# Metode Pencarian Minimax untuk Permainan Nim

Oleh :

Rian Herdiani

10106414

# PENDAHULUAN (1)

**Latar Belakang**

- Seiring dengan semakin pesatnya kemajuan teknologi, industri hiburan kini mulai diminati. Salah satu yang paling banyak diminati adalah industri *games*. Banyaknya *game-game* baru yang bermunculan menandai semakin banyaknya peminat akan *game* itu sendiri. *Game* yang mendapat perhatian besar adalah *games* berbasis komputer.
- komputer dirancang dengan menggunakan *Artificial Intelligince*. Dengan diterapkannya AI pada komputer maka, komputer dapat mengambil langkah-langkah yang tepat yang dapat memberikan keuntungan pada komputer dan merugikan pemain manusia ketika bermain.
- minimax merupakan metode yang digunakan untuk menentukan pilihan agar dapat memperkecil kemungkinan kehilangan nilai maksimal.

# PENDAHULUAN (2)

- **Identifikasi Masalah**

  Bagaimana merancang aplikasi permainan nim dengan menerapkan *Artificial Intelligence* menggunakan metode pencarian minimax dan mengimplementasikan hasil rancangan tersebut?

- **Tujuan**

  Untuk merancang suatu aplikasi permainan nim berbasis komputer dengan menerapkan *Artificial Intelligence* menggunakan metode minimax dan mengimplementasikan hasil rancangan

- **Batasan Masalah**

- Aplikasi yang dirancang akan dibuat dengan menggunakan perangkat lunak Delphi 7.
- Implementasi dari hasil rancangan akan ditambahkan statistik jalannya permainan dan rekomendasi langkah terbaik yang dapat diikuti oleh pemain manusia.
- Aplikasi ini tidak terkoneksi dengan jaringan.

# PENDAHULUAN

- **Metodologi Penelitian**

1. Tahap pengumpulan data
   a. Studi Pustaka
   b. Observasi
2. Tahap pembuatan perangkat lunak.
   a. Rekayasa Sistem
   b. Analisis
   c. Desain
   d. Pengkodean
   e. Pemeliharaan
   f. Pengujian

# LANDASAN TEORI

## II.1 Definisi Permainan

*Games* adalah bagian fundamental dari eksistansi manusia. Secara bahasa "*games*" mempunyai makna sindiran yang mengacu pada aktifitas yang bukan "*games*" sesungguhnya. Menurutnya apabila ingin memahami tentang "*games*" dan desain "game" yang harus dilakukan pertama kali ialah menentukan pokok orientasinya. Kemudian menentukan seluruh karakteristik "*game*" yang ada. (Chris Crawford, 1982).

# Landasan teori

**II.1.1 Klasifikasi Permainan**

Berdasarkan jenis *platform*, permainan digolongkan menjadi:

1. *Arcade games*
2. *PC games*
3. *Consule games*
4. *Handheld games*
5. *Mobile games*

Sedangkan berdasarkan *genre* permainannya, dikelompokkan menjadi:

1. Aksi – *Shooting*
2. *Fighting* (pertarungan)
3. Aksi – Petualangan
4. Petualangan
5. Simulasi
6. *Role Playing*
7. *Strategy*
8. *Puzzle*
9. Simulasi kendaraan
10. Olahraga

# Landasan teori

## II.2 Definisi dan Konsep Permainan NIM

Nim merupakan suatu permainan strategi matematika yang dimainkan oleh dua orang pemain. dimana setiap pemain harus memindahkan objek secara bergiliran dari tumpukan yang berbeda.

Nim biasanya dimainkan sebagai *misere game*, dimana pemain yang mengambil objek terakhir maka ialah yang kalah. Nim juga dapat dimainkan secara normal (biasa), yaitu pemain yang mengambil objek terakhir menang.

### II.2.1 Sejarah Permainan Nim

Permainan ini sebenarnya sudah ada sejak zaman dahulu. Kabarnya permaiann ini merupakan permaian dari Cina yang bernama *Tsyanshidzi* atau "mengambi batu". Asal kata "Nim" itu sendiri sebenarnya berasal dari German yaitu *nimm* yang artinya "take" atau "ambil". Di Eropa permainan nim ini sudah ada sejak awal abad ke-16. Salah seorang professor matematika dari Harvard, Charles Bouton, ialah yang mengembangkan teori permainan nim ini.

# Landasan teori

**II.3 *Combinatorial Game Theory***

adalah salah satu teori matematika yang hanya mempelajari permainan dengan dua pemain yang memiliki posisi dimana setiap pemain bermain bergantian untuk memperoleh kemenangan.

Pada umumnya, yang disebut sebagai permainan kombinatorial ialah yang memiliki sayarat seperti :

1. Permainan memiliki tepat 2 pemain.
2. Pada umumnya pemain yang melakukan langkah terakhir ialah yang menang (seperti permainan nim).
3. Pemain bermain secara bergliran.
4. Permianan memiliki akhir, tidak berlangsung terus menerus.
5. Tidak ada informasi yang disembunyikan dari pemain.
6. Permainan tidak berdasarkan pada keberuntungan.

# Landasan teori

**II.4 Definisi *Artificial Intelligence* (AI)**

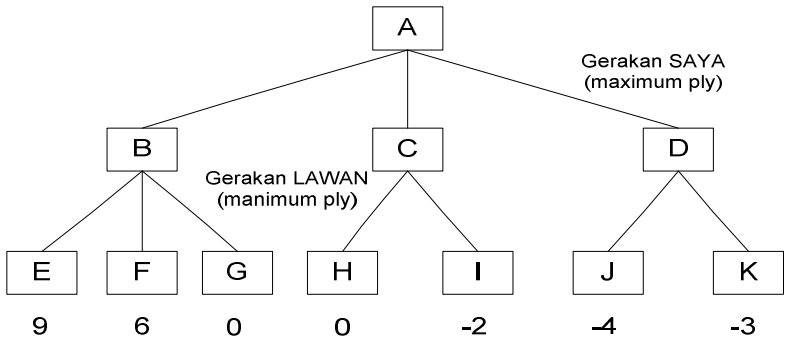
Definisi kecerdasan buatan apabila ditinjau dari berbagai macam sudut:

1. Sudut pandang kecerdasan, Bahwa kecerdasan buatan akan membuat mesin menjadi 'cerdas' atau mampu melakukan seperti yang dilakukan manusia.
2. Sudut pandang penelitian, Kecerdasan buatan merupakan suatu pembelajaran tentang bagaimana membuat komputer agar dapat melakukan sesuatu sebaik yang dikerjakan manusia.
3. Sudut pandang bisnis, Kecerdasan buatan adalah kumpulan peralatan sangat *powerfull* dan metodologis dalam menyelesaikan masalah-masalah bisnis.
4. Sudut pandang pemrograman, Kecerdasn buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

# Landasan teori

**II.5 Minimax**

- Dalam bidang kecerdasan buatan salah satu teknik permainan yang terkenal adalah Minimax. Teorema Minimax (von Neumann, 1928), bahwa:
- *"A strategy tells what an agent will do in every possible situation and strategies may be pure (deterministic) or mixed (probabilistic)"*
- Metode Minimax menggunakan *depth-first search* dengan kedalaman terbatas. Fungsi evaluasi yang digunakan adalah fungsi evaluasi statis, dengan mengasumsikan bahwa lawan akan membuat langkah terbaik yang mungkin djalankan.




