BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada analisis sistem akan mencakup analisis masalah, pengenalan dan analisis *game* sejenis, analisis kebutuhan non-fungsional, dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Seperti yang dijelaskan dalam bab sebelumnya, adanya beberapa hal yang melatarbelakangi dibuatnya game ini adalah sebagai berikut :

1. Banyaknya kesalahan mengeja suatu kata.
2. Pembelajaran sendiri dirasakan kurang efisien dan kurang menarik dibandingkan dilakukan secara berkelompok

Berdasarkan permasalahan tersebut, maka akan dibangunnya sebuah aplikasi *word game* berbasis *Mobile* dengan menambahkan bahasa Indonesia dan fitur mode *multiplayer.*

3.1.2 Analisis *Game* Sejenis

Analisis *game* sejenis dilakukan untuk mendapatkan konsep permainan dan perbandingan dengan *game* yang telah ada. Analisis ini dilakukan dengan caramelakukan observasi terhadap*game* yang memiliki kesamaan*gameplay* dan genre dengan *game* yang akan dibangun.

Pada*websiteandroid market* terdapat beberapa*gameandroid* yang memiliki jenis yang sama dengan "*see the word*", *game* tersebut diantaranya, yaitu "*WordWash*" dan "*Word*s with Bees".

3.1.2.1 *GameWord Wash*

GameWordwash merupakan salah satu *wordgame* yang terdapat di *GooglePlay*. *Game* ini menggunakan tema gelembung-gelembung sabun yang di dalamnya terdapat sebuah huruf.Berikut adalah tampilan permainan *Wordwash.*

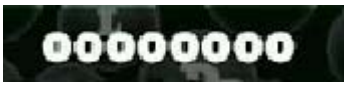
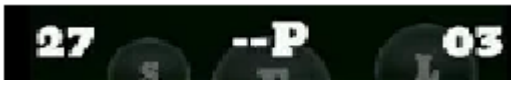


Gambar 3.1 *Game Word Wash*

a. Konten-Konten Pada *GameWord wash*

Berikut adalah konten-konten yang terdapat pada permainan *gameWord wash* dapat dilihat pada tabel 3.1:

Tabel 3.1 Konten-Konten Pada *Game Word Wash*

No	Nama	Gambar	Penjelasan
1	Gelembung huruf		Gelembung huruf merupakan karakter huruf, pemain dapat menyentuhnya untuk menyusunnya menjadi sebuah kata
2	Papan skor		Papan skor merupakan display untuk melihat poin yang telah dicapai pemain
3	Tampilan konten waktu, kata, jumlah kata		Berikut merupakn tampilan konten waktu yang berada di posisi

			sebelah kiri, konten kata merupakan dislpay kata yang dibuat oleh pemain yang berada di tengah dan terakhir jumlah kata merupakan jumlah kata yang harus dibuat oleh pemain yang berada di sebalah kanan.

b. Skenario Permainan

Skenario permainan pada *GameWord wash* dapat disusun oleh beberapa hal, antara lain:

1. Alur Sistem

Sistem akan menginisialisasi seluruh entitas yang diperlukan untuk memulai permainan, seperti karakter pemain yang papan permaian, gelembung huruf, papan skor, tampilan waktu, penampil kata yang telah disusun pemain dan jumlah kata yang harus dibuat. Karakter gelembung huruf akan diacak dan akan terus bermunculan dengan berbagai jenis huruf yang muncul, pada saat semua konten telah muncul maka permainan siap dimainkan.

2. Control Permainan

a) Permainan menggunakan sistem touch screen sebagai alat input untuk mengontrol karakternya.

b) Pemain menekan atau menyentuh gelembung huruf untuk menyusun huruf-huruf agar menjadi sebuah kata.

3. Misi Permainan

Pemain harus membuat kata dengan jumlah yang ditentukan sebelum waktu habis untuk mendapatkan poin yang tinggi dan melanjutkan ke level selanjutnya.

4. ***Gameplay Word wash***

a) Karakter utama adalah gelembung huruf yang harus disusun menjadi sebuah kata.

b) Untuk membentuk kata-kata tersebut, gelembung huruf harus disentuh sesuai dengan susunan huruf dalam sebuah pengejaan kata.

c) Target jumhlah kata yang harus dibuat berbeda beda tiap levelnya.

d) Waktu yang diberikan tiap levelnya akan diakumulasikan jika pada level sebelumnya masih tersedia waktu setelah target jumlah kata dapat dicapai.

e) Apabila pemain tidak dapat membuat sejumlah kata dalam waktu yang telah ditentukan, maka permainan akan langsung berhenti.

f) Pada awal permainan, akan ada tampilan petunjuk cara bermain.

5. **Kekurangan Permainan**

Menurut rating yang diperoleh dari google playstore, game ini masih mendapatkan rating 3.9 dari 5 rating yang disediakan.Selain itu, menurut user yang sudah memainkan game ini salah satu kekurangan game ini adalah kata yang dibuat hanya terdiri dari tiga huruf dan sering terjadi lag pada saat muncul dan bergeraknya gelembung huruf.Berikut adalah tampilan perolehan rating game word wash.



Gambar 3.2 Rating Game word wash

3.1.2.2 *Game* Susun Huruf

*Game*susun huruf merupakan *wordgame* yang terdapat di *googleplay* dengan menggunakan bahasa Indonesia yang didalamnya terdapat huruf-huruf acak.Berikut adalah tampilan permainan susun huruf.



Gambar 3.3 Game Susun Huruf

a. Konten-Konten Pada *Game*Susun Huruf

Berikut adalah konten-konten yang terdapat pada permainan *game*Susun Huruf dapat dilihat pada tabel 3.2:

Tabel 3.2 konten-konten Pada*Game* Susun Huruf

No	Nama	Gambar	Penjelasan
1	Acak ulang		Tombol untuk mengacak ulang huruf yang ditampilkan pada permainan ini.
2	Karakter huruf		Ini merupakan tampilan salah satu huruf yang harus disusun menjadi sebuah kata yang benar

3	tampilan kata yang berhasil disusun		menampilkan kata yang telah berhasil disusun oleh pemain yang berada di pojok kiri atas.
4	waktu		Tampilan batas waktu untuk menyusun huruf-huruf acak sampai menjadi kata yang benar.
5	Tombol centang		Tombol yang ditekan setelah huruf sudah disusun guna untuk mengecek apakah susunan huruf itu sudah benar atau belum.

b. Skenario Permainan

Skenario permainan pada *Game*Susun Huruf dapat disusun oleh beberapa hal, antara lain:

1. Alur Sistem

Sistem akan menginisialisasi seluruh entitas yang diperlukan untuk memulai permainan, seperti karakter huruf acak, tombol centang, tombol refresh dan tampilan waktu Waupun kata yang berhasil disusun yang akan digunakan dalam permainan. Sistem akan mengacak letak dari karakter huruf tersebut. Setelah seluruh entitas yang diperlukan sudah diinisialisasi, maka permainan pun dapat segera dimulai.

2. Kontrol Permainan

a) Permainan ini menggunakan sistem touch screen sebagai alat input untuk mengontrol karakternya.

b) Menyusun huruf-huruf acak yang terdapat dilayar menjadi sebuah kata.

3. Misi Permainan

Menyusun huruf-huruf menjadi sebuah kata yang tepat sebelum waktu habis.

***4. Gameplay* Susun Huruf**

a) Pemain harus menyusun huruf-huruf yang telah diacak enjadi sebuah kata yang tepat.
b) Huruf-huruf yang disusun akan ditampilkan di atas papan permainan.
c) Pemain menekan tombol ceklis untuk memeriksa apakah susunan kata itu tepat atau tidak.
d) Pemain dapat menggunakan tombol acak ulang untuk mendapatkan pengacakan lain dari huruf-huruf tersebut.
e) Apabila pemain berhasil menyusun kata sebelum waktu habis maka pemain dinyatakan menang dan dapat melanjutkan ke permainan selanjutnya.
f) Dan apabila pemain tidak dapat menyusun huruf-huruf tersebut setelah waktu habis maka pemain dinyatakan kalah.

3.1.3 Analisis Pembangunan *Game See the word*

Sebelum membangun aplikasi*game*ini terlebih dahulu dibuat perancangannya. Perancangan yang dibuat merupakan kerangka dari sebuah*game* yang akan dibangun, di dalam perancanganterdapat penjelasan secara rinci mengenai *game* yang akan dibangun.

3.1.3.1 Pengenalan *Game*

Aplikasi*game* yangakan dibangun diberi nama "*see the word*" yang bergenre *cassual* dan *puzzle*. Aplikasi yang dibuat ini memiliki beberapa keunggulan dibandingkandengan aplikasi*game* sejenis yang lainnya. Kelebihan tersebut antara lain :

1. *Game* ini dibuat tidak hanya dengan bahasa Inggris melainkan ditambah dengan bahasa Indonesia.

2. *Game* ini terdapat mode *turn base multiplayer*, sehingga dapat dimainkan oleh 2 *player* dengan beda*device* yang koneksinya menggunakan *Bluetooth*.

3.1.3.2 *Gameplay*

Gameplay yang terdapat dalam *game see the word* ini akan dibedakan sesuai dengan mode permainannya, yaitu mode *single player* dan *multiplayer*.

a. Mode single player

1. Karakter huruf akan disediakan secara acak.
2. Akan disediakan 3 nyawa untuk setiap permainan.
3. Pemain diharuskan membuat suatu kata dengan merangkai huruf-huruf yang disediakan di papan permainan.
4. Dengan menyentuh tiap huruf, huruf tersebut akan berpindah pada bagian atas papan huruf untuk dibuat sebuah kata dengan menyentuh huruf lainnya.
5. Setelah kata sudah terbentuk pemain harus menekan tombol submit untuk memerikasa kata tersebut apakah sesuai aturan permainan dan terdapat dalam kamus.
6. Apabila kata tersebut benar maka huruf-huruf akan kembali diganti dengan huruf yang lain.
7. apabila kata tersebut salah maka nyawa akan berkurang.
8. Tekan tombol clear untuk mengembalikan posisi huruf ke semula.
9. Game akan berakhir jika nyawa telah habis.
10. Pemain akan memenangkan permainan jika semua huruf telah digunakan dalam pembuatan kata.

b. Mode Multiplayer

1. Pemain harus mempunyai lawan main untuk permainan mode multiplayer
2. Sebelum permainan dimulai, kedua pemain diharuskan menghubungkan ponselnya dengan media *Bluetooth*.
3. *Player*1akan berperan sebagai *server*.

4. *Player*2akan berperan sebagai *client*.
5. Kedua pemain akan bermain secara bergantian.
6. Setiap pemain yang berhasil membuat kata dan kata tersebut akan dikirmkan ke lawan mainnya.
7. Huruf yang telah digunakan player 1 akan berubah warna menjadi biru sedang huruf yang digunakan oleh player 2 akan menjadi berwarna merah.
8. Apabila *player* 1 menggunakan huruf yang telah dipakai *player* 2 maka *player* 1 akan bertambah nilai dan nilai *player* 2 akan berkurang, begitupun sebaliknya.
9. *Game*akan berakhir jika terjadi koneksi *Bluetooth* antar perangkat terputus, salah satu pemain menang atau mengalami hasil imbang dimana kedua pemain mempunyai nilai yang sama.
10. Salah satu pemain akan memenangkan permainan jika pemain tersebut lebih banyak dalam memnggunakan huruf untuk membuat kata.
11. Permainan akan berakhir seri jika jumlah point yang dikmpulkan kedua pemain sama.

3.1.3.3 Analisis tingkat kesulitan

Pada game *see the word* terdapat dua mode permainan single player dan multiplayer. Untuk mode permainan akan disediakan dalam beberapa tingkat kesulitan dengan adanya beberapa level yang disediakna dalam permainannya.

Pada tingkatan kesulitan pada mode single player ini akan diberikan level 1 sampai level 5, dimana pemain akan diberikan batas penggunakaan kata-kata untuk pembuatan suatu kata pada tiap levelnya. berikut penjelasan tingkatan tiap levelnya.

a. Level 1 pemain harus membuat kata yang terdiri dari minimal 4 huruf.
b. Level 2 pemain harus membuat kata yang terdiri dari minimal 5 huruf.
c. Level 3 pemain harus membuat kata yang terdiri dari minimal 6 huruf.
d. Level 4 pemain harus membuat kata yang terdiri dari minimal 7 huruf.

e. Level 5 pemain harus membuat kata yang terdiri dari minimal 8 huruf

Pembatasan pembutan huruf yang terdapat pada tiap level itu berdasarkan rata-rata penggunaan jumlah huruf dalam suatu kata yang dijelaskan pada sub bab2.2 kamus bahasa Indonesia.

Pemain akan dapat melanjutkan apabila ke level selanjutnya apabila berhasil membuat 10 kata yang benar dan terdapat dalam kamus.

Dan pemain akan kalah jika 3 nyawa yang disediakan itu telah habis, nyawa tersebut akan berkurang jika pemain melakukan kesalahan dalam pembuatan kata misalnya tidak terdapat dalam kamus, pengejaan yang salah dan mengulang kata yang sudah dipakai sebelumnya.

Untuk mode permainan multiplayer tingkat kesulitan harus mengalahkan lawan main dengan sebanyak-banyaknya menggunakan huruf-huruf acak untuk membuat suatu kata dengan kata yang dibuat terdiri dari 4 huruf sampai kata yang terdiri 8 huruf.

3.1.3.4 Analisis Konfigurasi Tileset dalam Pengacakan Huruf

Salah satu permasalahan yang ditemukan dalam pembuatan tugas akhir ini adalah pengacakan huruf dalam papan permainan, pengacakan dalam sebuah *word game* ini tidak bisa dilakukan dengan menggunakan pengacakan huruf sembarangan karena akan menimbulkan dimana huruf-huruf tersebut tidak dapat dirangkai menjadi sebuah kata. misalnya kendala yang pada pembuatan game ini yaitu dimana game ini menyediakan 5x5 tileset yang berarti terdapat 25 huruf yang disediakan dalam pembuatan suatu kata. Apabila dalam tileset itu dilakukan pemberian huruf acak secara sembarangan maka ditemukannya suatu masalah dimana huruf-huruf tersebut bisa berupa hanya terdiri dari 2 jenis huruf ataupun juga huruf-huruf tersebut berupa huruf konsonan semua yang terutama dalam bahasa Indonesia tidak adanya suatu kata yang dapat dibuat terdiri dari huruf konsonan semua yang mengakibatkan suatu kata tidak dapat dibentuk dapat dilihat pada gambar 3.4.

B	G	W	F	Z
X	C	Q	T	H
S	Y	V	M	A
X	C	M	L	P
O	J	S	G	R

A	A	F	F	A
D	A	A	F	F
F	F	A	D	A
F	A	F	A	F
A	F	F	F	A

Gambar 3.4 Kesalahan Hasil Pengacakan Huruf

Melihat permasalahan diatas maka diperlukan suatu pengacakan huruf yang mengacu pada suatu kata-kata yang terdapat dalam suatu bahasa. untuk itu dalam pembuatan game ini akan menggunakan sistem pengacakan huruf yang telah diterapkan pada game sejenis yaitu pendistribusian pada huruf permainan *scrabble* yang telah disediakan dalam berbagai jenis bahasa, karena setiap bahasa mempunyai sistem pendistribusian atau pengacakan huruf yang berbeda, yang dilihat dari rata-rata frekuensi kemunculan huruf dalam sebuah kata-kata tersebut yang terdapat dalam suatu bahasa.

Dalam eksperimen yang akan dilakukan, akan digunakan beberapa variasi konfigurasi *tileset* yang dirasa cukup representative baik untuk bahasa Inggris dan bahasa Indonesia yang telah dijelaskan pada subbab 2.2.1 dan 2.2.2.

Berikut pseudocode untuk pengacakan/pendistribusian huruf pada tilset dapat dilihat pada gambar 3.5.

Pseudocode pengacakan dan pendistribusian huruf

```
Function Letter[]
generateIndonesiaLetterGrid()
return Letter[]
    grid as Letter[]
    i as integer
begin
    grid <- new Letter[25];
```

```
r <- new Random();
for (int i=0;i<GRID_ITEMS;i++) loop
    int randomNum<-r.nextInt((97 - 1) + 1) + 1;
    if (randomNum >= 1 and randomNum <= 19)then
            grid[i]<-new Letter('A');
    else if (randomNum >= 20 and randomNum <= 28)then
            grid[i]<-new Letter('N');
    else if (randomNum >= 29 and randomNum <= 36)then
            grid[i]<-new Letter('E');
    else if (randomNum >= 37 and randomNum <= 44)then
            grid[i]<-new Letter('I');
    else if (randomNum >= 45 and randomNum <= 49)then
            grid[i]<-new Letter('U');
    else if (randomNum >= 50 and randomNum <= 54)then
            grid[i]<-new Letter('T');
    else if (randomNum >= 55 and randomNum <= 58)then
            grid[i]<-new Letter('R');
    else if (randomNum >= 59 and randomNum <= 61)then
            grid[i]<-new Letter('O');
    else if (randomNum >= 62 and randomNum <= 64)then
            grid[i]<-new Letter('S');
    else if (randomNum >= 65 and randomNum <= 67)then
            grid[i]<-new Letter('K');
    else if (randomNum >= 68 and randomNum <= 70)then
            grid[i]<-new Letter('M');
    else if (randomNum >= 71 and randomNum <= 74)then
            grid[i]<-new Letter('D');
    else if (randomNum >= 75 and randomNum <= 77)then
            grid[i]<-new Letter('G');
    else if (randomNum >= 78 and randomNum <= 80)then
            grid[i]<-new Letter('L');
```

```
        else if (randomNum >= 81 and randomNum <= 82)then
                grid[i]<-new Letter('H');
        else if (randomNum >= 83 and randomNum <= 84)then
                grid[i]<-new Letter('P');
        else if (randomNum >= 85 and randomNum <= 88)then
                grid[i]<-new Letter('B');
        else if (randomNum >= 89and randomNum <= 90)then
                grid[i]<-new Letter('Y');
        else if (randomNum >= 91 and randomNum <= 91)then
                grid[i]<-new Letter('F');
        else if (randomNum >= 92 and randomNum <= 92)then
                grid[i]<-new Letter('V');
        else if (randomNum >= 93 and randomNum <= 95)then
                grid[i]<-new Letter('C');
        else if (randomNum >= 96 and randomNum <= 96)then
                grid[i]<-new Letter('W');
        else if (randomNum >= 97 and randomNum <= 97)then
                grid[i]<-new Letter('J');
        else if (randomNum >= 98 and randomNum <= 98)then
                grid[i]<-new Letter('Z');
        else then
                grid[i]<-new Letter('A');
        end if
    end loop
    return grid;
end
```

Gambar 3.5 Pseudocode pengacakan dan Pendistibusian Huruf

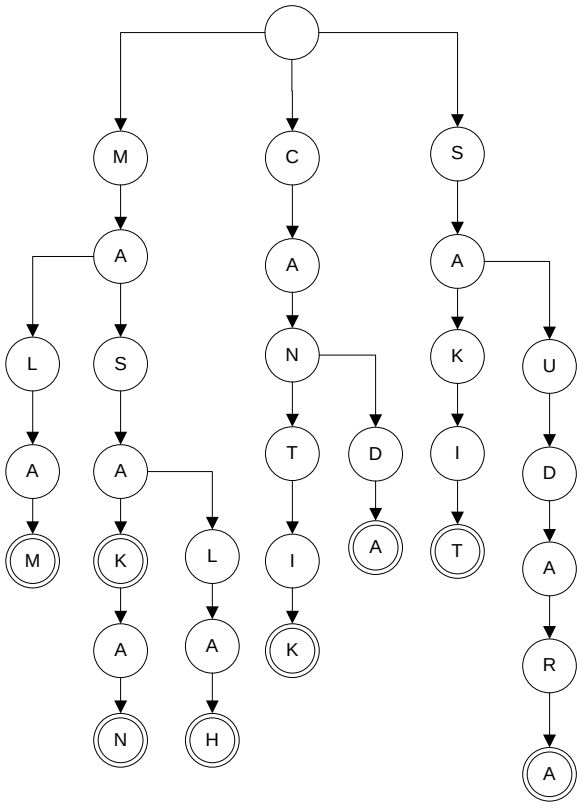
3.1.3.5 Analisis WordTree

Data-data kamus yang dimiliki adalah berupa kata-kata yang memiliki panjang huruf yang beragam sehingga apabila dilakukan pencarian

*squential*menggunakan data seperti ini maka diperlukan waktu yang amat lama sehingga kita perlu mengkonversi data-data tersebut menjadi *graph(node* dan *edge)* yang dapat mempercepat proses pencarian.

Banyak variasi teknik algoritma dan *heuristik* yang digunakan untuk pencarian kita lebih cepat. Elemen utama dari algoritmanya adalah representasi *lexicon*-nya.*lexicon*direpresentasikan sebagai *graph.Graph* berupa *node* sebagai *state* yang menetukan diterima atau tidaknya suatu *input* dan *edge* sebagai penghubung/jembatan antar *node.Node* awal digambarkan dengan *node* yang memiliki *edge* tanpa *node* asal. *Node* yang digambarkan berupa lingkaran dengan dua lapis merupakan *node* akhir yang apabila *input* berhenti di *node* tersebut maka *input* tersebut valid.

Menggambarkan *lexicon*/kamus sebgai *tree* yang *edge*-nya diberi label huruf. Setiapkata pada *lexicon* berhubungan dengan *path* dari *roo*t.saat dua kata berawal dari jalan yang sama dari *path* mereka. *Node* akhir dari *path* mereka disebut *terminal node*, *node* diberi tanda.data struktur seperti ini disebut pohon huruf atau *wordtree.*



Gambar 3.6 Sebuah *lexicon* menggunakan *Tree*

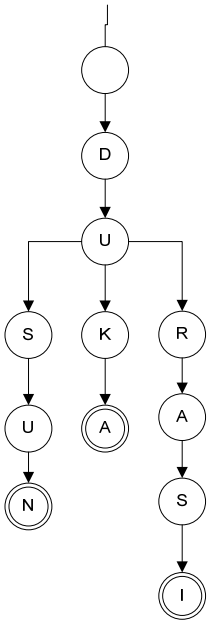
Dapat dilihat pada gambar 3.6 yaitu berupa gambar *wordtree* yang pada permainan ini akan digunakan untuk membuat kamus kata untuk kata-kata yang telah berhasil pemain buat dalam permainan, sehingga pemain tidak dapat menggunakan kata yang sama. penggunanaan*wordtree* ini untuk meminimalkan penyimpanan suatu kata dalam memori.

Permainan akan menampilkan board yang didalamnya terdapat huruf-huruf acak, seperti terlihat pada gambar 3.7. kata-kata yang dibuat pemain menggunakan huruf tersebut akan disimpan dalam wordtree.

N	A	O	P	U
D	E	K	N	O
A	G	E	I	A
R	C	M	L	S
U	S	S	D	R

Gambar 3.7 tileset

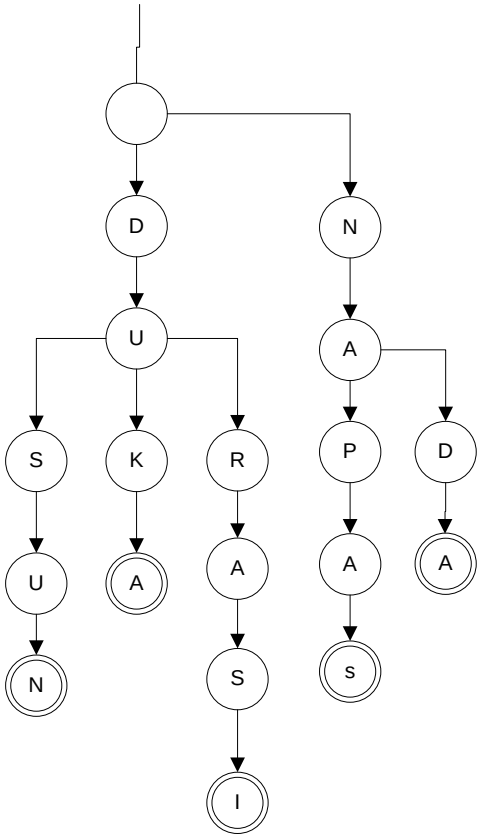
Misal pemain membuat kata DUKA, DUSUN dan DURASI.kata-kata tersebut memiliki karakteristik yang sama pada dua huruf pertama ‘DU’pada kata DUKA, DUSUN dan DURASI,sehingga dapat disatukan. Maka akan terbentuk wordtree seperti pada gambar 3.8



Gambar 3.8 *WordTree* 1

Sedangkan untuk huruf pada 2 kata yang karakteristiknya berbeda walaupun memiliki huruf yang sama namun harus tetap di bedakan *edge*-nya. Hal ini terjadi pada kata’NAPAS’dan ‘DUSUN’, walaupun kedua kata tersebut memiliki hururf 'N' namun 'N' pada ’NAPAS’ bukanlah edge yang menunjukan teminal node, padahal 'N' pada 'DUSUN’ adalah *edge* yang menunjukan *terminal*

node. Oleh karena itu edge seperti ini tetap harus dipisahkan, dapat dilihat pada gambar 3.9.



