# BAB 2

# LANDASAN TEORI

## 2.1 Game

*Game* merupakan kata dalam bahasa Inggris yang berarti permainan. *Game* adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing*. Suatu cara belajar yang digunakan dalam menganalisa interaksiantara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional. Teori *game* pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944.

Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi : "*Game* terdiri atas sekumpulan peraturan yang membangun situasibersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan.

Peraturan-peraturan menentukan kemungkinantindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.“(*Theory of Games and Economic Behavior* (*3d ed.* 1953)”[2].

### 2.1.1 Jenis-Jenis Game

Berikut ini akan dijabarkan beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya, mesin yang menjalankannya. Jenis *game* diantaranya adalah [3]:

1. *Game* PC

   *Game* yang dimainkan pada PC (Personal Computer) yang memiliki kelebihan yaitu memiliki tampilan antarmuka yang baik untuk input maupun output, output visual kualitas tinggi karena layar komputer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televise biasa.

2. *Game* Console

   *Game* yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah pribadi, seperti Microsoft Xbox 360, Nintendo Wii dan sebagainya.

3. *Game* Arcade

   *Game* yang dijalankan pada mesin dengan input dan output audio visual yang telah terintegrasi dan tersedia ditempat-tempat umum, seperti mal, bandara dan sebagainya.

4. *Game* Online

   *Game* yang hanya dapat dimainkan secara online melalui LAN atau internet.

### 2.1.2 Genre Game

*Game* dibagi menjadi beberapa genre, diantaranya yaitu :

1. *Action Shooting* (tembak- menembak)

   Menembak , memukul , bisa juga tusuk-tusukan, tergantung cerita dan tokoh di dalamnya, video *game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari *game* jenis ini adalah tembak, tembak dan tembak.

   Contoh : GTA dan Crysis.

2. *Fighting* (pertarungan)

   Ada yang mengelompokan video *game* fighting di bagian Aksi, namun penulis berpendapat berbeda, jenis ini memang memerlukan kecepatan refleks dan koordinasi mata dan tangan, tetapi inti dari *game* ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya), pengenalan karakter dan timing sangatlah penting, iya, combo-pun menjadi esensial untuk mengalahkan lawan secepat mungkin.

   Contoh : Mortal Kombat dan Tekken.

3. *Adventure* (Petualangan)

Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris, atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya,

itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam video *game* jenis ini.

Contoh : Kings Quest, dan Space Quest.

4. *Strategy* (strategi)

   Video *game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusal pelatihan tempur, tergantung dari tema ceritanya. Kebanyakan *game* stategi adalah *game* perang. Contoh : Warcraft.

5. *Simulation* (Simulasi)

   *Video Game* jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. *Video Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas.

6. *Puzzle* (menyusun)

   Video *game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula *game* jenis ini adalah juga unsur *game* dalam video *game* petualangan maupun *game* edukasi

7. *Sport game* (Olahraga)

   *Game* ini merupakan adaptasi dari kenyataan, membutuhkan kelincahan dan juga strategi dalam memainkannya. *Game* berupa kompetisi antara dua pemain atau lebih, di mana pemain dapat berupa individual atau tim. Contoh *game* tipe ini antara lain sepakbola, bola basket, tenis, dan bilyard.

8. *RPG (Role Playing Game)*

   *Video game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam *game*, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan.

9. *Education* (edukasi)

   *Game* edukasi merupakan paket software yang menciptakan kemampuan pada lingkungan *game* yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur *game* secara teliti untuk mengembangkan kemampuannya. Developer yang membuatnya, harus memperhitungkan berbagai hal agar *game* ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan design visual ataupun animasinya [3].

## 2.2 Game Edukasi

*Game* edukasi adalah adalah *game* yang khusus dirancang untuk mengajarkan user suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya, Hurd dan Jenuings (2009:6). *Game* jenis ini mengajarkan anak-anak atau pun orang dewasa beberapa bentuk pembelajaran dengan cepat dan mudah diserap. Sementara pembelajaran di sekolah konvensional dirasa lambat dan membosankan.

Penerapan education *game* bermula dari perkembangan video *game* yang sangat pesat dan menjadikannya sebagai media efektif yang interaktif dan banyak dikembangkan di perindustrian. Melihat kepopuleran *game* tersebut, para pendidik berpikir bahwa mereka mempunyai kesempatan yang baik untuk menggunakan

komponen rancangan *game* dan menerapkannya pada kurikulum dengan penggunaan industri berbasis *game*. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan, Hurd dan Jenuings (2009:4). Salah satu contoh *game* edukasi yang dikembangkan oleh mahasiswa ialah*game* Brain Racer.[12].

### 2.2.1 Kriteria Game Edukasi

Menurut Hurd dan Jenuings (2009:5), perancangan Education *game* yangbaik haruslah memenuhi kriteria dari education *game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah education *game*, yaitu[12]:

1. Nilai Keseluruhan (Overall Value)
   Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur timer.
2. Dapat Digunakan (Usability)
   Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Apliksi ini merancang sistem dengan interface yang user friendly sehingga user dengan mudah dapat mengakses aplikasi.
3. Keakuratan (Accuracy)
   Keakuratan diartikan sebagai bagaimana kesuksesan model atau gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (Appropriateness)

   Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapatdiadaptasikan terhadap keperluan user dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan user untuk membantu pemahaman user dalam menggunakan aplikasi.

5. Relevan (Relevance)

   Relevan artinya dapat mengaplikasikan isi *game* ke target user. Agar dapat

   relevan terhadap user, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektifitas (Objectives)

   Objektifitas menentukan tujuan user dan kriteria dari kesuksesan atau kegagalan. Dalam aplikas ini objektivitas adalah usaha untuk mempelajari hasil dari *game*.

