BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktik

2.1.1 Sejarah CCA (Cimahi Creative Association)

Sejak Cimahi menjadi pemerintahan kota pada tahun 2001, telah dirasakan bahwa industry yang berkembang sudah tidak mendapatkan lagi dukungan sumber daya alam sehaingga pada tahun 2004 pemerintah Kota Cimahi melakukan riset bekerja sama dengan Institut Teknologi Bandung (ITB), yang merekomendasikan bahwa Kota Cimahi harus mengembangkan Teknologi Informasi sehingga pada tahun 2005 dideklarasikan "Cimahi Cyber City".

Namun pada perkembangannya pada tahun 2009 hasil riset yang dilakukan oleh "PT. Sembilan Matahari" merekomendasikan bahwa Kota Cimahi harus mengembangkan Animasi dan Film. Sejalan dengan perkembangan didunia dengan fenomena Industri Kreatif, maka pemerintah kota mendeklarasikan untuk mengembangkan Industri Kreatif dengan unggulan IT, Animasi, dan Film. Hal tersebut didukung oleh sumber daya manusia dan potensi pasar yang besar. Maka untuk mendukung rencana tersebut pemerintah kota melaksanakan FGD Industri Kreatif yang pertama yang tujuannya menghimpun pelaku kreatif. Ternyata hasilnya cukup menjanjikan karena jumlah pelaku kreatif cukup signifikan dan memiliki kemampuan yang baik. Pada FGD ke-2 yang dilaksanakan pada tanggal 29 September 2009 maka dibentuk lah Cimahi Creative Association (CCA) berikut pemilihan ketuanya. Pada FGD ke-3, tanggal 28 Oktober 2009, kelembagaan dan kepengurusan CCA diresmikan dengan SK Walikota No. 530/KEP 304 –

PENMO/2009. Sedangkan pada FGD ke-4 CCA menyusun Anggaran Dasar (AD), Anggaran Rumah Tangga (ART), serta rencana kerja tahun berikutnya.

2.1.2 Logo CIMAHI CREATIVE ASSOCIATION

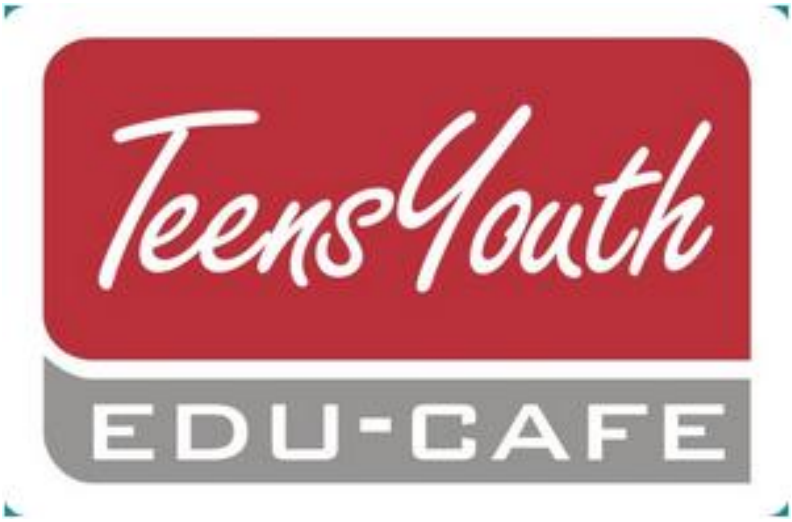
Berikut gambar logo CIMAHI CREATIVE ASSOCIATION :