Gambar 3.9 *WordTree* 2

Berikut pseudocode untuk *build tree* dapat dilihat pada gambar 3.10.

pseudocode *build tree*

function add(word as string) return integer
i as integer
currentNode as Node
c as character
next as Node
begin
currentNode <- mRoot;
if (word == null) then

```
                return INVALID_WORD_ERROR;
        for i <- 0; i < word.length(); i<-i+1 loop
                c <- word.charAt(i);
                next = currentNode.get(c);
                if (next == null) then
                        next <- new Node();
                        currentNode.add(c, next);
                end if
                currentNode <- next;
        end loop;
        mWordIndex <- mWordIndex+1
        currentNode.mIndex <- mWordIndex;
        return currentNode.mIndex;
end
```

Gambar 3.10 Pseudocode *Build Tree*

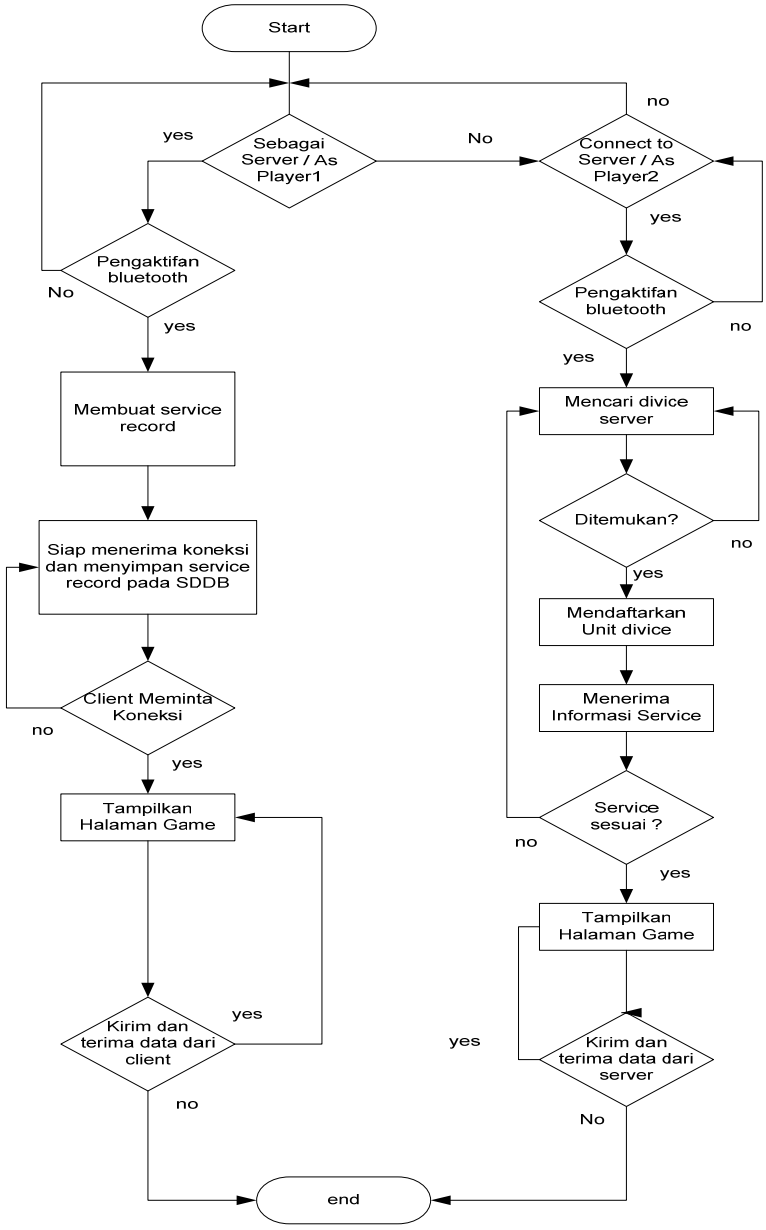
3.1.3.6 Koneksi *Bluetooth* pada mode *multiplayer*

Pada mode *multiplayer* yang tersedia pada *gamesee the word player* akan harus memilih sebagai *player1* atau *player2*. Jika *Player* memilih "*As Player1*" maka program akan menjalankan fungsi sebagai *server* yang membuat *service* untuk *client*. Sedangkan jika *player*memilih "*AsPlayer2*" maka program akan menjalankan fungsi sebagai *client* yang akan membuat koneksi kepada *server*.

Server merupakan suatu tempat yang digunakan untuk membuat sebuah *service* bagi *client* agar dapat melakukan koneksi, menentukan mode koneksi yang akan dipakai *client*, menetapkan *security* dalam jaringan yang dibangun, serta memberi informasi kepada *client* tentang *service* yang ada. *Server* mengawali kerjanya dengan menentukan mode yang akan dipakai yang kemudian dilanjutkan dengan membuat *servicerecord* agar *client* dapat menemukan *server* dan dapat melakukan koneksi. Setelah itu *server* pun akan siap menerima koneksi dari *client* dan menyimpan *servicerecord* ke dalam *SDDB* (*Service discovery database*). Dan

jika ada permintaan koneksi dari *client*, maka server siap untuk mengirim dan menerima data.

Pertama *client* melakukan pencarian unit *device* (*device* discovery), jika *client* tidak menemukan satu pun unit *device*, maka pencarian dapat dilakukan kembali sampai sebuah unit *device* ditemukan.Setelah itu *client* mendaftarkan semua unit yang berhasil ditemukan agar *user* dapat memilih serta membangun koneksi dengan unit tersebut. Kemudian *client* melakukan pencarian *service* dari unit *device* tersebut (*service* discovery) dan *client*akan menerima informasi dari *service* yang ada. Jika *service* telah ditemukan dan sesuai dengan *client*, maka *client* dapat melakukan koneksi pada *service* tersebut dan *client* pun siap untukmengirim dan menerima data.



Gambar 3.11 *Flowchart* Koneksi *Bluetooth*

3.1.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan sistem dilakukan untuk menentukan kebutuhan apa saja yang diperlukan dalam membuat aplikasi *game* yang akan dibuat. Analisis kebutuhan sistem yang diperlukan antara lain:

1. Analisis Kebutuhan Perangkat Lunak
2. Analisis Kebutuhan Perangkat Keras
3. Analisis Pengguna

3.1.4.1 Analisis Kebutuhan Perangkat Lunak

Dalam membangun sebuah *game* diperlukan perangkat-perangkat lunak yang dapat mendukung proses pembuatannya. Adapun perangkat lunak yang dibutuhkan antara lain:

1. Sistem operasi *Microsoft Windows*.
2. *Java SE Development Kit* 7 untuk *Windows* x86.
3. *Eclipse IDE for Java Developer.*
4. *Android SDK.*
5. *AndroidDevelopment Tools (ADT) Plugin for Eclipse*.
6. *Adobe Photoshop CS5.*

Untuk menguji aplikasi *game* "*See the word*" menggunakan *smartphone* dengan perangkat lunak sebagai berikut:

1. Sistem Operasi *Android* OS, v4.1.4 *(Jellybean).*

3.1.4.2 Analisis Kebutuhan Perangkat Keras

Perangkat keras (PC) yang digunakan dalam pembangunan aplikasi *game* "*See the word*" memiliki spesifikasi sebagai berikut :

1. *Processor 2.6 GHz.*
2. *RAM 4 GB 6400.*
3. *VGA AMD ATI Radeon 3300.*
4. *Harddisk 600 GB.*
5. Monitor *LED 14 inch*.
6. Perangkat antarmuka *mouse* dan *keyboard* standar.
7. *Speaker* multimedia standar.

Untuk menguji aplikasi *game* "*See the word*" menggunakan *smartphone* dengan spesifikasi sebagai berikut:

1. *CPU 1.20 MHzProcessor.*
2. *Memori Internal 1.7 GB, Eksternal microSD 8GB.*
3. *RAM* dengan kapasitas 370MB.
4. Layar dengan ukuran 240 x 320 *pixels.*

5. *Bluetooth*

3.1.4.3 Analisis Pengguna

Berdasarkan hasil penelitian, spesifikasi minimal bagi pengguna yang dapat menggunakan aplikasi ini adalah sebagai berikut:

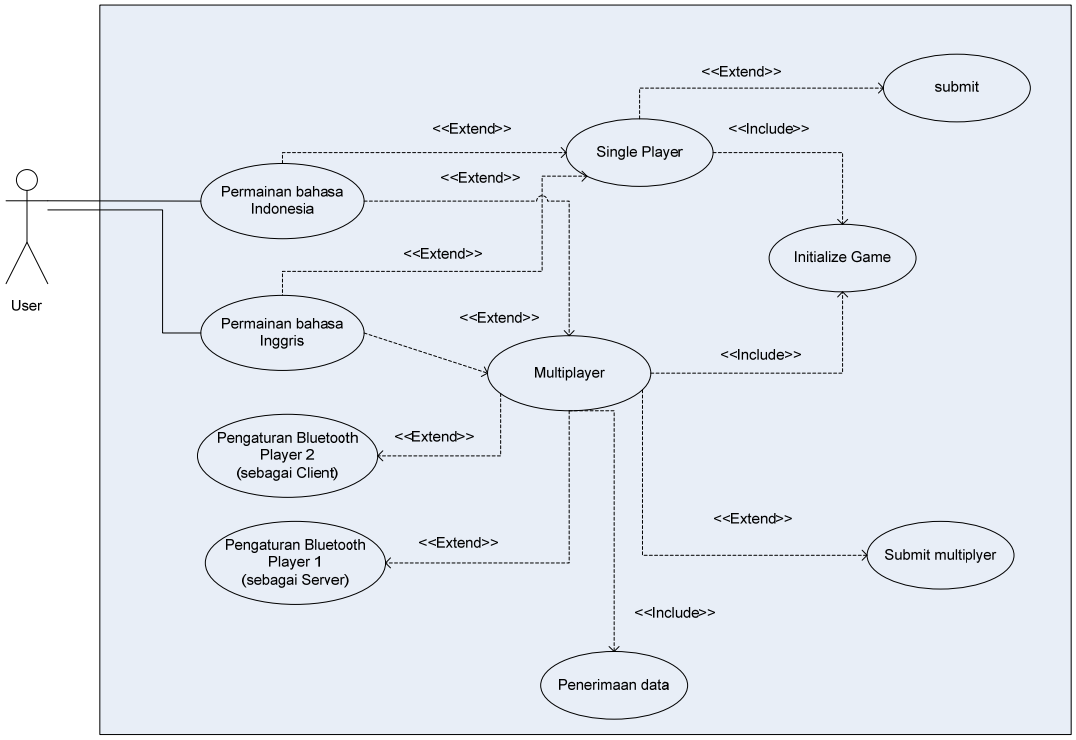
1. Peran : *User*
2. Usia : Lebih dari 7 tahun
3. Tingkat Keterampilan : Dapat mengoperasikan *smartphone*
4. Pengalaman : Pernah memainkan aplikasi *game*

3.1.5 Analisis Kebutuhan Fungsional

Kebutuhan fungsional dari aplikasi *game* "*See the word*" yang akan dibuat terdiri dari penggambaran spesifikasi kebutuhan menggunakan *tool* pemodelan yang berorientasi objek yaitu *Unified Modeling Language(UML).*

3.1.5.1 *Diagram Use Case*

Berdasarkan spesifikasi kebutuhan perangkat lunak yang ada, maka dibuat *use case* yang bertujuan untuk menggambarkan peran dari pengguna dan interaksinya dengan aplikasi serta interaksi dengan sistem lain. *Use case* tidak menunjukan suatu aksi diimplementasikan tetapi hanya menggambarkan fungsionalitas yang diharapkan ada pada perangkat lunak.



Gambar 3.12 Diagram *Use case*

3.1.5.1.1 Definisi Use case

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. *Use case* yang ada pada *game* yang dibangun akan dijelaskan pada table 3.3

Tabel 3.3 Definisi *Use case*

No	*Use case*	Deskripsi
1	Permainan bahasa Indonesia	menggambarkan proses *user* untuk memulai permaianan dengan menggunakan bahasa Indonesia
2	Permainan bahasa Inggris	menggambarkan proses *user* untuk memulai permaianan dengan menggunakan bahasa Inggris
3	*Single player*	menggambarkan aktifitas *user* bermain *game* dengan mode *single player*
4	*Multiplayer*	Menggambarkan aktifitas *user* mengatur perangkat

		untuk memulai bermain *game* dalam *modemultiplayer*
5	*Initialize Game*	Proses penyediaan tampilan *board game*
6	Pengaturan*Bluetooth player* 1(Sebagai *server)*	Proses pengaktipan dan konfigurasi*Bluetooth* yang dilakukan *user* pada saat dalam *mode multiplayer* sebagai*player 1.*
7	Pengaturan*Bluetooth player* 2(Sebagai *client*)	Proses pengaktipan dan konfigurasi*Bluetooth* yang dilakukan *user* pada saat dalam mode *multiplayer* sebagai*player2.*
8	*Submit*	menggambarkan aktifitas sistem dalam memeriksa kata yang dibuat oleh *user* dengan menekan tombol *submit*
9	*Submit* Multi*player*	menggambarkan aktifitas sistem dalam memeriksa kata yang dibuat oleh *user* dan mengirimkan data kepada lawan main dengan menekan tombol *submit*
10	Menerima data	Menggambarkan aktiftas sistem dalam menerima data dari pemain lain pada saat mode permainan *multiplayer*

3.1.5.2 Skenario *Use case*

Bagian ini diisi dengan skenario (*flow of event*) untuk beberapa *use case* utama, yang menggambarkan urutan interaksi aktor dengan *use case* tersebut, dari awal sampai akhir. Skenario *use case* yang terdapat pada *game* ini adalah sebagai berikut.

3.1.5.2.1 Skenario *Use case* Permainan Bahasa Indonesia

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use case*permainan bahasa Indonesia yang akan dijelaskan pada tabel 3.4.

Tabel 3.4 Skenario Use case Permainan Bahasa Indonesia

Identifikasi

Nomor *Use case*	1
Nama	Permainan Bahasa Indonesia
Deskripsi	Menggambarkan proses *user* untuk memulai permaianan dengan menggunakan bahasa Indonesia
Trigger	*External*
Aktor	*User*

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu *'IndWord'*.	
	b. Tampilkan sub menu *'Indword'*, Menampilkan menu mode permainan '*single player* dan '*multiplayer*'.
c. Memilih mode permainan Klik menu "*single palyer*".	
	d. Menampilkan Mode permainan *Single player.*

Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu *'IndWord'*.	
	b. Tampilkan sub menu *'Indword'*, Menampilkan menu mode permainan '*single player* dan '*multiplayer*'.
c. Memilih mode permainan Klik menu "*Multiplayer*".	
	d. Menampilkan Submenu *multiplayer.*

3.1.5.2.2 Skenario *Use case* PermainanBahasa Inggris

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use case*permainan bahasa Inggris yang akan dijelaskan pada tabel 3.5.

Tabel 3.5 Skenario Use case Permainan bahasa Inggris

Identifikasi	
Nomor *Use case*	2
Nama	Permainan Bahasa Inggris
Deskripsi	Menggambarkan proses *user* untuk memulai permaianan dengan menggunakan bahasa Inggris
Trigger	*External*
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu *'EngWord'*.	
	b. Tampilkan sub menu *'EngWord'*, Menampilkan menu mode permainan '*single player* dan '*multiplayer*'.
c. Memilih mode permainan Klik menu "*single palyer*".	
	d. Menampilkan Mode permainan *Single player.*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu 'Eng*Word*'.	
	b. Tampilkan sub menu 'Eng*Word*', Menampilkan menu mode permainan '*single player* dan '*multiplayer*'.

c. Memilih mode permainan Klik menu "*Multiplayer*".	
	d. Menampilkan Submenu *multiplayer*.

3.1.5.2.3 Skenario *Use case Single Player*

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use casesingle player* yang akan dijelaskan pada tabel 3.6.

Tabel 3.6 Skenario *Use case Single player*

Identifikasi	
Nomor *Use case*	3
Nama	*Single player*
Deskripsi	Proses yang dilakukan untuk memulai *game* dengan mode *single player*.
Trigger	*External*
Aktor	*User*
Skenario Normal	
Aksi Aktor	**Reaksi Sistem**
c. Klik menu mode permainan "*single player*".	
	d. Tampilkan arena permainan mode *single player.*
	e. *Initialize* permainan.

3.1.5.2.4 Skenario *Use case Multiplayer*

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use caseMultiplayer* yang akan dijelaskan pada tabel 3.7.

Tabel 3.7 Skenario Use case Multiplayer

Identifikasi	
Nomor *Use case*	4

Nama	*Multiplayer*
Deskripsi	Proses yang dilakukan untuk memulai *game* dengan mode multi *player*
Trigger	*External*
Aktor	*User*
Skenario Normal	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu mode permainan *Multiplayer*.	
	b. Menampilkan sub menu *multiplayer*.
c. Memilih sebagai *player* 1 Klik "*Asplayer* 1".	
	d. Menampilkan pengaturan Bluetooth *player* 1.
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu mode permainan *Multiplayer*.	
	b. Menampilkan sub menu *multiplayer*.
c. Memilih sebagai *player*2, Klik "*As player* 2".	
	d. Menampilkan pengaturan *Bluetoothplayer* 2

3.1.5.2.5 Skenario *Use case Initialize Game*

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use caseInitialize Game* yang akan dijelaskan pada tabel 3.8.

Tabel 3.8 Skenario *Use case Initialize Game*

Identifikasi	
Nomor *Use case*	5
Nama	*Initialize Game*
Deskripsi	Proses penyediaan tampilan board game.
Trigger	*Internal*
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	a. Menampilkan *board* permainan.
b. *User* merangkai huruf membentuk sebuah kata.	
	c. Menampilkan kata yang dibuat *user*.
d. Klik tombol "*submit*".	

3.1.5.2.6 Skenario *Use case* Pengaturan *Bluetooth Player 1*

Skenario use case ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use case*pengaturan *Bluetooth* yang akan dijelaskan pada tabel 3.9.

Tabel 3.9 Skenario Use case Pengaturan *Bluetooth Player* 1

Identifikasi	
Nomor *Use case*	6
Nama	Pengaturan *Bluetooth player 1*
Deskripsi	Proses pengaktipan dan konfigurasi*Bluetooth* yang dilakukan *user* pada saat dalam mode *multiplayer* sebagai *player1*.
Trigger	*External*
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

a. Klik menu "*player 1*" untuk sebagai sebagai *server.*	
	b. Mengaktifkan *Bluetooth.*
	c. Menunggu terhubung dengan *client.*
	d. Menerima permintaan sambungan dari perangkat lain *(Client).*
	e. Berhasil terhubung.
	f. *Initialize* permainan.
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu "*player1"* untuk sebagai sebagai *server.*	
	b. Mengaktifkan *Bluetooth.*
	c. Menunggu terhubung dengan *client.*
	d. Menerima permintaan sambungan dari perangkat lain *(Client).*
	e. Gagal terhubung.
	f. Mengulang kembali ke urutan C sampai berhasil terhubung.

3.1.5.2.7 Skenario *Use case* Pengaturan *Bluetooth Player 2*

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use case*pengaturan *Bluetooth* yang akan dijelaskan pada tabel 3.10.

Tabel 3.10 Skenario Use case Pengaturan *Bluetooth Player* 2

Identifikasi	
Nomor *Use case*	7
Nama	Pengaturan *Bluetooth player* 2
Deskripsi	Proses pengaktipan dan konfigurasi*Bluetooth* yang

	dilakukan *user* pada saat dalam mode *multiplayer*sebagai *player 2.*
Trigger	*External*
Aktor	*User*

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu "*player* 2" untuk sebagai sebagai *client.*	
	b. Mengaktifkan *Bluetooth.*
	c. Mencari perangkat lain.
	d. Menampilkan perangkat yang ditemuka.
e. Memilih perangkat yang ditemukan, klik "nama perangkat".	
	f. Menyambungkan ke perangkat yang dipilih.
	g. Berhasil terhubung.
	h. *Initialize* permainan.

Skenario Aternatif 1	
Aksi Aktor	**Reaksi Sistem**
a. Klik menu "*player* 2" untuk sebagai sebagai *client.*	
	b. Mengaktifkan *Bluetooth.*
	c. Mencari perangkat lain.
	d. Perangkat tidak ditemukan.
e. Klik tombol *"scan"* untuk mengulangi pencarian perangkat lain.	

Skenario Alternatif 2

Aksi Aktor	Reaksi Sistem
a. Klik menu "*player* 2" untuk sebagai sebagai *client.*	
	b. Mengaktifkan *Bluetooth.*
	c. Mencari perangkat lain.
	d. Menampilkan perangkat yang ditemukan.
e. Memilih perangkat yang ditemukan, klik "nama perangkat".	
	f. Menyambungkan ke perangkat yang dipilih.
	g. Gagal terhubung.
h. Kembali memilih perangkat yang berhasil ditemukan, Klik "nama perangkat".	

3.1.5.2.8 Skenario *Use case Submit*

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use casesubmit* yang akan dijelaskan pada tabel 3.11.

Tabel 3.11 Skenario *Use case Submit*

Identifikasi	
Nomor *Use case*	8
Nama	Submit
Deskripsi	menggambarkan aktifitas sistem dalam memeriksa kata yang dibuat oleh *user* dengan menekan tombol *submit*
Trigger	*external*
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

a. Klik tombol *"Submit"*.	
	b. Memeriksa kata yang dibuat oleh *user* terdapat dalam kamus dan sesuai aturan main.
	c. Menyimpan kata kedalam *word tree.*
	d. Menghitung skor.
	e. Menampilkan skor.
	f. Mengembalikan huruf-huruf kedalam *board* dengan memberikan warna.
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
a. Klik tombol *"Submit"*.	
	b. Memeriksa kata yang dibuat oleh *user* tidak terdapat dalam kamus dan tidak sesuai dengan aturan.
	c. Menampilkan pesan kata tidak dapat terdapat dalam kamus.
d. Membuat kata baru	
e. Kembali klik tombol *"submit"*	

3.1.5.2.9 Skenario *Use case Submit multiplayer*

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use casesubmit multiplayer* yang akan dijelaskan pada tabel 3.12.

Tabel 3.12 Skenario *Use case Submit multiplayer*

Identifikasi	
Nomor *Use case*	9
Nama	Submit *multiplayer*

Deskripsi	menggambarkan aktifitas sistem dalam memeriksa kata yang dibuat oleh *user* dan mengirimkan data kepada lawan main dengan menekan tombol *submit*
Trigger	*eksternal*
Aktor	*User*

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
a. Klik tombol *"Submit"*	
	b. Memeriksa kata yang dibuat oleh *user* terdapat dalam kamus atau kata sudah digunakan sebelumnya.
	c. Menyimpan kata ke dalam *word tree*
	d. Menghitung skor
	e. Mengembalikan huruf-huruf ke dalam *board* dengan memberikan warna
	f. Menonaktifkan papan permainan
	g. Mengirimkan data kata yang berhasil dibentuk kepada lawan main

Skenario Alternatif 1	
Aksi Aktor	**Reaksi Sistem**
a. Klik tombol *"Submit"*	
	b. Memeriksa kata yang dibuat oleh *user*,tidak terdapat dalam kamus kata sudah digunakan sebelumnya.
	c. Menampilkan pesan kata tidak terdapat dalam kamus.
d. Klik tombol *"pass"*	

	e. Menonaktifkan papan permainan
Skenario Alternatif 2	
Aksi Aktor	**Reaksi Sistem**
a. Klik tombol *"Submit"*	
	b. Memeriksa kata yang dibuat oleh *user*,tidak terdapat dalam kamus kata sudah digunakan sebelumnya.
	c. Menampilkan pesan kata tidak terdapat dalam kamus.
d. Klik tombol *"ok"*	
e. Membuat kata baru	

3.1.5.2.10 Skenario *Use case*Menerima Data

Skenario *use case* ini menjelaskan interaksi antara aktor, yaitu *user* dengan *use case* menerima data yang akan dijelaskan pada tabel 3.13.

Tabel 3.13 Skenario Use case Menerima Data

Identifikasi	
Nomor *Use case*	10
Nama	Menerima data
Deskripsi	Menggambarkan aktiftas sistem dalam menerima data dari pemain lain pada saat mode permainan *multiplayer*
Trigger	*internal*
Aktor	*User*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	a. Menerima data dari perangkat lawan
	b. Menyimpan dalam *word tree*
	c. Mengaktifkan papan permainan
d. Membuat kata	

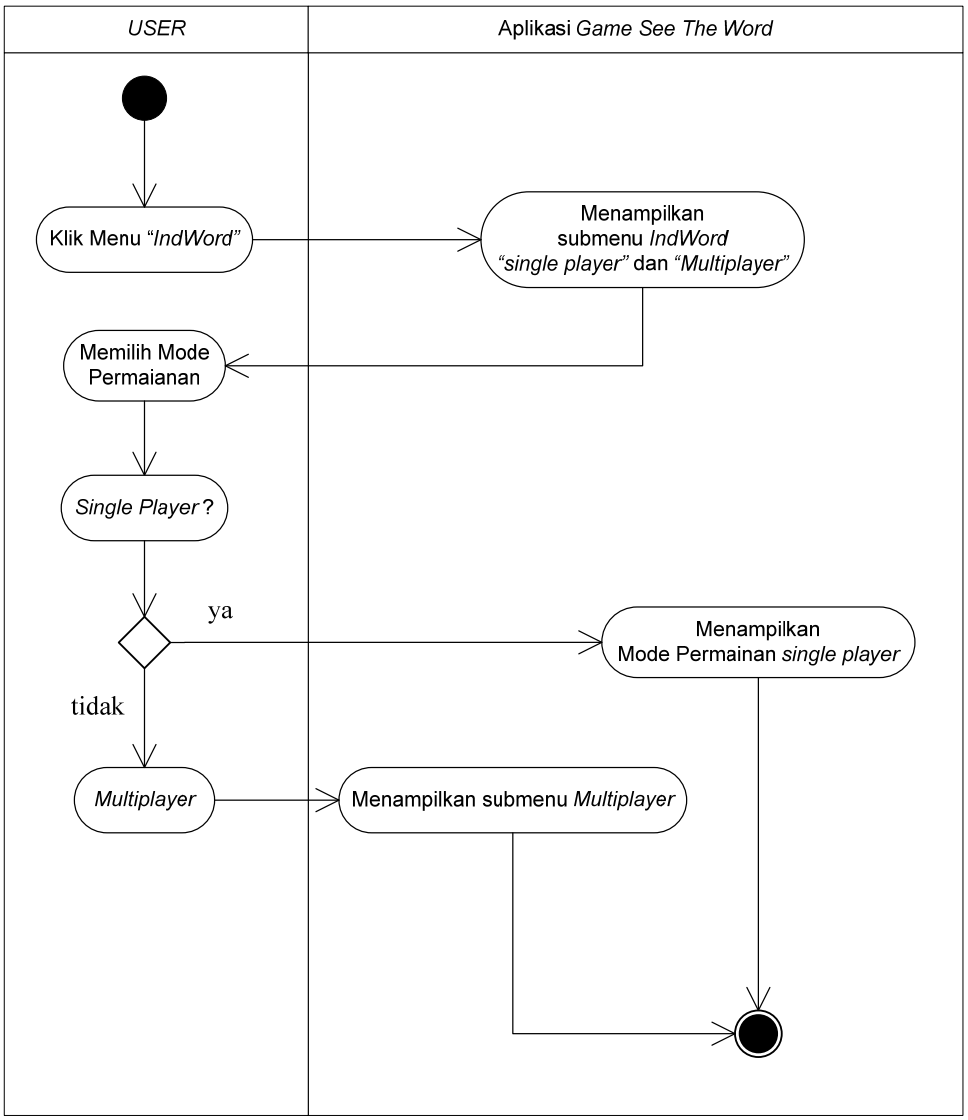
e. Klik tombol *"submit"*	

3.1.5.3 *Activity Diagram*

Activitydiagram berisi penjelasan lebih lengkap mengenai urutan suatu proses yang terjadi antara pelaku dan sistem. Aktivitas yang pertama selalu dimulai dari pelaku, kemudian akan mendapat respon dari sistem yang nantinya akan menampilkan akhir dari proses tersebut.