7. Umpan Balik (Feedback)

   Untuk membantu pemahaman user bahwa *game* (performance) mereka sesuai dengan objek *game* atau tidak, feedback harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan *game*.

#### 2.2.1.1 Hubungan Konsep Game Edukasi Dengan Belajar

*Game* komputer merupakan media yang dekat dengan anak, dengan media *game* komputer diharapkan tidak hanya bermain saja melainkan anak dapat belajar.Belajar tidak hanya dapat dilakukan di sekolah saja, namun dapat dilakukan dimana-mana, seperti di rumah ataupun dilingkungan masyarakat. Irwanto (1997:105) berpendapat bahwa belajar merupakan proses perubahan dari belum mampu menjadi sudah mampu dan terjadi dalam jangka waktu tertentu. Sedangkan menurut Mudzakir (1997:34) belajar adalah suatu usaha atau kegiatan yang bertujuan mengadakan perubahan di dalam diri seseorang, mencakup perubahan tingkah laku, sikap, kebiasaan, ilmu pengetahuan, keterampilan dan sebagainya.

Kelebihan *Game* Edukasi dibanding *game* lainya adalah:

1. Tidak membuat anak malas belajar karena efek ketergantungan terhadap *game* sehingga lupa akan belajar.
2. Dapat mendorong siswa untuk meningkatkan pengetahuan dan kemampuan mengenai komputer.
3. Meningkatkan motivasi belajar anak.
4. Menyesuaikan materi dengan kemampuan anak.
5. Memberika hiburan pada anak ketika jenuh saat belajar secara konvensional.
6. Mengasah syaraf motorik dan sensorik pada anak.

## 2.3 Algoritma Fisher-Yates

Shuffle Fisher-Yates (bernama setelah Ronald Fisher dan Frank Yates ), juga dikenal sebagai shuffle Knuth (setelah Donald Knuth ), adalah sebuah algoritma untuk menghasilkan permutasi acak dariseperangkat terbatas dalam hal biasa, untuk secara acak mengocok set. Sebuah varian dari shuffle Fisher-Yates, yang dikenal sebagai algoritma Sattolo itu, dapat digunakan untuk menghasilkan acak siklus panjang *n* sebagai gantinya. Benar dilaksanakan, shuffle Fisher-Yates adalah berisi , sehingga setiap permutasi adalah kemungkinan yang sama. Versi modern dari algoritma ini adalah juga agak efisien, hanya membutuhkan waktu sebanding dengan jumlah barang yang dikocok dan tidak ada ruang penyimpanan tambahan.

Proses dasar dari Fisher-Yates menyeret mirip dengan memilih secara acak tiket bernomor keluar dari cab, atau kartu dari setumpuk, satu demi satu sampai tidak ada yang lebih kiri. Apa algoritma tertentu adalah menyediakan cara untuk melakukan ini secara numerik dengan cara yang efisien dan ketat itu, benar dilakukan, menjamin hasil yang bias.

Shuffle Fisher-Yates, dalam bentuk aslinya, digambarkan pada tahun 1938 oleh Ronald A.Fisher dan Frank Yates, dalam tabel buku statistik mereka untuk penelitian biologi, pertanian dan medis. (Kemudian edisi menjelaskan sebuah metode yang agak berbeda dikaitkan dengan CR Rao ). Metode mereka. didesain untuk dapat diimplementasikan dengan menggunakan pensil dan kertas, dengan meja precomputed angka acak sebagai sumber keacakan.

Metode dasar yang diberikan untuk menghasilkan permutasi acak dari angka 1 – *N* berjalan sebagai berikut:

Tuliskan nomor dari 1 sampai *N*.

Pilih bilangan acak *k* antara satu dan jumlah nomor unstruck tersisa (inklusif).

Dihitung dari low end, mencoret nomor *k* belum menyerang, dan menuliskannya di tempat lain.

Ulangi dari langkah 2 sampai semua angka telah menyerang.

Urutan nomor ditulis pada langkah 3 sekarang menjadi permutasi acak dari nomor asli.

Asalkan nomor acak memilih pada langkah 2 di atas adalah benar-benar acak dan tidak bias, sehingga akan dihasilkan menjadi permutasi. Fisher dan Yates Andrea untuk menggambarkan bagaimana mendapatkan angka acak tersebut dalam setiap rentang yang dikehendaki dari tabel yang diberikan dengan cara yang menghindari bias. Mereka juga menyarankan kemungkinan menggunakan metode sederhana - mengambil angka acak dari satu sampai *N* dan membuang setiap duplikat-untuk menghasilkan setengah pertama dari permutasi, dan hanya menerapkan algoritma yang lebih kompleks untuk babak yang tersisa, di mana memilih sejumlah duplikat akan jika tidak menjadi putus asa umum.[13].



Gambar 2. 1 Zynga Poker mengocok dek [13]

## 2.4 Perlombaan 17 Agustus

Hari peringatan Kemerdekaan Republik Indonesia, bertepatan tanggal 17 Agustus 1945, Pendiri bangsa Indonesia Soekarno-Hatta yang membacakan proklamasi tanda kemerdekaan bangsa Indonesia dari kolonialisme Belanda. Sekalipun Belanda belum mengakui kemerdekaan Indonesia pada tanggal 17 Agustus 1945 tapi sebagai bangsa yang sudah merdeka yaitu bangsa Indonesia, hal itu tidaklah menjadi halangan untuk menyatakan kemerdekaan. Biasanya tanggal 17 agustus yang masyarakat indonesia mengisi berbagai macam lomba-lomba yang mengikutsertakan anak-anak hingga orang dewasa untuk memeriahkan suasana tujuhbelasan yang sakral.

