BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada analisis sistem akan mencakup analisis masalah, *storyline*, *gameplay*, analisis tingkat kesulitan, analisis *scoring*, analisis algoritma, analisis kebutuhan *non*-fungsional dan analisis kebutuhan fungsional. Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.1.1 Analisis Masalah

Berdasarkan latar belakang masalah maka masalah yang dihadapi adalah pertumbuhan ekonomi yang dicapai tanpa memperhatikan pengelolaan hutan secara berkelanjutan dan manusia yang tidak peduli terhadap lingkungan yang menyebabkan banyak terjadinya kerusakan lingkungan.

3.1.2 *Game* Sejenis

Sebelum merancang *game* yang akan dibuat, terlebih dahulu dilakukan analisis pada *game* sejenis. *Game* sejenis yang bertemakan *go green* adalah *Game Plant It Green* dan *lorax garden*.

1. ***Game* Plant It *Green***

Game Plant It Green merupakan *game* simulasi gerakan *go green* pada sebuah kota. Dalam *game* ini pemain diharuskan mengelola lingkungan agar kota yang terkena polusi berat yaitu *Greenville*, kembali menjadi "hijau".

Pemain dapat membangun panel tenaga surya maupun tenaga angin, mengadakan program *bike to work* dan melakukan tindakan "hijau" lainnya.



Gambar 3.1 *Game Plant It Green*

a. *Story Line Game Plant It Green*

Dalam *game* ini pemain berperan sebagai walikota dengan janji mengubah cara lama dari masa lalu, membawa masa depan yang lebih "hijau". Tujuan dari permainan ini adalah membangun kembali dan merenovasi seluruh kota juga untuk mempromosikan rumah dengan teknologi hijau (eco-technology). Keterbatasan dari *game* ini adalah informasi penyebab *global warming* hanya diberikan sebelum permainan dimulai, dan terlalu rumit untuk target umur 9-10 tahun. Kelebihan dari *game* ini, pemain dapat mengetahui kota hijau adalah kota yang menggunakan kendaraan ramah lingkungan, rumah dengan teknologi hijau, serta pabrik-pabrik yang ramah lingkungan.

b. *Gameplay Game Plant It Green*

Game ini terdapat 45 *level*. Pertama pemain harus memilih rumah yang akan difasilitasi dengan *eco-technology* setelah itu pemain harus memilih fasilitas apa yang akan diterapkan pada rumah tersebut. Selanjutnya pemain harus memilih lahan kosong untuk dibangun rumah atau taman. Pada bagian pabrik-pabrik pemain harus mengganti tenaga listrik dengan membangun panel tenaga surya maupun tenaga angin**.**

c. *Scenario* dalam *game Plant It Green* dapat dilihat dari alur permainan sebagai berikut :

1. Pemain berada di kota yang disebut *Greenville* yang dipenuhi rumah tua dan belum terfasilitasi *eco-technology*.
2. Pemain harus mengubah kota menjadi kota hijau yang memiliki perumahan dan toko berfasilitas *eco-technology*.
3. Pemain harus membangun sarana rekreasi dan taman kota.
4. Pemain harus membangun kincir angin dan kincir air sebagai sumber tenaga listrik.
5. *Game* ini terdapat 45 lavel.

2. *Game Lorax Garden*

Game ini merupakan *game* simulasi berkebun dengan target usia 6 tahun ke atas. Permainan ini terinspirasi dari buku anak-anak tentang lingkungan yang dikarang oleh Dr. Seuss. Dalam permainan ini, pemain dituntut untuk bekerja menanam pohon dan bunga untuk melestarikan lingkungan.

a. Story line game lorax garden

Game ini menceritakan hutan Truffula yang tidak lagi terdapat banyak pohon dan bunga sehingga hewan-hewan lorax kehilangan habitatnya. Pemain harus menanam pohon dan bunga sebanyak mungkin agar hutan Truffula kembali asri dan hewan-hewan lorax mendapatkan kembali habitatnya.

b. Gameplay game lorax garden

Untuk menanam pohon pemain memilih benih pohon, warna, bentuk daun, bentuk batang dan pola batang untuk setiap pohon yang akan ditanam kemudian siram denagn air dan tunggu hingga tumbuh besar, setelah tumbuh besar pohon yang sudah ditanam dipindahkan ke hutan Truffula. Sedangkan untuk menanam bunga, pilih benih bunga, sirami dengan air, jaga bunga dari gulma dan serangga. Buang gulma dan semprot serangga berbahaya dari bunga dengan racun serangga. Setelah tumbuh sempurna pindahkan bungan ke hutan Truffula.

c. *Scenario* dalam *game Lorax Garden* dapat dilihat dari alur permainan sebagai berikut :

1. Desain pohon Truffula pilih benih, warna, bentuk daun, bentuk batang, dan pola batang untuk setiap pohon. Sirami dengan air dan tunggu sampai pohon tumbuh besar kemudian pindahkan pohon yang ditaman ke hutan Truffula.
2. Menanam bunga, pilih benih bunga, sirami dengan air, perhatikan bunga hingga mekar, buang gulma, dan semprot serangga berbahaya dengan racun seranggan.

Secara garis besar perbandingan *game* sejenis diatas dijelaskan dalam tabel dibawah ini.

Tabel 3.1 Tabel Komponen *Game* Sejenis

Nama *Game*	**Plant It *Green***
Jenis *Game*	*Construction and Management Simulation* bermode misi
Level	*Game* ini terdapat 45 *level*
score	Menghasilkan uang dari pajak per-hari *resident* $ 6.100 Menghasilkan 1850 *energy credits* perhari dari *resident* Harga pembelian 5000 bahan bangunan $ 12.500 Harga pembelian 10.000 bahan bangunan $ 22.500 Harga pembelian 25.000 bahan bangunan $ 50.000 Harga pembelian 100.000 bahan bangunan $ 150.000 Membangun *park* membutuhkan 5000 bahan bangunan dan biaya $ 25.000 Membangun *Eco-Home* membutuhkan 25000 bahan bangunan dan biaya *$* 50000 Membangun *Pre-FabHome* membutuhkan 10000 bahan bangunan dan biaya *$* 25.000 Membangun *Eco-Apartments* membutuhkan 50000 bahan bangunan dan biaya *$* 75.000 Menghasilkan uang dari pajak per-hari industri $ 4.000 Menghasilkan 1750 *energy credits* perhari dari industri Membangun *solar energy* membutuhkan 10000 bahan bangunan dan biaya *$* 25.000 Membangun *wind power turbines* membutuhkan 25000 bahan bangunan dan biaya *$* 50.000 Membangun *hydropower plant* membutuhkan 5000 bahan bangunan dan biaya *$* 75.000 Dengan men-demolish coal power plant mendapatkan 10000 bahan bangunan
Karakter	Alicia *Green*, menyampaikan petunjuk permainan
Nama *Game*	**Lorax Garden**
Jenis *Game*	*Artificial Life*
Level	*Game* ini terdapat 23 *level*
score	Setiap menanam bunga mendapatkan 1 hati untuk mempercepat pertumbuhan pada penanaman pohon
Karakter	Lorax, menyampaikan petunjuk permainan

3.1.3 *Game* yang akan dibangun

Game yang akan dibangun berjudul *go green* bersama petualangan si ulat yang bergenre *petualangan dan edukasi*. *Game* ini dibangun dengan tema yang

sama pada *game* sejenis yaitu bertemakan *go green* dengan mengaplikasikan teknologi sebagai jembatan untuk menyajikan materi mengenai bagian dan fungsi pohon, proses *fotosintesis* dan menyampaikan pesan moral mengenai *global warming* dan penghijauan yang khususnya ditujukan kepada siswa-siswi SD kelas IV. Fitur-fitur yang akan dibuat pada *game* ini yaitu :

1. *Game* petualangan
2. Petunjuk
3. Materi pembelajaran
4. Quiz

3.1.4 Analisis *Game* yang Akan Dikembangkan

3.1.4.1 Storyline

Game go green bersama petualangan si ulat ini menceritakan petualangan ulat dalam proses *metamorfosis* dan proses *fotosintesis* pada pohon yang merupakan *area* petualangan. Dalam *game* ini terdapat karakter utama yaitu ulat, 4 karakter pembantu yaitu Mr. Akar, Mr. Batang, Mr. Daun, dan Mr. Sun, dan 4 karakter musuh yaitu burung hantu, raja burung hantu, polusi dan raja polusi. *Game* ini terbagi menjadi tiga bagian.

Bagian pertama yaitu perjuangan ulat pada *area* akar pohon. Pada bagian pertama ini ulat harus mendapatkan semua informasi terlebih dahulu sebelum bertemu Mr. Akar yang akan memberikan pertanyaan untuk dapat melanjutkan ke *level* 2. Informasi yang didapat adalah jenis-jenis akar, fungsi akar, bagian-bagian akar dan pesan moral *global warming* akan disampaikan setelah ulat menjawab semua pertanyaan dari Mr.Akar. Pada saat ulat bertemu Mr. Akar, ulat akan diberi tiga pertanyaan. Dari tiga pertanyaan, satu jawaban salah atau semua jawaban benar maka raja burung hantu akan pergi, sedangkan jika dua jawaban salah maka *game* akan berakhir karena ulat berhasil dimangsa raja burung hantu. Pada bagian ini tokoh musuh adalah burung hantu yang akan menghalang-halangi jalan.

Bagian kedua adalah *area* batang pohon. Pada batang pohon ulat akan mencari jalan keluar *menu*ju bagian berikutnya (*level* 3) dengan rintangan labirin. Di bagian kedua ini ulat harus mendapatkan informasi-informasi seperti pada *level* 1 selama perjalanannya sebelum menemui Mr. Batang. Informasi yang didapat

mengenai bagian-bagian pada batang, fungsi batang dan jenis batang. Pesan moral *global warming* akan disampaikan setelah ulat menjawab semua pertanyaan dari Mr. Batang. Ulat harus menemui Mr. Batang untuk dapat keluar dari labirin dengan cara menjawab tiga pertanyaan. Dari tiga pertanyaan, satu jawaban salah atau semua jawaban benar maka raja burung hantu akan pergi, sedangkan jika dua jawaban salah maka *game* akan berakhir karena ulat berhasil dimangsa raja burung hantu. Pada bagian kedua ini letak Mr. Batang akan berpindah-pindah sehingga dapat menghasilkan *path finding* yang berbeda-beda.

Bagian ketiga adalah *area* daun. Pada *area* ini ulat berubah menjadi kepompong dan mendapat serangan dari 9 burung hantu (*owl*), raja burung hantu, 9 awan gelap (polusi) dan raja polusi yang akan mengganggu proses *metamorfosis* ulat juga *fotosintesis* pada pohon. Untuk dapat mengalahkan musuhnya tembak 9 burung hantu dan 9 polusi dan harus dapat menjawab masing-masing tiga pertanyaan yang diberikan oleh Mr. Daun dan Mr. Sun untuk dapat mengalahkan raja burung hantu yang akan memangsa, dan melenyapkan raja polusi yang menghalangi matahari. Informasi akan muncul setiap 3 burung hantu dan 3 polusi datang menyerang kepompong.

3.1.4.2 *Game*play

Tugas utama pemain dalam memainkan *game* ini adalah untuk menyelesaikan misi-misi yang ada pada setiap bagian atau *level* pada *game.* Setiap *level* memiliki misi yang berbeda. Berikut ini akan dijelaskan misi-misi dari setiap *level* nya.

1. *Level* 1

   Misi ulat di *area* akar pohon :

   Ulat harus mendapatkan semua informasi sebagai bahan pelajaran untuk dapat menjawab tiga pertanyaan dari karakter pembantu Mr. Akar. Menjawab pertanyaan untuk dapat *menu*ju ke *level* berikutnya dan mengusir burung hantu (*owl*) yang akan memangsanya.

2. *Level* 2

   Misi ulat di *area* batang pohon :

Ulat berada dalam labirin, ulat harus mendapatkan semua informasi sebagai bahan pelajaran untuk dapat menjawab tiga pertanyaan dari karakter pembantu Mr. Batang. Selanjutnya ulat menemui Mr. Batang untuk mendapatkan petunjuk jalan keluar, keluar dari labirin, dan menjawab pertanyaan.

3. *Level* 3

Ulat harus menembak burung hantu dan polusi, menjawab tiga pertanyaan dari Mr. Daun dan Mr. Sun agar dapat mengalahkan raja burung hantu (*owl*) juga melenyapkan raja awan gelap (polusi) dan mengambil informasi setiap 3 burung hantu dan polusi datang menyerang. Setelah selesai mengalahkan musuh-musuhnya pohon dapat berfotosintesis karena cahaya matahari tidak lagi tertutup oleh raja awan gelap (polusi) yang akan menghasilkan udara yg bersih dan ulat akhirnya berhasil bermetamorfosis menjadi kupu-kupu karena tidak dapat dimangsa oleh raja burung hantu.

Tipe berupa *game* petualangan dan edukasi dimana karakter utama dapat digerakan oleh pemain untuk bisa bergerak ke kiri, ke kanan, ke atas dan ke bawah, loncat, dan menembak musuh. Sedangkan tipe ujian yang akan diberikan berupa ujian dengan tipe soal pilihan ganda.

3.1.4.3 Analisis Tingkat Kesulitan

Tingkat kesulitan pada *game* ini adalah sebagai berikut :

1. *Level* 1

Ulat akan mendapatkan penghalang dari anak burung hantu yang bergerak naik turun, harus mendapatkan semua kotak informasi terlebih dahulu untuk menemui Mr. Akar, diburu raja burung hantu di akhir *level* 1.

2. *Level* 2

Ulat harus mendapatkan semua informasi terlebih dahulu untuk menemui Mr. Batang dan keluar dari labirin untuk dapat melanjutkan ke *level* 3.

3. *Level* 3

Ulat mendapatkan serangan dari 9 burung hantu (*owl*) dan raja burung hantu yang akan menggagalkan metamorfosisnya dan ulat harus dapat melenyapkan 9 awan gelap (polusi) dan raja polusi agar pohon dapat berfotosintesis.

3.1.4.4 Analisis *Scoring*

Analisis *scoring* adalah analisa untuk memberikan penilaian kepada *user* yang menyelesaikan *game*. Analisis *scoring go green* bersama petualangan si ulat adalah sebagai berikut.

Tabel 3.2 *Scoring*

Level	**Keterangan**	***Score***
Level 1 *area* akar	- Terdapat 3 pertanyaan - Terdapat 3 kotak informasi - Mengusir raja burung hantu - daun	- Setiap pertanyaan 5 - Setiap kotak informasi 5 - mengusir raja burung hantu 25 - satu daun 10
Level 2 *area* batang (labirin)	- Terdapat 3 pertanyaan - Terdapat 3 kotak informasi - daun	- Setiap pertanyaan 5 - Setiap kotak informasi 5 - satu daun 10
Level 3 *area* daun	- Menjawab 6 pertanyaan - Terdapat 6 informasi - Membunuh 9 burung hantu - Melenyapkan 9 polusi - Mengalahkan raja burung hantu - Melenyapkan raja polusi	- Setiap pertanyaan 5 - Setiap informasi 5 - Setiap membunuh satu burung hantu 5 - Setiap melenyapkan satu polusi 5 - Mengalahkan raja burung hantu 25 - Melenyapkan raja polusi 25

3.1.4.5 Analisis Algoritma A*

Algoritma A* merupakan algoritma untuk mencari rute terpendek. Algoritma ini digunakan untuk mencari rute terpendek antara tokoh pembantu dan pintu keluar pada labirin. Algoritma ini bekerja pada saat insialisasi *level* 2. Hasil algoritma akan ditampung sampai tokoh ulat me*menu*hi syarat untuk menemui tokoh pembantu. Berikut ini adalah *Map area* labirin pada *level* 2.

y/x	1	2	3	4	4	5	6	7	8	9	10	11	12	13	15	16	17	18
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
4	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
5	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
6	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108
7	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
8	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
9	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162
10	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
11	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198
12	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216
13	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234
14	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252
15	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270
16	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288
17	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306
18	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324
19	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342
20	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
21	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378
22	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396
23	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414
24	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432
25	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
26	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468
27	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486
28	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504
29	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522
30	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540
31	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
32	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576

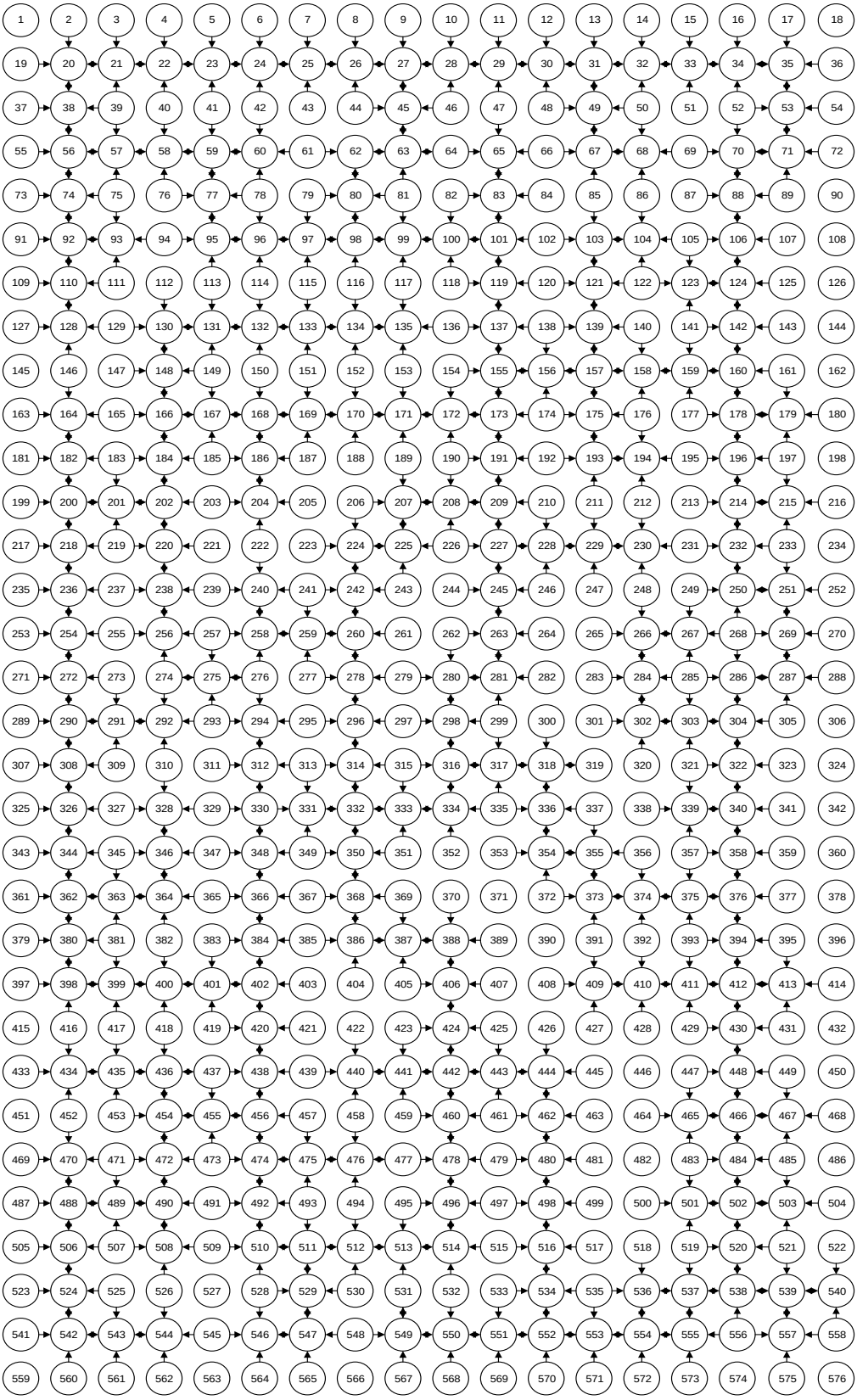
Gambar 3.1 *Map* Keseluruhan

Labirin merupakan array yang di inisilisasi terhadap sekumpulan *node-node* yang terhubung. Luas labirin yaitu 18 x 32 = 576 *node*, setiap 1 *node* akan mewakili 80 x 48 *pixel*.

Tabel 3.3 Keterangan Warna Pada *Map* Labirin

Warna	Keterangan
	Block atau penghalangan
	Posisi ulat
	Posisi tokoh pembantu (sebelum mulai menampilkan *map* terdapat 4 posisi untuk tokoh pembantu, dimana posisi tokoh pembantu hanya akan diletakan secara acak di salahsatu 4 posisi tersebut)
	Posisi Pintu Keluar
	Rute terpendek hasil perhitungan A*

Berdasarkan pada gambar 3.2, gambar 3.3 adalah tampilan *map* labirin keseluruhan berupa *node-node* terhubung agar dapat terlihat bagaimana keterhubungan antar *node* yang ada pada gambar 3.2.



Gambar 3.2 *Node* Yang Terhubung

Pada pencarian algoritma A* fungsi *heuristic* yang digunakan adalah manhattan. Dimana pada pencarian ini mencari 4 sisi yaitu atas, bawah, kiri, dan kanan dengan fungsi evaluasi f(n) = g(n) + h(n).

h(n) = D*(dx + dy)

Keterangan :

dx = *abs*(*node*.x – *goal*.x)

dy = *abs*(*node*.y – *goal*.y)

D adalah jarak antara *node*, jarak antara node pada labirin yang diinisialisasi 10.

