

APLIKASI PERMAINAN OTHELLO DENGAN BAHASA PEMROGRAMAN PYTHON

Rori Setiawan
L2F301470

Informatika dan Komputer
Jurusan Teknik Elektro
Fakultas Teknik Universitas Diponegoro

Abstrak

Aplikasi permainan othello merupakan sebuah permainan sederhana yang dapat dijumpai pada perangkat-perangkat seperti handphone, atau game console. Permainan othello menggunakan sebuah papan yang dibagi menjadi petak-petak kecil dan bidak-bidak permainan berupa biji-biji berwarna hitam dan putih. Dalam tugas akhir ini bertujuan untuk menghadirkan sebuah aplikasi permainan othello yang dapat dimainkan untuk melawan komputer, juga dapat digunakan untuk melawan pemain lain yang terhubung pada jaringan komputer.

Untuk mendukung pembuatan aplikasi permainan othello ini perlu dilakukan analisa kebutuhan aplikasi dengan menggunakan diagram use case dan membuat pemodelan dengan menggunakan UML. Aplikasi permainan othello ini ditulis dengan menggunakan bahasa pemrograman python 2.4.2 untuk sistem operasi windows.

Pengujian yang dilakukan terhadap aplikasi permainan othello meliputi pengujian pada permainan lokal, permainan online dan pengujian dengan pemberian kondisi khusus. Hasil akhir dari tugas akhir ini adalah sebuah aplikasi permainan othello pada komputer yang dapat dimainkan secara lokal, maupun online.

Kata kunci:

Othello, python, jaringan komputer.

I. PENDAHULUAN

1.1 LATAR BELAKANG

Aplikasi permainan othello sudah dikenal oleh banyak orang, terutama pemilik handphone, atau game console. Permainan othello merupakan permainan catur sederhana yang merupakan modifikasi dari permainan catur Cina. Permainan othello menggunakan sebuah papan yang terbagi menjadi petak-petak kecil untuk menempatkan bidak-bidak permainan yang berupa biji-biji berwarna hitam dan putih.

Aplikasi permainan ini ditulis dengan menggunakan bahasa pemrograman python versi 2.4.2 untuk sistem operasi windows dan modul pygame versi 1.6.

1.2 MAKSUD DAN TUJUAN

Membuat aplikasi permainan othello dengan menggunakan bahasa pemrograman python.

1.3 BATASAN MASALAH

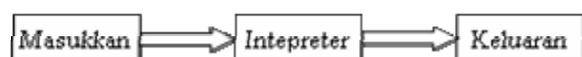
Dalam tugas akhir ini hanya membahas hal-hal sebagai berikut:

1. Menggunakan sistem operasi Ms. Windows.
2. Membuat aplikasi permainan othello dengan ukuran 10 x10 petak.
3. Menggunakan bahasa pemrograman python.
4. Tidak membahas jaringan komputer secara rinci.
5. Tidak membahas modul-modul tambahan secara rinci.

II. LANDASAN TEORI

2.1 BAHASA PEMROGRAMAN PYTHON

Bahasa pemrograman Python merupakan bahasa pemrograman yang bersifat intepreter. Gambar 2.1 menunjukkan langkah dari bahasa intepreter. Setiap baris perintah dalam bahasa python akan langsung dieksekusi tanpa melalui proses kompilasi terlebih dahulu.



Gambar 2.1 Langkah proses bahasa intepreter

2.2 PERMAINAN OTHELLO

Permainan othello merupakan permainan catur sederhana yang menggunakan papan permainan dan bidak-bidak permainan. Papan permainan terbagi menjadi petak-petak kecil dengan jumlah dan ukuran yang sama sebagai media untuk menempatkan bidak permainan. Bidak permainan berupa biji-biji berwarna hitam dan putih.

Permainan othello hanya dapat dimainkan oleh dua orang pemain saja. Setiap pemain bergerak berdasarkan giliran. Permainan othello diawali dengan menempatkan dua buah bidak hitam dan putih pada tengah-tengah papan permainan dengan posisi bidak saling bersilangan. Gambar 2.x menunjukkan posisi awal permainan othello.

