BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan kegiatan penguraian suatu sistem informasi yang utuh dan nyata ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk mengidentifikasi serta mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang mungkin terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga mengarah kepada solusi dengan kebutuhan serta perkembangan teknologi.

3.1.1 Analisis *Game* Sejenis

Analisis yang dilakukan merupakan observasi dari game sejenis untuk melihat cara bermain, tujuan dan ada tidaknya kekurangan atau kelebihan *game* itu sendiri. Hasil observasi tersebut untuk membantu dalam penelitian. Game yang dianalisis adalah *little farm.*

Little farm merupakan *game* yang ber*genre casual game* dengan mekanisme *matching.* Pada *game* ini pemain bertugas menyiram semua tanaman pada saat yang sama dengan kondisi 4 petak dari tanaman tersebut sejajar atau cocok membentuk segi empat, setelah itu tanaman akan dimasukkan ke dalam truk stok tanaman dan menghasilkan *score*.

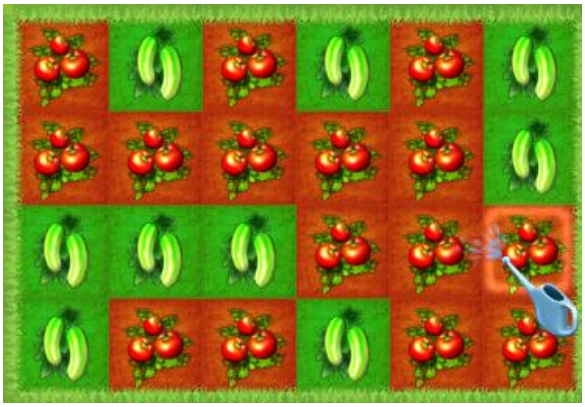
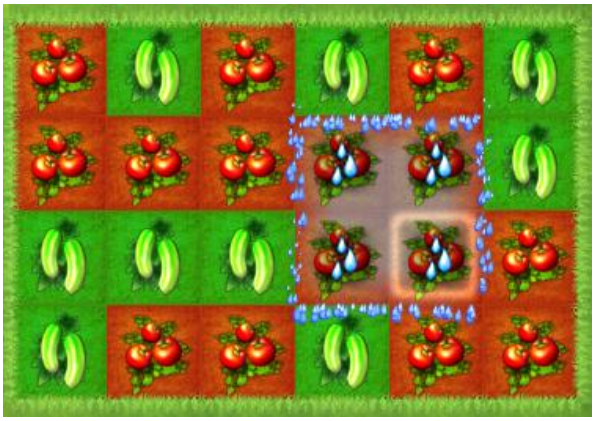


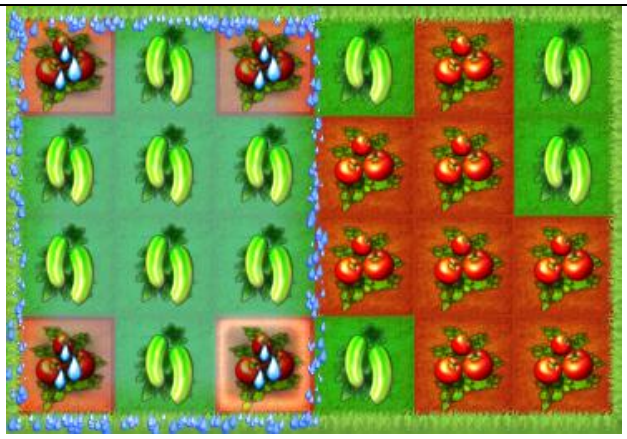
Gambar 3-1 *Game Little Farm* [18]

Pada *game* ini *player* bisa dinyatakan naik *level* jika sudah mengisi stok truk tanaman hingga penuh. Waktu permainan pada setiap *level*nya relatif lama dan sama. *player* tidak mengetahui batas waktu pada setiap *level*nya.

Berikut skenario *game little farm* :

Tabel 3-1 Skenario *game little farm*

Skenario *Little Farm*
1. Pada awal *game* pemain memasukkan nama. 2. Pada permainan lahan sudah disediakan dan siap untuk disiram 3. Pada setiap petak terdapat tanaman yang berbeda dan pada lahan terdapat 2 tanaman yang berbeda 4. Pemain diminta untuk menyirami tanaman yang akan dijual dengan aturan mengidentifikasi kotak pencocokan tanaman di lahan yang dapat digunakan yang berbentuk empat sudut persegi atau persegi panjang.  



5. Syarat naik *level* adalah mengisi stok truk hingga penuh. Setiap stok truk dinyatakan penuh jika tanaman yang masuk sejumlah 30 buah. Pemain diberikan kesempatan untuk mengisi stok truk tanaman sebelum waktu permainan selesai. Pada setiap *level* jumlah truk yang harus dipenuhi berbeda. Waktu permainan yang diberikan pada setiap level sama, yaitu 10 menit.



6. Pada setiap *level* jumlah truk yang harus dimuat menjadi *goal* pada setiap *level*.



7. Kemudian jika stok truk penuh, maka permainan pada *level* tersebut selesai dan hasil skor ditampilkan dilayar.



Kemudian permainan bisa dilanjutkan ke *level* berikutnya.

3.1.2 Analisis Masalah

Berdasarkan analisis yang dilakukan terhadap *game* yang sejenis yaitu *game little farm* dapat disimpulkan bahwa dalam game tersebut terdapat beberapa hal yang kurang diantaranya syarat untuk naik *level* hanya dengan mengisi stok truk hingga penuh sebelum waktu permainan selesai. Selain itu batas waktu permainan yang seragam sehingga pemain akan lebih santai dalam menyelesaikan permainan pada setiap *level*. Hal ini menyebabkan permainan tidak terlalu menantang untuk diselesaikan. Untuk itu diperlukan *rule gameplay* yang lebih bervariasi pada *game* ini agar *game* menjadi lebih menarik untuk diselesaikan. Metode *forward chaining* yang berbasis aturan berfungsi untuk membuat *rule gameplay* yang lebih bervariasi pada *game* ini.

3.1.3 Analisis Game yang Akan Dikembangkan

Analisis *game* yang akan dikembangkan merupakan bagian yang mendeskripsikan *game* yang akan dikembangkan. Pada bagian ini terdiri dari arsitektur, *storyline*, *gameplay*.

3.1.3.1 Arsitektur Game

Aplikasi ini dibangun dalam bentuk dua dimensi (2D), serta memiliki fitur seperti pada Tabel 3-2.

Tabel 3-2 Fitur *Game*

Fitur *Game*
1. Sistem *single player*
2. Jenis tanaman yang disirami 6 tanaman
3. Pada setiap *level*, *player* bisa menyirami 2 jenis tanaman
4. Penyelesaian *game* pada tiap *level* memiliki bermacam-macam kombinasi
5. Pada setiap *level player* bisa memperoleh *gold medal* atau *silver medal*
6. Di akhir *game player* bisa memperoleh *gold trophy* atau *silver trophy*

3.1.3.2 *Storyline*

Storyline merupakan cerita pada game dimana pada *storyline* sebuah *game* akan ditentukan bagaimana desainnya dan aturan permainannya. Pada game ini, pemain akan berperan sebagai seorang pemuda yang diberi lahan perkebunan yang merupakan warisan dari kakeknya. Lahan perkebunan tersebut sudah memiliki banyak tanaman yang bisa diolah. Pemain diminta untuk mengurus lahan perkebunan tersebut dengan menyiraminya dan menjualnya.

Pada suatu saat ada seorang pengusaha tanaman menantang pemain untuk menjual tanaman dengan jumlah yang telah ditentukan dengan batas waktu tertentu. Jika tantangan tersebut berhasil diselesaikan maka pemain memperoleh *trophy* dan gelar sebagai "*little farmer*". Disanalah mulai permainan seorang pemuda yang menyirami tanamannnya dan menjualnya sesuai dengan tantangan yang diberikan oleh pengusaha. Berdasarkan cerita tersebut maka game ini berjudul "*little farmer*".

3.1.3.3 *Gameplay*

Gameplay dilakukan untuk menggambarkan aturan-aturan dalam *game*. *Gameplay* yang akan dibangun terdapat pada tabel 3-3.

Tabel 3-3 *Gameplay*

Gameplay
1. Di awal *game player* memilih menu mulai 2. Kemudian *game* mengenalkan cara bermain pada *game little farmer.* 3. Lalu pemain siap untuk bermain 4. Pada setiap *level* pemain diberikan lahan dengan petak 4 x 6 5. Pada setiap petak terdapat tanaman yang berbeda dan pada lahan terdapat 2 tanaman yang berbeda 6. Pemain diminta untuk menyirami tanaman yang akan dijual dengan aturan 4 tanaman harus berada pada posisi yang cocok dan sejajar membentuk persegi atau persegi panjang.  



7. Pemain diberikan target penyelesaian permainan dengan kombinasi tertentu dan waktu tertentu.
8. Pada setiap level memiliki waktu permainan yang berbeda untuk level 1 diberikan waktu 30 detik, untuk level 2 diberikan waktu 45 detik dan untuk level 3 diberikan waktu 60 detik.
9. Pada *game little farmer* skor pada masing-masing tanaman berbeda. Pada level 1 Jika pemain menyiram dan menjual tomat akan mendapatkan skor 20 untuk setiap tomatnya, untuk terong akan mendapatkan skor 25 untuk setiap terongnya. Pada level 2 Jika pemain menyiram dan menjual timun akan mendapatkan skor 30 untuk setiap timunnya, labu akan mendapatkan skor 40 untuk setiap labunya. Pada Level 3 Jika pemain menyiram dan menjual kubis akan mendapatkan skor 50 untuk setiap kubisnya, dan jagung akan mendapatkan skor 60 untuk setiap jagungnya.
10. Pada setiap *level* memiliki kombinasi tersendiri untuk diselesaikan.
11. Pemain bisa mengumpulkan skor dan menempatkan dirinya ke papan skor tertinggi.
12. Di akhir jika pemain berhasil maka akan mendapatkan gelar *little farmer*

Berikut ini tabel perbandingan target penyelesaian permainan pada *game sejenis* dan *game* yang akan dikembangkan :

Tabel 3-4 Tabel perbandingan target penyelesaian permainan pada *game sejenis* dan *game* yang akan dikembangkan