Kriteria untuk me*menu*hi *exact cost* pada pencarian di labirin adalah sebagai berikut :

g(n) = ((*goal*.x) – ((*node*.x)) * ((*goal*.x) – (*node*.x)) + ((*goal*.y) – (*node*.y)) * ((*goal*.y)) – (*node*.y))

Penginisialisasian *f* (*n*) dapat dilakukan pada awal pembuatan *node* karena pada fungsi yang digunakan hanya membutuhkan *start node* dan *goal node* sehingga dalam pencarian pada labirin ini *f* (*n*) akan terdapat pada setiap *node* yang nantinya dapat memepermudah pencarian sehingga didapatkan procedure A* dalam pseuducode sebagai berikut.

```
Procedure Astar(input startNode:Node,endNode:Node,
nodes:Array[0...JumlahNode] of Node, input/output
CLOSEDLIST:List, OPENLIST:List output tempPath:List)
DEKLARASI
  cur : Integer{penampung node dengan nilai terkecil di
openlist dan penampung dari start node}
  des : Integer{penampung node tujuan atau goal}
  min  : Integer
  nextNode : Integer
  tempNeighbors : List
  tempMin : Node
ALGORITMA
  cur ← startNode.id
  des ← endNode.id
  while (cur ≠ des) do
    min ← undefined
    CLOSEDLIST ← push(cur)
    deleteOpen(cur)
    for q ← 0 to q < nodes[cur - 1].neighbors.length do
      tempMin ← nodes[cur - 1].neighbors[q]
      if (not cekClosed(tempMin.id))then
        tempNeighbors ← push(tempMin.id)
        if ((min = undefined) or (min > tempMin.f))then
          min ← tempMin.f
          nextnode ← tempMin.id
        endif
      endif
    endfor
    for q ← 0 to q < OPENLIST.length do
      if ((min = undefined) or
           (nodes[OPENLIST[q] - 1].f ≤ min)) then
        nextnode ← OPENLIST[q]
      endif
    endfor
    for q ← 0 to q < tempNeighbors.length do
      OPENLIST ← push(tempNeighbors[q])
    endfor
    cur ← nextnode
    tempPath ← push(cur)
  endwhile
```

Setelah mendapatkan *tempPath* dibutuhkan procedure untuk pembacaan yang benar yang akan menghasilkan *path*.

```
Procedure reconstructorPath(input tempPath:List, starts:Integer
output path:List)
DEKLARASI
   f    :Integer
   temp :Integer
ALGORITMA
   for f ← tempPath.length – 2 down to f ≥ 0 do
      temp ← tempPath[f]
      if (cekDes(path) = true) then
         path ← push(starts)
         break
      else
         if (cekTtg(temp) = true) then
            path ← push(temp)
         endif
      endif
   endfor
   path←reverse()
```

Pencarian pada labirin yang akan diambil salah satu contohnya dengan *start node* pada *node* ke 549 dan *goal node* pada *node* ke 540 sebagai berikut.

y/x	1	2	3	4	4	5	6	7	8	9	10	11	12	13	15	16	17	18
17	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306
18	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324
19	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342
20	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
21	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378
22	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396
23	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414
24	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432
25	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
26	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468
27	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486
28	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504
29	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522
30	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540
31	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
32	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576

Gambar 3.3 *Map* Mencari Rute Terpendek

Berikut ini adalah pemeriksaan terhadap *neighbor* dari *current node* 549 pada gambar 3.4.

Tabel 3.4 Hasil Pemeriksaan Berdasarkan *Neighbor*

***Current Node* 549**	
Neighbor Node yang diperiksa dan f(n)	[**(550,113),** (531,117)] Perhitungan *Neighbor Node* 550 d = *Math.sqrt*((b.x) - (a.x)) * ((b.x) - (a.x)) + ((b.y) - (a.y)) * ((b.y) - (a.y)) d = *Math.sqrt*((18) - (10)) * ((18) - (10)) + ((30) - (31)) * ((30) - (31)) d = 23.627416997969522 dx = *Math.abs*((*node*.x - *goal*.x)) dx = *Math.abs*((10 - 18)) dx = 8 dy = *Math.abs*((*node*.y - *goal*.y)) dy = *Math.abs*((31 - 30)) dx = 1 h = D * (dx + dy) h = 10 * (8 + 1) h = 90 f = d + h f = 23.627416997969522 + 90 f = 113.62741699796952 Perhitungan *Neighbor Node* 531 d = Math.sqrt((b.x) - (a.x)) * ((b.x) - (a.x)) + ((b.y) - (a.y)) * ((b.y) - (a.y)) d = Math.sqrt((18) - (9)) * ((18) - (9)) + ((30) - (30)) * ((30) - (30)) d = 27 dx = Math.abs((node.x - goal.x)) dx = Math.abs((9 - 18)) dx = 9 dy = Math.abs((node.y - goal.y)) dy = Math.abs((30 - 30)) dx = 0 h = D * (dx + dy) h = 10 * (9 + 0) h = 90 f = d + h f = 27 + 90 f = 117
Node yang dipilih sebagai *best node*	**(550,113)**

Tabel 3.5 adalah hasil A* berdasarkan gambar 3.4.

Tabel 3.5 Hasil A*

Cur	**OPEN (*Node*,f(n))**	**CLOSED**	***Neighbor Node* yang diperiksa dan f(n) (*Node*,f(n))**	**temp*Path***
549	-	[549]	[**(550,113),** (531,117)]	[550]
550	[(531,117)]	[549, 550]	[**(551,99)**]	[550, 551]
551	[(531,117)]	[549, 550, 551]	[**(552,85)**]	[550, 551, 552]
552	[(531,117)]	[549, 550, 551, 552]	[**(553,72)**, (534,74)]	[550, 551, 552, 553]

553	[(531,117), (534,74)]	[549, 550, 551, 552, 553]	[**(554,59)**]	[550, 551, 552, 553, 554]
554	[(531,117), (534,74)]	[549, 550, 551, 552, 553, 554]	[**(555,46)**, (536,48)]	[550, 551, 552, 553, 554, 555]
555	[(531,117), (534,74), (536,48)]	[549, 550, 551, 552, 553, 554, 555]	[**(537,35)**]	[550, 551, 552, 553, 554, 555, 537]
537	[(531,117), (534,74), (536,48)]	[549, 550, 551, 552, 553, 554, 555, 537]	[(536,48), **(538,22)**]	[550, 551, 552, 553, 554, 555, 537, 538]
538	[(531,117), (534,74), (536,48)]	[549, 550, 551, 552, 553, 554, 555, 537, 538]	[**(539, 11)**, (520, 33)]	[550, 551, 552, 553, 554, 555, 537,538, 539]
539	[(531,117), (534,74), (536,48), (520,33)]	[549, 550, 551, 552, 553, 554, 555, 537, 538, 539]	**[(540,0)**, (557,22)]	[550, 551, 552, 553, 554, 555, 537,538, 539, 540]

Setelah *tempPath* sudah ditemukan kemudian akan diperbaiki pembacaannya oleh *procedure reconstructorPath.* Berikut adalah hasil dari *procedure reconstructorPath* terhadap labirin.

y/x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
17	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306
18	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324
19	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342
20	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
21	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378
22	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396
23	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414
24	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432
25	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
26	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468
27	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486
28	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504
29	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522
30	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540
31	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
32	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576

Gambar 3.4 *Map* Dengan Hasil Perhitungan A*

3.1.5 Analisis Kebutuhan *Non* Fungsional

Analisis kebutuhan *non*-fungsional meliputi analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, serta analisis *user.*

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras adalah semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan dengan perangkat lunak (*software*) yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya.

Spesifikasi minimum perangkat keras yang dibutuhkan agar dapat menjalankan *game go green* bersama petualangan si ulat ini adalah sebagai berikut :

1. *Processor* dengan kecepatan 1.8 Ghz
2. Memori 256 MB
3. *Space* kosong di *harddisk* 11 MB
4. *VGA card*
5. Monitor
6. *Mouse*
7. *Keyboard*
8. *Speaker*

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa dapat berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan dalam membangun aplikasi *game* ini adalah sebagai berikut.

1. Sistem Operasi *Windows* (*XP, Vista, 7, 8*)
2. *Flash player 10*

3.1.5.3 Analisis *User*

Analisis *user* akan mencakup analisis beberapa parameter terhadap calon *user* dari aplikasi.

1. *Users Knowlegde and Experience* dari target *user* yang akan menggunakan *game*.

Tabel 3.6 Analisis Klasifikasi *Knowledge And Experience*

Educational level	*Reading Level*	*Typing Skills*
Game ini dapat digunakan oleh siswa-siswi kelas empat	*Game* ini dapat digunakan oleh siswa-siswi kelas empat sekolah	*Game* ini tidak memerlukan *skill* yang

sekolah dasar.	dasar dengan reading *level* sedang.	tinggi.
Computer Literacy	***Task Experience***	***System Experience***
Game ini dapat digunakan oleh pengguna yang memiliki kemampuan komputer yang sedang.	*Game* ini dapat digunakan oleh pengguna yang memiliki pengalaman penggunaan computer dan *game* yang mudah.	*Game* ini dapat digunakan oleh pengguna yang memiliki pengalaman penggunaan komputer dan *game* yang mudah.
Application Experience	***Native Language***	***Use Of Other System***
Game inidapat digunakan dalam sistem operasi *Windows xp, vista,* 7 dan 8.	*Game* ini menggunakan satu bahasa yaitu bahasa Indonesia.	*Game* ini dapat dijalankan tanpa perlu meng-*install* aplikasi lain.

2. *Users Physical Characteristic*

Keadaan fisik seseorang memungkinkan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal-hal yang harus diperhatikan terhadap *user* dari karakteristik fisiknya untuk dapat menggunakan aplikasi adalah *Handedness, Color Blindness, dan Gender*. *Handedness* adalah kemudahan dalam menggunakan *keyboard* dan input *device* (tangan kanan, tangan kiri). *Color Blindness* adalah warna pada layar. Gender adalah aplikasi yang dibuat untuk wanita atau pria.

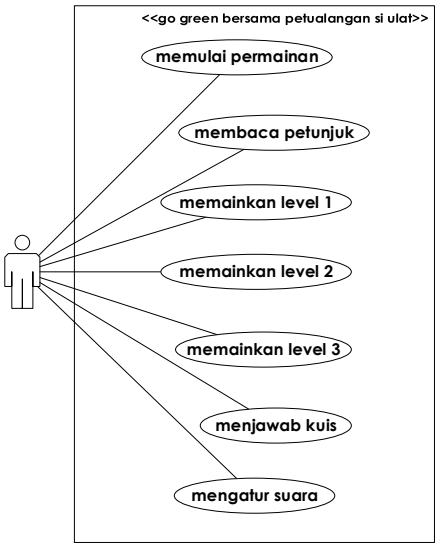
Tabel 3.7 *User*s Physical Characteristic

Age	9-10 tahun
Gender	Pria dan Wanita
Handedness	Kanan dan Kiri
Color Blind	*User* yang tidak dapat membedakan warna yang satu dengan yang lainnya (butawarna) *menu*ngkinkan dapat menggunakan aplikasi ini, karena dalam karakter yang dibuat dapat dikenali dengan mudah melalui bentuknya akan tetapi penggunaannya tidak akan optimal karena dalam *game* ini terdapat banyak sekali perbedaan warna yang *menu*njang interaksi dan ketertarikan.

3.1.6 Analisis Kebutuhan Fungsional

3.1.6.1 Use case

Use case adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Berikut *use case diagram* dari *game go green* bersama petualangan si ulat.



Gambar 3.5 *Use case Diagram Go Green* Bersama Petualangan Si Ulat

1. Definisi *Actor*

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use case Diagram*.

Tabel 3.8 Definisi *Actor*

No.	*Actor*	Deskripsi
1.	Pemain	Pemain merupakan orang yang memainkan *game* dan berinteraksi langsung dengan antar muka *game*.

2. Definisi *Use case*

Definisi *Use case* berfungsi untuk menjelaskan fungsi *Use case* yang terdapat pada *Use case Diagram*.

Tabel 3.9 Definisi *Use case*

No.	*Use case*	Deskripsi
1.	memulai permainan	**Kode *Use case* :** UC-01 **Deskripsi Singkat:** proses untuk memulai permainan **Aktor :** pemain **Trigger :** pemain memilih tombol "main" **Prekondisi :** *game* sudah *terinstall* dan siap untuk dimainkan **Pascakondisi :** menampilkan permainan
2	Membaca petunjuk	**Kode *Use case* :** UC-02 **Deskripsi Singkat:** proses untuk memberitahu pemain bagaimana cara memainkan permainan **Trigger :** sistem menampilkan petunjuk **Prekondisi :** pemain sudah memasuki *area* permainan **Pascakondisi :** sistem menampilkan permainan
3	Memainkan *level* 1	**Kode *Use case* :** UC-03 **Deskripsi Singkat: :** Proses memainkan permainan pada *level* 1 **Trigger :** sistem menampilkan permainan *level* 1 **Prekondisi :** pemain sudah membaca petunjuk *level* 1

		Pascakondisi : sistem menampilkan kuis *level* 1
4	Memainkan *level* 2	**Kode *Use case* :** UC-04 **Deskripsi Singkat: :** Proses memainkan permainan pada *level* 2 **Trigger :** sistem menampilkan permainan *level* 2 **Prekondisi :** pemain sudah membaca petunjuk *level* 2 **Pascakondisi :** sistem menampilkan pesan moral *level* 2
5	Memainkan *level* 3	**Kode *Use case* :** UC-05 **Deskripsi Singkat: :** Proses memainkan permainan pada *level* 3 **Trigger :** sistem menampilkan permainan *level* 3 **Prekondisi :** pemain telah membaca petunjuk *level* 3 **Pascakondisi :** sistem menampilkan pesan moral *level* 3
6	Menjawab kuis	**Kode *Use case* :** UC-06 **Deskripsi Singkat:** proses menjawab kuis **Trigger :** sistem menampilkan kuis **Prekondisi :** sistem menampilkan kuis **Pascakondisi :** sistem menampilkan pesan moral
7	Melihat *menu* bantuan	**Kode *Use case* :** UC-07 **Deskripsi Singkat:** proses untuk melihat cara memainkan permainan **Trigger :** mengklik tombol bantuan pada menu utama **Prekondisi :** pemain berada pada *menu* utama **Pascakondisi :** sistem menampilkan *menu* bantuan
8	Mengatur suara	**Kode *Use case* :** UC-08 **Deskripsi Singkat:** proses untuk menampilkan pengaturan suara **Trigger :** mengklik tombol pengaturan **Prekondisi :** pemain berada pada *menu* utama **Pascakondisi :** sistem menampilkan *menu* pengaturan suara

3.1.6.2 Scenario Use Case dan Activity Diagram

Berikut ini akan dijelaskan scenario *use case* dan *activity diagram* pada *game go green* bersama petualangan si ulat.

a. *Scenario Use case* dan *Activity Diagram* Memulai Permainan

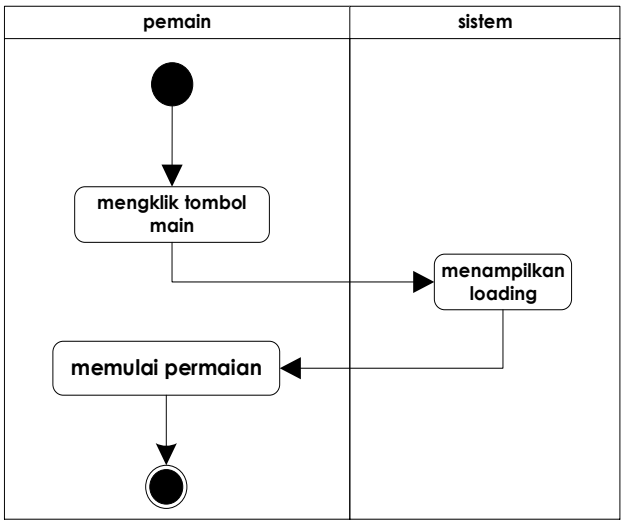
Nama *Use case* : memulai permainan (UC-01)

Berikut tabel *Scenario use case* memulai permainan.

Tabel 3.10 *Use case Scenario* Memulai Permainan

Kode Use case : **UC-01**	
***Scenario* normal** : **pemain membuka aplikasi *game* untuk memulai permainan**	
Aksi Aktor	Reaksi Sistem
1. Memilih tombol main	
	2. Menampilkan *loading*
3. Mulai memainkan permainan	

AC-01 *Activity Diagram* untuk *Use case* UC-01 memulai permainan



Gambar 3.6 AC-01 *Activity Diagram* untuk *Use case* UC-01 memulai permainan

b. *Scenario Use case* dan *Activity Diasgram* Membaca Petunjuk

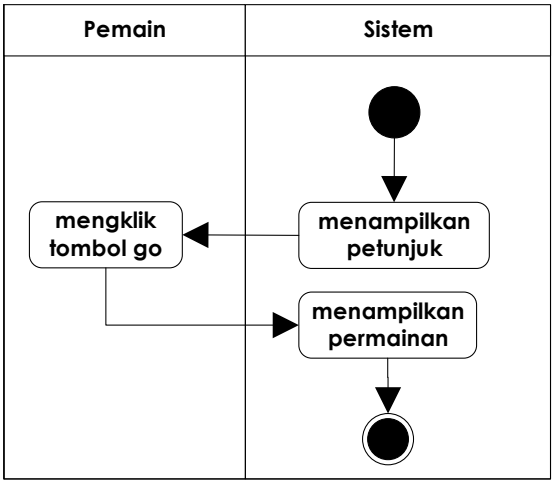
Nama *Use case* : membaca petunjuk (UC-02)

Berikut tabel *scenario use case* membaca petunjuk.

Tabel 3.11 *Scenario Use case* Membaca Petunjuk

Kode Use case : **UC-02**	
Scenario normal : **pemain membaca petunjuk**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan petunjuk
2. Memilih tombol *go*	
	3. Menampilkan permainan

AC-02 *Activity Diagram* untuk *Use case* UC-02 membaca petunjuk



Gambar 3.7 AC-02 *Activity Diagram* untuk *Use case* UC-02 membaca petunjuk

c. *Scenario Use case* dan *Activity Diagram* Memainkan *Level* 1

Nama *Use case* : memainkan *level* 1 (UC-03)

Berikut tabel *scenario use case* memainkan permainan *level* 1.

