BAB III
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1 Analisis Masalah

Berdasarkan penelitian yang dilakukan sebelumya oleh Hary Fernando dari Institut Teknologi Bandung dengan menerapkan algoritma burt force dan greedy [2], Pada persoalan kasusnya algoritma *burt force* pada permainan domino pemain hanya perlu meletakkan kartu domino dimana saja asalkan sesuai, sehingga dapat dikeluarkan tapi tidak peduli urutan atau hal-hal lain. Algoritma *greedy* persimpangan-persimpangan yang ada akan menentukan langkah *greedy* yang akan diambil selanjutnya, penerapan algoritma greedy lebih baik dari *burt force*, tetapi jika kita telah mengeluarkan kartu dengan nilai terbesar yang kita miliki, hal tersebut belum menjamin bahwa kartu-kartu selanjutnya yang akan dikeluarkan akan mudah / cepat habis, bisa saja natinya pemain membutuhkan kartu-kartu besar sebagai penghubung kartu yang satu dengan kartu yang lain sehingga semua kartu dapat keluar. maka hasilnya mungkin lebih buruk dari algoritma *brute force* yang jelas-jelas pasti menghasilkan solusi permasalahan. Maka algoritma greedy dalam mengambil keputusan kurang optimal sehingga akan menyebabkan kemungkinan kartu yang dikeluarkan lebih buruk.

3.2 Analisis *Game*

Permainan yang akan dibangun berjudul domino. Pada permainan domino setiap pemain dapat mengetahui kartu apa saja yang dipegangnya dan tidak bisa melihat kartu yang dipegang oleh lawan, selain itu kondisi kartu yang sudah dikeluarkan juga dapat dilihat oleh semua pemain sehingga dari kartu yang sudah dikeluarkan bisa menjadi patokan bagi pemain selanjutnya yang mendapat giliran bermain untuk mengeluarkan kartu yang berkaitan dengan kartu yang dikeluarkan sebelumnya.

Algoritma yang digunakan pada permainan ini adalah algoritma *minimax* yang pencariannya menggunakan metode *Divide and Conquer*. Pada permainan domino dengan menggunakan algoritma *minimax* yang pencariannya menggunakan metode *Divide and Conquer* dapat menulusuri semua kemungkinan langkah hingga kedalaman tertentu, dan mampu menganalisa segala kemungkinan posisi permainan untuk menghasilkan sebuah keputusan yang terbaik. Semua strategi lawan akan dihitung dengan algoritma yang sama dan seterusnya. Untuk setiap langkahnya, NPC *(Non-Player Character)* akan memilih langkah terbaik dalam mengambil keputusan.

3.2.1 *Gameplay*

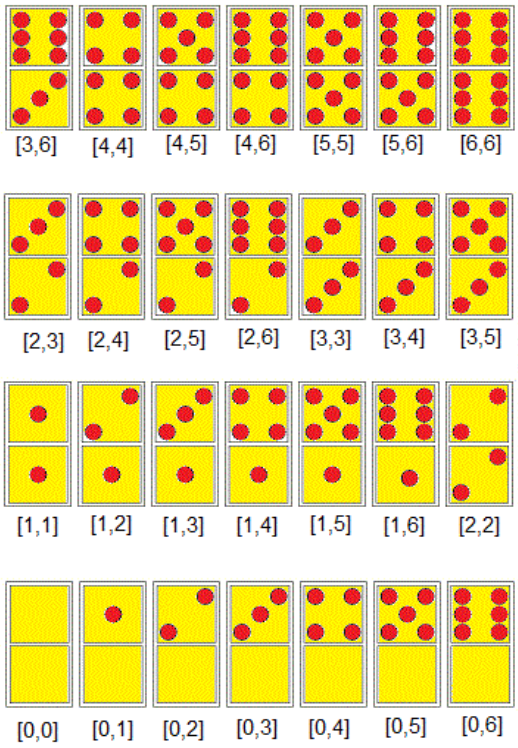
Cara memainkan permainan domino dengan meletakan kartu domino yang bernilai paling kecil terlebih dahulu yaitu kartu kosong, kemudian diikuti oleh pemain lain dengan menyambung kartu domino dengan nilai yang bersesuaian sehingga membentuk suatu pola yang tidak terputus. Penurunan kartu dilakukan terus hingga kartu domino habis atau tidak ada lagi kartu yang dapat diturunkan, Jika yang diturunkan adalah kartu balak, yaitu kartu yang kedua ujungnya bernilai sama, maka akan terbentuk percabangan dari kartu ini dan pemasukan kartu berikutnya dapat dimulai dari bercabangan ini dan bercabangan bertambah. Dalam permainannya, mungkin saja terdapat cabang yang banyak sehingga nantinya pemain harus menentukan pilihan kartu mana yang dibuang, permainan berakhir jika terdapat pemain yang dominonya telah habis atau semua pemain tidak dapat lagi menyambungkan balok-balok itu lagi. Pemain dengan nilai yang paling sedikit adalah pemenangnya [2].

3.3 Analisis Masukan

Pada bagian ini menjelaskan data masukan yang dibutuhkan algoritma yang akan diperoleh dari *game* Domino :

1. Kartu

   kartu domino merupakan kartu yang berukuran kecil berbentuk persegi panjang yang mempunyai jumlah kartu sebanyak 28 kartu. setiap kartu akan memiliki pasangannya masing-masing, seperti pada gambar 3.1.



Gambar 3. 1. Kartu Domino berjumlah 28

   a. tampilKartu merupakan varibel yang dideskripsikan untuk menyimpan kartu domino. tampilKartu = [Banyak Kartu].
   b. acakNumber merupakan varibel yang deskripsikan untuk melakukan pengacakan kartu domino sebanyak 28 kartu, acakNumber = [0..27].

2. Waktu

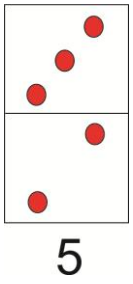
waktu yang digunakan pada *game* ini yaitu waktu ketika melakukan pembagian kartu pada setiap pemain. waktuAwal merupakan variabel yang dideskripsikan untuk memberikan jeda ketika melakukan pembagian kartu domino. kostanta waktuAwal yang diberikan yaitu 56. waktuAwal = 56.

3.4 Analisis Struktur Data

Array adalah sebuah struktur datayang terdiri atas banyak variabel dengan tipe data sama, dimana masing-masing elemen variabel mempunyai nilai indeks, dengan memberi indeks pada variabel untuk membedakan antara yang satu dengan yang lain. Untuk menandakan nilai pada sebuah kartu menggunakan array.

Contoh :

Misalkan 3-2 bernilai 5 seperti pada gambar 3.2.



Gambar 3. 2. Kartu 3-2 yang ada pada meja (*table*)

P = 3

Q = 2

Maka:

Nilai P pada kartu domino dimasukan kedalam indeks ke 0 pada temp array kiri, dan Q dimasukan kedalam indeks ke 0 pada temp array kanan.