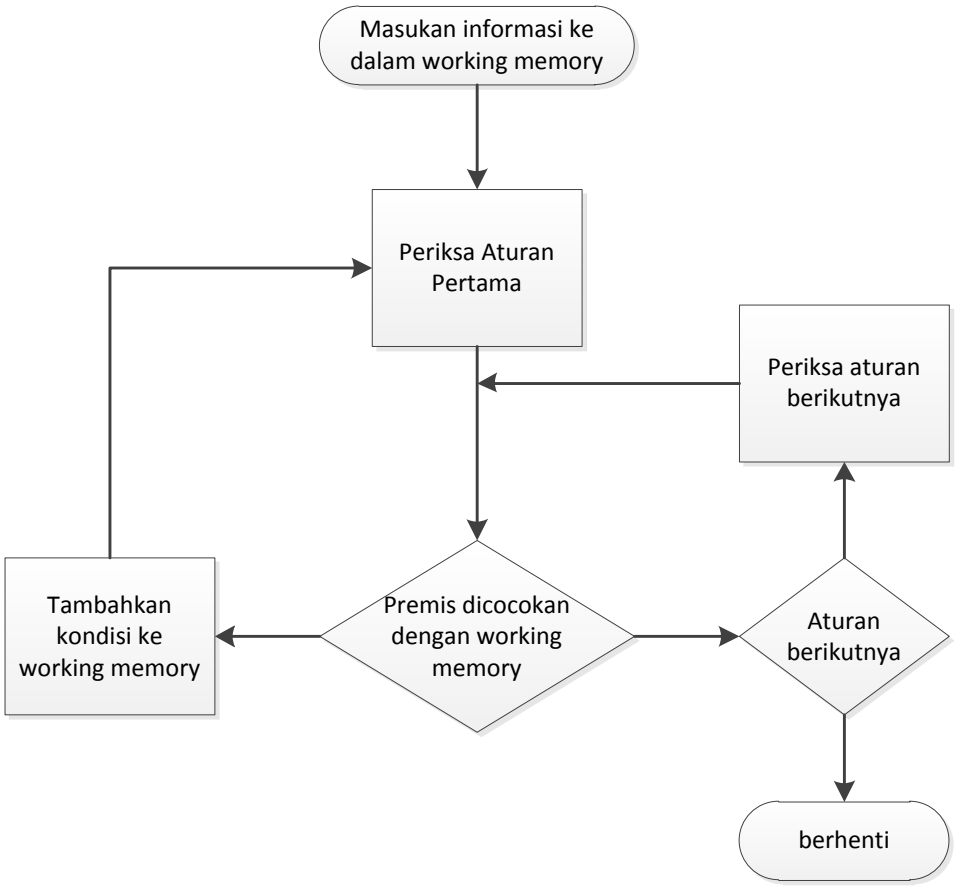
***Game* sejenis**				***Game* yang akan dikembangkan**			
Level	**Waktu**	**Kombinasi Penyelesaian**	***Reward***	***Level***	**Waktu**	**Kombinasi penyelesaian**	***Reward***
1-1	10 Menit	menjual tanaman yang dimasukkan ke 1 truk (1 truk = 30 tanaman)	Mendapatkan skor dan melanjutkan ke level berikutnya	1	30 Detik	Menjual tomat ≥ 30	*Silver Medal*
						Menjual terong ≥ 40 **dan** Mendapatkan skor ≥ 1000	*Gold Medal*
1-2	10 Menit	Menjual tanaman yang dimasukkan ke 2 truk (2 truk = 60 tanaman)	Mendapatkan skor dan melanjutkan ke level berikutnya	2	45 Detik	Menjual timun ≥ 40 **dan** Mendapatkan skor ≥ 1200	*Silver Medal*
						Menjual labu ≥ 40 **dan** Mendapatkan skor ≥ 1600	*Gold Medal*
1-3	10 Menit	Menjual tanaman yang dimasukkan ke 3 truk (3 truk=90 tanaman)	Mendapatkan skor dan melanjutkan ke level berikutnya	3	60 Detik	Menjual kubis ≥ 40 **dan** Mendapatkan skor ≥ 2000	*Silver Trophy*
						Menjual jagung ≥ 50 **dan** Mendapatkan skor ≥ 3000	*Gold Trophy*

3.1.4 Analisis Metode

Metode yang digunakan dalam pengembangan *game little farmer* adalah metode *forward chaining*. Metode *forward chaining* merupakan strategi pencarian yang memulai proses pencarian dari sekumpulan data atau fakta, dari data-data

tersebut dicari suatu kesimpulan yang menjadi solusi dari permasalahan yang dihadapi.

Operasi dari sistem *forward chaining* dimulai dengan memasukan sekumpulan fakta yang diketahui kedalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai gol atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui.



Gambar 3-2 Alur Metode *Forward Chaining*

Setiap informasi pada alur metode *forward chaining* diwakilkan dengan premis, kemudian premis tersebut dikombinasikan membentuk sebuah *rule* yang digunakan.

Seperti pada yang dijelaskan pada *gameplay little farmer* bahwa terdapat kombinasi tertentu untuk menyelesaikan *game* pada setiap *level*nya. Dan pemain diminta untuk menyirami tanaman dan setelah disiram maka otomatis tanaman yang disiram akan terjual. Sehingga pada setiap *level*nya akan melihat hasil

review yang diperoleh oleh *player* pada *level* tersebut. Hal ini akan menjadi premis dalam *rule* yang akan digunakan.

Premis menurut kamus besar bahasa indonesia adalah kalimat yang dijadikan dasar penarikan kesimpulan di dalam logika. Premis menjadi dasar pengetahuan yang digunakan pada metode *forward chaining*.

Berikut premis yang digunakan dalam *game little farmer*:

Tabel 3-5 Tabel Premis

Variabel	Premis
P1	menjual tomat lebih dari atau sama dengan 30
P2	Mendapatkan skor *level* 1 lebih dari atau sama dengan 1000
P3	menjual terong lebih dari atau sama dengan 40
P4	Silver medal *level* 1
P5	Gold medal *level* 1
P6	*Level* 1 selesai
P7	menjual timun lebih dari atau sama dengan 40
P8	Mendapatkan skor *level* 2 lebih dari atau sama dengan 1200
P9	menjual labu lebih dari atau sama dengan 40
P10	Mendapatkan skor *level* 2 lebih dari atau sama dengan 1600
P11	Silver medal *level* 2
P12	Gold medal *level* 2
P13	*Level* 2 selesai
P14	menjual kubis lebih dari atau sama dengan 40
P15	menjual jagung lebih dari atau sama dengan 50
P16	Mendapatkan skor *level* 3 lebih dari atau sama dengan 2000
P17	Mendapatkan skor *level* 3 lebih dari atau sama dengan 3000
P18	Silver trophy
P19	Gold trophy
P20	Game selesai

P1 : Premis 1 P2 : Premis 2

P3 : Premis 3

P4 : Premis 4

P5 : Premis 5

P6 : Premis 6

P7 : Premis 7

P8 : Premis 8

P9 : Premis 9

P10 : Premis 10

P11 : Premis 11

P12 : Premis 12

P13 : Premis 13

P14 : Premis 14

P15 : Premis 15

P16 : Premis 16

P17 : Premis 17

P18 : Premis 18

P19 : Premis 19

P20 : Premis 20

Berikut ini deskripsi *rule* :

Tabel 3-6 Deskripsi *Rule*

Nomor *Rule*	Deskripsi *Rule*
R1	**IF** P1 **THEN** P4
R2	**IF** P2 **AND** P3 **THEN** P5
R3	**IF** P4 **THEN** P6
R4	**IF** P5 **THEN** P6
R5	**IF** P7 **AND** P8 **THEN** P11
R6	**IF** P9 **AND** P10 **THEN** P12
R7	**IF** P11 **THEN** P13
R8	**IF** P12 **THEN** P13
R9	**IF** P14 **AND** P16 **THEN** P18
R10	**IF** P15 **AND** P17 **THEN** P19
R11	**IF** P18 **THEN** P20
R12	**IF** P19 **THEN** P20

R1 : *Rule* 1

R2 : *Rule* 2

R3 : *Rule* 3

R4 : *Rule* 4

R5 : *Rule* 5

R6 : *Rule* 6

R7 : *Rule* 7

R8 : *Rule* 8

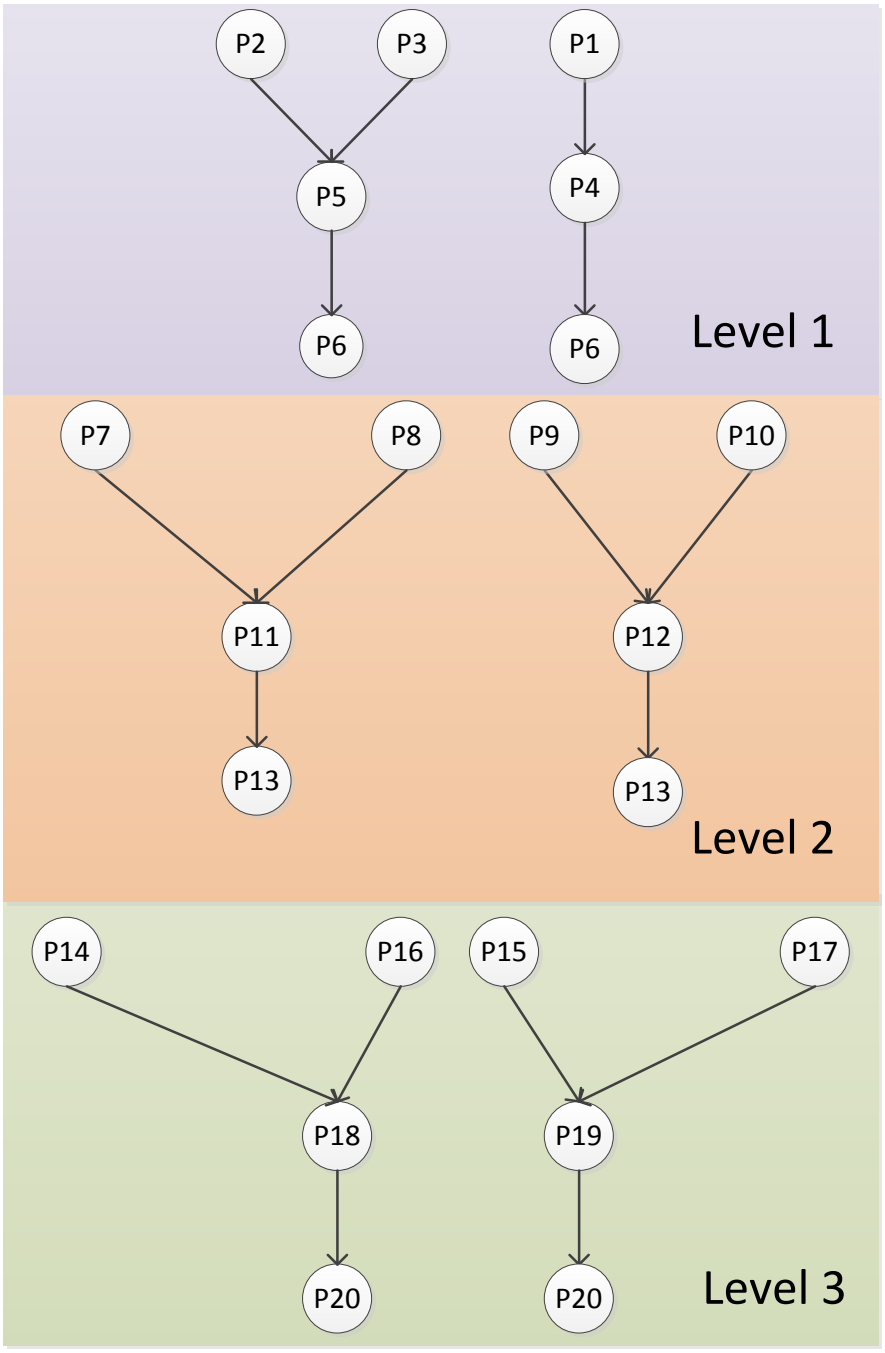
R9 : *Rule* 9

R10 : *Rule* 10

R11 : *Rule* 11

R12 : *Rule* 12

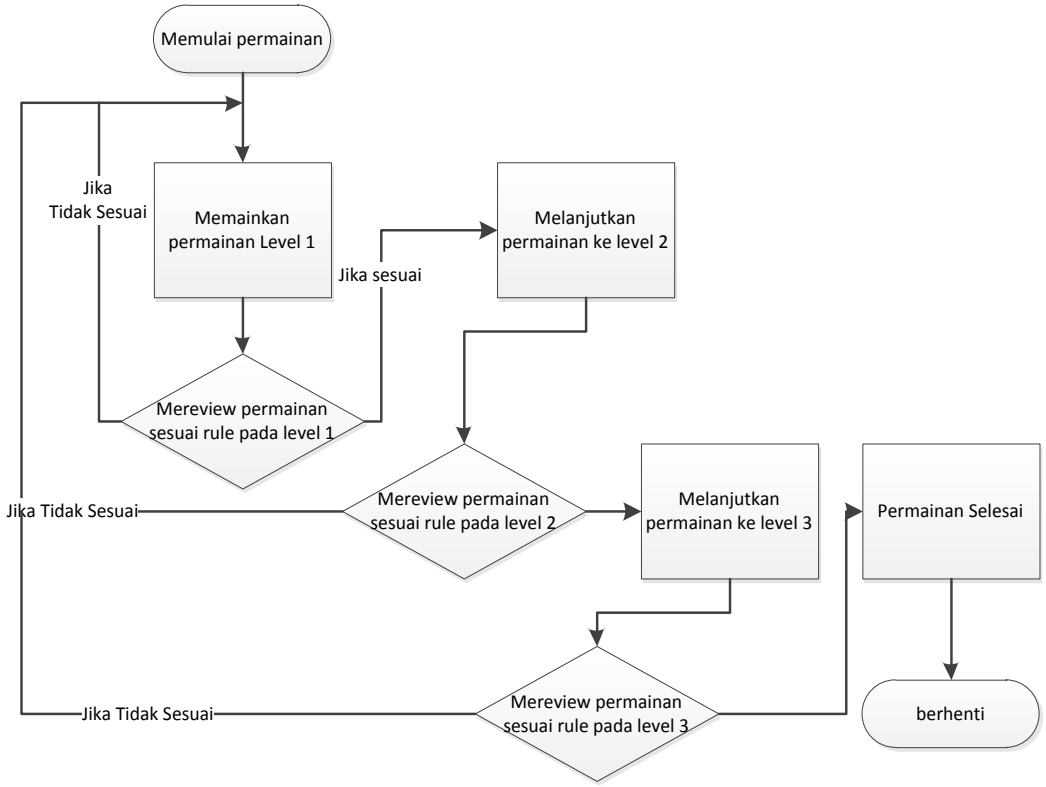
Berikut ini gambaran alur *rule game little farmer* :