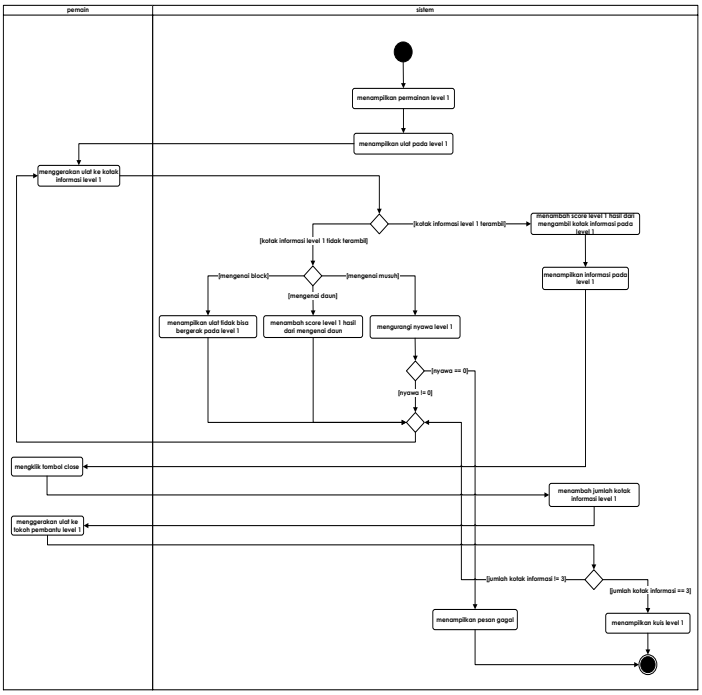
Tabel 3.12 *Scenario Use case* Memainkan *Level* 1

Kode Use case : **UC-03**	
Scenario normal : **pemain memainkan *level* 1**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 1
	2. Menampilkan ulat pada level 1
3. Menggerakan ulat ke kotak informasi *level* 1	
	4. Menambah *score level* 1 hasil dari mengambil kotak informasi pada *level* 1
	5. Menampilkan informasi pada *level* 1
6. Memilih tombol *close*	
	7. Menambah jumlah kotak informasi pada *level* 1
8. Menggerakan ulat ke tokoh pembantu *level* 1	
	9. Menampilkan kuis pada *level* 1
***Scenario alternative* 1 : kotak informasi tidak terambil, ulat mengenai daun**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 1
	2. Menampilkan ulat pada *level* 1
3. Menggerakan ulat ke kotak informasi *level* 1	
	4. Menambah *score level* 1 hasil dari mengenai daun
5. Menggerakan ulat ke kotak informasi *level* 1	
	6. Menampilkan informasi pada *level* 1
7. Memilih tombol *close*	
	8. Menambah jumlah kotak informasi pada *level* 1
9. Menggerakan ulat ke tokoh pembantu *level* 1	
	10. Menampilkan kuis pada *level* 1
***Scenario alternative* 2 : kotak informasi tidak terambil, ulat mengenai block**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 1
	2. Menampilkan ulat pada *level* 1
3. Menggerakan ulat ke kotak informasi *level* 1	
	4. Menampilkan ulat tidak bisa bergerak pada *level* 1
5. Menggerakan ulat ke kotak informasi *level* 1	
	6. Menampilkan informasi pada *level* 1
7. Memilih tombol *close*	
	8. Menambah jumlah kotak informasi pada *level* 1
9. Menggerakan ulat ke tokoh pembantu *level* 1	
	10. Menampilkan kuis pada *level* 1
***Scenario alternative* 3 : kotak informasi tidak terambil, ulat mengenai musuh dengan kondisi nyawa tidak sama dengan nol**	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 1
	2. Menampilkan ulat pada *level* 1
3. Menggerakan ulat ke kotak informasi *level* 1	
	4. Mengurangi nyawa *level* 1
5. Menggerakan ulat ke kotak informasi *level* 1	
	6. Menampilkan informasi pada *level* 1
7. Memilih tombol *close*	
	8. Menambah jumlah kotak informasi pada *level* 1
9. Menggerakan ulat ke tokoh pembantu *level* 1	
	10. Menampilkan kuis pada *level* 1
***Scenario alternative* 4 : kotak informasi tidak terambil, ulat mengenai musuh dengan kondisi nyawa sama dengan nol**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 1
	2. Menampilkan ulat pada *level* 1
3. Menggerakan ulat ke kotak informasi *level* 1	
	4. Mengurangi nyawa pada *level* 1
	5. Menampilkan pesan gagal

d. *Activity Diagram Use case* dan *Activity Diagram* Memainkan *Level* 1

AC-03 *Activity Diagram* untuk *Use case* UC-03 memainkan *level* 1



Gambar 3.8 AC-03 *Activity Diagram* untuk *Use case* UC-03 memainkan *level* 1

e. *Scenario Use case* dan *Activity Diagram* Memainkan *Level* 2

Nama *Use case* : memainkan *level* 2 (UC-04)

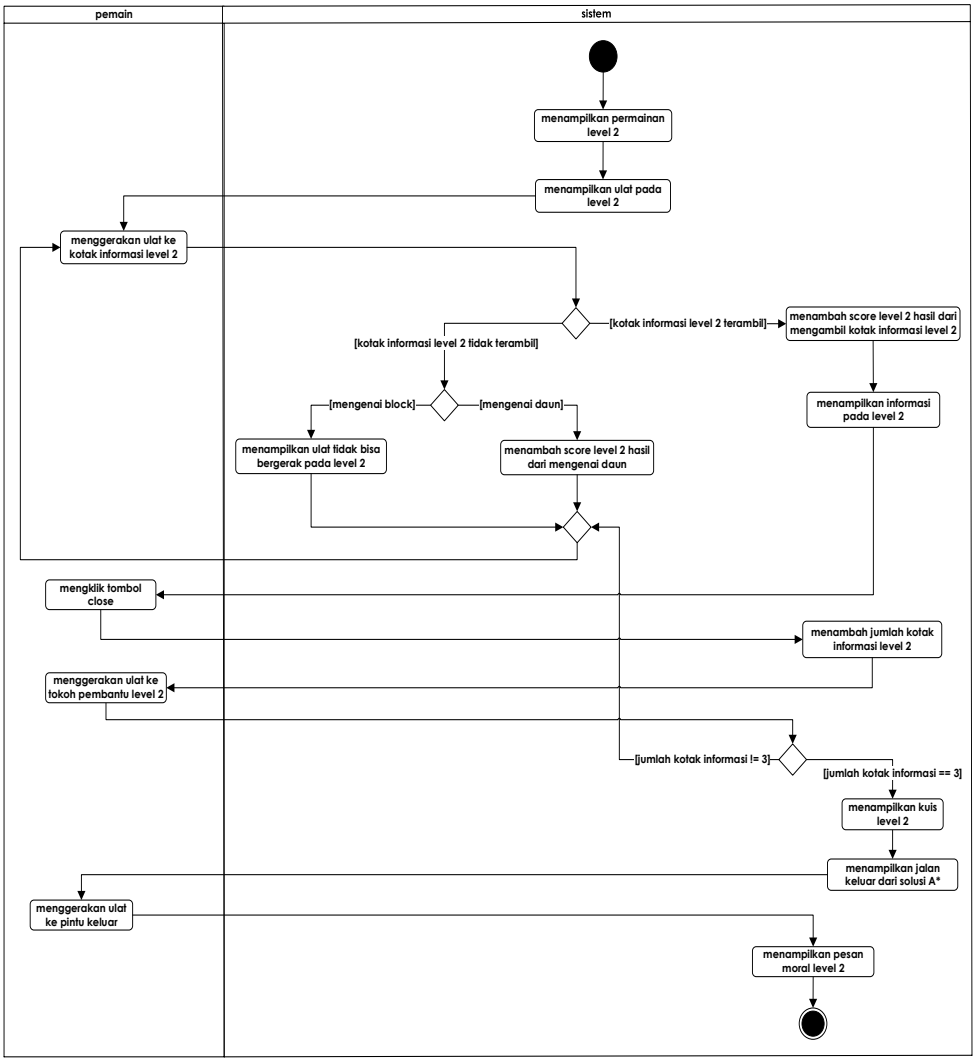
Berikut tabel *scenario use case* memainkan *level* 2

Tabel 3.13 *Scenario Use case* Memainkan *Level* 2

Kode Use case : **UC-04**	
Scenario normal : **pemain memainkan *level* 2**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 2
	2. Menampilkan ulat pada *level* 2
3. Menggerakan ulat ke kotak informasi *level* 2	
	4. Menambah *score level* 2 hasil dari mengambil kotak informasi pada *level* 2
	5. Menampilkan informasi pada *level* 2
6. Memilih tombol *close*	
	7. Menambah jumlah kotak informasi pada *level* 2
8. Menggerakan ulat ke tokoh pembantu *level* 2	
	9. Menampilkan kuis level 2
	10. Menampilkan jalan keluar dari solusi A*
11. Menggerakan ulat ke pintu keluar	
	12. Menampilkan pesan moral level 2
***Scenario alternative* 1 : kotak informasi tidak terambil, ulat mengenai block**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 2
	2. Menampilkan ulat pada *level* 2
3. Menggerakan ulat ke kotak informasi *level* 2	
	4. Menampilkan ulat tidak bisa bergerak pada *level* 2
5. Menggerakan ulat ke kotak informasi *level* 2	
	6. Menambah *score level* 2 hasil dari mengambil kotak informasi pada *level* 2
	7. Menampilkan informasi pada *level* 2
8. Memilih tombol *close*	
	9. Menambah jumlah kotak informasi pada *level* 2
10. Menggerakan ulat ke tokoh pembantu *level* 2	
	11. Menampilkan kuis level 2
	12. Menampilkan jalan keluar dari solusi A*
13. Menggerakan ulat ke pintu keluar	
	14. Menampilkan pesan moral level 2
***Scenario alternative* 2 : kotak informasi tidak terambil, ulat mengenai daun**	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan permainan *level* 2
	2. Menampilkan ulat pada *level* 2
3. Menggerakan ulat ke kotak informasi *level* 2	
	4. Menambah *score level* 2 hasil dari mengenai daun

5. Menggerakan ulat ke kotak informasi *level* 2	
	6. Menambah *score level* 2 hasil dari mengambil kotak informasi *level* 2
	7. Menampilkan informasi pada *level* 2
8. Memilih tombol *close*	
	9. Menambah jumlah kotak informasi pada *level* 2
10. Menggerakan ulat ke tokoh pembantu *level* 2	
	11. Menampilkan pertanyaan level 2
	12. Menampilkan jalan keluar dari solusi A*
13. Menggerakan ulat ke pintu keluar	
	14. Menampilkan pesan moral level 2

AC-04 *Activity Diagram* untuk *Use case* UC-04 memainkan *level* 2



Gambar 3.9 AC-04 *Activity Diagram* untuk *Use case* UC-04 memainkan *level* 2

f. *Scenario Use case* dan *Activity Diagram* Memainkan *Level* 3

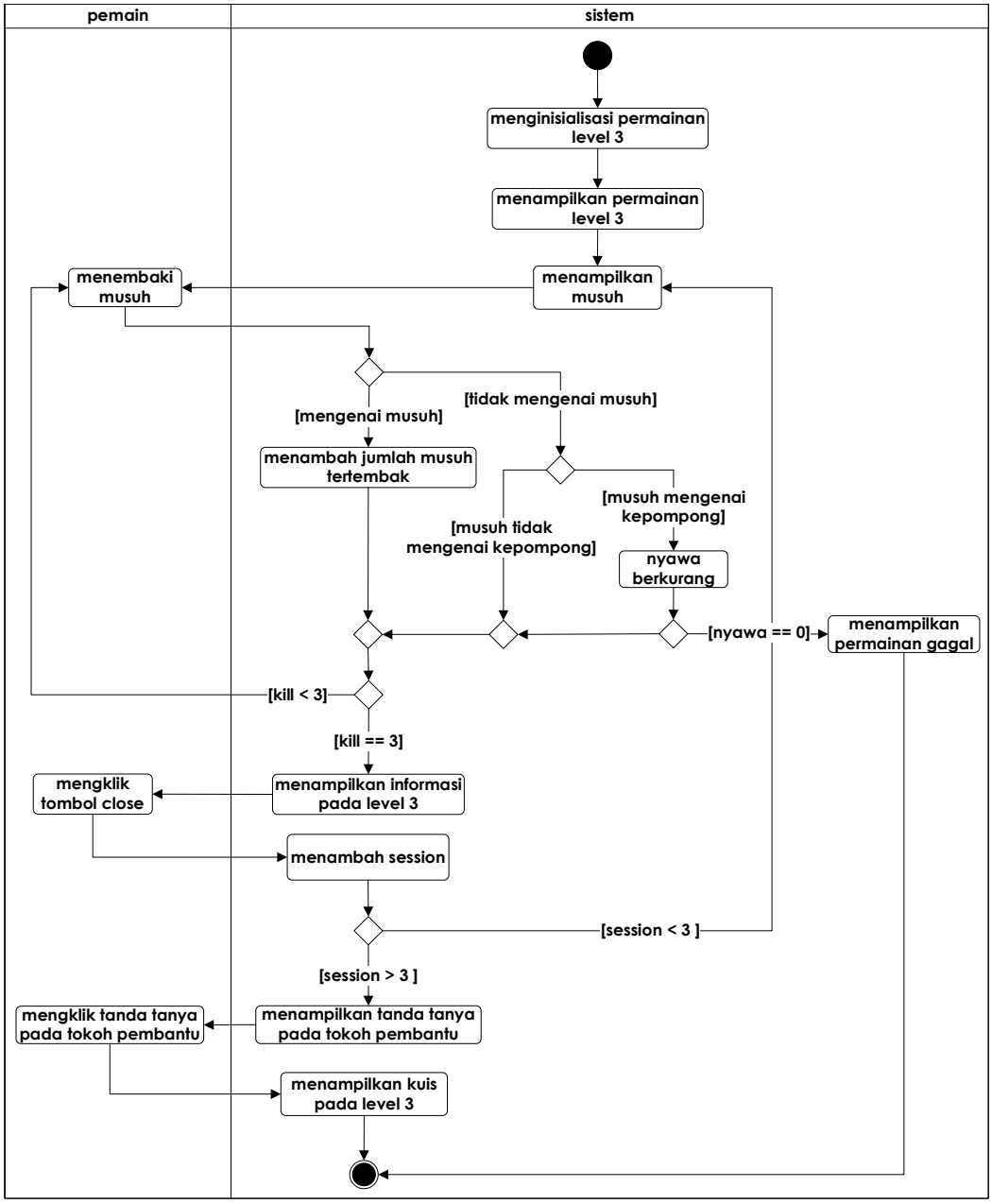
Nama *Use case* : memainkan *level* 3 (UC-5)

Berikut tabel *scenario use case* memainkan *level* 3.

Tabel 3.14 *Use case Scenario* Memainkan *Level* 3

Kode Use case : **UC-05**	
Scenario normal : **pemain memainkan *level* 3**	
Aksi Aktor	Reaksi Sistem
	1. Menginisialisasi permainan *level* 3
	2. Menampilkan permainan *level* 3
	3. Menampilkan musuh
4. Menembaki musuh	
	5. Menambah jumlah musuh tertembak
	6. Musuh mati sama dengan 3 maka menampilkan informasi pada level 3
7. Memilih tombol *close*	
	8. Menambah *session*
	9. *Session* lebih dari 3 maka menampilkan tanda tanya pada tokoh pembantu
10. Memilih tanda tanya pada tokoh pembantu	
	11. Menampilkan kuis pada *level* 3
***Scenario alternative* 1 : pemain memainkan permainan dengan kondisi musuh mengenai kepompong dan nyawa sama dengan nol**	
Aksi Aktor	Reaksi Sistem
	1. Menginisialisasi permainan *level* 3
	2. Menampilkan permainan *level* 3
	3. Menampilkan musuh
4. Menembaki musuh	
	5. Musuh mengenai kepompong maka nyawa berkurang
	6. Nyawa sama dengan nol maka menampilkan permainan gagal

AC-05 *Activity Diagram* untuk *Use case* UC-05 memainkan *level* 3



Gambar 3.10 AC-05 *Activity Diagram* untuk *Use case* UC-05 memainkan level 3

g. *Scenario Use case* dan *Activity Diagram* Menjawab Kuis

Nama *Use case* : menjawab kuis (UC-06)

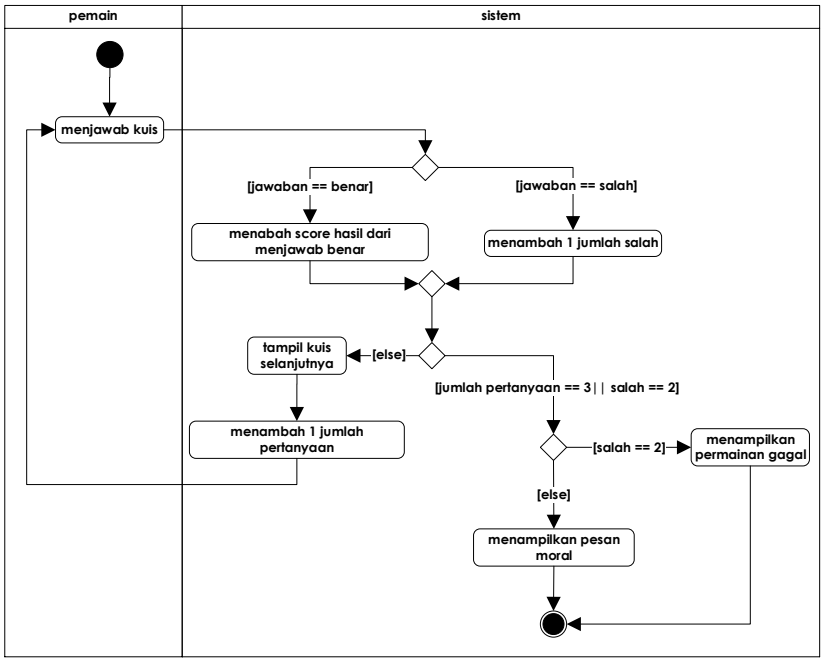
Berikut tabel *scenario use case* menjawab kuis.

Tabel 3.15 *use case scenario* menjawab kuis

Kode Use case : **UC-06**	
Scenario normal : **pemain menjawab kuis**	
Aksi Aktor	Reaksi Sistem

1. Menjawab kuis	
	2. Menambah *score* hasil dari menjawab benar
	3. Jumlah pertanyaan sudah sama dengan 3 dengan salah tidak sama dengan 2 maka menampilkan pesan moral
***Scenario alternative* 1 : gagal menjawab kuis**	
Aksi Aktor	Reaksi Sistem
1. Menjawab kuis	
	2. Menjawab salah maka menambah 1 jumlah salah
	3. Jumlah pertanyaan belum sama dengan 3 dengan salah tidak sama dengan 2 maka tampil kuis selanjutnya
	4. Menambah 1 jumlah pertanyaan
	5. Jumlah pertanyaan sudah sama dengan 3 dengan salah sama dengan 2 maka permainan gagal

AC-06 *Activity Diagram* untuk *Use case* UC-06 menjawab kuis.



Gambar 3.11 AC-06 *Activity Diagram* untuk *Use case* UC-06 menjawab kuis

h. *Scenario Use case* dan *Activity Diagram* Melihat *Menu* Bantuan

Nama *Use case* : melihat *menu* bantuan (UC-07)

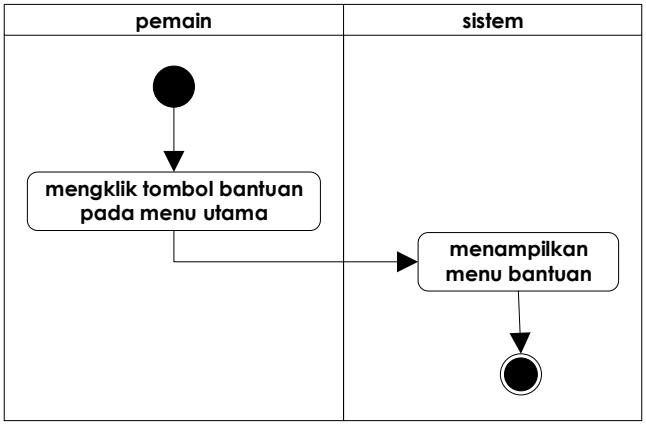
Berikut tabel *scenario use case* melihat *menu* bantuan.

Tabel 3.16 *use case scenario* melihat *menu* bantuan

***Kode Use case :* UC-07**	
***Scenario* normal : pemain melihat *menu* bantuan**	
Aksi Aktor	Reaksi Sistem
1. Memilih tombol bantuan pada *menu* utama	

	2. Menampilkan *menu* bantuan

AC-07 *Activity Diagram* untuk *Use case* UC-07 melihat *menu* bantuan.



Gambar 3.12 AC-07 *Activity Diagram* untuk *Use case* UC-07 melihat *menu* bantuan

i. *Scenario Use case* dan *Activity Diagram* Mengatur Suara

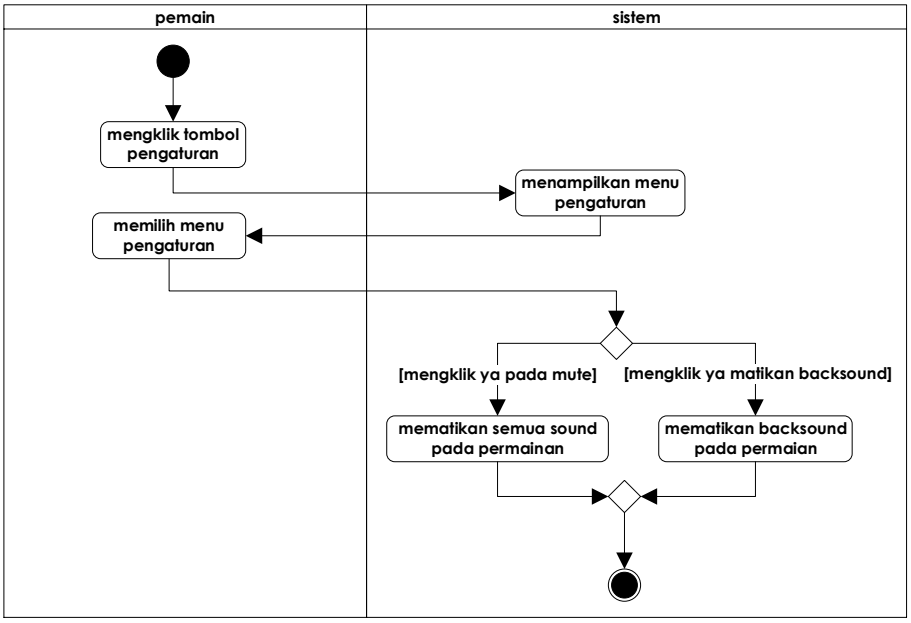
Nama *Use case* : mengatur suara (UC-08)

Berikut tabel *scenario use case* mengatur suara.

