

MEMBANGUN GAME

DETECTIVE CODE OUT

- **Oleh:**
- **FADLI FADLILLAH SULAIMAN**
- **10108579**

Reviewer 1:
Nelly Indriani W,S .Si.,M.T.

Reviewer 2 :
Irfan Maliki, S.T.,M.T

Reviewer 3 :
Sufaatin, S.T.

LATAR BELAKANG MASALAH

- **Game sejenis “jump”**

- **Musuh bersifat pasif dalam bergerak dan menyerang**
- **Tidak Adanya Level Sehingga Terkesan Monoton**

PERUMUSAN MASALAH



Bagaimana mengembangkan game dengan perilaku musuh yang berbeda dari game yang di analisis.

MAKSUD DAN TUJUAN

- **MAKSUD dan TUJUAN**

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas akhir ini adalah untuk membangun game Detective Code Out dengan menambahkan tantangan sebagai berikut:

- **1. Menambahakan keunikan *game* berupa misi memecahkan code.**
- **2. Menambahkan *behavior* musuh dengan kecerdasan buatan.**

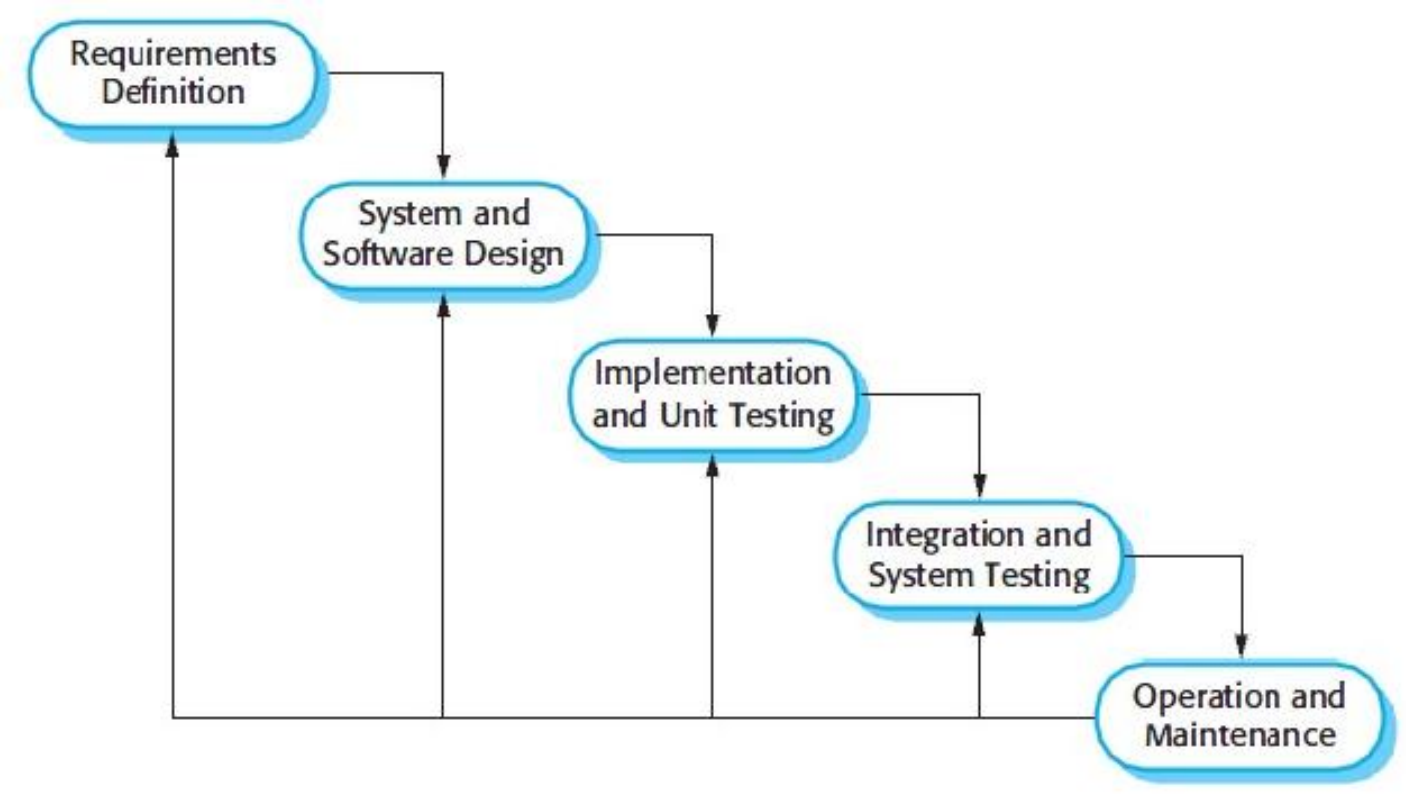
BATASAN MASALAH

- Game 2 dimensi Detective code out yang dibangun dengan genre side scrolling.
- Game dibangun memiliki 3 level.
- NPCs (non playable characters) yang akan dibuat bersifat artificial Intelligence, dengan menggunakan metode algoritma A* dalam melakukan pencarian terhadap pemain.

BATASAN MASALAH

- Game ini bersifat single player.
- Perancangan perangkat lunak menggunakan tools Greenfoot dengan bahasa pemograman java.
- Aplikasi game akan berjalan pada mesin atau computer PC yang dipasang JVM (Java Virtual Machine).
- Menggunakan pemodelan berorientasi objek dengan tools UML.
- Pengguna yang memainkan berumur 11 tahun keatas.

PENGEMBANGAN PERANGKAT LUNAK



ANALISIS GAME SEJENIS

- JUMP



ANALISIS GAME DETECTIVE CODE OUT

- Story Line
- Analisis Tingkat Kesulitan
- Entitas Yang Terlibat Dalam Game
- Story Board

STORY LINE

- **Detective yang bernama Brian harus mengungkap kasus pembunuhan seorang direktur bank. Setelah diselidiki ternyata moggi adalah pelaku utamanya.**

ANALISIS TINGKAT KESULITAN

- Dalam game ini, terdapat 3 level yang masing-masing level dibedakan berdasarkan alur cerita, tingkat kesulitan rintangan yang harus dihadapi, jumlah dan jenis enemy yang bertambah. Di setiap peningkatan level, user akan dihadapkan pada enemy dengan tingkat kesulitan yang akan bertambah.

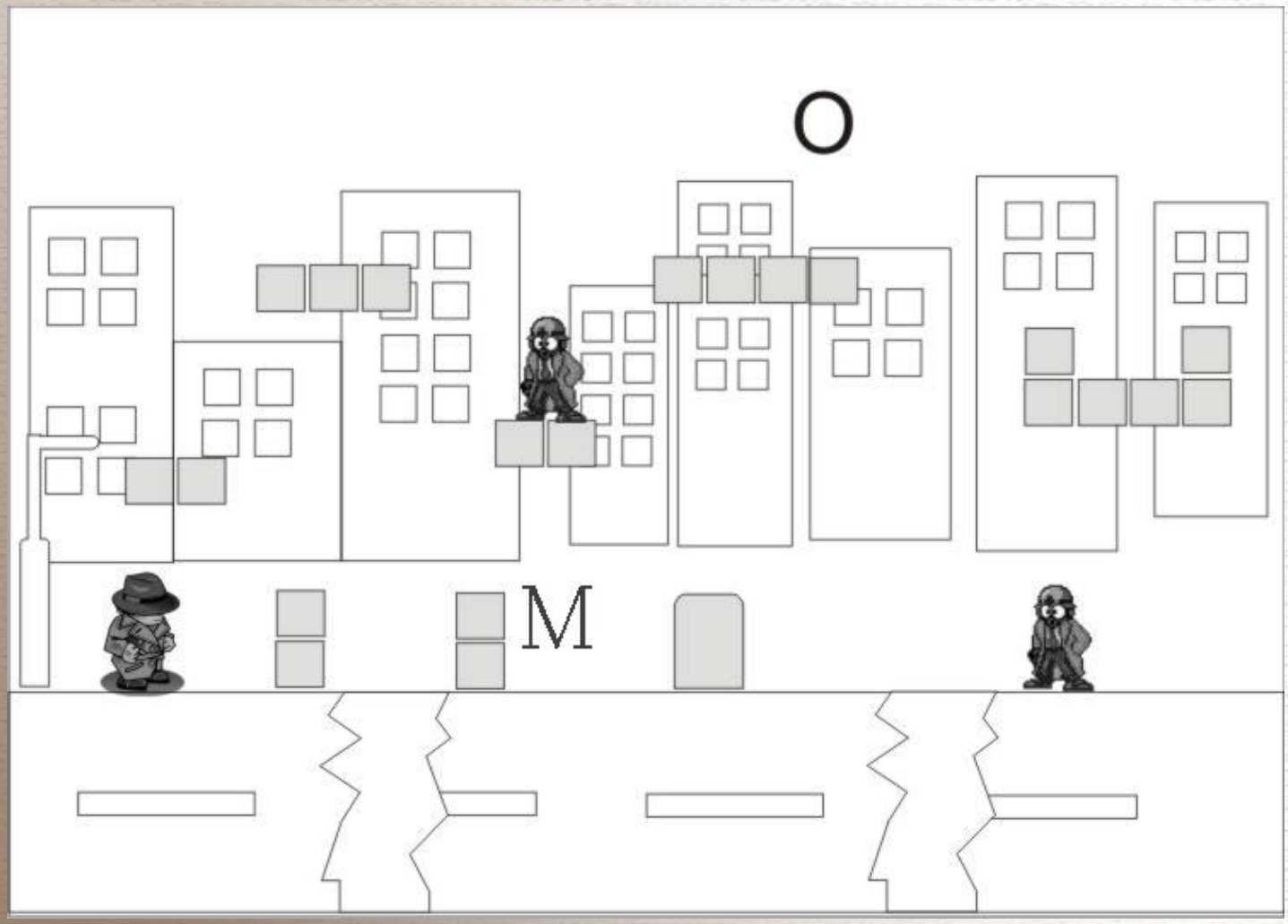
Entitas yang terlibat dalam permainan

NO.	NAMA	GAMBAR	PENJELASAN
1.	Brian		Brian adalah karakter player utama pada game, memiliki kemampuan untuk melompat, menghindar, dan juga bertarung dengan musuh
2.	Jack		Jack merupakan musuh pada level 1, jack memiliki kemampuan untuk mengejar Brian jika berada pada radius jarak penglihatannya, melompat dan bertarung dengan Brian
3.	Billy		Billy merupakan musuh pada level 2, Billy memiliki kemampuan untuk mengejar Brian jika berada pada radius penglihatannya, melompat dan bertarung dengan Brian
4.	Moggi		Moggi merupakan musuh di level 3. Moggi merupakan bos dari jack dan Billy.

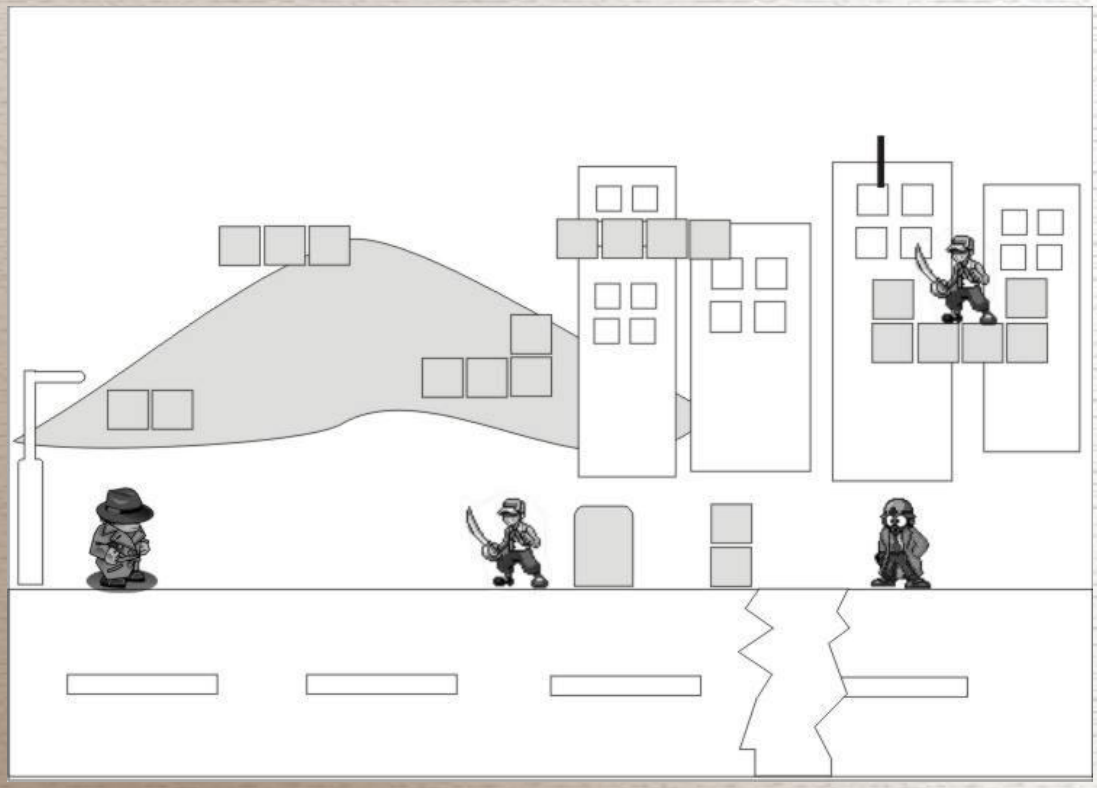
STORY BOARD

- Level 1.
- Level 2.
- Level 3.