Kebanyakan dari kita, masyarakat Indonesia, sampai sekarang mungkin masih ada yang belum mengerti makna dari perlombaan 17 Agustusan, kita hanya menganggap bahwa perlombaan itu hanya sebuah *game* semata, marilah kita bersama-sama mencoba untuk lebih mengetahui dan mengenali makna dari perlombaan 17 Agustusan ini.

Tujuan dari perlombaan 17 adalah untuk selain menghibur juga mempererat kekerabatan antar anggota masyarakat serta menjujung tinggi nilai sportifitas dan juga menjunjung nilai kecintaan terhadap bernegara kita. Semua orang semua gigih menghadapi setiap pertandingan untuk mendapatkan sebuah hasil kemenangan, mencontoh para pejuang yang gigih memperjuangkan agar bangsa kita ini bangsa indonesia menjadi bangsa yang mampu berdiri dan bersaing dengan negara-negara lain sehingga dapat mendapatkan pengakuan secara mendunia.

**”Keberhasilan para pendahulu dalam mencapai kemerdekaan Indonesia bukanlah akhir dari sebuah perjuangan. Mari kita hias kemerdekaan ini dengan pembangunan dan kesejahteraan bagi rakyat Indonesia”.**

Berikut ini merupakan daftar nama beberapa lomba yang paling sering diperlombakan dalam acara perlombaan tujubelasan di indonesia :

a. Lomba menangkap ayam / bebek
b. Lomba balap karung
c. Lomba balap kelereng
d. Lomba balap makan kerupuk
e. Lomba memasukkan benang ke dalam jarum
f. Lomba panjat pinang
g. Lomba memasukkan pensil ke dalam botol
h. Lomba gebuk bantal
i. Lomba gigit duit
j. Lomba tarik tambang
k. Lomba rebutan kursi
l. Lomba sepak bola pria memakai daster
m. Lomba giring bola dengan terong
n. Lomba balap bakiak panjang
o. Lomba balap gendong
p. Lomba rias wajah dengan mata ditutup

### 2.4.1 Lomba Menangkap Ayam

Lomba menangkap hewan (ayam, belut, ikan, bebek) adalah pertandingan kompetisi dengan beberapa peserta. Para peserta berlomba untuk mengejar hewan (ayam, belut, ikan, bebek), dan pemenangnya adalah peserta yang paling cepat dalam menangkap hewan tersebut. Tantangan dari lomba ini adalah peserta harus terjatuh, basah basahan, beradu lari dengan peserta lainya. Biasanya jaman dahulu hewan disimbolkan sebagai pemimpin koloni Belanda dan sekarang disimbolkan sebagai koruptor.




Gambar 2. 2 Perlombaan Menangkap Hewan [4]

Makna perlombaan menangkap ayam ini adalah semangat untuk menangkap hewan di simbolkan sebagai penguasa (Penjajah dan Koruptor) yang selalau menindas, merusak,merampas hasil bumi dan kerja keras rakyat indonesia yang mengakibatkan banyak rakyat yang kelaparan. Sebagai bentuk pelampiasan hewan menjadi simbol tangkap para koruptor dan penguasa yang mementingkan kehidupannya pribadinya sendiri.

### 2.4.2 Lomba Balap Karung

Balap karung adalah salah satu lomba tradisional yang populer pada hari kemerdekaan Indonesia. Sejumlah peserta diwajibkan memasukkan bagian bawah badannya ke dalam karung kemudian berlomba sampai ke garis akhir. Meskipun sering mendapat kritikan karena dianggap memacu semangat persaingan yang tidak sehat dan sebagai kegiatan hura-hura, balap karung tetap banyak ditemui, seperti juga lomba panjat pinang, sandal bakiak, dan makan kerupuk. Lomba balap karung juga diapresiasi oleh pendatang dari luar negeri dengan langsung terlibat dalam perlombaan ini.



Gambar 2. 3 Perlombaan Balap Karung [4]

Makna perlombaan balap karung : Saat penjajahan, sebagian besar rakyat mengalami penderitaan sangat berat. Bahan pakaian diambil kaum penjajah, yang tertinggal adalah plastik, karet, dan karung. Mau tidak mau, rakyat hanya mengenakan pakaian berasal dari karung goni. Kain yang berserat kasar tersebut menimbulkan gatal-gatal di kulit sehingga saat tibanya kemerdekaan disambut

rakyat dapat berpakaian layak kembali. Sebagai bentuk pelampiasan kekesalan terhadap penggunaan kain berbahan karung, maka rakyat menginjak-injak karung. Namun makna lain dari balap karung adalah, betapa sulitnya berlari ketika kedua kaki terkungkung. Sejauh apapun "melompat" tetap akan mengalami kesulitan akibat dihalang-halangi. Terkadang bisa saja terjerambab. Hal itu pula yang menyulitkan kemajuan bangsa Indonesia dalam kungkungan penjajah.