Gambar 3-3 *Rule Game little Farmer*

3.1.4.1 Analisis *Rule*

Analisis *rule* meliputi alur penerapan *rule* pada *game little farmer* dan gambaran penggunaan *rule* dalam *metode forward chaining.* Berikut ini alur penerapan *rule* pada *game little farmer* :



Gambar 3-4 Alur penerapan *rule* pada *game little farmer*

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem.Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis perangkat pikir/*user*.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras digunakan untuk mengetahui spesifikasi minimun perangkat keras dalam membangun *game little farm* pada tabel 3-7 dan menjalankan *game little farm* pada tabel 3-8.

Tabel 3-7 Spesifikasi Perangkat Keras Pembangun

Nama Perangkat	Spesifikasi
Processor	1.80Ghz

RAM	4GB
Hard disk	50GB
Display	Resolusi 1366 x 768
Keyboard	Standar
Mouse	Standar

Tabel 3-8 Spesifikasi Perangkat Keras Pengguna

Nama Perangkat	**Spesifikasi**
Processor	1.80Ghz
RAM	512 MB
Hard disk	100 MB
Display	Resolusi 800 x 600
Keyboard	Standar
Mouse	Standar

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak digunakan untuk mengetahui spesifikasi perangkat lunak dalam menjalankan dan membangun *game litle farmer*, seperti pada tabel 3-9 dan tabel 3-10.

Tabel 3-9 Spesifikasi Perangkat Lunak Pembangun

Nama Perangkat Lunak	**Spesifikasi**
Sistem Operasi	Microsoft Windows XP
Tools	Greenfoot 2.3.0
	JDK 7
	Adobe Photoshop CS3

Tabel 3-10 Spesifikasi Perangkat Lunak Pengguna

Nama Perangkat Lunak	**Spesifikasi**
Sistem Operasi	Microsoft Windows XP, 7
Tools	JDK 7

3.1.5.3 Analisis Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir digunakan untuk mengetahui spesifikasi pengguna untuk dapat memainkan *game little farm*, seperti pada tabel 3-11.

Tabel 3-11 Spesifikasi Pengguna

Peran	Pemain
Usia	7 Tahun ke atas
Tingkat Keterampilan	Dapat menggunakan komputer
Kondisi Fisik	Tidak buta total dan buta warna

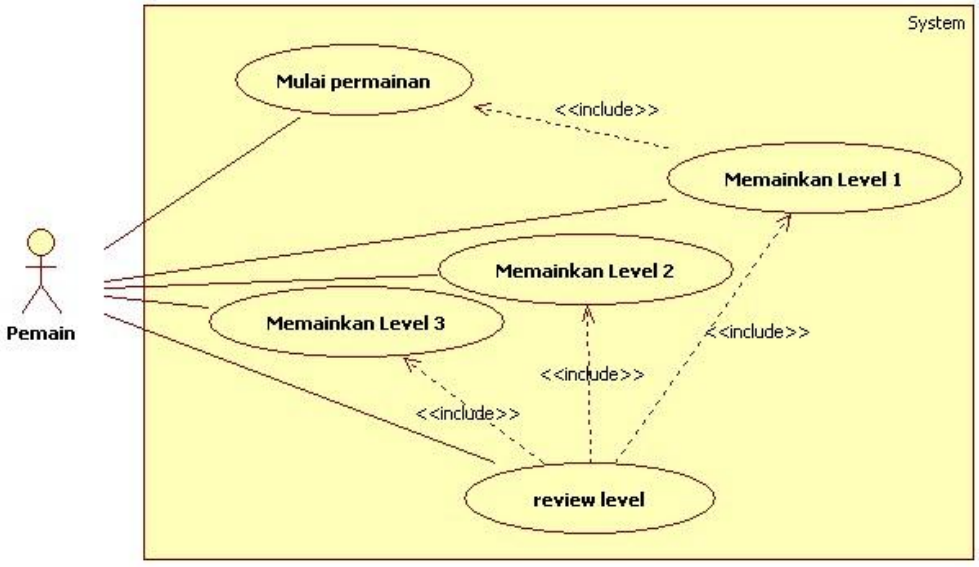
3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui gambaran sistem yang akan dibangun. Analisis kebutuhan fungsional terdiri dari pendefinisan fungsi *game*, penggambaran alir aktivitas pada *game* dan penggambaran interaksi antar objek pada *game*.

3.1.6.1 Pendefinisian Fungsi *game*

A. *Use Case Diagram*

Penggambaran fungsi game berdasarkan interaksi antar aktor dan objek pada sistem digambarkan dengan menggunakan diagram use case yang terdapat pada gambar 3-4.



Gambar 3-5 *Use Case Diagram*

B. Definisi Aktor

Definisi aktor berfungsi menjelaskan peran setiap aktor yang terdapat pada *use case*. Definisi aktor ditunjukan pada Tabel 3-12.

Tabel 3-12 Definisi Aktor

No.	Aktor	Deskripsi
1.	Pemain	User / pengguna / orang yang memainkan game ini

C. Definisi *Use Case*

Definisi *use case* berfungsi menjelaskan tiap-tiap proses dan objek yang terdapat pada *use case*. Definisi *use case* ditunjukan pada Tabel 3-13.

Tabel 3-13 Definisi *Use Case*

No.	*Use Case*	Deskripsi
1.	Mulai Permainan	Proses memulai permainan
2.	Memainkan *Level* 1	Proses memainkan *level* 1 terdapat fungsi yang menampilkan waktu, skor, lahan, penyiram, tanaman yang disiram yaitu tomat dan terong.
3.	Memainkan *Level* 2	Proses memainkan *level* 2 terdapat fungsi yang menampilkan waktu, skor, lahan, penyiram, tanaman yang disiram yaitu timun dan labu.
4.	Memainkankan *Level* 3	Proses memainkan *level* 3 terdapat fungsi yang menampilkan waktu, skor, lahan, penyiram, tanaman yang disiram yaitu kubis dan jagung.
5.	Review *level*	Proses review *level*

D. Skenario Use Case

Skenario *use case* menunjukan proses apa yang terjadi pada setiap bagian di dalam *use case* tersebut, dimana user memberikan perintah pada bagian-bagian *use case* dan respon apa yang diberikan oleh sistem kepada *user* setelah *user* memberikan perintahnya.

Tabel 3-14 Skenario *use case* mulai permainan

Identifikasi	
Nomor	1
Nama	Mulai Permainan
Tujuan	Memulai permainan dari awal
Deskripsi	Proses memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih mulai	2. Menampilkan intro
	3. Menampilkan petunjuk

Tabel 3-15 skenario *use case* memainkan level 1

Identifikasi	
Nomor	1

Nama	Memainkan
Tujuan	Memainkan permainan *level* 1
Deskripsi	Proses menampilkan *level* 1
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Arena permainan ditampilkan
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan goal *level* 1
2. menggerakan kursor	3. Menampilkan penyiram tanaman
6. menyirami tomat	4. menampilkan tanaman tomat yang akan disirami
7. menyirami terong	5. menampilkan tanaman terong yang akan disirami
	8. menampilkan stok tanaman terjual
	9. menampilkan skor
	10. menampilkan waktu

Tabel 3-16 skenario *use case* memainkan level 2

Identifikasi	
Nomor	1
Nama	Memainkan
Tujuan	Memainkan permainan *level* 2
Deskripsi	Proses memainkan *level* 2
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Arena permainan ditampilkan
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan goal *level* 2
2. menggerakan kursor	3. Menampilkan penyiram tanaman
6. menyirami timun	4. menampilkan tanaman timun yang akan disirami
7. menyirami labu	5. menampilkan tanaman labu yang akan disirami
	8. menampilkan stok tanaman terjual
	9. menampilkan skor
	10. menampilkan waktu

Tabel 3-17 skenario *use case* memainkan level 3

Identifikasi	
Nomor	1
Nama	Memainkan
Tujuan	Memainkan permainan *level* 3
Deskripsi	Proses memainkan *level* 3
Aktor	Pemain
Skenario Utama	

Kondisi Awal	Arena permainan ditampilkan
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan goal *level* 3
2. menggerakan kursor	3. Menampilkan penyiram tanaman
6. menyirami kubis	4. menampilkan tanaman kubis yang akan disirami
7. menyirami jagung	5. menampilkan tanaman jagung yang akan disirami
	8. menampilkan stok tanaman terjual
	9. menampilkan skor
	10. menampilkan waktu

Tabel 3-18 Skenario *use case review level*

Identifikasi	
Nomor	1
Nama	*Review level*
Tujuan	Me*review* permainan
Deskripsi	Me*review* permainan untuk lanjut ke *level* berikutnya
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memilih lanjut	2. Menampilkan Data Permainan
3. Memilih Data	4. Menampilkan Petunjuk
5. Memilih keluar	6. Menampilkan pesan keluar aplikasi
Skenario Tambahan	
Kondisi Awal	**Aktor berada di pesan keluar permainan**
1. Memilih Ya	2. Keluar aplikasi
3. Memilih Tidak	4. Menampilkan Menu Utama

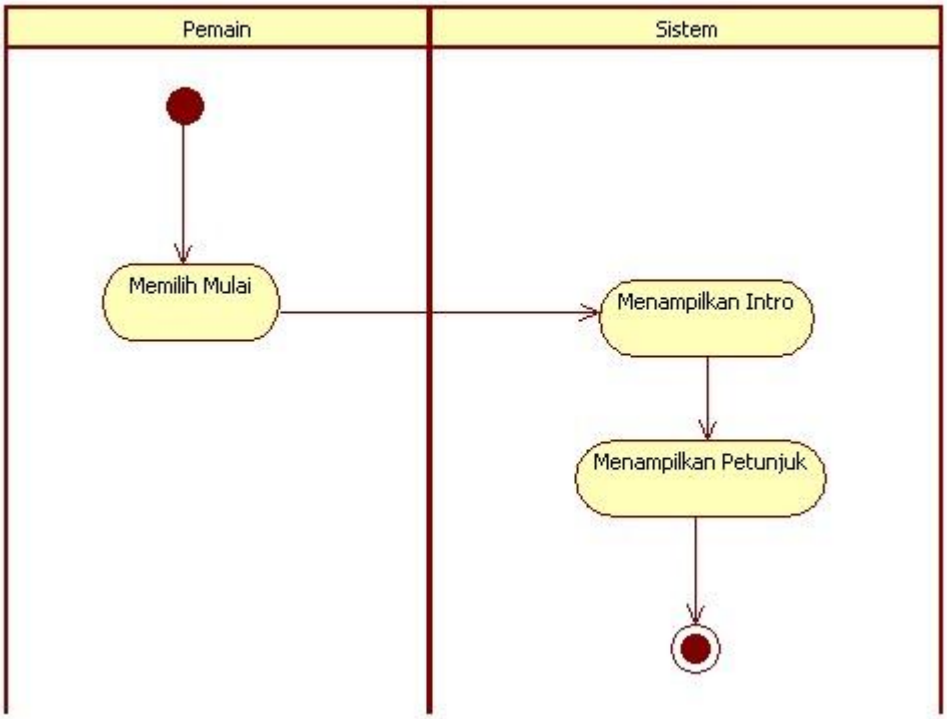
3.1.6.2 Penggambaran *Activity Diagram*

Penggambaran alir aktivitas pada *game* menggambarkan berbagai alir aktivitas dalam *game* yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Penggambaran alir aktivitas pada *game* digambarkan menggunakan *Activity Diagram*. *Activity* diagram merupakan *state* diagram khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behavior* internal sebuah sistem (dan interaksi antar subsistem) secara eksak,

tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *Use Case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *Use Case* menggambarkan bagaimana *actor* menggunakan sisitem untuk melakukan aktivitas.