3.1.5.3.1 Permainan Bahasa Indonesia

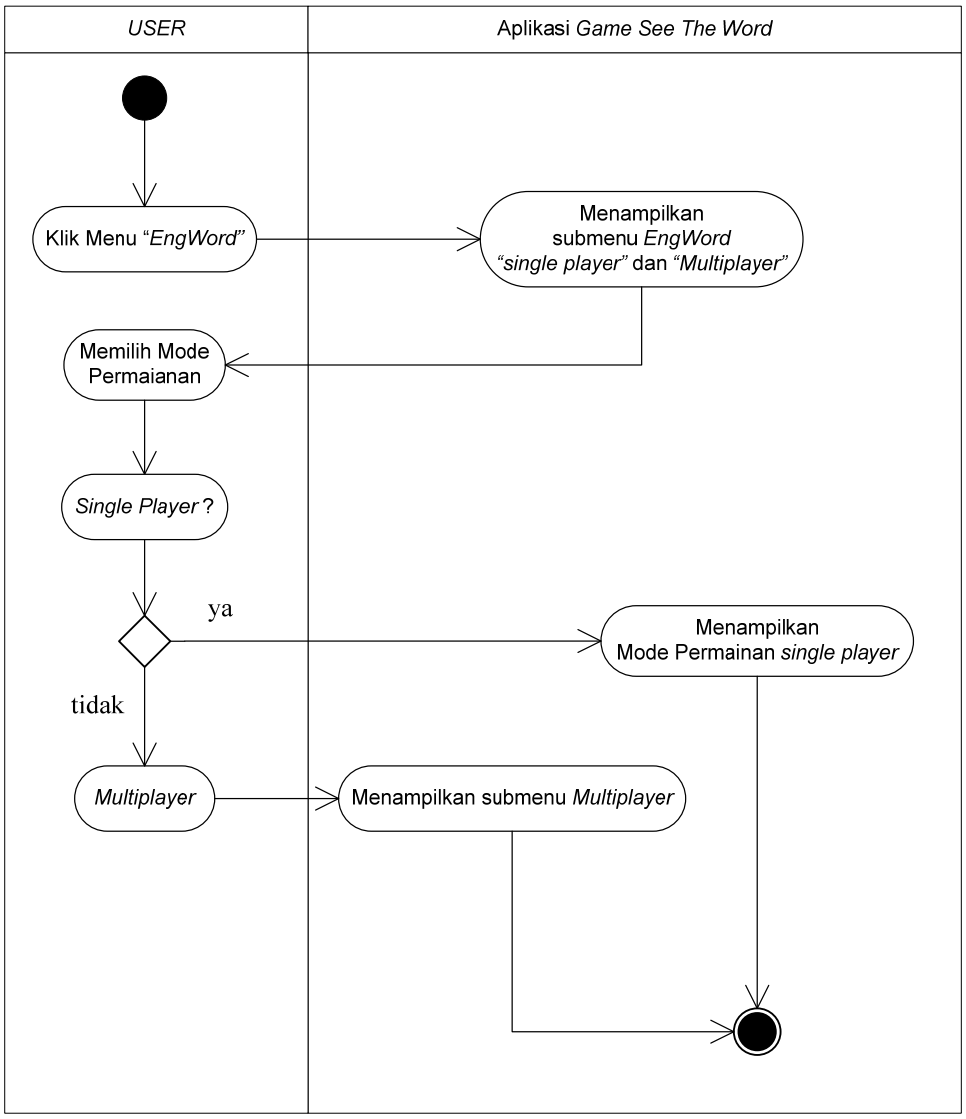
Proses awal penyajian *game* yang terjadi saat *user* memilih salah satu jenis *game* dengan menggunakan bahasa Indonesia kemudian sistem akan menampilkan submenu selanjutnya untuk memilih mode permainan *single player* atau *multiplayer* dan selanjutnya sistem menampilkan area permainan sesuai mode *game* yang dipilih *user*.



Gambar 3.13 *Activity Diagram* Permainan Bahasa Indonesia

3.1.5.3.2 Permainan Bahasa Inggris

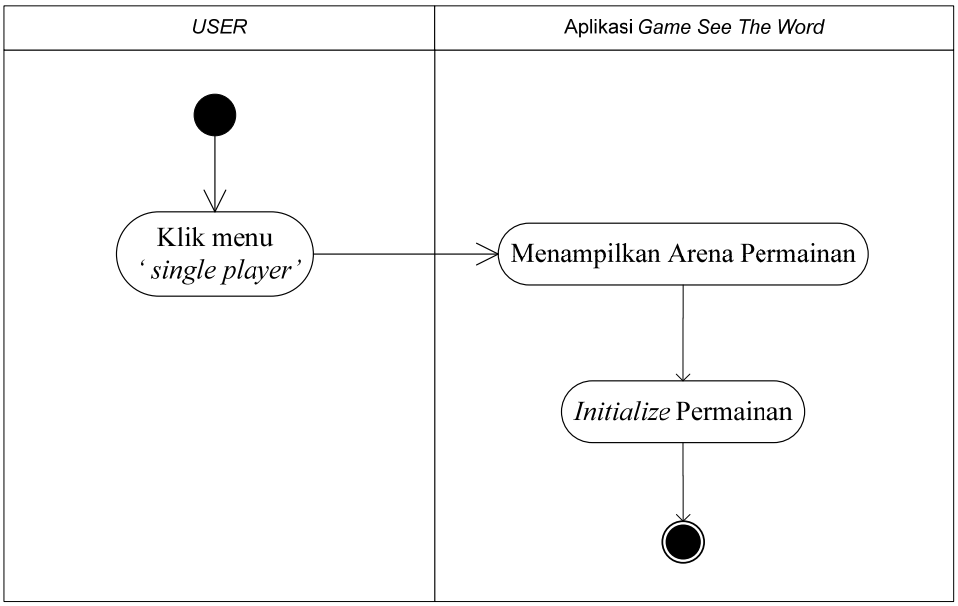
Proses awal penyajian *game* yang terjadi saat *user* memilih salah satu jenis *game* dengan menggunakan bahasa Inggris kemudian sistem akan menampilkan sub menu selanjutnya untuk memilih mode permainan *single player* atau *multiplayer* dan selanjutnya sistem menampilkan area permainan sesuai mode *game* yang dipilih *user*.



Gambar 3.14 *Activity Diagram* Permainan Bahasa Inggris

3.1.5.3.3 Single Player

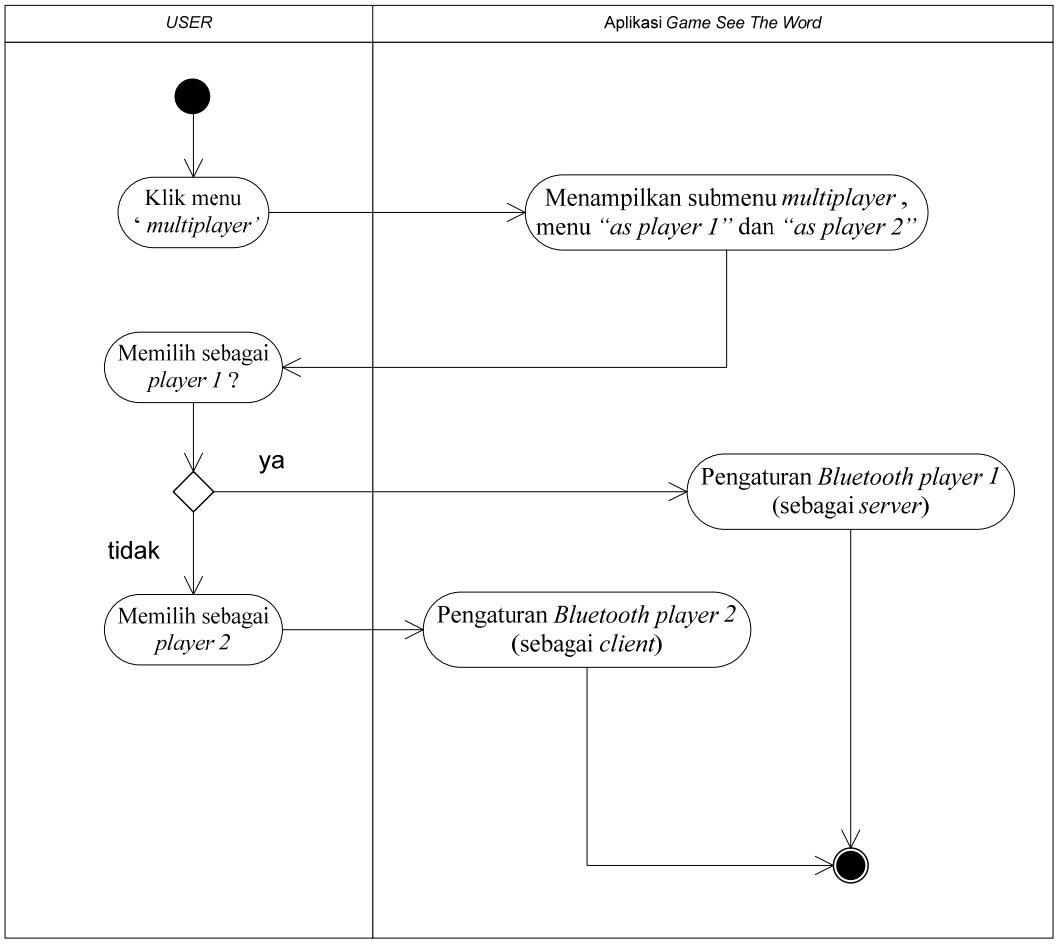
Proses ini terjadi pada saat *user* memilih menu "*single player*" pada sub menu dar *Indword* atau juga *Engword*, sistem akan menampilkan arena bermain dalam mode *single player* dan *user* dapat memulai permainan. Proses permainan mode *single player* ini dapat dilihat pada gambar dibawah ini.



Gambar 3.15 *Activity Diagram Single Player*

3.1.5.3.4 Multiplayer

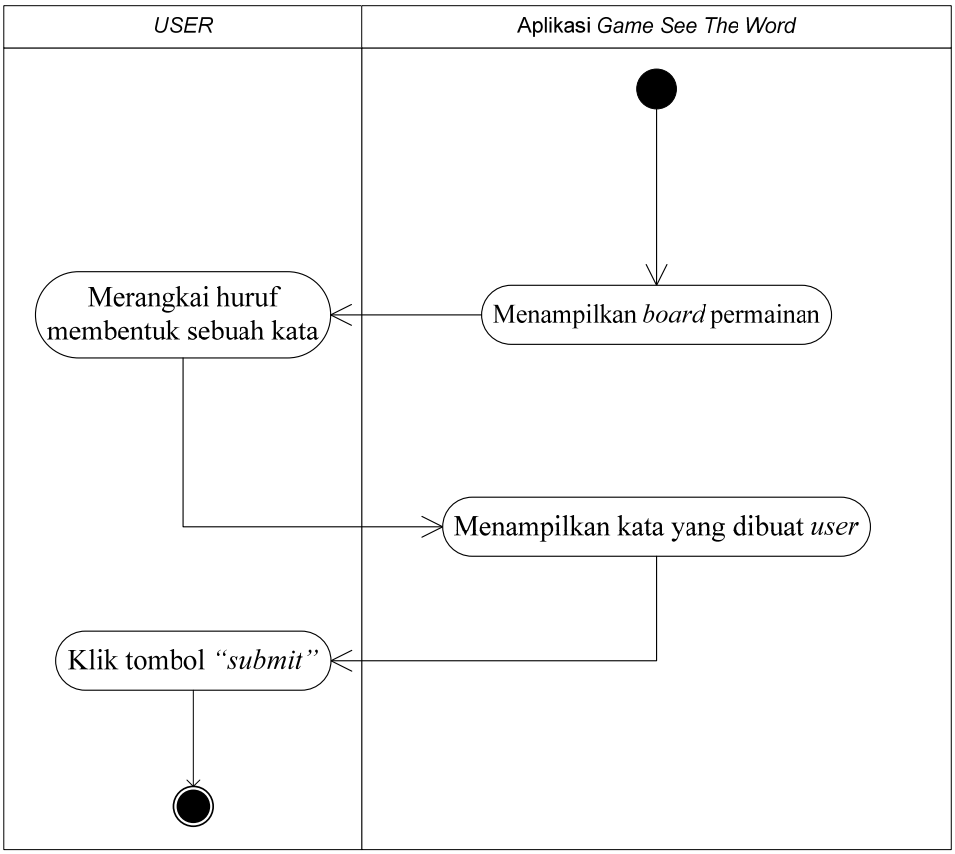
Prosesini terjadi pada saat *user* memilih menu "*Multiplayer*" pada sub menu *IndWord* atau juga *EngWord*, sistem akan menampilkan arena bermain dalam mode *single player* dan *user* dapat memulai permainan sebelum permainan ini sistem akan mengaktifkan dan menyambungkan perangkat antar *user*. Proses permainan mode *multiplayer* ini dapat dilihat pada gambar dibawah ini.



Gambar 3.16 ***Activity Diagram Multiplayer***

3.1.5.3.5 Initialize Game

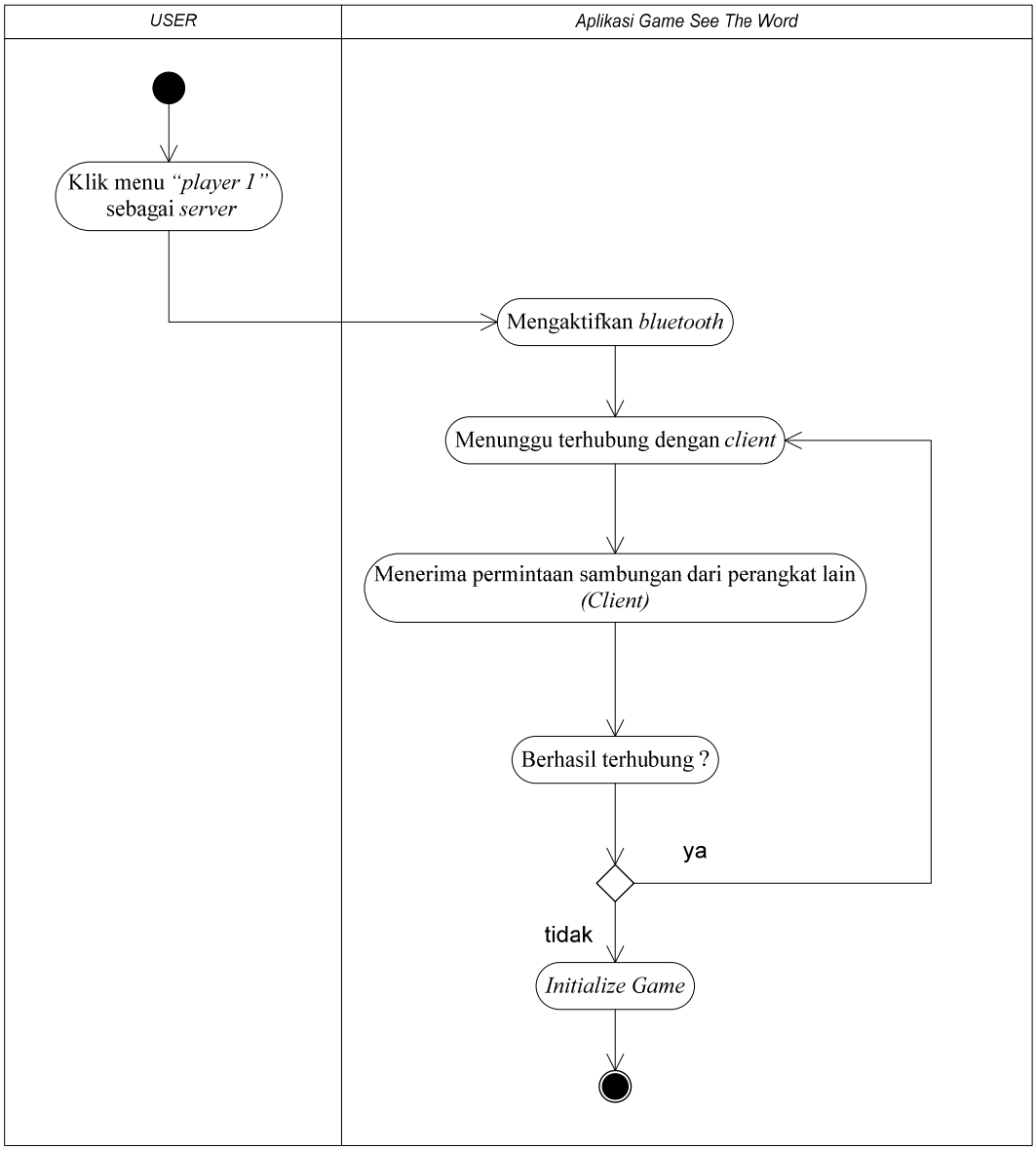
Pada proses ini sistem akan menampilkan *board* permainan dan *user* siap untuk memulai permainan. Proses ini dapat dilihat pada gambar dibawah ini.



Gambar 3.17 ***Activity Diagram Initialize Game***

3.1.5.3.6 Pengaturan *Bluetooth Player 1*

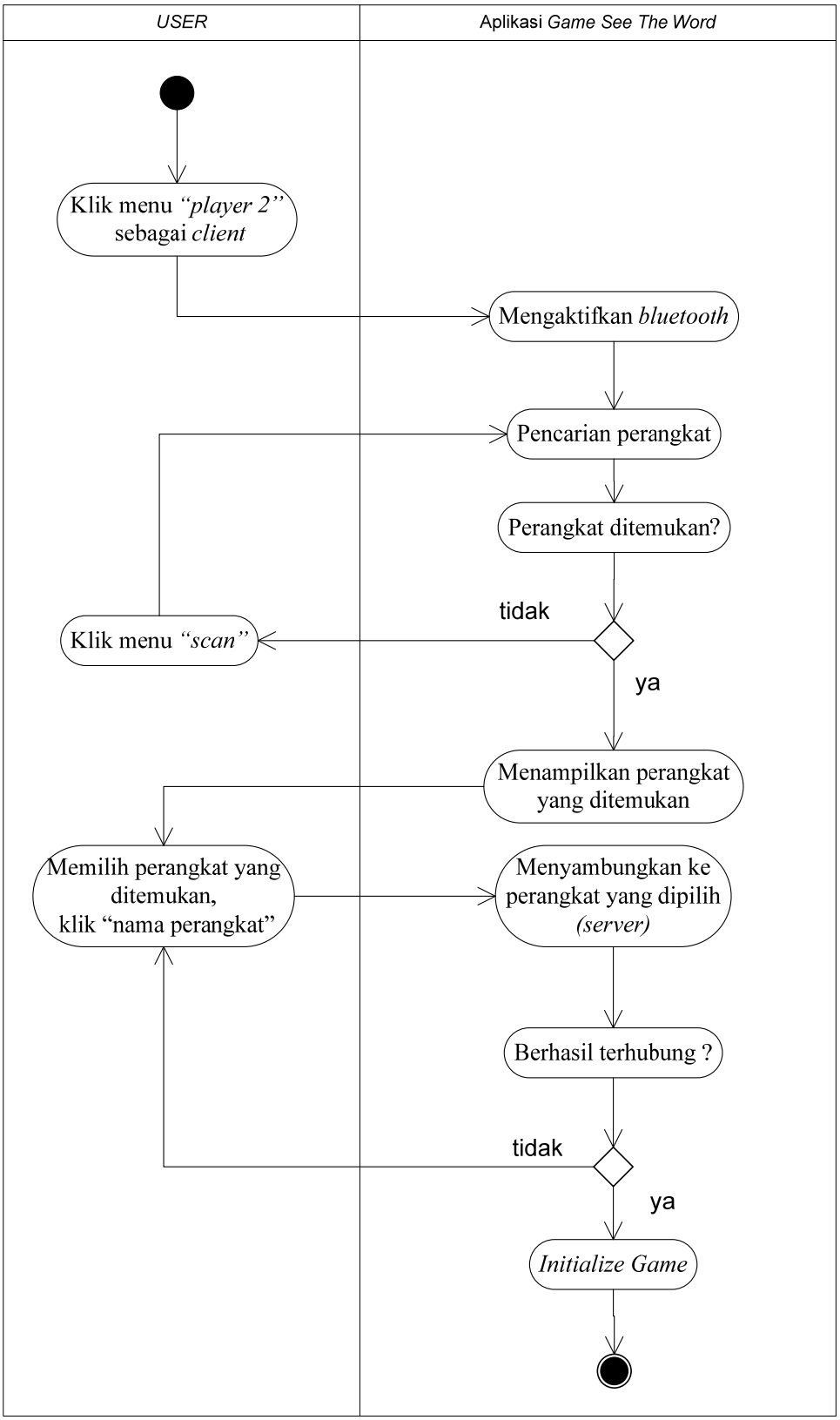
Prosesini terjadi dengan terlebih dahulu user memilih sebagai *player 1 (server)*, lalu *Bluetooth*akan diaktifkan dan menyambungkan dengan perangkat lain apabila proses sambungan antar perangkat gagal maka sistem akan kembali menunggu sambungan dari perangkat lain.



Gambar 3.18 *Activity Diagram*Pengaturan *Bluetooth Player 1*

3.1.5.3.7 Pengaturan *Bluetooth Player 2*

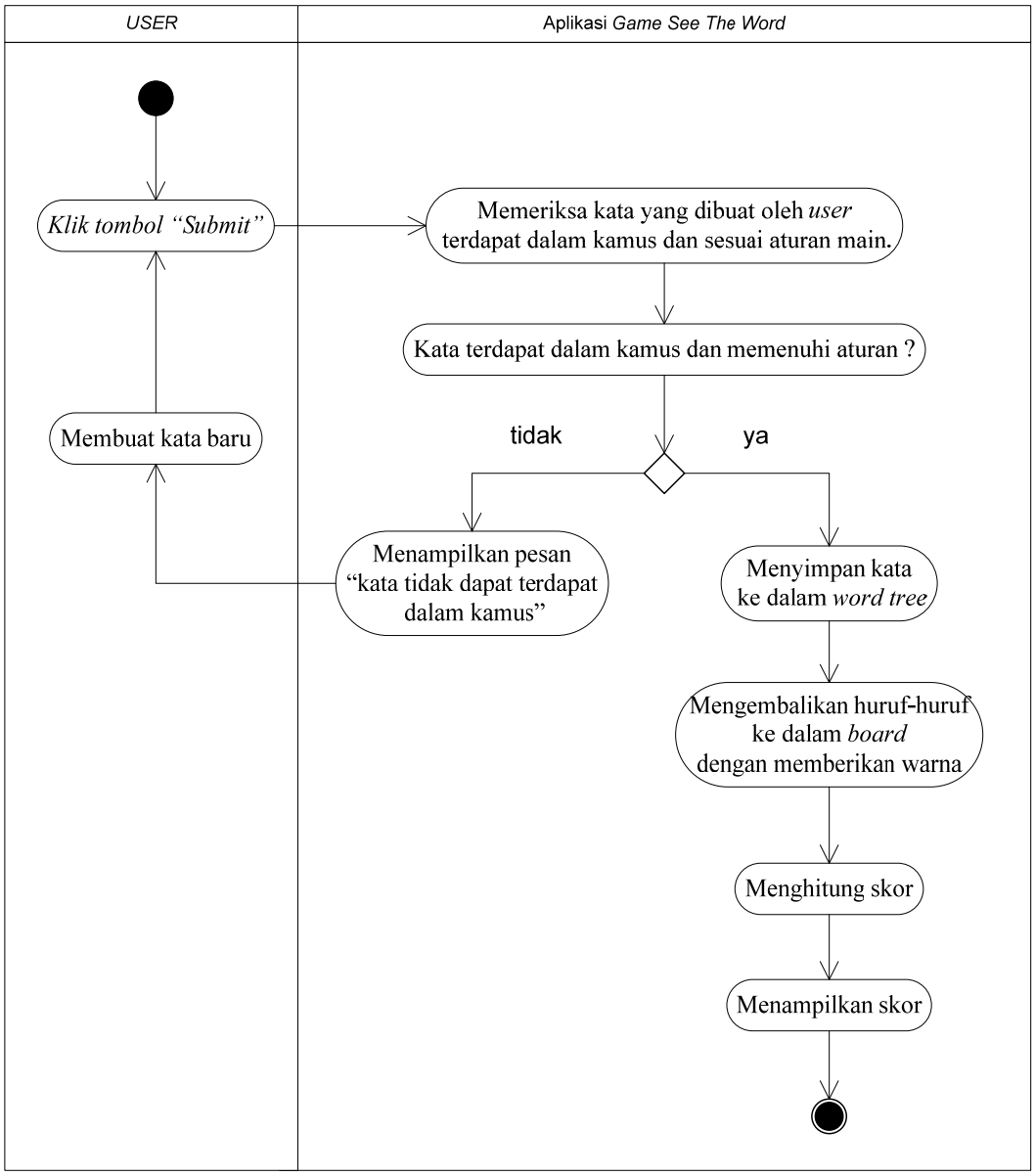
Prosesini terjadi dengan terlebih dahulu user memilih sebagai *player* 2 *(client),* lalu *bluetooth*akan diaktifkan dan mencari perangkat lainnya apabila berhasil menemukan maka akan ditampilkan untuk dipilih oleh *user* dengan mengklik nama perangkatnya. Proses pengaturan *Bluetooth player* 2 dapat dilihat pada gambar dibawah ini.