Pemain hanya boleh bergerak jika dapat mengambil alih bidak lawan. Pengambil alihan bidak lawan dilakukan dengan meletakan sebuah bidak pada papan permainan dengan posisi menggapit posisi bidak lawan.

Permainan akan berakhir bila seluruh papan permainan sudah terisi penuh. Pemain dengan jumlah bidak terbanyak pada papan permainan akan menjadi pemenang permainan.

2.3 JARINGAN KOMPUTER

Jaringan komputer adalah sekumpulan komputer dengan jumlah dua komputer atau lebih yang dapat berkomunikasi satu sama lain dan dapat menggunakan peralatan seperti media penyimpanan dan pencetak (*printer*) secara bersama-sama.

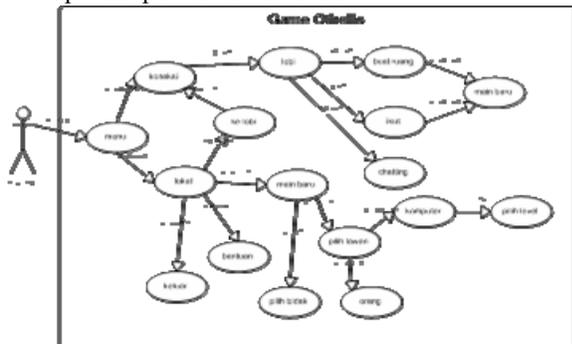
Setiap komputer dalam jaringan memiliki sebuah alamat yang unik yang akan membedakan sebuah komputer dengan yang lain. Alamat komputer ini berupa alamat fisik dan alamat logika. Alamat fisik terdapat pada kartu antarmuka jaringan yang berupa alamat MAC yang sudah ditentukan oleh produsen pada saat produksi. Alamat logika berupa alamat IP yang bersifat dinamis. Alamat IP berupa empat bilangan desimal yang dipisahkan dengan tanda titik. Alamat IP ini dapat diubah sesuai keperluan. Komunikasi antara

komputer dalam jaringan dilakukan dengan menggunakan socket. Socket merupakan gabungan/kombinasi alamat logika dan port.

III. PERANCANGAN APLIKASI PERMAINAN OTHELLO.

3.1 ANALISA KEBUTUHAN APLIKASI

Untuk mendapatkan informasi yang diperlukan dalam pembuatan aplikasi digunakan diagram use case untuk menganalisa kebutuhan aplikasi dilihat dari sisi pengguna. Gambar 3.1 menunjukkan diagram use case dari aplikasi permainan othello.



Gambar 3.1 Diagram Use case aplikasi permainan othello.

3.2 ALUR PEMROGRAMAN

Aplikasi permainan othello terdiri dari dua aplikasi terpisah yang terdiri dari aplikasi server permainan dan aplikasi klien.

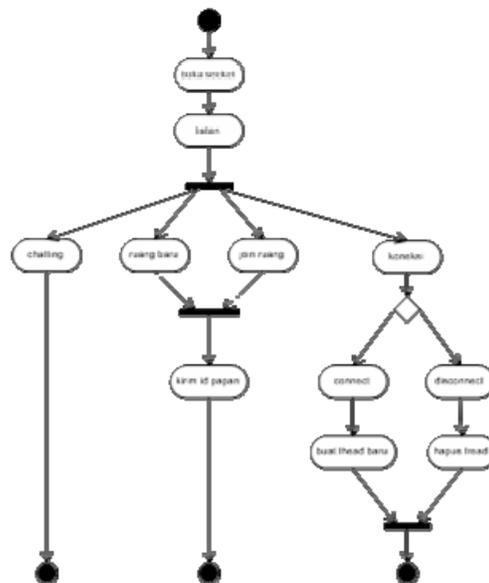
3.2.1 Aplikasi Server permainan

Aplikasi server permainan dirancang untuk selalu aktif, dan dibuat dengan berbasis teks untuk mengurangi beban proses yang dilakukan CPU. Aplikasi server permainan terdiri dari dua bagian, yaitu:

1. Server loop: berfungsi untuk menangani koneksi klien dan membuat thread baru untuk klien.
2. klien loop: berfungsi untuk hubungan antar klien yang terhubung dengan server permainan.