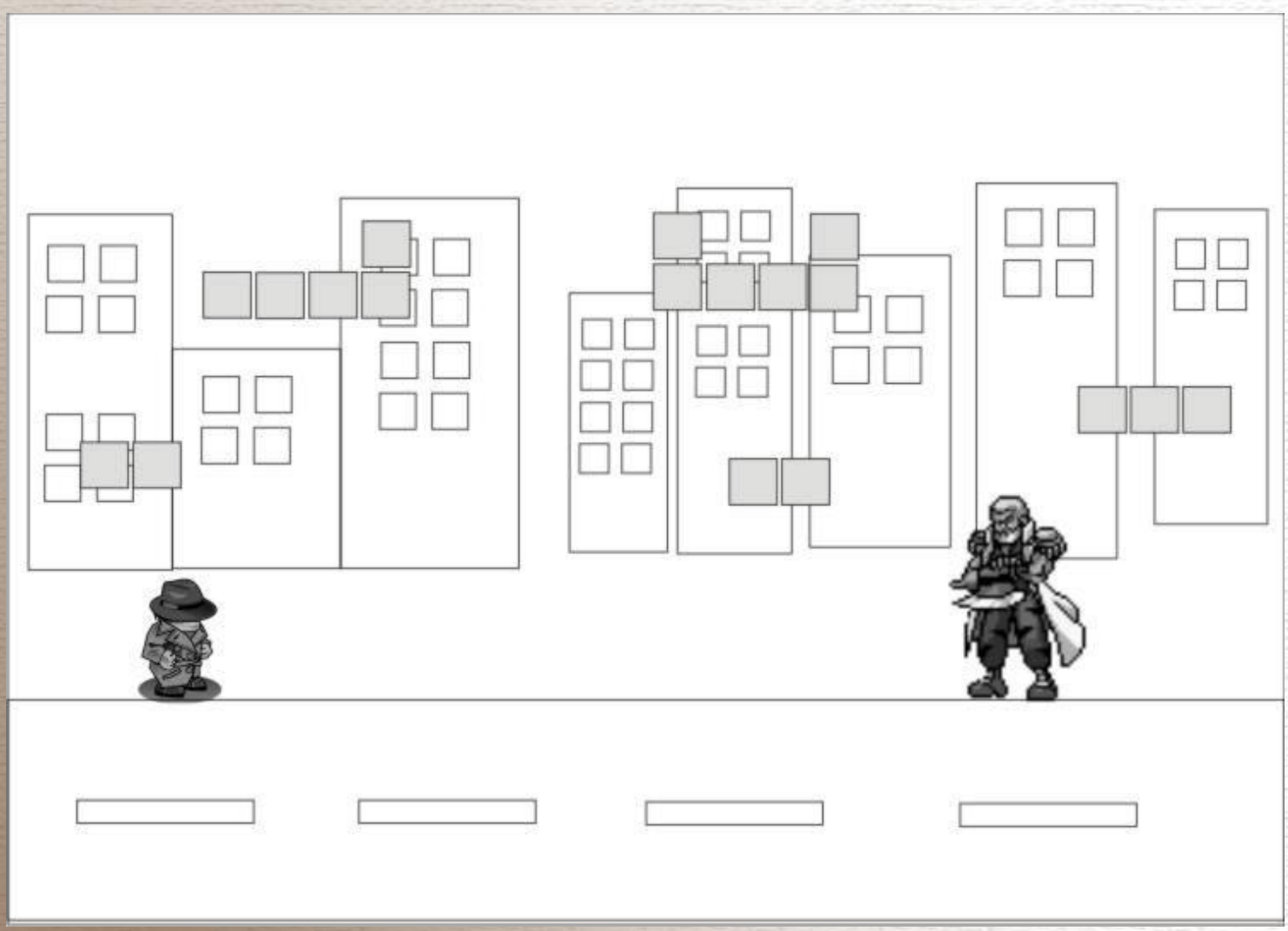
STORY BOARD LEVEL 1



STORY BOARD LEVEL 2



STORY BOARD LEVEL 3



ANALISIS ALGORITMA A*

- **Algoritma A* melakukan pencarian jarak terpendek dengan biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi matematika dituliskan sebagai :**

$$f(n) = g(n) + h(n).$$

- **Dengan perhitungan ini algoritma A* adalah complete dan optimal.**

ANALISIS ALGORITMA A*

Gambar Arena Permainan pada *Game Detective Code Out*

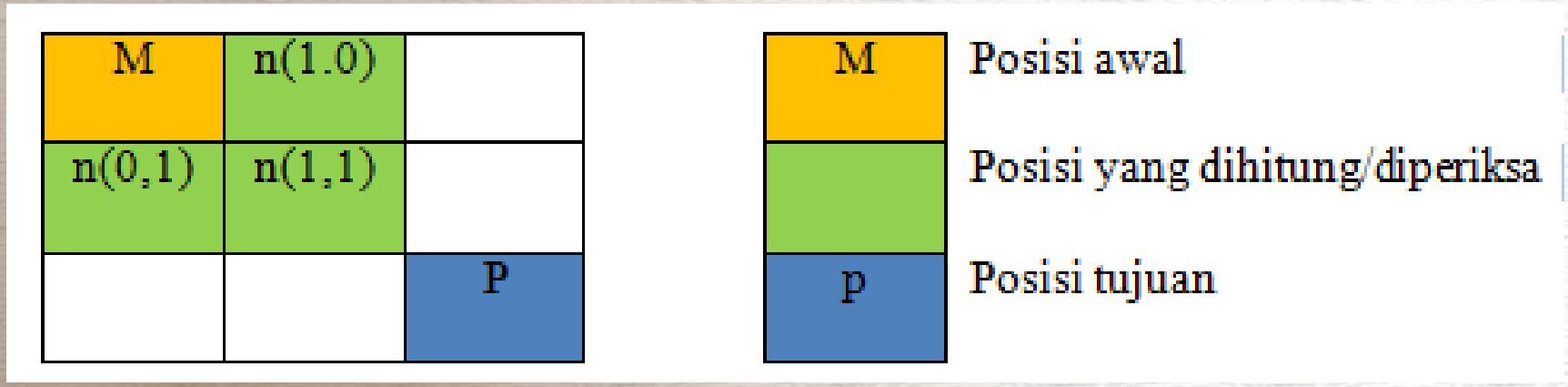
(0.0)	(1,0)	(2.0)
(0,1)	(1,1)	(2,1)
(0,2)	(1,2)	(2,2)