Tabel 3.17 *Scenario Use case* Mengatur Suara

***Kode Use case :* UC-08**	
Scenario normal : **pemain mengatur suara**	
Aksi Aktor	Reaksi Sistem
1. Memilih tombol pengaturan	
	2. Menampilkan *menu* pengaturan
3. Memilih *menu* pengaturan	
***Scenario alternative* 1 : mematikan backsound**	
Aksi Aktor	Reaksi Sistem
1. Memilih tombol pengaturan	
	2. Menampilkan *menu* pengaturan
3. Memilih ya pada matikan *backsound*	
	4. Mematikan *backsound* permaian
***Scenario alternative* 2 : mematikan semua *sound* pada permainan**	
Aksi Aktor	Reaksi Sistem
1. Memilih tombol pengaturan	
	2. Menampilkan *menu* pengaturan
3. Memilih ya pada *mute*	
	4. Mematikan semua *sound* pada permaian

AC-08 *Activity Diagram* untuk *Use case* UC-08 mengatur suara

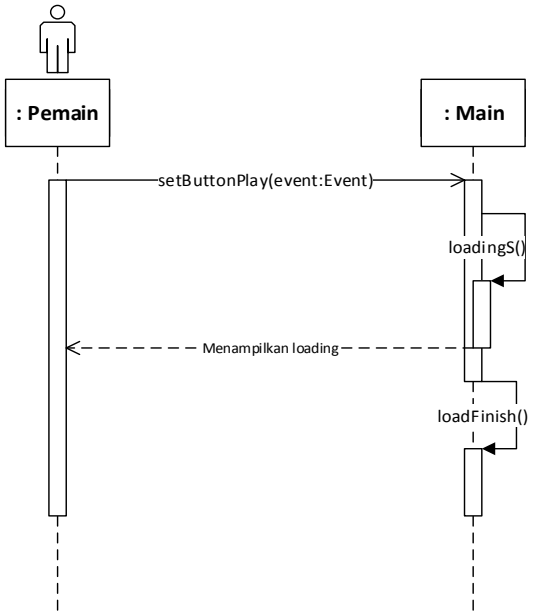


Gambar 3.13 AC-08 Activity Diagram untuk Use case UC-08 mengatur suara

3.1.6.3 Sequence *Diagram*

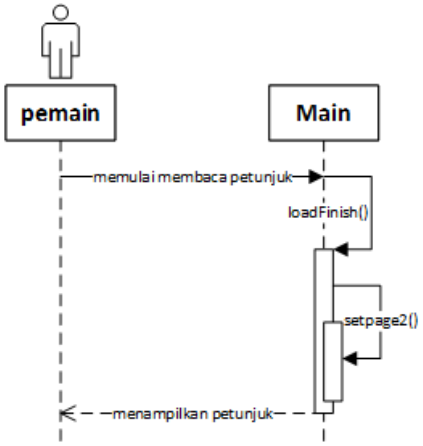
Sequence diagram menggambarkan kelakuan objek *class* pada *use case* dengan mendeskripsikan *life line* dan *message* yang dikirimkan dan diterima antara *object* atau *class*.

1. ***Sequence Diagram* Memulai Permainan**



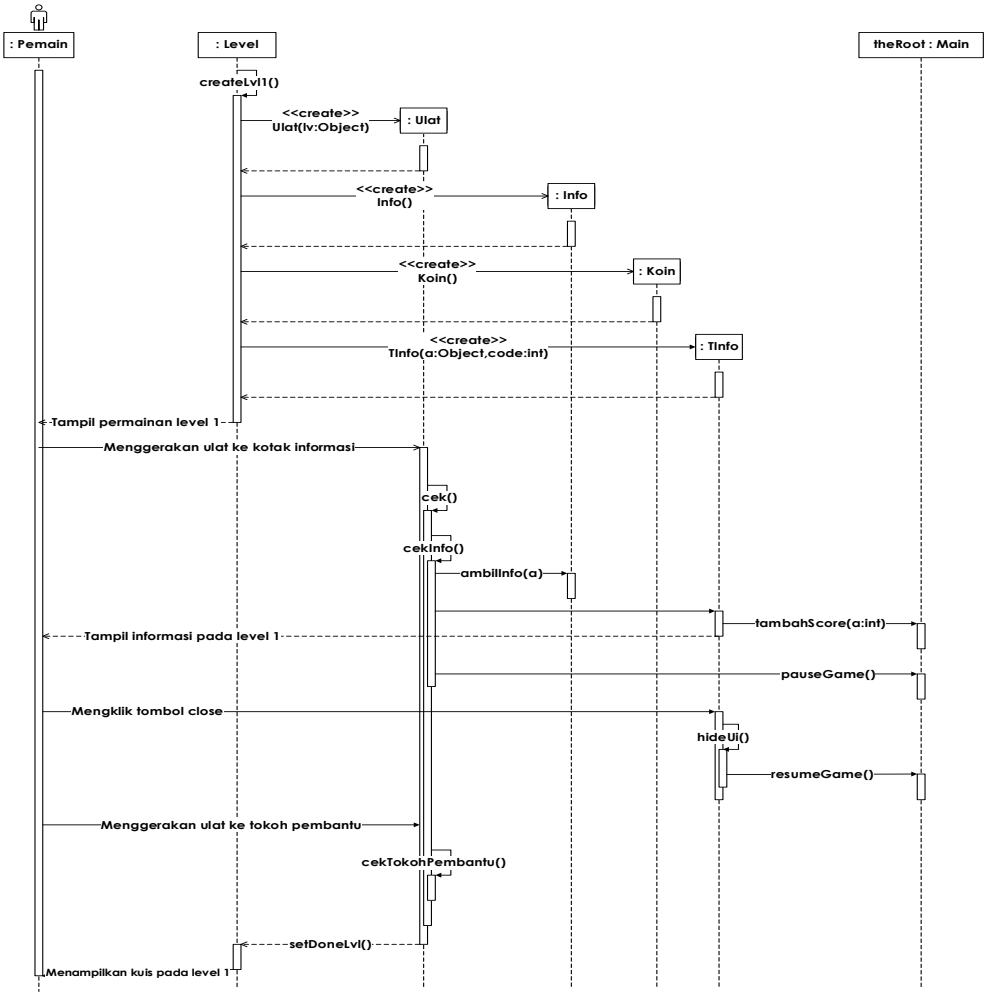
Gambar 3.14 *Sequence Diagram* Memulai Permainan

2. *Sequence Diagram* Membaca Petunjuk



Gambar 3.15 *Sequence Diagram* Membaca Petunjuk

3. *Sequence Diagram* Memainkan *Level* 1

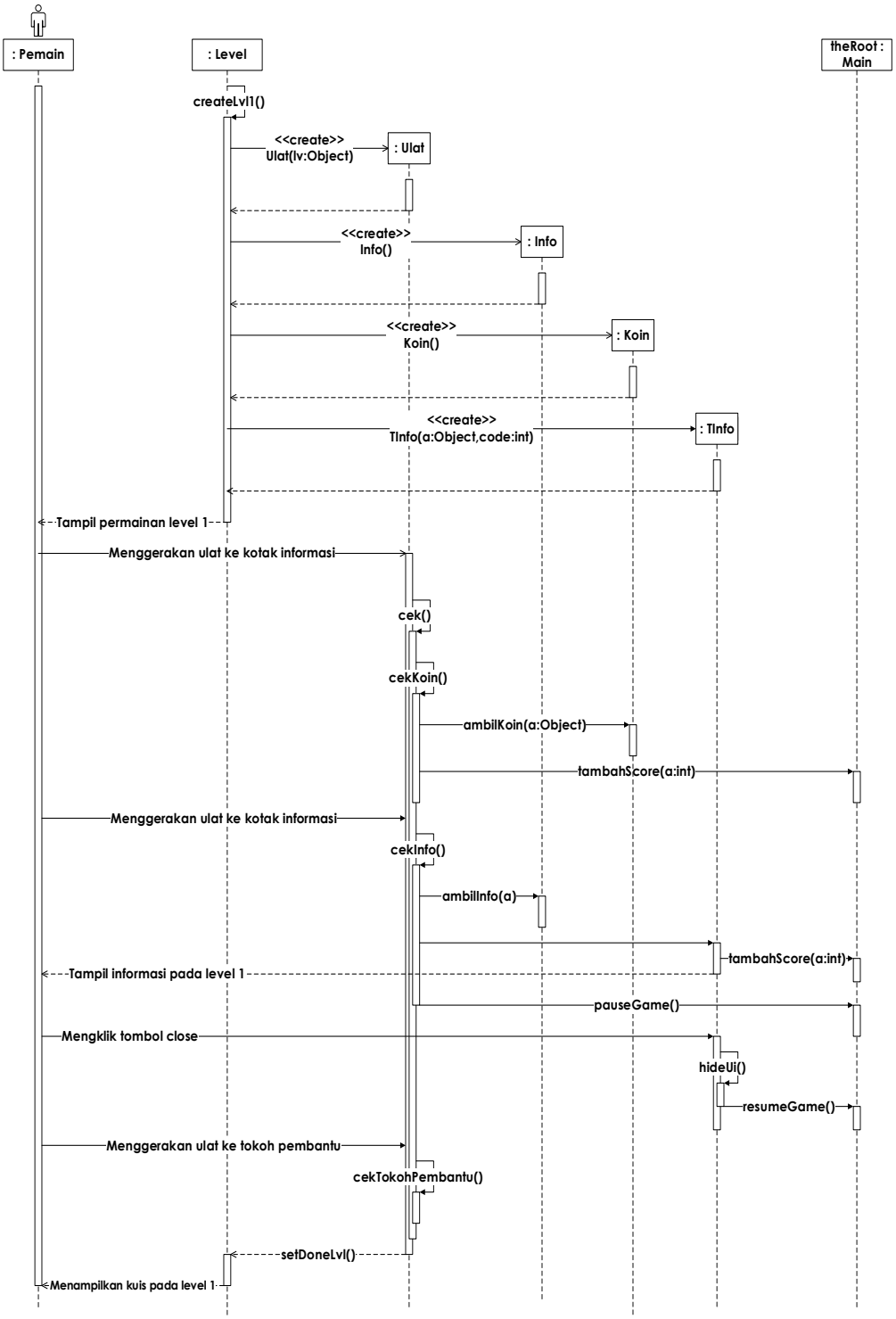


Gambar 3.16 *Sequence Diagram* Memainkan *Level* 1

a. Sequence Diagram Alternative Memainkan *Level* 1

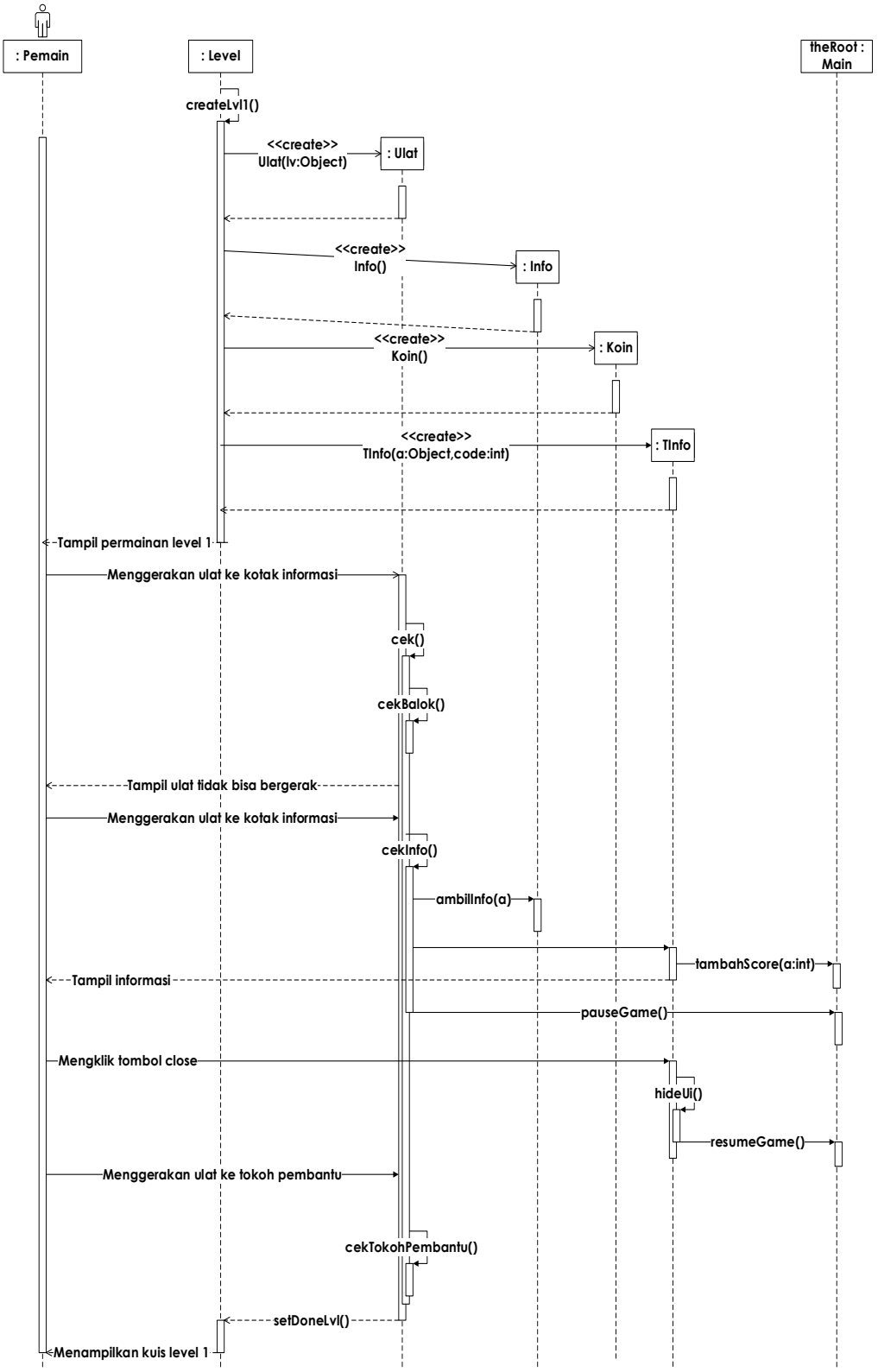
Ada 4 *sequence diagram scenario alternative* memainkan *level* 1, yaitu :

Scenario alternative 1 : kotak informasi tidak terambil, ulat mengenai daun



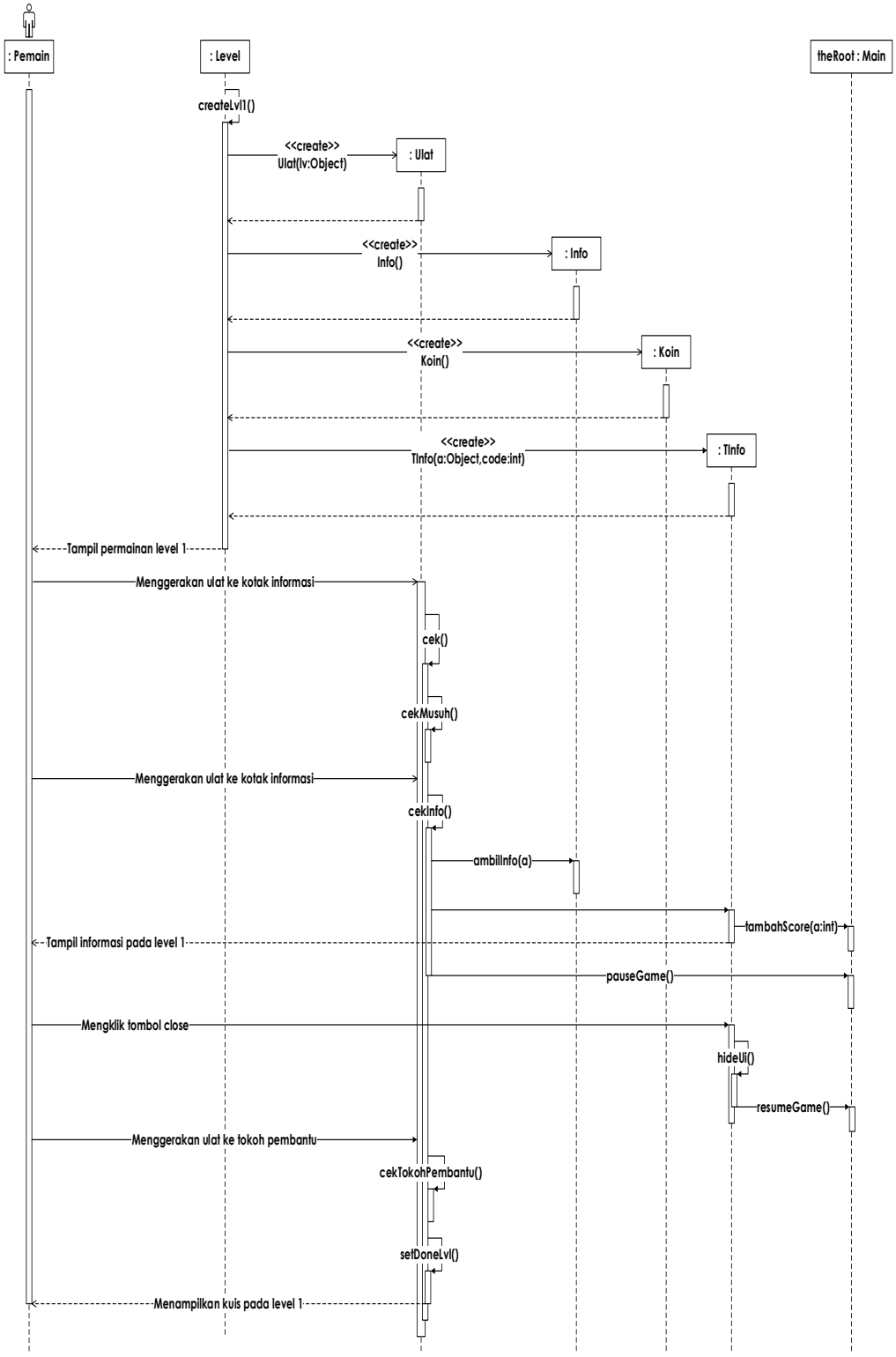
Gambar 3.17 *Sequence Diagram Alternative* 1 Memainkan *Level* 1

Scenario alternative 2 : kotak informasi tidak terambil, ulat mengenai *block*



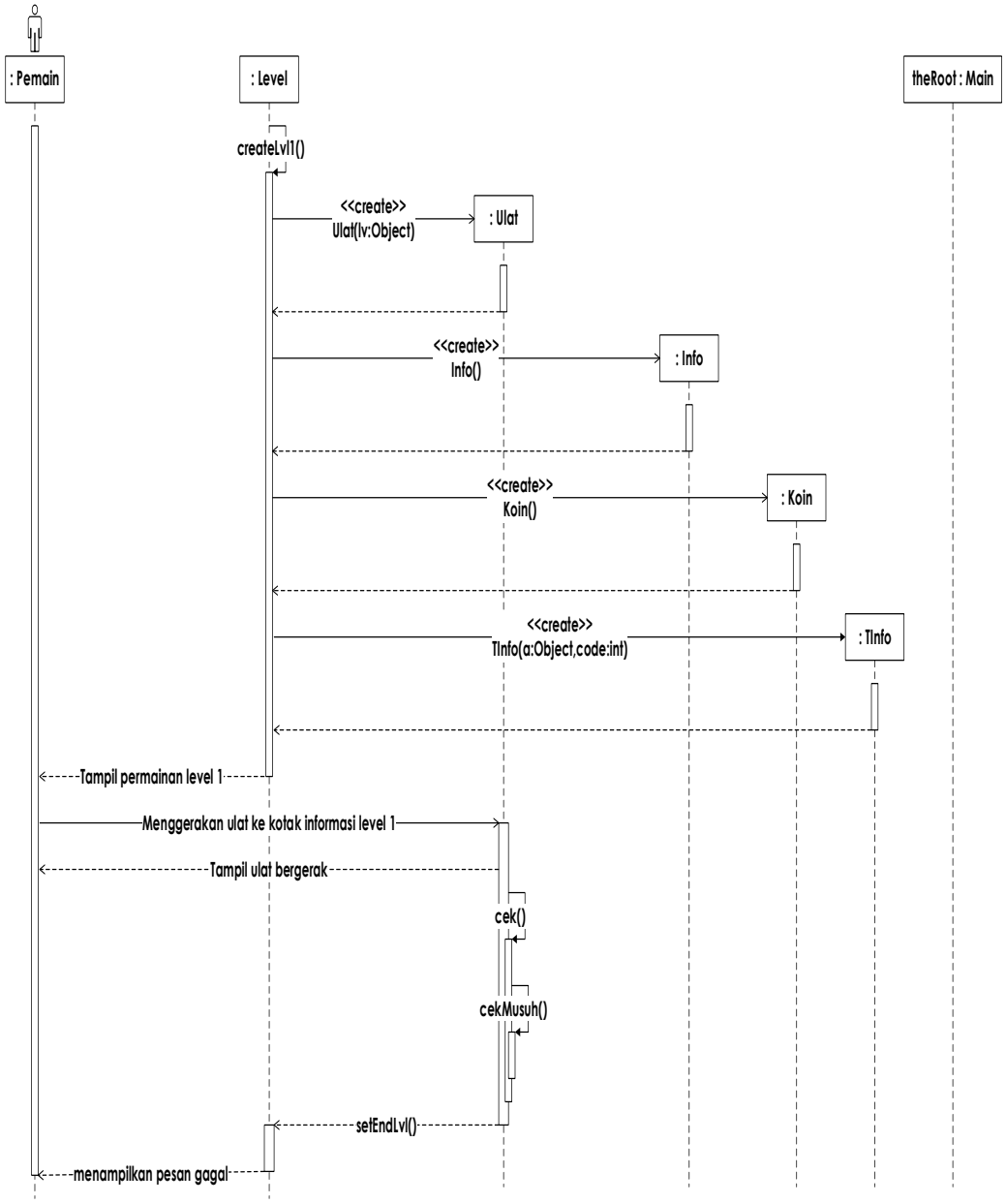
Gambar 3.18 *Sequence Diagram Alternative* 2 Memainkan *Level* 1

Scenario alternative 3 : kotak informasi tidak terambil, ulat mengenai musuh dengan kondisi nyawa tidak sama dengan nol