Aplikasi server permainan bersifat pasif, yang akan memberi jawaban bila ada permintaan dari klien. Gambar 3.2 dan 3.3 menunjukkan class diagram dan activity diagram (UML) aplikasi server permainan.

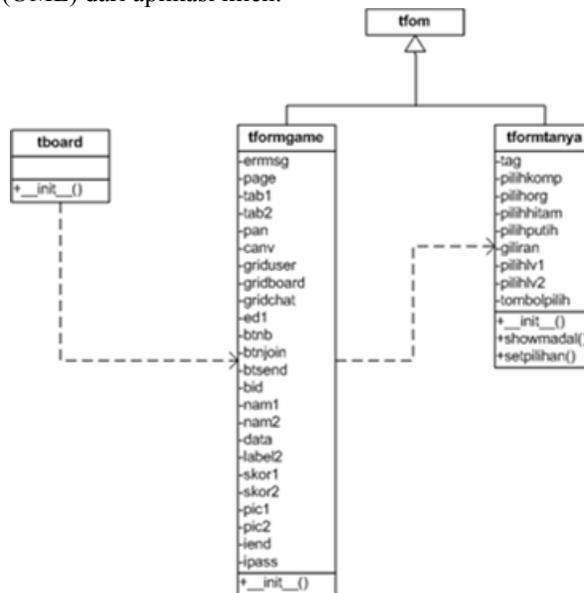
Gambar 3.2 Class diagram server permainan.



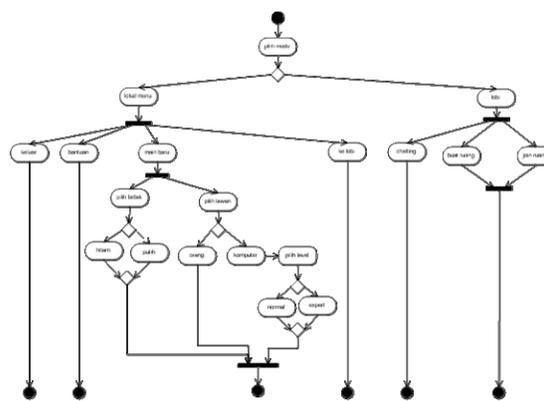
Gambar 3.3 Activity diagram server permainan.

3.2.2 Aplikasi permainan othello (klien)

Aplikasi klien menjadi inti dari aplikasi permainan othello. Aplikasi klien dibuat untuk dapat dimainkan secara *on-line* dalam jaringan komputer atau dimainkan pada komputer lokal. Gambar 3.4 dan 3.5 menunjukkan class diagram dan activity diagram (UML) dari aplikasi klien.



Gambar 3.4 Class Diagram aplikasi klien.



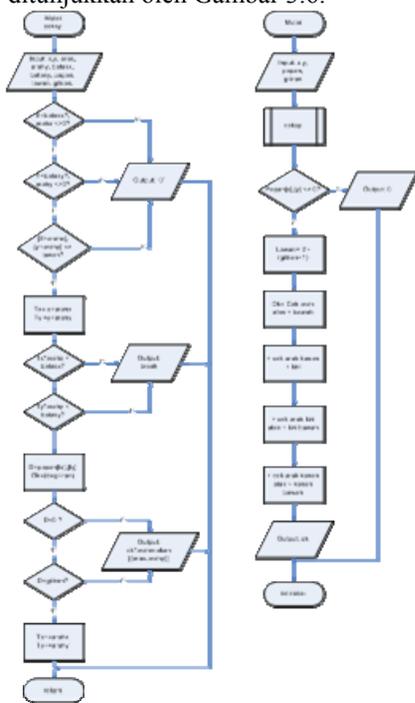
Gambar 3.5 Activity diagram aplikasi klien.