TempArrKiri : Array = [3,…,…,…,…,…,…]

TempArrKanan : Array = [2,…,…,…,…,…,…]

Hasil = TempArrKiri [0] + TempArrKanan [0]

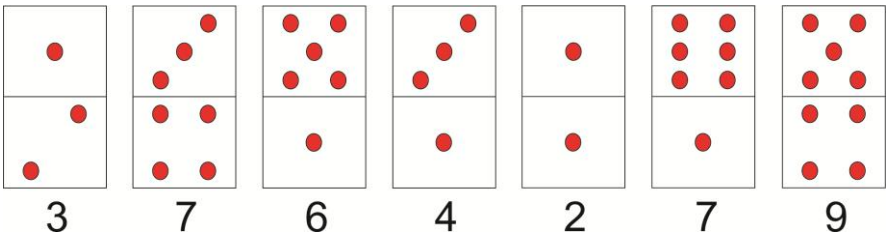
Return . Hasil

3.5 Analisis Algoritma

Analisis algoritma yang dilakukan dalam penelitian ini adalah algoritma *minimax* yang pencariannya menggunakan metode *Divide and Conquer*. Algoritma tersebut akan diterapkan pada NPC *(Non Player Character)* untuk menentukan langkah NPC *(Non Player Character)* dalam mengambil keputusan, untuk setiap langkahnya seluruh kemungkinan akan ditelusuri satu-persatu hingga kedalaman tertentu, dengan tujuan untuk mendapatkan langkah terbaik dalam mengambil keputusan dari semua posisi yang ada.

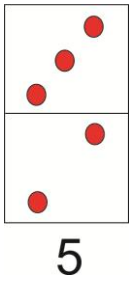
Berikut ini adalah contoh kasus dalam penerapan algrotima *minimax* dengan pecarian metode *Divide and Conquer* :

Misalkan sebuah NPC (*Non Player Character*) memiliki kartu seperti pada gambar 3.3.



Gambar 3. 3. Kartu awal yang dimiliki NPC (*Non Player Character*)

Pada meja domino sudah terdapat kartu 3-2 seperti pada gambar 3.4.



Gambar 3. 4. kartu 3-2 yang sudah ada pada meja (*table*)

Maka kartu berapakah yang harus dikeluarkan oleh NPC (*Non Player Character*) agar mendapatkan hasil yang terbaik ?

Dengan Algoritma *minimax* menggunakan metode *Divide and Conquer*, nilai MIN dan MAX dapat ditentukan dengan melakukan pencarian hingga kedalaman tertentu, dan dapat menganalisa segala kemungkinan posisi permainan untuk menghasilkan sebuah keputusan yang terbaik, nilai MAX akan dimabil dalam keputusan akhir, karena pada permainan domino apabila mengeluarkan kartu yang bernilai paling besar, maka di akhir permainan lebih besar kemungkinan meninggalkan kartu yang bernilai kecil, karena pada permainan domino kartu yang bernilai paling kecil atau ditinggalkan sesedikit mungkin itulah pemenangnya.

Langkah 1 :

Kartu yang dimiliki oleh NPC *(Non Player Character)* dibagi menjadi 2 bagian secara rekursif, supaya lebih mudah dalam menentukan pencarian nilai min dan max.

Jumlah kartu awal yang dimiliki NPC ada 7, maka :

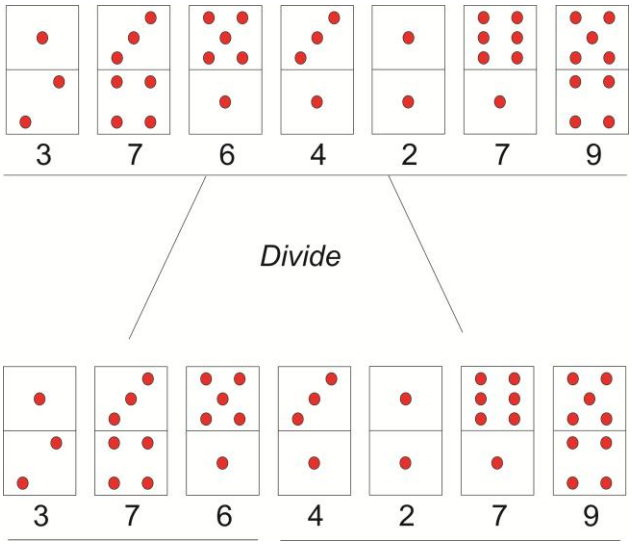
K= (i+j) div 2

K= (1+7) : 2

= 8 : 2

= 4

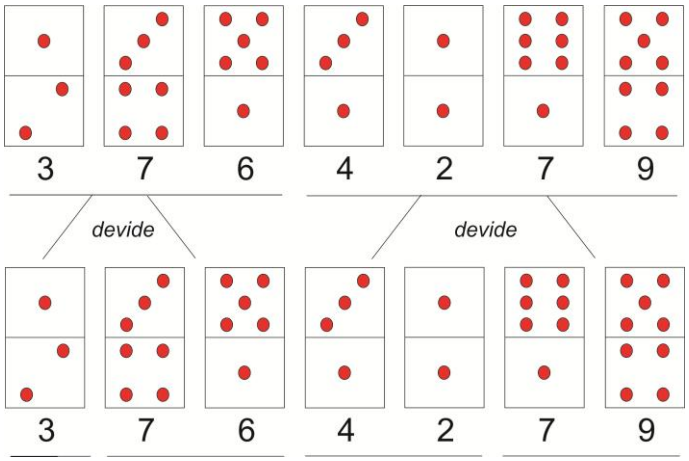
Untuk menentukan nilai MIN dan MAX, maka jumlah kartu dibagi menjadi 2 bagian secara rekursif, maka dari nilai 8 terjadi perpotongan pada kartu ke 4, seperti pada gambar 3.5.



Gambar 3. 5. Kartu yang telah di-*divide* bagian 1

Langkah 2 :

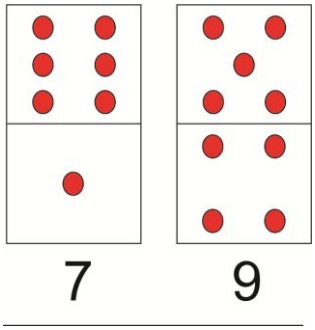
Untuk menentukan nilai MIN dan MAX maka kartu yang sudah dibagi 2 secara rekursif seperti gambar di atas, dibagi lagi menjadi 2 bagian secara rekursif. Agar lebih mudah mencari nilai MIN dan MAX untuk mencocokan dengan kartu yang sudah ada pada meja domino (*table*), dapat dilihat pada gambar 3.6.



Gambar 3. 6. Kartu yang telah di-*devide* bagian 2

Langkah 3 :

Lakukan pencocokan kartu pada gambar 3.7 sampai gambar 3.12 dengan kartu yang ada pada meja domino (*table*) .