## 2.5 Aplikasi Pembangun Perangkat Lunak

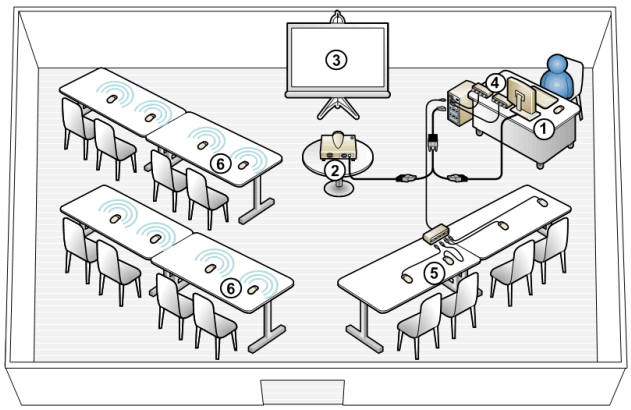
### 2.5.1 Windows Multipoint Mouse SDK

Microsoft MultiPoint SDK merupakan sebuah framework yang menyediakan lingkungan pengembangan dimana para developer dapat mengembangkan aplikasi yang memungkinkan hingga 25 mouse secara simultan bekerja dalam satu komputer. Multipoint SDK terutama digunakan untuk membangun aplikasi edukasi untuk sekolah-sekolah yang memiliki keterbatasan dalam hal infrastruktur teknologi sehingga makin banyak siswa-siswa yang dapat bersentuhan dengan komputer. Pilot program ini sudah diujicobakan di India oleh Microsoft Research menunjukkan manfaat dari teknologi ini untuk beberapa subjek, teknologi pembelajaran kolaboratif seperti Mutlipoint meningkatkan proses pembelajaran jika dibandingkan dengan skenario satu siswa satu komputer[7].




Gambar 2. 4 Logo Multipoint SDK [7]

Menurut Penelitian Microsoft dalam kelas besar, tidak semua siswa berpartisipasi dalam diskusi kelas. Hal ini membuat sulit bagi guru untuk secara akurat mengukur apakah siswa memahami materi yang disajikan. MultiPoint menyajikan kesempatan unik bagi para guru untuk secara berkala jajak pendapat siswa untuk mengetahui tingkat pemahaman pada waktu tertentu. Jajak pendapat itu dapat mengambil bentuk kuis di mana jawaban yang benar diberikan poin. Skenario penggunaan di atas menyajikan contoh aplikasi MultiPoint yang Anda dapat membangun untuk membantu siswa bersenang-senang dan belajar sambil memanfaatkan sumber daya komputasi yang tersedia bagi mereka.




Gambar 2. 5 Skenario Pengunaan Multipoint [7]

### 2.5.2 Microsoft Visual C# 2010 Express

Microsoft Visual Studio Express adalah kumpulan freeware lingkungan pengembangan terpadu (IDE) yang dikembangkan oleh Microsoft yang merupakan versi ringan dari Microsoft Visual Studio. Gagasan edisi ekspres menurut Microsoft adalah untuk menyediakan efisiensi, mudah digunakan dan mudah dipelajari bagi pengguna IDE selain pengembang perangkat lunak profesional, yaitu seperti penggemar dan mahasiswa.

Sejalan dengan semakin populer sejak Visual Studio 2005 Express Editions, edisi ini akan selalu bebas biaya. Visual Studio 2008 Express Editions memerlukan Windows XP atau versi Windows berikutnya; Windows NT 4.0, Windows 2000 dan Windows 9x tidak lagi didukung. Visual Studio 2005 Express Editions dapat diinstal pada Windows 2000 SP4. Pada akhir April 2009,dan yang paling terbaru adalah Visual Studio 2010 Expres yang dapat mendukung sistem operasi windows xp, vista, 7 (*Seven*).[8]




Gambar 2. 6 Logo Microsoft Visual Studio 2010 Express [8]

### 2.5.3 Pemrograman C#

C# (dibaca : C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif

kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Specification, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar. Gambar 2.7 merupakan gambar logo C#.[9]




Gambar 2. 7 Logo C# (dibaca: C sharp) [9]

### 2.5.4 WPF (Windows Presentation Foundation)

WPF (Windows Presentation Foundation) adalah teknologi baru yang dirilis oleh Microsoft bersamaan dengan .NET Framework 3.0 (versi sekarang adalah 3.5). Selain WPF .NET 3.0 juga menyertakan beberapa teknologi baru yaitu WF (Workflow Foundation) untuk pemodelan dan WCF (Windows Communication Foundation) yang merupakan pengembangan dari Web Services.

WPF merupakan next generation graphic platform yang memungkinkan anda membuat advance UI (User Interface) yang memadukan document, media, 2D dan 3D, dan animation yang memiliki kemiripan dengan pemrograman web.WPF dapat digunakan pada platform windows seperti Windows XP, Vista, 7 (*seven*) dan Windows Server 2007 / 2010.

Sebelum ada teknologi WPF untuk mengembangkan aplikasi windows anda harus menggunakan beberapa teknologi yang berbeda, misalnya untuk menambahkan form user control seperti button kedalam aplikasi anda harus menggunakan komponen Windows Form, untuk menambahkan 2D Graphic anda harus menggunakan library GDI+, untuk 3D Graphic anda harus menggunakan library DirectX atau OpenGL.

WPF didesain untuk menyediakan solusi yang utuh untuk pengembangan aplikasi, WPF menyediakan integrasi dari teknologi yang berbeda, dengan WPF anda dapat menggabungkan komponen vector graphic, complex animation, dan media kedalam aplikasi. Fitur yang baru dalam WPF (.NET 3.5) adalah fitur 3D Graphic yang sudah built-in berbasis pada teknologi DirectX.

### 2.5.5 XAML (Extensible Application Markup Language)

WPF menggunakan XAML untuk merepresentasikan UI dan user interaction. XAML dibangun dari XML dan penulisanya mirip dengan XHTML yang sering anda gunakan dalam membuat aplikasi berbasis web. XAML jugamendukung prinsip code-behind seperti pada aplikasi ASP.NET. Dengan menggunakan XAML, designer dan programmer dapat bekerja bersama dalam

mengembangkan aplikasi, pertama designer membuat design UI kemudian programmer melanjutkan dengan menambah kodenya dalam bahasa yang disupport oleh .NET seperti C# / VB.NET.