M		
		P

Keterangan : **P : pemain**
M : musuh

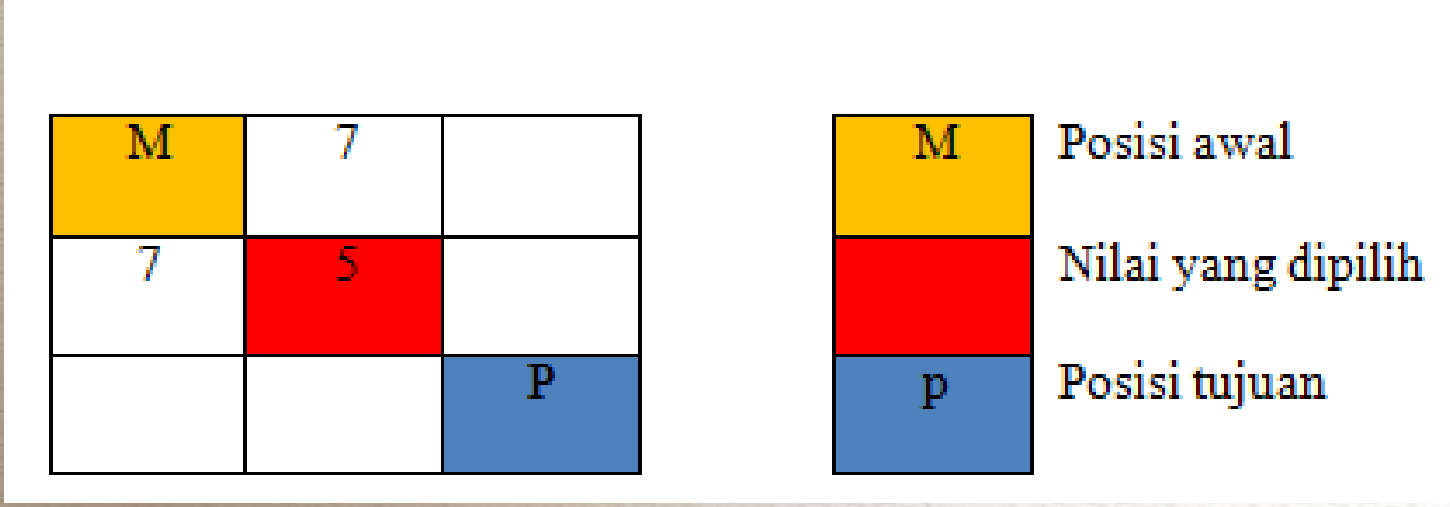
ANALISIS ALGORITMA A*

- **Langkah pertama**



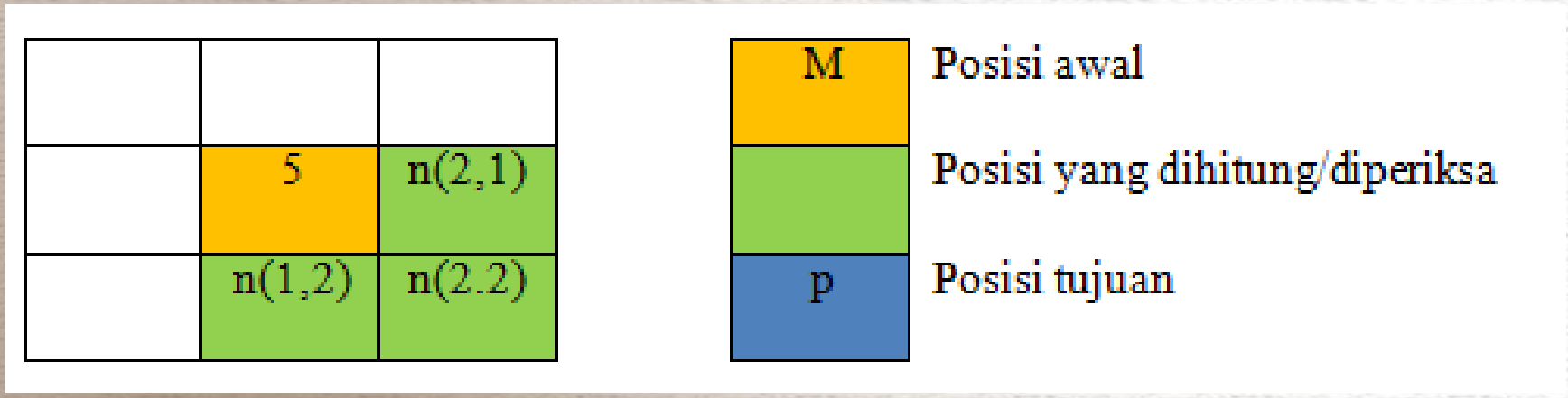
ANALISIS ALGORITMA A*

- **Hasil Langkah pertama**



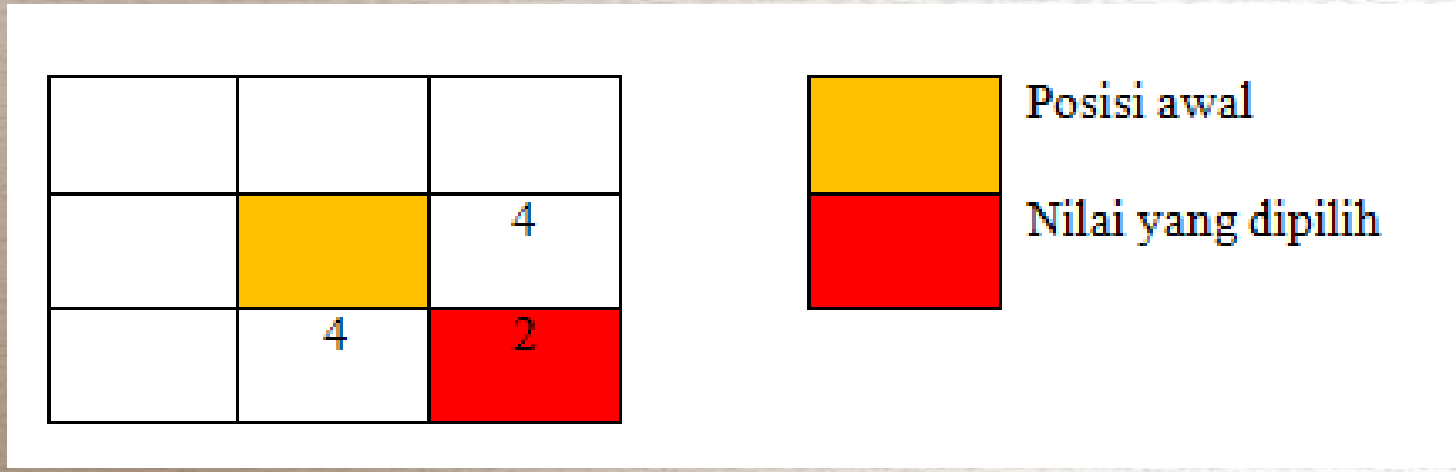
ANALISIS ALGORITMA A*

- **Langkah Kedua**



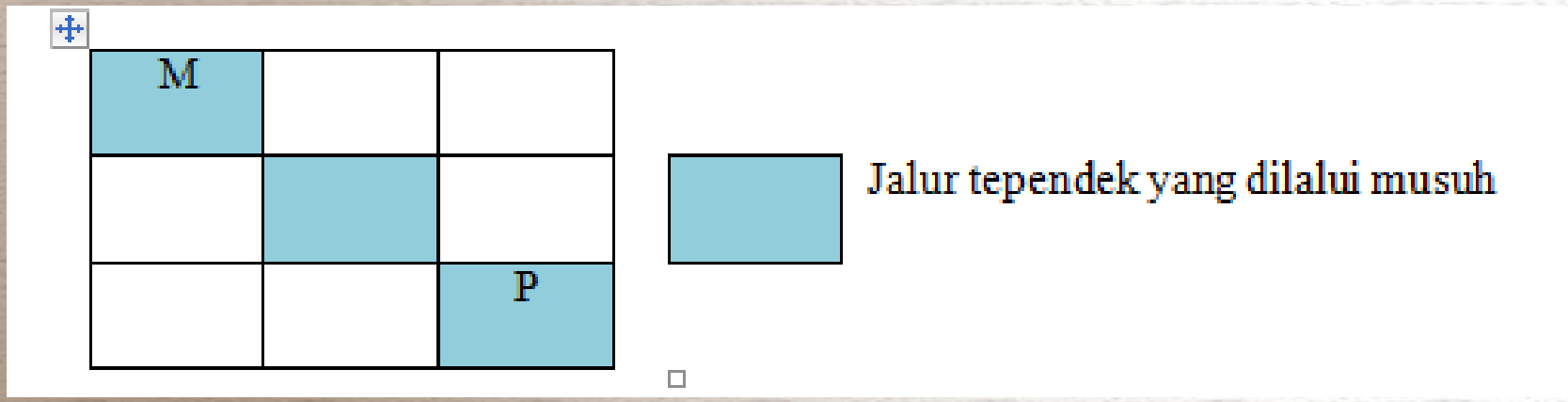
ANALISIS ALGORITMA A*

- **Hasil langkah kedua**



ANALISIS ALGORITMA A*

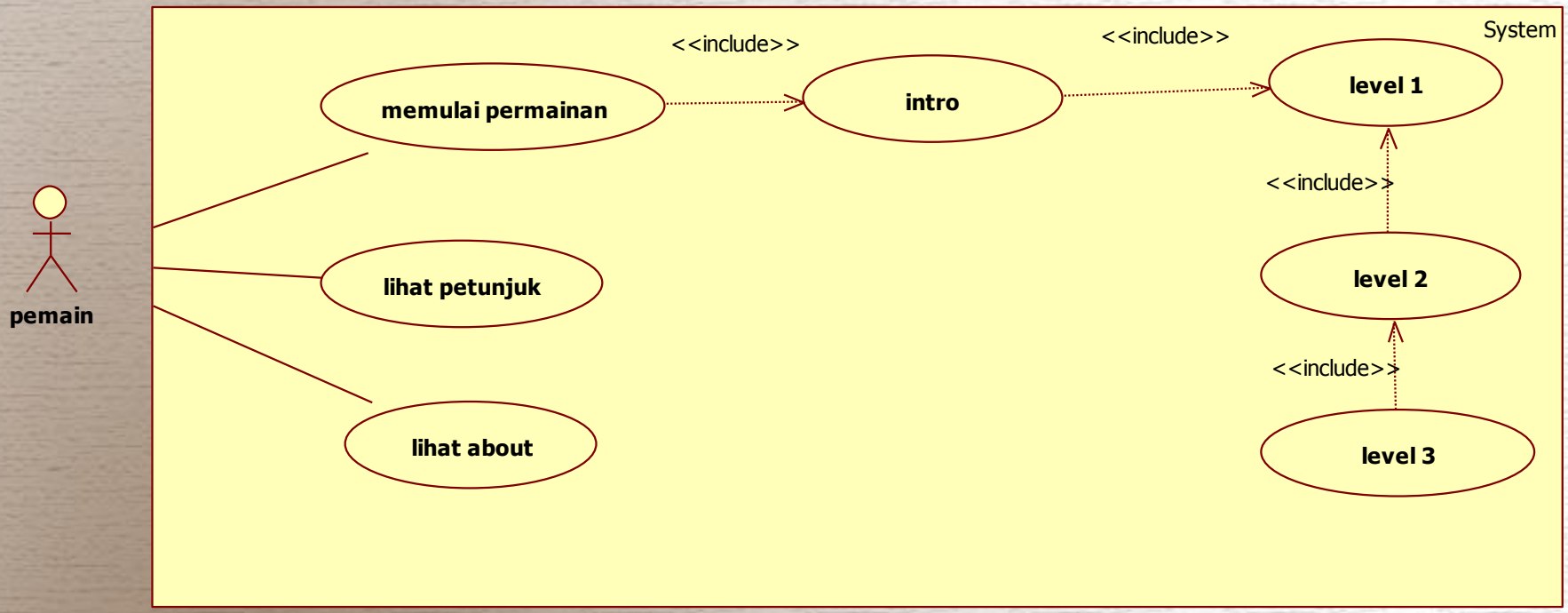