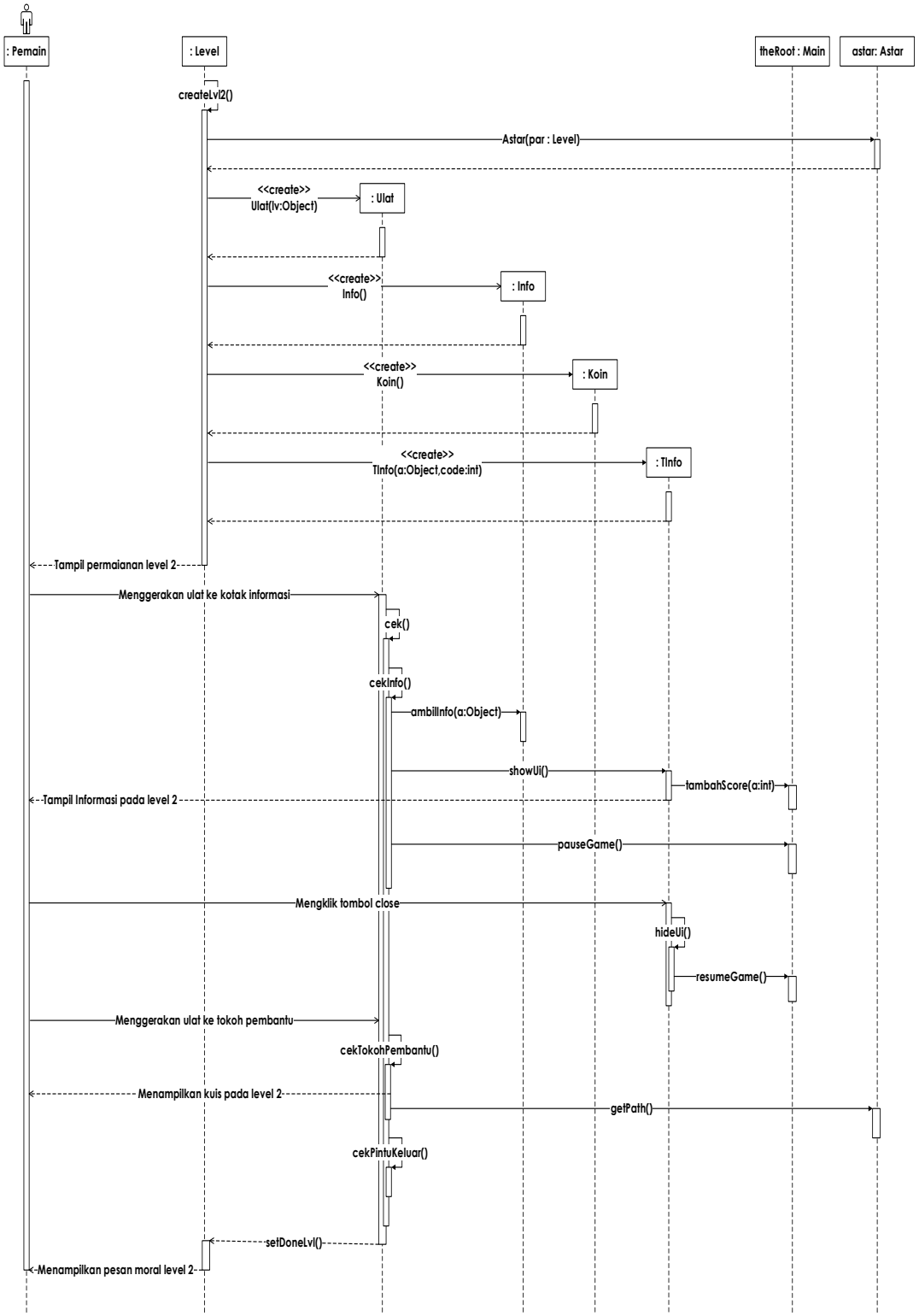
Gambar 3.19 *Sequence Diagram Alternative* 3 Memainkan *Level* 1

Scenario alternative 4 : kotak informasi tidak terambil, ulat mengenai musuh dengan kondisi nyawa sama dengan nol



Gambar 3.20 *Sequence Diagram Alternative* 4 Memainkan *Level* 1

4. *Sequence Diagram* Memainkan *Level* 2

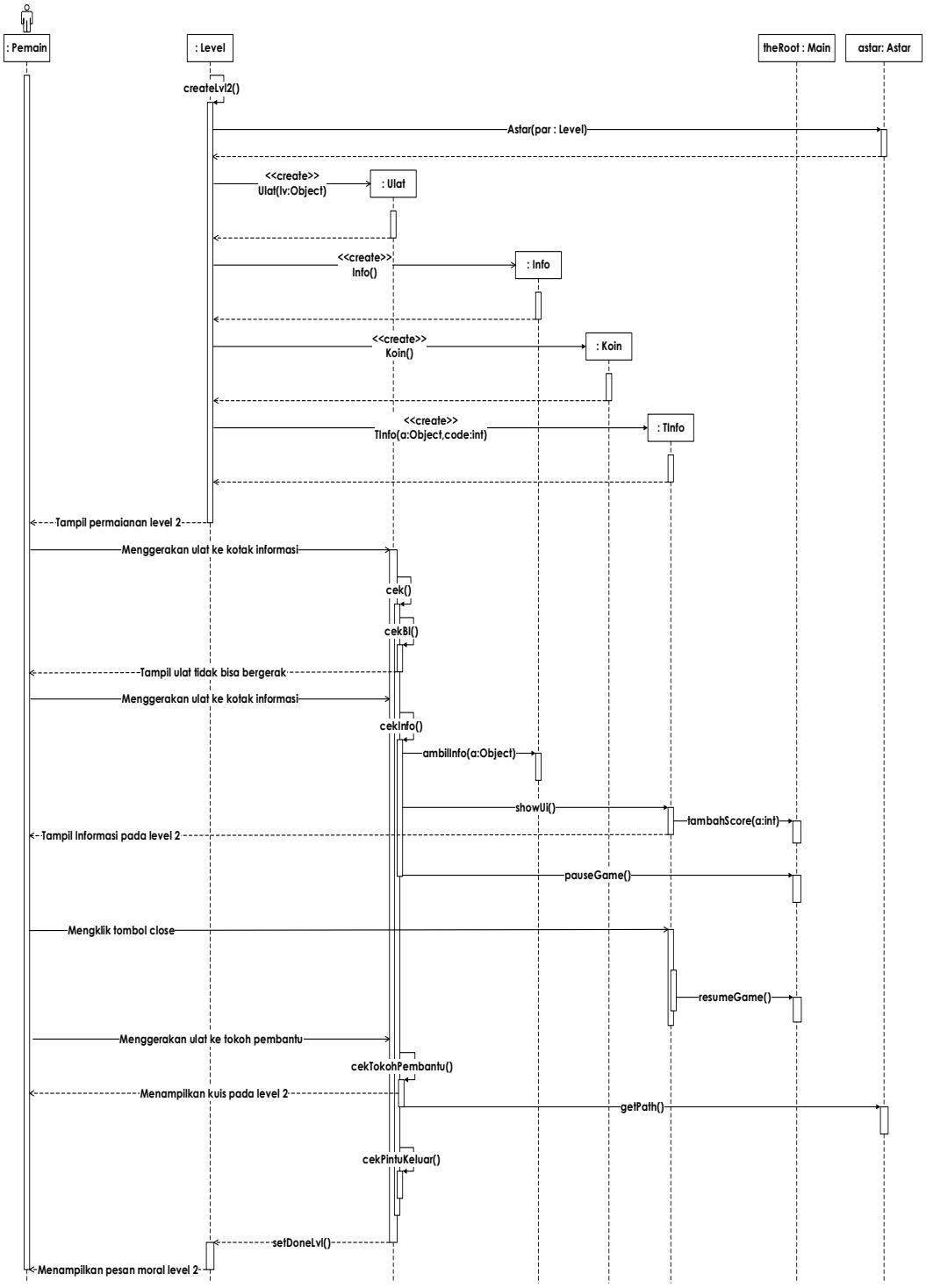


Gambar 3.21 *Sequence Diagram* Memainkan *Level* 2

a. *Sequence Diagram Alternative* Memainkan *Level* 2

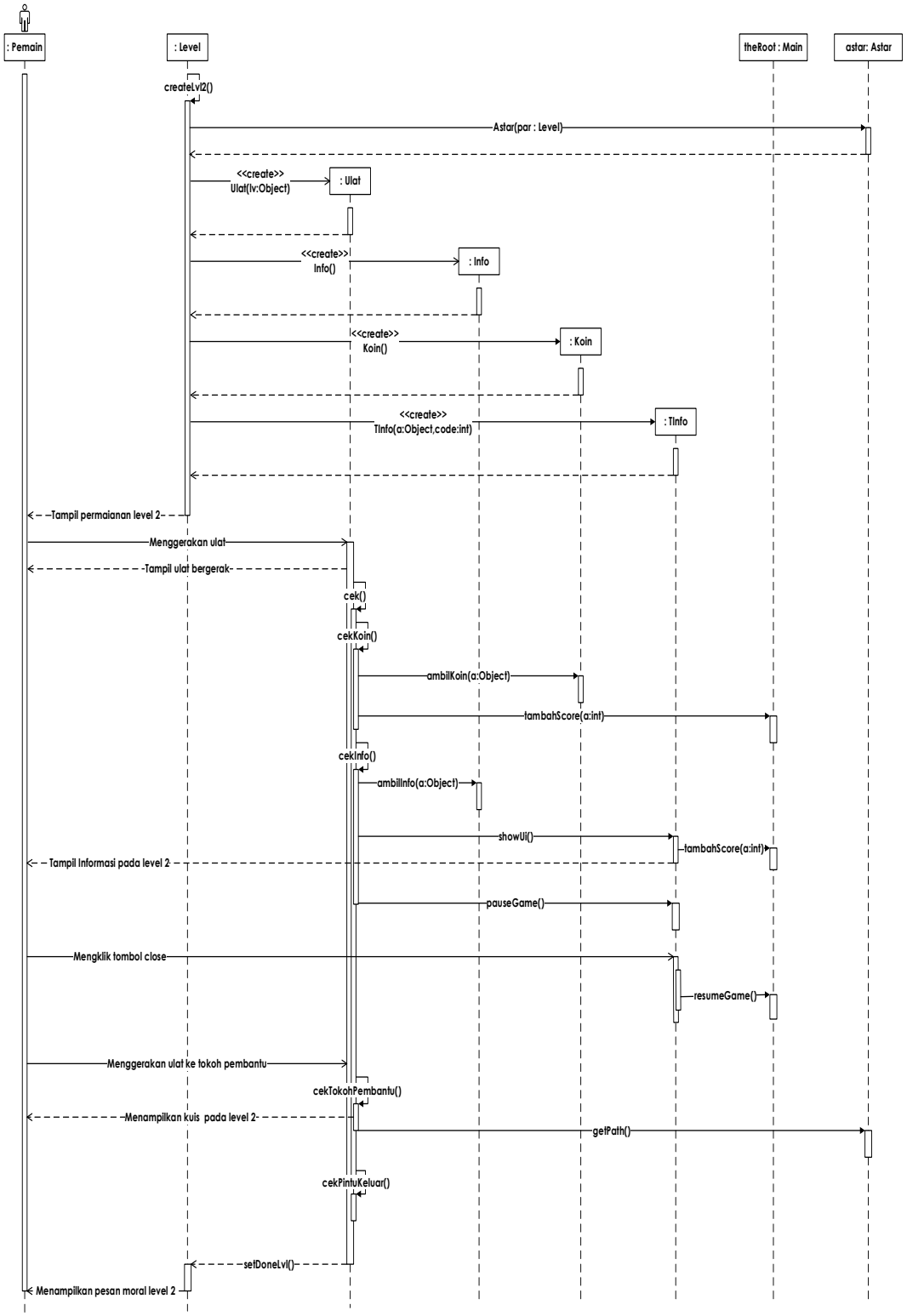
Ada 2 *sequence diagram scenario alternative* memainkan *level* 2, yaitu :

Scenario alternative 1 : kotak informasi tidak terambil, ulat mengenai block



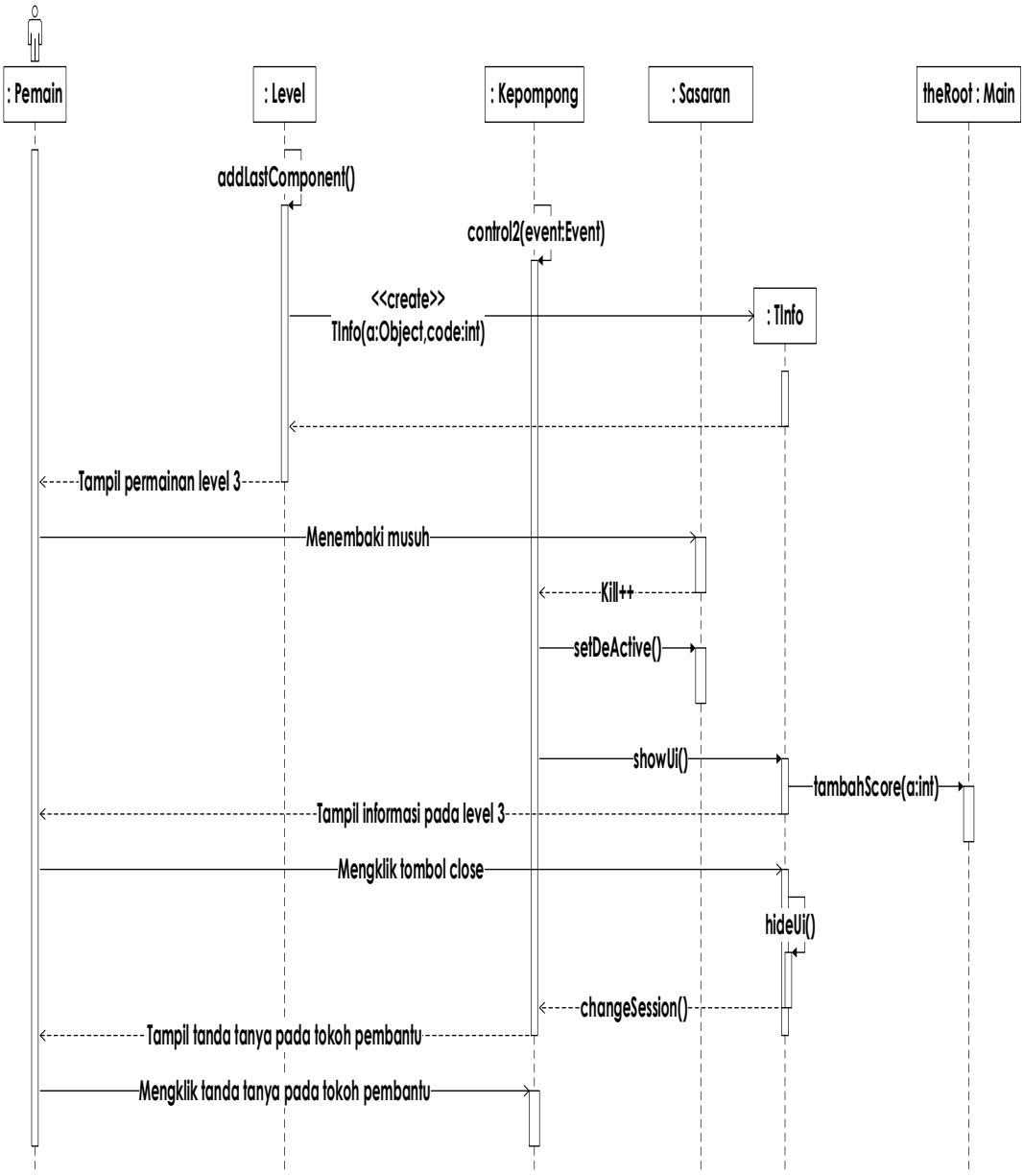
Gambar 3.22 *Sequence Diagram Alternative* 1 Memainkan *Level* 2

Scenario alternative 2 : kotak informasi tidak terambil, ulat mengenai daun



Gambar 3.23 *Sequence Diagram Alternative* 2 Memainkan *Level* 2

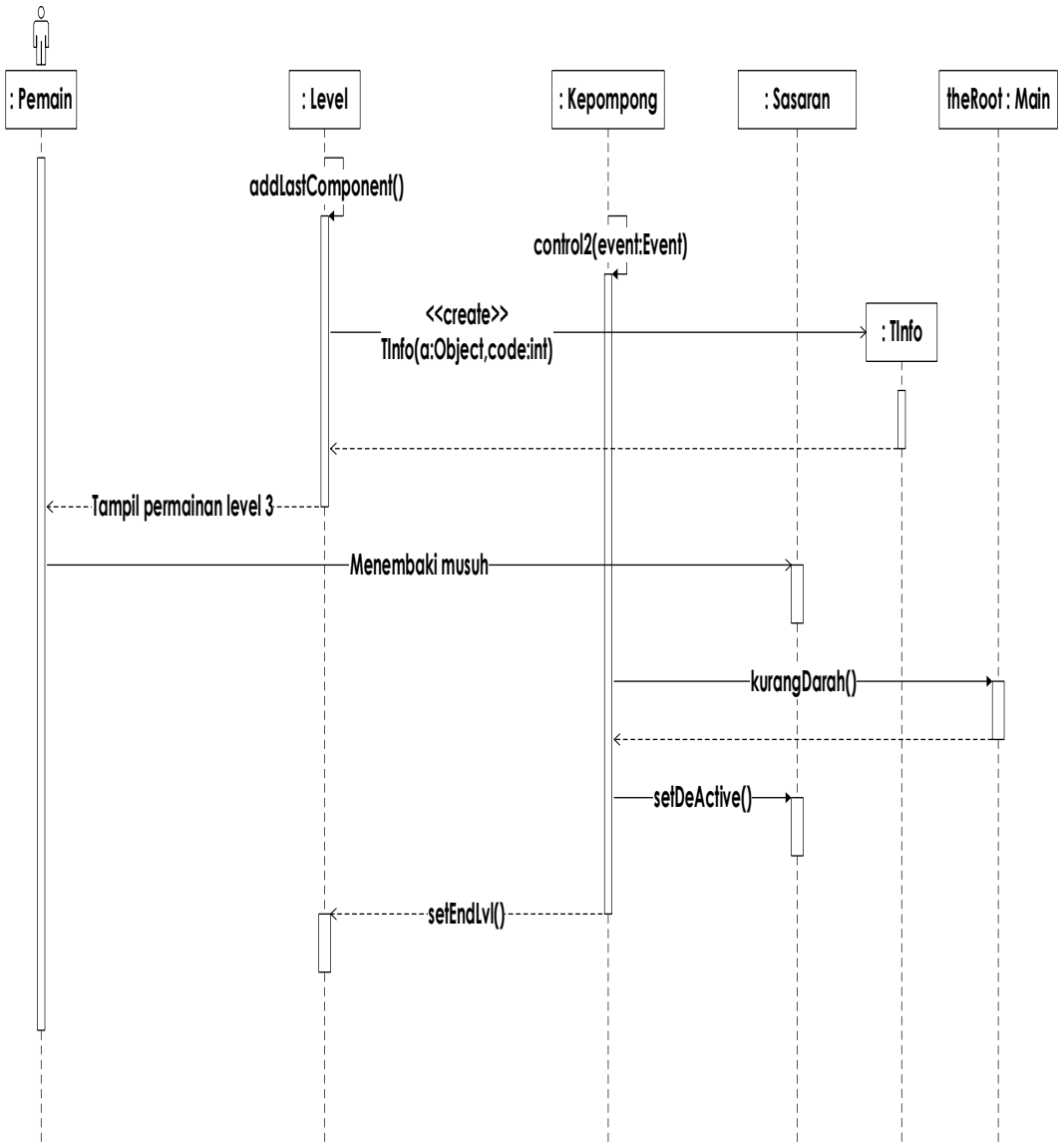
5. *Sequence Diagram* Memainkan *Level* 3



Gambar 3.24 *Sequence Diagram* Memainkan *Level* 3

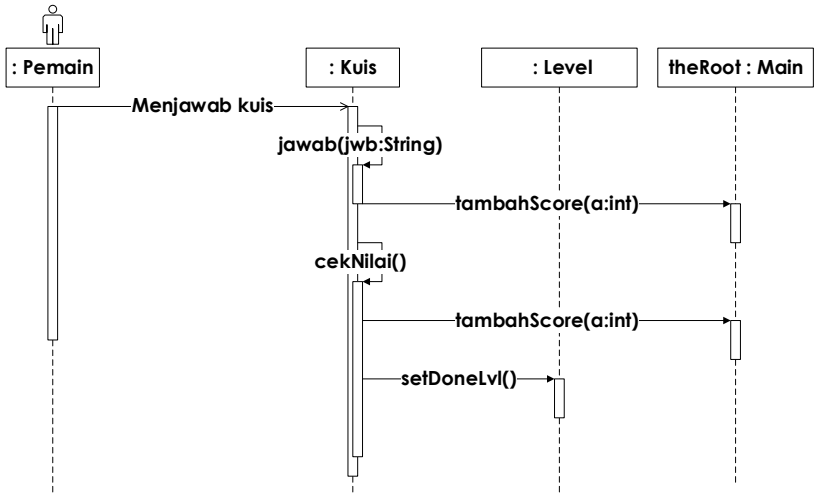
a. *Sequence Diagram Alternative* Memainkan *Level* 3