3.2.3 Algoritma program

Algoritma merupakan urutan langkah-langkah penyelesaian masalah yang disusun secara logis, ditulis

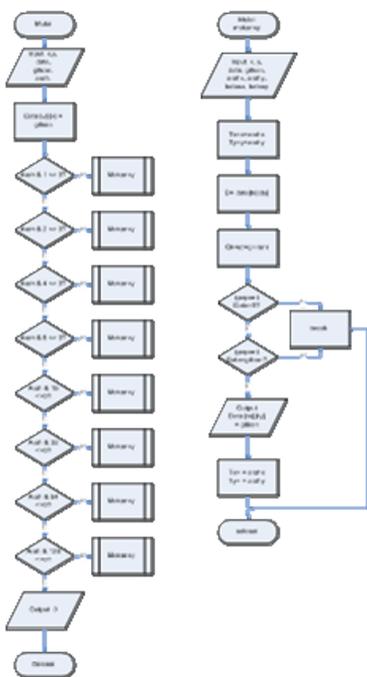
dengan notasi yang mudah dimengerti. Algoritma yang digunakan dalam aplikasi permainan othello antara lain sebagai berikut:

1. Algoritma pengecekan langkah dilakukan untuk menentukan sebuah bidak dapat ditempatkan pada petak papan permainan dan juga menentukan arah 'makan' bidak. Pengecekan ini dilakukan dengan menggunakan algoritma yang ditunjukkan oleh Gambar 3.6.



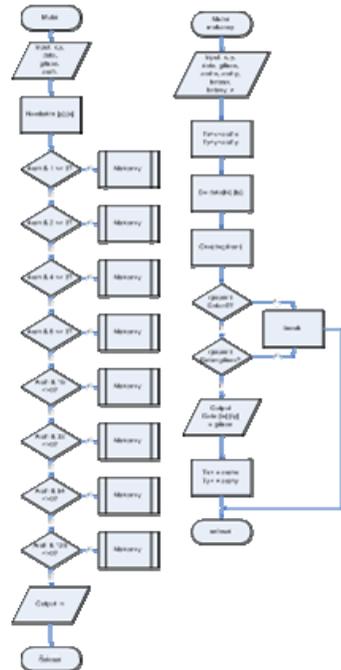
Gambar 3.6 Algoritma pengecekan langkah

2. Algoritma memakan bidak lawan: Algoritma memakan bidak lawan ini digunakan untuk dapat mengambil alih bidak lawan pada papan permainan yang didapatkan dari posisi yang dipilih dan dengan arah makan yang didapatkan dari pengecekan langkah. Gambar 3.7 menunjukkan algoritma yang digunakan untuk memakan bidak lawan.



gambar 3.7 Algoritma memakan bidak.

3. Algoritma menghitung bobot: Proses menghitung bobot hanya dilakukan pada permainan dengan komputer sebagai lawan main. Komputer akan membandingkan bobot dari setiap titik yang mungkin dapat diambil. Gambar 3.8 menunjukkan algoritma menghitung bobot.



Gambar 3.8 Algoritma menghitung bobot.

4. Algoritma tingkatan kesulitan permainan: Proses tingkatan kesulitan permainan hanya dilakukan pada permainan dengan komputer sebagai lawan main. Komputer akan menggunakan rekursi untuk dapat mengambil langkah dengan keuntungan yang terbesar. Gambar 3.9 menunjukkan algoritma rekursi tingkat kesulitan permainan.



Gambar 3.9 Algoritma tingkat kesulitan permainan.

IV. IMPLEMENTASI DAN PEMBAHASAN

4.1 APLIKASI SERVER PERMAINAN

Aplikasi server terdiri dari beberapa bagian yaitu:

1. Server socket

Server socket merupakan bagian dari aplikasi server yang akan melayani permintaan koneksi dari klien. Server socket dibuat dengan perintah-perintah sebagai berikut:

```
TcpPort=10000
serverSocket=socket.socket(socket.AF_INET
,socket.SOCK_STREAM)
serverSocket.bind(('',TcpPort))
serverSocket.listen(5)
thread.start_new_thread(serverloop,(0,serverSocket,TcpPort))
```

2. Server loop

Server loop merupakan bagian dari server permainan yang selalu mendengarkan pada socket yang digunakan untuk melayani koneksi dari klien. Jika ada permintaan koneksi, server akan membuat sebuah thread baru untuk klien tersebut. Server loop dibentuk dengan perintah-perintah sebagai berikut:

```
def serverloop(self,s,port):
    while running:
        conn,addr=s.accept()
        if conn:

thread.start_new_thread(clientloop,(conn,addr))
```

