* merupakan sebuah perangkat lunak lengkap yang dapat melakukan pengembangan aplikasi baik itu bisnis maupun personal ataupun kompebeb aplikasinya, *visual studio* mencakup kompiler ,SDK, IDE, dan MSDN Library. kompiler yang dimasukan kedalam paket Visual studio antara lain Visual C++, Visual #, Visual Basic, Visual Basic NET, Visual InterDev, Visual J++, dan Visual SourceSafe. (wahana, 2013) Berikut beberapa kelebihan dan kekurangan *Microsoft visual studio*, fitur – fitur penting yang terdapat pada *Microsoft visual studio* serta langkah – langkah dalam menjalankan Visual studio.

Berikut beberapa kelebihan dan kekurangan *Visual studio*, fitur – fitur penting yang terdapat pada *Visual studio 2012* serta langkah – langkah dalam menjalankan *Visual studio*

### 1. Kelebihan *Visual studio*

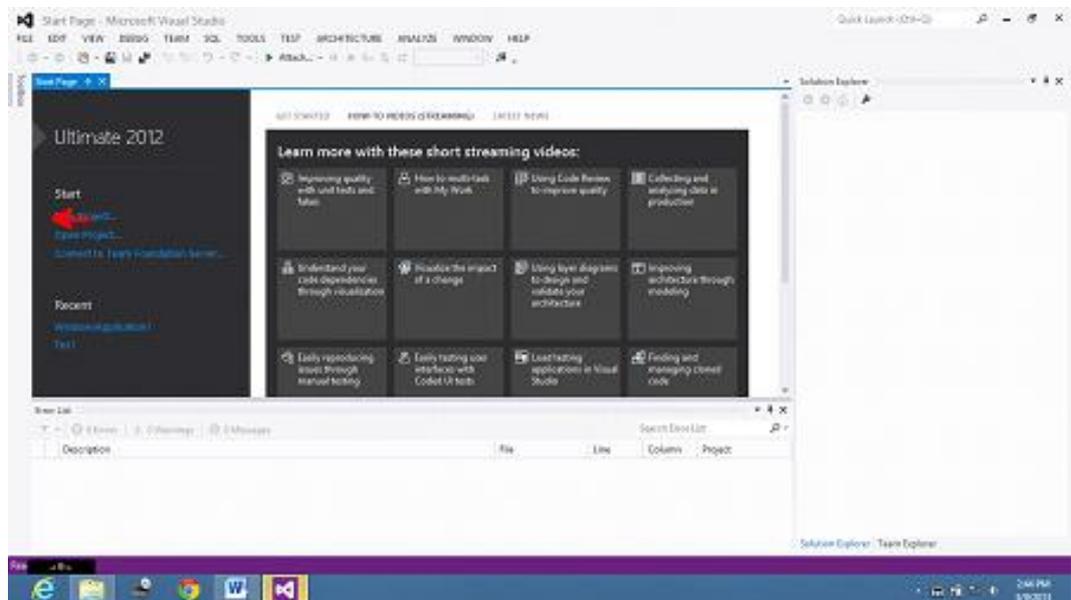
Salah satu yang menjadi kelebihan *Visual studio* adalah mempunyai banyak fitur yang membuat tampilannya elegan dan sangat user friendly.

### 2. Kekurangan *Visual studio*

*Visual studio* bahasa pemrograman yang powerfull, lebih lambat dibandingkan bahasa pemrograman lain dan tidak punya pendukung untuk membuat report bawaan sendiri.

### 3. Bagian-bagian *Visual studio*.

1. Interface atau antarmuka *Visual studio* berisi menu toolbar, toolbox, from, project explore seperti gambar dibawah ini.