- **Jalur terpendek**



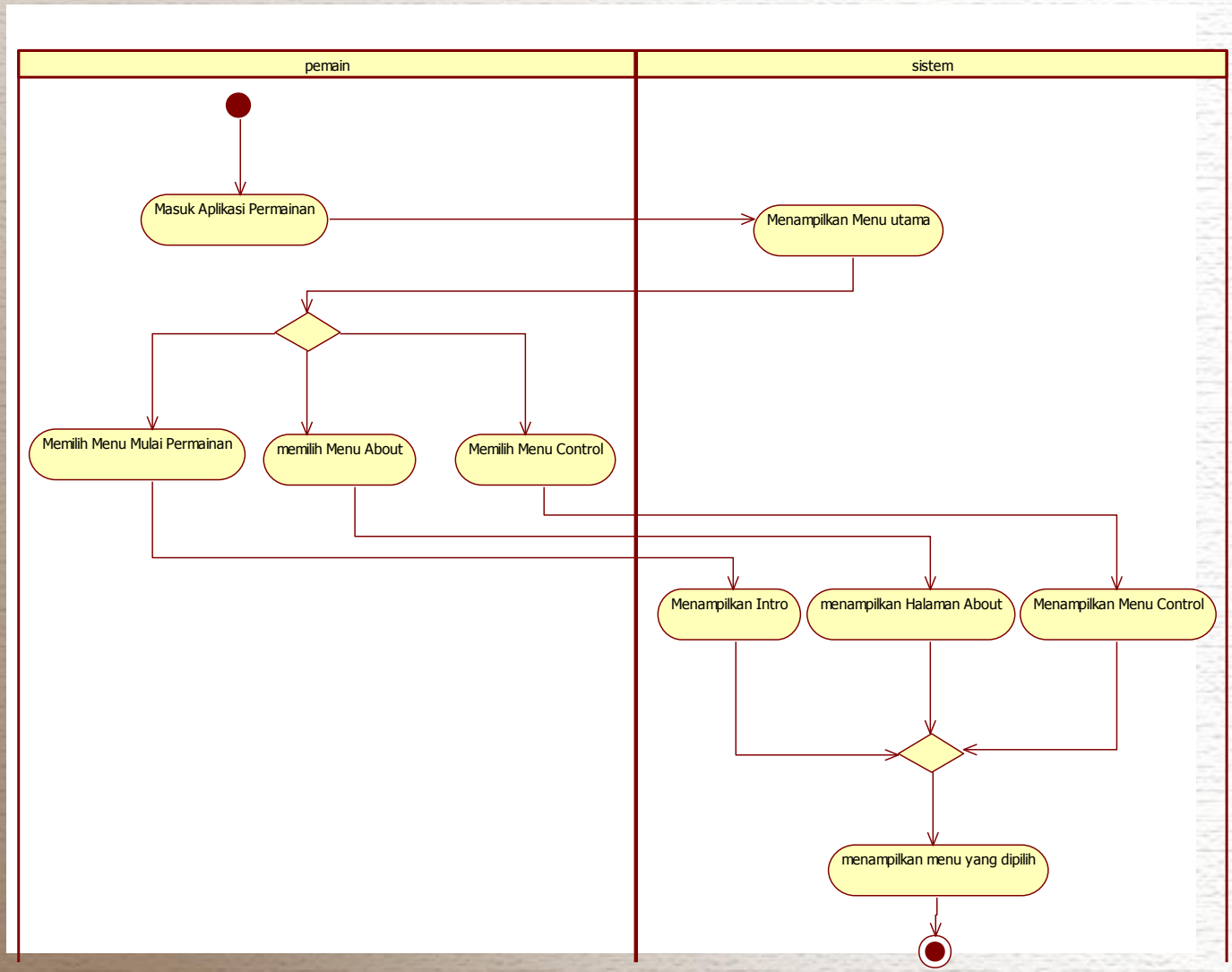
ANALISIS KEBUTUHAN FUNGSIONAL

- Use Case Diagram
- Activity Diagram
- Sequence Diagram
- Class Diagram

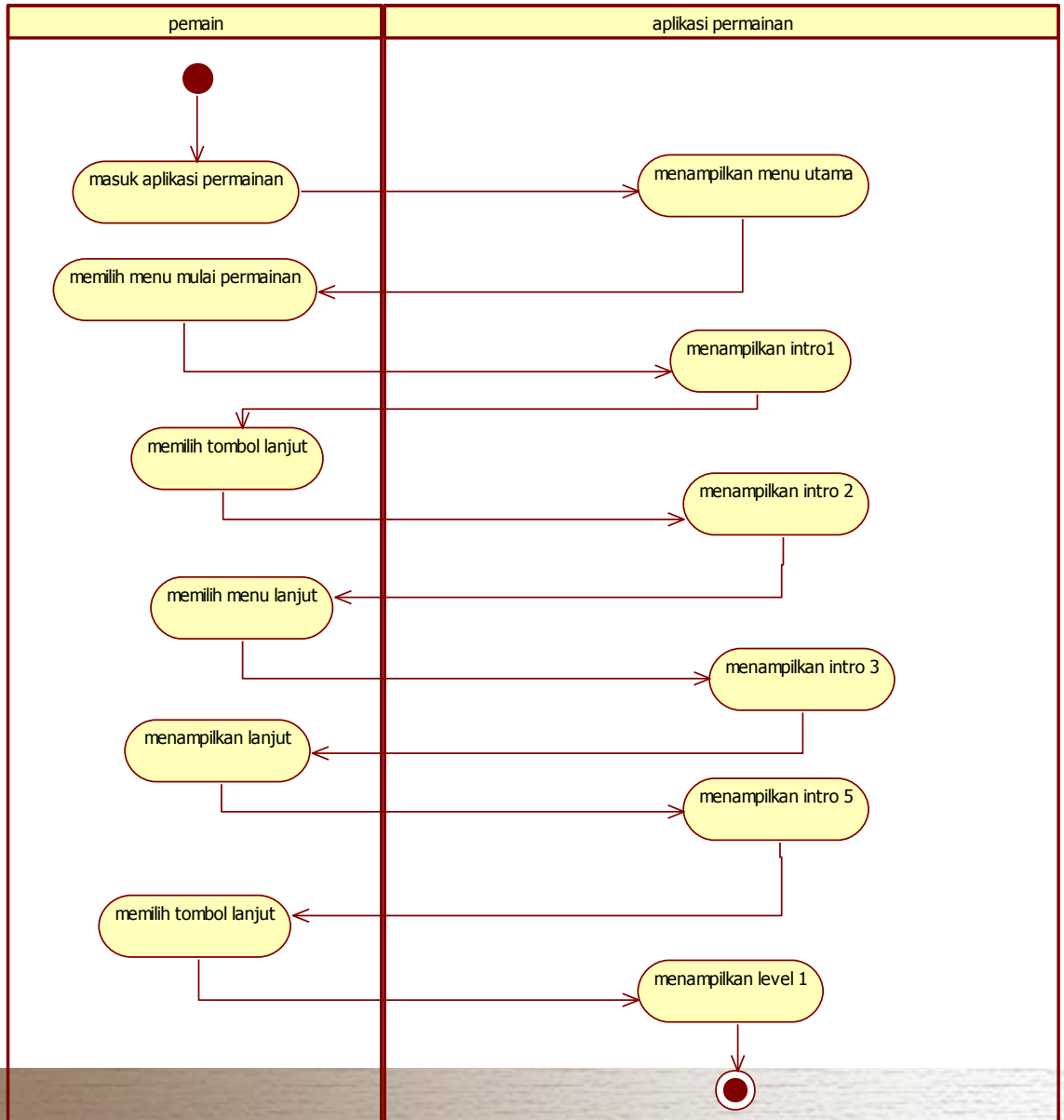
Use Case Diagram



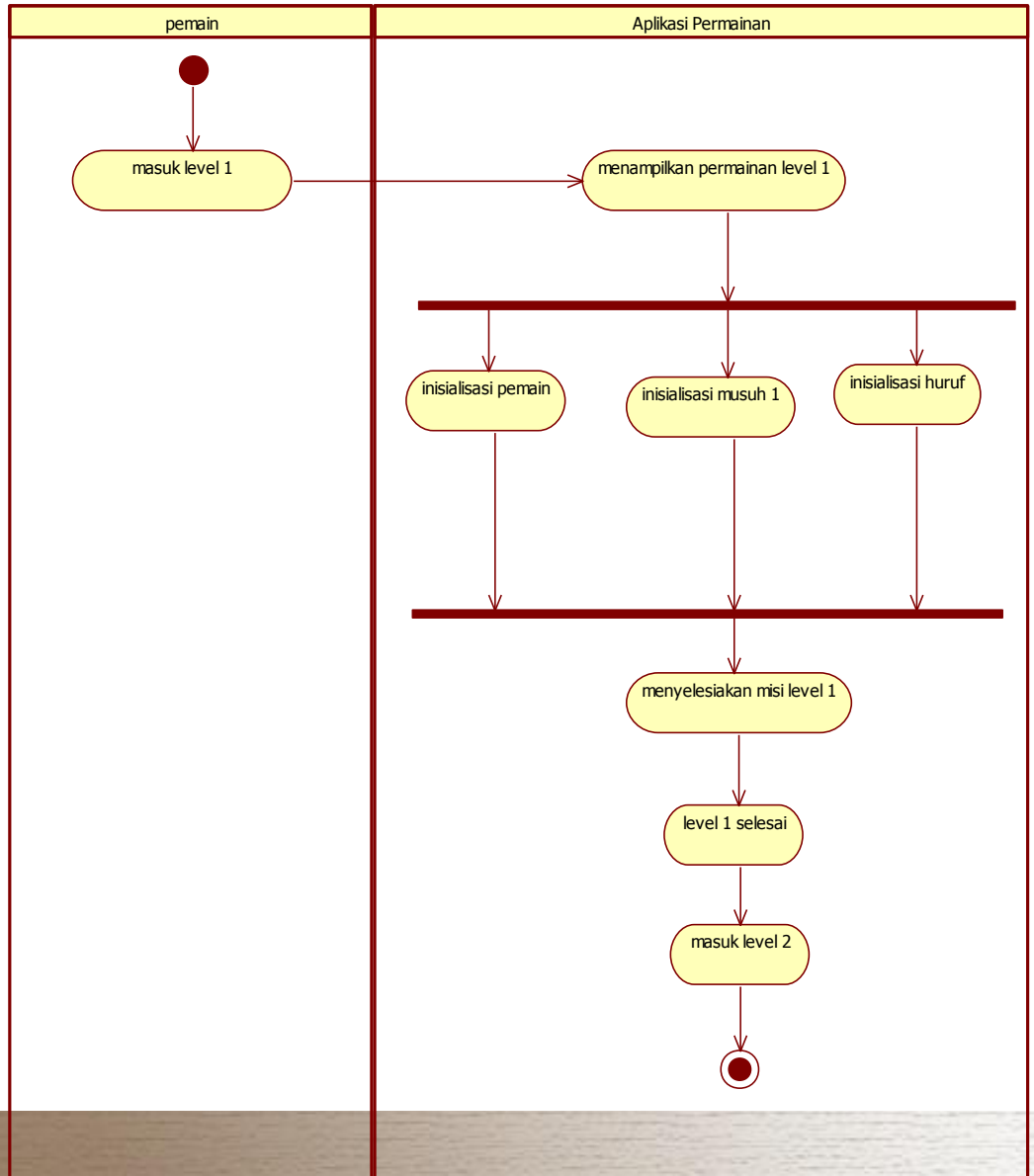
Activity Diagram Mulai Permainan



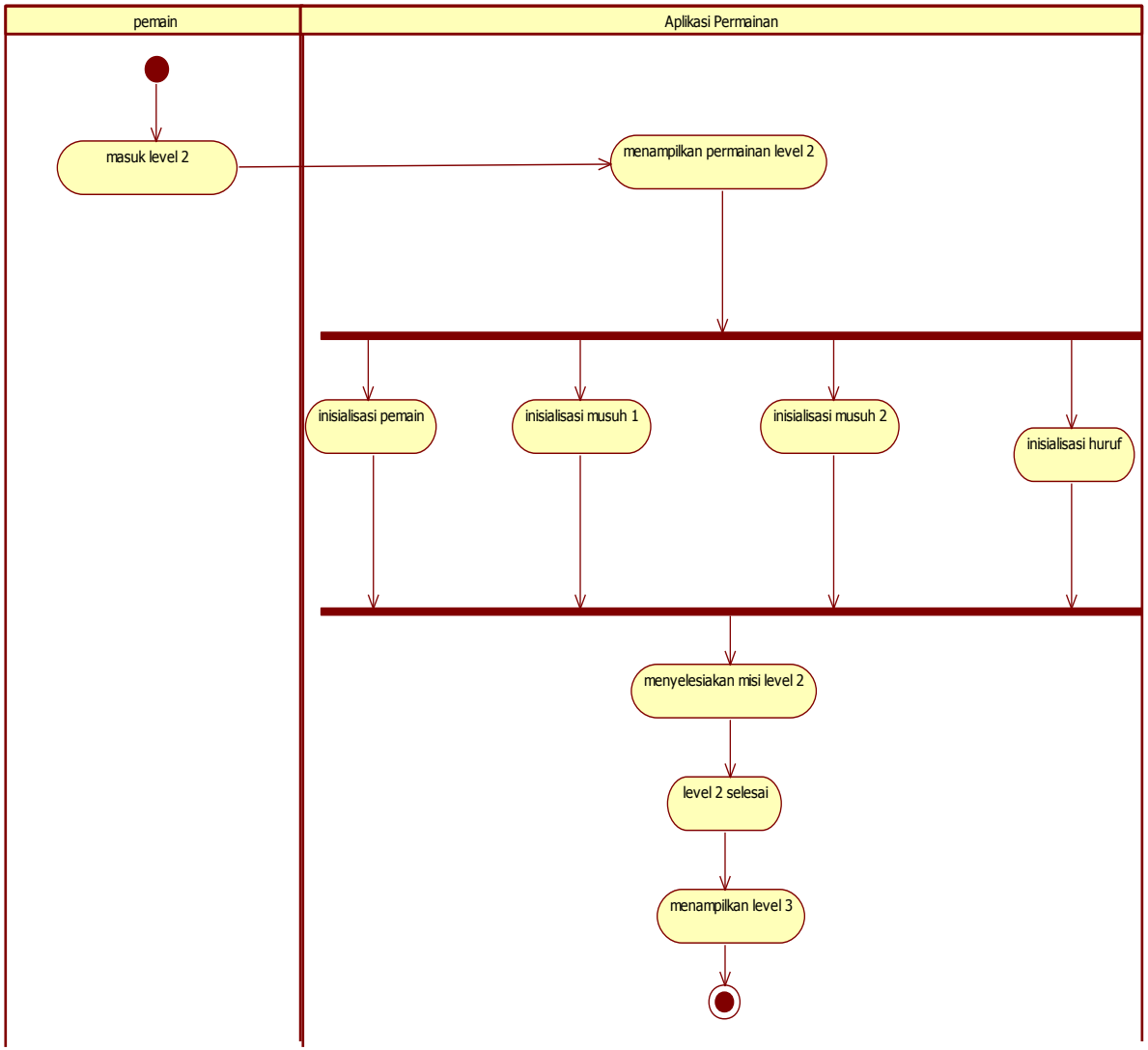
Activity Diagram Intro



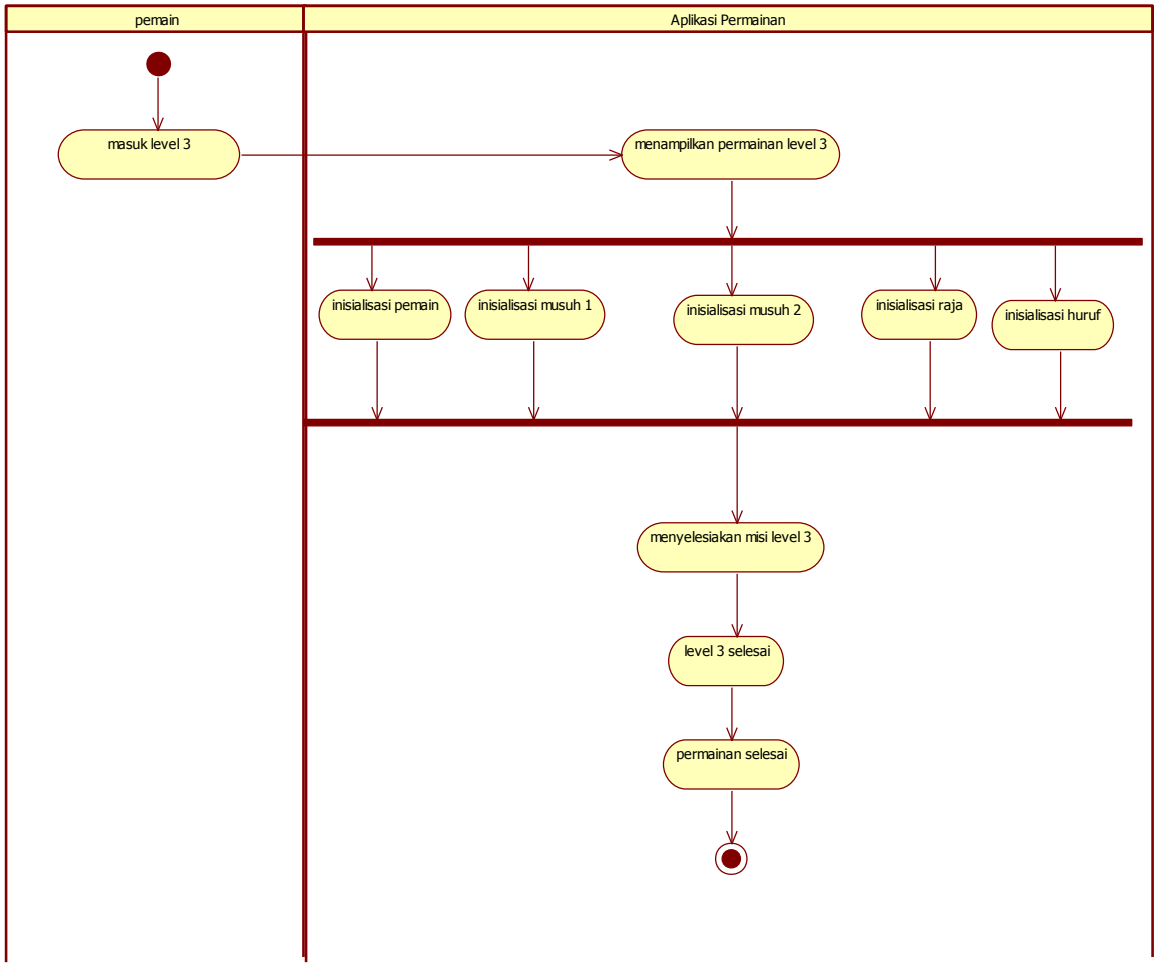
Activity Diagram Level 1



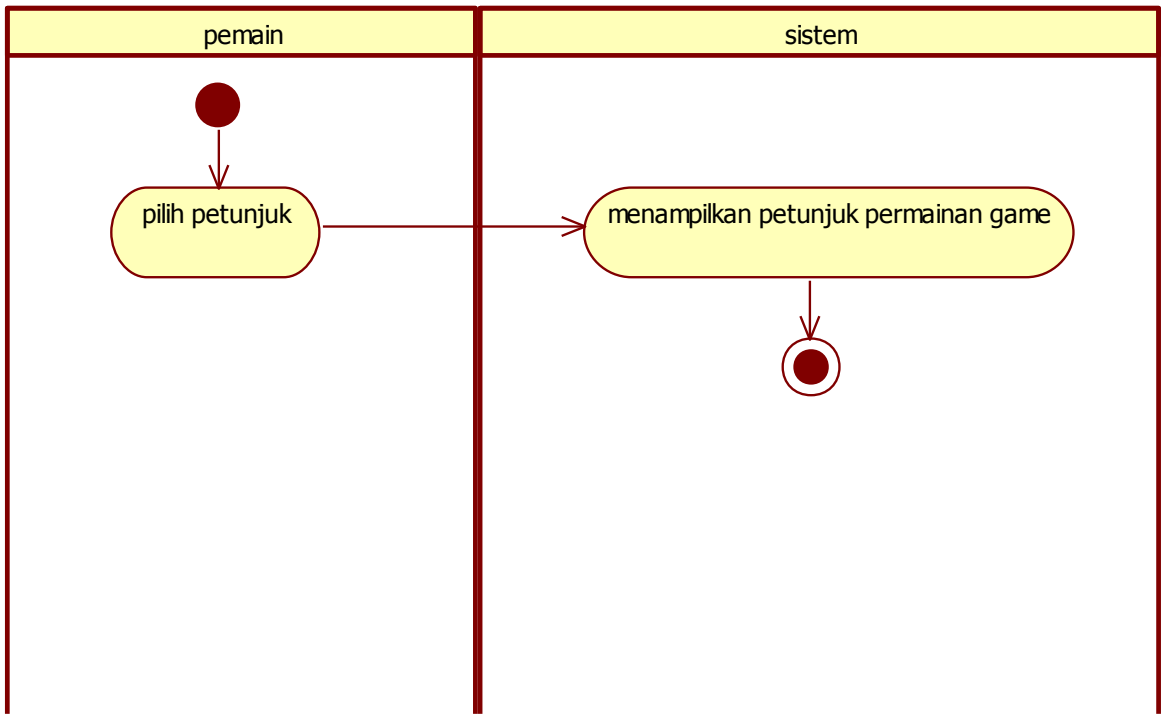