Gambar 2. 8 Logo XAML

### 2.5.6 XML (eXtensible Markup Language)

XML adalah singkatan dari eXtensible Markup Language. Merupakan pengembangan dari HTML dan penyederhanaan dari SGML (standard generalized markup language). Banyak digunakan dalam information exchange karena sifatnya yang universal. Direkomendasikan oleh W3C (World Wide Web Consortium) pada tahun 1998.

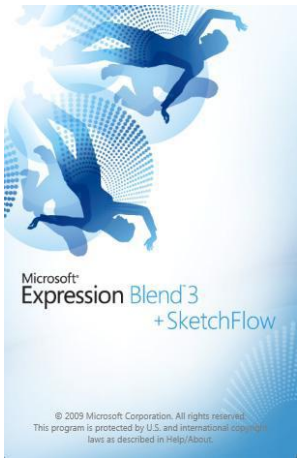
Keunggulan dari XML adalah sebagai berikut :

1. Sederhana strukturnya
2. Universalitas – software maupun hardware
3. Extensible
4. Scalable – mampu memisahkan data dan design
5. Integritas

### 2.5.7 Microsoft Expression Blend

Microsoft Expression Blend adalah tools untuk membuat user interface sebuah aplikasi berbasis Windows ataupun Silverlight. Blend bisa juga dikolaborasikan dengan Microsoft Expression Design, sebuah tools untuk membuat image dan lain-lain. Sehingga bisa dihasilkan sebuah "cutting-edge" user interfaces.

Alat ini ditujukan agar designer dapat berkolaborasi dengan developer dalam suatu proyek pembuatan aplikasi. Menggunakan "bahasa" yang sama. Format yang dihasilkan dapat digunakan oleh developer yang menggunakan Visual Studio. Bahkan dalam Blend, kita bisa melakukan test project aplikasi yang dibuat.




Gambar 2. 9 Microsoft Exprsion Blend

## 2.6 Analisis Sistem Berorientasi Objek

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunkan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara

struktur data dab perilaku dalam suatu entitas. Model berorientasi objek bermanfaat untuk memahami masalah, komunikasi dengan ahli aplikasi, pemodelan suatu organisasi, meyiapkan dokumentasi serta perancangan program dan basis data. Pertama-tama suatu model analisis dibuat untuk menggambarkan aspek dasar dari domain aplikasi, dimana model tersebut berisi objek yang terdapat dalam domain aplikasi termasuk deskripsi dari keterangan objek dan perilakunya.

### 2.6.1 OOP (Object Oriented Programing)

Pemrograman berorientasi objek (*object-oriented programming* disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar.

Konsep dasar dari Pemrograman Berorientasi Objek menekankan konsep berikut:

1. Class - kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode

yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP.

2. Objek - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. Abstrak - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
4. Enkapsulasi - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. Polimorfisme - melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim.
6. Inheritas (Inheritance)- Mengatur polimorfisme dan enkapsulasi denganmengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dariobjek yang sudah ada objek-objek ini dapat membagi dan memperluasperilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut(bahasa berbasis-objek tidak selalu memiliki inheritas).

### 2.6.2 UML (Unified Modeling Language)

UML (Unified Modeling Language) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun system (Flowler, 2006). Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya.

UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.




Gambar 2. 10 Logo UML [5]

UML suatu bahasa yang digunakan untuk menentukan,memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi.UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasiobjek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikianUML dapat digunakan untuk memahami dan mendokumentasikan setiap sisteminformasi. Penggunaan UML dalam industri terus meningkat, ini merupakanstandar terbuka yang menjadikannya sebagai bahasa pemodelan yang umumdalam industri peranti lunak dan pengembangan sistem.

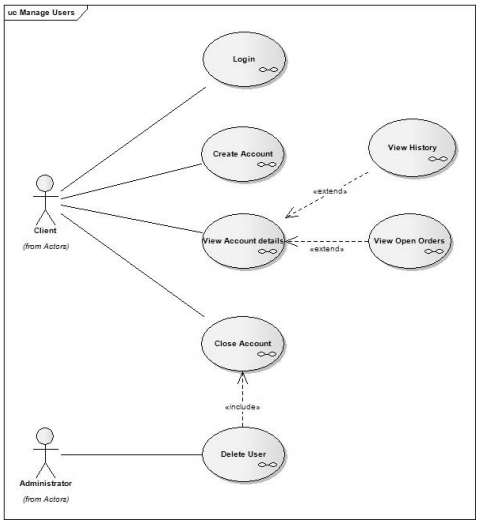
#### 2.6.2.1 Bagian – Bagian UML

Bagian-bagian utama dari UML adalah view, diagram, model element, dan general mechanism [5].

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

a) *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 case merupakan 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.

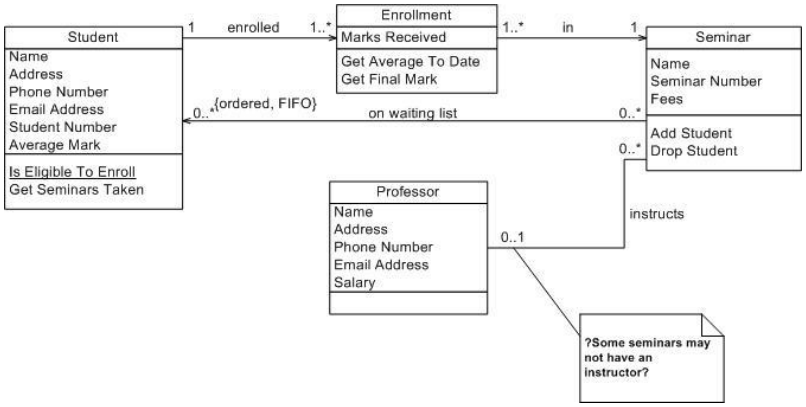



Gambar 2. 11 *Use Case*

b) *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.