3. klien loop

klien loop merupakan bagian dari server permainan yang digunakan untuk keperluan hubungan antar klien yang terhubung dengan server permainan. Perintah-perintah berikut digunakan untuk membentuk klien loop.

```
def clientloop(sock,addr):
    ...
    try:req=so.recv(1024)
    except:break
    clientlock.acquire()
    req=req.split('#')
    ...
    clientlock.release()
    so.close()
```

4. Bagian koneksi

Bagian koneksi server akan mencatat data-data dari klien yang terhubung ke server permainan. Bagian koneksi dibentuk dengan perintah-perintah sebagai berikut:

```
def connect(self,sock,nick,ip,port):
    print 'Client connect:',nick
    if nick in self.player:return
'ERR#Nama nick ini sudah ada !'

self.player[nick]=[sock,nick,ip,port,0]
return 0
```

4.2 APLIKASI PERMAINAN OTHELLO (KLIEN)

Pada aplikasi klien terdapat bagian-bagian yang berperan penting seperti berikut:

1. bagian pengecekan langkah

Bagian pengecekan langkah ini berfungsi untuk menguji posisi petak papan permainan yang diambil oleh pemain, harus memenuhi peraturan

permainan. Pengecekan langkah dibentuk dari perintah-perintah sebagai berikut:

```
def ceklangkah(x,y,data,giliran):
    def
cekxy(x,y,data,giliran,lawan,arahx,batasx
,arahy,batasy):
    if (arahx<>0 and x==batasx) or
(arahy<>0 and y==batasy):return 0
    if
data[y+arahy][x+arahx]<>lawan:return 0
    tx=x+arahx*2
    ty=y+arahy*2
    while 1:
        diluar=(-tx*arahx<-batasx) or
(-ty*arahy<-batasy)
        if diluar:break
        d=data[ty][tx]
        ok=d==giliran
        if d==0 or d==giliran: return
ok*arahmakan[(arahx,arahy)]
        tx+=arahx
        ty+=arahy
    return 0
```

2. Bagian memakan bidak:

Bagian memakan bidak adalah bagian dari aplikasi Othello untuk mengambil alih bidak lawan menjadi milik pemain. Kode program berikut digunakan untuk mengambil alih bidak lawan.

```
def domakan(x,y,data,giliran,arah):
    def
makanxy(x,y,data,giliran,arahx,batasx,arahy,batasy):
        tx=x+arahx
        ty=y+arahy
        while not ((-tx*arahx<-batasx) or
(-ty*arahy<-batasy)):
            d=data[ty][tx]
            ok=d==giliran
            if d==0 or d==giliran: break
            data[ty][tx]=giliran
            tx+=arahx
            ty+=arahy
            if anim:
                if formgame.data==data:
formgame.canv.refresh()
                time.wait(100)
                app.do()
    return 0
```

3. Bagian menghitung bobot.

Bagaian ini merupakan bagian khusus untuk permainan dengan komputer sebagai lawan. Proses menghitung bobot merupakan bagian paling menentukan dalam hal penentuan prioritas langkah yang akan dipilih oleh komputer. Kode program berikut ini digunakan untuk menghitung bobot nilai dari tiap kemungkinan langkah.

```
def dobobot(x,y,data,giliran,arah):
    def
makanxy(x,y,data,giliran,arahx,batasx,arahy,batasy,n):
        tx=x+arahx
        ty=y+arahy
        while 1:
            diluar=(-tx*arahx<-batasx) or
(-ty*arahy<-batasy)
```

```

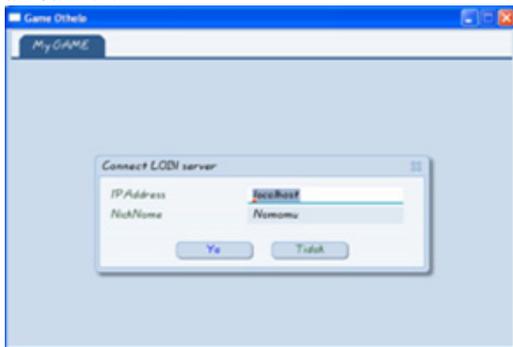
if diluar:break
d=data[ty] [tx]
ok=d==giliran
if d==0 or d==giliran: break
n+=nilaititik[ty] [tx]
tx+=arahx
ty+=arahy
return n

```

4.3 HASIL DAN PEMBAHASAN

1. Tampilan awal

Pada awal dijalankan aplikasi klien akan menampilkan sebuah dialog untuk pilihan jenis permainan yang akan dilakukan. Tampilan awal dari aplikasi permainan othello ditunjukkan pada Gambar 4.1.