Gambar 1 Logo CCA

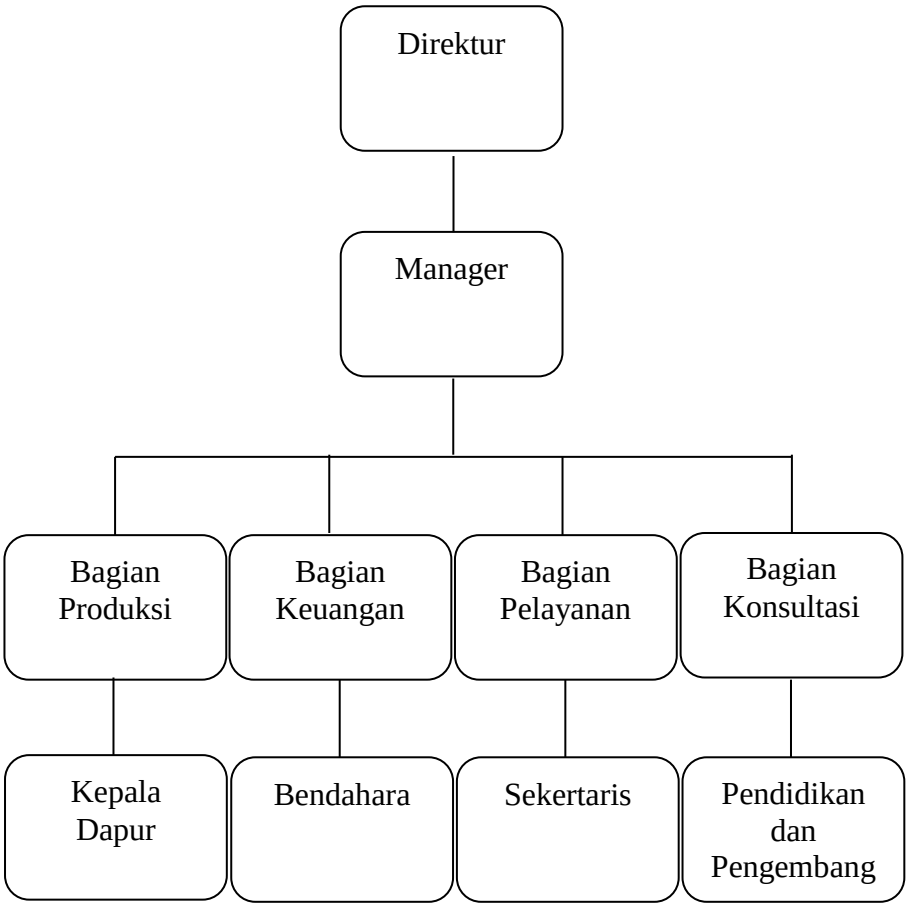
2.1.3 Logo Teens Youth Educational Café

Berikut adalah Logo Teens Youth Educational Café :



Gambar 2 Logo Teens Youth Educational Café

2.1.4 Struktur Organisasi dan Job Description di Teens Youth Educational Café



Gambar 3 Struktur Organisasi T-Youth Café

2.1.5 Bagan Struktur Organisasi

Keterangan :

Direktur : Farid

Manager : Toyib

Kepala Dapur : Riska

Bendahara : Niken Oktaviani

Sekretaris : Andina Rostaviani

Pendidikan dan Pengembangan : Weny Windya Hapsyary

Pendidikan dan Pengembangan : Sinta

2.2 Landasan Teori

2.2.1 VB.Net



Gambar 4 VB.Net

2.2.1.1 Mengenal VB.NET

VB.NET adalah salah satu bahasa pemograman yang popular saat ini . Kenapa disebut .NET? Apakah ada hubungannya dengan internet? Kenapa ada kata ".Net" nya? Ternyata setelah diselisik tidak ada hubungannya lho dengan internet, mungkin dalam pikiran pembaca semuanya, ini berarti ada hubungannya dengan internet (hehe… setidaknya itu yang saya pikirkan begitu pertama kali

dihadapkan dengan VB.NET). VB.NET tentu berbeda dengan VB.6. Pendahulunya itu tidak begitu powerfull dibandingkan dengan VB.NET karena secara struktur sudah berbeda, berikut beberapa perbedaan VB.NET dan VB6 :.