Sequence diagram alternative 1, pemain memainkan permainan dengan kondisi musuh mengenai kepompong dan nyawa sama dengan nol



Gambar 3.25 *Sequence Diagram Alternative* Memainkan *Level* 3

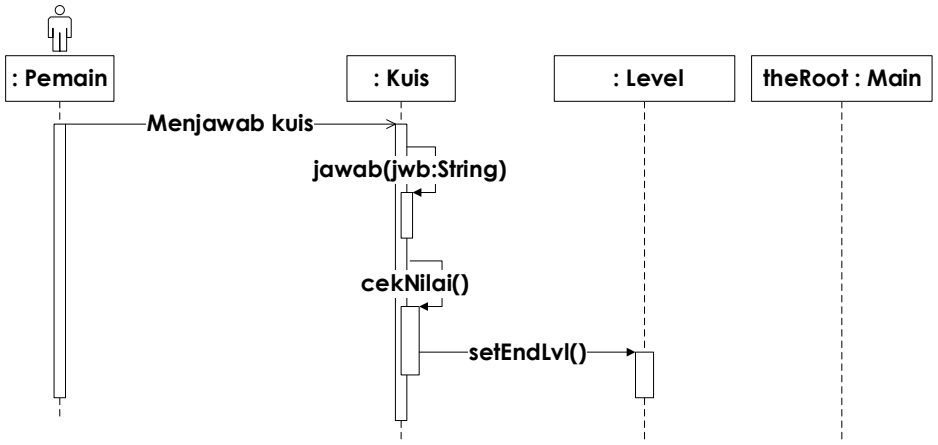
6. *Sequence Diagram* Menjawab Kuis



Gambar 3.26 *Sequence Diagram* Menjawab Kuis

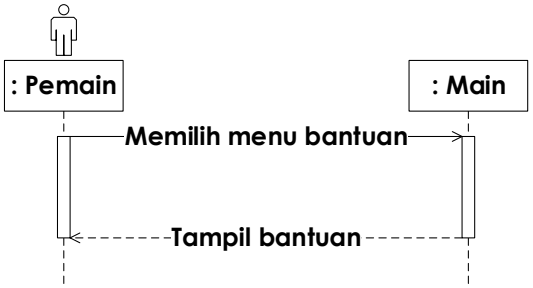
a. Sequence Diagram Alternative Menjawab Kuis

Sequence diagram alternative 1, gagal menjawab kuis



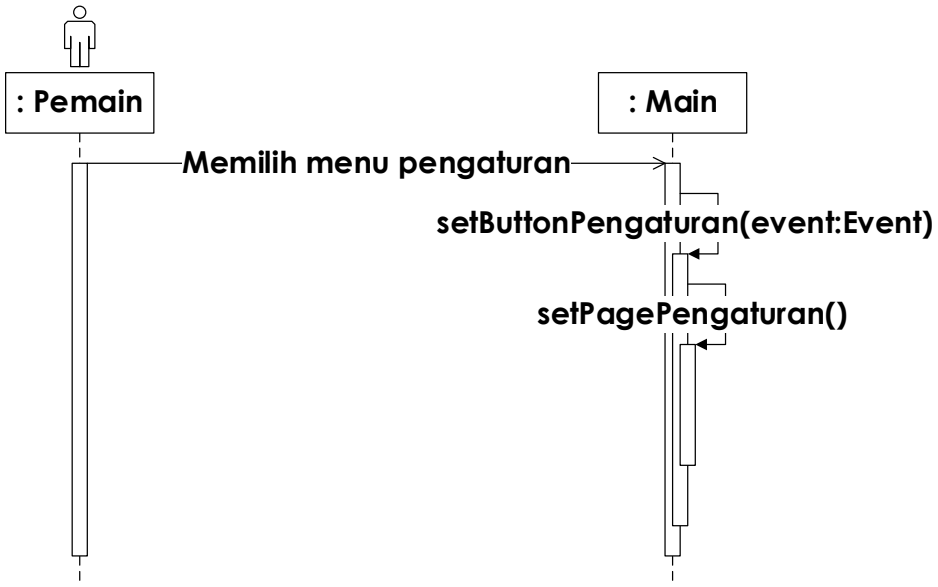
Gambar 3.27 *Sequence Diagram Alternative* Menjawab Kuis

7. *Sequence Diagram* Melihat *Menu* Bantuan



Gambar 3.28 *Sequence Diagram* Melihat *Menu* Bantuan

8. *Sequence Diagram* Mengatur Suara

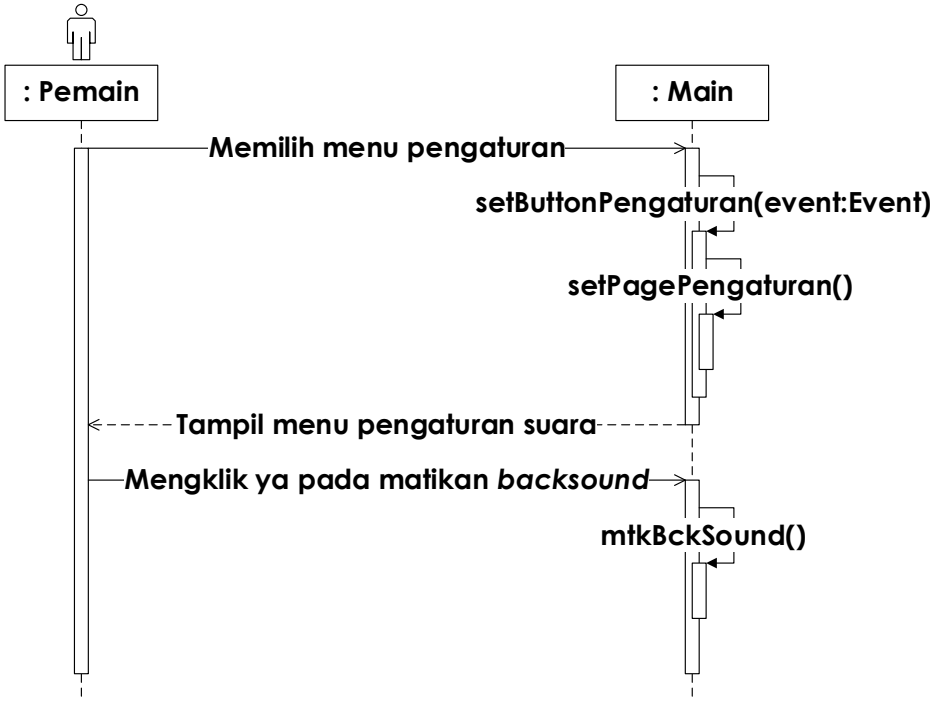


Gambar 3.29 *Sequence Diagram* Mengatur Suara

a. *Sequence Diagram Alternative* Mengatur Suara

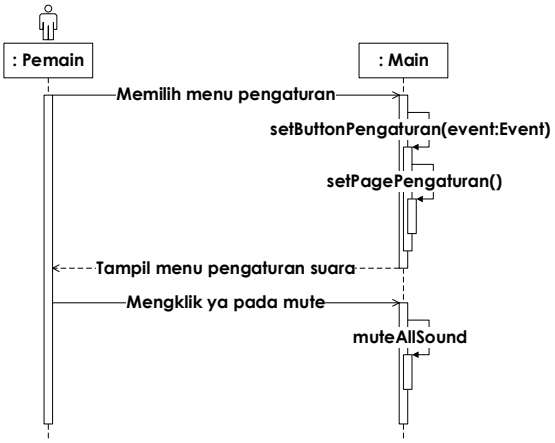
Ada 2 *sequence diagram alternative* mengatur suara, yaitu :

Alternative 1 : mematikan *backsound*



Gambar 3.30 *Sequence Diagram Alternative* 1 Mengatur Suara

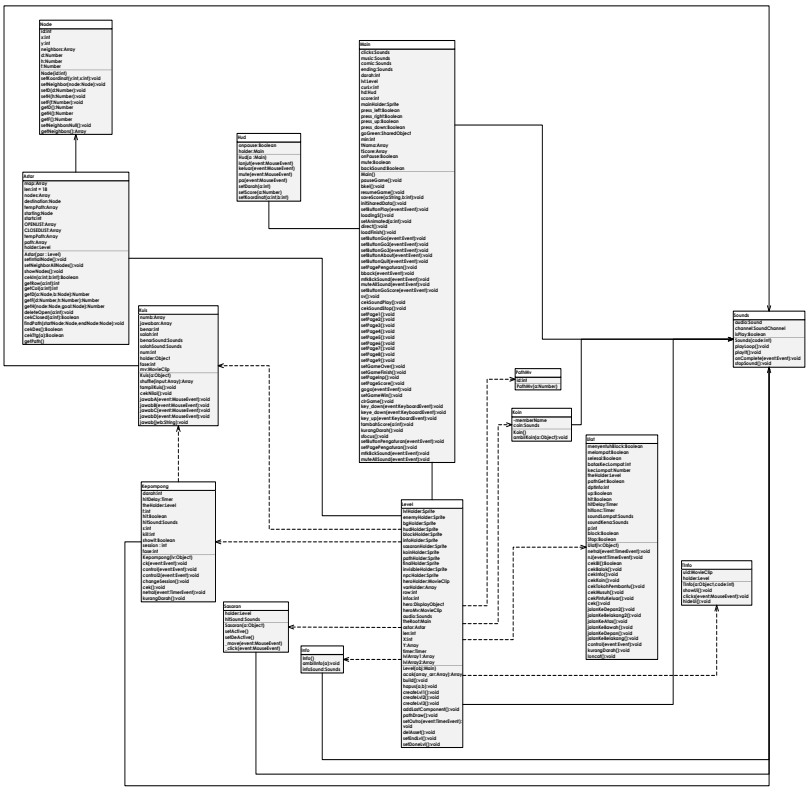
Alternative 2 : mematikan semua *sound* pada permainan



Gambar 3.31 *Sequence Diagram Alternative* 2 Mengatur Suara

3.1.6.4 Class Diagram

Diagram kelas atau atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Berikut ini adalah perancangan struktur sistem yang terdapat pada *game go green* bersama petualangan si ulat, yang digambarkan dengan *class diagram*.



Gambar 3.32 *Class Diagram*

3.2 Perancangan Sistem

Di dalam bagian ini akan dijelaskan perancangan komponen, perancangan antarmuka, perancangan pesan dan jaringan semantik dan perancangan metode dari *game go green* bersama petualangan si ulat.

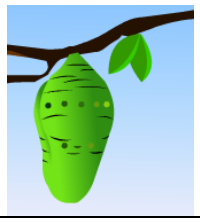
3.2.1 Perancangan Komponen

Dalam bagian ini dijelaskan karakter yang terdapat pada *game* dan *storyboard game go green* bersama petualangan si ulat.

3.2.1.1 Karakter

Dalam bagian ini dijelaskan karakter-karakter yang terdapat pada *game go green* bersama petualangan si ulat, seperti yang dapat dilihat pada tabel berikut ini.

Tabel 3.18 Karakter

No	Nama	Gambar	Keterangan
1	Ulat		• Merupakan karakter utama yang akan dimainkan oleh pemain • Mati jika menabrak burung hantu dan salah menjawab pertanyaan dari karakter pembantu • Karakter utama ini memiliki 3 Healthpoint
2	kepompong		• Merupakan karakter utama yaitu ulat yang berubah menjadi kepompong pada *level* 3 • Mati jika salah menjawab pertanyaan • Mati jika ditabrak burung hantu dan polusi
3	Kupu-kupu		• Merupakan karakter utama ulat yang berhasil bermetamorfosis karena dapat menjawab pertanyaan dan mengalahkan semua musuh-musuhnya selama berlangsung
4	Mr. akar		• Merupakan karakter pembantu pada *level* 1 • Memberi pertanyaan • Memberi pesan moral tentang *global warming* dan *go green*.

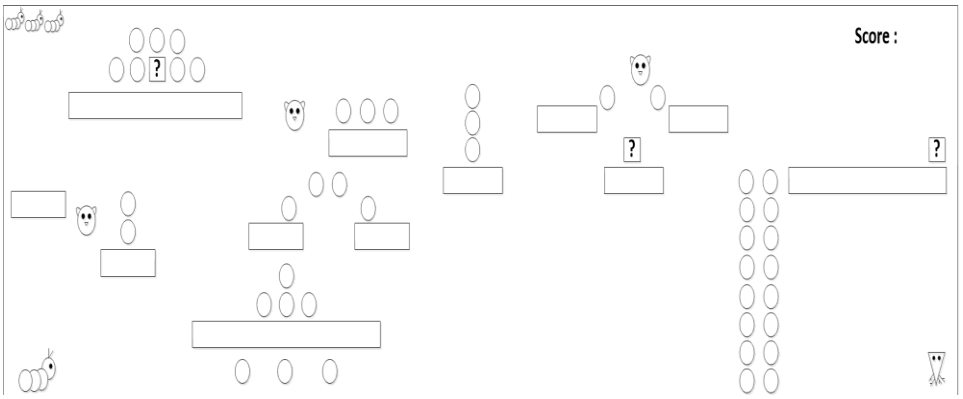
5	Mr. batang		• Merupakan karakter pembantu pada *level* 2 • Memberi pertanyaan dan *path finding* • Memberi pesan moral tentang *global warming* dan *go green*.
6	Mr. daun		• Merupakan karakter pembantu pada *level* 3 • Memberi pertanyaan • Memberi pesan moral tentang *global warming* dan *go green*.
7	Mr. sun		• Merupakan karakter pembantu pada *level* 3 • Memberi pertanyaan • Memberi pesan moral tentang *global warming* dan *go green*
8	Burung hantu		• Merupakan karakter musuh pada *level* 1 dan *level* 3 • Pada *level* 1 menghalang-halangi jalan • Pada *level* 3 mengganggu kopompong
9	Raja burung hantu		• Merupakan karakter musuh yang ingin memangsa ulat pada *level* 1 dan *level* 3
10	polusi		• Merupakan karakter musuh pada *level* 3 • Mengganggu dengan cara menabrak kopompong
11	Raja polusi		• Merupakan karakter musuh pada *level* 3 • *Menu*tupi Mr. Sun
12	Kotak informasi		• Berisi informasi-informasi untuk dapat menjawab pertanyaan pada setiap level

3.2.1.2 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* ide cerita dapat disampaikan kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji sehingga menghasilkan persepsi yang sama.

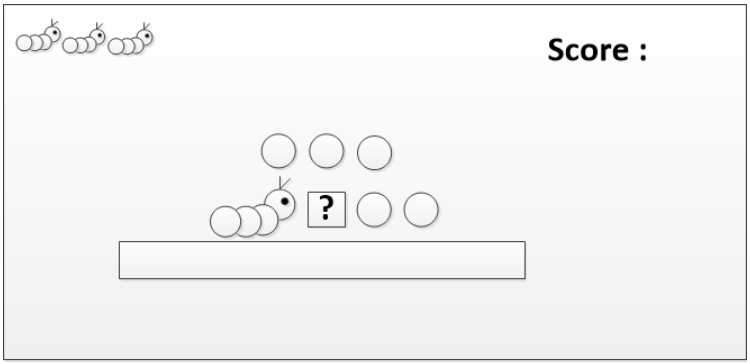
Pada bagian ini akan dijelaskan *storyboard* dari *gameplay go green* bersama petualangan si ulat *level* 1, *level* 2 *level* 3, pemberian materi, dan pertanyaannya.

1. *Storyboard level* 1



Gambar 3.33 Storyboard *Level* 1

Pada *level* 1 terdapat 3 kotak informasi. Kotak informasi tersebut harus diambil oleh pemain.



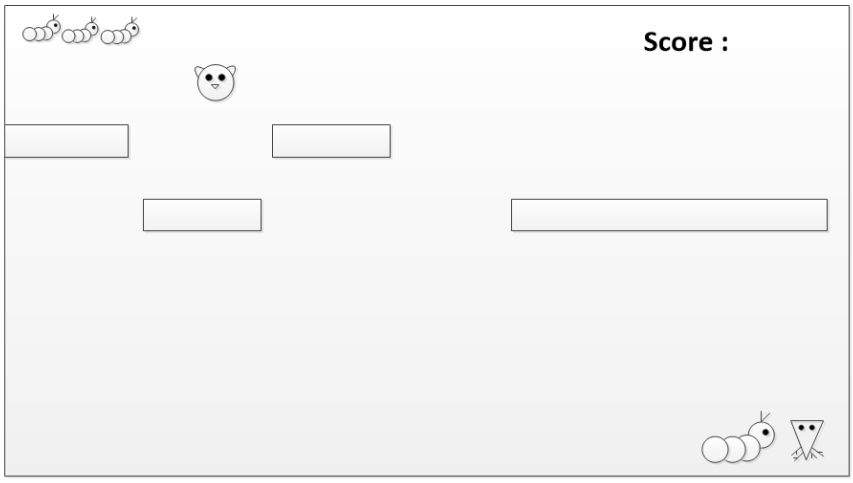
Gambar 3.34 Kotak Informasi *Level* 1

Ketika pemain mendapatkan kotak informasi, akan muncul penjelasan materi.



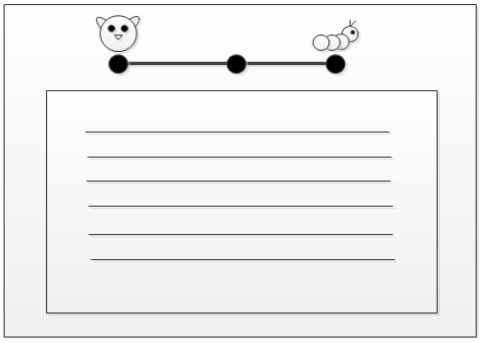
Gambar 3.35 Penjelasan Materi *Level* 1

Dalam penjelasan materi terdapat pula gambar penjelasan. Untuk kembali melanjutkan pemain harus menekan *GO*.



Gambar 3.36 Bertemu Tokoh Pembantu *Level* 1

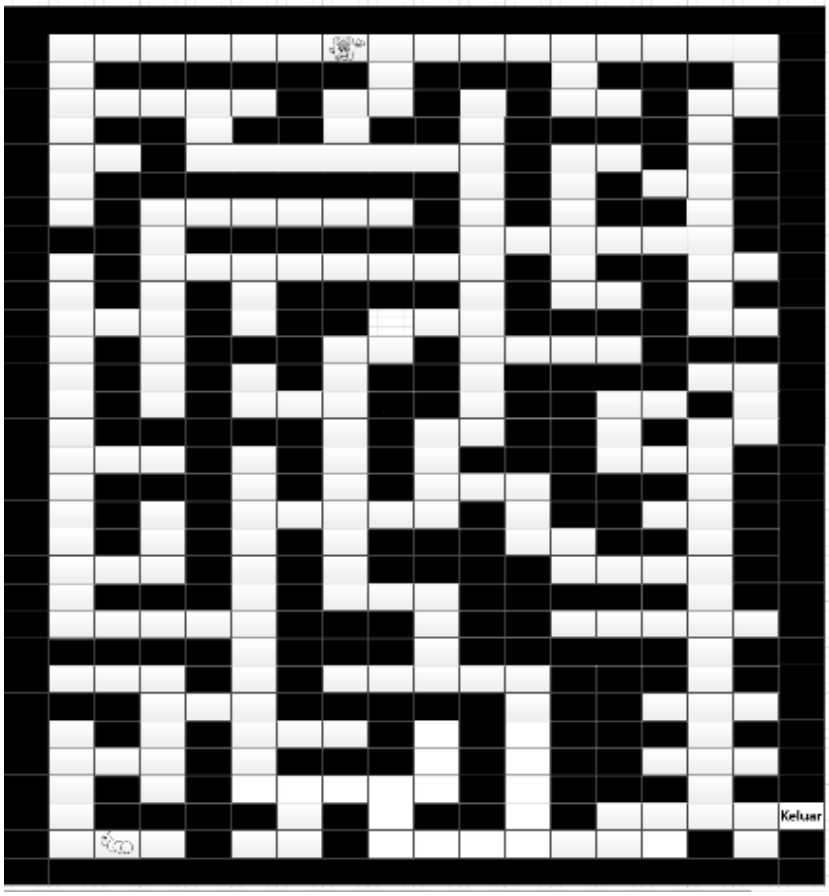
Setelah semua kotak informasi didapatkan pemain, pemain harus menemui Mr. Akar untuk menjawab pertanyaan.



Gambar 3.37 Pertanyaan *Level* 1

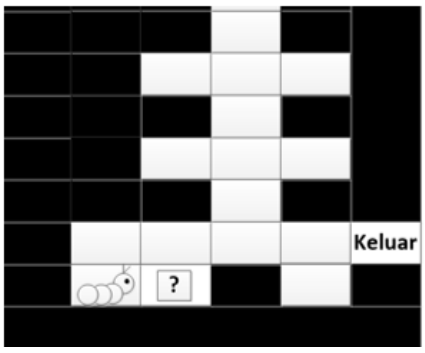
Pemain harus menjawab pertanyaan dari Mr.Akar. bentuk pertanyaan adalah pilihan ganda.