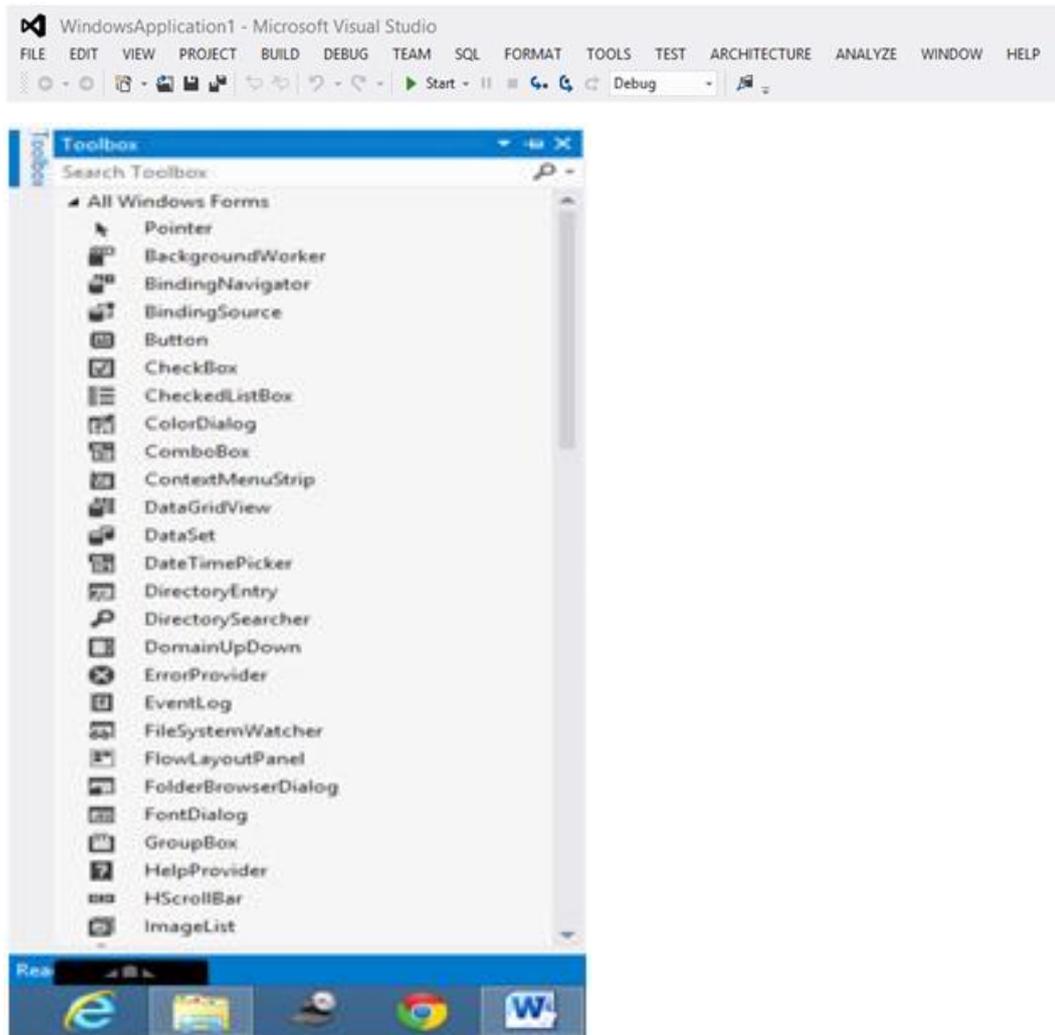


Gambar 2.4 Antar muka *Visual studio*

Dalam pembuatan aplikasi menggunakan visual basic dilakukan dengan form kemudian diberi script dalam komponen yang diperlukan. form disusun oleh komponen-komponen yang ada dalam toolbox dan setiap komponen harus diatur melalui propertinya pada jendela property.