Pohon pelacakan minimax

# Landasan teori

## II.6 Borland Delphi

Borland Delphi adalah sebuah alat pegembangan aplikasi-aplikasi untuk sistem operasi Microsoft Windows. Dan sangat mudah digunkan untuk membuat suatu program yang berbasiskan GUI (*Graphical User Interface*) atau *console*.

Beberapa kelebihan yang dapat diambil apabila menggunakan Borland Delphi, antara lain:

- Delphi mendukung Pemrograman Beroientasi Object (*Object Orented Programming*)
- Pengembangan aplikasi secara cepat (RAD)
- Menggunakan bahasa tingkat tinggi
- Hasil dari proses kompilasi berupa sebuah file yang dapat dieksekusi sehingga mempermudah dalam pendistribusian program dan mengurangi banyanya file pendukung
- Delphi menyediakan banyak sekali komponen yang dapat digunakan. Komponen juga dapat bersumber dari pihak ketiga yang biasanya disertai dengan dokumentasi, source code dan lain-lain, yang sifatnya komersil atau free.
- Mendukung banyak database server (MySQL, SQL Server, Interbase, Oracle dan sebagainya) sehingga dapat mempermudah dalam membuat aplikasi database.

# Landasan teori

## II.7 *Unified Modeling Laguage* ( UML )

Menurut Martin Fowler dalam bukunya yanng berjudul *"UML Distilled Edisi 3 Panduan Singkat Bahasa Pemodelan Objek Standar"* ditulis bahwa UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).

Kelebihan dari UML ( *Unified Modeling Languang* ) adalah tidak hanya terbatas untuk object oriented software development saja tetapi juga lebih dari itu, UML dapat diterapkan untuk pengembangan software yang lainnya seperti data modeling. UML dapat digunakan untuk menggambarkan perkembangan secara lengkap dari relasi dan objek database relasional dari kebutuhan bisnis hingga physical data model.

# ANALISIS & PERANCANGAN

- **Analisis Masalah**

  Analisis Komponen Permainan

  Analisis Strategi nim

  Analisis Langkah Minimax

- **Analisis Komponen Permainan**

  **1.** Pemain -> dimainkan oleh 2 orang, bergiliran

  **2.** Cara bermain -> mengambil benda dengan jumlah yang tidak ditentukan pada tumpukan

  3. Tipe Permainan -> normal (biasa), *misere*(Yang terakhir mengambil objek, kalah)

  **4.** Permainan selesai -> apabila tidak ada objek yang tersisa dalam tumpukan

# ANALISIS & PERANCANGAN

- **Analisis Strategi Nim**

Ilustrasi permainan:

- ada 10 objek, yang terakhir mengambil objek adalah pemenangnya.
- Asumsi, kita = pemain ke-1.
- Berapa buah objek yang harus diambil agar kita dapat giliran terakhir yang mengambil objek?



- Permainan dimulai...

*pemain ke-1, mengambil 3 objek, sisa 7 objek



*Pemain ke-2, misal mengambil 5 objek



*Pemain ke-1, sisa 2 objek, ambil semua objek, dan menang...

# ANALISIS & PERANCANGAN

Nim dengan 3 baris,

Baris 1 

Baris 2 

Baris 3 

Bagaimana strateginya?

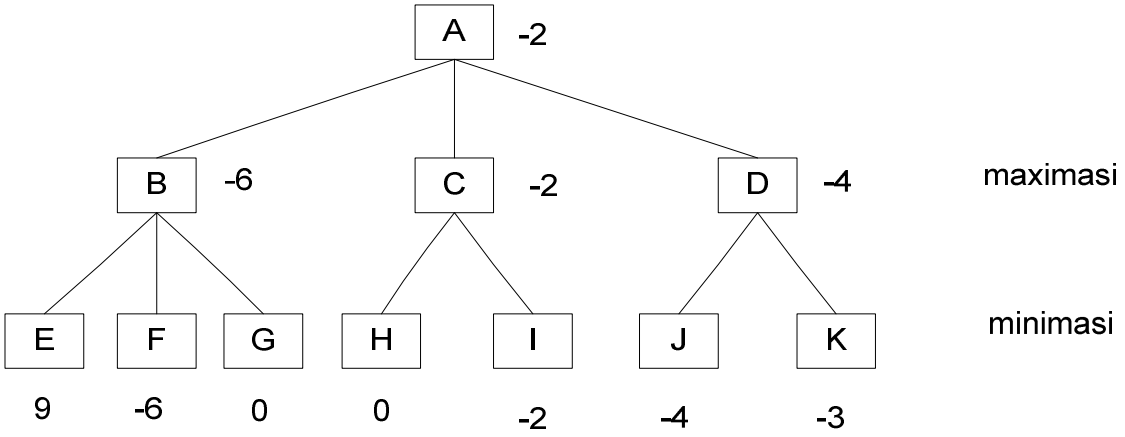
Gunakan *binary digital sum*

| Tumpukan (baris) | Jumlah objek (desimal) | Jumlah objek (biner) | Jumlah objek(Biner) xor *nim-sum* | Hasil penjumlahan |
|---|---|---|---|---|
| **1** | **2** | **010** | **010  100 = 110** | **6** |
| **2** | **3** | **011** | **011 100 = 111** | **7** |
| **3** | **5** | **101** | **101  100 = 001** | **1** |
| ***Nim-sum*** | **2 xor 3 xor 5 = 4** | **100** | | |

# ANALISIS & PERANCANGAN

- **Analisis Langkah Kerja Minimax**

**Konsep awal**




**Implementasi *nimtree*...**

# ANALISIS & PERANCANGAN

- Nim dengan 2 tumpukan baris, jumlah objek 1 dan 3, atau diinisialkan dengan *state*(1,3)
  1. ●
  2. ● ● ●
- Asumsi pemain ke-1 = max, max mendapat giliran bermain pertama
- Langkah yang mungkin dijalankan max adalah:

1. Mengambil 1 objek dari tumpukan pertama, sehingga menyisakan 3 buah objek pada tumpukan ke-2. Kita inisialkan dengan *state* (3), atau

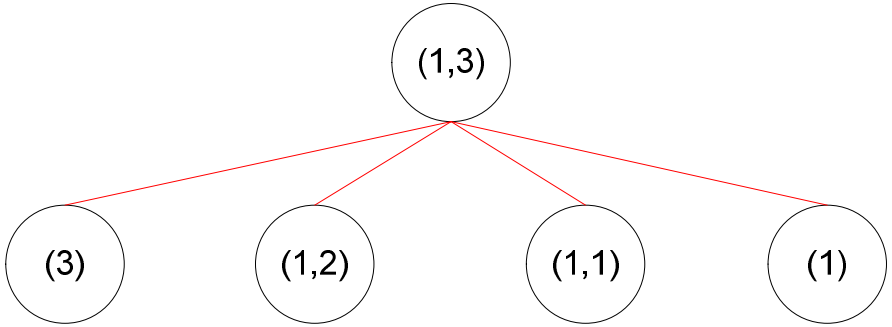
2. Mengambil 1 objek pada tumpukan ke-2, sehingga menyisakan 1 pada tumpukan pertama dan 2 objek pada baris ke-2, (1,2). Atau

3. Mengambil 2 objek pada tumpukan ke-2, sehingga menyisakan 1 objek pada tumpukan pertama dan 1 objek pada tumpukan ke-2, (1,1). Atau