Activity Diagram Level 2



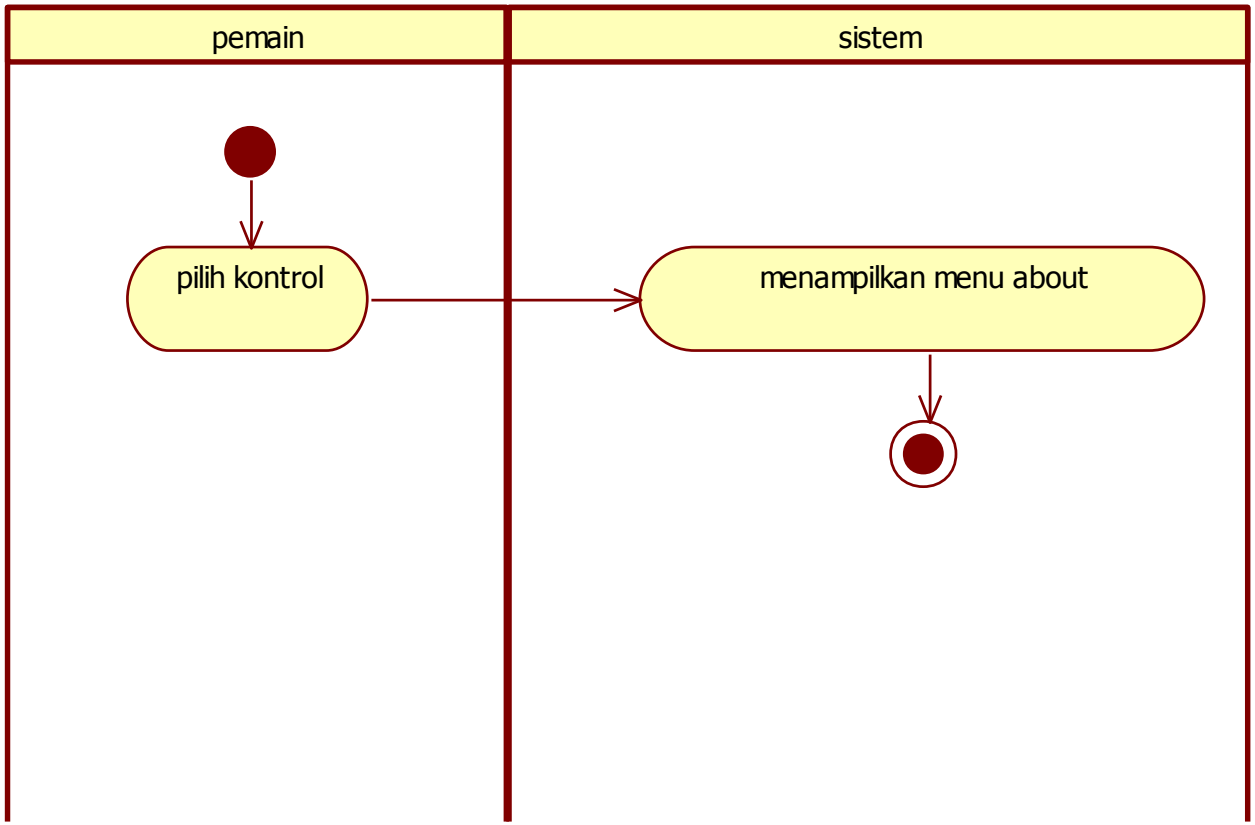
Activity Diagram Level 3



Activity Diagram Lihat petunjuk



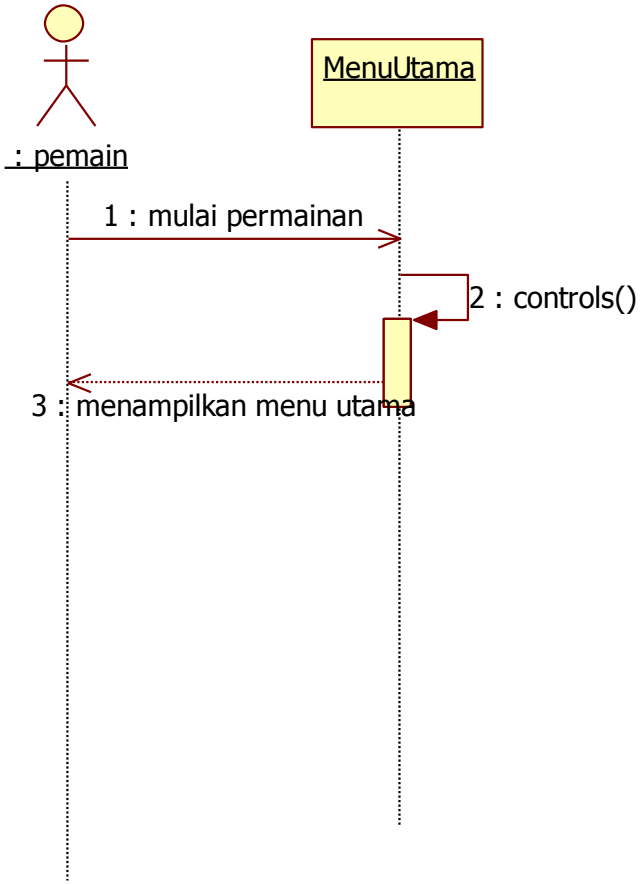
Activity Diagram Lihat about



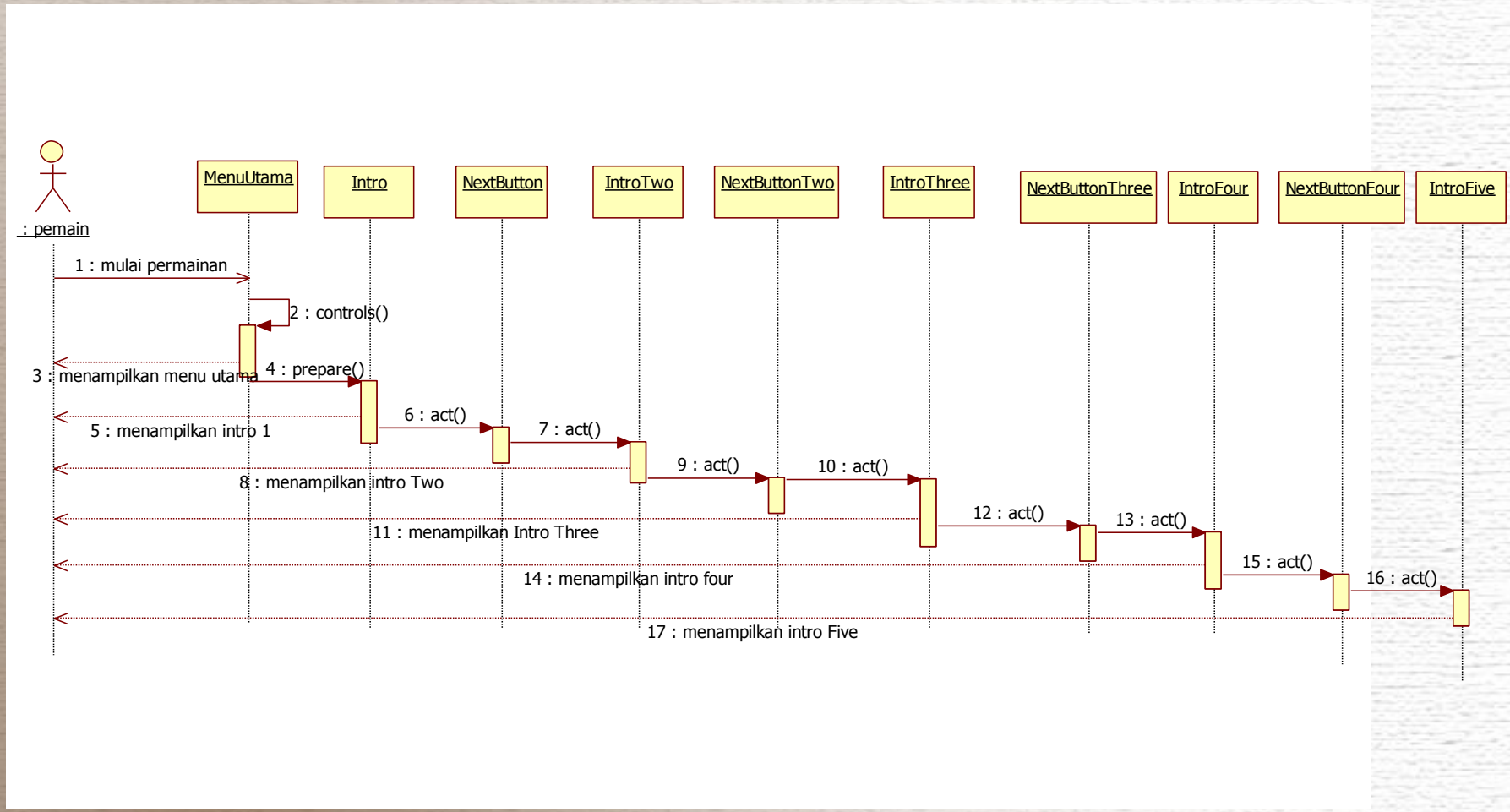
SEQUENCE DIAGRAM

- Sequence diagram mulai permainan baru
- Sequence diagram level 1
- Sequence diagram level 2
- Sequence diagram level 3
- Sequence diagram lihat petunjuk
- Sequence diagram lihat about