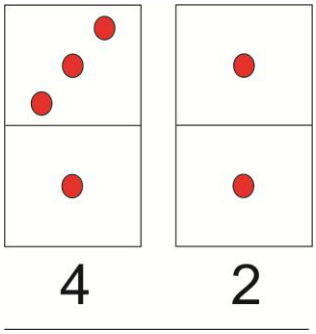
Gambar 3. 7. Kartu 6-1 dan 5-4

Q: Apakah kartu 6-1 (7) cocok dengan kartu yang ada pada meja domino 3-2 ?

A: Tidak cocok, karena 6 maupun 1 tidak ada satupun yang cocok dengan kartu yang ada pada meja (table) yaitu 3-2.

Q: Apakah kartu 5-4 (9) cocok dengan kartu yang ada pada meja domino 3-2 ?

A: Tidak cocok, karena 5 maupun 4 tidak ada satupun yang cocok dengan kartu yang ada pada meja (table) yaitu 3-2.



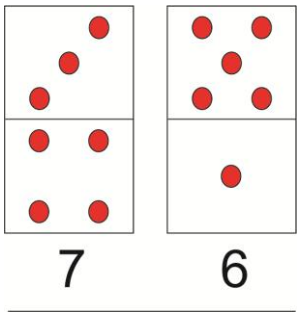
Gambar 3. 8 Kartu 3-1 dan 1-1

Q: Apakah kartu 3-1 (4) cocok dengan kartu yang ada pada meja domino 3-2

A: cocok, karena 3 memiliki kecocokan dengan kartu yang ada pada meja (table) yaitu 3-2, maka 3-1 disimpan pada temp sementara.

Q: Apakah kartu 1-1 (2) cocok dengan kartu yang ada pada meja domino 3-2

A: Tidak cocok, karena 1 maupun 1 tidak ada satupun yang cocok dengan kartu yang ada pada meja (table) yaitu 3-2



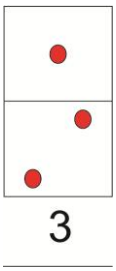
Gambar 3. 9 kartu 3-4 dan 5-1

Q: Apakah kartu 3-4 (7) cocok dengan kartu yang ada pada meja domino 3-2

A: cocok, karena 3 memiliki kecocokan dengan kartu yang ada pada meja (table) yaitu 3-2, maka 3-4 disimpan pada temp sementara.

Q: Apakah kartu 5-1 (6) cocok dengan kartu yang ada pada meja domino 3-2

A: Tidak cocok, karena 5 maupun 1 tidak ada satupun yang cocok dengan kartu yang ada pada meja (table) yaitu 3-2



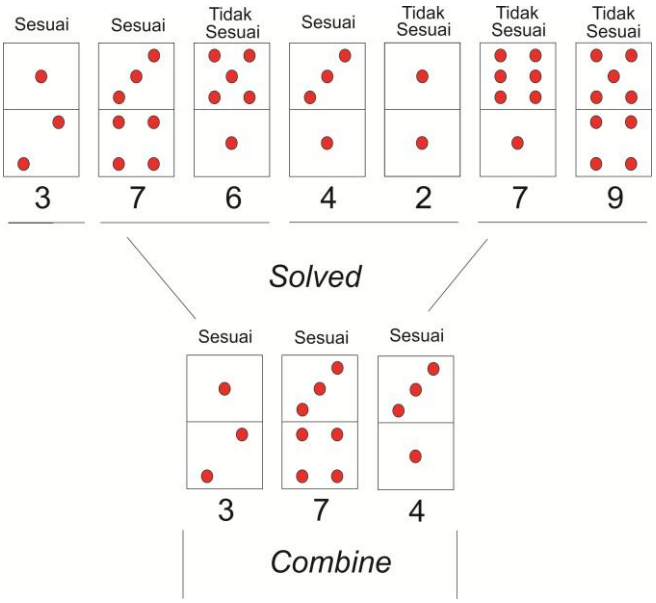
Gambar 3. 10 kartu 1-2

Q: Apakah kartu 1-2 (3) cocok dengan kartu yang ada pada meja domino 3-2

A: cocok, karena 2 memiliki kecocokan dengan kartu yang ada pada meja (table) yaitu 3-2, maka 1-2 disimpan pada temp sementara.

Langkah 4 :

setelah dilakukan pencocokan, maka kartu yang sudah disimpan pada temp digabungkan untuk menentukan nilai minimum dan maksimum, seperti pada gambar berikut 3.11.

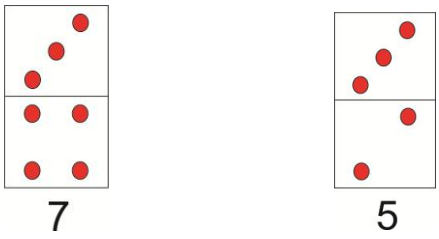


Gambar 3. 11. kartu Solved & Combine

Diantara 1-2 (3) , 3-4 (7) dan 3-1 (4) didapatkan hasil sebagai berikut :

a. 1-2 (3) → kartu yang memiliki nilai paling minimum, karena 1-2 mempunyai nilai 3.
b. 3-4 (7) → kartu yang memiliki nilai paling maksimum, karena 3-4 mempunyai nilai 7.

Maka 3-4 (7) adalah kartu yang akan dikeluarkan agar mendapatkan hasil yang optimum.



Gambar 3. 12. kecocokan kartu 3-4 yang dikeluarkan dan 3-2 yang sudah ada pada meja

3.6 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak digunakan untuk mengetahui spesifikasi perangkat lunak yang digunakan untuk membangun dan memainkan permainan domino, spesifikasi perangkat lunak yang digunakan untuk membangun permainan ini adalah sebagai berikut :

1. Sistem Operasi Windows 7 Ultimate 32 bit
2. Adobe Flash CS 5
3. Adobe Photoshop CS 4
4. Corel Draw X6

Sedangkan perangkat lunak yang dibutuhkan untuk memainkan permainan ini adalah *adobe flash player*, dengan spesifikasi yaitu : Microsoft ® Windows ® XP (32 bit), Windows Server 2008 (32 bit), Windows Vista ® (32 bit), Windows 7 (32 bit dan 64 bit), Windows 8 (32 bit dan 64 bit), atau Windows Server 2012 (64 bit).

3.6.1 Analisis Kebutuhan Perangkat Keras

Analisis perangkat keras digunakan untuk mengetahui spesifikasi perangkat keras yang digunakan untuk membangun dan memainkan permainan domino, perangkat keras yang digunakan dalam analisis ini adalah Asus A43S, untuk detail spesifikasinya adalah sebagai berikut :

1. *Processor* Intel® core ™ i3 CPus @ -2310M (4 CPUs) @2.10Ghz
2. *Memory* (RAM) 4096MB
3. *Display Memory* 1024 MB
4. *Display Mode* 1366 x 768 (32 bit) (60Hz)
5. *Input Device Mouse* dan *Keyboard*

Sedangkan untuk spesifikasi perangkat keras yang dibutuhkan untuk dapat memainkan permainan ini adalah sebagai berikut [10] :

1. Procesor : Pentium 4 > 1Ghz atau setara
2. Memori : 512Mb
3. Hardisk : 20Gb
4. Vga : 64Mb
5. Keyboard : Standar
6. Mouse : Standar

3.6.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi kebutuhan sistem meliputi analisis perangkat keras atau hardware, analisis perangkat lunak atau software, dan analisis pengguna.