4. Mengambil seluruh objek pada tumpukan ke-2, sehingga hanya menyisakan 1 pada tumpukan pertama. Kita inisialkan (1).

# ANALISIS & PERANCANGAN

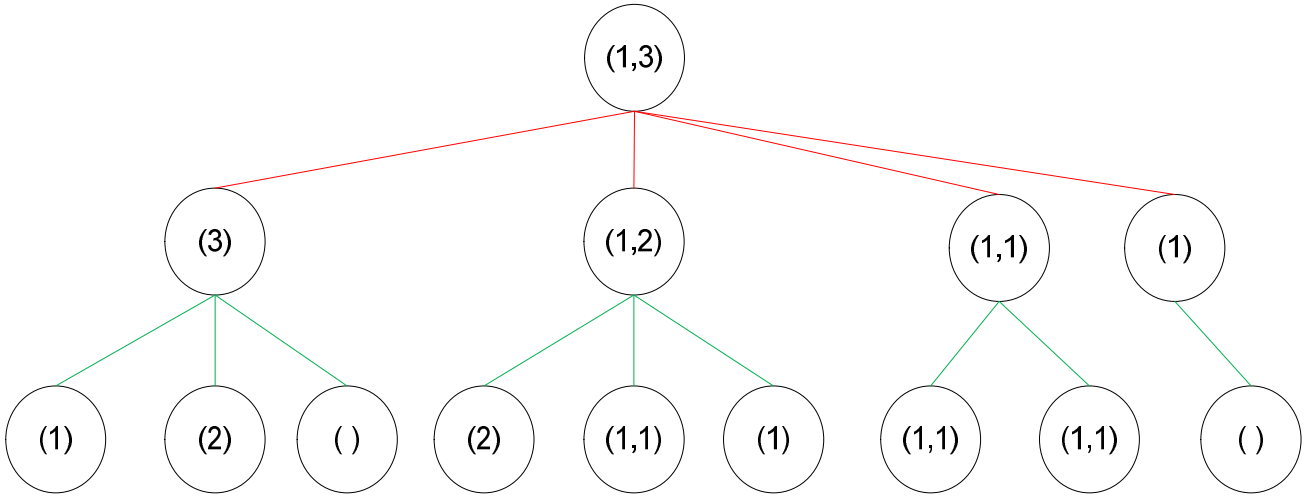
- Represenatasi *tree*




MAX

MIN

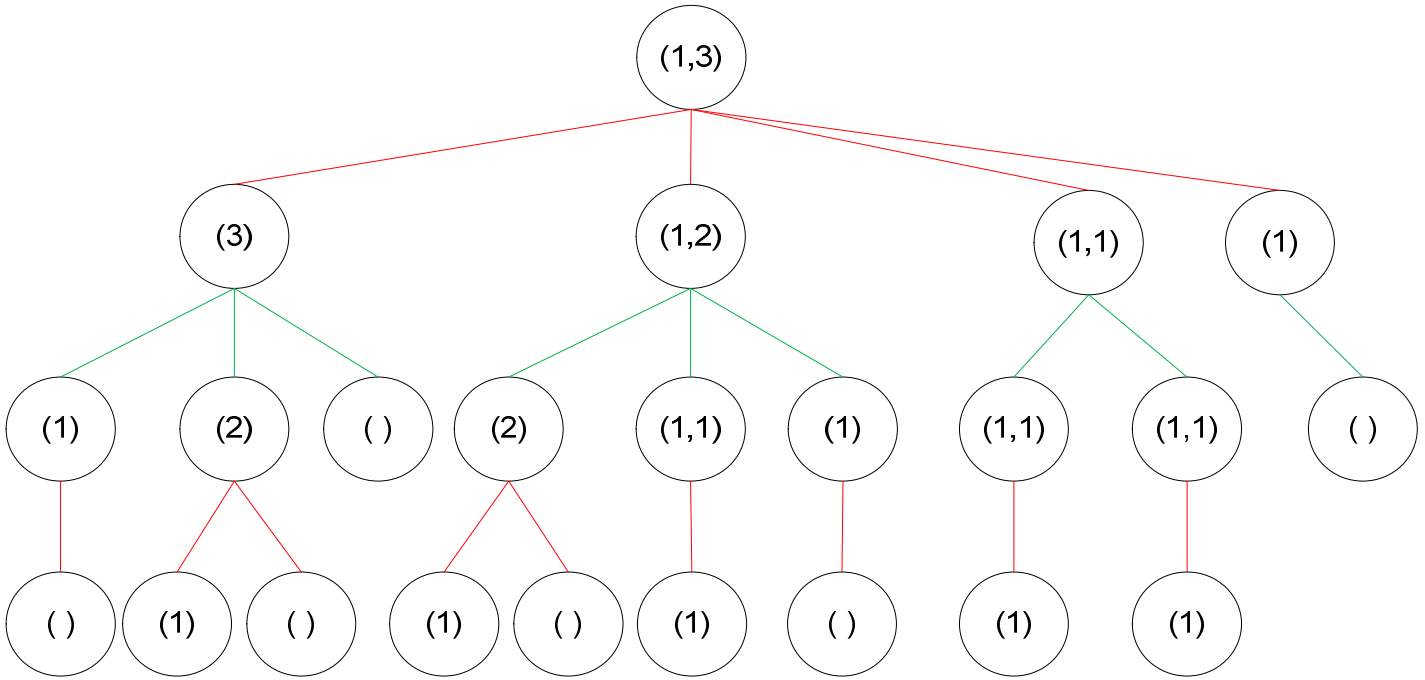
# ANALISIS & PERANCANGAN




MAX

MIN