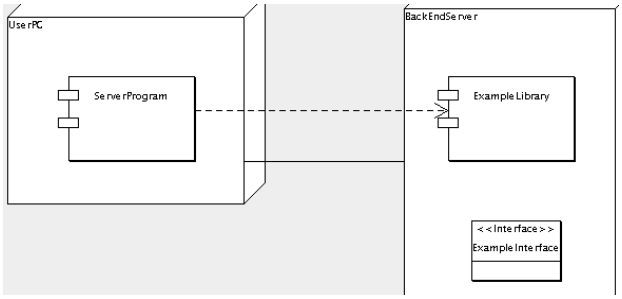
Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system. Class diagram dapat dilihat pada Gambar 2.12




Gambar 2. 12 Logo *Class Diagaram*

c) Component Diagram

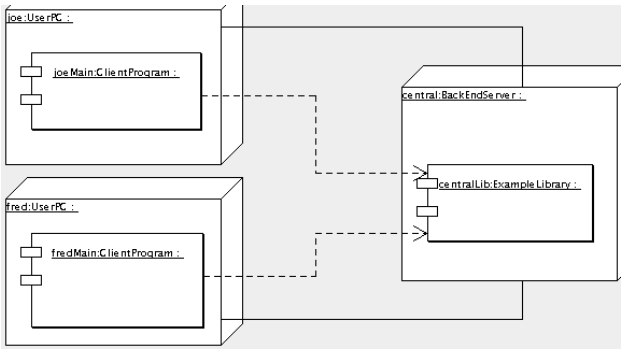
Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis. Komponent merupakan implementasi software dari sebuah atau lebih class. Komponent dapat berupa source code, komponent biner, atau executable component. Sebuah komponent berisi informasi 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. Component diagram dapat dilihat pada Gambar 2.13.




Gambar 2. 13 *Component Diagram*

d) 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, executable component dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

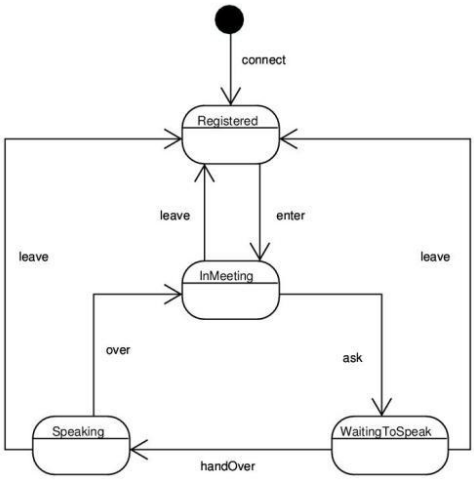



Gambar 2. 14 *Deployment Diagram*

e) 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

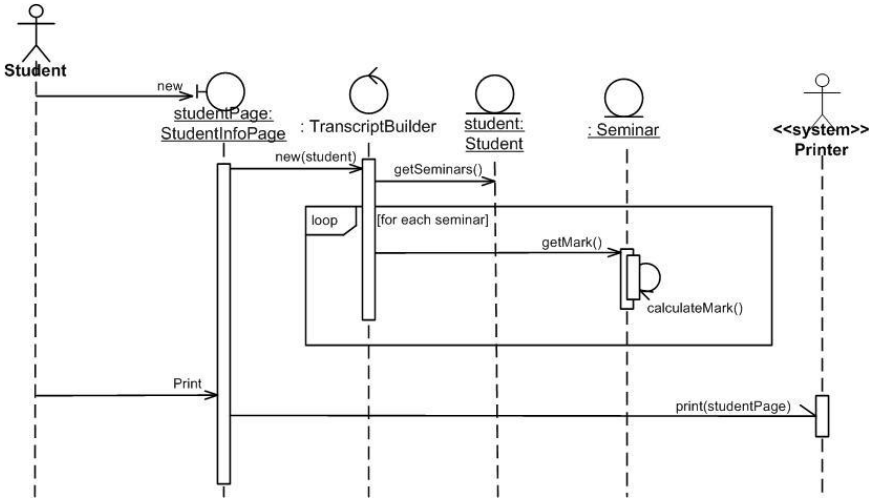
digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.




Gambar 2. 15 *State Diagram*

f) Sequence Diagram

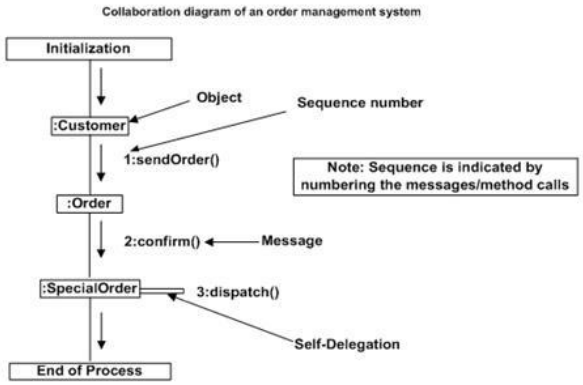
Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.