A. *Activity* Diagram mulai Permainan

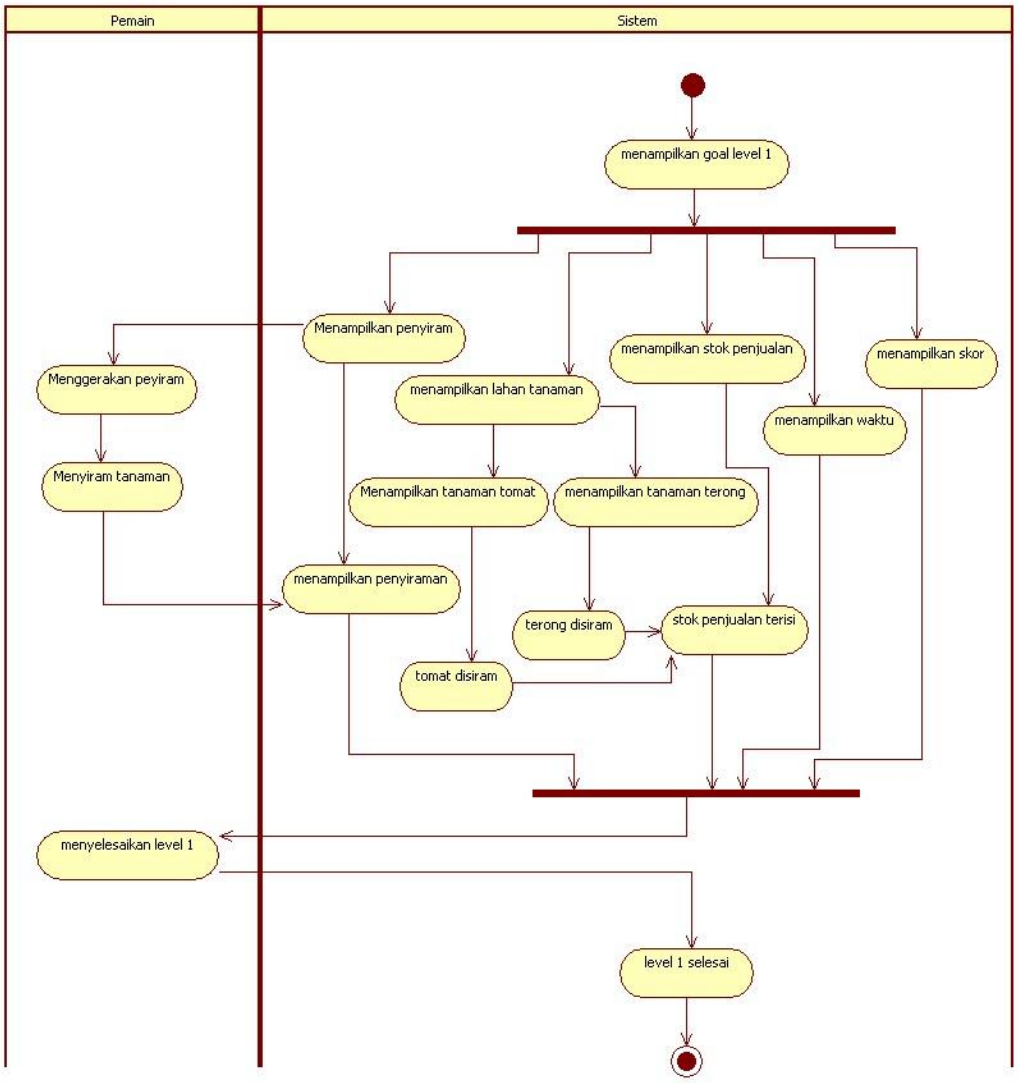
Aktivitas ini terjadi saat permainan dijalankan, sistem akan menampilkan intro permainan.



Gambar 3-6 *Activity* Diagram Mulai Permainan

B. *Activity* Diagram Memainkan *Level* 1

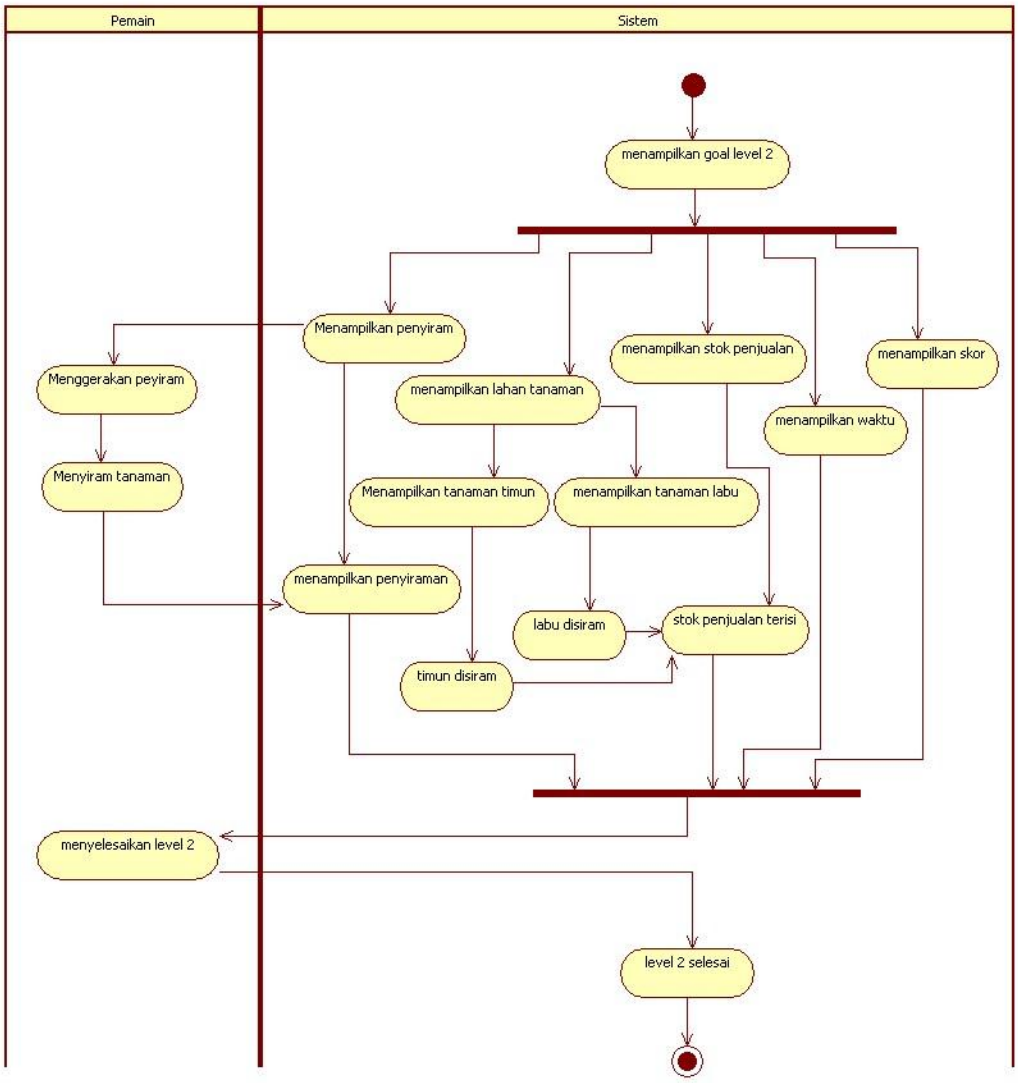
Aktivitas ini terjadi saat menampilkan *Level* 1 berupa goal *level*1, tampilan lahan, waktu, peyiram, tanaman, stok penjualan yang harus diisi.



Gambar 3-7 *Activity* Menampilkan *Level* 1

C. *Activity* Diagram Memainkan *Level* 2

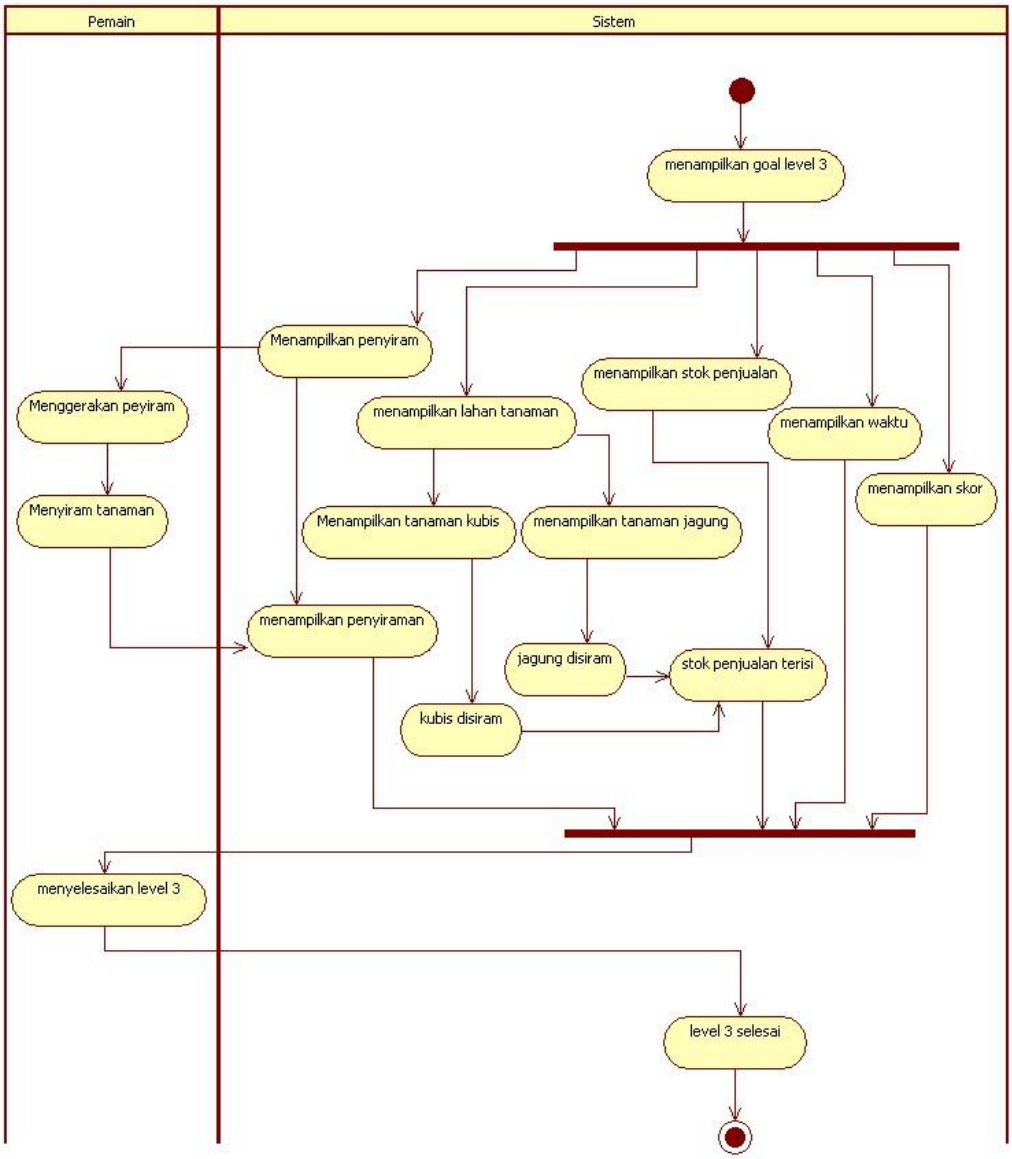
Aktivitas ini terjadi saat menampilkan *level* 2 berupa goal *level* 2, tampilan lahan, waktu, peyiram, tanaman, stok penjualan yang harus diisi.