3.6.3 Analisis Pengguna

Analisis pengguna digunakan untuk mengetahui spesifikasi pengguna agar dapat memainkan permainan domino. Spesifikasi minimal bagi pengguna agar dapat memainkan permainan domino dapat dilihat pada tabel 3.1 :

Tabel 3. 1. Spesifikasi Pengguna

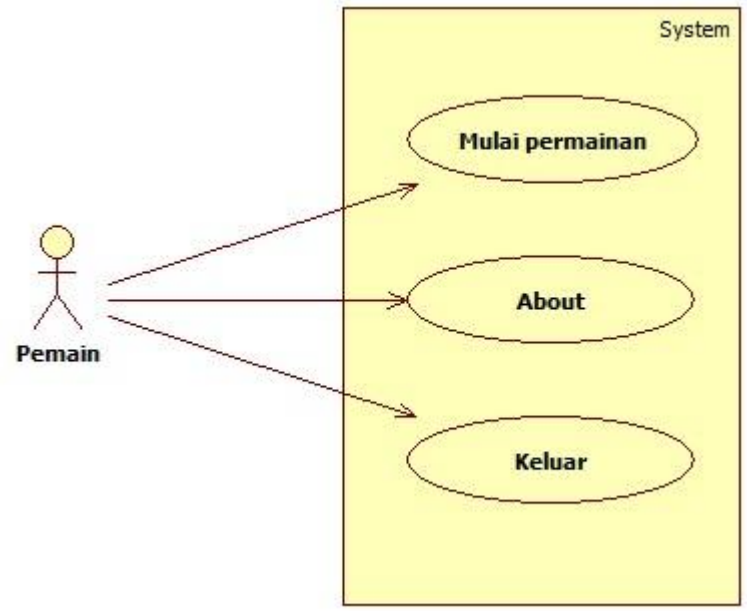
Karakter Fisik Pengguna	
Umur	Tidak ada batasan umur
Gender	Laki-laki dan Perempuan
Disabilities	Tidak buta dan tidak mempunyai cacat fisik seperti cacat tangan
Pengetahuan dan Pengalaman Pengguna	
Computer literacy	Dapat menggunakan dan mengoperasikan Komputer

3.6.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun. Tahap awal yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML (*Unified Modeling Language*). Dimana tahap-tahap perancangan yang dilakukan dalam membangun aplikasi *game* domino antara lain *use case diagram, activity diagram, class diagram,* dan *sequence diagram*.

3.6.4.1 *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Berikut adalah *use case diagram* permainan domino dapat dilihat pada gambar 3.13.



Gambar 3. 13. *Use Case Diagram* Permainan Domino

3.6.4.1.1 Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada *use case diagram*. Deskripsi pendefinisian aktor pada permainan domino adalah :

a. Aktor : Pemain
b. Deskripsi : Orang yang memaikan permainan

3.6.4.1.2 Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada *use case diagram*. Deskripsi pendefinisian *use case* pada permainan domino dapat dilihat pada tabel 3.2.

Tabel 3. 2. Tabel *use case* pada permainan domino

No	*Use case*	Deskripsi
1	Mulai permainan	Merupakan proses untuk memulai permainan
2	Petunjuk *About*	Merupakan proses untuk menunjukan profil pembuat *game*
3	Keluar permainan	Merupakan proses untuk keluar dari permainan

3.6.4.1.3 Skenario *Use Case*

Skenario *use case* menunjukan proses apa saja yang terjadi pada setiap bagian di dalam *use case* tersebut, dimana user memberikan perintah pada setiap bagian *use case* kemudian sistem meresponnya. Skenario *use case* yang terdapat pada pembangunan aplikasi ini antara lain adalah sebagai berikut:

1. *Use Case Screnario* Mulai Permainan

   Use case scenario ini menjelaskan interaksi antara pemain dengan *use case* mulai bermain. Berikut adalah penjelasan dari *use case scenario* mulai permainan yang dapat dilihat pada tabel 3.3.

Tabel 3. 3. *Use Case Scenario* Mulai Permainan

Identifikasi	
Nomor	1
Nama	Mulai Permainan
Tujuan	Memilih Permainan
Deskripsi	Proses untuk memulai permainan

Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama *game* domino
Aksi Aktor	Reaksi Sistem
1. Memilih menu mulai permainan	
	2. Menampilkan arena permainan 3. Kocok kartu 4. Membagikan kartu
5. Memilih kartu	6. Melempar kartu 7. Menyimpan kartu yang dilempar 8. Lap 9. Div 2 10. Mencocokan kartu 11. Menggabungkan kartu yang sudah dicocokan (*combine)* 12. Mencari nilai maksimum 13. Simpan kartu
Kondisi Akhir	Permainan selesai

2. *Use Case Scenario About*

   Use case scenario ini menunjukan profil pembuat game. Berikut adalah penjelasan dari *use case scenario about* yang dapat dilihat pada tabel 3.4.

Tabel 3. 4. *Use Case Scenario About*

Identifikasi	
Nomor	2
Nama	*About*
Tujuan	Menampilkan *About*
Deskripsi	Proses untuk menunjukan profil pembuat *game*
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama *game* domino
Aksi Aktor	Reaksi Sistem
1.Memilih menu about	
	2.Sistem menampilkan menu *about*
Kondisi Akhir	*About* ditampilkan

3. *Use Case Scenario* Keluar

   Use case scenario ini menjelaskan interaksi antara pemain dengan *use case* keluar. Berikut adalah penjelasan dari *use case scenario* keluar permainan yang dapat dilihat pada tabel 3.5.