1. VB.NET menggunakan framework
2. VB.NET menggunakan dataset, sedangkan VB6 recordset
3. VB.NET menggunakan XML(bisa dijadikan pengganti database) VB6 tidak bisa.
4. VB.NET komponennya (textbox dll) mengalami penambahan-penambahan, VB6 cuma itu-itu saja kecuali menggunakan OCX atau instal komponen tambahan
5. VB.NET bahasanya dipermudah(kalau ada salah syntak akan ada pemberitahuannya secara langsung kalau ada error) VB6 tunggu di F5(debug) baru tau kalau ada yang error
6. VB.NET bisa membuat file .dll / .ocx (Class Library) tanpa harus d registrasikan d system32 klo VB6 harus d registrasikan dulu.
7. Perbedaan mendasar pada VB.NET dibandingkan VB 6 adalah .NET framework itu sendiri dan konsep OOP.
8. Teknologi .NET framework dengan arsitekturnya sangat berbeda dengan teknologi sebelumnya seperti pada VB 6. .NET framework memiliki compiler untuk masing-masing bahasa; Untuk VB.NET ada VB compiler, C# ada C# compiler, dan compiler bahasa lainnya yang merubah source code menjadi Intermediate Languge (IL)
9. VB.NET source code --(VB compiler)--> IL --(CLR-JIT compiler)--> native code : CLR adalah sebuah Runtime yang berfungsi sebagai virtual machine dan CLR melalui Just In Time (JIT) compiler nya, akan melakukan kompilasi terhadap IL menjadi native code OS. Sementara pada VB 6 seperti ini: VB 6 source code --(compiler)--> native code
10. Dilihat dari segi OOP : Mindset seorang developer VB 6 dengan procedural programming nya berbeda dengan developer VB.NET yang OOP minded. Oleh karena itu,. Walaupun bahasa yang digunakan pada

VB dan VB.NET relatif sama, tapi konsep procedural programming dan OOP di dalamnya sangat berbeda.

2.2.1.2 NET Framework

.NET Framework adalah teknologi inti yang menyediakan berbagai library untuk digunakan oleh aplikasi di atasnya. Komponen inti .NET Framework adalah Common Language Runtime (CLR) yang menyediakan run time environment untuk aplikasi yang dibangun menggunakan Visual Studio .NET, terlepas dari jenis bahasa pemrogramannya. Dengan adanya CLR tersebut, programmer dapat menikmati consistent object model dalam mengakses berbagai komponen library.

Dengan demikian penggunaan bahasa pemrograman dalam dunia .NET adalah lebih ke masalah selera atau taste, dan bukan pada kelebihan maupun kekurangan masing-masing bahasa. Mengapa ? Karena semua bahasa pemrograman yang mensupport .NET mengakses library yang sama di dalam .NET Framework, dengan object model yang konsisten, dengan run time file yang sama. Bahasa adalah sekedar skin atau theme, bukan senjata sakti. Bagi seorang .Net Developer, pemahaman terhadap konsep dan object model .NET Framework adalah jauh lebih penting daripada bahasa pemrograman itu sendiri.

2.2.1.3 Keuntungan Framework .NET

Berikut beberapa keuntungan dari Framework .NET :

- ✓ Mudah

Kemudahan di sini lebih ke arah pada kemudahan bagi para developer untuk membuat aplikasi yang dijalankan pada lingkungan Framework .NET. Beberapa hal yang merepotkan developer pada saat membuat aplikasi, telah di hilangkan atau di ambil alih kemampuannya oleh Framework .NET, misalnya masalah *lifetime* sebuah objek yang biasanya luput dari perhatian developer pada saat proses pembuatan aplikasi. Masalah ini telah ditangani dan diatur secara otomatis oleh Framework .NET melalui komponen yang bernama *Garbage Collector* yang bertanggung jawab untuk mencari dan membuang objek yang sudah tidak terpakai secara otomatis.

- ✓ Efisien

Kemudahan pada saat proses pembuatan aplikasi, akan berimplikasi terhadap efisiensi dari suatu proses produktivitas, baik efisien dalam hal waktu pembuatan aplikasi atau juga efisien dalam hal lain, seperti biaya (*cost*).

- ✓ Konsisten

Kemudahan-kemudahan pada saat proses pembuatan aplikasi, juga bisa berimplikasi terhadap konsistensi pada aplikasi yang kita buat. Misalnya, dengan adanya BCL, maka kita bisa menggunakan objek atau Class yang dibuat untuk aplikasi berbasis windows pada aplikasi berbasis web. Dengan adanya kode yang bisa dintegrasikan ke dalam berbagai macam aplikasi ini, maka konsistensi kode-kode aplikasi kita dapat terjaga.

- ✓ Produktivitas

Semua kemudahan-kemudahan di atas, pada akhirnya akan membuat produktivitas menjadi lebih baik. Produktivitas naik, terutama produktivitas para developer, akan berdampak pada meningkatnya produktivitas suatu perusahaan.