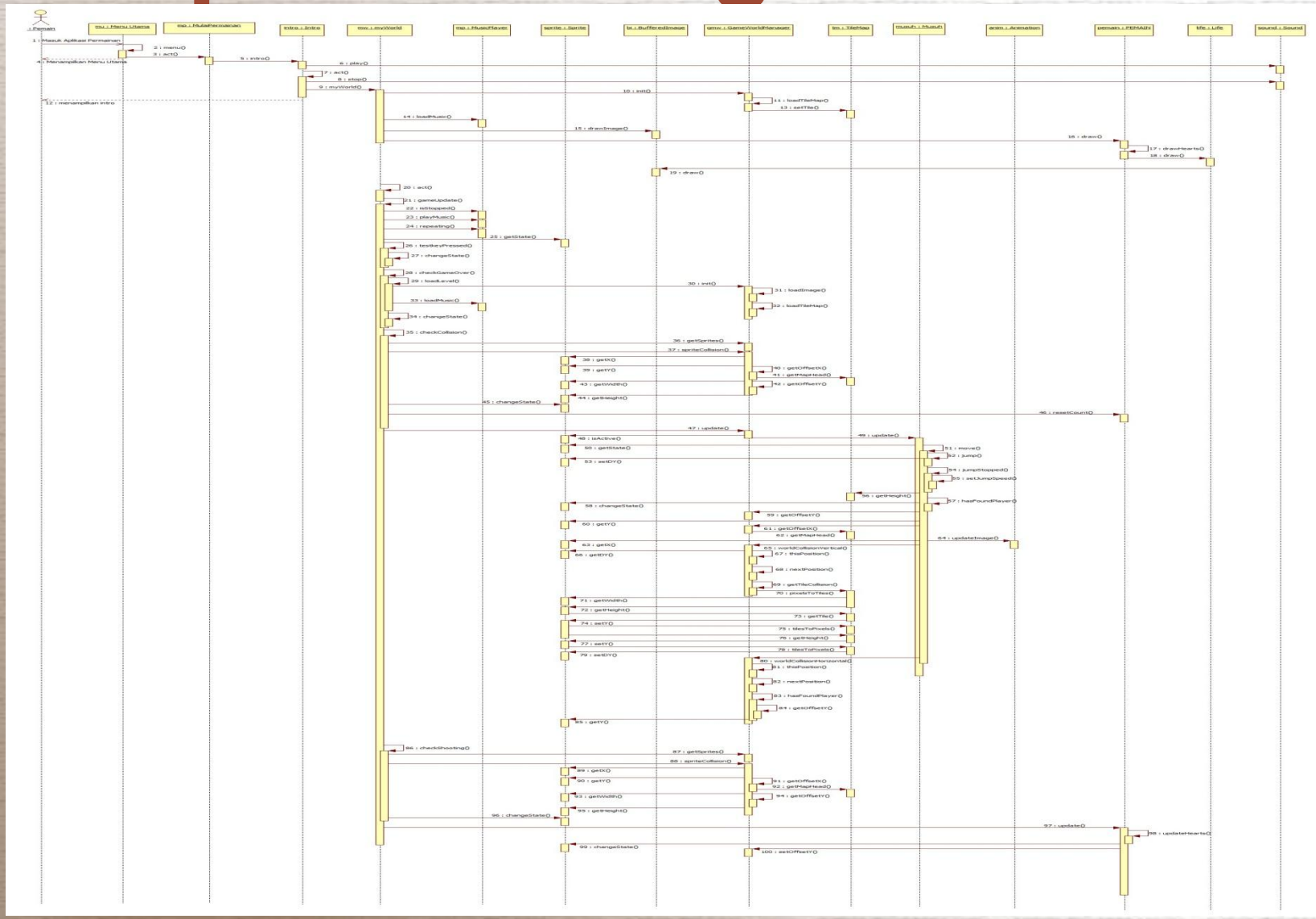
Sequence Diagram Mulai Permainan Baru



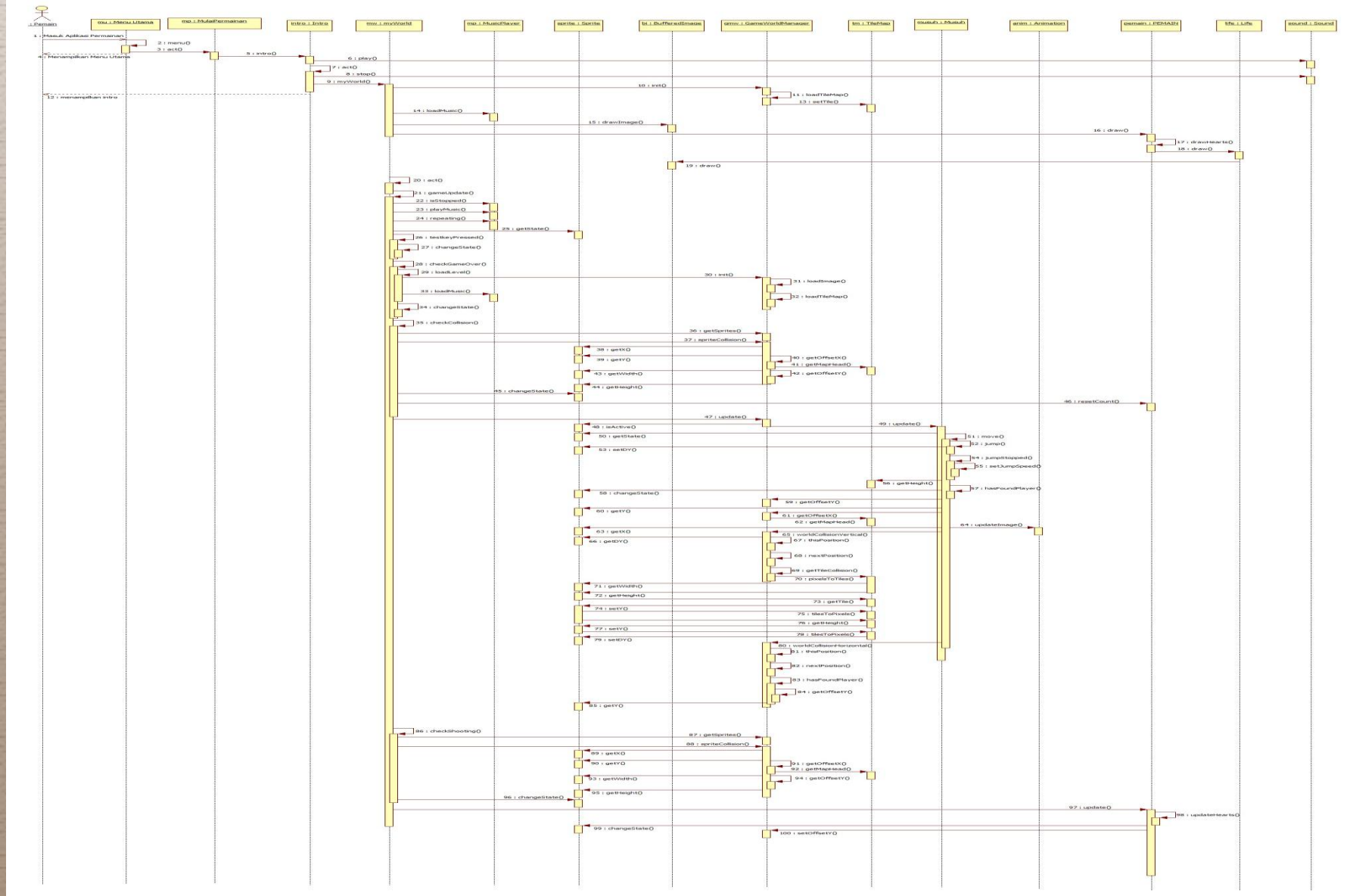
Sequence Diagram Intro



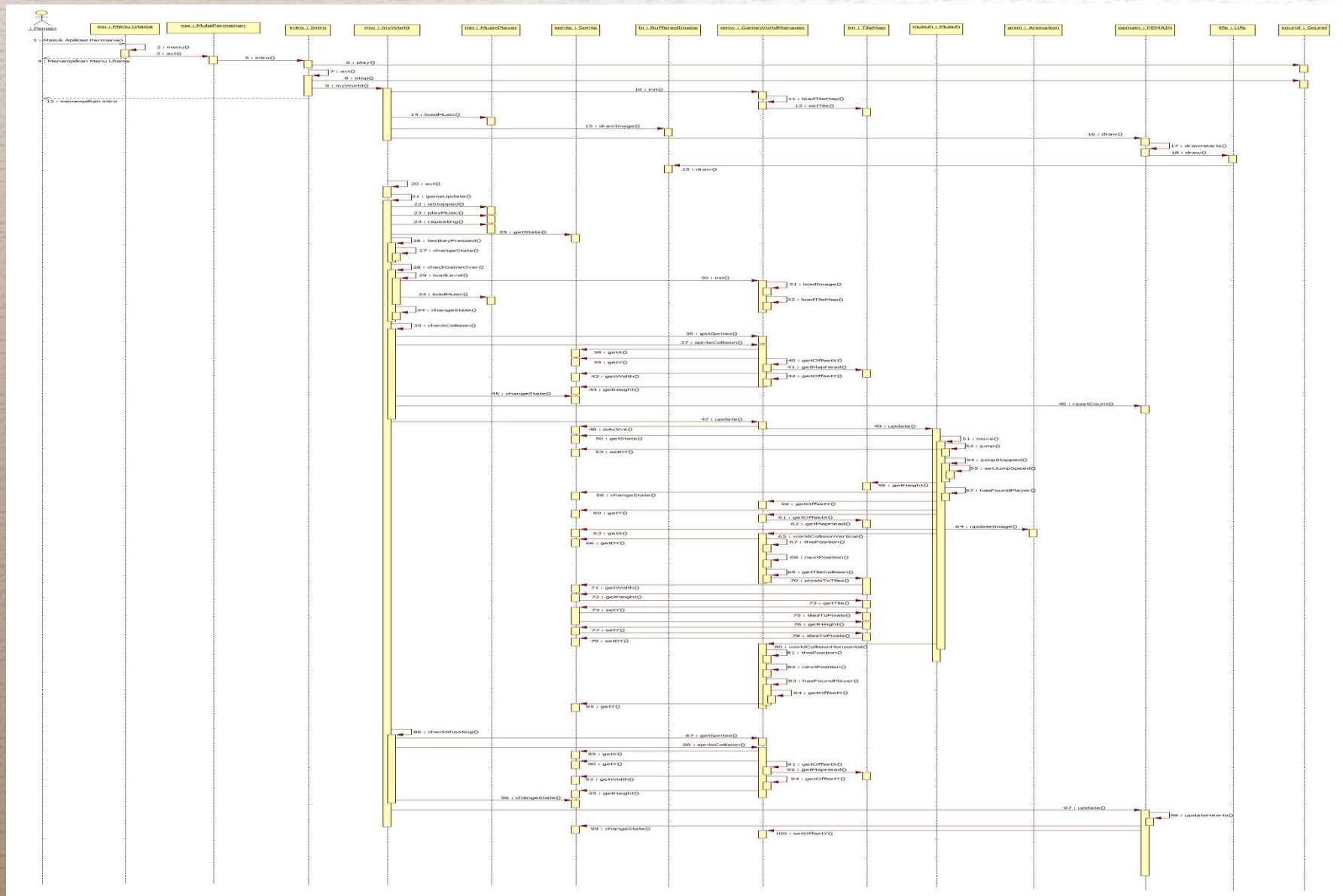
Sequence Diagram Level 1



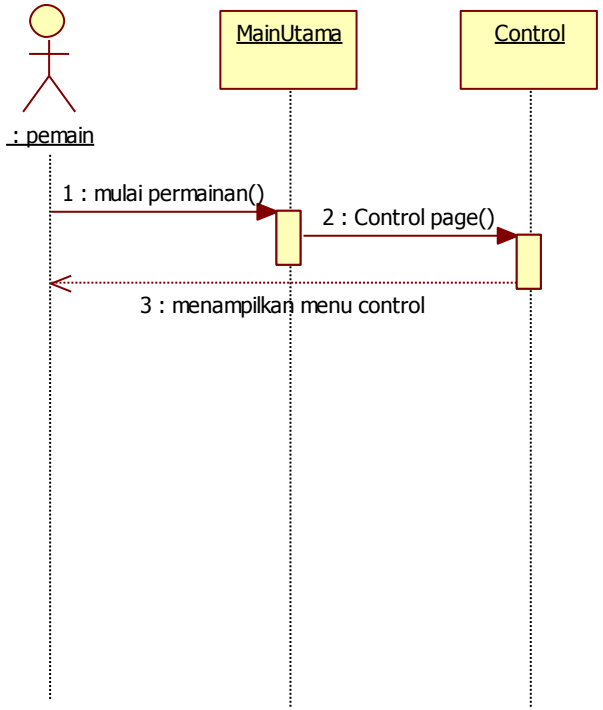
Sequence Diagram Level 2



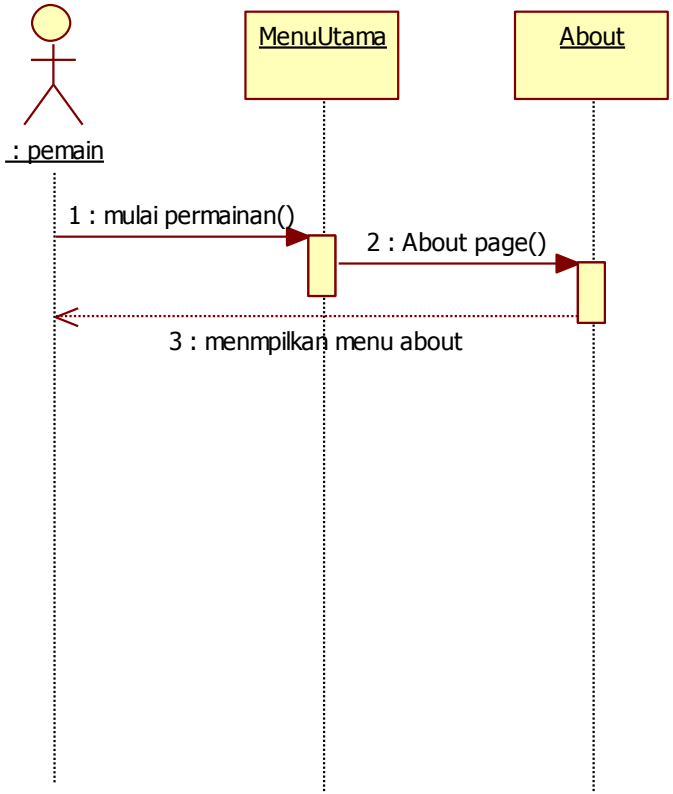
Sequence Diagram Level 3



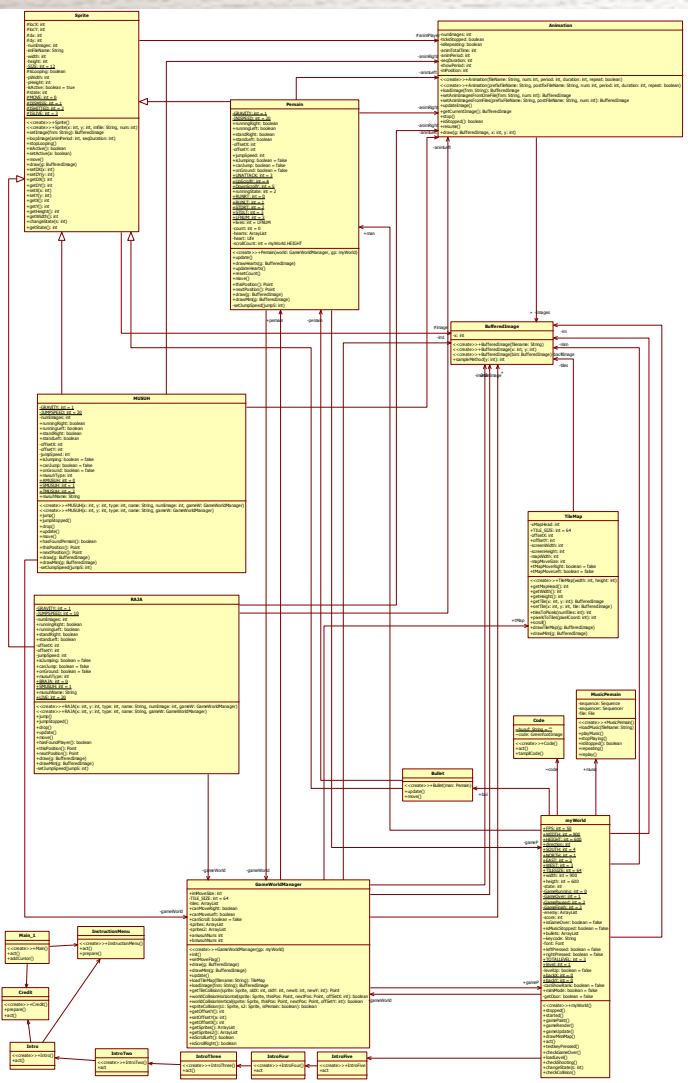
Sequence Diagram Lihat Petunjuk



Sequence Diagram Lihat About



Class Diagram



Implimentasi

- **Implementasi Antar Muka**

Implementasi Antar Muka



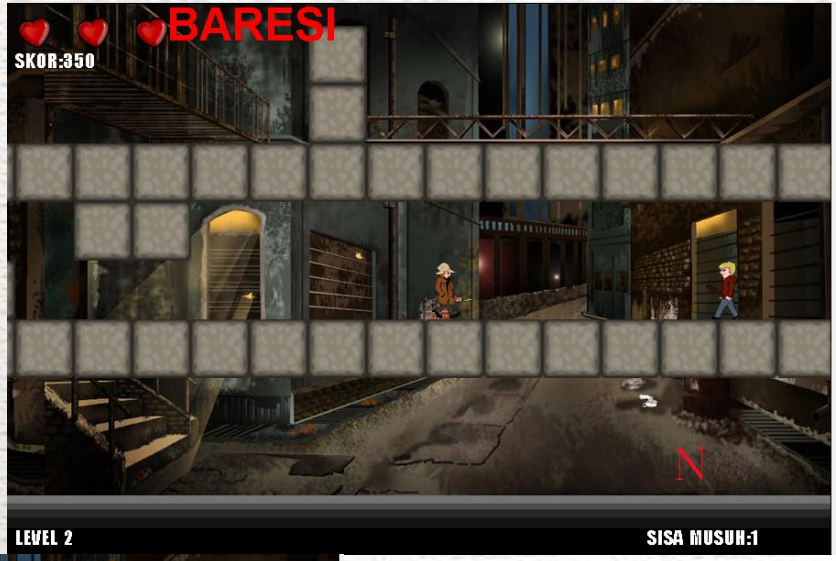






Implementasi Antar Muka







Implementasi Antar Muka





Pengujian sistem

1. **Pengujian *Alpha***
2. ***Pengujian Betha***

Pengujian Alpha

- Pengujian Black Box
- Pengjian white Box

Pengujian Black Box

- Pengujian mulai permainan
- Pengujian Tampil Kontrol
- Pengujian Tampil About

Pengujian mulai permainan

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih Mulai	Menampilkan *Intro 1*	Dapat menampilkan *Intro 1*	[√] Diterima [ ] Ditolak
Pilih lanjut intro 1	Menampilkan intro 2	Dapat menampilkan intro 2	[√] Diterima [ ] Ditolak
Pilih lanjut intro 2	Menampilkan intro 3	Dapat menampilkan intro 3	[√] Diterima [ ] Ditolak
Pilih lanjut intro 3	Menampilkan intro 4	Dapat menampilkan intro 4	[√] Diterima [ ] Ditolak
Pilih lanjut intro 4	Menampilkan intro 5	Dapat menampilkan intro 5	[√] Diterima [ ] Ditolak
Pilih lanjut intro 5	Menampilkan map level 1	Dapat menampilkan map level 1	[√] Diterima [ ] Ditolak
Menyelesaikan level 1	Menampilkan map level 2	Dapat menampilkan map level 2	[√] Diterima [ ] Ditolak
Menyelesaikan level 2	Menampilkan map level 3	Dapat menampilkan keterangan menang	[√] Diterima [ ] Ditolak

Pengujian Tampil kontrol

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih Kontrol	Menampilkan informasi mengendalikan permainan	Dapat menampilkan informasi mengendalikan permainan	[√] Diterima [ ] Ditolak
Pilih kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Diterima [ ] Ditolak
Pilih skip	Menampilkan intro 1	Dapat menampilkan intro 1	[√] Diterima [ ] Ditolak

Pengujian Tampil About

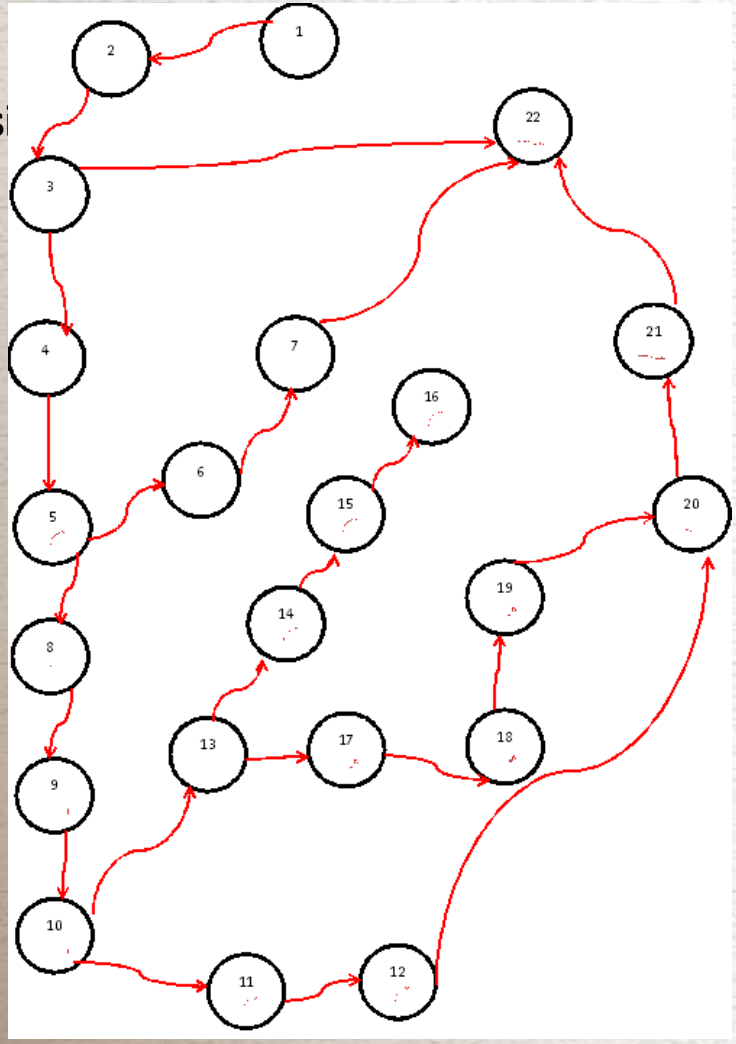
Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih About	Menampilkan informasi pembuat aplikasi permainan	Dapat menampilkan informasi pembuat aplikasi permainan	[√] Diterima [ ] Ditolak
Klik kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Diterima [ ] Ditolak
Klik skip	Menampilkan intro 1	Dapat menampilkan intro 1	[√] Diterima [ ] Ditolak

White Box

- Algoritma A*
- Collision Detction

White Box Algoritma A*

- **Kerangan :**
- **= Menggambarkan kondisi**
- **= Menggambarkan aksi**



White Box-Cyclomatic Complexity (V(G)

- V(G) = E – N + 2
- V(G) = 26 – 22 + 2
- V(G) = 6
- Di mana:
- E = jumlah edge pada grafik alir
- N = jumlah node pada grafik alir

Independent path

- Path 1 = 1-2-3-4-5-6-7-22
- Path 2 = 1-2-3-22
- Path 3 = 1-2-3-4-5-8-9-10-11-12-20-21-22
- Path 4 = 1-2-3-4-5-6-7-8-9-10-13-14-15-16-20-21-22
- Path 5 = 1-2-3-4-5-8-9-10-13-17-18-19-20-21-22
- Path 6 = 1-2-3-4-5-8-9-10-11-12-20-9-10-11-12-20-21-22

Graph Matrices

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Hasil
1		1																					0
2			1																				0
3				1																		1	1
4					1																		0
5						1		1															1
6							1																0
7																						1	0
8									1														0
9										1													0
10											1		1										1
11												1											0
12																				1			0
13														1			1						1
14															1								0
15																1							0
16																				1			0
17																		1					0
18																			1				0
19																				1			0
20									1												1		1
21																						1	0
22																							0
Hasil																							5

Graph Matrices

- V(G) = Jumlah Graph Matric + 1
- V(G) = 5 + 1
- V(G) = 6

Kesimpulan-white box

- Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 6, maka dapat disimpulkan bahwa pada proses pencarian dengan algoritma A* (A *star*) berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

Pengujian Betha

- Kuesioner
- **Kuesioner disebarkan menggunakan teknik *Simple Random Sampling* kepada 30 orang dari umur 11 tahun ke atas**
- **Kuesioner terdiri dari 6 pernyataan dengan menggunakan skala 1 sampai 5.**

$$P = \frac{S}{skorideal} x100\%$$

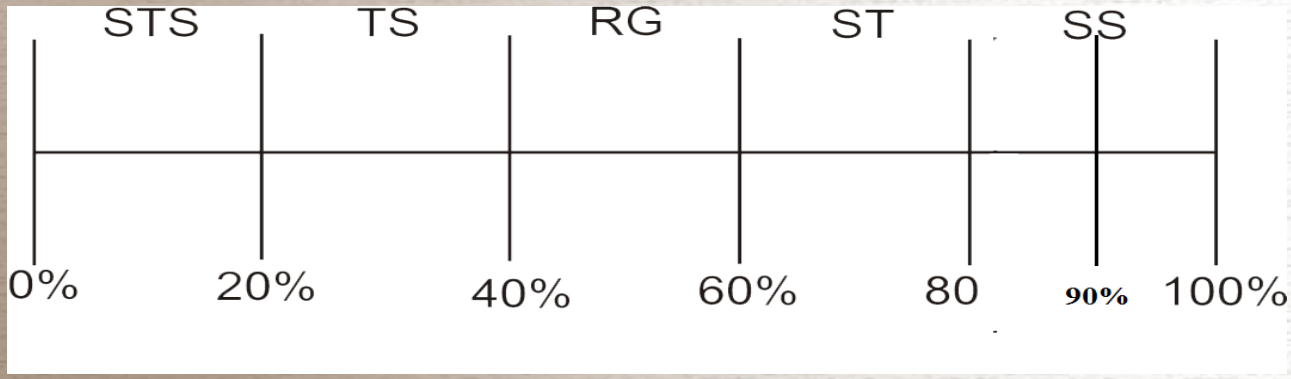
P = Nilai persentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang di tetapkan jawaban