USER
Aplikasi Game See The Word
Klik menu "player 2" sebagai client
Mengaktifkan bluetooth
Pencarian perangkat
Perangkat ditemukan?
tidak
Klik menu "scan"
ya
Menampilkan perangkat yang ditemukan
Memilih perangkat yang ditemukan, klik "nama perangkat"
Menyambungkan ke perangkat yang dipilih (server)
Berhasil terhubung ?
tidak
ya
Initialize Game

Gambar 3.19 *Activity Diagram*Pengaturan *Bluetooth Player 2*

3.1.5.3.8 Submit

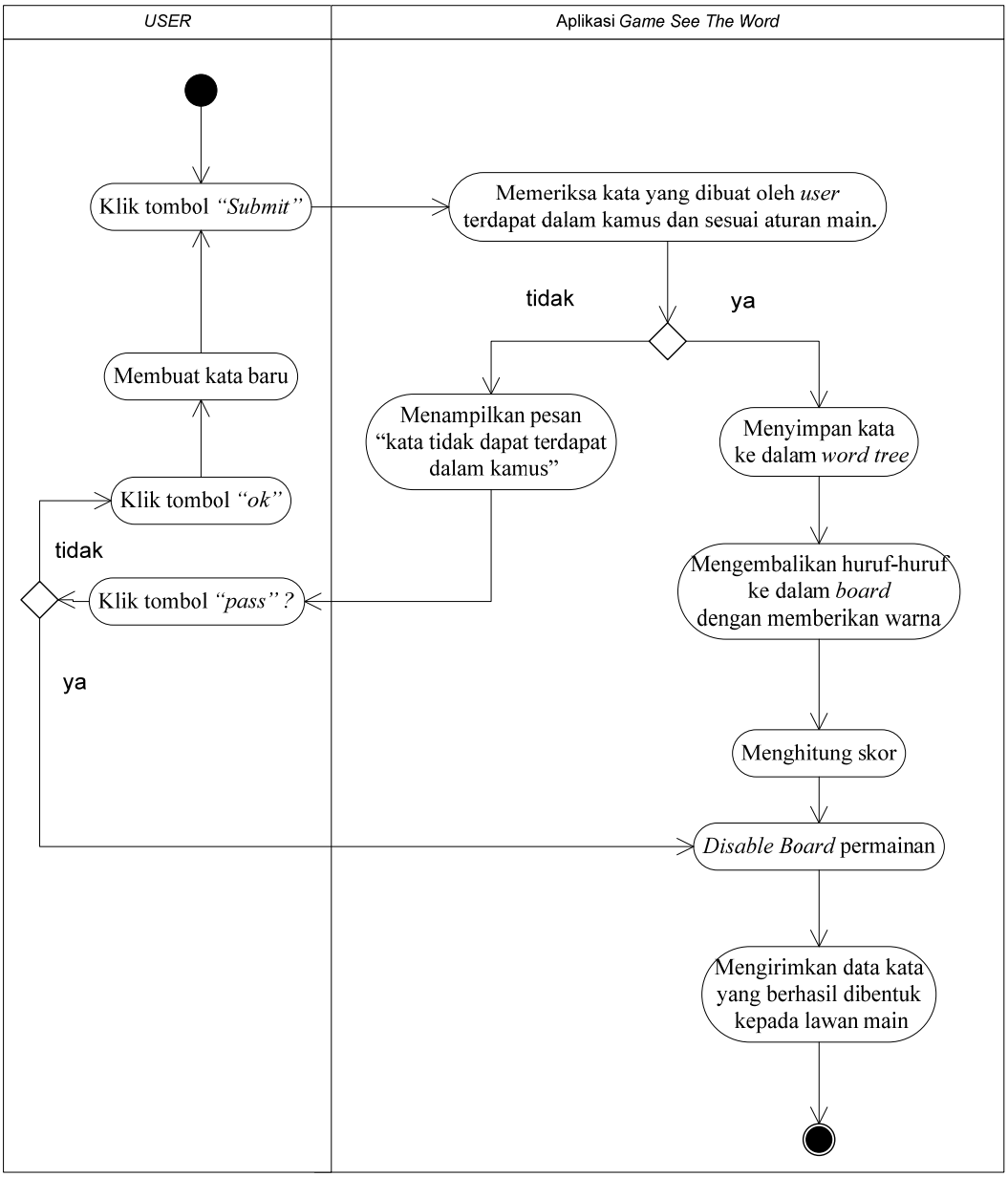
Prosesini akan terjadi pada saat *user* mengklik tombol *"submit"* dan sistem akan memeriksa kata yang telah dibuat oleh *user* apakah kata tersebut sesuai dengan aturan permainan, terdapat dalam kamus dan sudah digunakan sebelumnya. Prosesini dapat dilihat pada gambar dibawah ini.



Gambar 3.20 *Activity Diagram Submit*

3.1.5.3.9 Submit Multiplayer

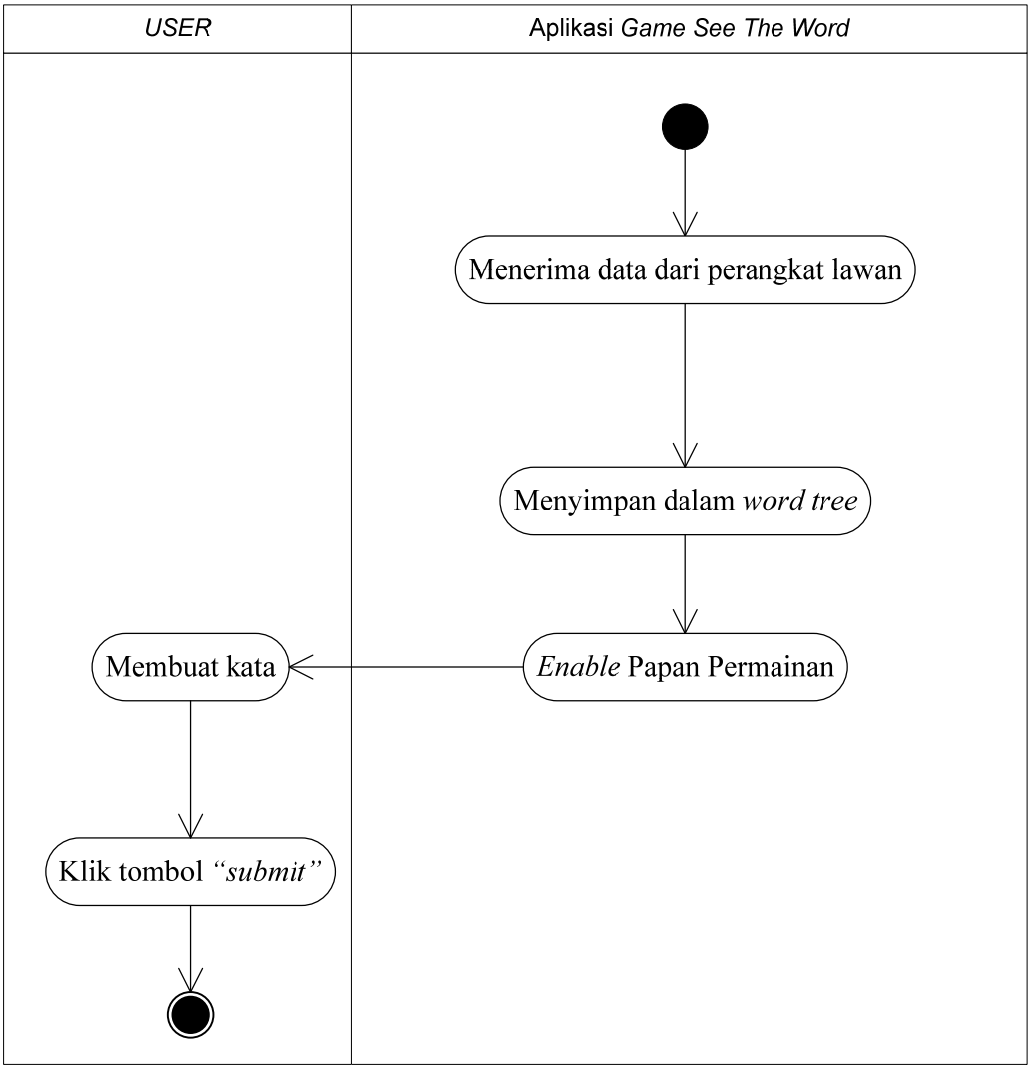
Proses ini akan terjadi pada saat *user* mengklik tombol "*submit*" dan sistem akan memeriksa kata yang telah dibuat oleh *user* apakah kata tersebut sesuai dengan aturan permainan dan terdapat dalam kamus dan sistem akan mengirimkan data kata yang berhasil dibuat *user* ke perangkat lawan main. Prosesini dapat dilihat pada gambar dibawah ini.



Gambar 3.21 ***Activity Diagram Submit Multiplayer***

3.1.5.3.10 Penerimaan Data

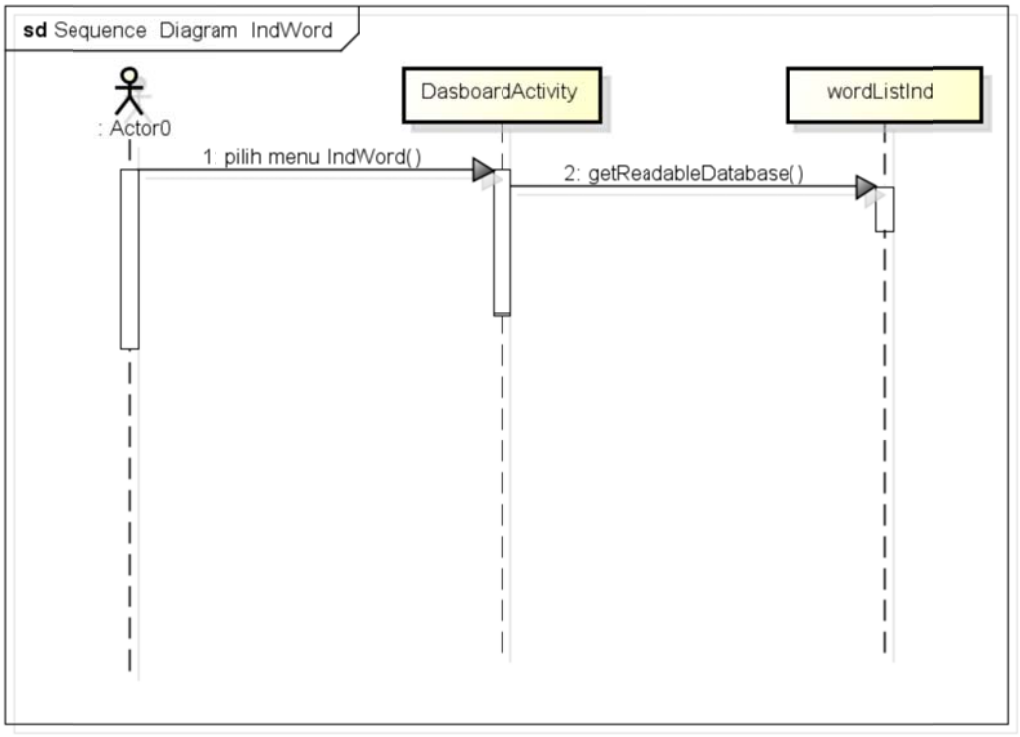
Proses ini akan terjadi pada saat *user* menerima data dari lawan main dan memulai giliran bermain dengan mengaktifkan papan permainan. Prosesini dapat dilihat pada gambar dibawah ini.



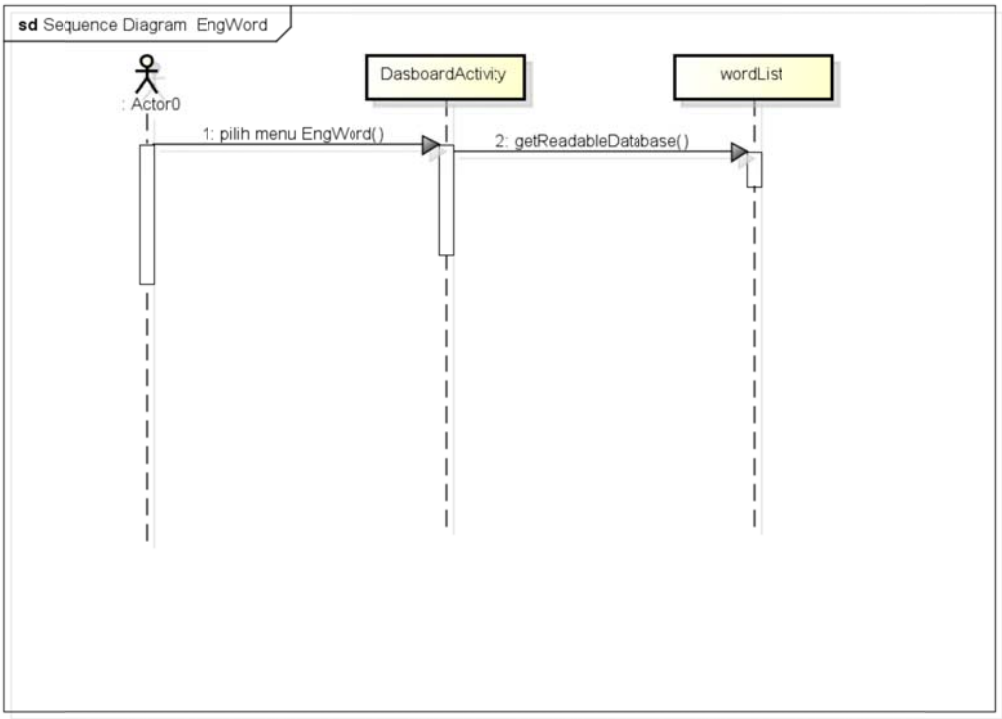
Gambar 3.22 *Activity Diagram* Penerimaan

3.1.5.4 *Sequence Diagram*

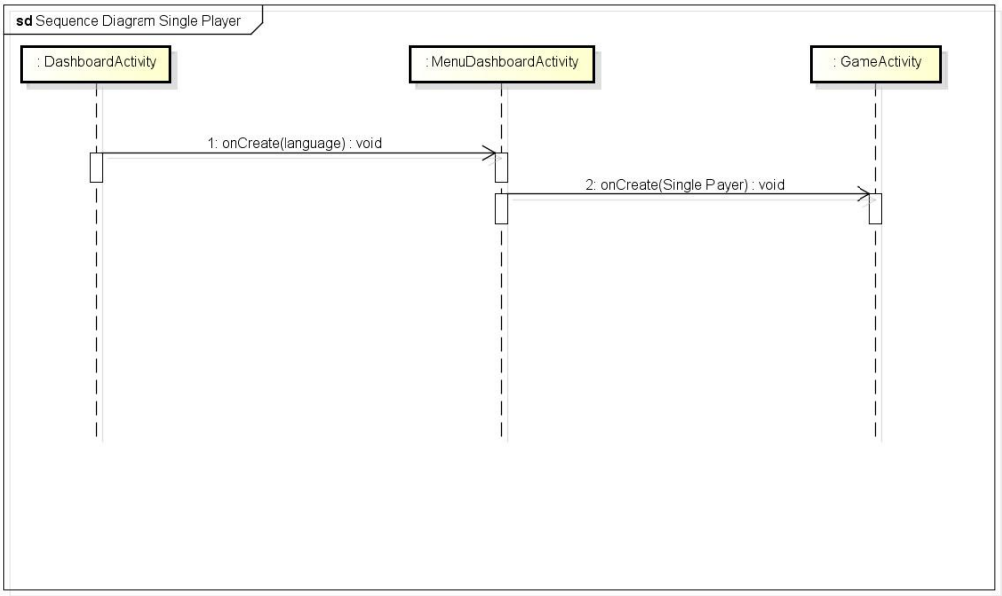
*Sequence diagram*mengambarkan perilaku objek pada *usecase* dengan mendeskripsikan waktu hidup dan *message* yang dikirimkan dan diterima antar objek. Berikut merupakan *sequence diagram* yang terdapat pada *game* ini.



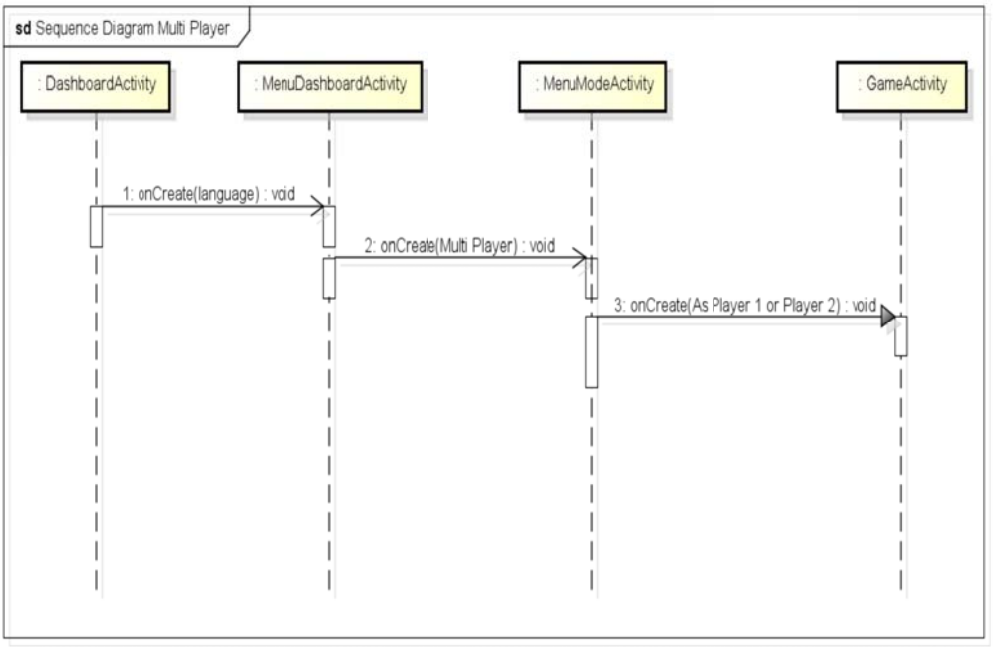
Gambar 3.23 *Sequence Diagram* untuk *Use Case* Permainan Bahasa Indonesia



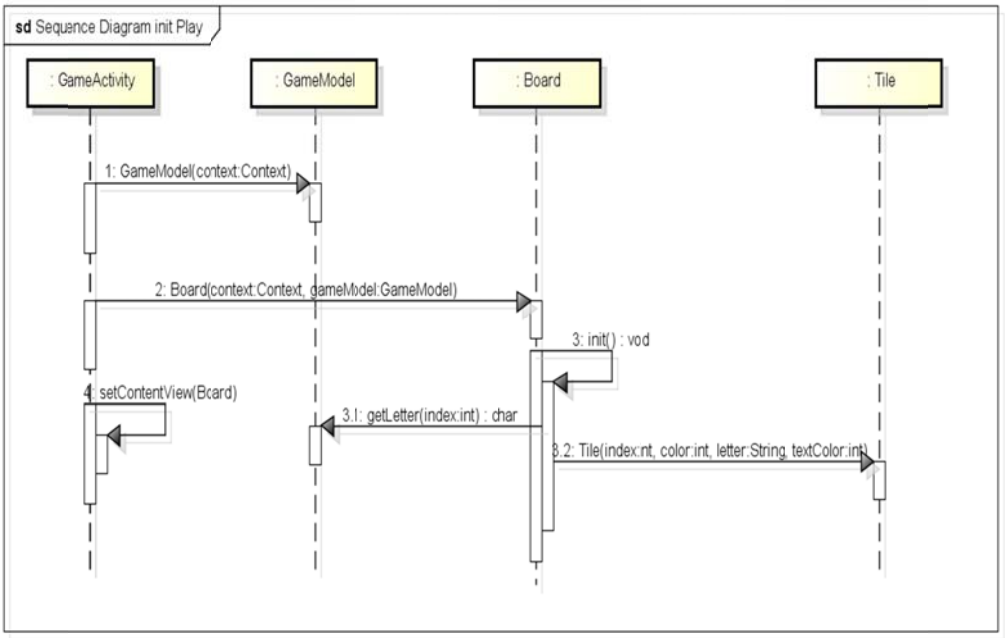
Gambar 3.24 *Sequence Diagram* untuk *Use Case* Permainan Bahasa Inggris



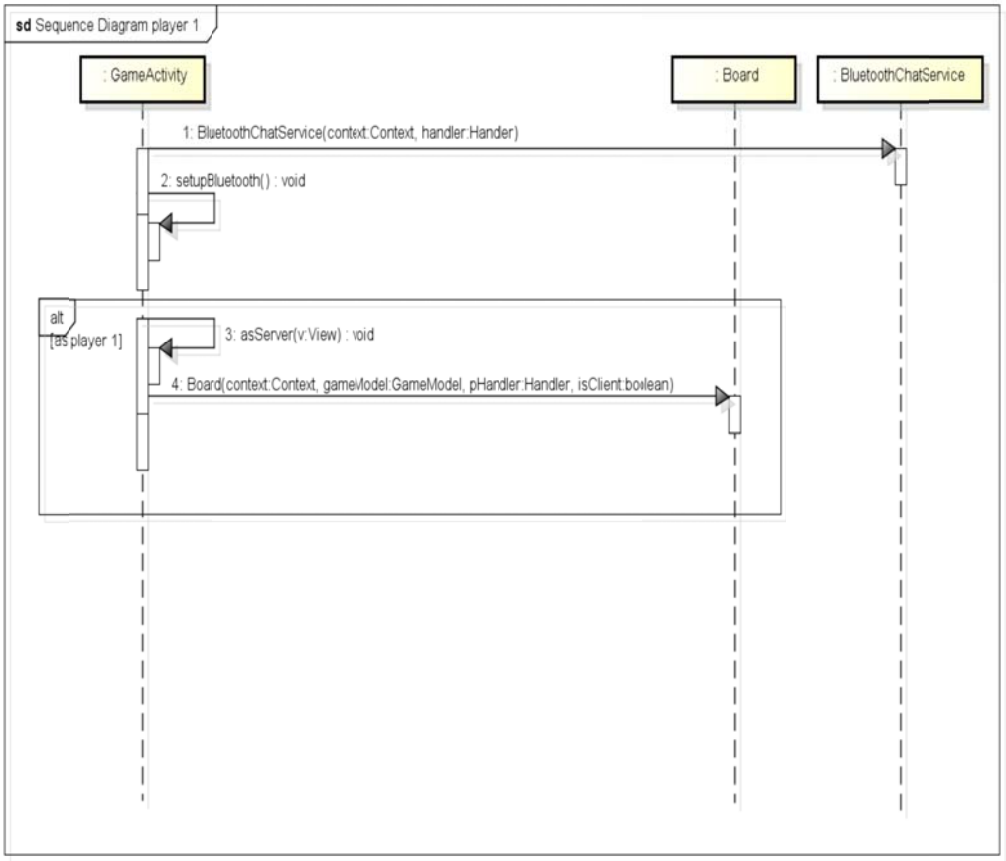
Gambar 3.25 *Sequence Diagram* untuk *Use Case Single Player*



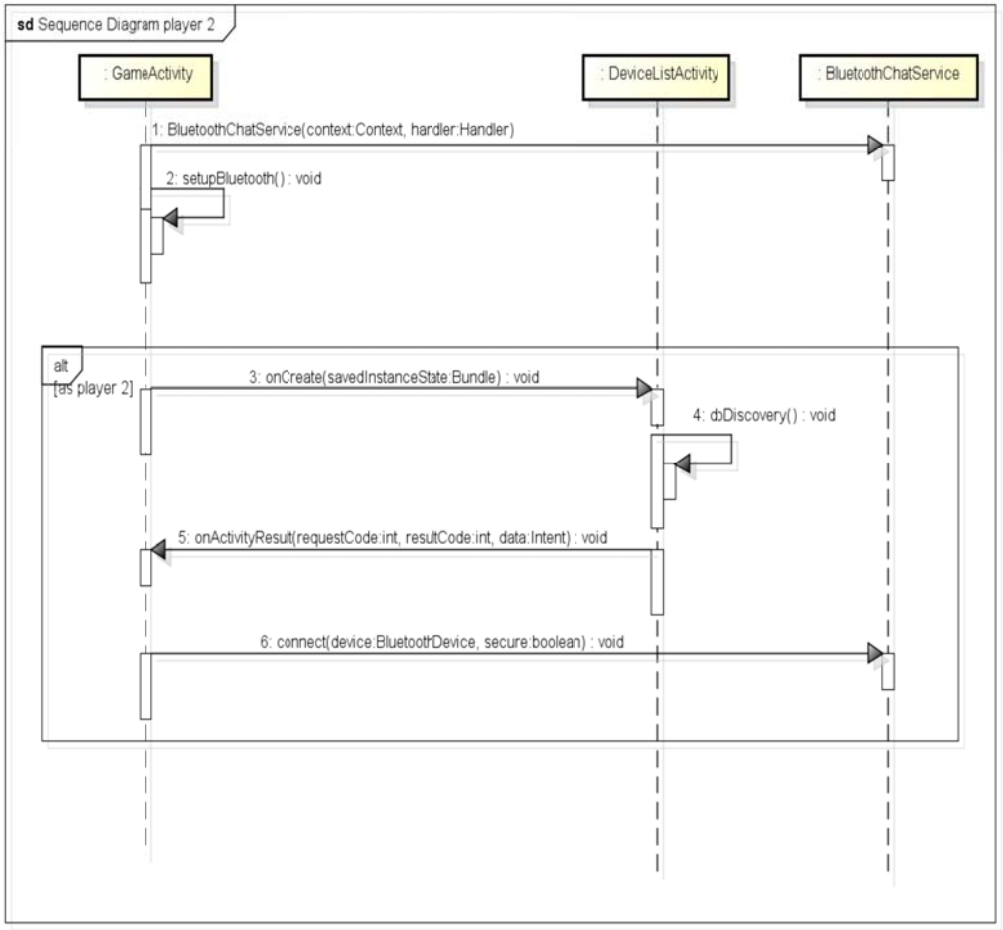
Gambar 3.26 *Sequence Diagram* untuk *Use Case Multiplayer*