2.2.1.4 Struktur Program Visual Basic 2010

Struktur program VB.NET terdiri dari 3 bagian, yaitu :

- ✓ Statement Option : digunakan untuk menangani masalah kode di aplikasi
  - o Option Explicit : Option Explicit on/off digunakan untuk menangani apakah variabel harus dideklarasi (on) atau tidak (Off). Sebagai contoh ketik contoh program di latihan 1 a dan 1 b. Defaultnya adalah Option Explicit On.
  - o Option Strict : Option strict digunakan untuk memperperbolehkan konversi data dari suatu variabel ke variabel lain. Option Strict On membolehkan konversi variabel sedangkan Option Strict Off tidak diperbolehkan defaultnya adalah option strict off.
- ✓ Statement Imports : Imports berguna untuk mendefinisikan class sehingga memudahkan pengetikan class tanpa harus namespace berulangkali.
- ✓ Class : Class merupakan tempat mendefinisikan algoritma yang anda buat.

2.2.2 Microsoft Access



Gambar 5 Microsoft Access

2.2.2.1 Pengertian Microsoft Access

3 Microsoft Access adalah suatu program aplikasi basis data komputer relasional yang digunakan untuk merancang, membuat dan mengolah berbagai jenis data dengan kapasitas yang besar.Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007.

4 Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/programmer yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

2.2.2.2 Komponen Utama (Object)

1. **Table**

Table adalah objek utama dalam database yang digunakan untuk menyimpan sekumpulan data sejenis dalam sebuah objek. Table terdiri atas :

a. *Field Name* : atribut dari sebuah table yang menempati bagian kolom.

b. *Record* : Isi dari field atau atribut yang saling berhubungan yang menempati bagian baris.

2. **Query (SQL / Structured Query Language)**

   Query adalah bahasa untuk melakukan manipulasi terhadap database. Digunakan untuk menampilkan, mengubah, dan menganalisa sekumpulan data. Query dibedakan menjadi 2, yaitu :

   a. *DDL* (Data Definition Language) digunakan untuk membuat atau mendefinisikan obyek-obyek database seperti membuat tabel, relasi antar tabel dan sebagainya.

   b. *DML* (Data Manipulation Language) digunakan untuk manipulasi database, seperti : menambah, mengubah atau menghapus data serta mengambil informasi yang diperlukan dari database.

3. **Form**

   Form digunakan untuk mengontrol proses masukan data (input), menampilkan data (output), memeriksa dan memperbaharui data.

4. **Report**

   Form digunakan untuk menampilkan data yang sudah dirangkum dan mencetak data secara efektif.

2.2.2.3 Tipe Data

Field - field dalam sebuah tabel harus ditentukan tipe datanya. Ada beberapa tipe data dalam Access, yaitu :

1. **Text**

   Text digunakan untuk field alfanumeric (misal : nama, alamat, kode pos, telp), sekitar 255 karakter tiap fieldnya.

2. **Memo**

Memo dapat menampung 64000 karakter untuk tiap fieldnya, tapi tidak bisa diurutkan/diindeks.

3. **Number**

   Number digunakan untuk menyimpan data numeric yang akan digunakan untuk proses perhitungan matematis.

4. **Date/Time**
5. **Currency**
6. **Auto Numbe*r***
7. **Yes/No**
8. **OLE Object**

   OLE Object digunakan untuk eksternal objek, seperti bitmap atau file suara.

9. **Hyperlink**
10. **Lookup Wizard**

    Jika menggunakan tipe data ini untuk sebuah field, maka bisa memilih sebuah nilai dari tabel lain atau dari sebuah daftar nilai yang ditampilkan dalam combo box.

2.2.3 Sistem Informasi

2.2.3.2 Pengertian Sistem Informasi

Sistem informasi yaitu suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang tergorganisasi. Biasanya suatu perusahan atau badan usaha menyediakan semacam informasi yang berguna bagi manajemen. Sebagai contoh: Perusahaan toko buku mempunyai sistem informasi yang menyediakan informasi penjualan buku-buku setiap harinya, serta stock buku-buku yang tersedia, dengan informasi tersebut, seorang manajer bisa membuat kebutusan, stock buku apa yang harus segera mereka sediakan

untuk toko buku mereka, manajer juga bisa tahu buku apa yang paling laris dibeli konsumen, sehingga mereka bisa memutuskan buku tersebut jumlah stocknya lebih banyak dari buku lainnya.