Gambar 4.1 Tampilan awal aplikasi klien.

2. Tampilan menu permainan lokal

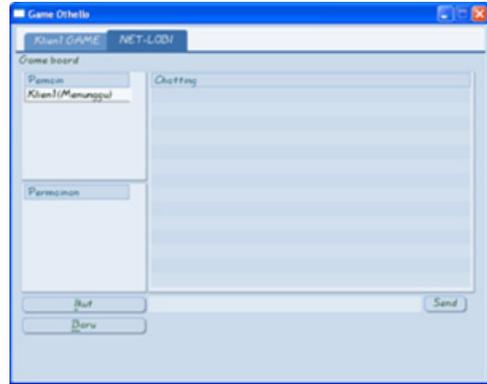
Menu permainan lokal akan dimunculkan jika pada dialog pilihan, pemain memilih tidak terhubung dengan server permainan. Menu permainan lokal terdiri dari mulai baru, bantuan, ke lobi, keluar. Gambar 4.2 menunjukkan tampilan menu permainan lokal.



Gambar 4.2 Tampilan menu permainan lokal.

3. Tampilan menu permainan online

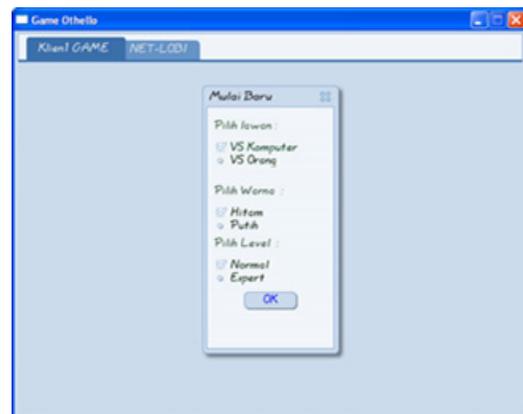
Bila pemain memilih untuk terhubung dengan server permainan maka akan muncul menu permainan yang terdiri dari menu lokal dan lobi jaringan. Gambar 4.3 menunjukkan tampilan lobi jaringan.



Gambar 4.3 Tampilan menu permainan online.

4. Tampilan mulai permainan baru (lokal)

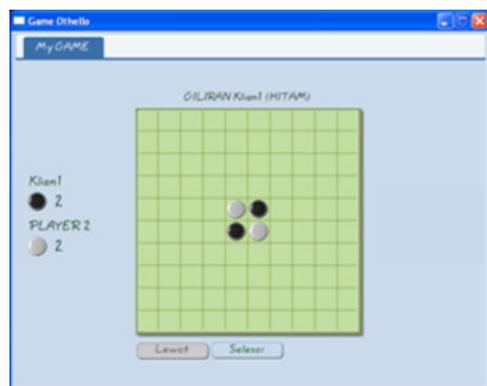
Bila pemain memilih untuk memulai permainan baru lokal, maka akan muncul sebuah menu untuk permainan lokal yang terdiri dari pilihan lawan main, pilihan bidak, dan pilihan tingkat kesulitan yang khusus digunakan jika memilih melawan komputer. Gambar 4.4 Menunjukkan tampilan menu pilihan permainan baru lokal.



Gambar 4.4 Tampilan menu permainan baru lokal.

5. Tampilan awal permainan

Permainan dimulai dengan menempatkan masing-masing dua bidak hitam dan putih pada tengah-tengah papan permainan dengan posisi saling bersilangan. Gambar 4.5 menunjukkan posisi awal permainan.



Gambar 4.5 Tampilan posisi awal permainan.