Gambar 2. 16 *Sequence Diagram*

g) Collaboration Diagram

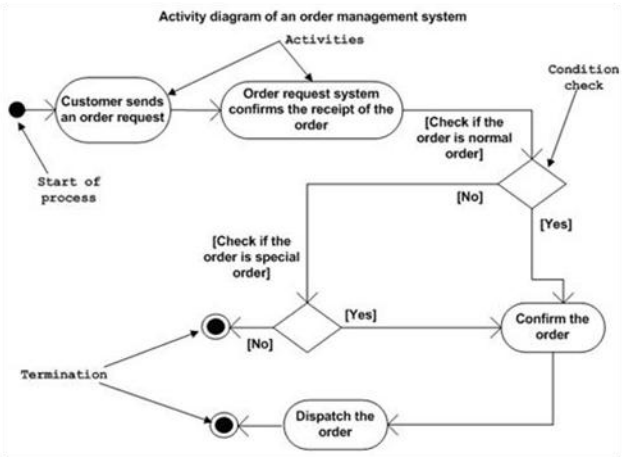
Menggambarkan kolaborasi dinamis seperti sequence diagram. Dalam menunjukkan pertukaran pesan, collaboration diagrams menggambarkan object dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan sequencediagrams, tapi jika penekanannya pada konteks gunakan collaboration diagram. Collaboration diagram dapat dilihat pada Gambar 2.17




Gambar 2. 17 *Colaboration Diagram*

a) 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 case atauv interaksi.




Gambar 2. 18 *Activityn Diagram*

## 2.7 Metode Pengujian

Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak,mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan.

Perangkat lunak dapat diuji dengan dua cara, yaitu :

1. Pengujian dengan menggunakan data uji untuk menguji semua elemen program (data internal, loop, logika, keputusan dan jalur). Data uji

dibangkitkan dengan mengetahui struktur internal (kode sumber) dari perangkat lunak.

2. Pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional perangkat lunak bekerja dengan baik. Data uji dibangkitkan dari spesifikasi perangkat lunak.

### 2.7.1 Pengujian Black Box

Test case ini bertujuan untuk menunjukkan fungsi PL tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya.

Pengujian black-box berfokus pada persyaratan fungsional PL. Pengujian inimemungkinkan analis system memperoleh kumpulan kondisi input yg akan mengerjakan seluruh keperluan fungsional program.

Tujuan metode ini mencari kesalaman pada [13]:

1. Fungsi yg salah atau hilang
2. Kesalahan pada interface
3. Kesalahan pada struktur data atau akses database
4. Kesalahan performansi
5. Kesalahan inisialisasi dan tujuan akhir
6. Metode ini tidak terfokus pada struktur kontrol seperti pengujian whitebox tetapi pada domain informasi.

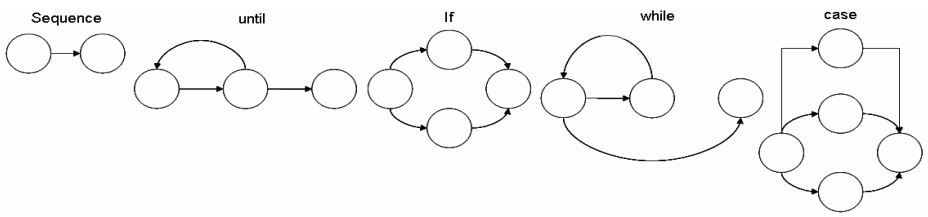
### 2.7.2 Pengujian White Box

*White Box Testing* atau pengujian *glass box* adalah metode desain *test case* menggunakan struktur kontrol desain prosedural untuk mendapatkan *test case.* Dengan menggunakan metode *White Box* analisis sistem akan memperoleh Test Case sebagai berikut:

a. Menjamin seluruh independent path di dalam modul yang dikerjakan sekurang-kurangnya sekali
b. Mengerjakan seluruh keputusan logikal
c. Mengerjakan seluruh loop yang sesuai dengan batasannya
d. Mengerjakan seluruh struktur data internal yang menjamin validitas.

**1. Uji Coba Basis Path**

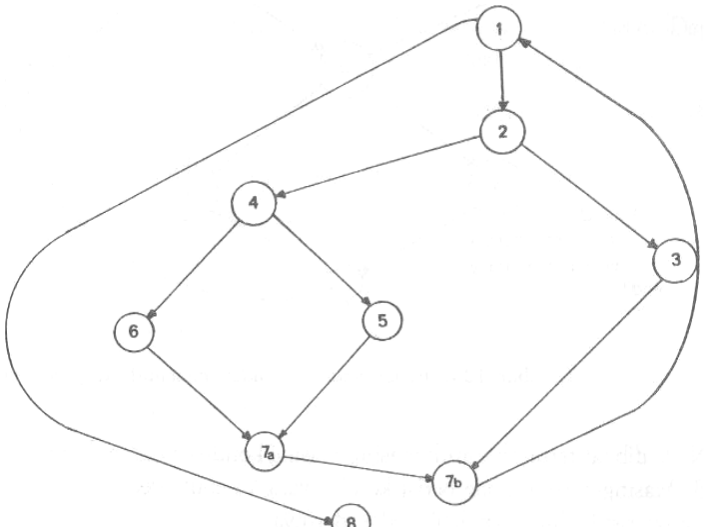
Merupakan teknik uji coba *white box* yang diusulkan Tom McCabe. Metode ini memungkinkan perancang test case mendapatkan ukuran kekompleksan logical dari perancangan prosedural dan menggunakan ukuran ini sebagai etunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Test case yang didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba.