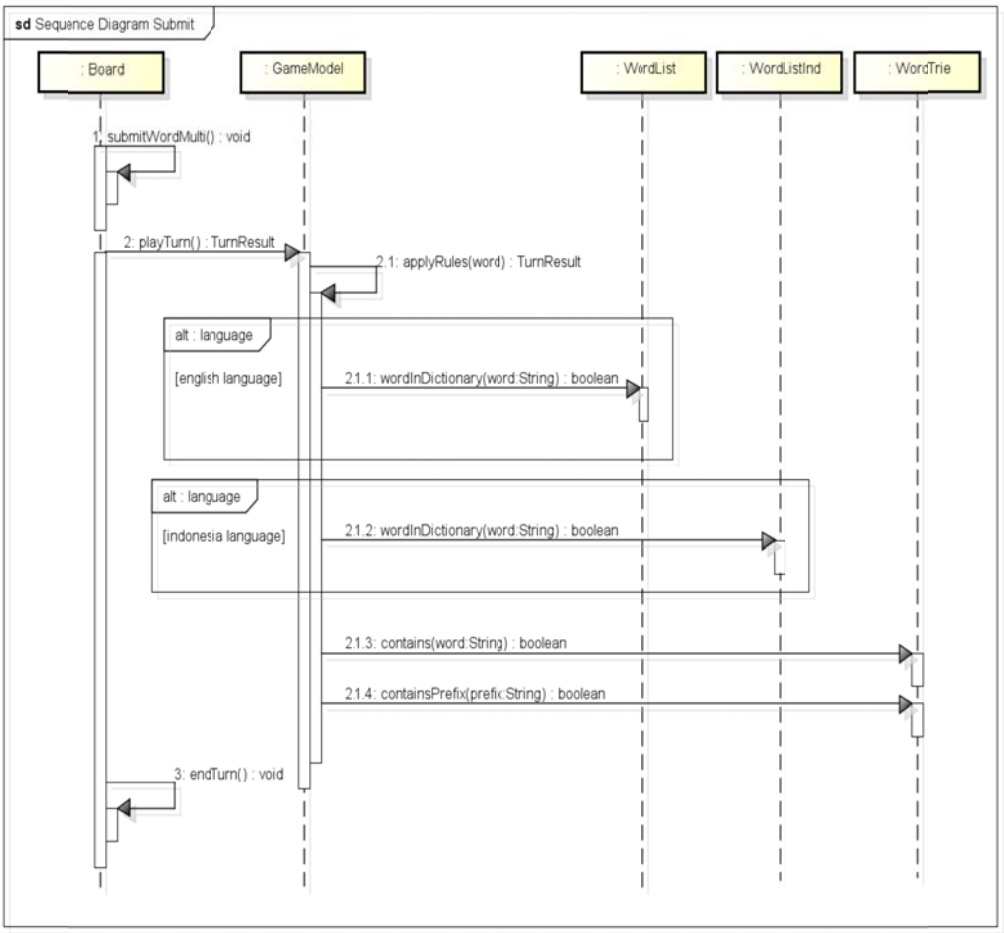
Gambar 3.27 *Sequence Diagram* untuk *Use Case Initialize Game*



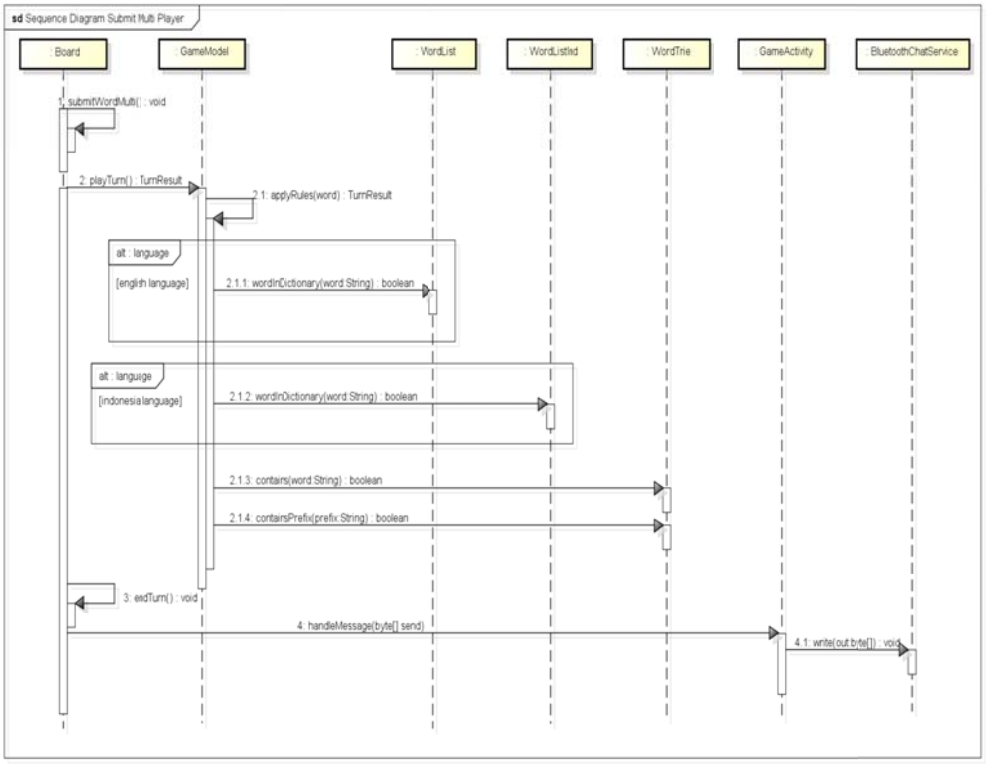
Gambar 3.28 *Sequence Diagram* untuk *Use Case* Pengaturan *Bluetooth Player 1*



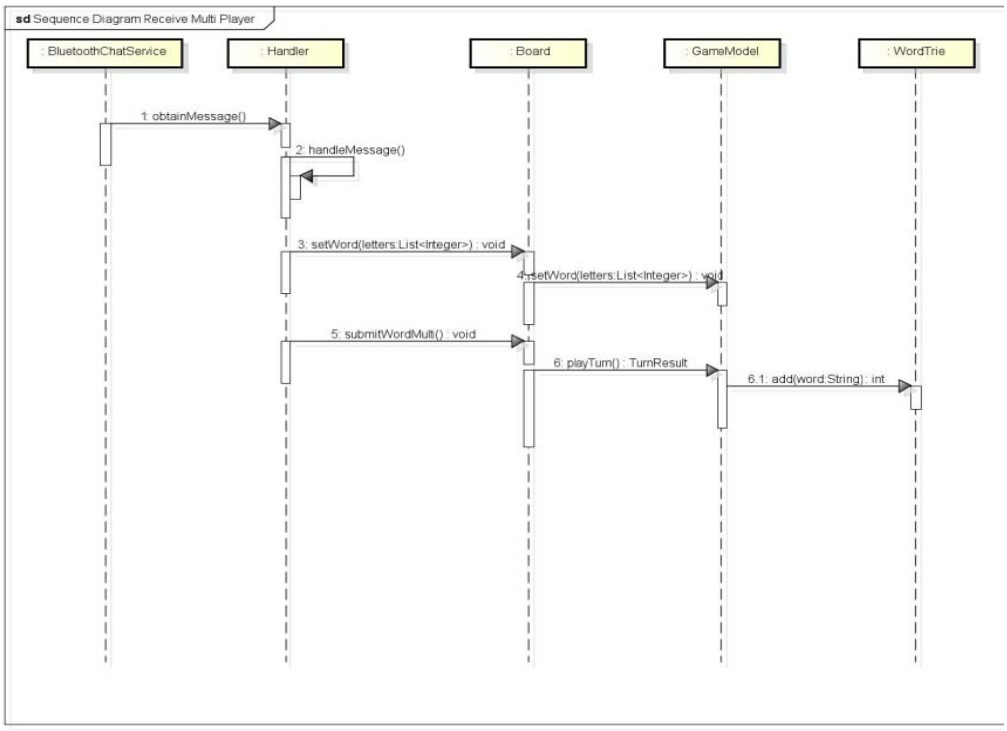
Gambar 3.29 *Sequence Diagram* untuk *Use Case* Pengaturan *Bluetooth Player 2*



Gambar 3.30 *Sequence Diagram* untuk *Use Case Submit*



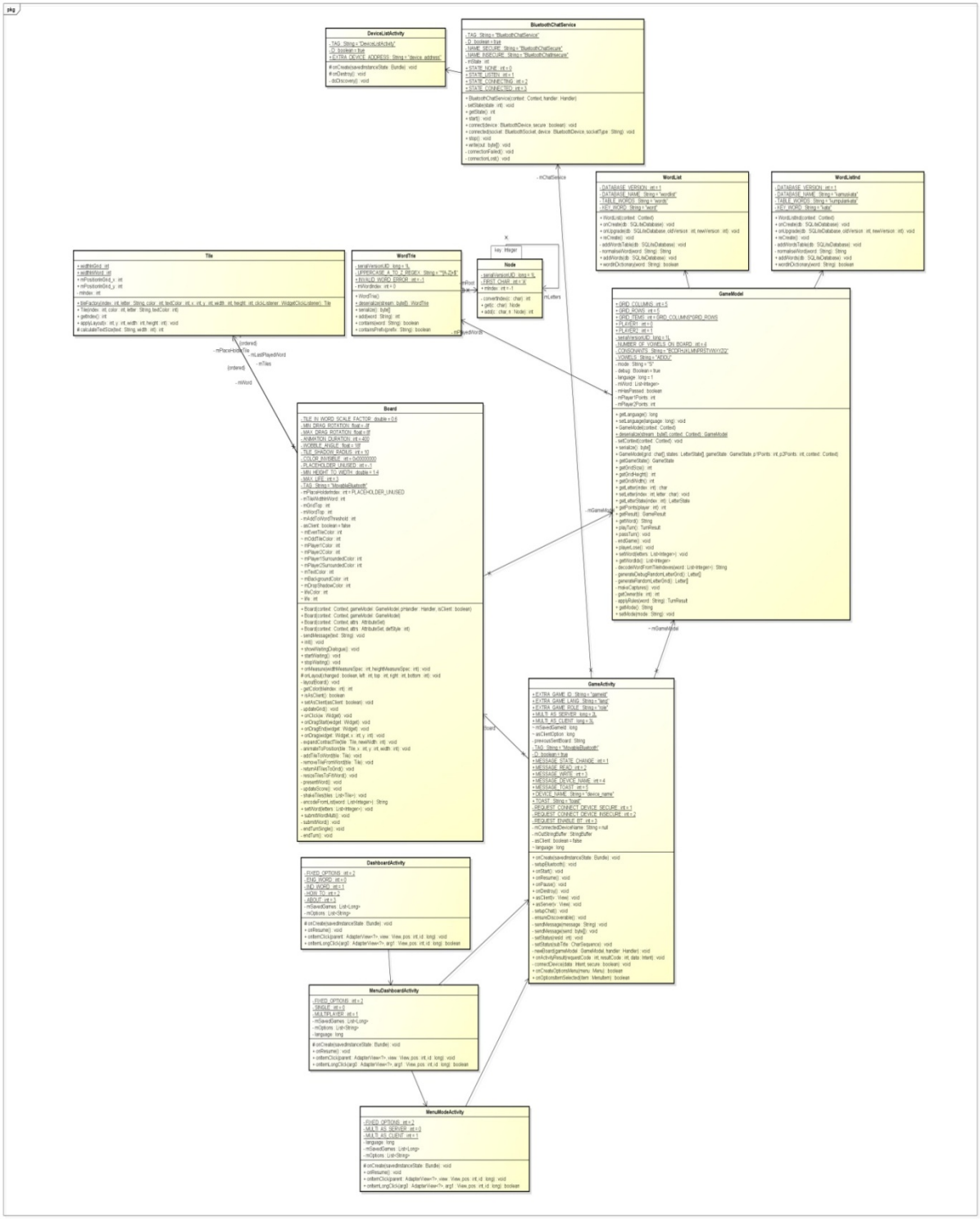
Gambar 3.31 *Sequence Diagram* untuk *Use Case Submit Multiplayer*



Gambar 3.32 *Sequence Diagram* untuk *Use Case* Penerimaan Data

3.1.5.5 *Class Diagram*

Berikut merupakan*class diagram*yang terdapat pada aplikasi *game* "*see the word"*



Gambar 3.33 ***Class Diagram game See the Word***

Deskripsi untuk masing-masing class akan dijelaskan lebih lanjut pada tabel-tabel berikut ini:

Tabel 3.14 Deskripsi *Class Tile*

Class	**Tile**	
Ket.	*Class* untuk mengatur penempatan tile huruf	
Metode	**Nama**	**Sifat**
	tileFactory(index : int,letter : string,color : int,textColor : int,X : int, Y : intwidht : int,height : int,clickListener : WidgetClickListener) : Tile	Public
	Tile(index : int, color : int, String letter : int, textColor : int)	Public
	getIndex() : int	Public
	applyLayout(x : int, y: int, width : int, height : int) : void	Public
	calculateTextSize(text : String, width : int) : int	Protected

Tabel 3.15 Deskripsi *Class Board*

Class	**Board**	
Ket.	Class untuk Mengontrolredering dari papan, termasuk huruf grid, kata, Skor, dan tombol "clear" dan tombol "submit" .	
Metode	**Nama**	**Sifat**
	Board(context : Context, gameModel :GameModel,pHandler:Handler ,isClient:boolean)	Public
	Board(context : Context, gameModel :GameModel)	Public
	Board(context : Context,attrs : AttributeSet)	Public
	Board(context : Context, attrs : AttributeSet ,defStyle : int)	Public
	sendMessage(text :String) : void	private
	init() : void	Public
	showWaitingDialogue() : void	Public
	startWaiting() : void	Public
	stopWaiting() : void	Public
	onMeasure(widthMeasureSpec : int, heightMeasureSpec : int) : void	public
	onLayout(changed:boolean, left : int,top :int,right : int,bottom : int) : void	private
	layoutBoard() : void	private

	getColor(tileIndex :int) : int	private
	isAsClient() : Boolean	public
	setAsClient(asClient : boolean) : void	public
	updateGrid() : void	private
	onClick(w : Widget) : void	public
	onDragStart(Widget : widget) : void	public
	onDragEnd(Widget: widget) : void	public
	onDrag(Widget: widget,x:int, y: int) : void	public
	expandContractTile(Tile: tile,newWidth: int) : void	private
	animateToPosition(Tile : tile,x:int, y:int,width : int) : void	private
	addTileToWord(Tile: tile) : void	private
	removeTileFromWord(Tile : tile) : void	private
	returnAllTilesToGrid() : void	private
	resizeTilesToFitWord() : void	private
	presentWord() : void	private
	updateScore() : void	private
	shakeTiles(tiles :List<Tile>) : void	private
	encodeFromList(List<Integer> word) : string	private
	setWord(letters:List<Integer>) : void	public
	submitWordMulti() : void	public
	submitWord() : void	private
	endTurnSingle() : void	private
	endTurn() : void	private

Tabel 3.16 Deskripsi *Class GameModel*

Class	**GameModel**	
Ket.	Class untuk Mengelola keadaan permainan, termasuk keadaan grid, kata, kata-kata yang dimainkan sebelumnya, dan skor pemain.	
Metode	**Nama**	**Sifat**
	getLanguage() : long	Public
	setLanguage(language:long) : void	Public
	deserialize(stream:byte[] ,context:Context) : GameModel	Public
	setContext(context:Context) : void	private
	serialize() :byte[]	Public
	GameModel(grid:char[],states : LetterState[],gameState: GameState,p1Points :int, p2Points : int, context:Context)	Public

	getGameState() : GameState	Public
	getGridSize() : int	Public
	getGridHeight() : int	Public
	getGridWidth() : int	Public
	getLetter(index:int) : char	Public
	setLetter(index:int,letter : char) : void	Public
	getLetterState(index:int) : LetterState	Public
	getPoints(player:int) : int	Public
	getResult(): GameResult	Public
	getWord() :String	Public
	playTurn() : TurnResult	Public
	passTurn() : void	Public
	endGame() : void	private
	playerLose() : void	Public
	setWord(letters: List<Integer>) : void	Public
	getWordIdx() : List<Integer>	Public
	decodeWordFromTileIndexes(word:List<Integer>) :String	private
	generateDebugRandomLetterGrid() : Letter[]	private
	generateIndonesiaLetterGrid() : Letter[]	private
	generateRandomLetterGrid() : Letter[]	private
	makeCaptures() : void	private
	getOwner(int tile) : int	private
	applyRules(String word) : TurnResult	private
	getMode() :String	Public
	setMode(mode:String) : void	Public

Tabel 3.17 Deskripsi *Class Node*

Class	**Node**	
Ket.	Class untuk memnipulasi kata untuk disimpan dalam hashmap	
Metode	**Nama**	**Sifat**
	convertIndex(c:char) : int	private
	get(c:char) : Node	Public
	add(c:char, n : Node) : int	Public

Tabel 3.18 Deskripsi *Class WordTrie*

Class	**WordTrie**	
Ket.	Class untuk membuat dan mengakses word tree	
Metode	**Nama**	**Sifat**

	WordTrie()	Public
	serialize() : byte[]	Public
	add(word : String) :int	Public
	contains(word : String) : boolean	Public
	containsPrefix(prefix :String) : boolean	Public

Tabel 3.19 Deskripsi *Class WordList*

Class	**WordList**	
Ket.	Class untuk Mengelola pembuatan dan akses ke database daftar kata, yang dapat digunakan untuk menentukan apakah string tertentu adalah kata dalam bahasa Inggris.	
Metode	**Nama**	**Sifat**
	WordList(context:Context)	Public
	onCreate(db : SQLiteDatabase) : void	Public
	onUpgrade(db : SQLiteDatabase,oldVersion : int,newVersion ; int) : void	Public
	reCreate() : void	Public
	addWordsTable(db: SQLiteDatabase) : void	private
	normaliseWord(String word) : string	private
	addWords(db: SQLiteDatabase) : void	public
	wordInDictionary(word :String) : boolean	public

Tabel 3.20Deskripsi *Class WordListInd*

Class	**WordListInd**	
Ket.	Class untuk Mengelola pembuatan dan akses ke database daftar kata, yang dapat digunakan untuk menentukan apakah string tertentu adalah kata dalam bahasa Indonesia.	
Metode	**Nama**	**Sifat**
	WordList(context:Context)	Public
	onCreate(db : SQLiteDatabase) : void	Public
	onUpgrade(db : SQLiteDatabase,oldVersion : int,newVersion ; int) : void	Public
	reCreate() : void	Public

	addWordsTable(db: SQLiteDatabase) : void	private
	normaliseWord(String word) : string	private
	addWords(db: SQLiteDatabase) : void	public
	wordInDictionary(word :String) : boolean	public

Tabel 3.21 Deskripsi *Class DashBoardActivity*

Class	**DashBoardActivity**	
Ket.	Class untuk menampilkan menu utama, yang memungkinkan penciptaan permainan baru, memuat permainan sebelumnya, atau membuka petunjuk.	
Metode	**Nama**	**Sifat**
	onCreate(savedInstanceState :Bundle) : void	protected
	onResume() : void	Public
	onItemClick(parent : AdapterView<?> ,view : View,pos : int,id : long) : void	Public
	onItemLongClick(arg0 : AdapterView<?>, arg1: View,pos : int ,id :long) : boolean	Public

Tabel 3.22 Deskripsi *Class MenuDashBoardActivity*

Class	**MenuDashBoardActivity**	
Ket.	Class untuk menampilkan menu utama, yang memungkinkan penciptaan permainan baru, memuat permainan sebelumnya, atau membuka petunjuk.	
Metode	**Nama**	**Sifat**
	onCreate(savedInstanceState :Bundle) : void	protected
	onResume() : void	Public
	onItemClick(parent : AdapterView<?> ,view : View,pos : int,id : long) : void	Public
	onItemLongClick(arg0 : AdapterView<?>, arg1: View,pos : int ,id :long) : boolean	Public

Tabel 3.23 Deskripsi *Class MenuModeActivity*

Class	**MenuModeActivity**
Ket.	Class untuk menampilkan menu utama, yang memungkinkan penciptaan permainan baru, memuat permainan sebelumnya, atau membuka petunjuk.

Metode	Nama	Sifat
	onCreate(savedInstanceState :Bundle) : void	protected
	onResume() : void	Public
	onItemClick(parent : AdapterView<?> ,view : View,pos : int,id : long) : void	Public
	onItemLongClick(arg0 : AdapterView<?>, arg1: View,pos : int ,id :long) : boolean	Public

Tabel 3.24 Deskripsi *Class DeviceListAct*

Class	**DeviceListActivity**	
Ket.	Class untuk menampilkan setiap perangkat yang dipasangkan dan yang ditemukan.	
Metode	**Nama**	**Sifat**
	onCreate(savedInstanceState :Bundle) : void	protected
	onDestroy() : void	protected
	doDiscovery() : void	private

Tabel 3.25 Deskripsi *Class BluetoothChatService*

Class	**BluetoothChatService**	
Ket.	Class untuk mensetting dan mengatur koneksi Bluetooth dengan perangkat lain. Menerima permintaan untuk terhubung, menghubungkan dengan perangkat lain dan melakuan transfer data	
Metode	**Nama**	**Sifat**
	BluetoothChatService(context:Context, handler:Handler)	Public
	setState(int state) : void	private
	getState() : int	public
	start() : void	Public
	connect(device: BluetoothDevice,secure : boolean) : void	Public
	connected(socket: BluetoothSocket, device: BluetoothDevice,socketType : String) : void	Public
	stop() : void	Public
	write(out:byte[]) : void	Public
	connectionFailed() : void	private
	connectionLost() : void	private

Tabel 3.26 Deskripsi *Class GameActivity*

Class	**GameActivity**	
Ket.	Class untuk menampilkan papan permainan	
Metode	**Nama**	**Sifat**
	onCreate(savedInstanceState :Bundle) : void	Public
	setupBluetooth() : void	private
	onStart(): void	Public
	onResume() : void	Public
	onPause() : void	Public
	onDestroy() : void	Public
	asClient(v: View)	Public
	asServer(v: View)	Public
	setupChat() : void	private
	ensureDiscoverable() : void	private
	sendMessage(message: String) : void	private
	sendMessage(send: byte[]) : void	private
	setStatus(resId: int) : void	private
	setStatus(subTitle: CharSequence) : void	private
	newBoard(gameModel : GameModel ,handler : Handler) : void	private
	onActivityResult(requestCode: int,resultCode: int, data: Intent) : void	public
	connectDevice(data: Intent, secure ; boolean) : void	private
	onCreateOptionsMenu(menu :Menu) : Boolean	public
	onOptionsItemSelected(item:MenuItem) : Boolean	public

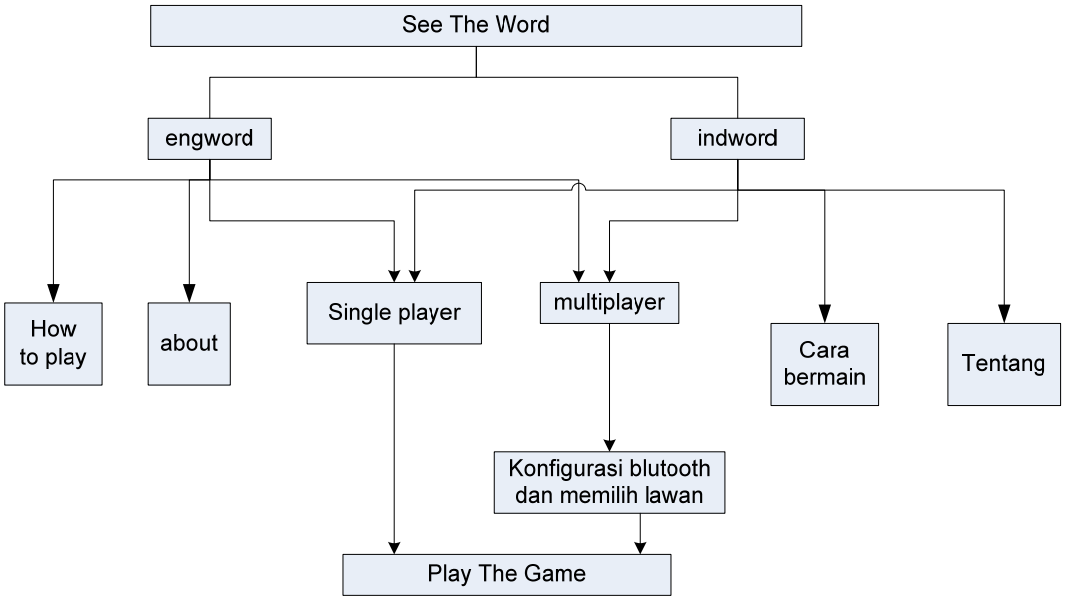
3.2 Perancangan Sistem

Tahap perancangan merupakan kelanjutan dari proses analisis. Hal ini dilakukan untuk mengatasi kekurangan yang ada.

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

3.2.1 Perancangan Struktur Menu

Dalam membangun sistem diperlukan perancangan struktur menu.Perancangan struktur menu dibuat dengan tujuan untuk mempermudah dalam merancang setiap bagian dari sistem yang akan dibangun, selain itu perancangan ini juga berguna untuk memudahkan *user* dalam melakukan navigasi, guna mengakses fasilitas yang terdapat dalam aplikasi *game* yang akan dibangun. Gambar di bawah ini merupakan perancangan struktur menu pada aplikasi *game* "*See the word*" yang akan dibangun.



Gambar 3.34 Struktur Menu

1. *Engword*

   Merupakan menu untuk permainan dengan menggunakan bahasa Inggris dan akan masuk ke menu selanjutnya untuk memilih mode permainan.

2. *Indword*

   Merupakan menu untuk permainan dengan menggunakan bahasaIndonesia dan akan masuk ke menu selanjutnya untuk memilih mode permainan.

3. *How to play /* Cara Bermaian

   Merupakan menuyang berfungsi untuk menampilkan informasi mengenai petunjuk cara menggunakan *game* yang akan dibangun.

4. *About /* Tentang

   Merupakan menu yang berfungsi untuk menampilkan informasi mengenai pengembang aplikasi*game* yang dibuat.