Gambar 3-8 *Activity* Menampilkan *Level* 2

D. *Activity* Diagram Memainkan *Level* 3

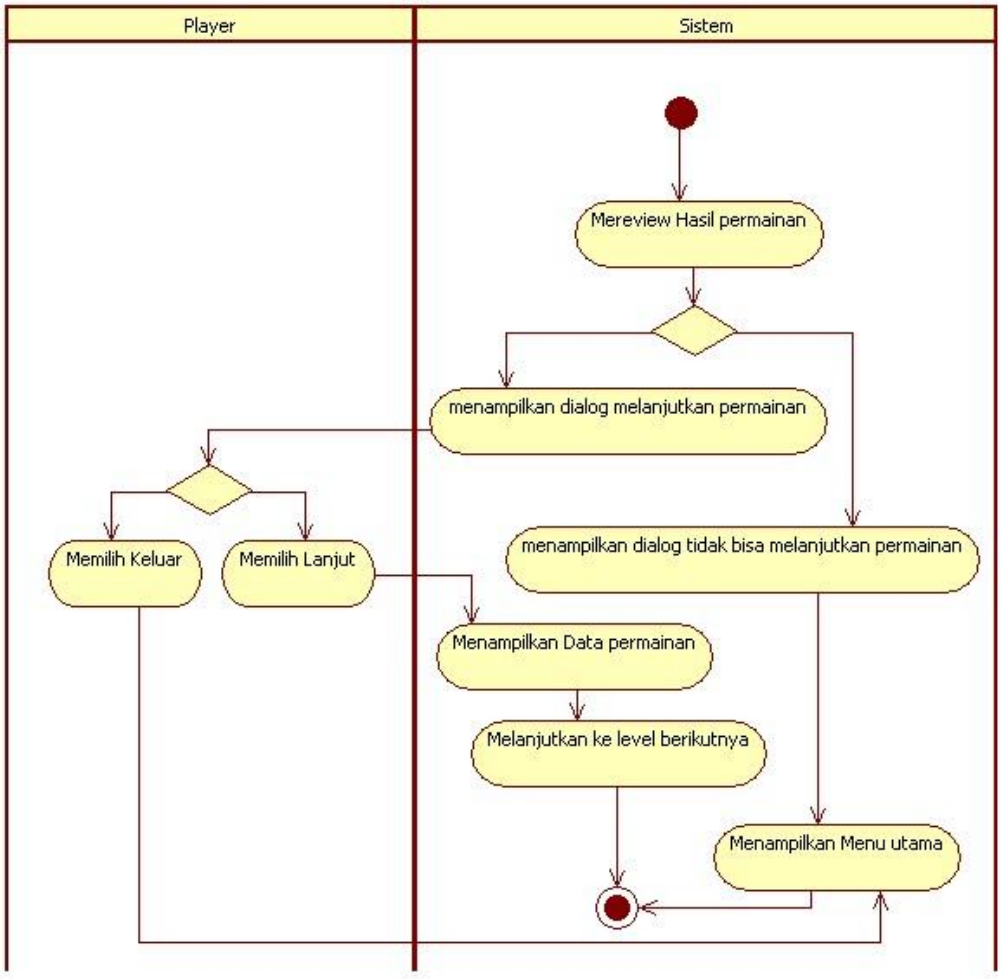
Aktivitas ini terjadi saat menampilkan *level* 3 berupa goal *level* 3, tampilan lahan, waktu, peyiram, tanaman, stok penjualan yang harus diisi.



Gambar 3-9 *Activity* Menampilkan *Level* 3

E. *Activity* Diagram *review level*

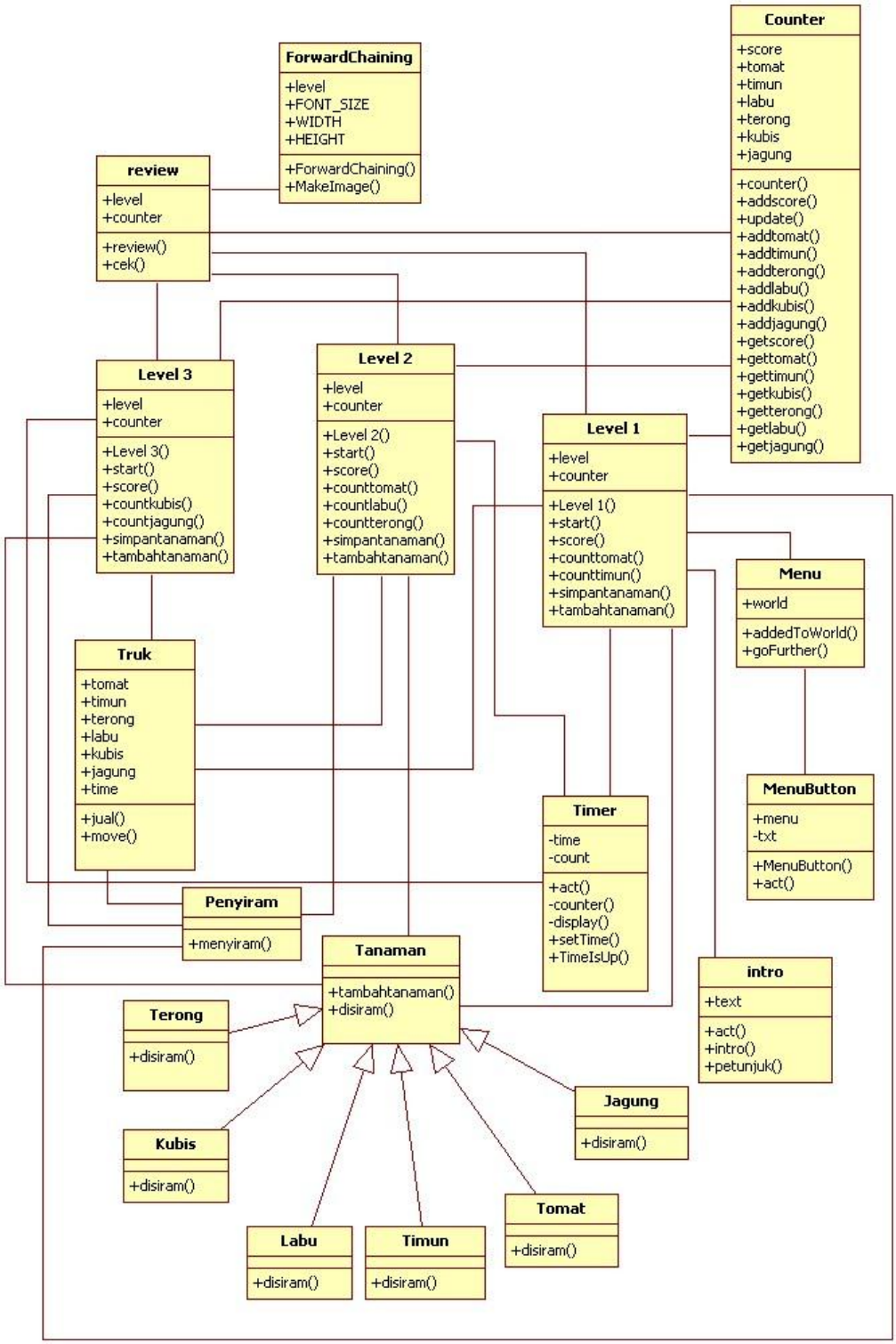
Aktivitas ini me*review* permainan untuk bisa melanjutkan permainan ke *level* berikutnya.



Gambar 3-10 *Activity* Diagram Review *Level*

3.1.6.3 Penggambaran *Class* Diagram

Kelas digambarkan dengan *class diagram*.Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi



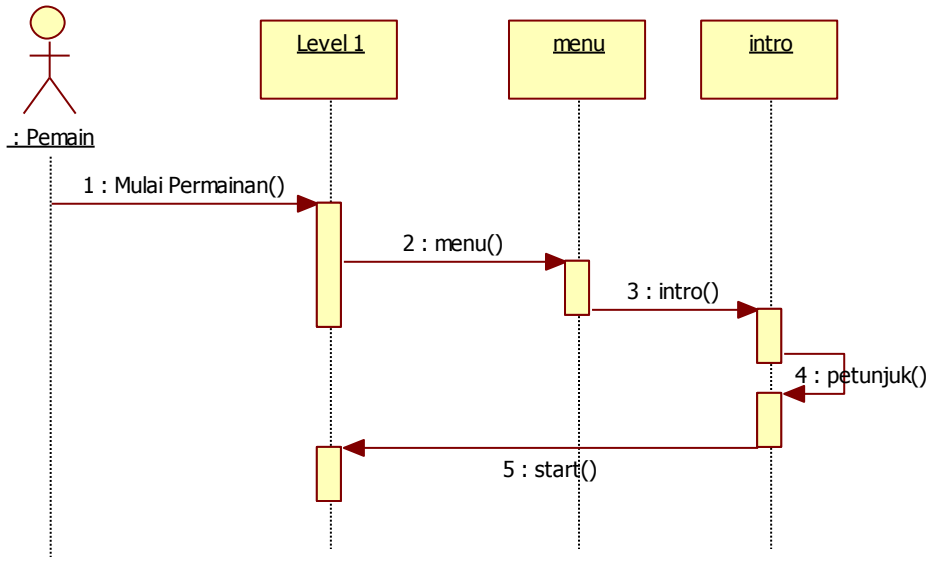
Gambar 3-11 Class Diagram

3.1.6.4 Penggambaran *Sequence* Diagram

Penggambaran interaksi antar objek pada *game* menggunakan *Sequence* Diagram.*Sequence* diagram terdiri atas dimensi *vertical* (waktu) dan dimensi horizontal (objek-objek yang terkait). Pada *sequence* diagram ini menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.