Gambar 019. Notasi Diagram Alir

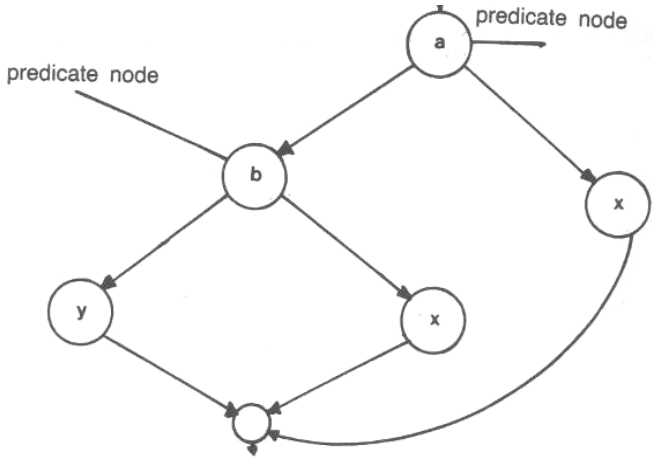
Lingkaran (node), menggambarkan satu/lebih perintah prosedural. Urutan proses dan keputusan dapat dipetakan dalam satu node. Tanda panah (edge), menggambarkan aliran kontrol. Setiap node harus mempunyai tujuan node. Region adalah daerah yang dibatasi oleh edge dan node. Termasuk daerah diluar grafik alir. Contoh menterjemahkan pseudo code ke grafik alir akan ditunjukan pada gambar 2.20

```
1 : do while record masih ada
baca record
2 : if record ke 1 = 0
3 : then proses record
simpan di buffer
naikan counter
4 : else if record ke 2 = 0
5 : then riser counter
6 : proses record
simpan pada file
7a: endif
endif
7b: end do
    8 : end
```




Gambar 2.20. Diagram Alir

Nomor pada pseudo code berhubungan dengan nomor node. Apabila ditemukan kondisi majemuk (compound condition) pada pseudo code pembuatan grafik alir menjadi rumit. Kondisi majemuk mungkin terjadi pada operator Boolean (AND, OR, NAND, NOR) yang dipakai pada perintah if. Berikut contoh gambar akan ditunjukan padagambar 2.21 berikut




Gambar 2.21 Logika gabungan

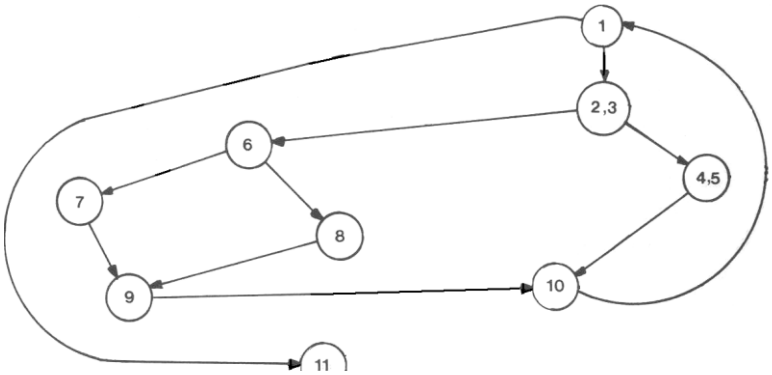
```
if A or B
then procedure x
else procedure y
endif
```

Node dibuat terpisah untuk masing-masing kondisi A dan B dari pernyataan IF A OR B. Masing-masing node berisi kondisi yang disebut pridicate node dan mempunyai karakteristik dua atau lebih edge darinya.

**2. Cyclomatic Complexity**

Cyclomatic complexity adalah metrik software yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba basis path, nilai yang dihitung untuk cyclomatic complexity menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.

Jalur independent adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.




Gambar 2.22 Contoh *Flowgraph*

Path 1 = 1 - 11

Path 2 = 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

Path 3 = 1 - 2 - 3 - 6 - 8 - 9 ...: 10 - 1 - 11

Path 4 = 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis set untuk diagram alir.

Cyclomatic complexity digunakan untuk mencari jumlah path dalam satu flowgraph. Dapat dipergunakan rumusan sebagai berikut：

1. Jumlah region grafik alir sesuai dengan cyclomatic complexity.

2. Cyclomatix complexity V(G) untuk grafik alir dihitung dengan rumus:

V(G) = E - N + 2

Dimana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

3. Cyclomatix complexity V(G) juga dapat dihitung dengan rumus:

V(G) = P + 1

Dimana P = jumlah predicate node pada grafik alir

Pada Gambar 2.5 dapat dihitung cyclomatic complexity:

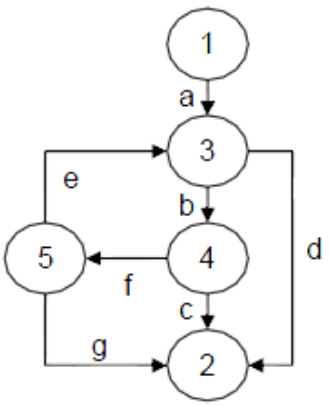
1. Flowgraph mempunyai 4 region
2. V(G) = 11 edge - 9 node + 2 = 4
3. V(G) = 3 predicate node + 1 = 4

Jadi cyclomatic complexity untuk flowgraph adalah 4.

### 3. Graph Metrik

Graph metrik merupakan software yang dikembangkan untuk membantu uji coba basis path atau struktur data. Graph metrik adalah matrik empat persegi yang mempunyai ukuran yang sama dengan jumlah node pada *flowgraph*. Masing-masing baris dan kolom mempunyai hubungan dengan node yang telah ditentukan dan pemasukan data matrik berhubungan dengan hubungan (*edge*) antar *node*.

Berikut ini adalah contoh sederhana pemakaian graph metrik dapat digambarkan pada gambar 2.6.




| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | a | | |
| 2 | | | | | |
| 3 | | | | b | |
| 4 | | c | | | f |
| 5 | | g | e | | |

Gambar 2.23 Penggunaan Graph Metrik

Hubungan bobot menyediakan tambahan informasi tentang aliran kontrol. Secara simpel hubungan bobot dapat diberi nilai 1 jika ada hubungan antara node atau nilai 0 jika tidak ada hubungan. Dapat juga hubungan bobot diberi tanda dengan:

a. Kemungkinan link (edge) dikerjakan
b. Waktu yang digunakan untuk proses selama traversal dari link
c. Memori yang diperlukan selama traversal link
d. Sumber daya yang diperlukan selama traversal link