2.2.3.3 Pengertian Sistem Informasi menurut beberapa Ahli

Pengertian sistem informasi menurut John F. Nash

Sistem Informasi adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksud menata jaringan komunikasi yang penting, proses atas transaksi-transaksi tertentu dan rutin, membantu manajemen dan pemakai intern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat.

Pengertian sistem informasi menurut Henry Lucas

Sistem Informasi adalah suatu kegiatan dari prosedurprosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam.

2.2.3.4 Komponen Sistem Informasi

1. komponen input input merupakan data yang masuk ke dalam sistem informasi.
2. komponen model kombinasi prosedur,logika,dan model matematika yang memproses data yang tersimpan di basis data dengan cara yang sudah di tentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen output output informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Kompone teknologi teknologi merupakan alat dalam sistem informasi, teknologi digunakan untuk menerima innput, menjalankan model, minyimpan dan mengakses data, menghasilkan dan mengirimkan

output dan memantu pengendalian sistem.

5. Komponen basis data merupakan kumpulan data yang saling berhubungan yang tersimpan didlm komputer denganmenggunakan softwre database.
6. Komponen kontrol pengendalian yang dirancang untuk menanggulangi gagguan terhadapsistem informasi.

2.2.3.5 Karakter Sistem informasi

1. sistem informsi memiliki komponen yang berupa subsistem yang merupakan elemen-elemen yang lebih kecil yang membentuk sistem informasi tersebut misalnya bagian input, proses, output. Contoh input adalah salesman memasukan data penjualan bulan ini, maka disana terdapat manusia yang melakukan pekerjaan input dengan menggunakan hardware keyboard dan menggunakan interface sebuah aplikasi laporan penjualan yang sudah di sediakan oleh sistem informasi tersebut.
2. Ruang lingkup sistem informasi yaitu rung lingkup yang ditentukan dari awal pembuatan yang meupakan gari bats lingkup kerja sistem tersebut sehingga sistem informasi tersebut tidak bersinggungan dengan sistem informasi lainnya.
3. Tujuan sistem informasi adalah hal pokok yang harus ditentukan dan dicapai dengan menggunakan sistem informasi tersebut, sebuah informasi dianggap berhasil apabila dapat mencapai tujuan tersebut.
4. Lingungan sistem informasi yaitu sesuatu yang berada diluar ruang lingkup sistemm informasi yang dapat mempengaruhi sistem informasi, hal ini urut dipertimbangkan pada saat perencanaann sistem informasi.

2.2.4 Data

Data merupakan deskripsi tentang benda, kejadian, aktivitas, dan transaksi yang tidak mempunyai makna sehingga tidak memiliki pengaruh langsung kepada pemakai. Data dapat berupa (Kadir, 2003) :

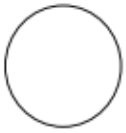
1. Data atau nilai yang terformat yaitu data dengan suatu format tertentu, seperti data yang menyatakan tanggal, jam, dan nilai mata uang.
2. Teks merupakan deretan huruf, angka, dan simbol khusus yang memiliki kombinasi tidak tergantung pada masing-masing item secara individual. Misalnya artikel surat kabar dan majalah.
3. Citra atau *image* merupakan data yang berbentuk gambar. Citra dapat berupa grafik, vektor, foto, hasil *rontgen*, tanda tangan dan lain-lain.
4. Audio adalah data yang berbentuk suara, seperti instrumen musik, suara orang atau binatang, gemericik air, suara angin, dan lain-lain.
5. Video merupakan data dalam bentuk sejumlah gambar yang bergerak, dan dapat juga dilengkapi dengan audio. Video juga dapat digunakan untuk mengabadikan suatu kejadian.

2.2.5 Data Flow Diagram (DFD)

Data flow diagram adalah sebuah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat bergerak dari input menjadi output (Pressman,2002).