A. *Sequence* Diagram mulai Permainan

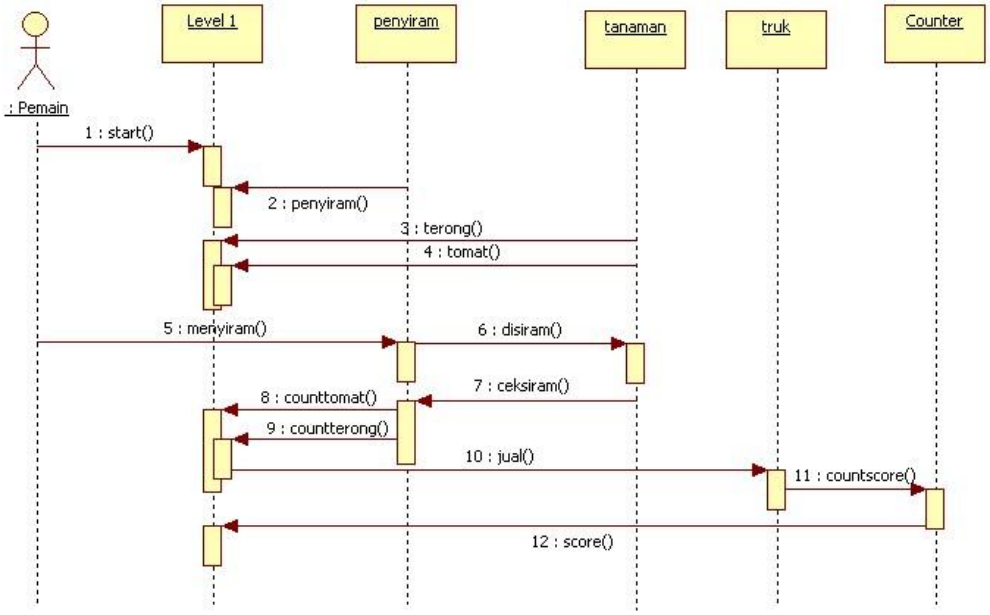
Interaksi ini terjadi saat permainan dijalankan, sistem akan menampilkan intro permainan.



Gambar 3-12 *Sequence* Diagram Mulai Permainan

B. *Sequence* Diagram Menampilkan Level 1

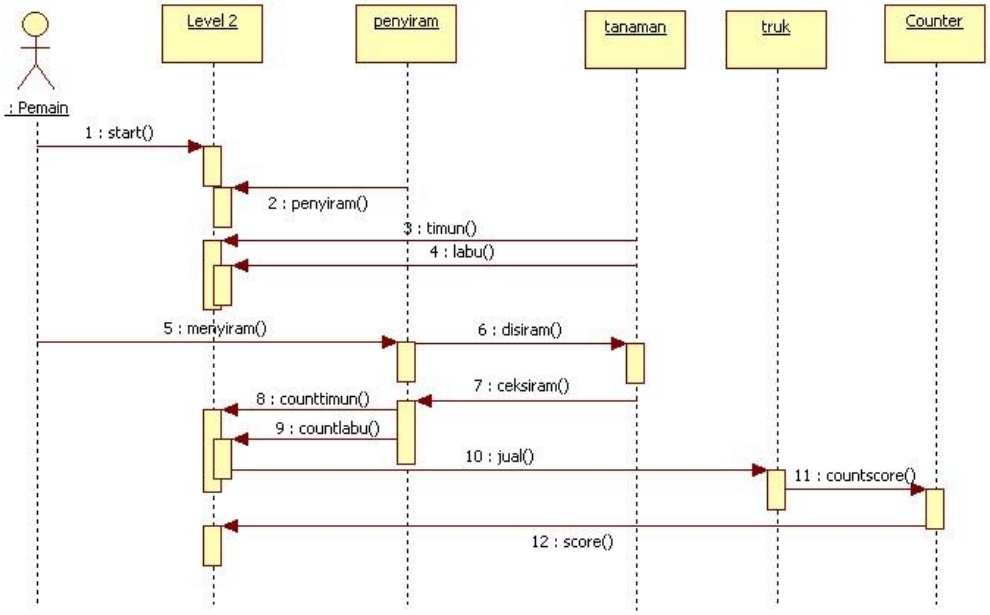
interaksi ini terjadi saat menampilkan permainan Level 1 berupa tampilan lahan, waktu, peyiram, tanaman, stok penjualan yang harus diisi.



Gambar 3-13 *Sequence* Diagram Menampilkan Level 1

C. *Sequence* Diagram Menampilkan Level 2

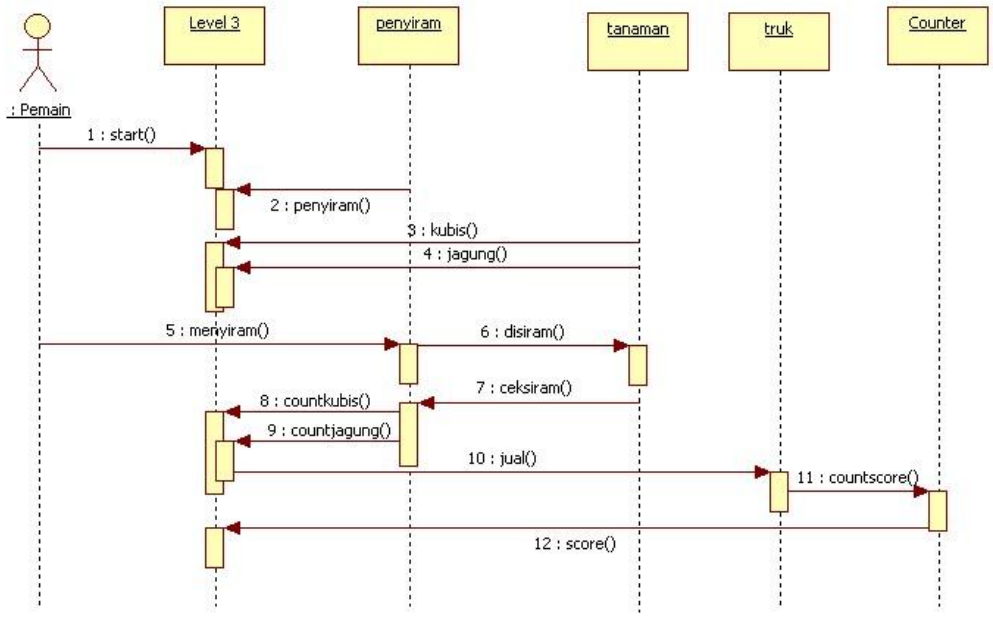
interaksi ini terjadi saat menampilkan permainan berupa tampilan lahan, waktu, peyiram, tanaman, stok penjualan yang harus diisi.



Gambar 3-14 *Sequence* Diagram Menampilkan Level 2

D. *Sequence* Diagram Menampilkan Level 3

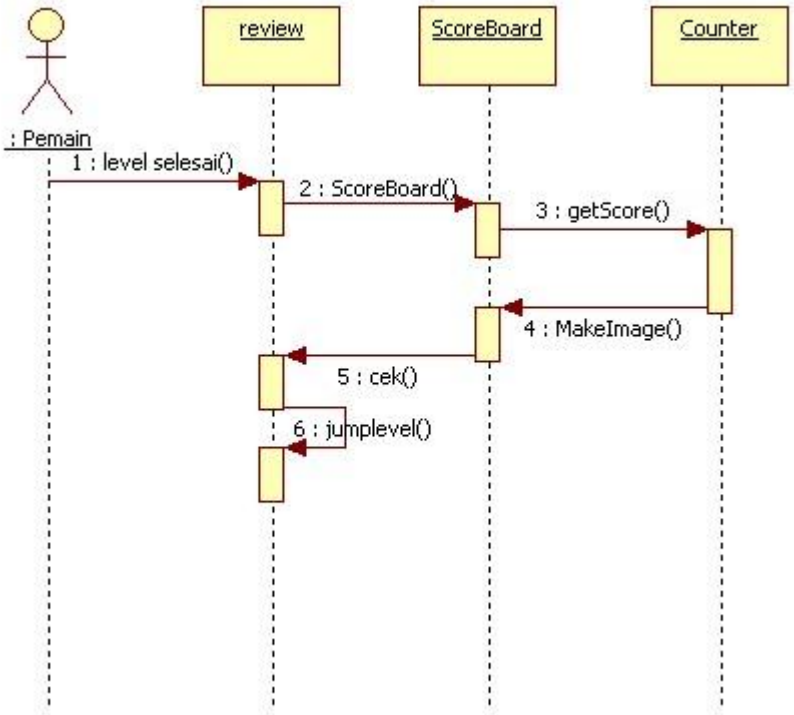
interaksi ini terjadi saat menampilkan permainan berupa tampilan lahan, waktu, peyiram, tanaman, stok penjualan yang harus diisi.



Gambar 3-15 *Sequence* Diagram Menampilkan Level 3

E. *Activity* Diagram *review level*

Interaksi ini untuk me*review* permainanan kemudian player bisa melanjutkan permainan ke *level* berikutnya atau kembali ke menu utama.



Gambar 3-16 *Sequence Diagram* Review *Level*

3.2 Perancangan Sistem

Tahap perancangan merupakan kelanjutan dari proses analisis dimana dilakukan perubahan-perubahan terhadap sistem yang sedang berjalan. Hal ini dilakukan untuk mengatasi kekurangan yang ada, memudahkan pekerjaan yang dilakukan oleh orang yang terlibat dan menghemat waktu pekerjaan.

Perancangan game ini meliputi perancangan struktur menu, perancangan antarmuka, perancangan pesan, jaringan semantik, perancangan prosedural. Semua perancangan yang telah disebutkan diatas harus saling berhubungan agar menciptakan suatu sistem yang baik.

3.2.1 Perancangan Komponen Permainan

Perancangan komponen permainan merupakan bagian pendeskripsian setiap komponen yang ada pada game baik objek, dan *story board*. Perancangan komponen permainan terdiri dari perancangan objek dan perancangan *story board*.

3.2.1.1 Perancangan Objek

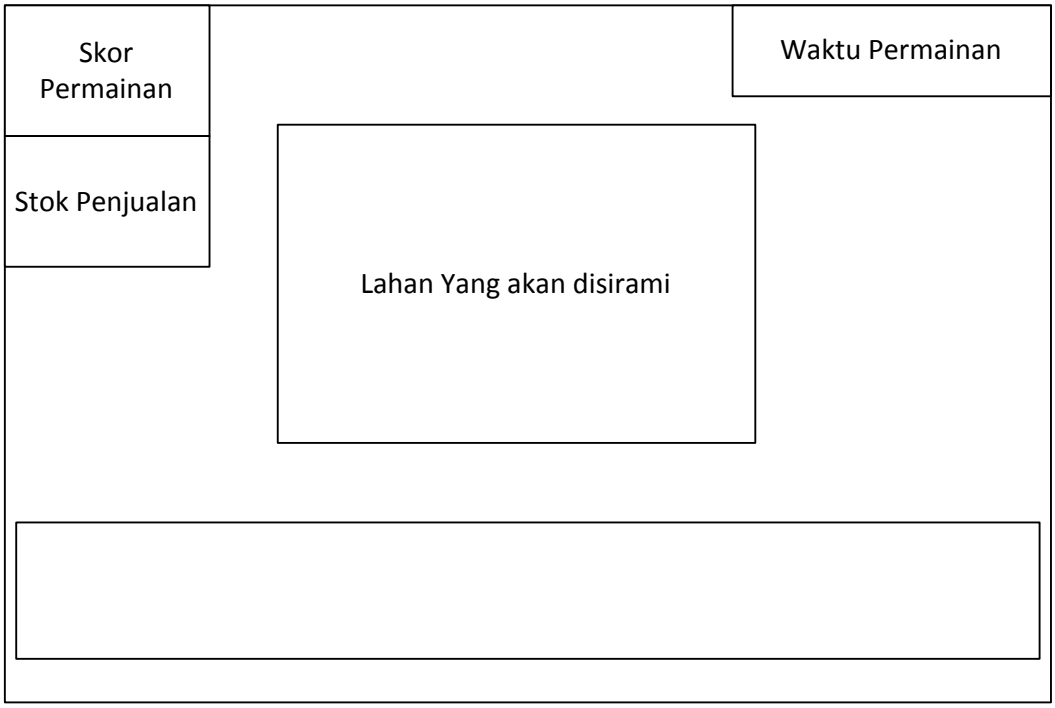
Berikut ini adalah perancangan objek yang akan diterapkan di aplikasi *game* yang akan dibangun.