5. *single player*

   Merupakan submenu dari *Indword* maupun *Engword,* yang fungsinya untuk menampilkan arena permainan dalm mode *single player*.

6. *Multiplayer*

   Merupakan submenu dari Ind*word* maupun eng*word,* yang fungsinya untuk menuju ke sub menu berikutnya sebelum menampilkan arena permainan dengan mode *multiplayer* dengan.

7. Konfigurasi Dan Memilih lawan

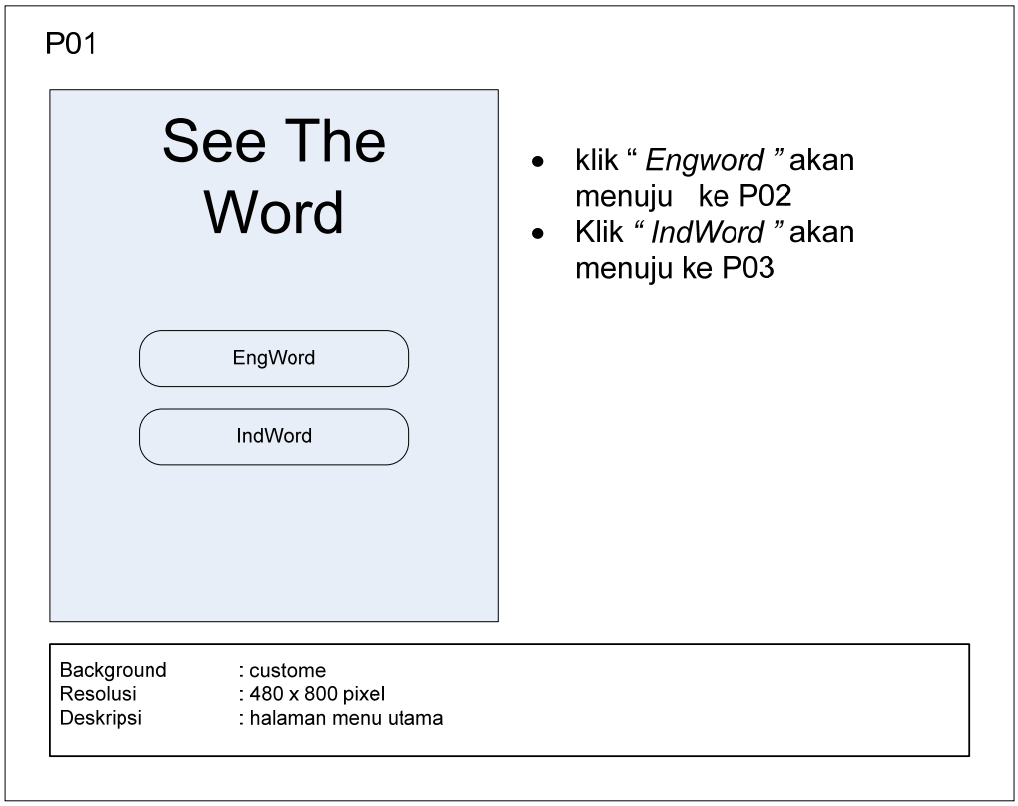
   Merupakan submenu dari *multiplayer* yang fungsinya untuk pengaturan*bluetooth* dan menyabungkan antar perangkat yang akan terlibat dalam permainan ini.

8. *Play the Game*

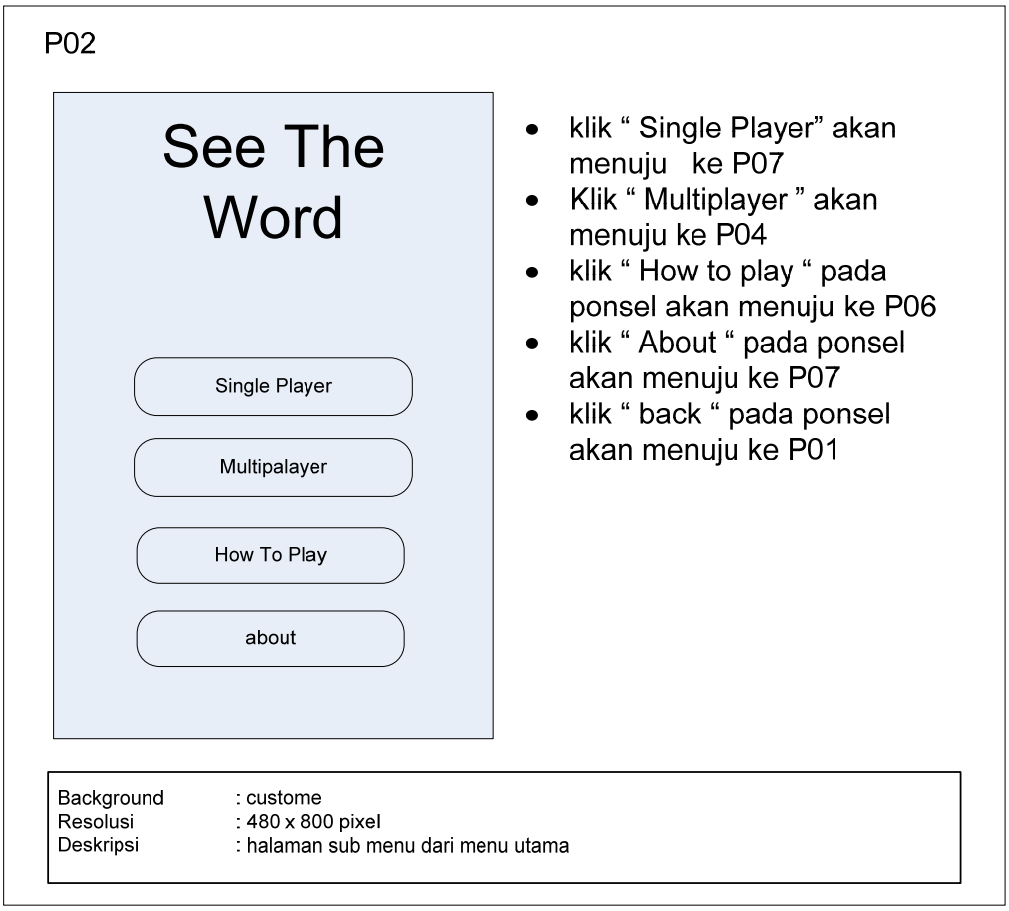
   Merupakan tampilan arena permaianan.

3.2.2 Perancangan Antar Muka

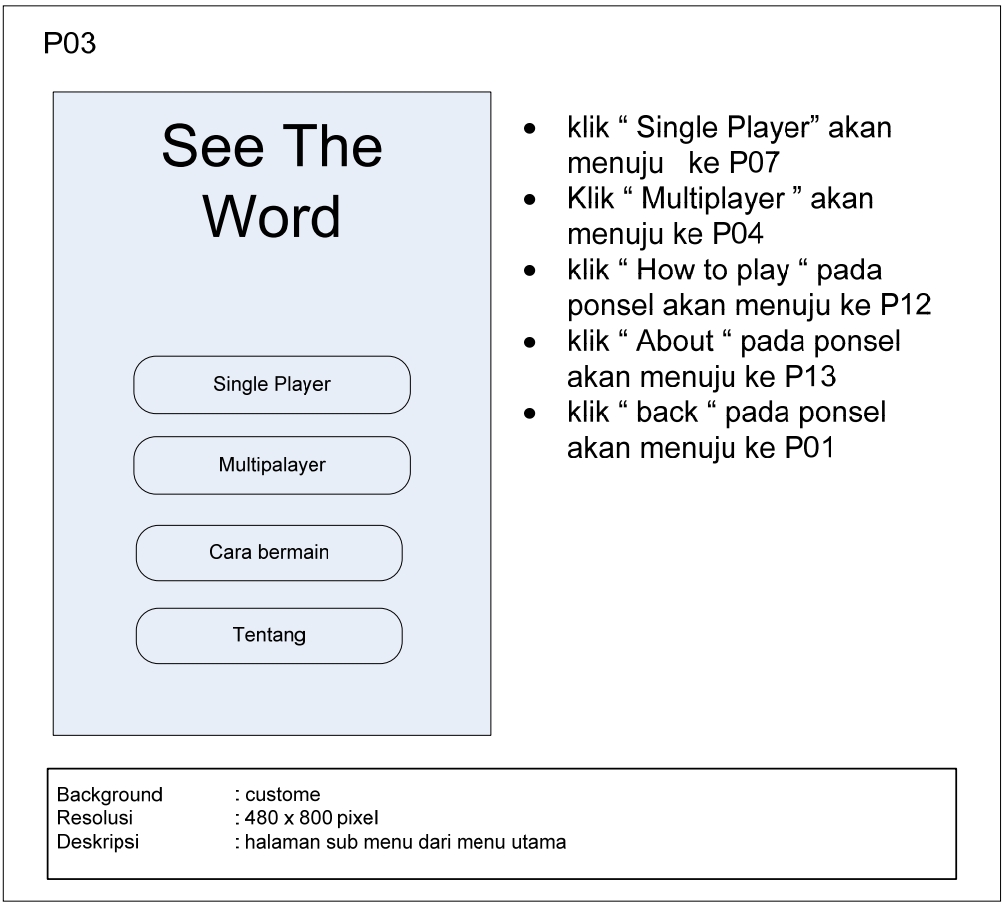
Perancangan antar muka merupakan gambaran yang dapat digunakan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut adalah perancangan antar muka dari *game* "*See the word*"



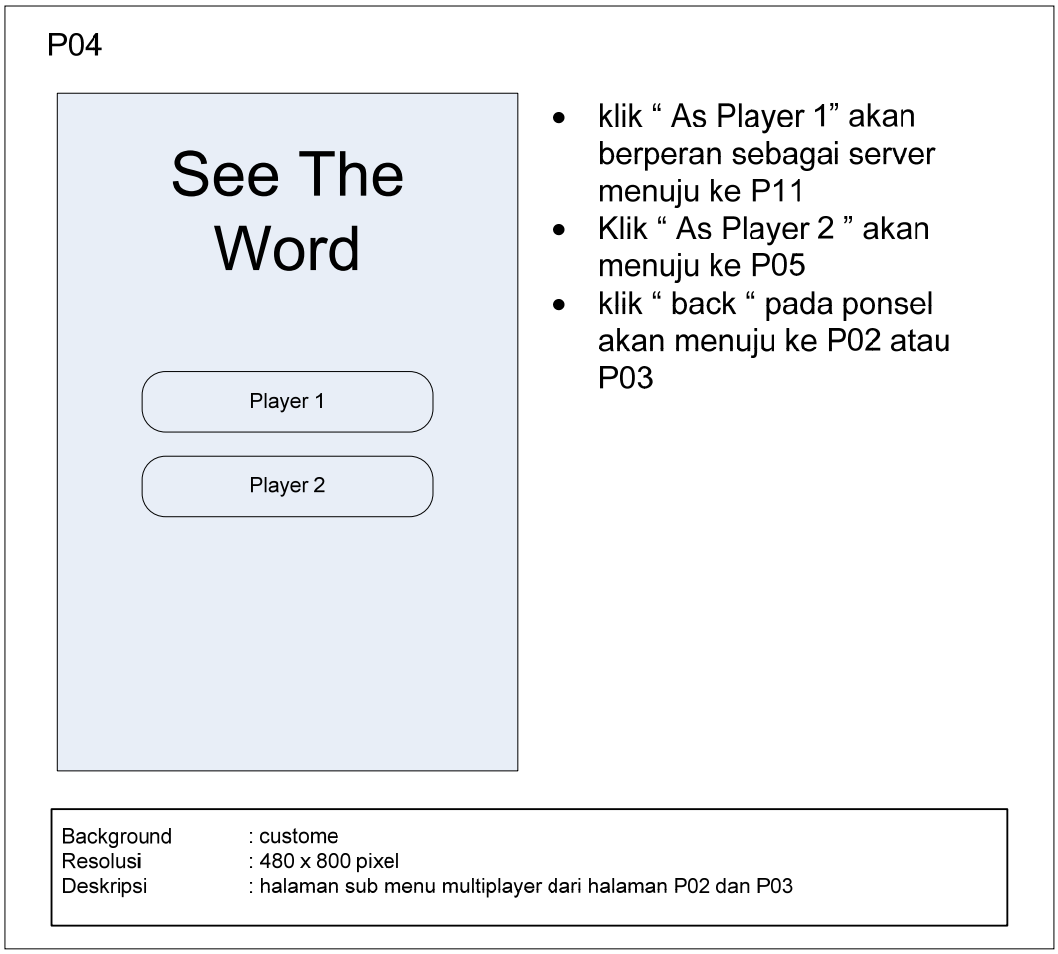
Gambar 3.35 Perancangan Antar Muka Menu Utama



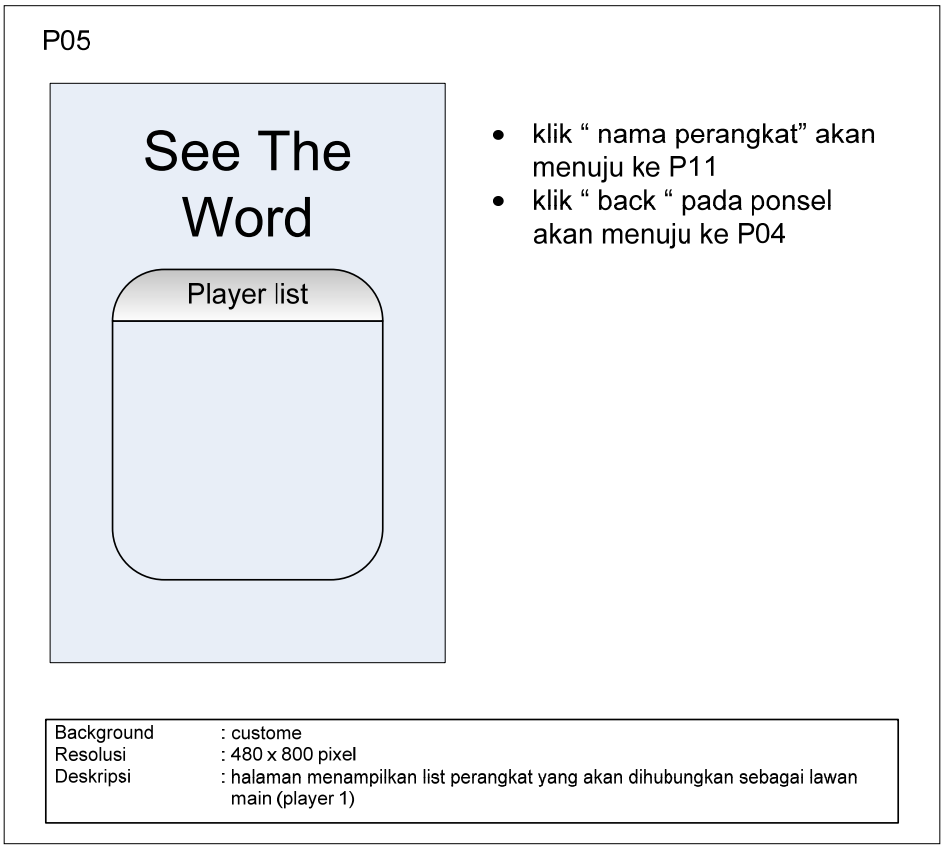
Gambar 3.36 Perancangan Antar Muka Submenu *Engword*



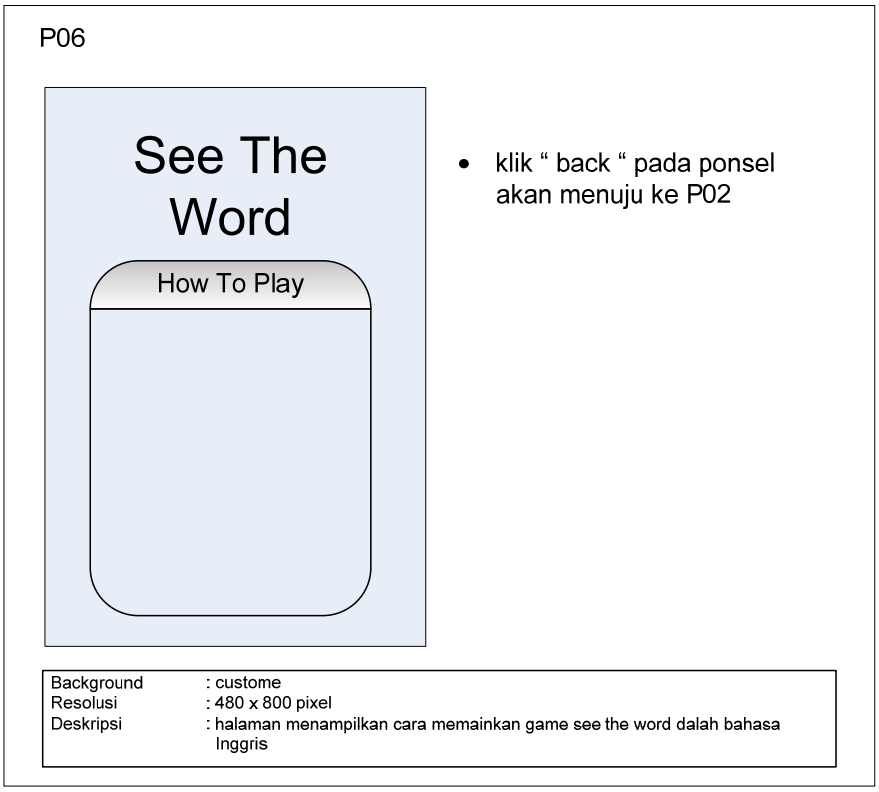
Gambar 3.37 Perancangan Antar Muka Submenu *Indword*



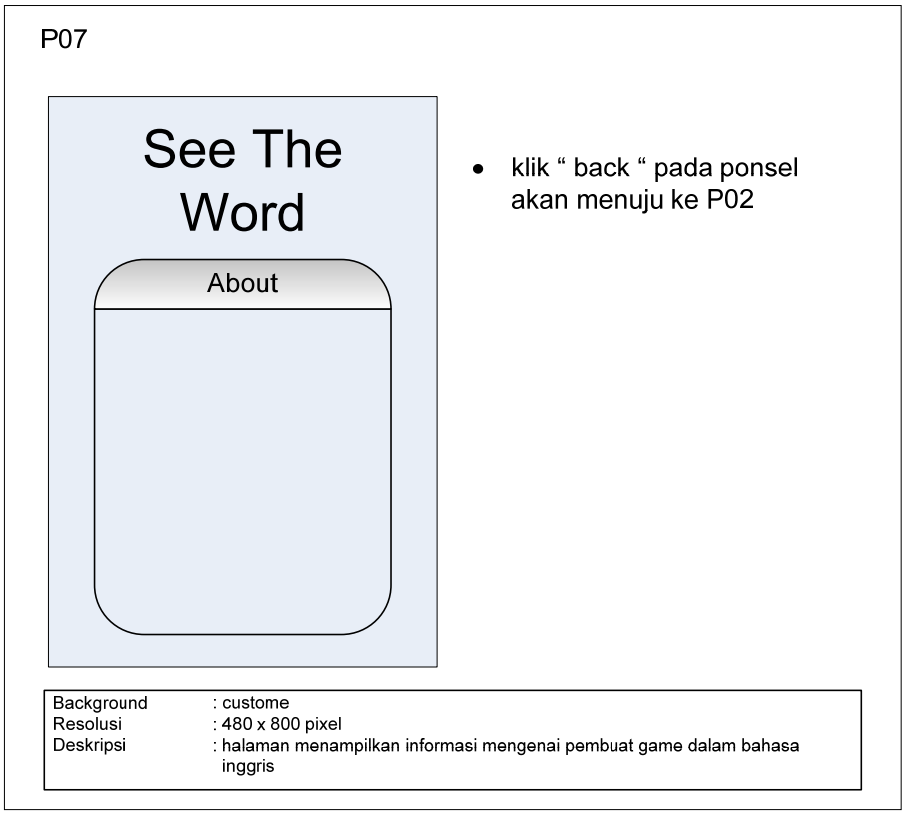
Gambar 3.38 Perancangan Antar Muka submenu *Multiplayer*



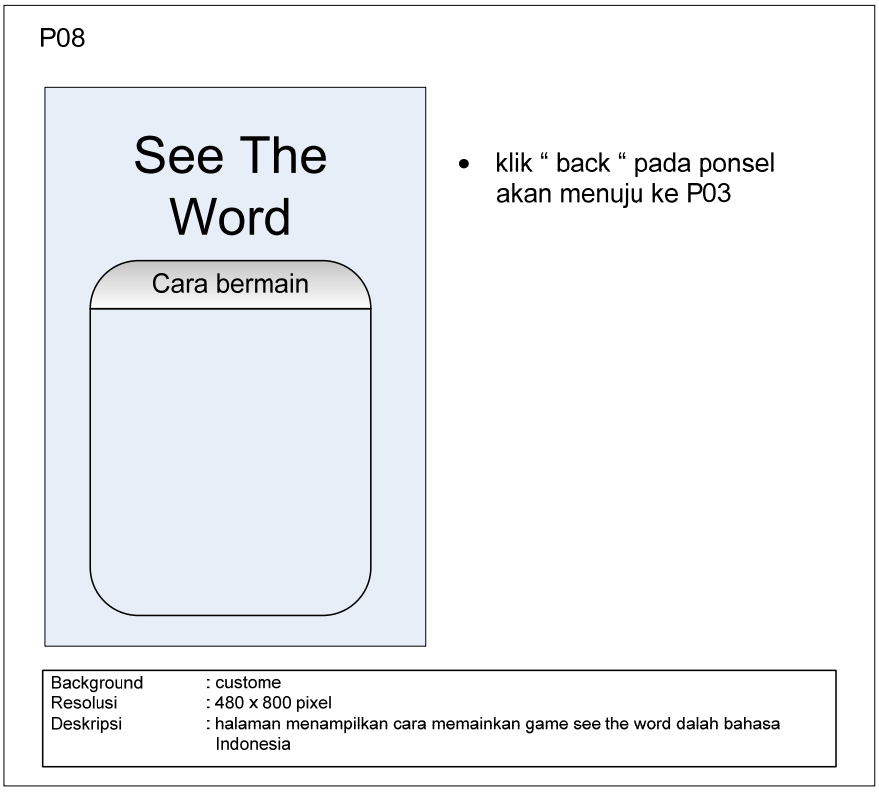
Gambar 3.39 Perancangan Antar Muka Menu pengaturan*Bluetooth player* 2



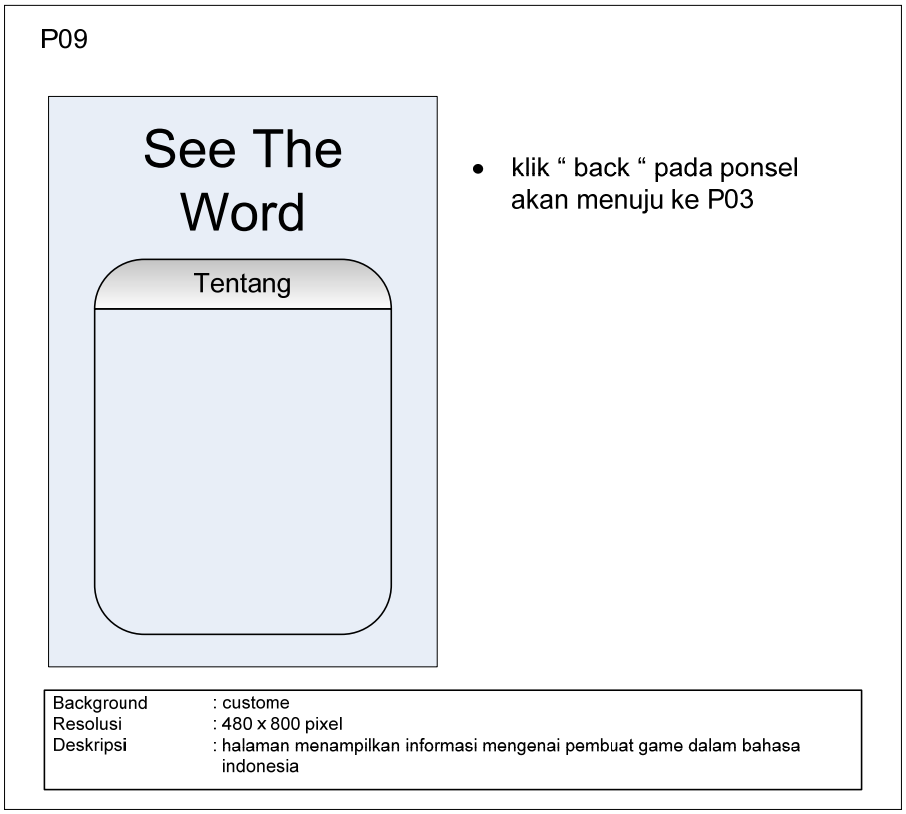
Gambar 3.40 Perancangan Antar muka halaman *How to Play*



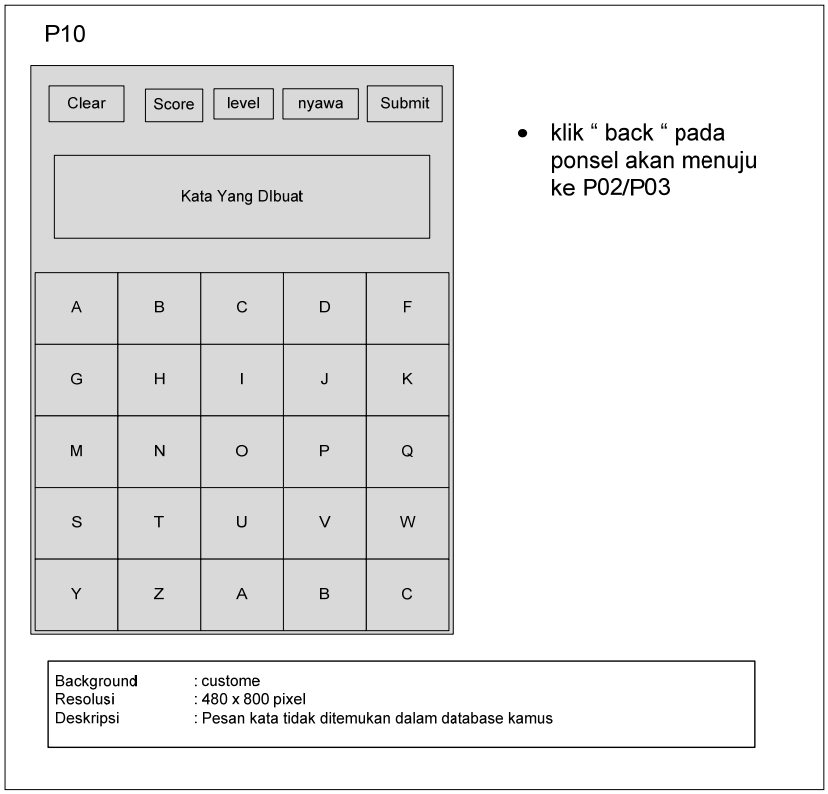
Gambar 3.41 Perancangan Antar Muka Halaman About