Skorideal = Nilai tertinggi dikalikan dengan jumlah sampel

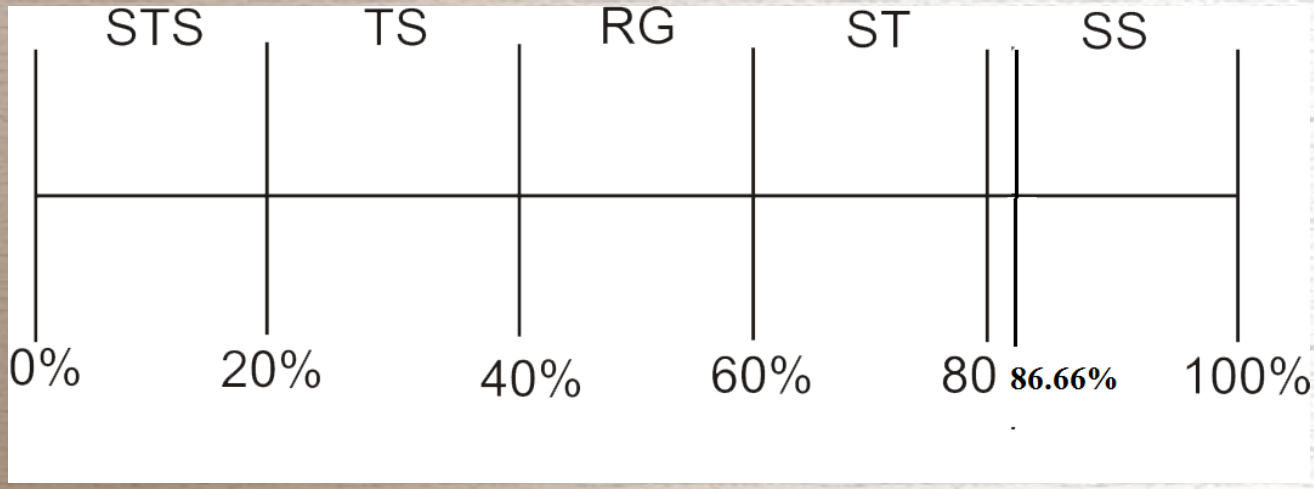
Interpretasi pertanyaan level ke 1

1. Tampilan aplikasi permainan *Detective Code out* secara umum sudah baik



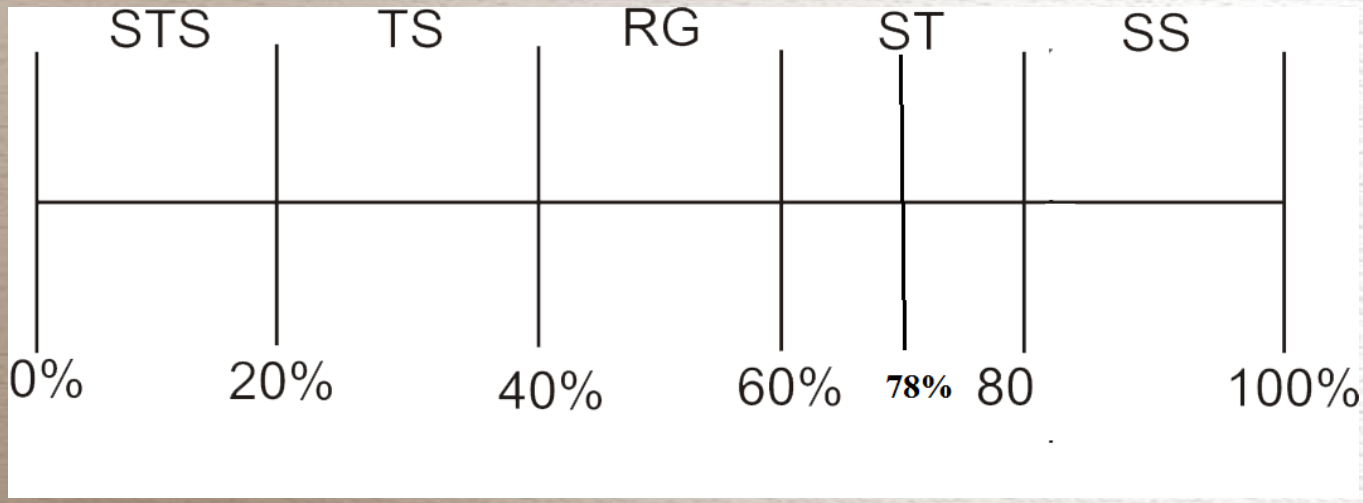
Interpretasi pertanyaan level ke 2

2. Musuh yang terdapat pada aplikasi permainan *Detective code out* sudah cerdas



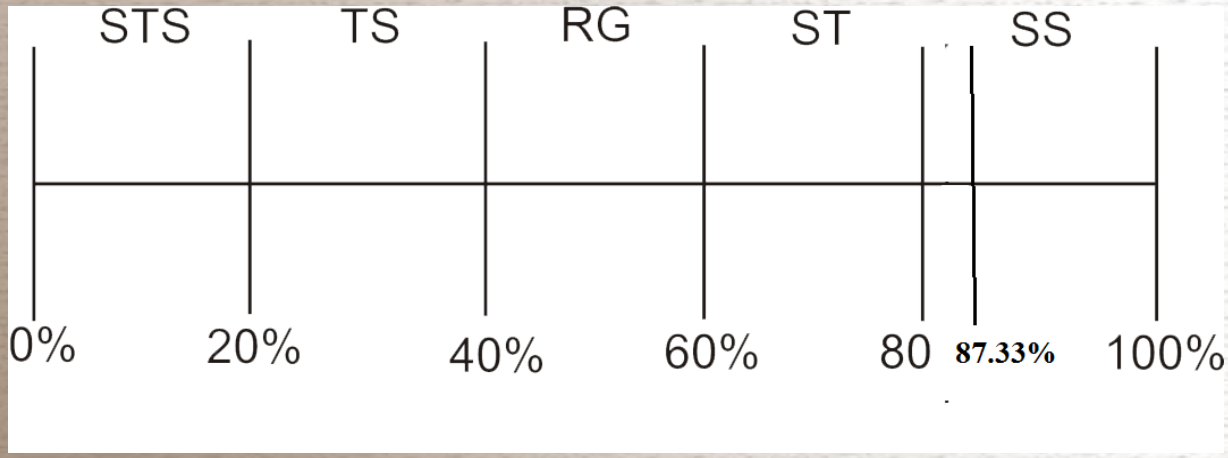
Interpretasi pertanyaan level ke 3

3. Musuh yang terdapat pada aplikasi permainan *Detective code out* sudah cerdas



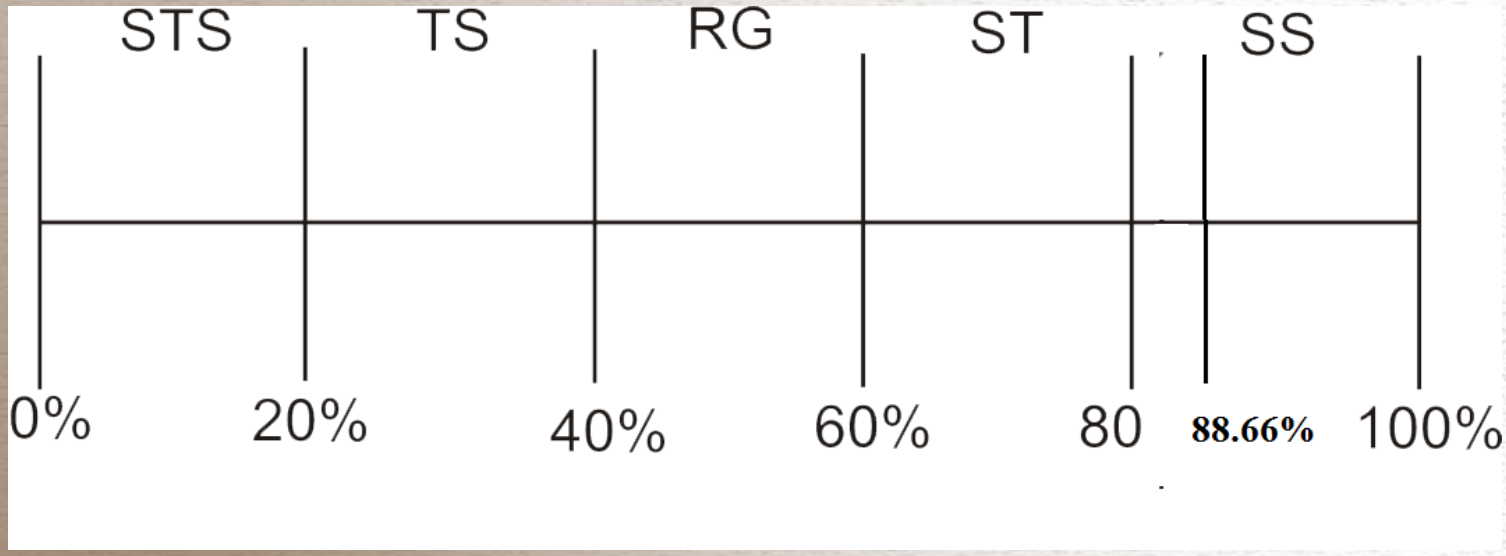
Interpretasi pertanyaan level ke 4

4. Cara pengoprasian dari mulai menu sampai menuju ke arena permainan *Detective code out* mudah dimengerti.



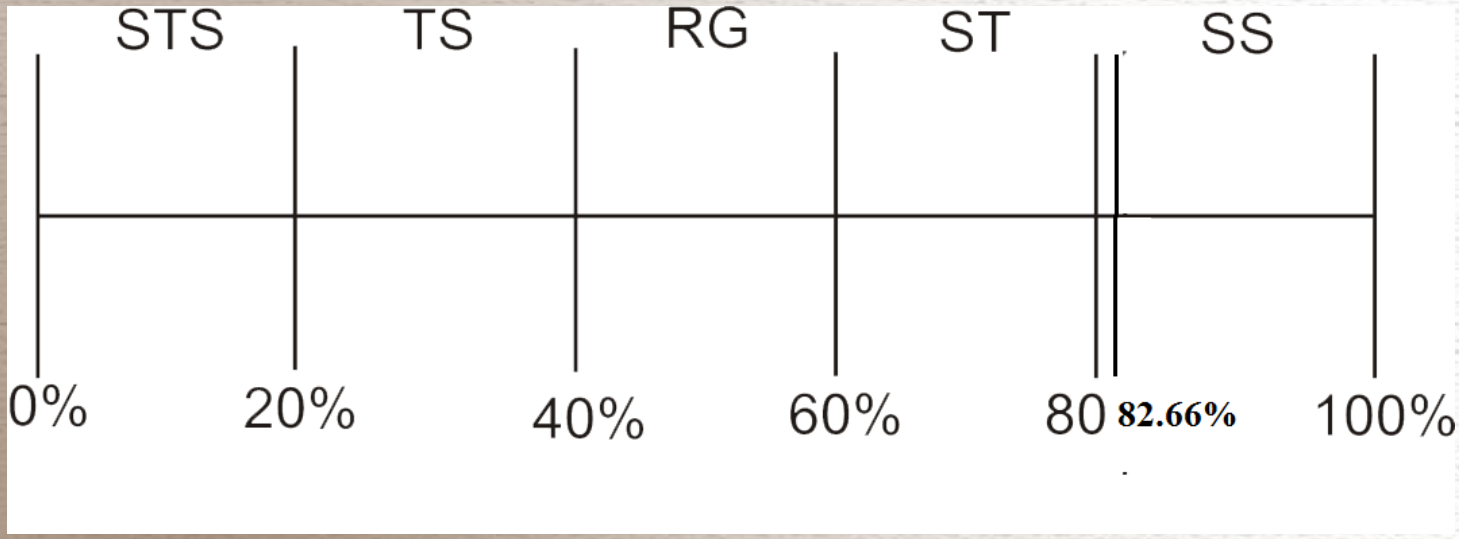
Interpretasi pertanyaan level ke 5

5. Aplikasi permainan D*etective Code Out* menarik untuk dimainkan



Interpretasi pertanyaan level ke 6

6. Kode-kode yang dikumpulkan mudah dipecahkan.



Kesimpulan

- Gamplay yang terdapat pada aplikasi permainan *Detective code out* sudah sesuai dengan detective.
- Musuh menjadi cerdas dengan mencari jalurnya sendiri untuk menuju lokasi pemain.

saran

- Game bisa dibuat 3 dimensi.
- Musuh bisa lebih sulit untuk dikalahkan.
- Game bisa dikembangkan kearah mobile.

DAFTAR PUSTAKA

- [1] Definition of Platform game , [online] tersedia http://web.archive.org/web/20070211211612/http://www.answerbag.com/t_view/269 (15 september 201)
- [2] Dlugosz, alexander, *Jump.* http://www.thejavahub.net/index.php?topic=1683.msg11436;topicseen#msg11436 (diakses pada tanggl 15 september 2012)
- [3] Sommerville, Ian (2007), *Software Engineering – Eight Edition,* Addison Wesley, Massachussets.

DAFTAR PUSTAKA

- [4] Electronic Arts Inc. 2012. *Medal of Honor Warfighter* http://www.medalofhonor.com/. Diakses pada tanggal 10 Oktober 2012 pukul 22.00 WIB
- [5] Thomas.(2006, Maret). *Genre and game studies : Toward a critical approach to video game genres*. *Simulation & Gaming, Vol. 37.* Retrieved 7 Oktober 2012, from http://www.academia.edu/385966/Genre_and_Game_Studies_Toward_a_Critical_Approach_to_Video_Game_Genres.

DAFTAR PUSTAKA

- [6] Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya).* Yogyakarta : Graha Ilmu.
- [7] Admin. (2012). *Algoritma A*.* http://www.digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=1010:algoritma-a-&catid=21:itp-informatika-teori-dan-pemograman&Itemid=14. diakses tanggal 20 September 2012 .
- [8] Davison, Andrew (2005), *Killer Game Programming in Java*, O'Reilly Media, Sebastopol.

DAFTAR PUSTAKA

- [9] Harbour, Jonathan S. (2008), *Beginning Java Programming : Second Edition,* Thomson Course Technology PTR, Boston.
- [10] Nugroho, Adi. (2005). *Rational Rose untuk Pemodelan Berorientasi Objek.* Bandung: Informatika.
- [11] Fowler, Martin. (2005). *UML Distilled edisi 3 Panduan Singkat Bahasa Pemodelan Object Standar.* Yogyakarta: Andi
- .

DAFTAR PUSTAKA

- [12] Suarga (2009), Dasar Pemrograman Komputer dalam Bahasa Java, Andi Offset, Yogyakarta
- [13] Henriksson, Poul dan Kolling, Michael. Greenfoot-Teach& learn java Programming http://www.greenfoot.org/
- [14] Sugiyono (2010), Metode Penelitian Kuantitatif, Kualitatif, dan R&D, ALFABETA, cv., Bandung.

TERIMA KASIH