Tabel 3. 5. *Use Case Scenario* Keluar

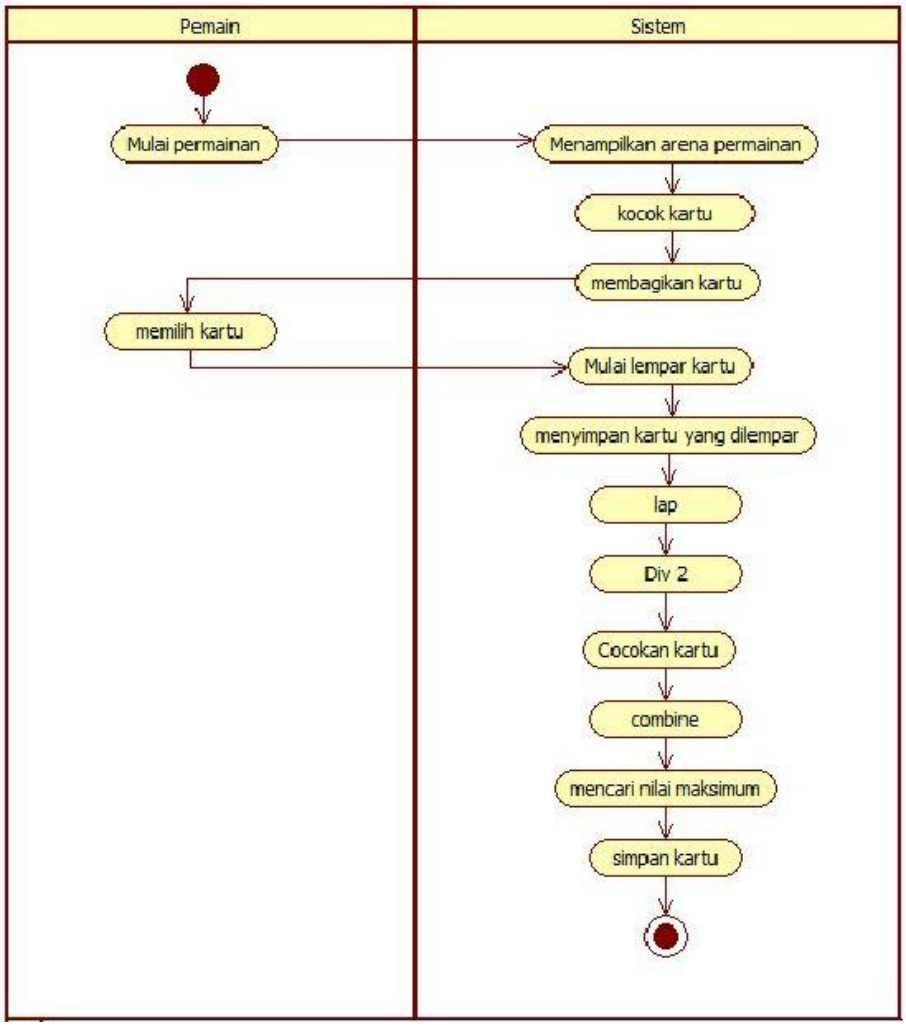
Identifikasi	
Nomor	3
Nama	Keluar
Tujuan	Menutup permainan
Deskripsi	Proses untuk menunjukan profil pembuat *game*
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	Reaksi Sistem
1.Memilih menu keluar	
	2.Menutup aplikasi
Kondisi Akhir	Aplikasi ditutup

3.6.4.2 *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah activity diagram dari permainan domino:

1. *Activity diagram* mulai permainan

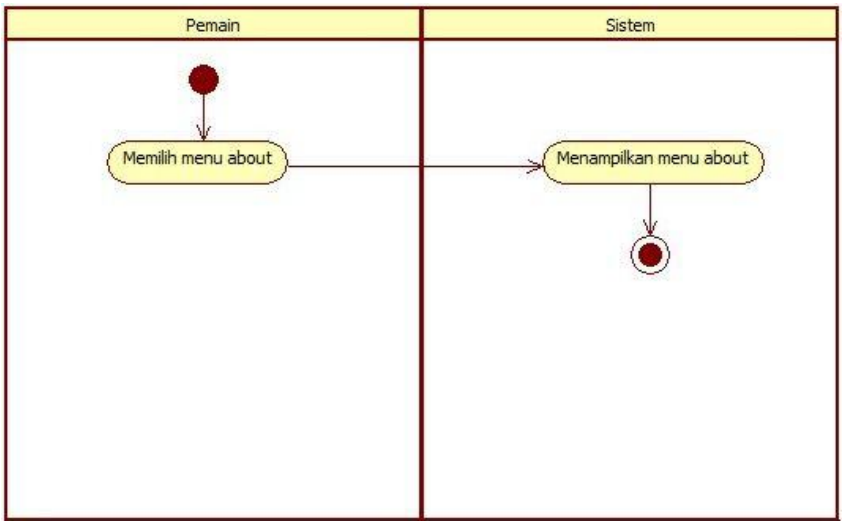
Pada proses ini sistem akan menampilkan arena permainan dimana pemain akan memulai dan menyelesaikan permainan. *Activity diagram* mulai permainan dapat dilihat pada gambar 3.14.



Gambar 3. 14. *Activity diagram* mulai permaina

2. *Activity diagram About*

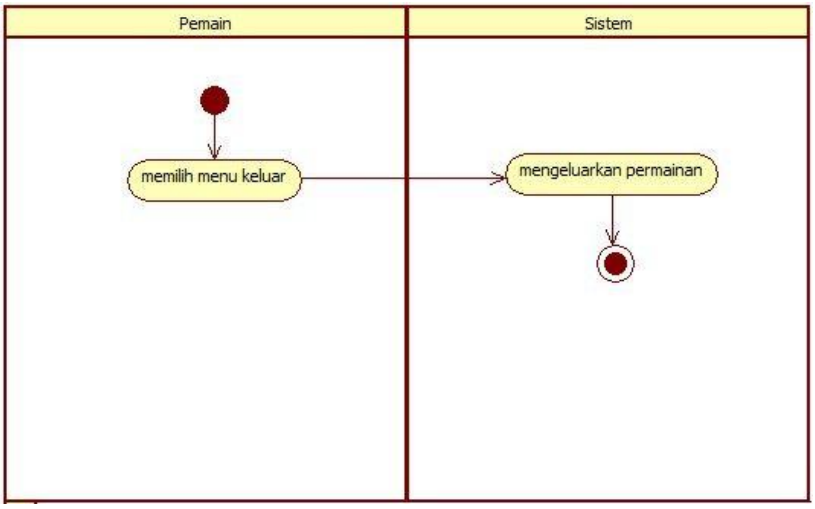
Pada proses ini sistem akan menampilkan profil pembuat *game*. *Activity diagram about* dapat dilihat pada gambar 3.15.