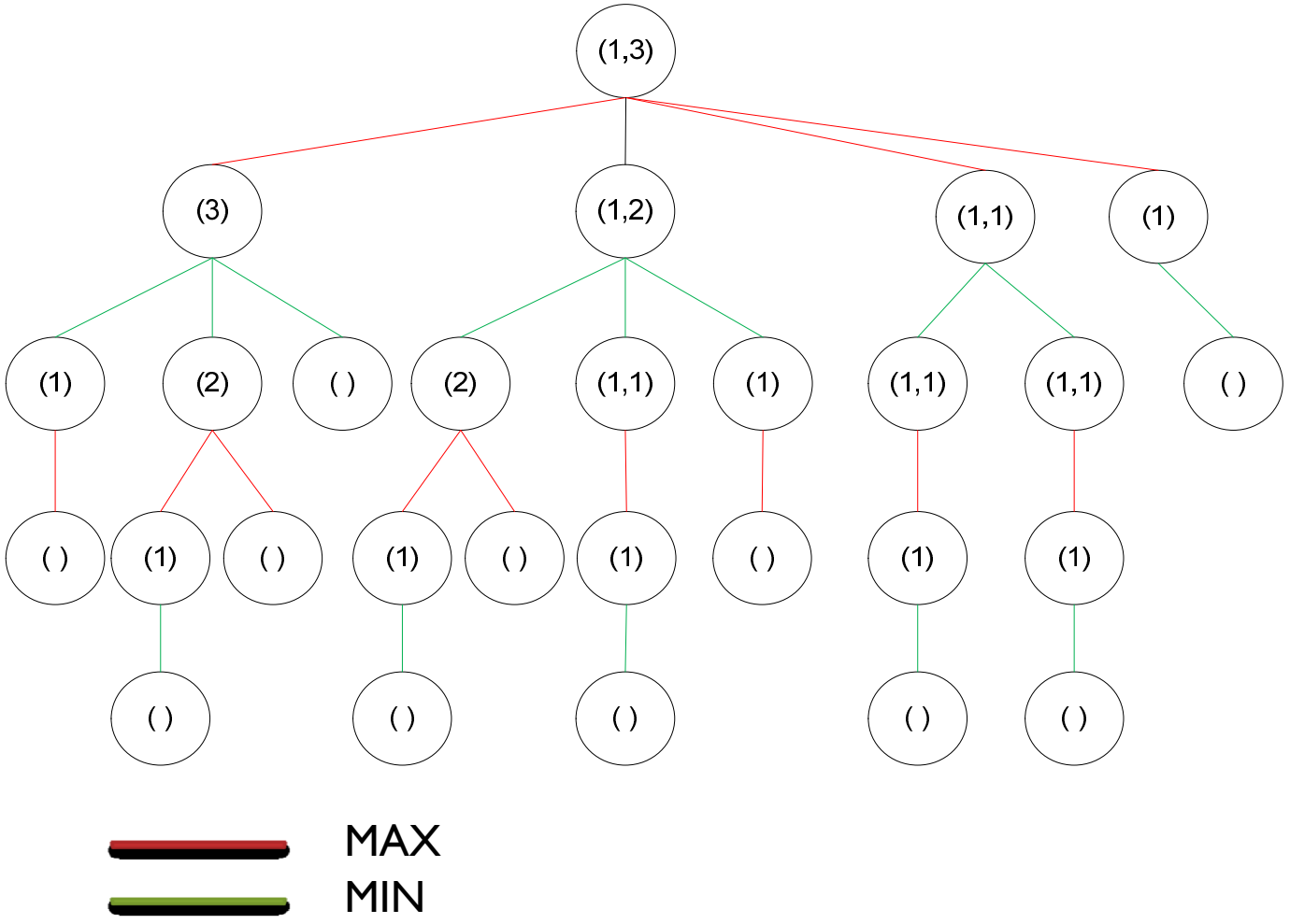
# ANALISIS & PERANCANGAN



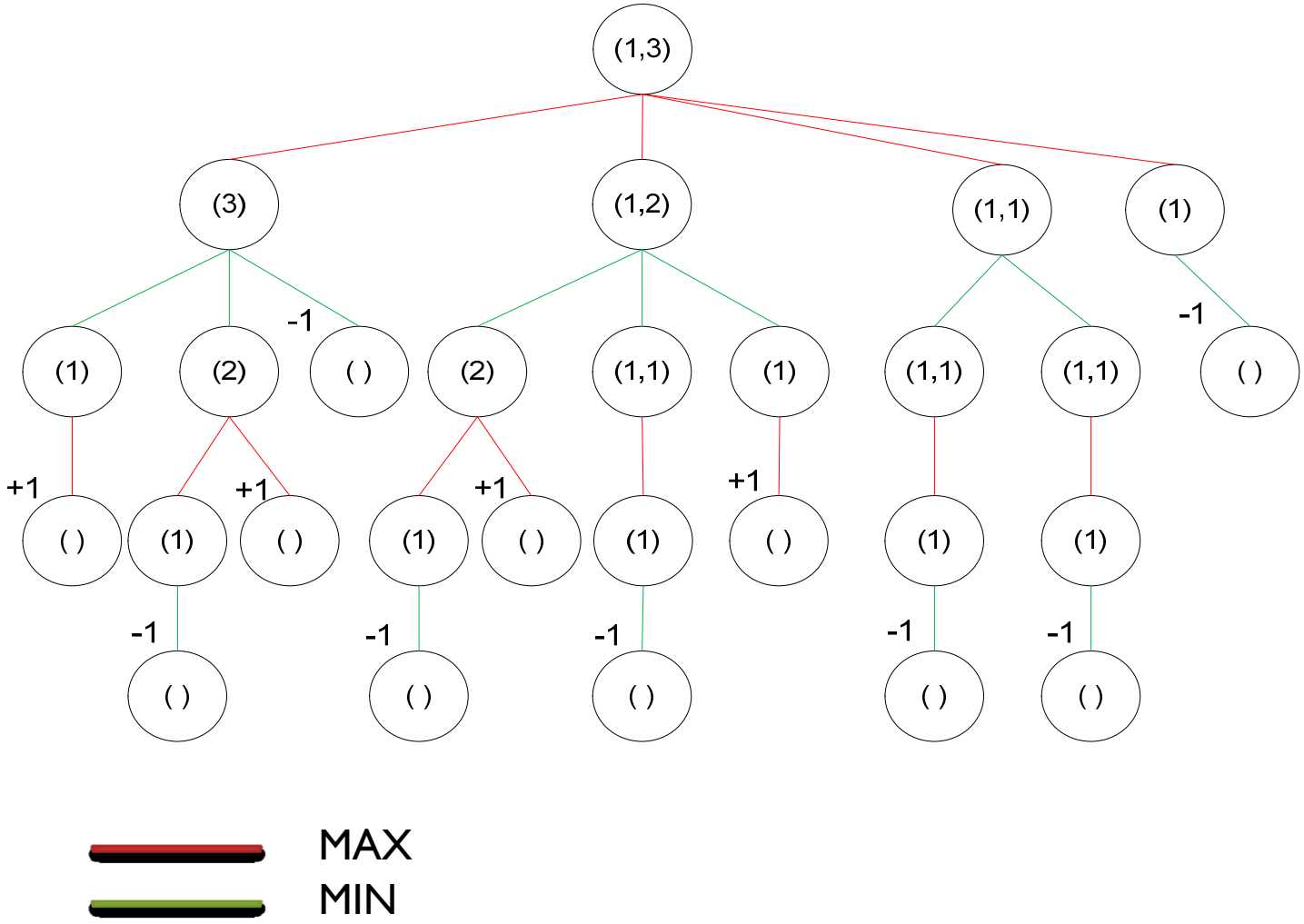

MAX

MIN

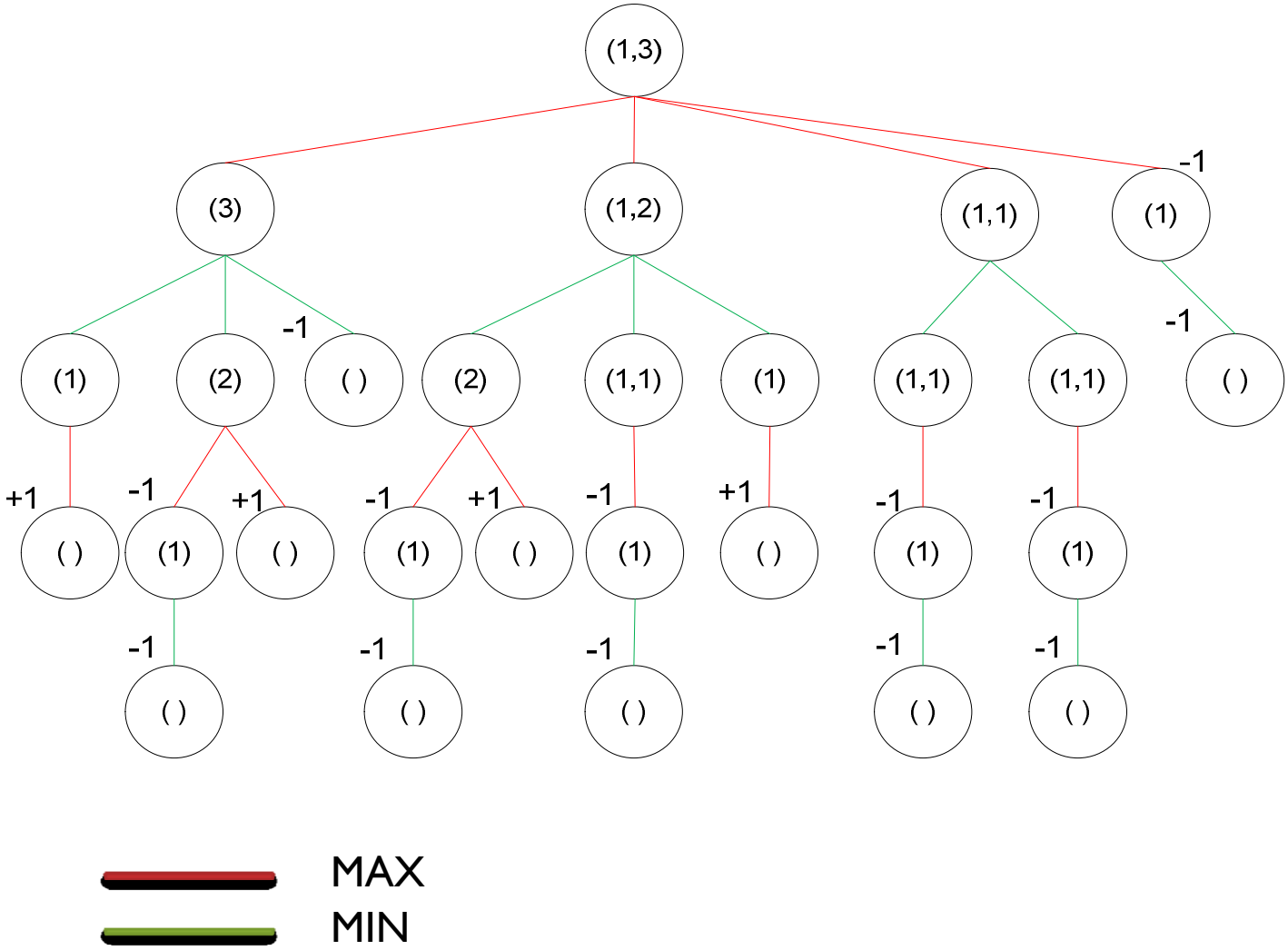
# ANALISIS & PERANCANGAN



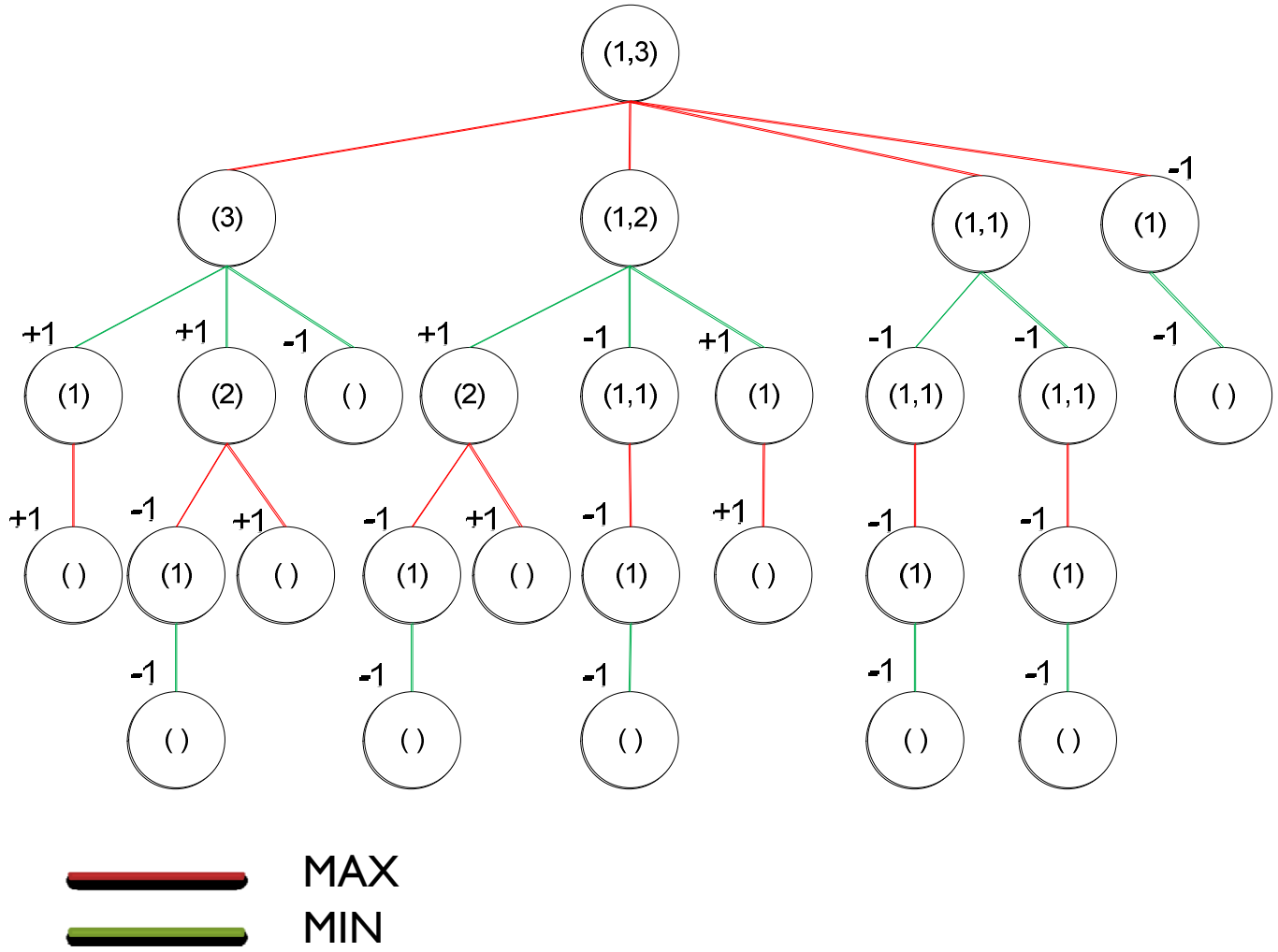
# ANALISIS & PERANCANGAN



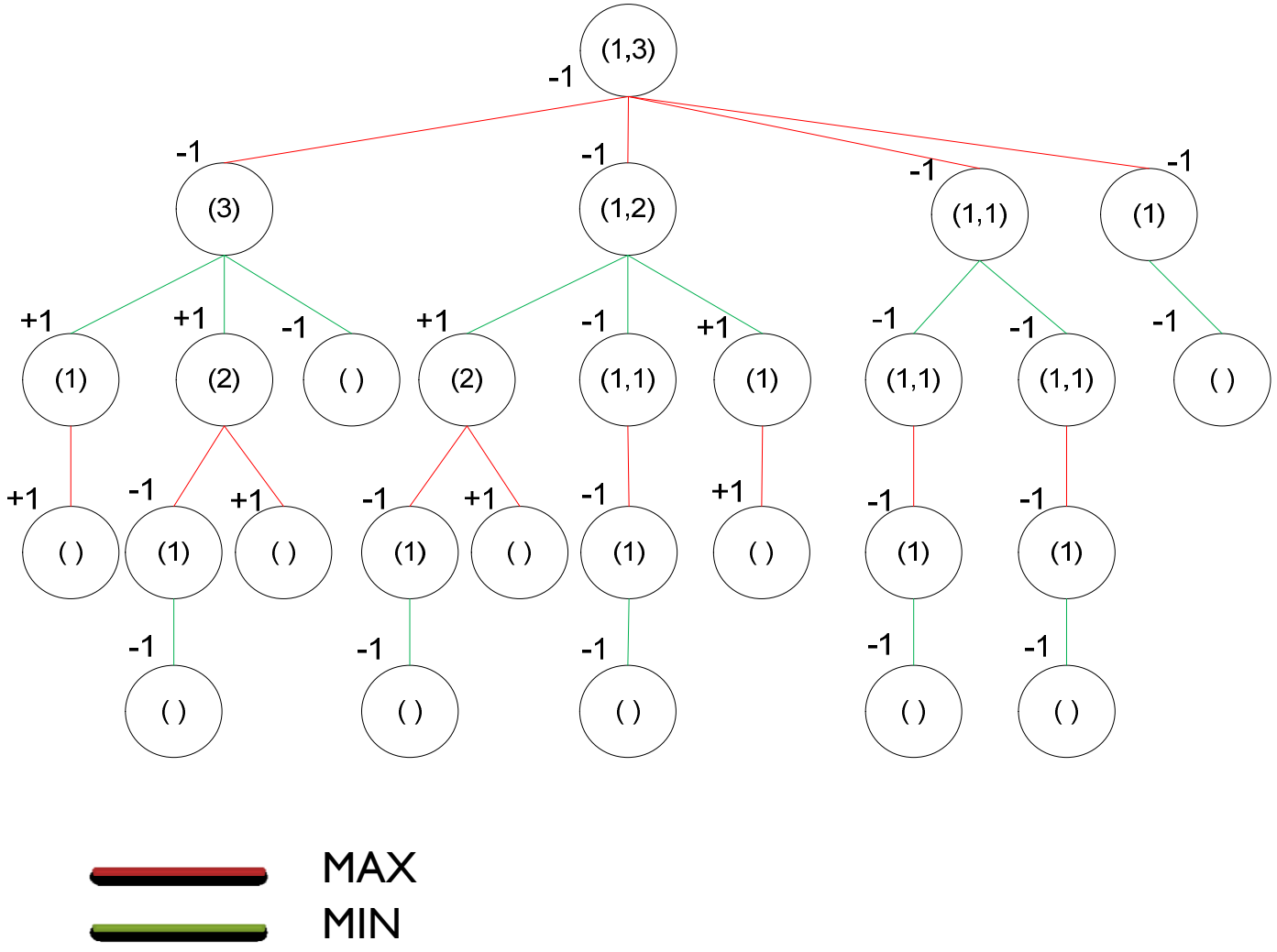