DFD dapat digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap tingkat abstraksi. Notasi dasar yang digunakan untuk menciptakan suatu DFD dijelaskan pada table dibawah ini :

Table 1 Tabel Notasi DFD

No	Gambar	Keterangan
1		Lingkaran menunjukkan sistem secara keseluruhan. Penamaan sebuah lingkaran dapat berupa kata, frase atau sebuah kalimat sederhana yang menjelaskan nama didtem itu sendiri.
2		Panah menunjukkan arah aliran data dari sistem ke entity luar atau sistem ke data store atau sebaliknya. Gambar panah diberinama untuk menunjukkan data yang mengalir melalui aliran tersebut.
3		Data store (penyimpanan) digambarkan dengan dua buah garis sejajar mendatar, menunjukkan sekumpulan data yang disimpan. Data store berfungsi untuk tempat penyimpanan data yang dapat digunakan dalam sistem baik sebagai input untuk melakukan suatu proses untuk kemudian digunakan oleh proses-proses lainnya di dalam sistem.
4		Terminator digambarkan dengan sebuah kotak persegi panjang yang berhubungan dengan sistem. Sebuah terminator dapat berupa orang, sekumpulan orang, departement dalam perusahaan atau organisasi yang sama tetapi berada di luar sistem yang dimodelkan.

DFD memiliki beberapa level yang merupakan hasil perjejangan dasri suatu system yang dibuat (level 0, 1, 2, dst).

1. **DFD level 0 (*diagram context*)**

   DFD level 0 menggambarkan seluruh elemen sistem dengan sebuah proses tunggal dengan data input atau output yang ditunjukan oleh anak panh yang masuk dan keluar secara berurutan.

2. **DFD level 1**

   DFD level 1 menggambarkan proses dan jalur informasi pada DFD level 0 yang telah detail.

3. **DFD level 2**

   DFD level 2 merupakan pengembangan alur data atau pemecahan proses menjadi proses terpisah-pisah, tapi masih berkaitan.

2.2.6 Basis Data

Basis data didefinisikan dalam sejumlah sudut pandang (Fathansyah, 1999), yaitu:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi beberapa kebutuhan.
3. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Untuk menggambarkan data dan menghubungkan data diperlukan model data, model data didefinisikan sebagai sekumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantic (makna) data dan batasan data (Fathansyah, 1999). Secara umum model data diglongkan menjadi 2 (dua) kelompok yaitu :

1. Model Logik Data berdasarkan *object* (*Object-base Logical Models*) terdiri dari
    a. Model Keterhubungan *Entitas* (*Entity Relationship Model*)
    b. Model Berorientasi *Object* (*Object Oriented Model*)
    c. Model Data *Semantik* (*Semantic Data Model*)
    d. Model Data Fungsional (*Functional Data Model*)
2. Model Logik Data berdasarkan *Record* (*Record-Based Logical Models*) terdiri dari:
    a. Model *Relasional* (*Relational Model*)
    b. Model *Hirarkhis* (*Hierarchical Model*)
    c. Model *Jaringan* (*Network Model*)

Dari sekian banyak cara permodelan data, dalam tulisan ini digunakan model keterhubungan *Entitas* (*Entity Relationship Model*)

untuk memodelkan data, dimana model keterhubungan *entitas* merupakan model data yang paling popular digunakan dalam perancangan *basisdata*.

2.2.7 Entity Relationship Diagram (ERD)

ERD adalah sebuah model data yang didasarkan atas presepsi dunia nyata, yang terdiri atas object dasar yang disebut dengan entitas dan hubungan (*relation*) antar object-object tersebut. Tujuan dari ERD adalah untuk mewakili object data dan hubungan mereka.

Sesuai dengan namanya ada 2 komponen utama pembentuk model keterhubungan entitas yaitu *entity* (*entitas*) dan *relasi* (*relation*). Entitas menyatakan suatu *object* yang memprasentasikan suatu himpunan atau sesuatu didunia nyata yang mempunyai peranan dalam sistem yang sedang dibangun, sedangkan relasi merupakan sebuah kumpulan dari beberapa entitas atau relasi yang memiliki tipe sama. Pada model entity relationship diagram hubungan antar file direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama dari masing-masing file.

Untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan atribut dalam satu file, yaitu :

1. *One to one relationship*

   Hubungan antara file pertama dan file kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan table dan relasi antar keduanya digambarkan dengan panah tunggal.

2. *One to Many relationship*

   Hubungan antara file pertama dan file kedua adalah satu berbanding banyak atau dapat pula dibalik banyak berbanding satu. Hubungan tersebut dapat digambarkan dengan panah banyak untuk menunjukan hubungan banyak tersebut.