Gambar 3. 15. *Activity diagram About*

3. *Activity diagram* keluar permainan

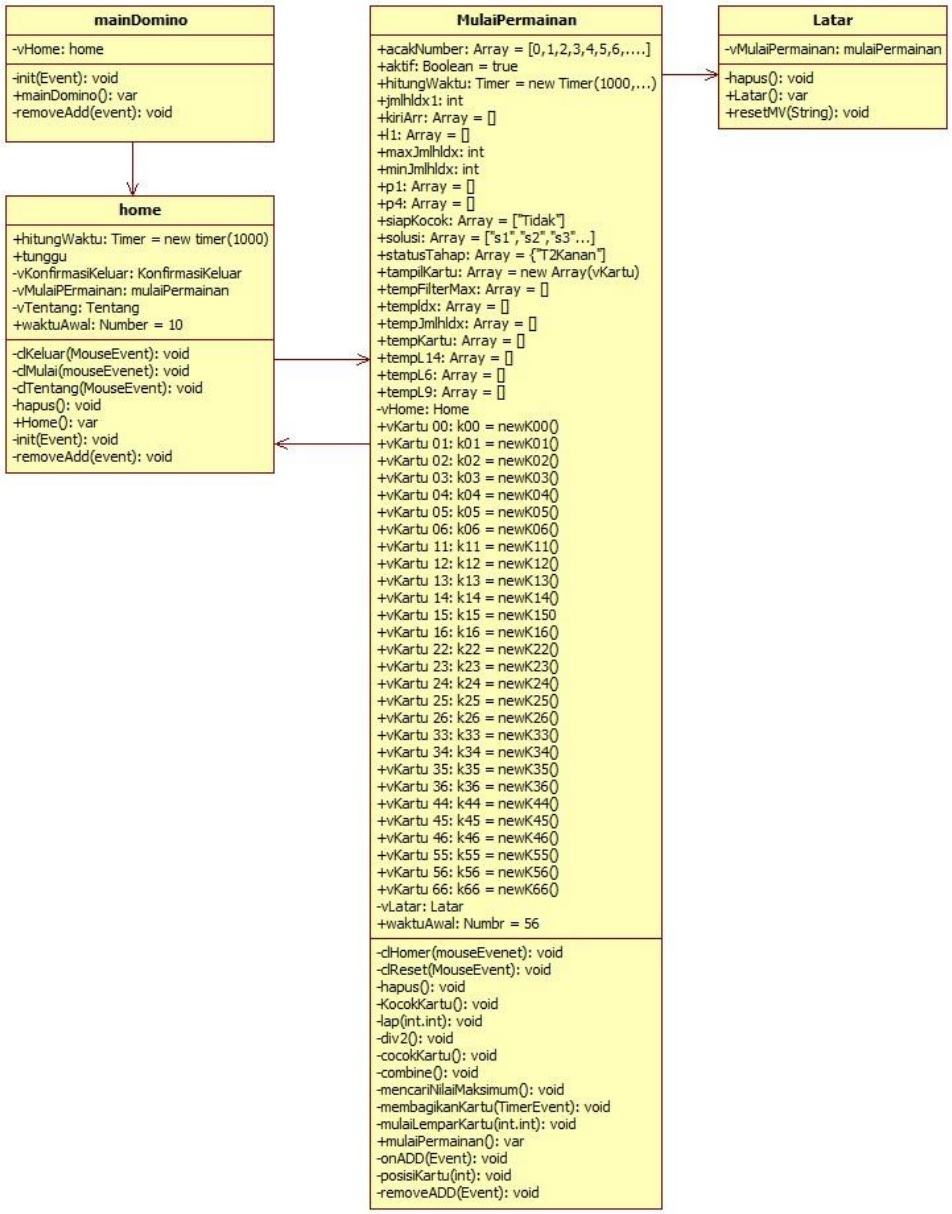
Pada proses ini sistem akan mengeluarkan permainan. *Activity diagram* keluar permainan dapat dilihat pada gambar 3.16.



Gambar 3. 16. *Activity diagram* keluar permainan

3.6.4.3 *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut adalah *class diagram* dari permainan domino:

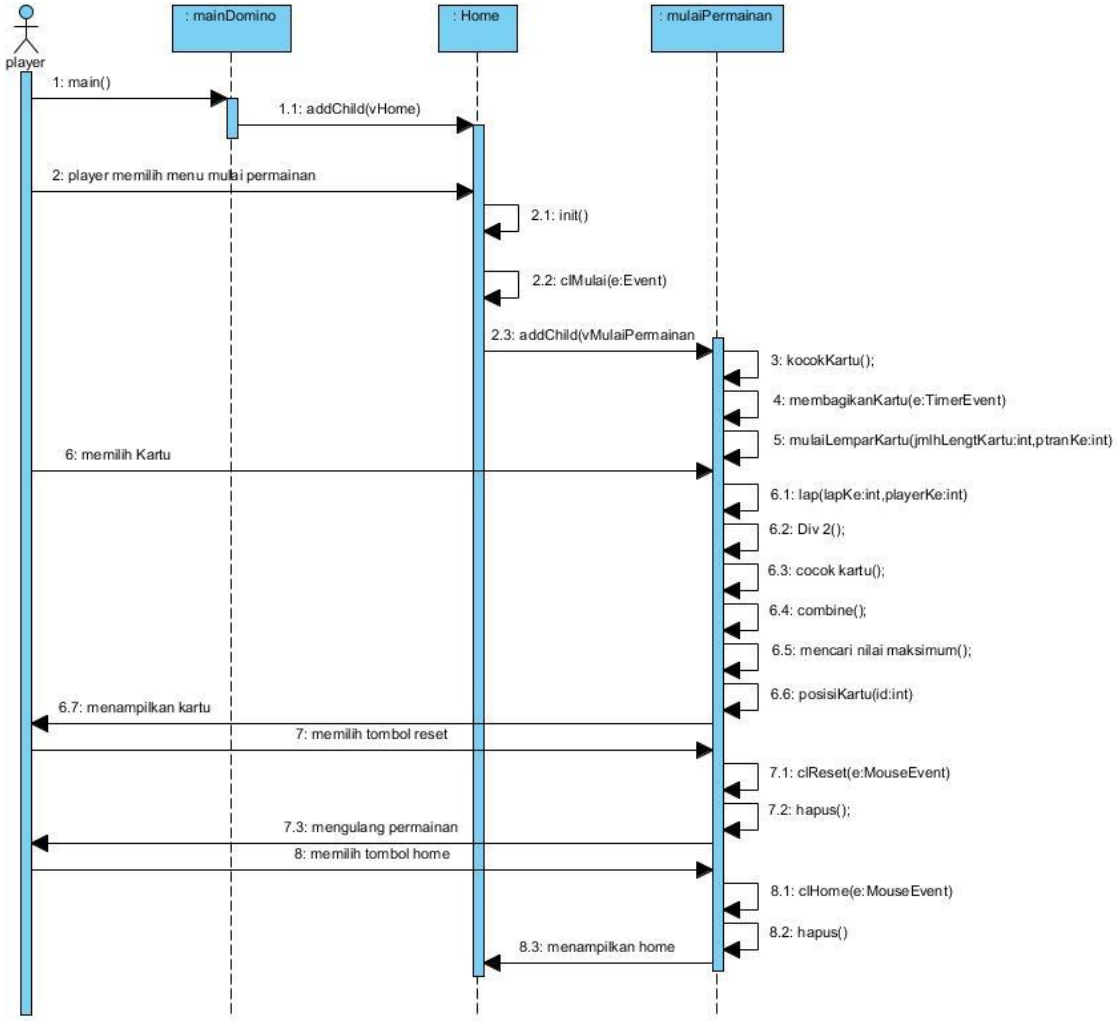


Gambar 3. 17. Class Diagram

3.6.4.4 *Sequence* Diagram

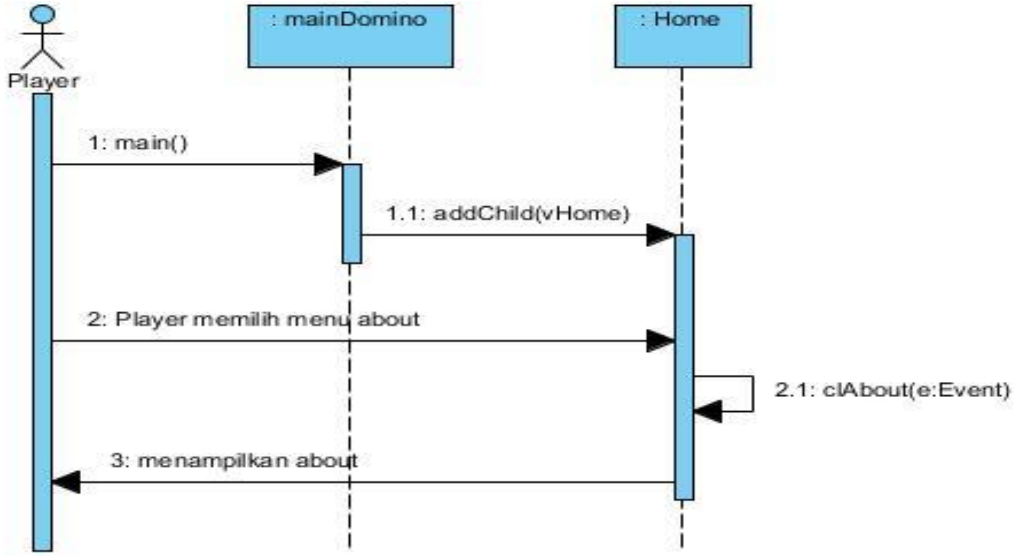
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah *sequence diagram* dari permainan domino:

1. *Sequence diagram* mulai permainan



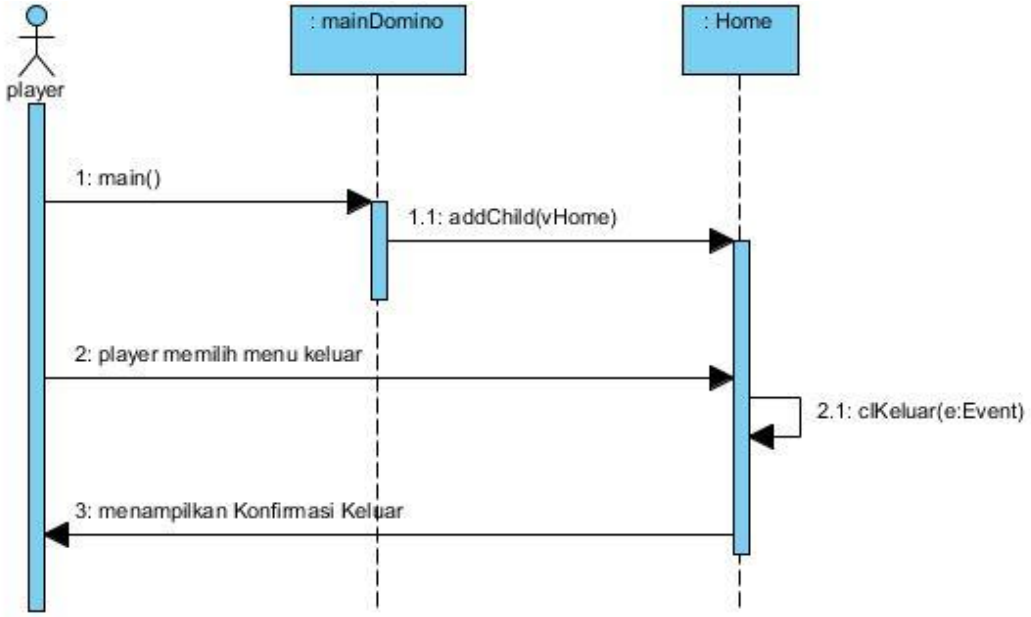
Gambar 3. 18. *Sequence diagram* Memulai permainan

2. *Sequence diagram about* permainan



Gambar 3. 19. *Sequence diagram* Menu *about*

3. *Sequence diagram* keluar permainan



Gambar 3. 20. *Sequence diagram* Keluar permainan

3.6.4.5 Perancangan Sistem

Perancangan sistem merupakan metodologi pengembangan suatu perangkat lunak yang dilakukan setelah melalui tahapan analisis.

3.6.4.5.1 Perancangan Komponen Permainan Domino

Perancangan komponen pada permainan onet terdiri dari pemain, titik awal dan titik tujuan yang akan dijelaskan sebagai berikut :

1. Pemain

   Permainan ini akan dimainkan oleh 4 pemain, satu pemain dan 3 AI.

2. Papan Domino

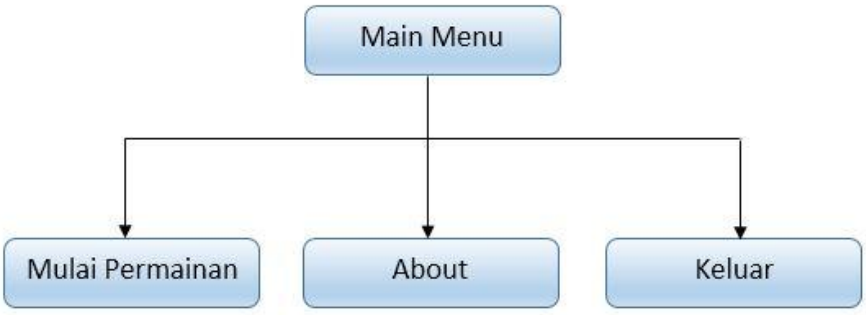
   Papan yang digunakan berupa 1 buah persegi.

3. Kartu Domino

   Kartu domino yang berjumlah 28 kartu, dengan warna biasanya masing-masing hitam dan putih, diletakkan pada perpotongan atau percabangan kartu domino

3.5.4.5.2 Perancangan Struktur Menu

Perancangan struktur menu dirancang untuk menyediakan fungsi-fungsi yang akan digunakan dalam *game* domino seperti pada gambar 3.21.