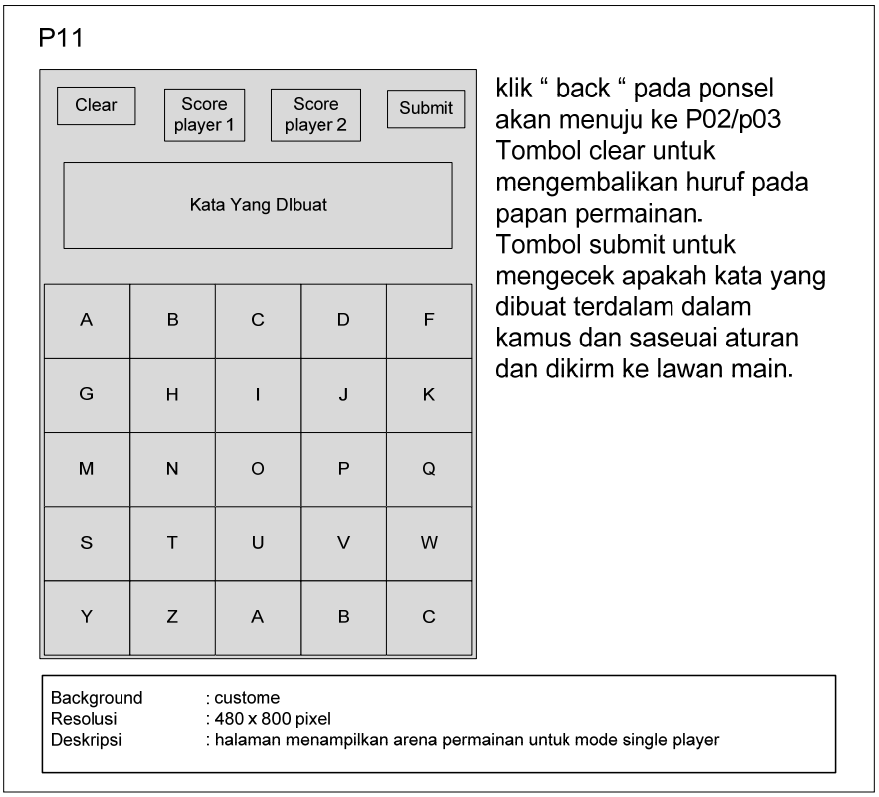
Gambar 3.42 Perancangan Antar Muka Halaman cara Bermain



Gambar 3.43 Perancangan Antar Muka Halaman Tentang



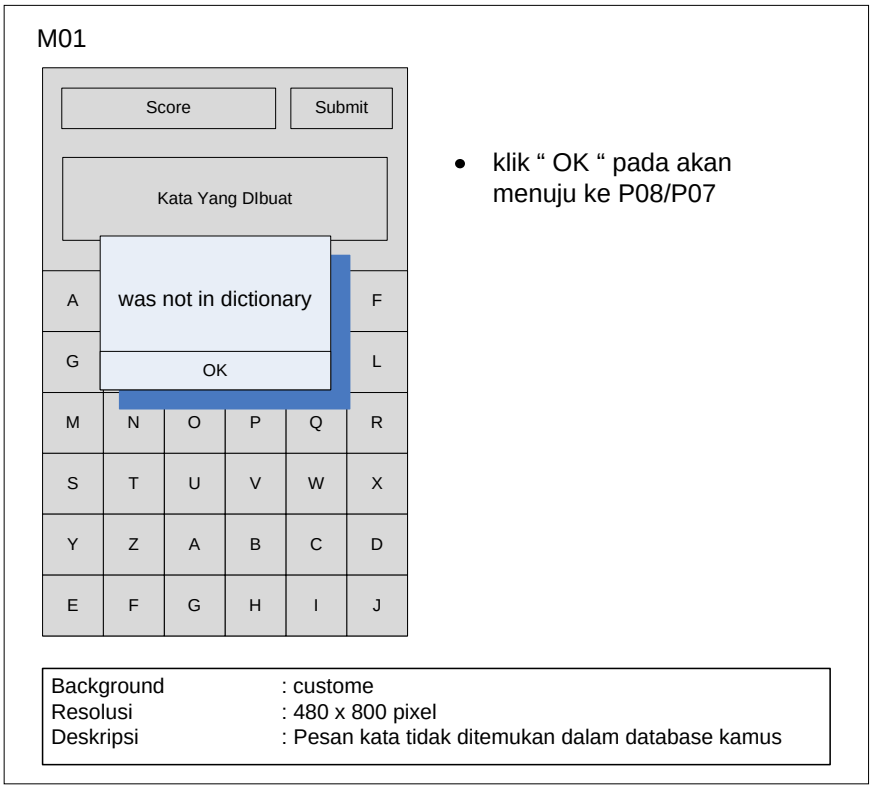
Gambar 3.44 Perancangan Antar Muka Halaman Permaina *Single Player*



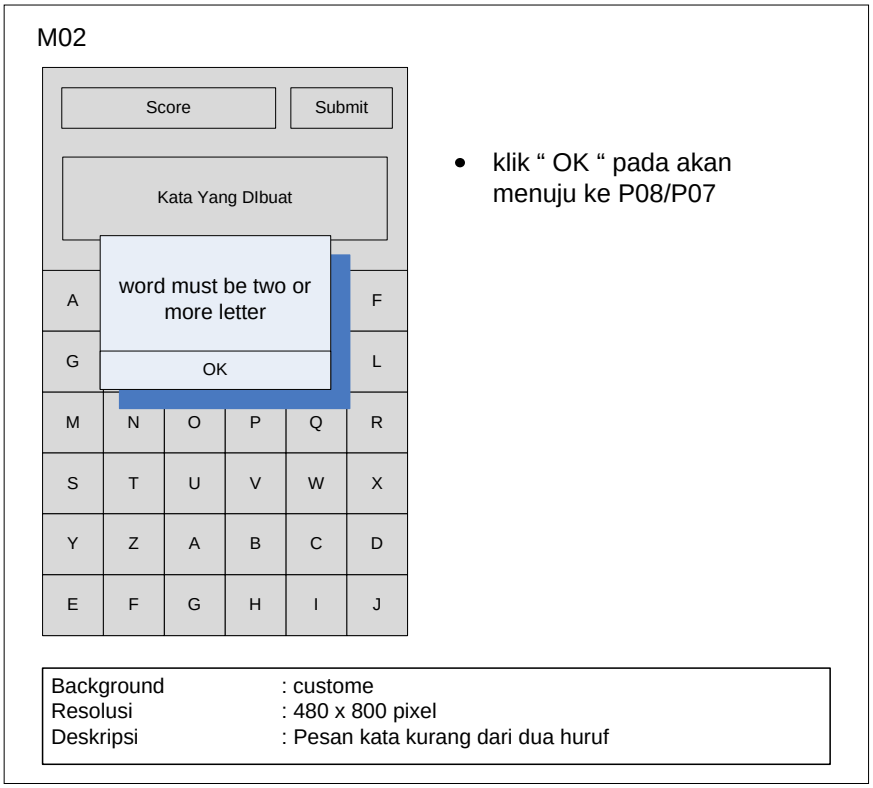
Gambar 3.45 Perancangan Antar Muka Halaman Permainan *Multiplayer*

3.2.3 Perancangan Pesan

Perancangan pesan adalah rancangan antar muka dari pesan yang akan ditampilkan apabila terjadi kesalahan dan pemberitahuan dalam proses yang dilakukan. Rancangan pesan dari game ini digambarkan sebagai berikut.

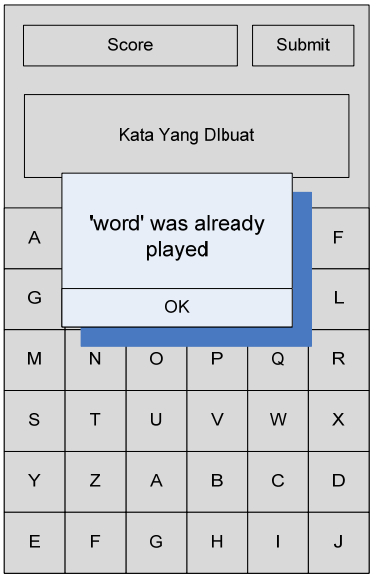


Gambar 3.46 Pesan Kata Tidak Terdapat Dalam Kamus



Gambar 3.47 Pesan panjang kata tidak seusai aturan

M03

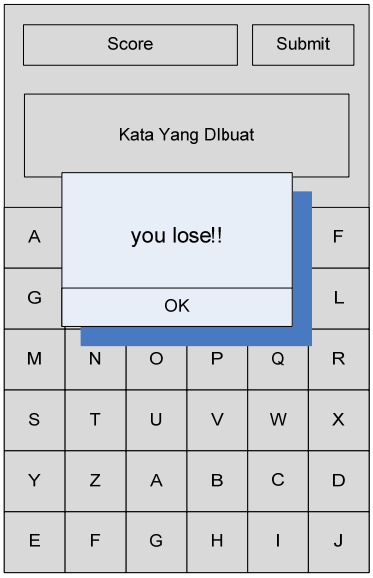


- klik " OK " pada akan menuju ke P08/P07

Background	: custome
Resolusi	: 480 x 800 pixel
Deskripsi	: Pesan kata telah dibuat sebelumnya

Gambar 3.48 Pesan Kata Sudah Digunakan Sebelumnya

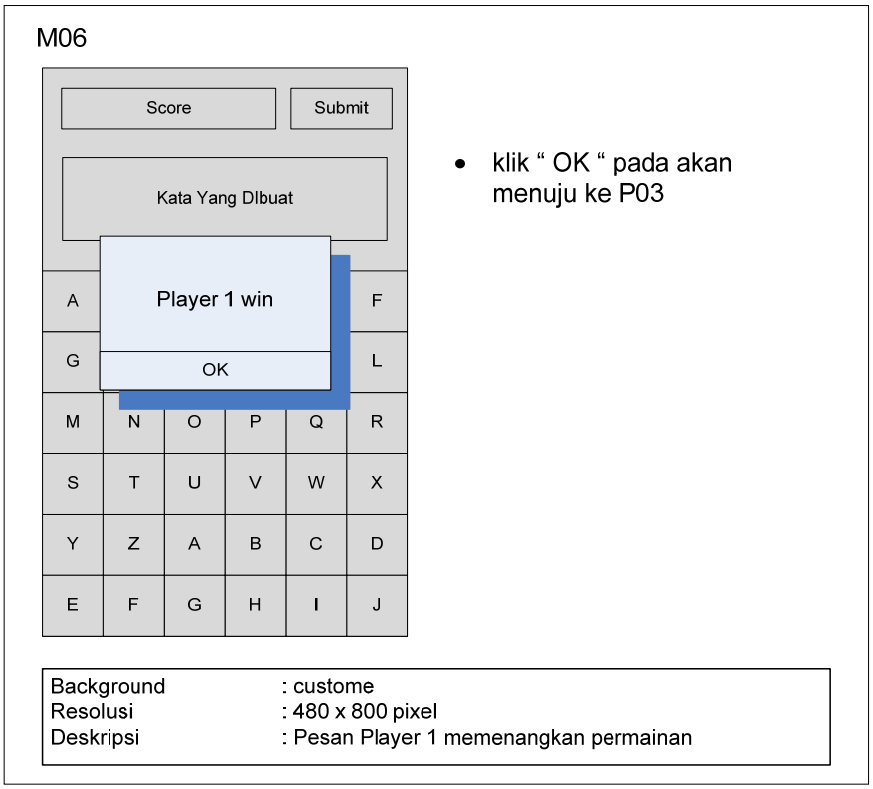
M04



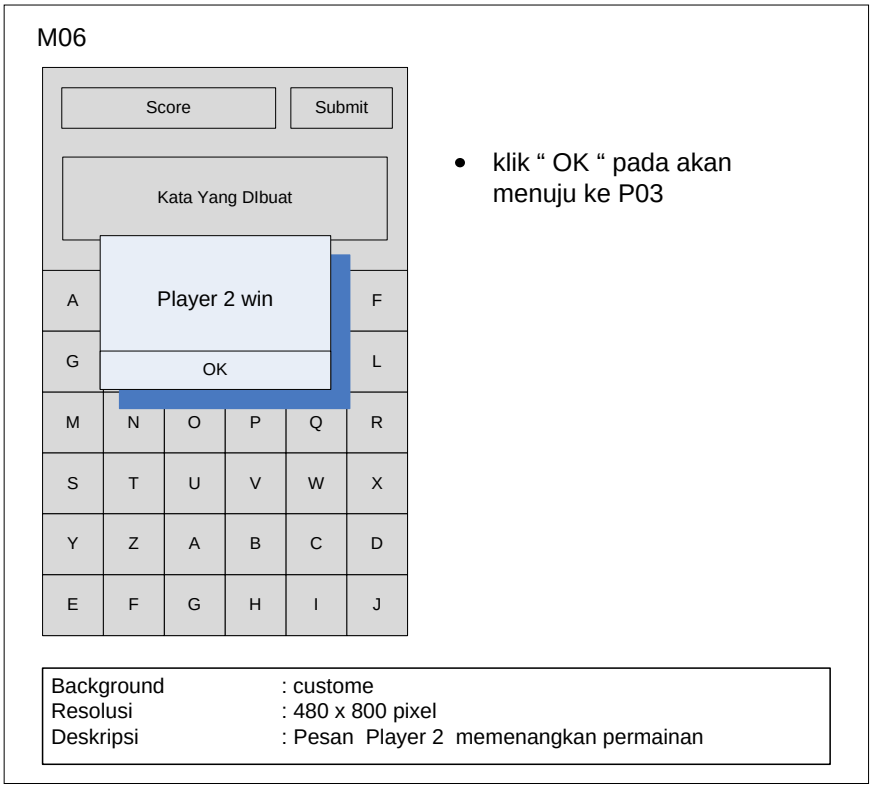
- klik " OK " pada akan menuju ke P02

Background	: custome
Resolusi	: 480 x 800 pixel
Deskripsi	: Pesan saat nyawa telah habis

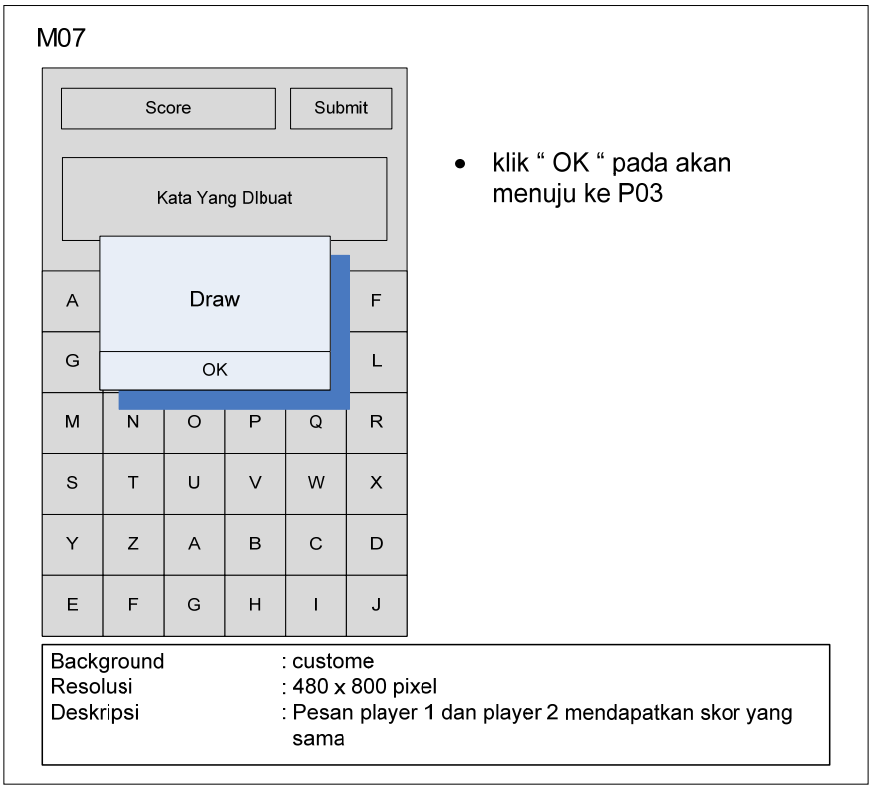
Gambar 3.49 Pesan Nyawa Telah Habis



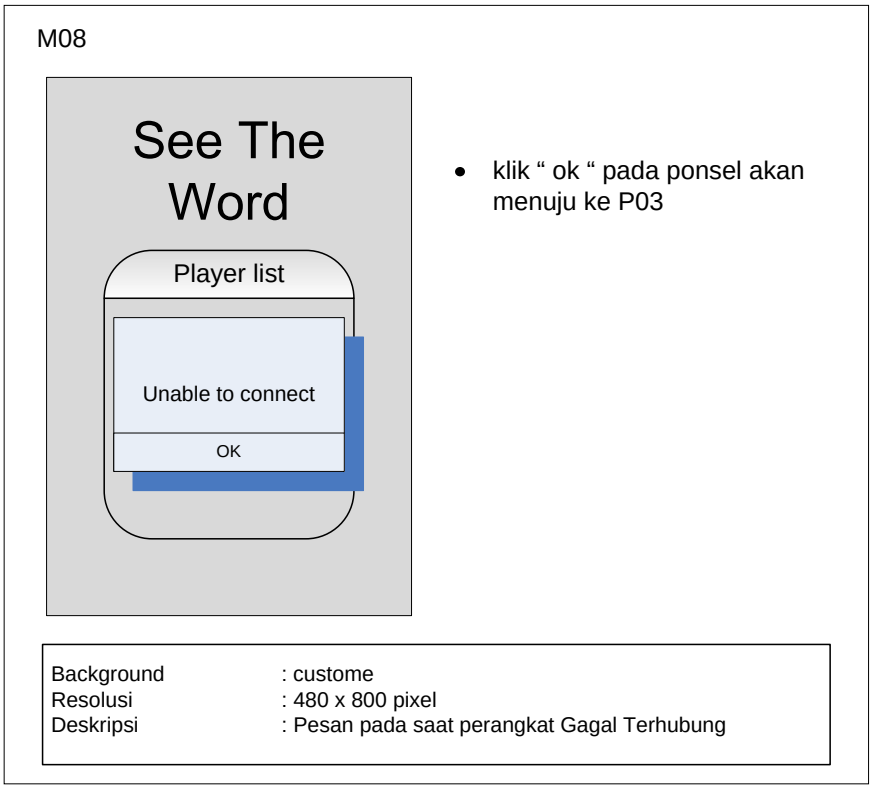
Gambar 3.50 Pesan *Player* 1 memenangkan Permainan



Gambar 3.51 Pesan *Player 2* Memenangkan Permainan



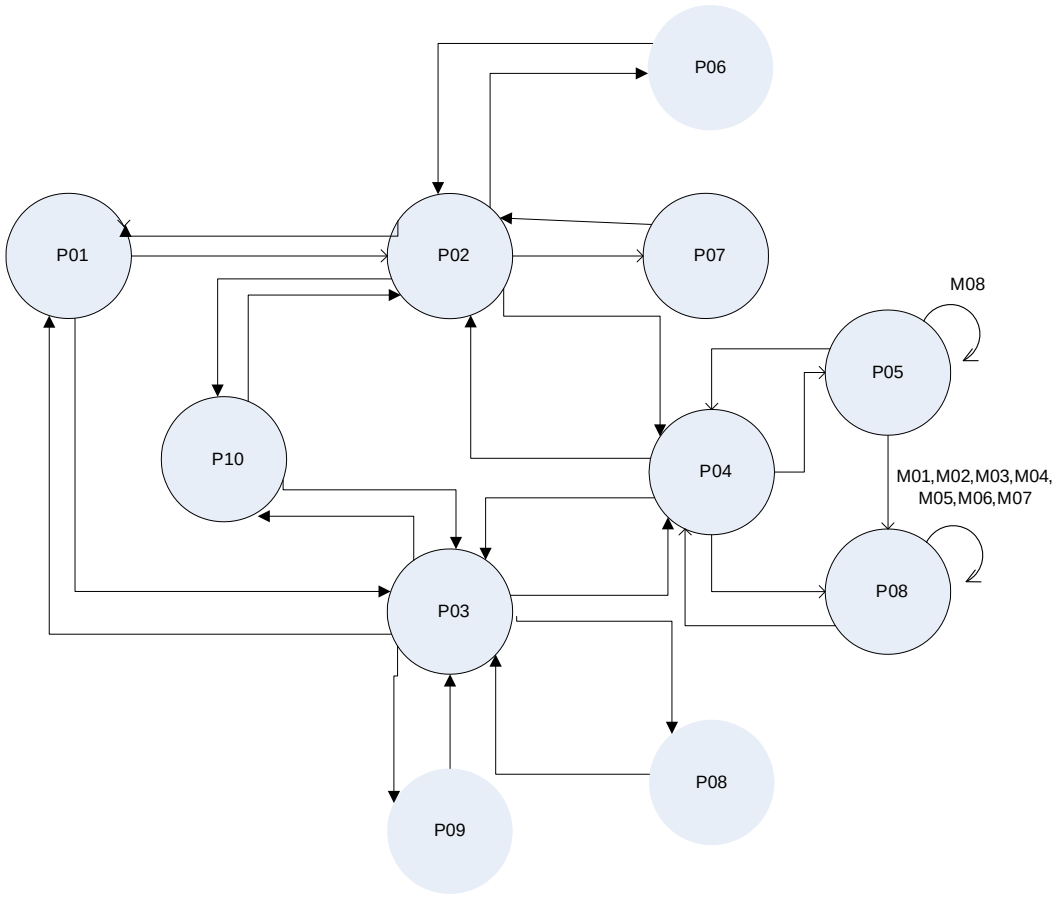
Gambar 3.52 Pesan *Draw*



Gambar 3.53 Pesan Perangkat *Player* 2 Gagal terhubung Dengan Perangkat *Player* 1

3.2.4 Jaringan Semantik

Jaringan semantik merupakan jaringan data dan informasi, yang menunjukan hubungan antar berbagai objek dimana informasi yang terhubung tersebut adalah informasi yang proporsional (suatu pernyataan yang dapat bernilai benar atau salah). Struktur jaringan semantik terdiri dari *node* atau simpul dan busur atau *arc* yang menghubungkannya. Simpul menyatakan objek sedangkan busur menyatakan*links*.*Links* dari jaringan semantik digunakan untuk menunjukan hubungan *(relationship)* antar simpul-simpul tersebut.



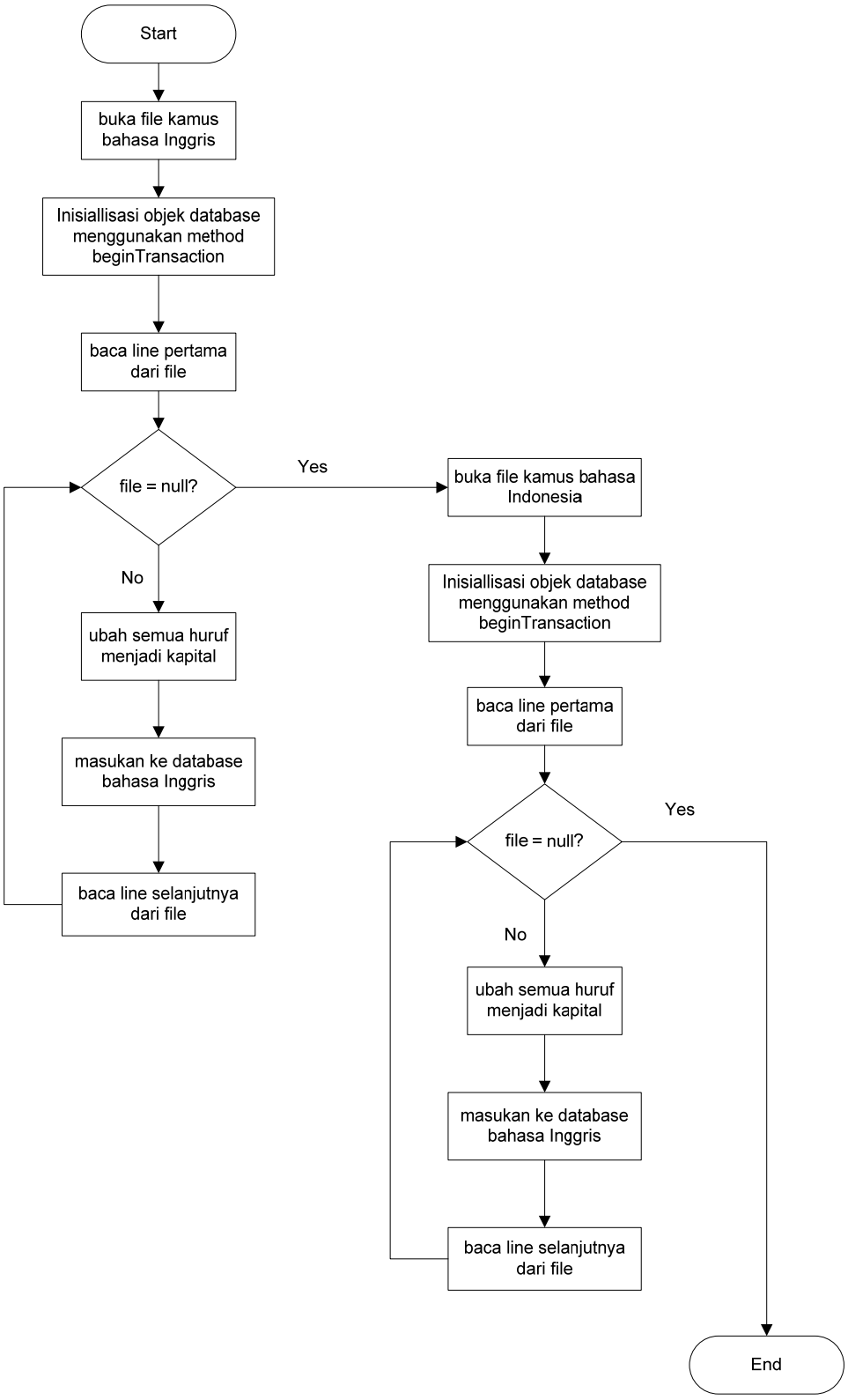
Gambar 3.54 Jaringan Semantik

3.2.5 Perancangan method

Perancangan metode menjelaskan tentang bagaimana program aplikasi berjalan.Perancangan method digambarkan dengan menggunakan flow chart.