Tabel 3-19 Perancangan Objek

Nama tanaman/objek	Ilustrasi
Tomat	
Timun	
Labu	
Terong	
Kubis	
Jagung	
Penyiram	

3.2.1.2 Perancangan *Storyboard*

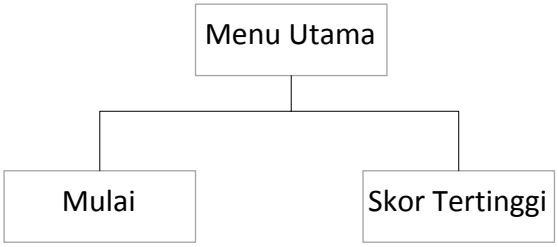
Storyboard merupakan sketsa gambar yang disusun secara berurutan berdasarkan naskah untuk menggambarkan lingkungan pada *game*. Pada game ini memiliki lahan yang sudah tertanam tanaman dan kemudian disirami dan akan mengisi stok truk, kemudian dijual dan menghasilkan *score*.



Gambar 3-17 *Story Board*

3.2.2 Perancangan Struktur Menu

Merupakan fitur-fitur yang disediakan pada sistem yang dirancang dan untuk mempermudah dalam penggunaan sistem nantinya.

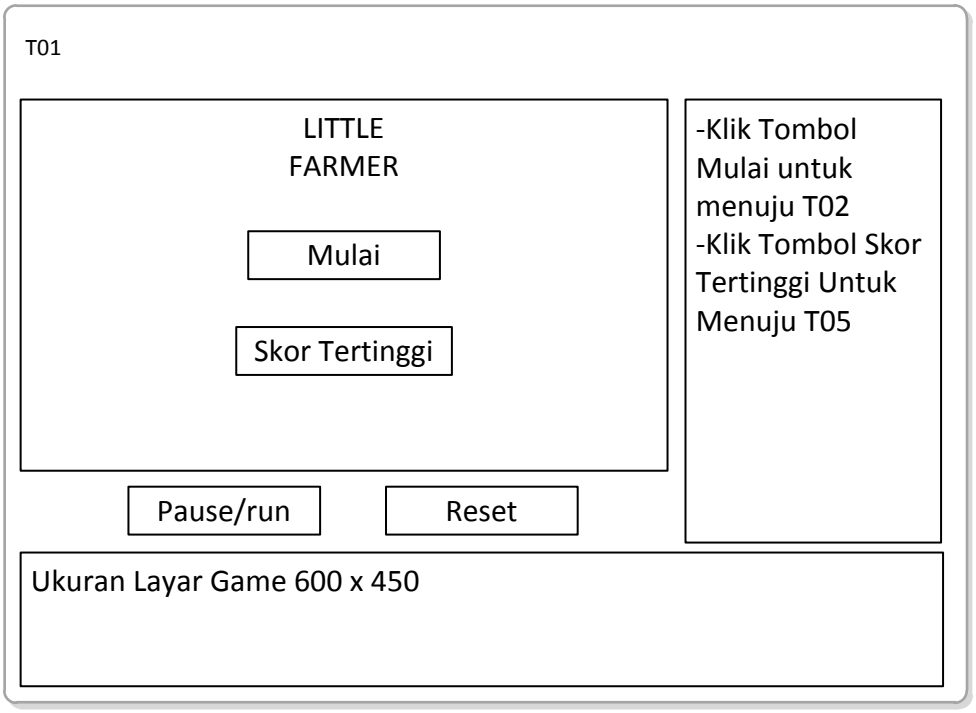


Gambar 3-18 Struktur Menu

3.2.3 Perancangan Antarmuka

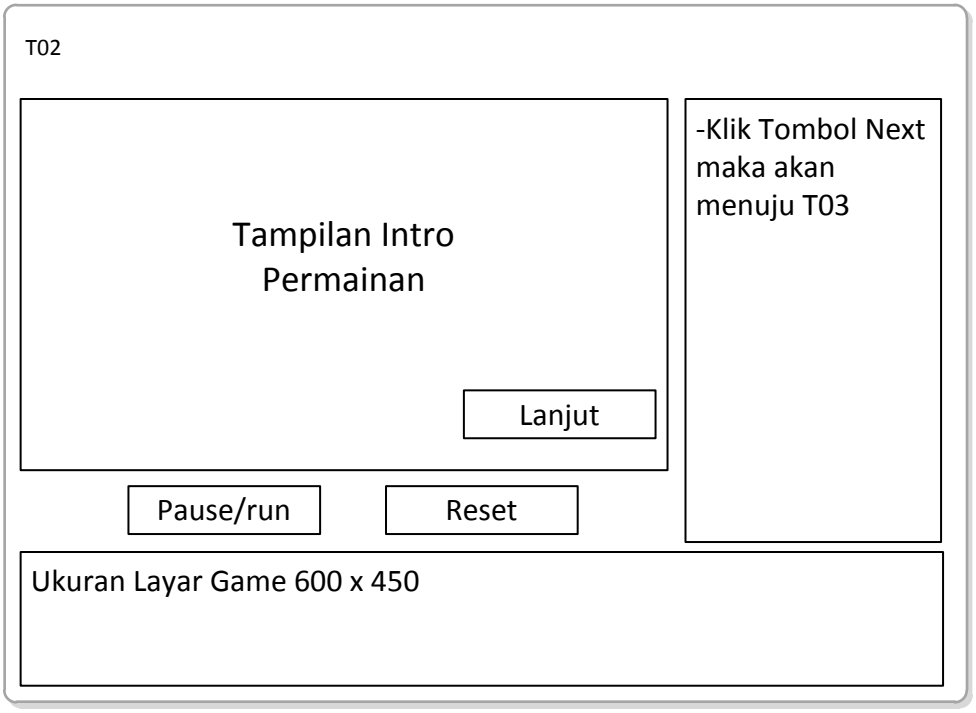
Perancangan antarmuka merupakan salah satu bagian penting dalam perancangan sistem karena nantinya antarmuka tersebut akan menjadi fasilitas yang menjembatani interaksi manusia dengan sistem.