## 2. toolbox dan menu

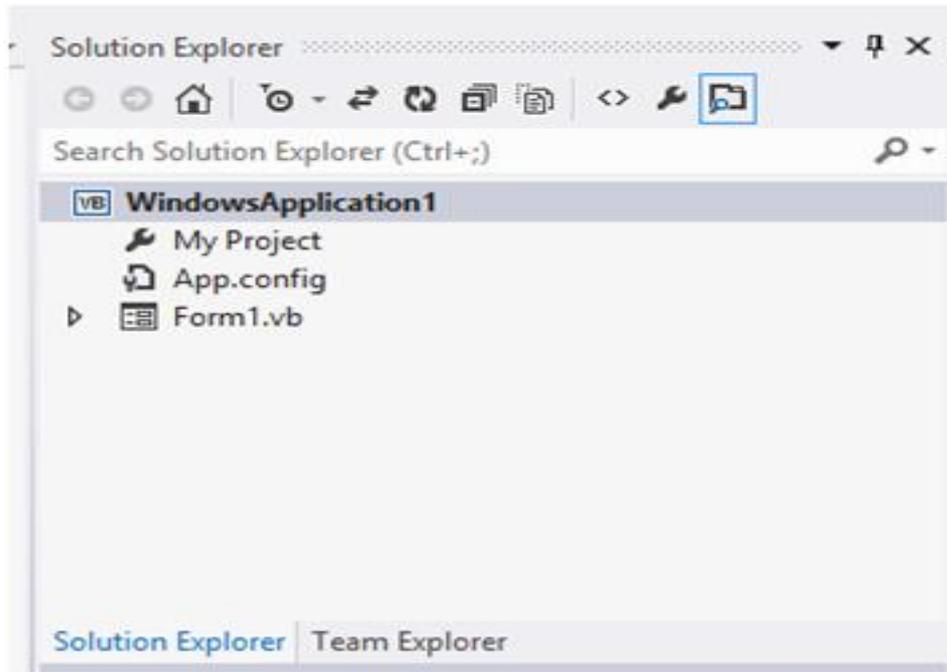
Toolbox dan menu visual studio 2012 yang nantinya digunakan untuk mengakses menu-menu yang ada pada visual basic, dan toolbox nya yang berisi button, textbox, label dan panel-panel lainnya.



Gambar 2.5 tampilan menu dan toolbox *Visual studio*

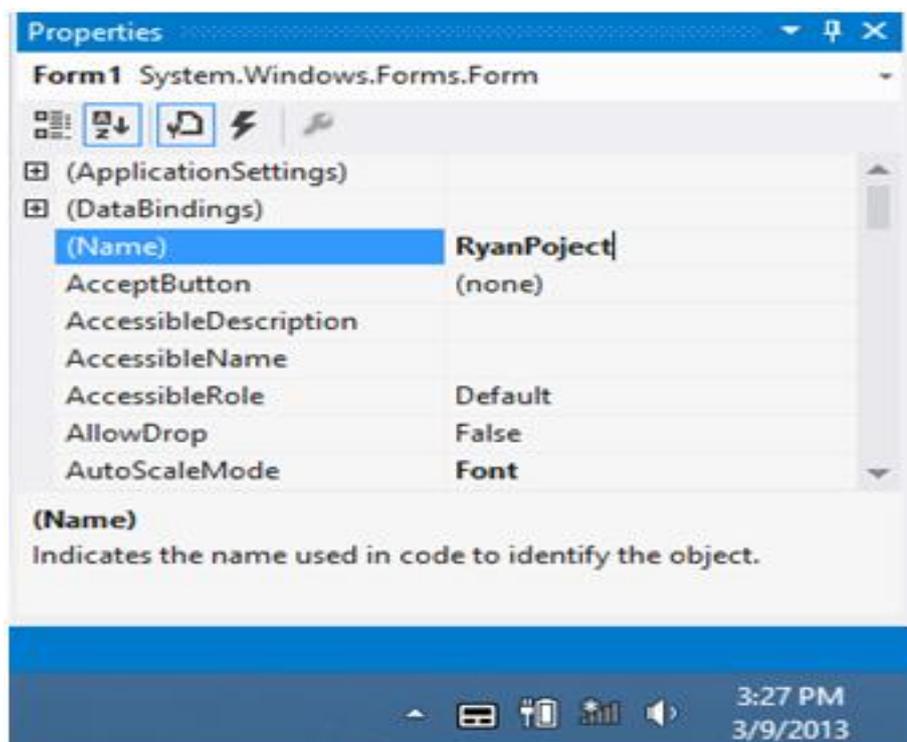
## 3 solution explorer, properties, dan coding area visual studio