3.2.5.1 *Pemuatan Database*

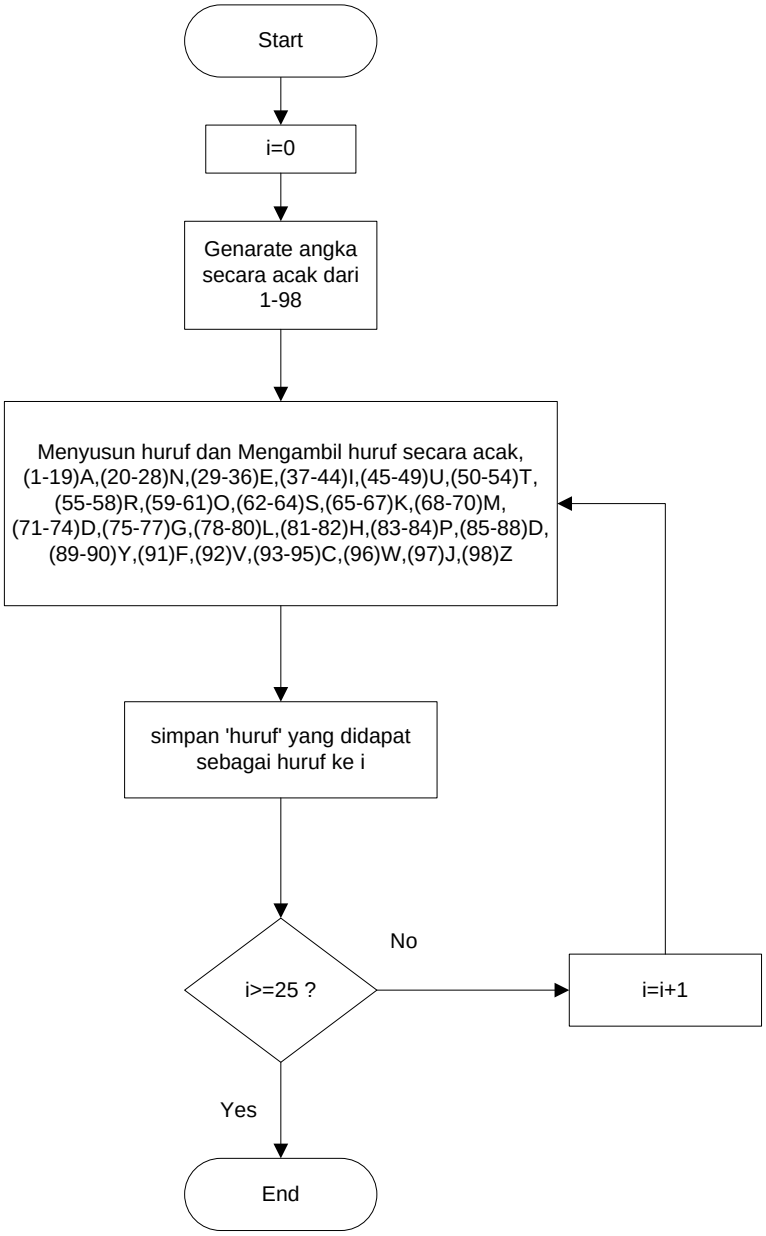
Pemuatan Databse merupakan proses pembuatan database kamus bahasa Indonesia dan bahasa Inggris. Proses ini akan dijelaskan dalam bentuk flow chart di bawah ini.



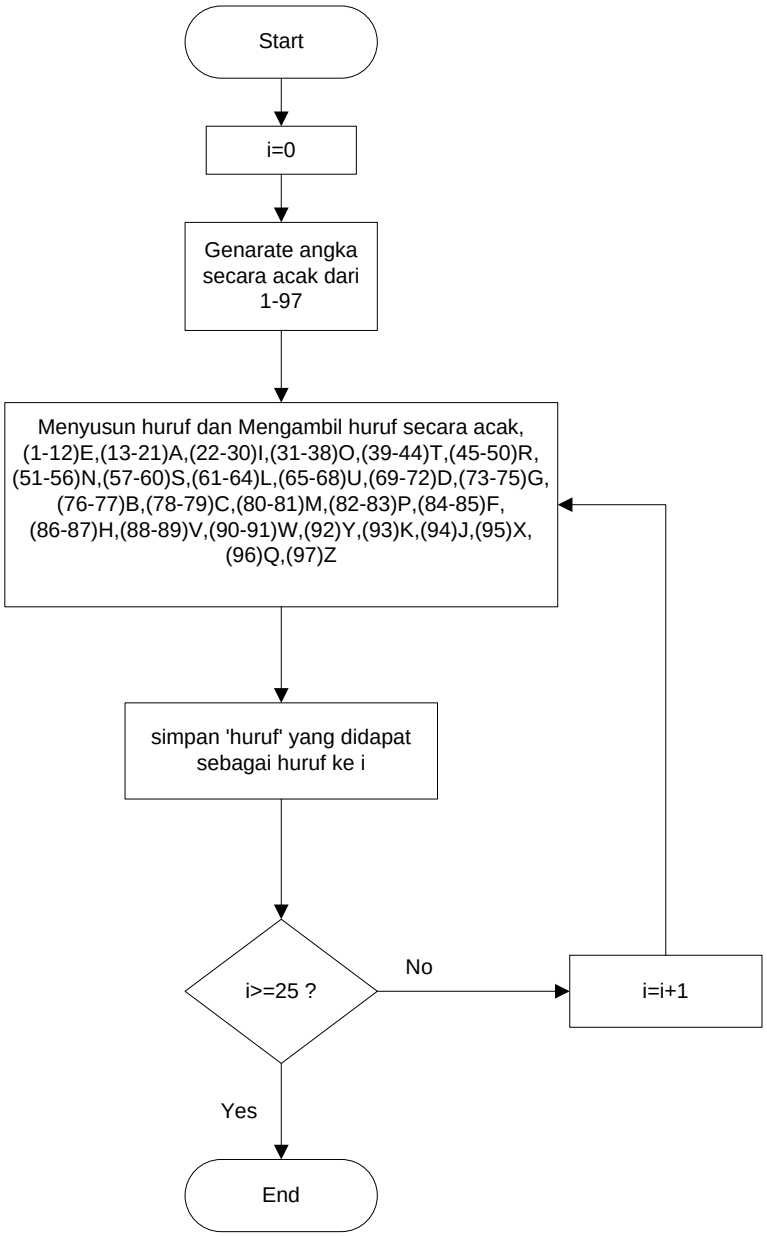
Gambar 3.55 Flowchart Method Pemuatan DataBase

3.2.5.2 Pendistribusianhuruf

Pendistribusian huruf merupakan proses penyedian dan pengacakan huruf secara random dengan aturan pendistribusian disesuaikan dengan permainan *scrabble*. Proses ini akan diperlihatkan dalam bentuk flowchart.



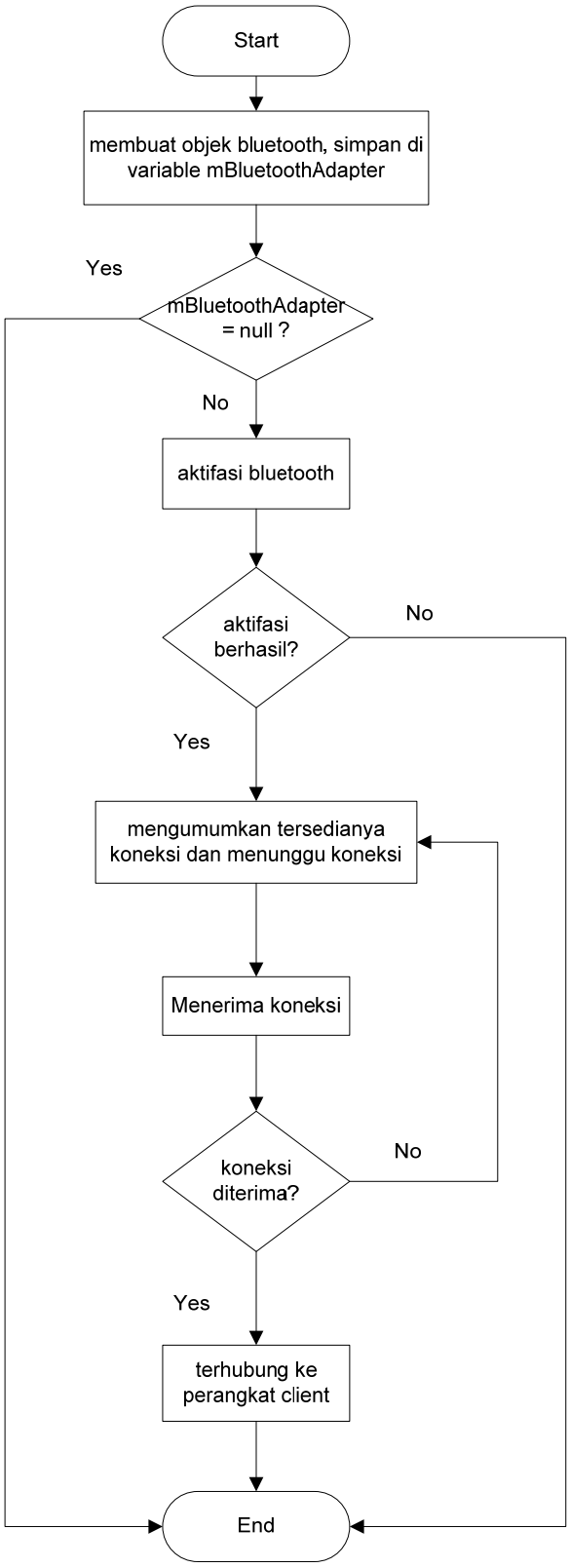
Gambar 3.56 Flowchart Method Pendistribuan Huruf Bahasa Indonesia



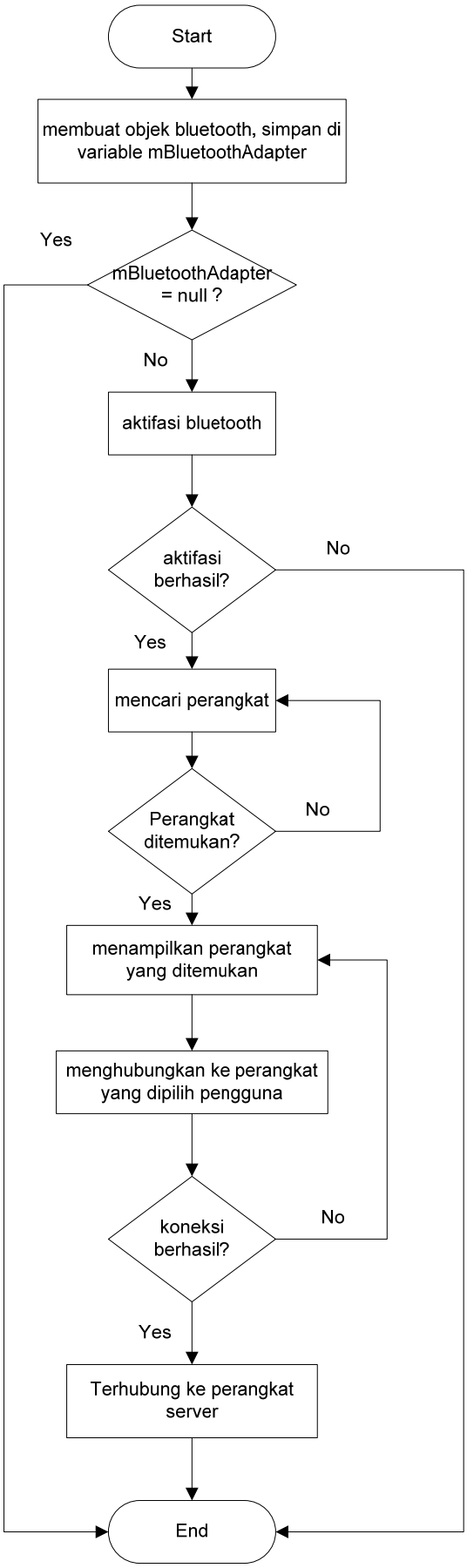
Gambar 3.57 Flowchart Method Pendistribuan Huruf Bahasa Inggris

3.2.5.3 Pengaturan *Bluetooth*

Pengaturan *Bluetooth* merupakan proses menghubungkan dua perangkat dalam mode permainan *multiplayer* yang satu bertindak sebgai *server* dan yang satunya lagi bertindak sebagai *client*. Proses ini akan diperlihatkan dalam bentuk flowchart.



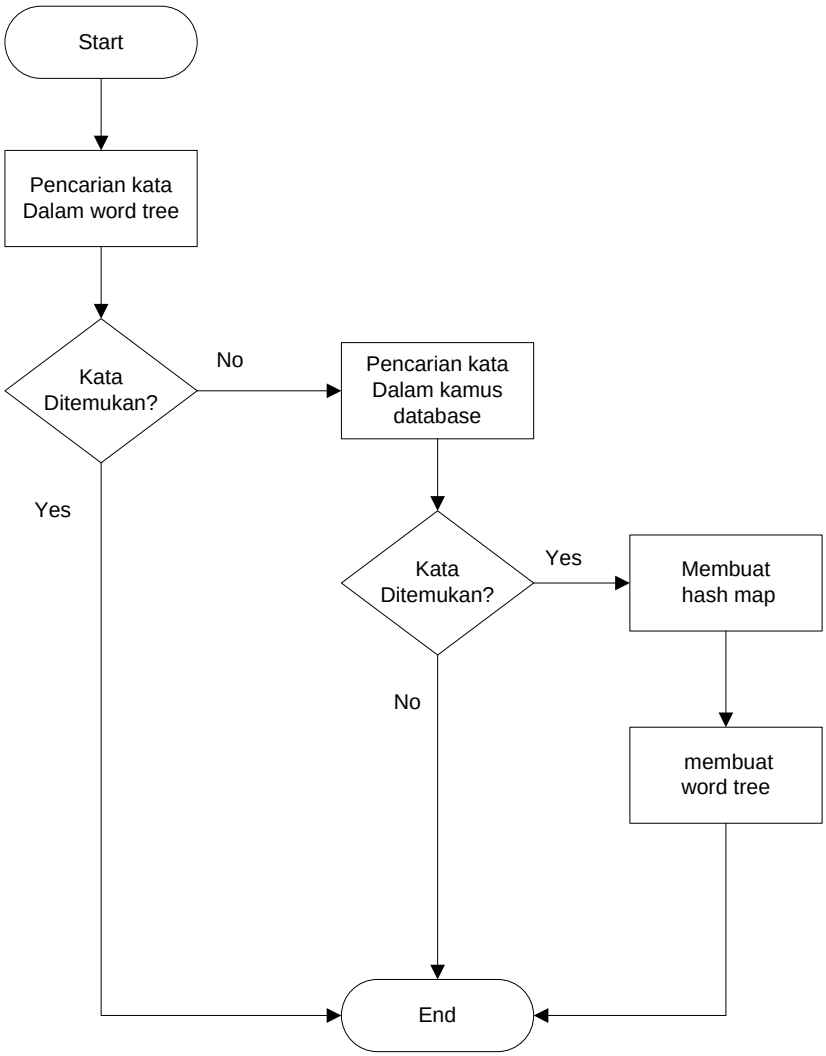
Gambar 3.58 Flowchart Methode Pengaturan *Bluetooth Player 1*


Start
membuat objek bluetooth, simpan di variable mBluetoothAdapter
Yes
mBluetoothAdapter = null ?
No
aktifasi bluetooth
aktifasi berhasil?
No
Yes
mencari perangkat
Perangkat ditemukan?
No
Yes
menampilkan perangkat yang ditemukan
menghubungkan ke perangkat yang dipilih pengguna
koneksi berhasil?
No
Yes
Terhubung ke perangkat server
End

Gambar 3.59 Flowchart Methode Pengaturan *Bluetooth Player 2*

3.2.5.4 Pencarian Kata Dalam *Database* Kamus

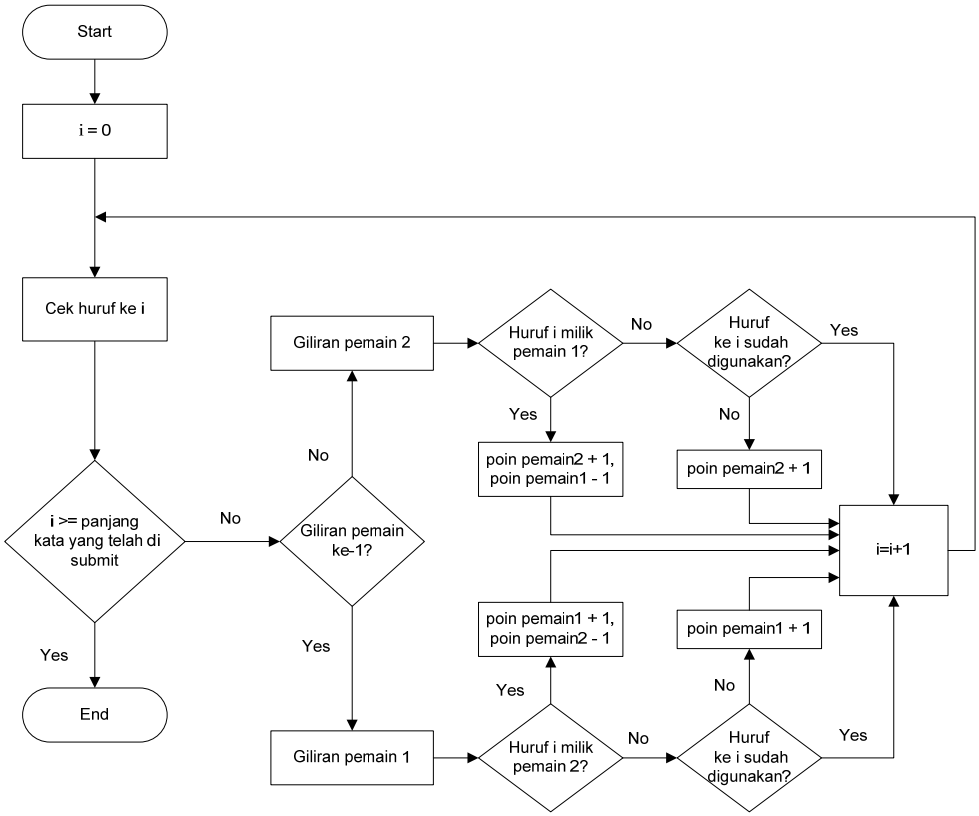
Ini merupakan proses pencarian suatu kata dalam kamus data base. Proses ini akan diperlihatkan dalam bentuk flowchart.



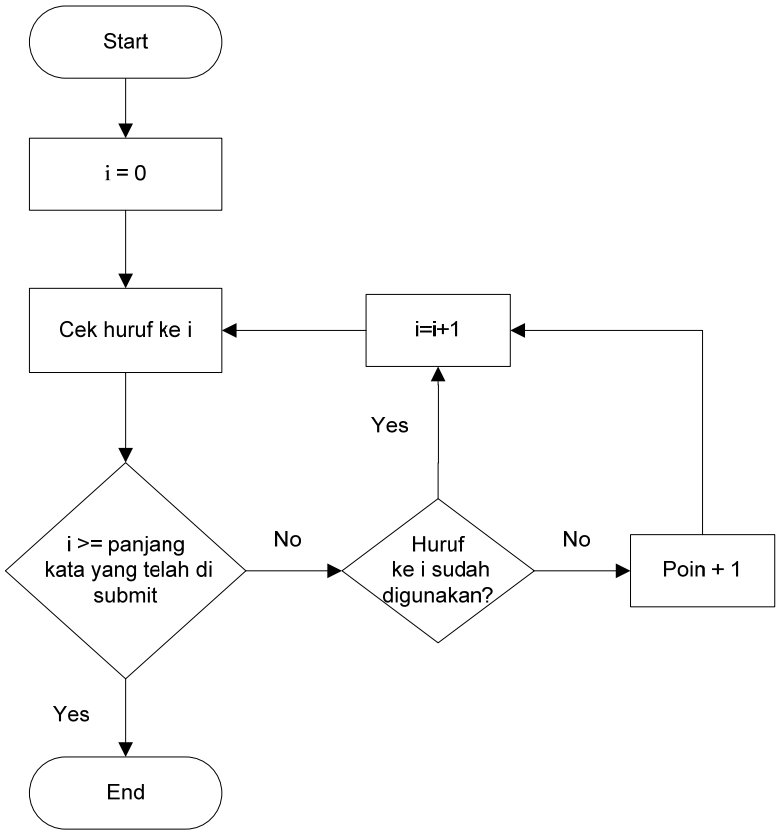
Gambar 3.60 Flowchart Methode Pencarian dalam Data Base

3.2.5.5 Penghitungan skor

Penghitungan skor merupakan proses penghitungan point yang didapat dari tiap kata yang telah disubmit. Proses ini akan digambar dalam bentuk flowchart.



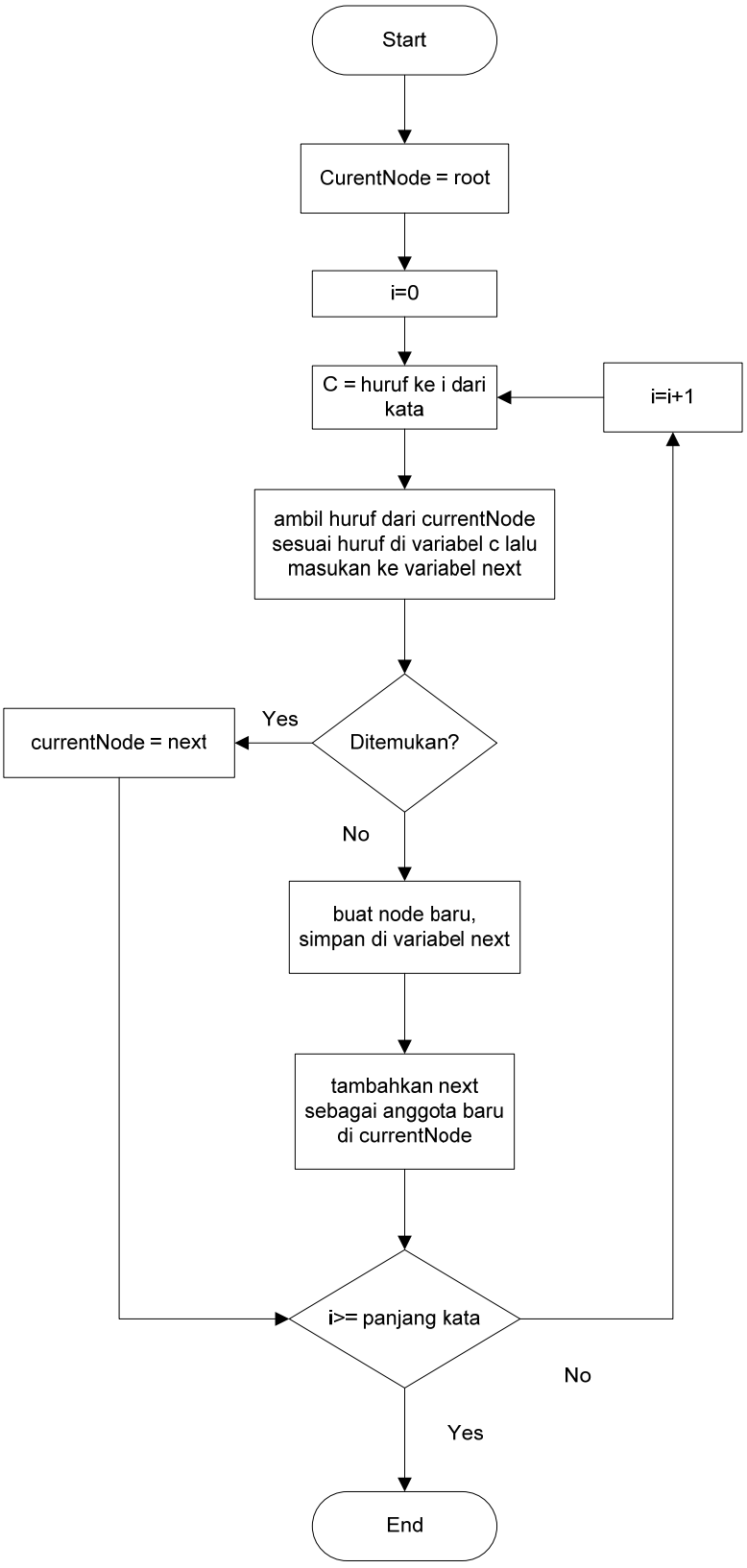
Gambar 3.61 Flowchart Methode Penghitung Skor Permainan *Multiplayer*



Gambar 3.62 Flowchart Methode Penghitung skor permainan *single player*

3.2.5.6 Pembuatan WordTree

WordTree merupakan kamus yang dibuat dalam bentuk tree kamus ini akan digunakan untuk kata-kata yang telah digunakan dalam permainan sehingga pemain tidak dapat mengulang kata yang sebelumnya sudah digunakan. Proses ini dapat dilihat pada gambar 3.60.



Gambar 3.63 Flowchart Methode *Build Word*