1. Menu Utama



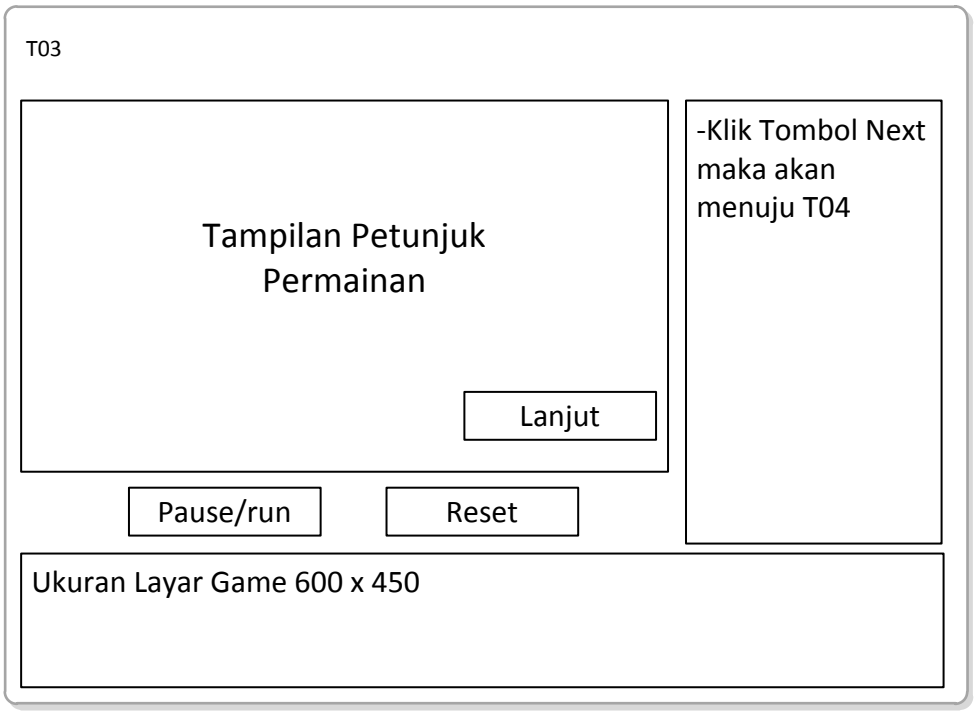
Gambar 3-19 Menu Utama

2. Intro Permainan



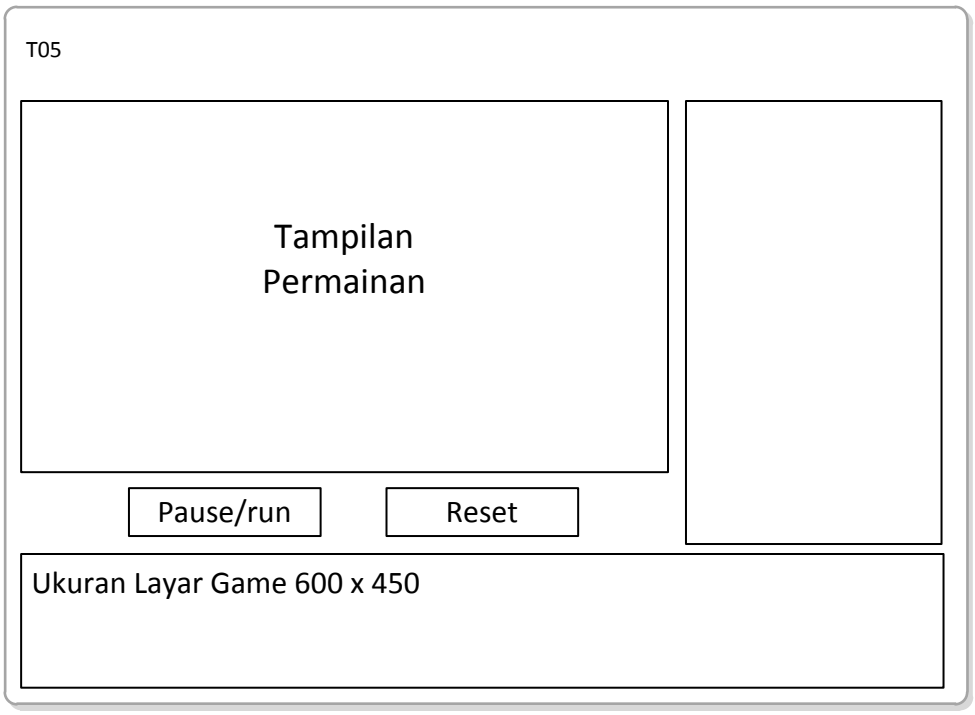
Gambar 3-20 Intro Permainan

3. Petunjuk Permainan



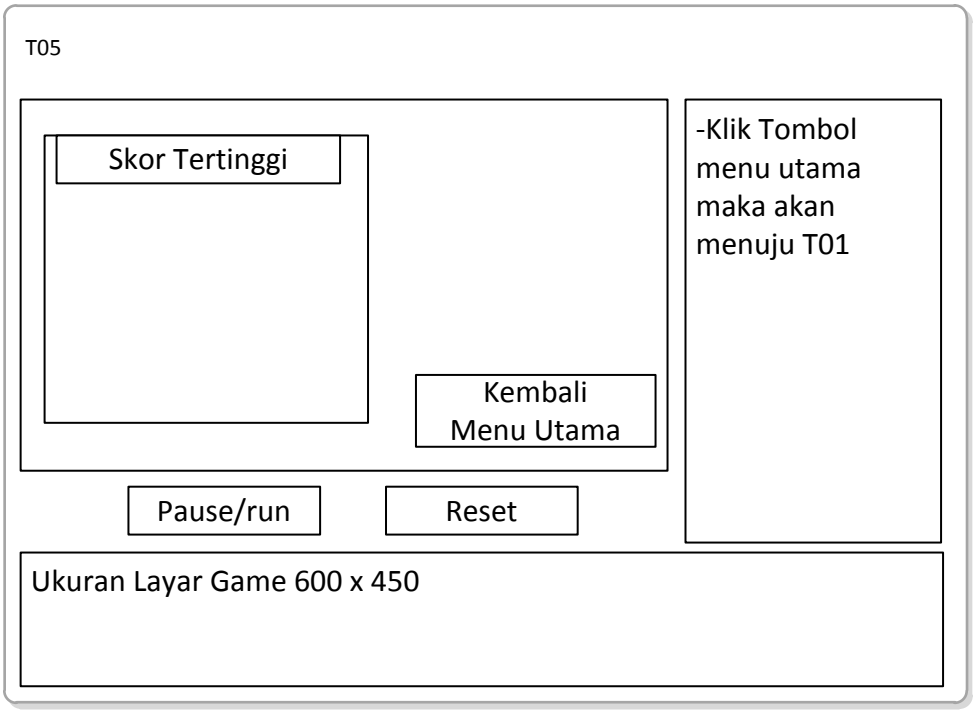
Gambar 3-21 Petunjuk Permainan

4. Tampil Permainan



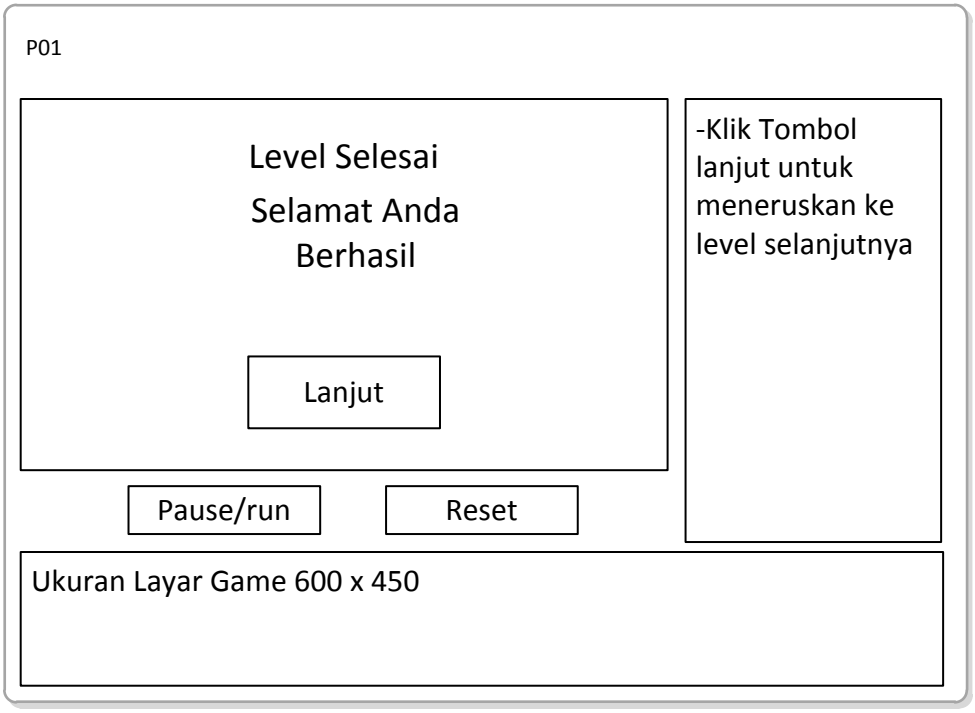
Gambar 3-22 Tampil Permainan

5. Tampil Skor Tertinggi



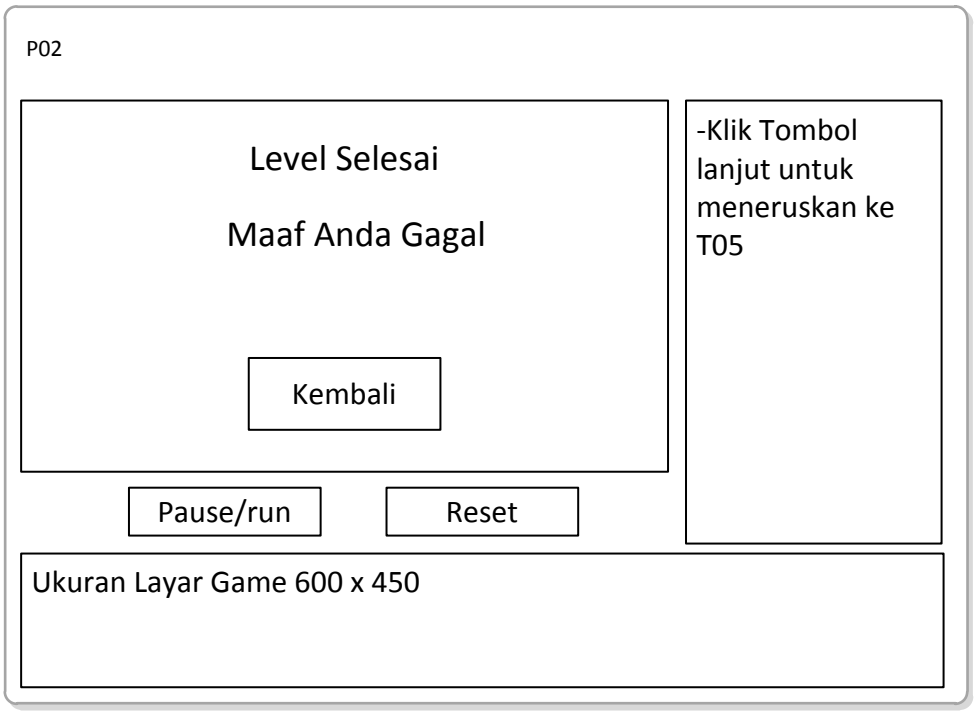
Gambar 3-23 Tampil Skor Tertinggi

6. Pesan *Level* Berhasil Diselesaikan



Gambar 3-24 Pesan *Level* Berhasil Diselesaikan

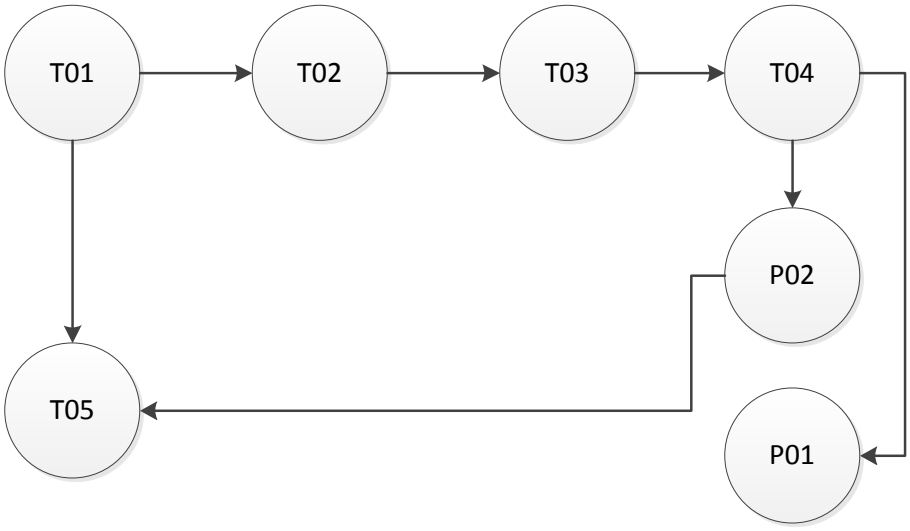
7. Pesan *Level* Gagal Diselesaikan



Gambar 3-25 Pesan *Level* Selesai

3.2.4 Jaringan Semantik

Setelah melakukan perancangan antar muka dan perancangan pesan maka dilakukanlah perancangan terhadap hubungan antar *form* dan pesan. Untuk menggambarkan hubungannya digambarkan dalam sebuah jaringan semantik



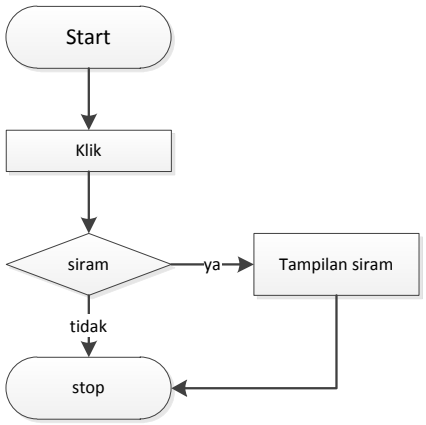
Gambar 3-26 Jaringan Semantik

3.2.5 Perancangan metode

Perancangan metode merupakan pemaparan proses pada method-method yang digunakan pada setiap kelas. Perancangan method digambarkan dengan menggunakan Flow Chart.

A. Method Siram()

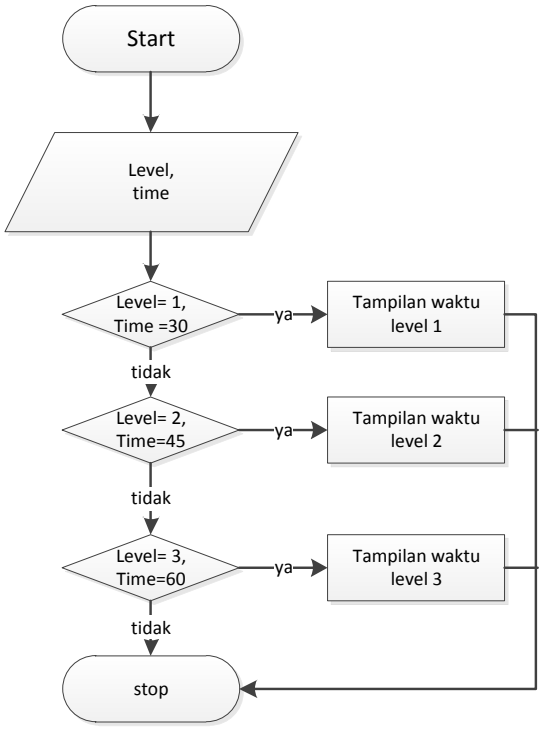
Method Siram() akan dipanggil ketika permainan dimulai.



Gambar 3-27 Method Siram()

B. Method SetTime()

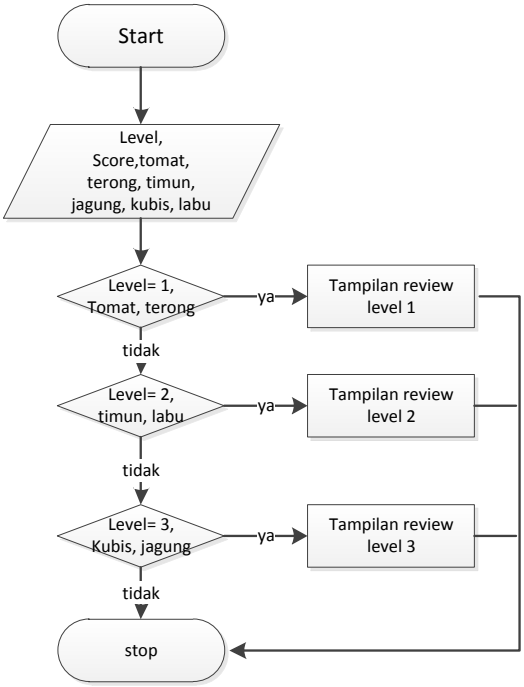
Method SetTime() akan dipanggil ketika permainan dimulai.



Gambar 3-28 Method SetTime()

C. Method Review()

Method Review() akan dipanggil ketika permainan selesai.



Gambar 3-29 Method Review()