2. *Storyboard level* 2



Gambar 3.38 Storyboard *Level* 2

level 2 adalah labirin, pada *area* ini terdapat 3 kotak informasi. Kotak informasi tersebut harus diambil oleh pemain.



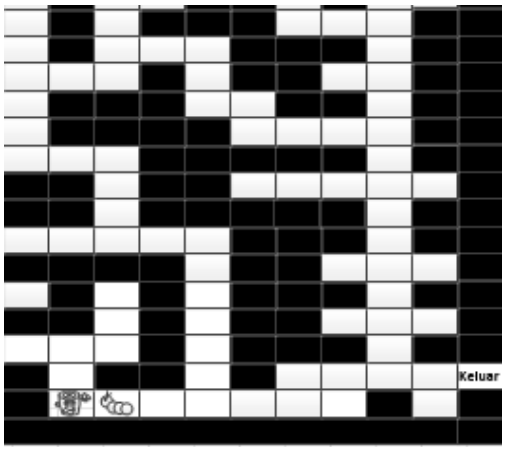
Gambar 3.39 Kotak Informasi *Level* 2

Ketika pemain mendapatkan kotak informasi, akan muncul penjelasan materi.



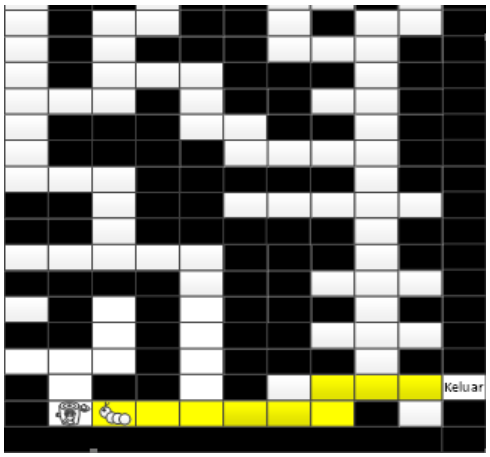
Gambar 3.40 Penjelasan Materi *Level* 2

Didalam penjelasan materi terdapat pula gambar penjelasan. Untuk kembali melanjutkan permainan pemain harus menekan *GO*.



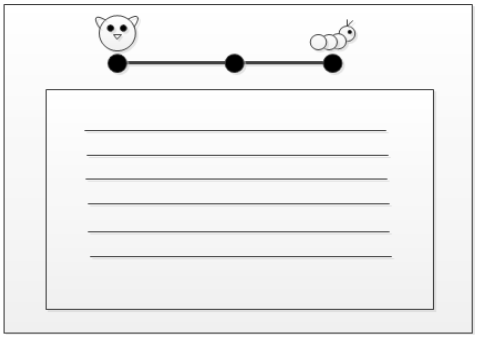
Gambar 3.41 Bertemu Tokoh Pembantu *Level* 2

Setelah pemain mendapatkan semua kotak informasi, pemain harus menemui Mr. Batang.



Gambar 3.42 Petunjuk Jalan Terpendek

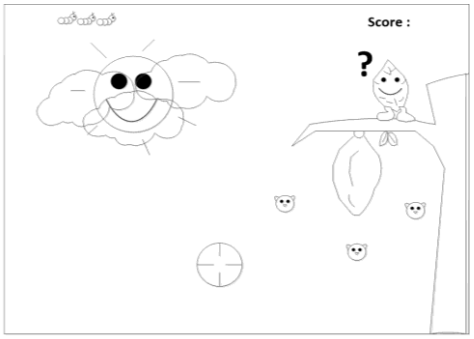
Pemain diberi petunjuk jalan terpendek dan diberi pertanyaan untuk dapat keluar dari labirin setelah semua kotak informasi terambil.



Gambar 3.43 Pertanyaan *Level* 2

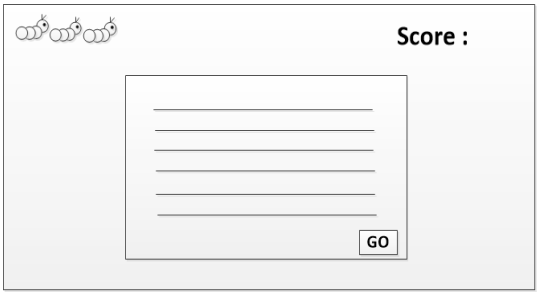
Pemain harus menjawab pertanyaan dari Mr. Barang. Bentuk pertanyaan adalah pilihan ganda.

3. *Storyboard level* 3



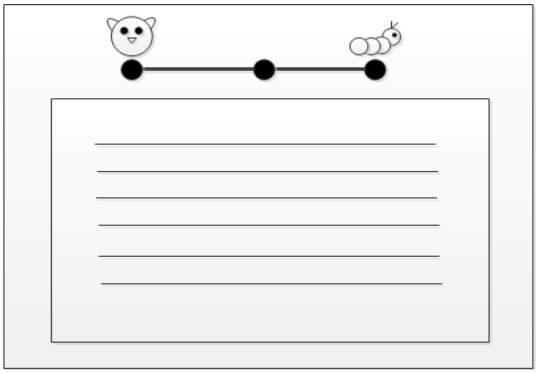
Gambar 3.44 *Level* 3 Musuh Burung Hantu

Pada *area level* 3 ini pemain harus melindungi kepompong dengan cara menembak 3 burung hantu dan pemain akan mendapatkan informasi. Setelah semua burung hantu tertembak pemain harus memilih tanda tanya pada Mr. Daun.



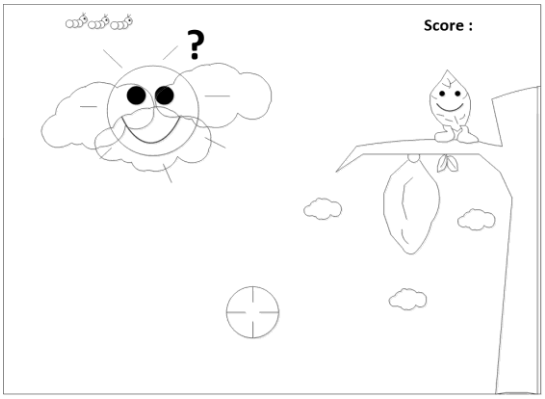
Gambar 3.45 Penjelasan Informasi *Level* 3

Dalam penjelasan materi terdapat pula gambar penjelasan. Untuk kembali melanjutkan permainan pemain harus menekan *GO*.



Gambar 3.46 Pertanyaan *Level* 3 Dari Tokoh Pembantu 1

Pemain harus menjawab pertanyaan dari Mr. Daun untuk dapat melanjutkan . Bentuk pertanyaan pilihan ganda.



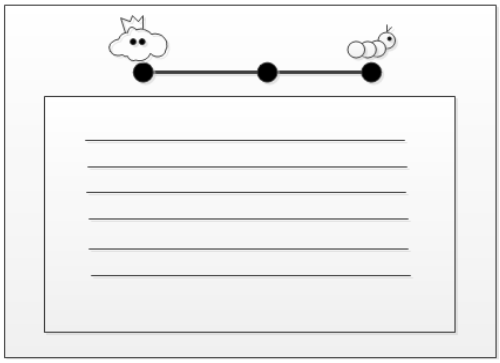
Gambar 3.47 *Level* 3 Musuh Polusi

Setelah menjawab pertanyaan dari Mr. Daun pemain harus melindungi kepompong dengan cara menembak 3 polusi dan pemain akan mendapatkan informasi. Setelah semua polusi tertembak pemain harus memilih tanda tanya pada Mr. Sun.



Gambar 3.48 Penjelasan Materi *Level* 3

Dalam penjelasan materi terdapat pula gambar penjelasan. Untuk kembali melanjutkan permainan pemain harus menekan *GO*.

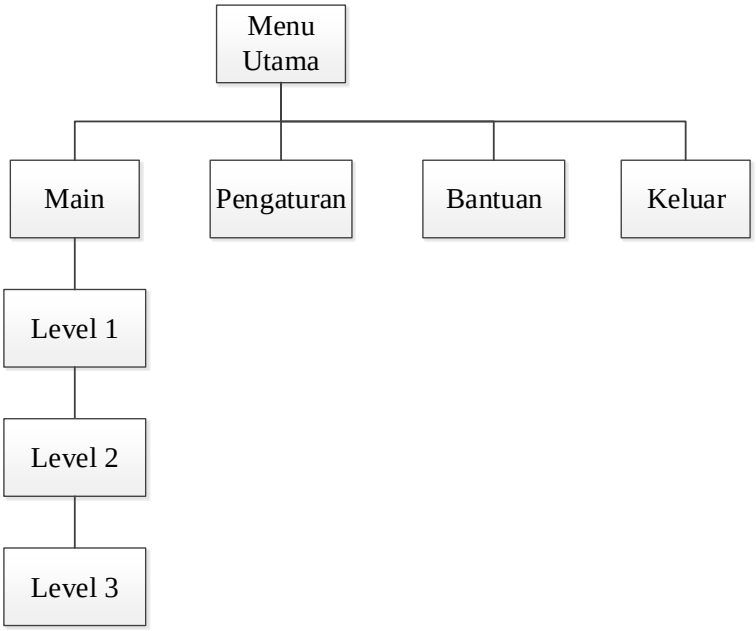


Gambar 3.49 Pertanyaan *Level* 3 Dari Tokoh Pembantu 2

Pemain harus menjawab pertanyaan dari Mr. Sun untuk dapat menyelesaikan adventure. Bentuk pertanyaan pilihan ganda.

3.2.2 Perancangan Struktur *Menu*

Perancangan struktur *menu game go green* bersama petualangan si ulat adalah sebagai berikut :



Gambar 3.50 Perancangan Struktur *Menu*

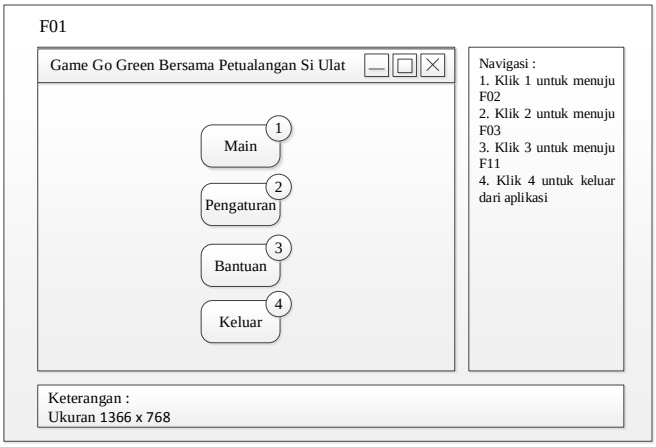
1. Main : *sub menu* untuk memulai *game go green* bersama petualangan si ulat.
2. Pengaturan : *sub menu* untuk melihat mengatur suara pada permainan.

3. Bantuan : *sub menu* untuk melihat cara memainkan permainan pada masing-masing *level*.
4. Keluar : *sub menu* untuk keluar dari aplikasi *game*.
5. Permainan *level* 1 : merupakan *sub menu* dari *menu* main, untuk menampilkan layar permainan pada *level* 1.
6. Permainan *level* 2 : merupakan *sub menu* dari *menu* main, untuk menampilkan layar permainan pada *level* 2 jika permainan *level* 1 sudah terselesaikan.
7. Permainan *level* 3 : merupakan *sub menu* dari *menu* main, untuk menampilkan layar permainan pada *level* 2 jika permainan *level* 2 sudah terselesaikan.

3.2.3 Perancangan Antarmuka

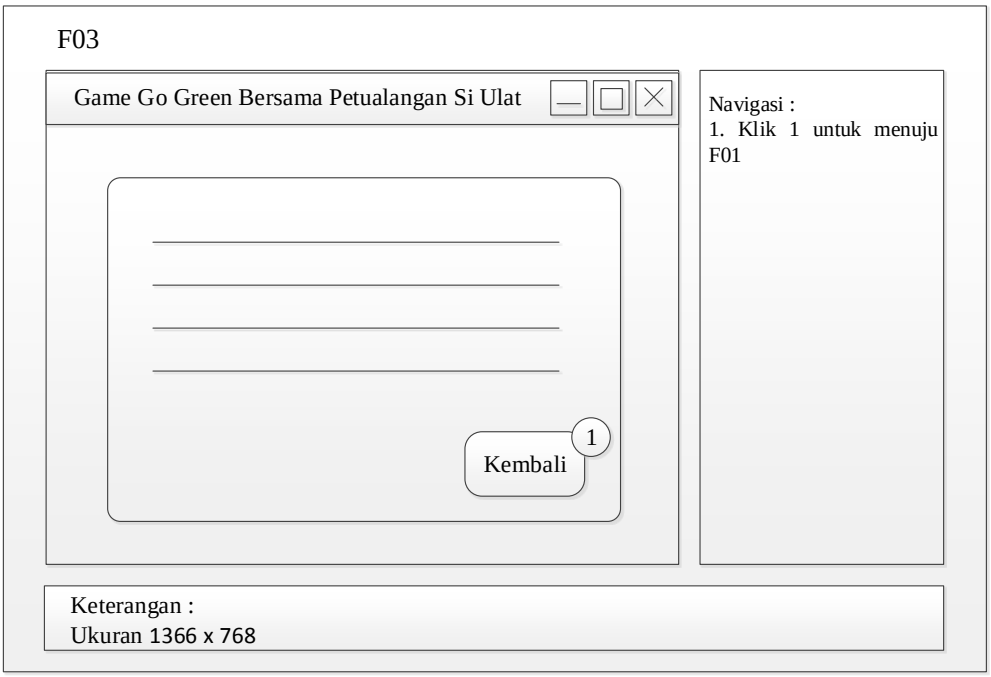
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi. Pada bagian ini akan dijelaskan perancangan antarmuka *menu* utama, perancangan antarmuka pengaturan, perancangan antarmuka bantuan, perancangan antarmuka *level* 1, perancangan antarmuka *level* 2, perancangan antarmuka *level* 3, perancangan antarmuka pertanyaan *level* 1, perancangan antarmuka pertanyaan *level* 2, perancangan antarmuka pertanyaan *level* 3, perancangan antarmuka mengisi form nama, dan perancangan antarmuka *score*.

1. Perancangan Antarmuka *Menu* Utama



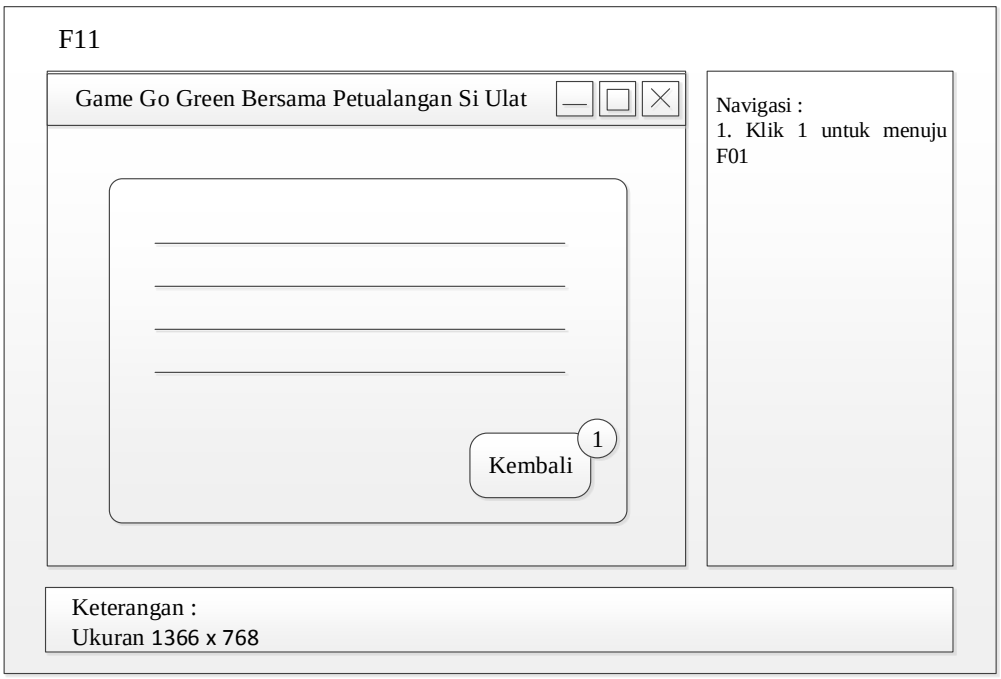
Gambar 3.51 Perancangan Antarmuka *Menu* Utama

2. Perancangan Antarmuka *Menu* Pengaturan



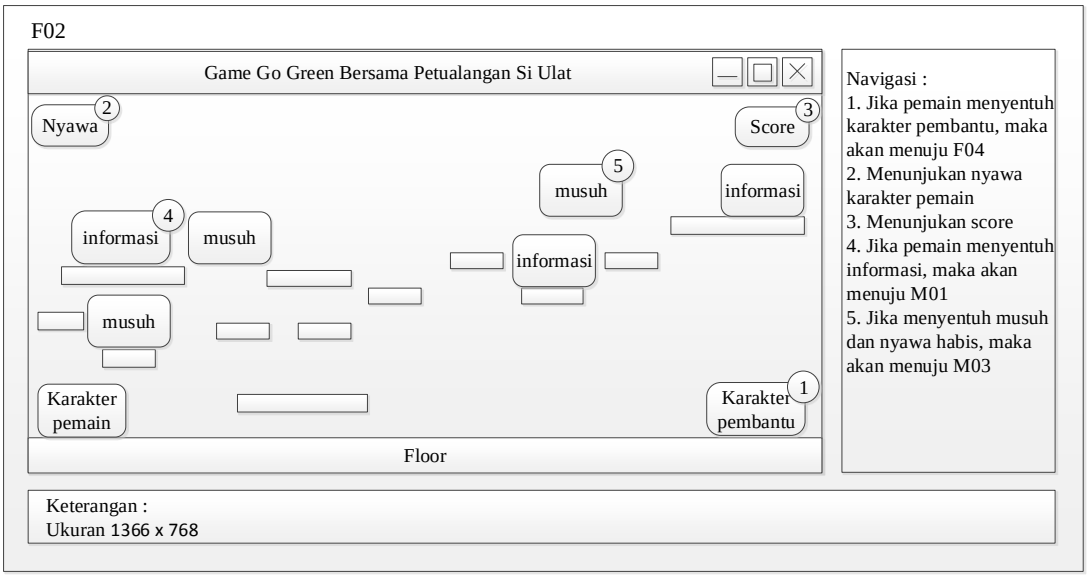
Gambar 3.52 Perancangan Antarmuka *Menu* Pengaturan

3. Perancangan Antarmuka *Menu* Bantuan



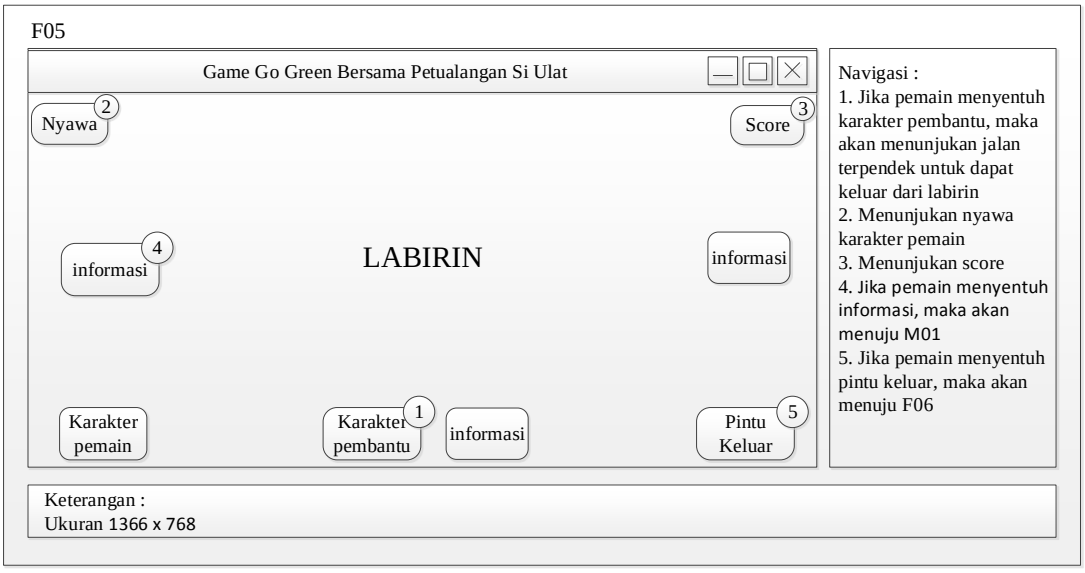
Gambar 3.53 Perancangan Antarmuka *Menu* Bantuan