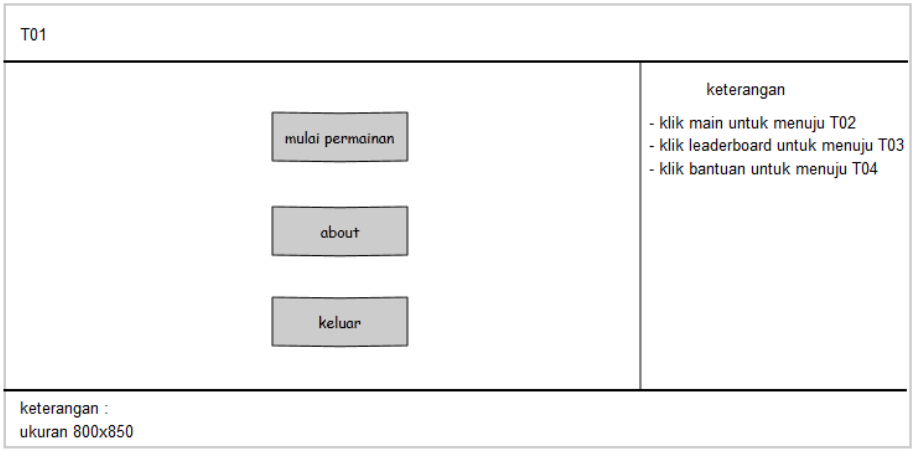
Gambar 3. 21. Struktur menu

3.6.4.5.3 Perancangan Antar Muka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimple-mentasikan aplikasi serta akan memudahkan pembuatan aplikasi:

1. Perancangan Antar Muka Main Menu

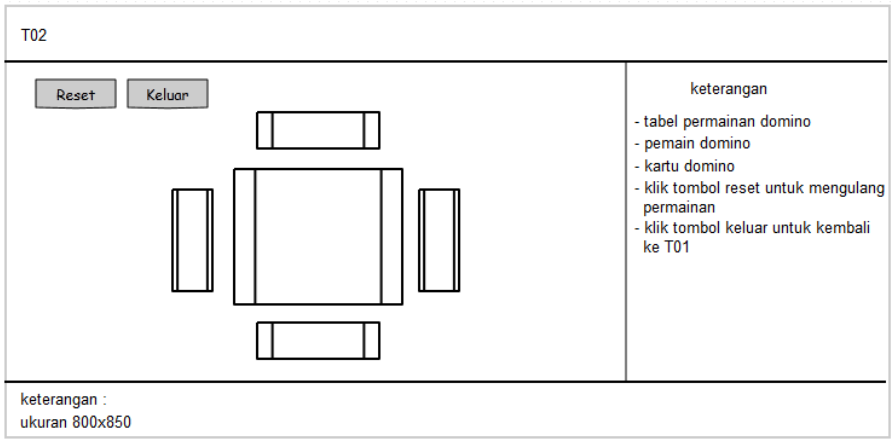
   Tampilan ini merupakan tampilan pertama saat aplikasi permainan domino dijalankan.



Gambar 3. 22 Perancangan antar muka main menu

2. Perancangan Antar Muka Permainan Domino

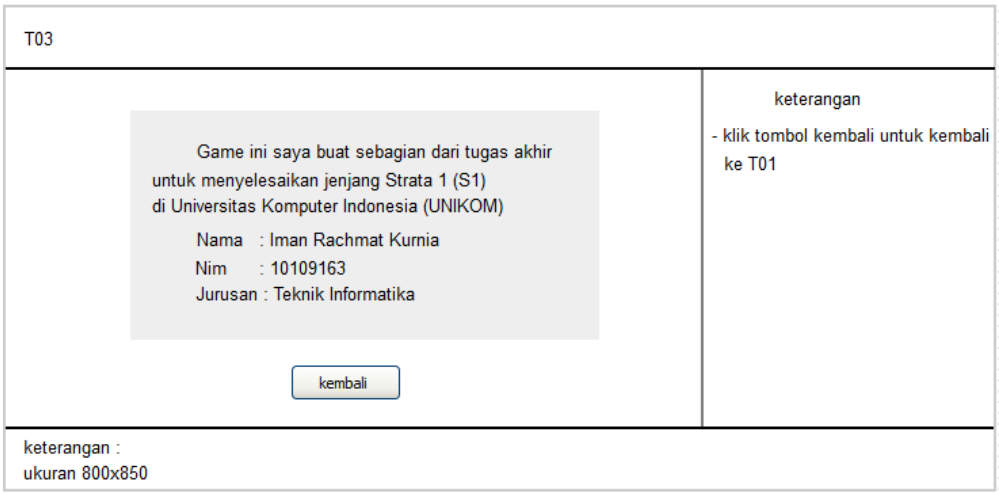
   Tampilan ini merupakan tampilan saat bermain domino.



Gambar 3. 23 Perancangan antar muka permainan

3. Perancangan Antar Muka Menu *About*

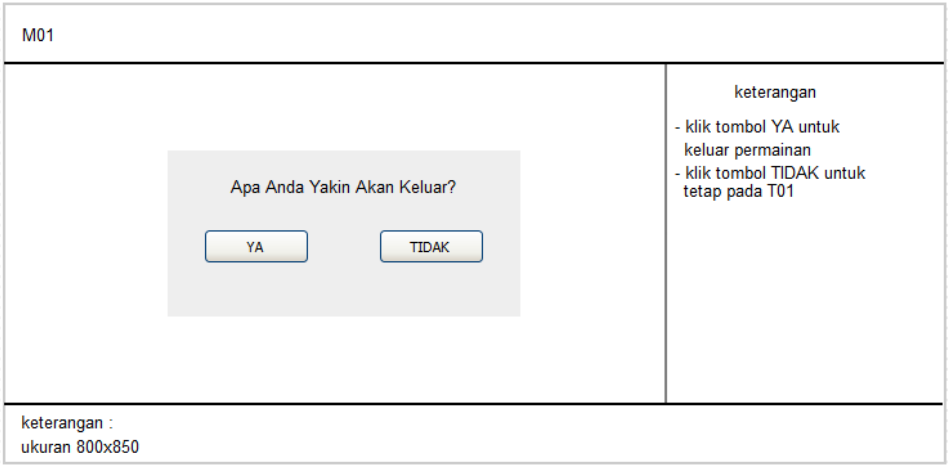
   Tampilan ini merupakan tampilan petunjuk permainan dan biodata pembuat aplikasi.



Gambar 3. 24 Perancangan antar muka menu *about*

4. Perancangan Antar Muka Menu Pesan Keluar

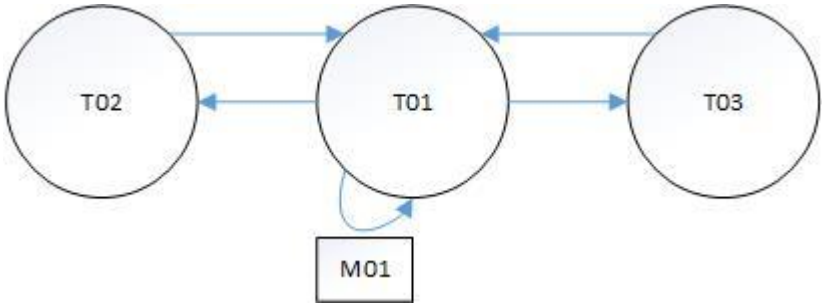
   Perancangan antarmuka ini adalah tampilan pesan yang muncul pada aplikasi seperti ditunjukan pada Gambar.



Gambar 3. 25 Tampilan Pesan

3.6.4.5.4 Jaringan Semantik

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran lingkaran yang menunjukan objek dan informasi tentang objek-objek tertentu. Jaringan semantik untuk *game* domino dapat dilihat pada gambar 3.26:



Gambar 3. 26 Jaringan semantik