3. *Many to many relationship*

Hubungan antara file pertama dan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan panah ganda untuk menunjukkan hubungan banyak tersebut.

Berikut table 2 akan menunjukkan simbol-simbol yang digunakan dalam ERD.

Table 2 Simbol - simbol ERD

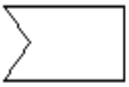
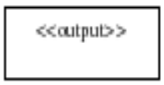
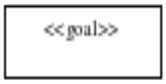
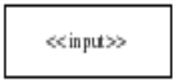
Gambar	Keterangan
	Entitas
	Hubungan
	Penghubung (link)
	One to Many
	One to One
	Many to Many

2.2.8 UML (Unified Modeling Language)

UML Adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

Table 3 Symbol – symbol UML.

Notasi	Deskripsi
 Actor	Menggambarkan pelaku / user dalam sebuah sistem. Pelaku ini meliputi manusia atau sistem komputer atau sub sistem lain yang memiliki method untuk melakukan sesuatu. *Contoh :* Customer, Manager, Application Server, dan lain-lain
 Event	Menggambarkan kejadian *(event)* atau peristiwa yang mendorong terjadinya proses. *Contoh :* Adanya Order, Jatuh Tempo, kebutuhan buku, dan lain-lain
 Process	Kumpulan aktivitas yang dirancang dalam tujuan tertentu untuk menghasilkan output khusus. Dapat juga dikatakan, proses merupakan rangkaian aktivitas dalam waktu dan tempat tertentu, yang memiliki awal dan akhir aktivitas; memiliki input dan output yang jelas. *Contoh :* Mengumpulkan data, Analisis Data, dan lain-lain
 Output	Sebuah proses dapat menghasilkan satu atau lebih output yang memenuhi kebutuhan eksternal atau sesuai dengan penggunaan internal. Output dapat berupa obyek fisik (seperti laporan *atau* invoice), rekomendasi atau hasil keseluruhan dari proses. Dalam beberapa kasus, output ini dapat menjadi sebuah *event* yang dapat mendorong terjadinya proses baru.
 Goal	Tujuan yang ingin dicapai dalam sebuah proses.
	Input, Resources dan Information yang digunakan dalam process.

Bagian-Bagian Uml

Bagian-bagian utama dari UML adalah view, diagram, model element, dan general mechanism.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis view dalam UML antara lain: use case view, logical view, component view, concurrency view,dan deployment view.

b. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. Actor yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

View ini digambarkan dalam use case diagramsdan kadang-kadang dengan activity diagrams. Viewini digunakan terutama untuk pelanggan, perancang (designer), pengembang (developer), dan penguji sistem (tester).

c. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object,danrelationship) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu.

View ini digambarkan dalam class diagrams untuk struktur statis dan dalam state, sequence, collaboration, dan activity diagram untuk model dinamisnya. View ini digunakan untuk perancang (designer) dan pengembang (developer).

d. Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya.

View ini digambarkan dalam component view dan digunakan untuk pengembang (developer).

e. Concurrency view

Membagi sistem ke dalam proses dan prosesor.View ini digambarkan dalam diagram dinamis (state, sequence, collaboration, dan activity diagrams) dan diagram implementasi (component dan deployment diagrams) serta digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

f. Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (nodes) dan bagaimana hubungannya dengan lainnya.

View ini digambarkan dalam deployment diagramsdan digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

g. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

1. Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Use casemerupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2. Class Diagram

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system.

3. Component Diagram

Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis.

Komponent merupakan implementasi software dari sebuah atau lebih class. Komponent dapat berupa source code, komponent biner, atau executable component. Sebuah komponent berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view.Sehingga component diagram merepresentasikan dunia riil yaitu component software yang mengandung component, interface dan relationship.

4. Deployment Diagram

Menggambarkan tata letak sebuah system secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes,executeable component dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

5. State Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh stateyang berbeda.

6. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antaraobject, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

7. Collaboration Diagram

Menggambarkan kolaborasi dinamis sepertisequence diagrams. Dalam menunjukkan pertukaran pesan, collaboration

diagrams menggambarkan objectdan hubungannya (mengacu ke konteks).Jika penekannya pada waktu atau urutan gunakansequencediagrams, tapi jika penekanannya pada konteks gunakan collaboration diagram.

8. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use caseatau interaksi.