4. Perancangan Antarmuka *Level* 1



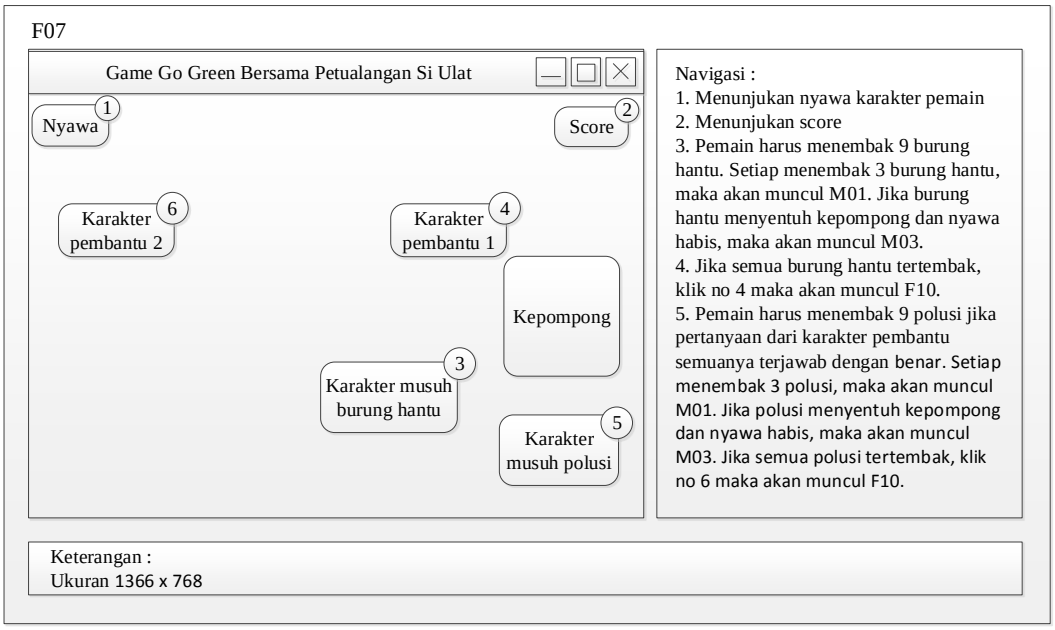
Gambar 3.54 Perancangan Antarmuka *Level* 1

5. Perancangan Antarmuka *Level* 2



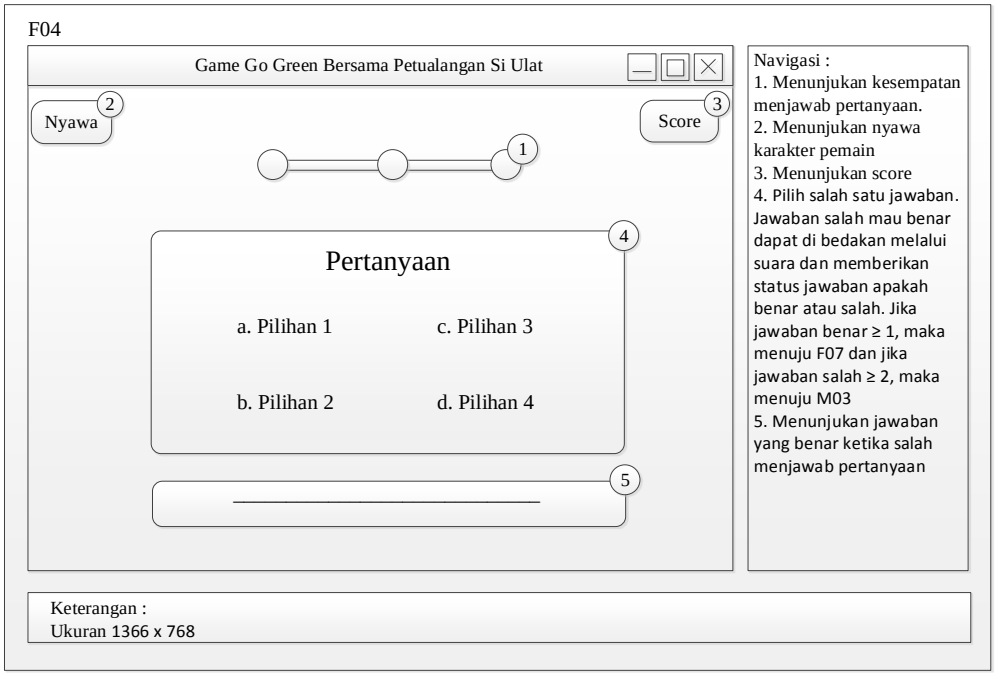
Gambar 3.55 Perancangan Antarmuka *Level* 2

6. Perancangan Antarmuka *Level* 3



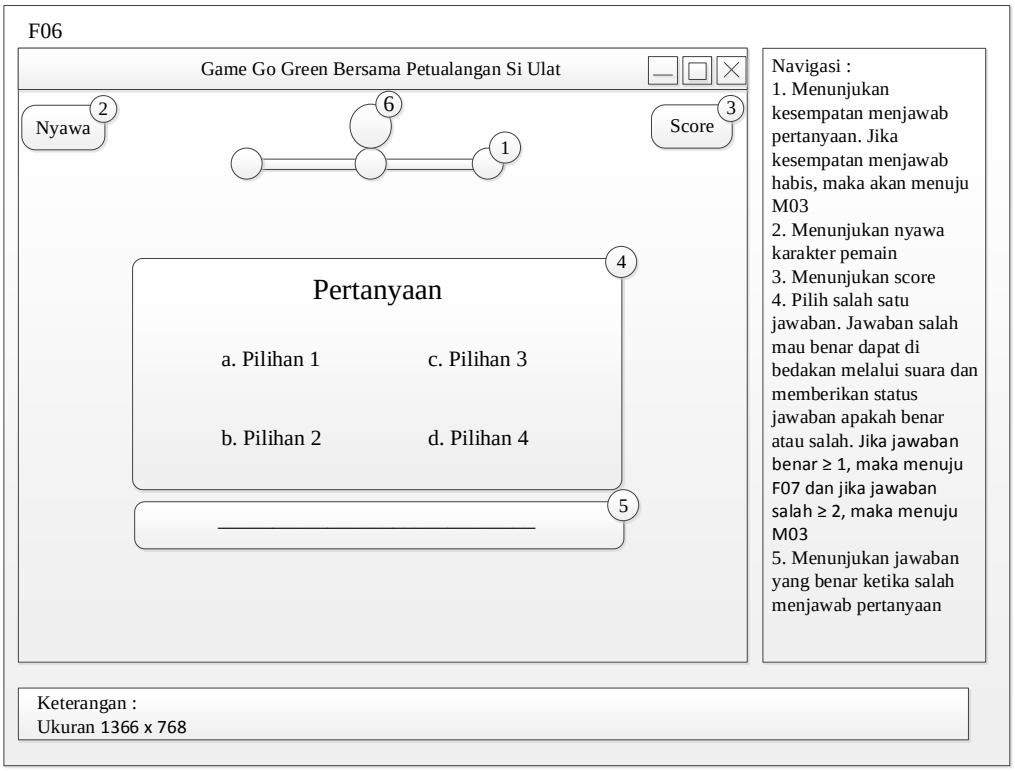
Gambar 3.56 Perancangan Antarmuka *Level* 3

7. Perancangan Antarmuka Pertanyaan *Level* 1



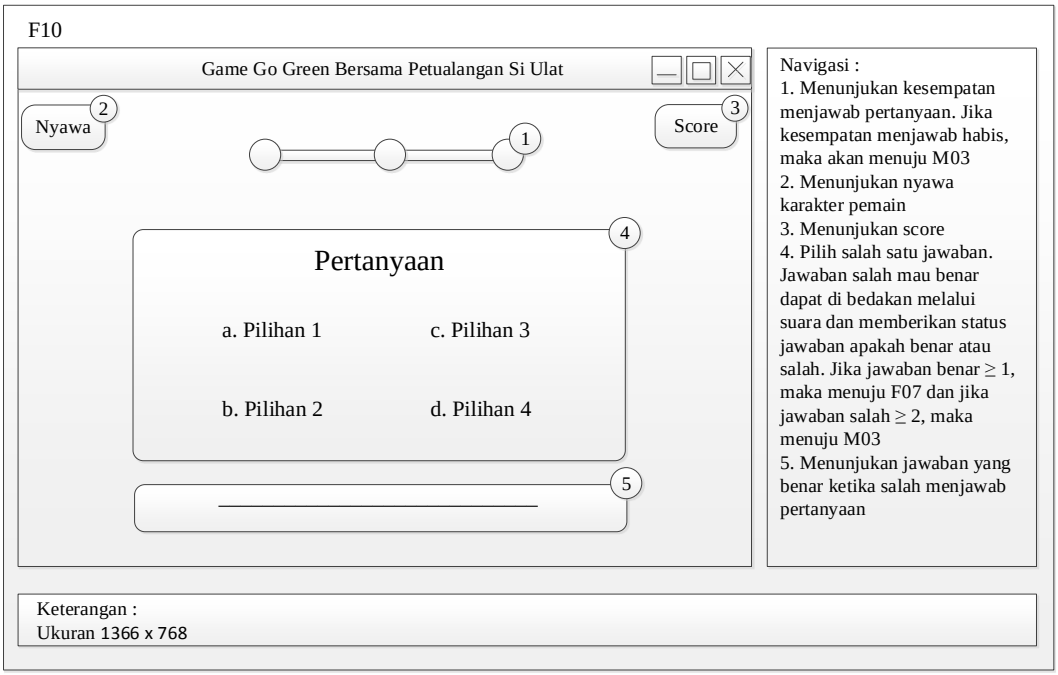
Gambar 3.57 Perancangan Antarmuka Pertanyaan *Level* 1

8. Perancangan Antarmuka Pertanyaan *Level* 2



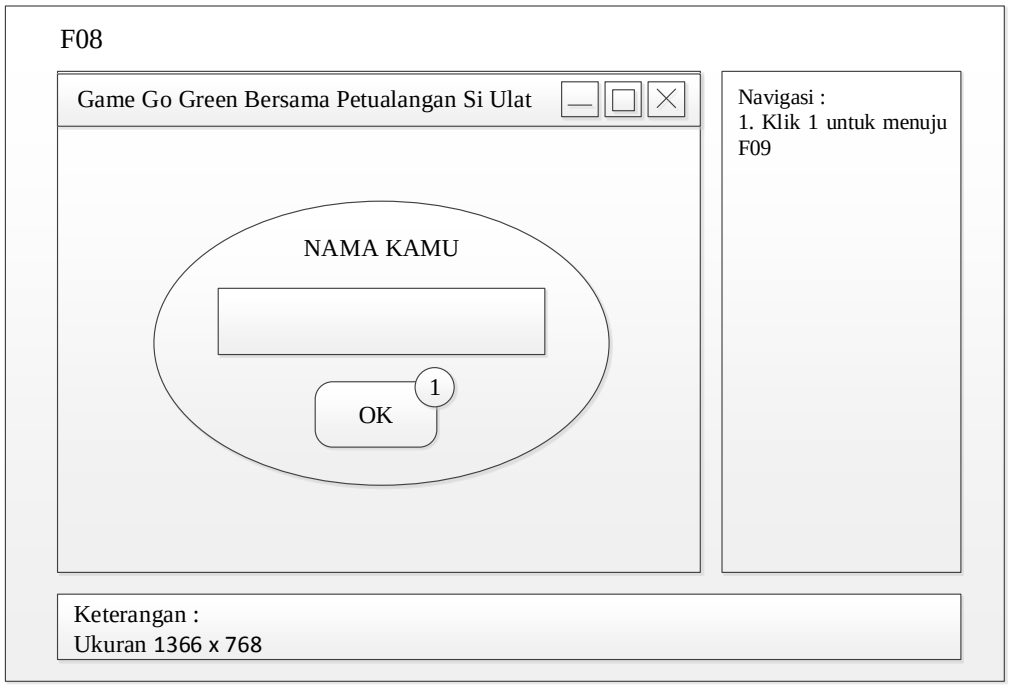
Gambar 3.58 Perancangan Antarmuka Pertanyaan *Level* 2

9. Perancangan Antarmuka Pertanyaan *Level* 3



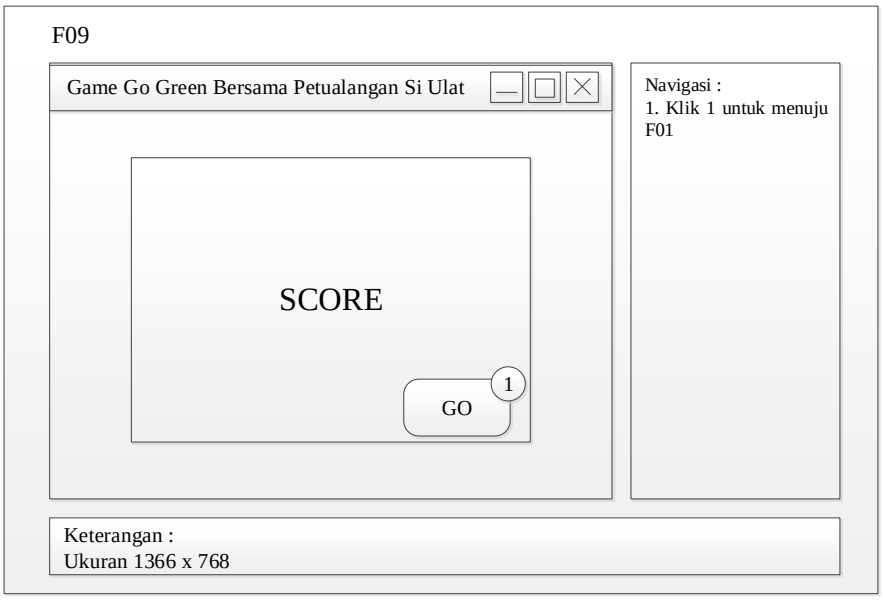
Gambar 3.59 Perancangan Antarmuka Pertanyaan *Level* 3

10. Perancangan Antarmuka Mengisi Form Nama



Gambar 3.60 Perancangan Antarmuka Mengisi Form Nama

11. Perancangan Antarmuka *Score*

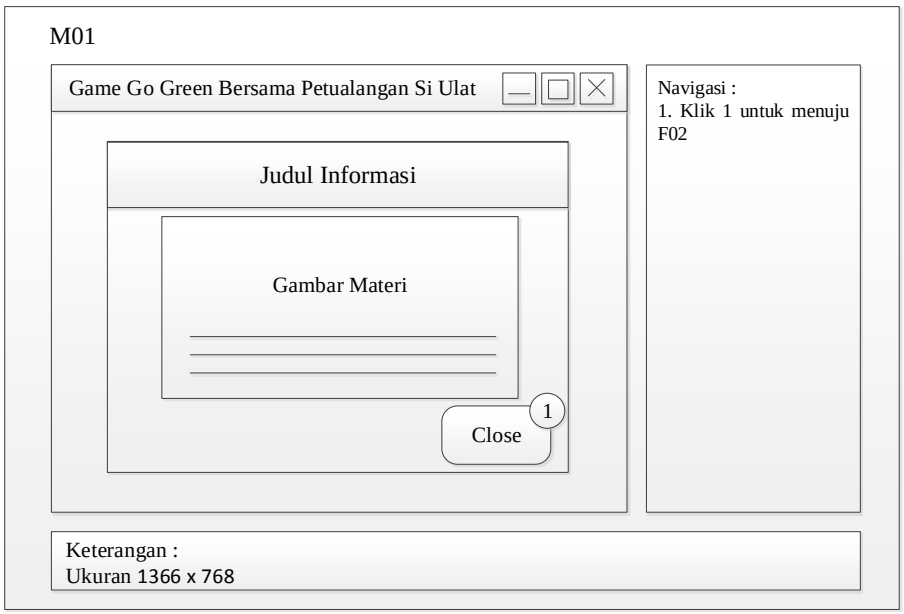


Gambar 3.61 Perancangan Antarmuka *Score*

3.2.4 Perancangan Pesan

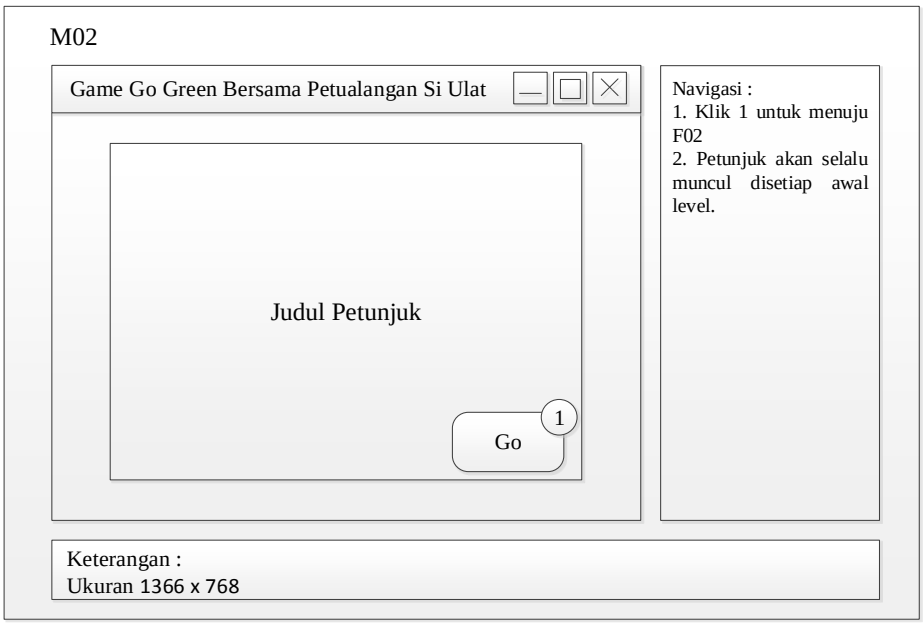
Pada bagian ini akan dijelaskan perancangan pesan yang terdapat pada *game go green* bersama petualangan si ulat yaitu perancangan pesan informasi, perancangan pesan petunjuk, dan perancangan pesan gagal.

1. Perancangan Pesan Informasi



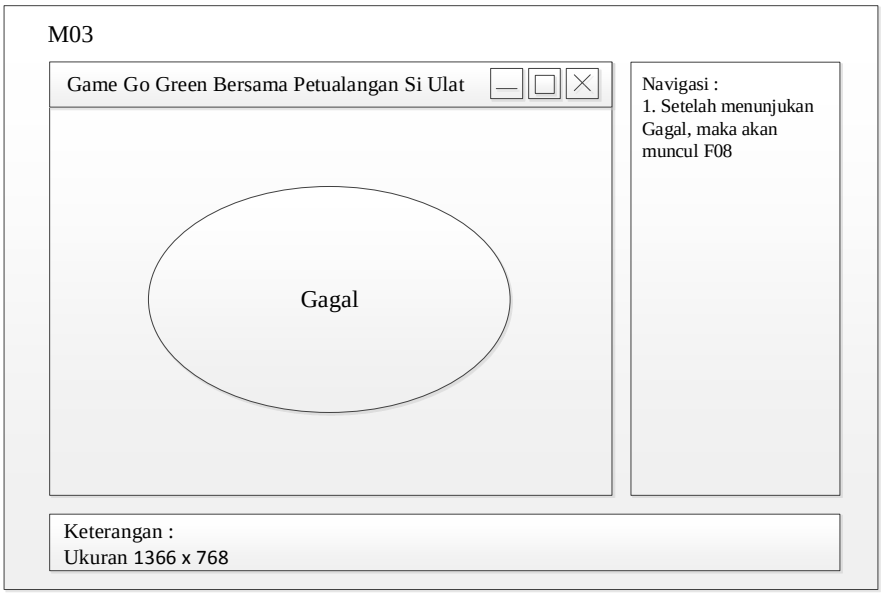
Gambar 3.62 Perancangan Pesan Informasi

2. Perancangan Pesan Petunjuk



Gambar 3.63 Perancangan Pesan Petunjuk

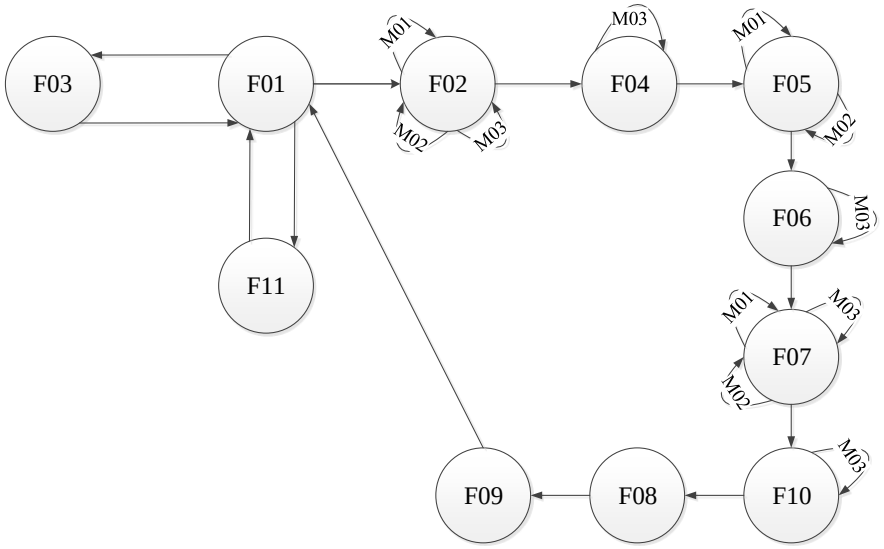
3. Perancangan Pesan *Game Over*



Gambar 3.64 Perancangan Pesan *Game Over*

3.2.5 Jaringan Semantik

Untuk mengetahui keterhubungan dari setiap tampilan yang dibuat dapat menggunakan jaringan semantik. Berikut adalah jaringan semantik pada *game go green* bersama petualangan si ulat.



Gambar 3.65 Jaringan Semantik