# ANALISIS & PERANCANGAN



# ANALISIS & PERANCANGAN



# ANALISIS & PERANCANGAN



# ANALISIS & PERANCANGAN

**Analisis Kebutuhan Non-Fongsional**

**1. Analisis Perangkat Keras**

- Processor intel pentium 4
- Memori minimal 128 MB
- Monitor resolusi 1024X768 pixel
- Mouse

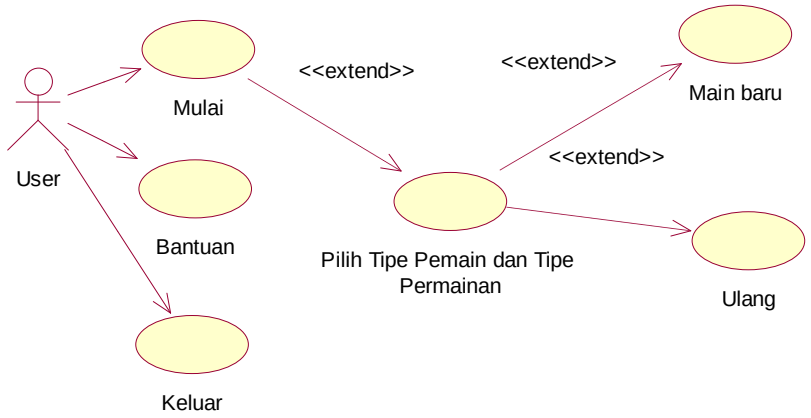