A. solution explorer merupakan item-item yang digunakan dalam pembuatan sebuah project



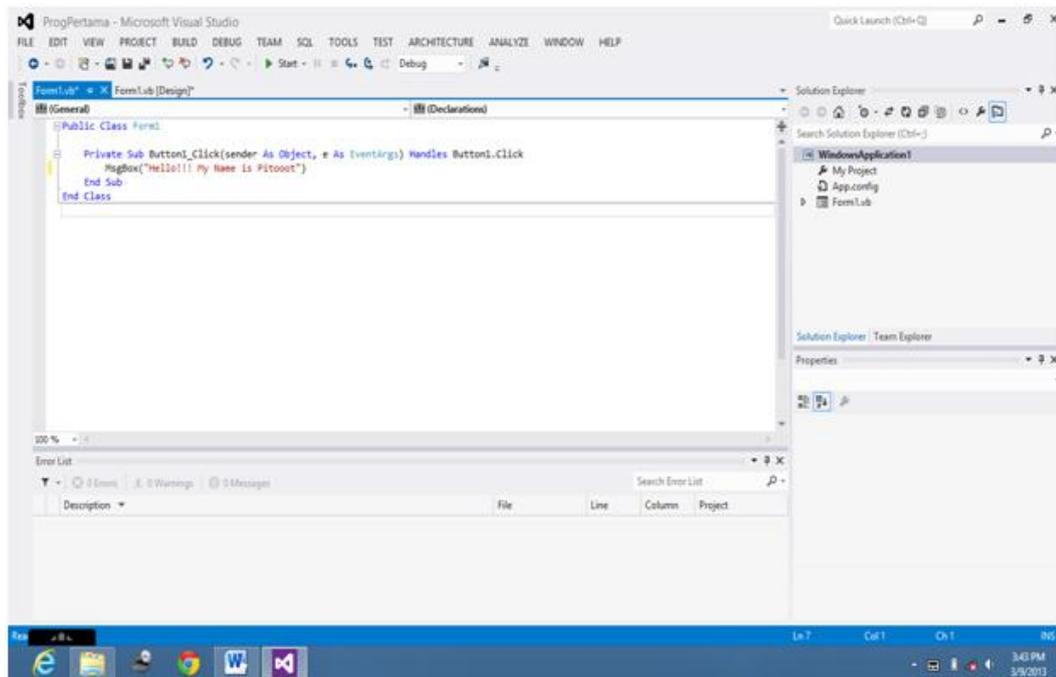
Gambar 2.6 tampilan solutions explorer *Visual studio*

B. properties merupakan kumpulan nilai/value dari sebuah object sebagai contoh button memiliki properties nama, colour, visible, dan sebagainya kesemua itu dapat anda tentukan nilai nya diproperties.



Gambar 2.7 tampilan Properties *Visual studio*

C. Area coding merupakan bagian dari area kerja Visual basic yang berfungsi untuk menempatkan syntax atau code-code program.



Gambar 2.8 tampilan Coding area *Visual studio*

## 2.6 Teori Black Box Testing

Black Box adalah pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan cek fungsional perangkat lunak. Jadi analogi seperti yang kita lihat kotak hitam, kita dapat melihat hanya penampilan luar, tanpa mengetahui apa dibalik bungkus hitamnya. Sama seperti pengujian kotak hitam, mengevaluasi hanya penampilan eksternal (antar muka), fungsional tanpa mengetahui apa yang sebenarnya terjadi dalam codingan.

### 2.6.1 Black Box Testing dapat menemukan error seperti :

- Fungsi atau logika yang tidak benar
- Error interface
- Error performance
- Kesalahan dalam struktur data atau akses database eksternal

### **2.6.2 Kelebihan Blck Box Testing :**

- Tidak perlu melihat source code secara detail
- Mendeteksi kesalahan pengetikan / typo
- Mendeteksi kesalahan design / user interface dari sebuah software / website
- Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
- Seorang tester tidak harus programmer

### **2.6.3 Kekurangan Blck Box Testing**

- Ketregantungan dengan dokumen dan design software tersebut
- Tidak sampai level code, sehingga tester tidak mengetahui level security dari software tersebut

Tujuan Black Box Testing adalah untuk mencari kesalahan / kegagalan dalam oprasi tingkat tinggi, yang mencakup kemampuan dari perangkat lunak, oprasional / tata laksana, skenario pemakai.