**2. Analisis kebutuhan perangkat lunak (*software*)**

- OS: Window
- Bahasa Pemrograman: Delphi 7.0

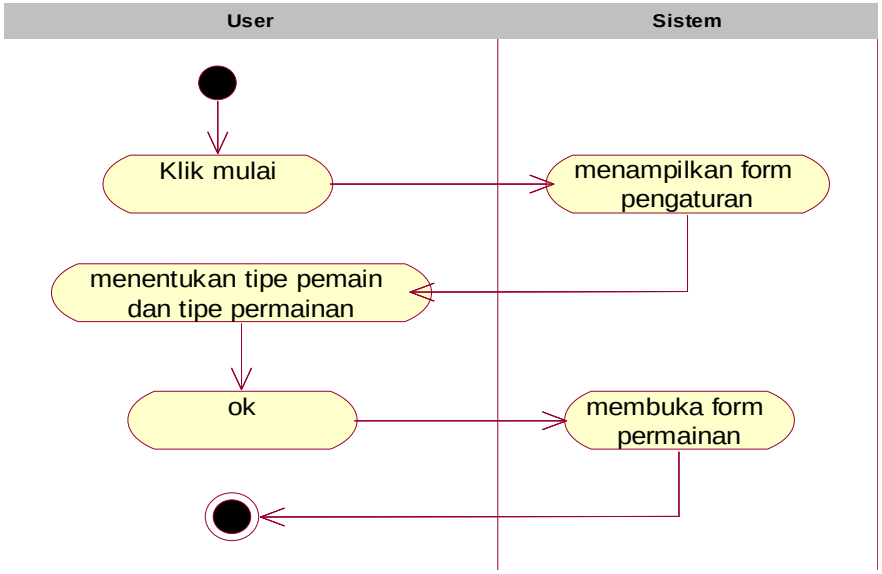
3. **Analisis pengguna**

- Pemain (*Player*) untuk memainkan permainan nim. Pengguna yang dapat mengerti dan memahami computer sehingga dapat menggunakan aplikasi yang dibnagun.

# Use Case Diagram

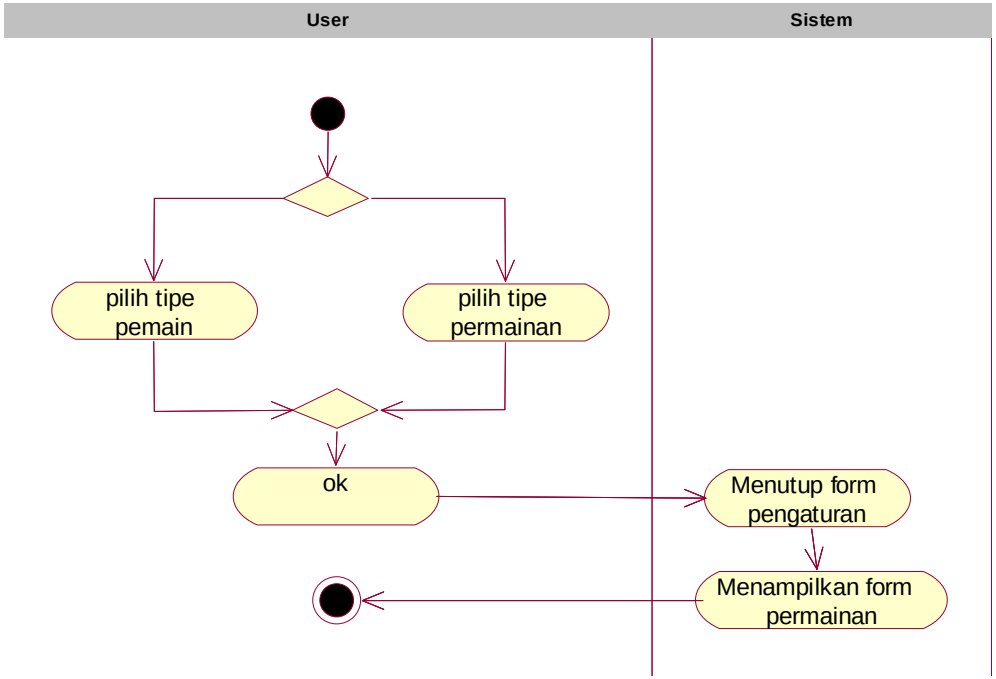


# ANALISIS & PERANCANGAN



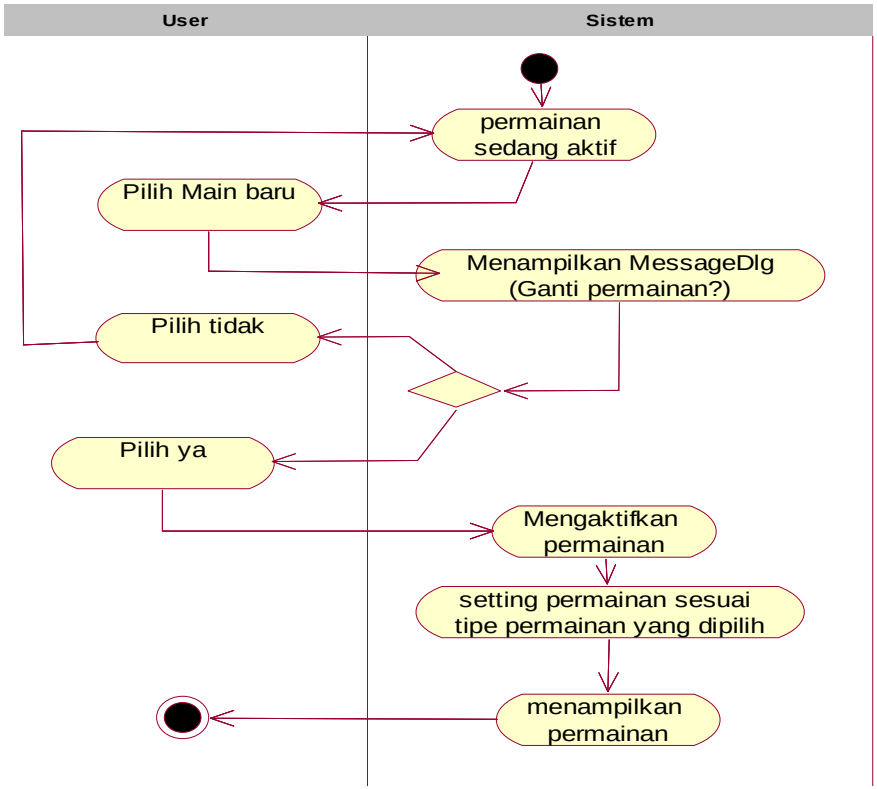

***Activity Diagram* Mulai**

# ANALISIS & PERANCANGAN



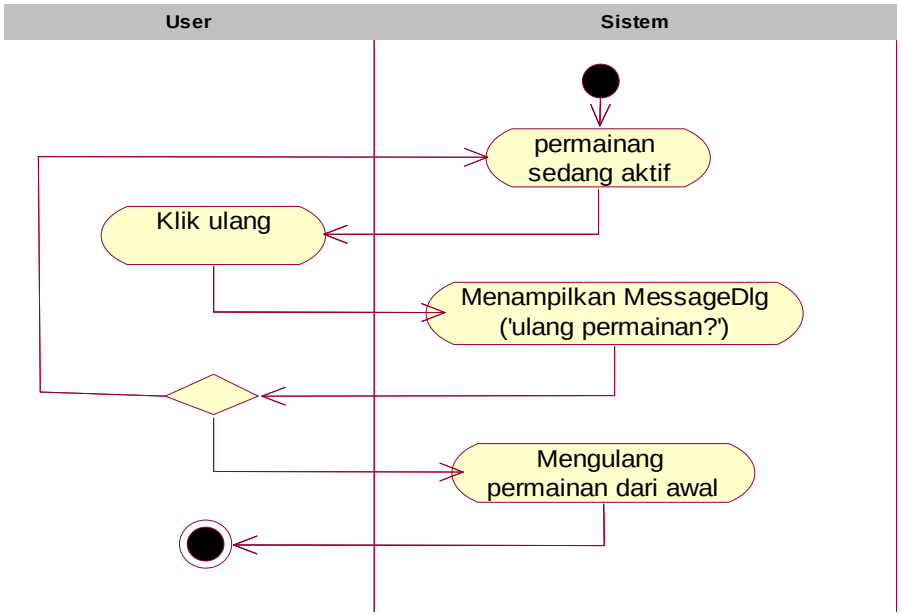

***Activity Diagram*** **Pilih Tipe Pemain dan Tipe permainan**

# ANALISIS & PERANCANGAN



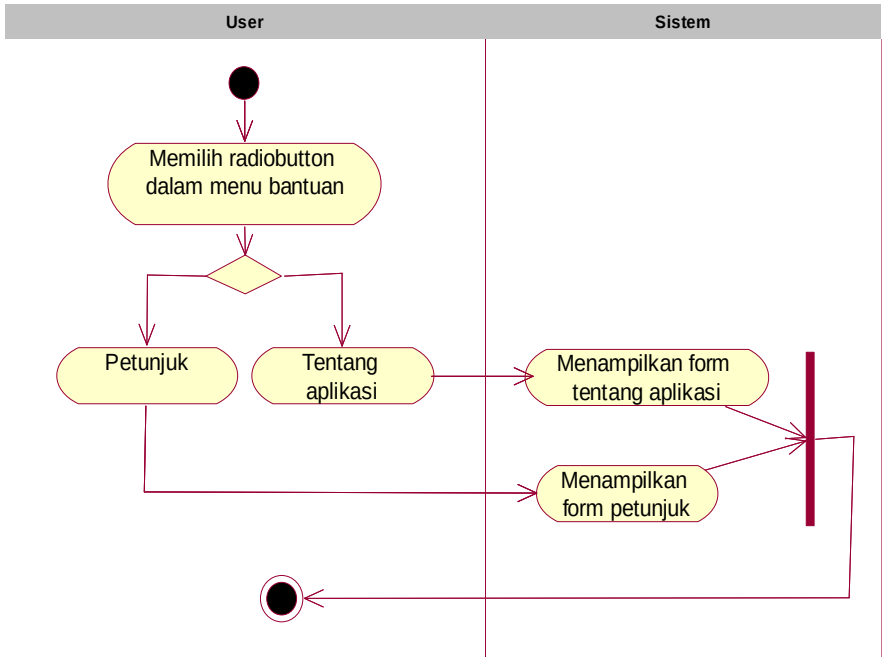

***Activity Diagram*** **Main baru**

# ANALISIS & PERANCANGAN



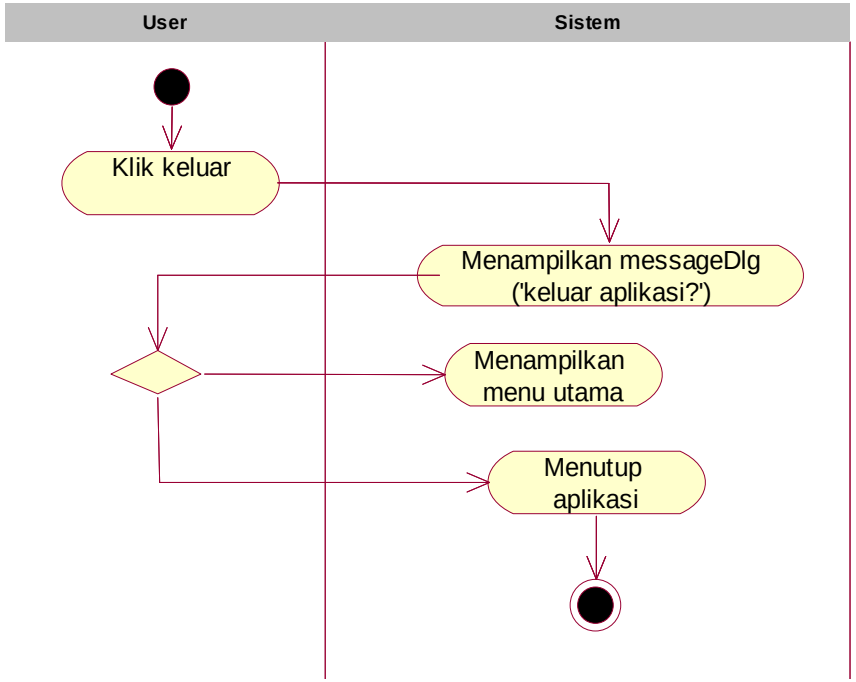

***Activity Diagram* Ulang**

# ANALISIS & PERANCANGAN




**Gambar III.18 *Activity Diagram* Bantuan**

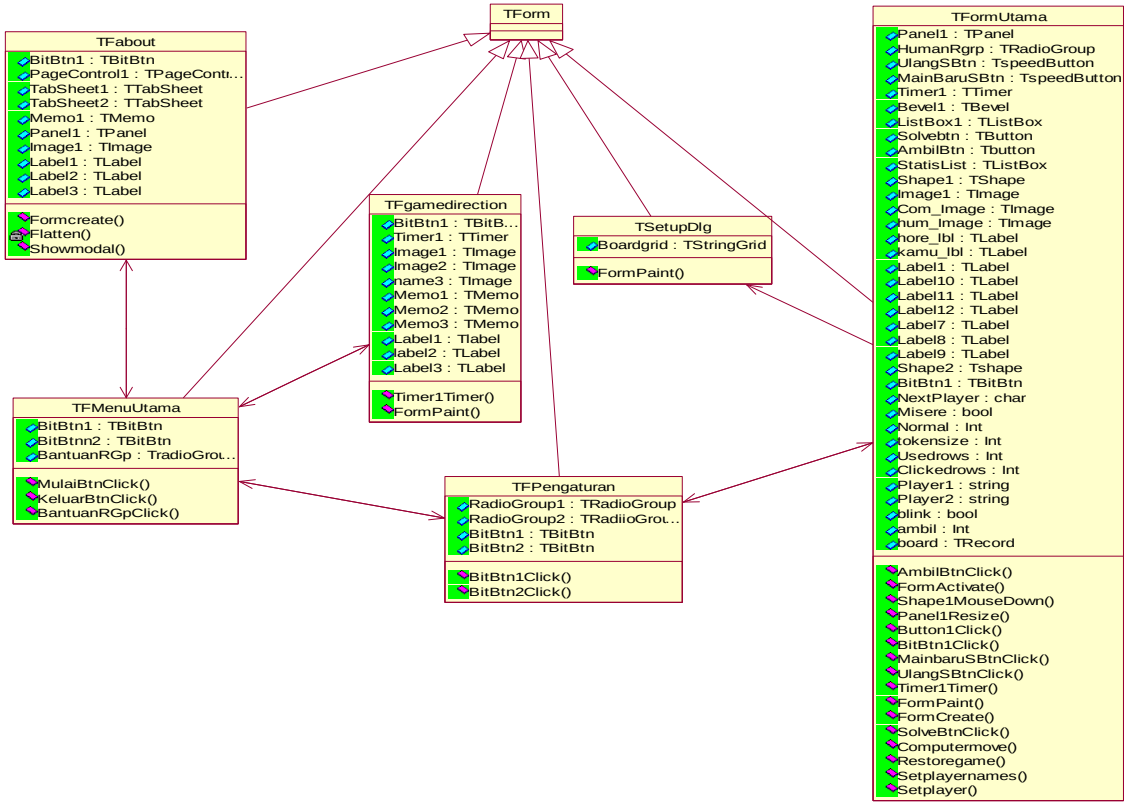
# ANALISIS & PERANCANGAN




***Activity Diagram* Keluar**

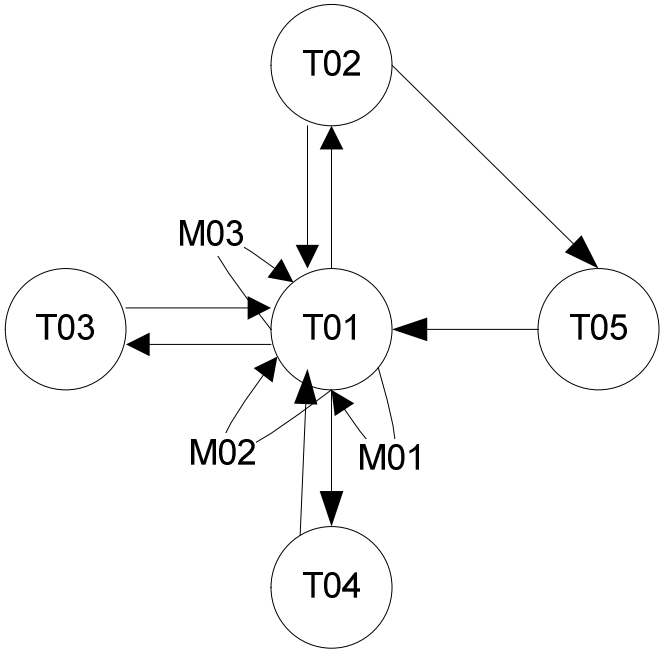
# ANALISIS & PERANCANGAN

- ***Class Diagram***



# ANALISIS & PERANCANGAN

- Jaringan Semantik




- T01 : form menu utama
- T02 : form pengaturan
- T03 : form utama
- T04 : form petunjuk
- T05 : form tentang aplikasi
- M01 : pesan konfirmasi keluar aplikasi
- M02 : pesan konfirmasi ganti permainan
- M03 : pesan konfirmasi ulang permainan

# IMPLEMENTASI DAN PENGUJIAN

Perangkat keras

Adapun Perangkat keras yang digunakan untuk pembuatan aplikasi adalah sebagai berikut:

- 1. processor 1,5GHz
- 2. memori minimal 128 MB
- 3. Harddisk minimal 250 GB
- 4. Monitor resolusi 1024X760 pixel
- 5. Mouse

Perangkat lunak

| Komputer | Spesifikasi perangkat lunak |
|---|---|
| Sistem Operasi | Microsoft Window 7 ultimate 32 bit |
| Bahasa Pemograman | Borland Delphi 7.0 |

# IMPLEMENTASI DAN PENGUJIAN

Pengujian blackbox

yaitu pengujian yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun. Berikut pengujian menggunakan metode *blackbox*.

Pengujian whitebox

pengujian yang didasarkan pada pengecekan terhadap detil perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

# TERIMA KASIH