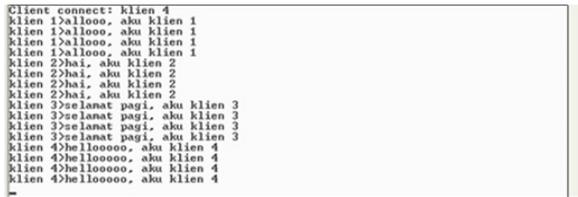
6. Tampilan server menerima koneksi

Server akan mencatat data-data klien yang melakukan koneksi dengan server permainan. Gambar 4.6 menunjukkan tampilan server saat menerima permintaan koneksi dari klien.



Gambar 4.6 Tampilan koneksi server permainan.

7. Tampilan server meneruskan pesan klien
Setiap pesan yang dikirim oleh klien akan diteruskan ke setiap klien yang terhubung dengan server permainan. Gambar 4.7 menunjukkan tampilan server saat meneruskan pesan dari klien.



Gambar 4.7 Tampilan server meneruskan pesan klien.

8. Hasil pengujian
Pengujian aplikasi dilakukan dengan menggunakan komputer dengan spesifikasi komputer sebagai berikut:

Server Permainan

prosesor : Intel Celeron 1,8 Ghz.
 memori : 256 Mb.
 VGA : Intel Extreme, 64 Mb share memori.
 Kartu jaringan : Realtek RTL 8139 PCI Fast Ethernet.
 Sistem Operasi : Microsoft Windows XP + SP2.
 IP : 200.200.200.3

Klien

prosesor : Intel Pentium II 450 Mhz.
 memori : 128 Mb.
 VGA : Cirrus Logic, 4 Mb.
 Kartu jaringan : Realtek RTL 8139 PCI Fast Ethernet.
 Sistem Operasi : Microsoft Windows XP + SP2.
 IP : 200.200.200.5

Tabel-tabel Berikut ini merupakan hasil pengujian terhadap aplikasi permainan othello. Hasil pengujian permainan lokal, permainan online dan pengujian dengan pengujian dengan pemberian kondisi khusus ditunjukkan secara berturut-turut pada tabel 4.1, 4.2, dan 4.3.

Tabel 4.1 hasil pengujian permainan lokal

| No | Jenis Pengujian | Hasil |
|----|--|----------|
| 1 | Menjalankan aplikasi | Berhasil |
| 2 | Memilih jenis permainan (lokal/LAN) | Berhasil |
| 3 | Memulai permainan baru | Berhasil |
| 4 | Melihat bantuan | Berhasil |
| 5 | Menuju lobi (koneksi ke server) | Berhasil |
| 6 | Memilih lawan main (CPU/Pemain lain) | Berhasil |
| 7 | Memilih karakter (warna bidak) | Berhasil |
| 8 | Memilih tingkat kesulitan (melawan Komputer) | Berhasil |
| 9 | Mengganti nama pemain | Berhasil |
| 10 | Pengecekan langkah | Berhasil |
| 11 | Mengambil alih bidak lawan | Berhasil |
| 12 | Tanggapan CPU (melawan CPU) | Berhasil |
| 13 | Melewatkan giliran | Berhasil |

| | | |
|----|-----------------------------|----------|
| 14 | Mengakhiri permainan | Berhasil |
| 15 | Menutup aplikasi dan keluar | Berhasil |

Tabel 4.2 Hasil pengujian pada permainan online.

| No | Jenis Pengujian | Hasil |
|----|--|----------|
| 1 | Menjalankan aplikasi server | Berhasil |
| 2 | Menjalankan aplikasi permainan Othello | Berhasil |
| 3 | Mengganti nama pemain | Berhasil |
| 4 | Melakukan koneksi ke server | Berhasil |
| 5 | Melihat lobi | Berhasil |
| 6 | Mengirim dan membaca pesan | Berhasil |
| 7 | Membuat ruang permainan baru | Berhasil |
| 8 | Bergabung ke ruang permainan | Berhasil |
| 9 | Memulai permainan jaringan | Berhasil |
| 10 | Mengakhiri permainan | Berhasil |
| 11 | Menutup koneksi ke server | Berhasil |
| 12 | Memulai permainan lokal | Berhasil |
| 13 | Mengakhiri permainan | Berhasil |
| 14 | Menutup aplikasi | Berhasil |

Tabel 4.3 Hasil pengujian dengan pemberian kondisi khusus.

(lihat halaman lampiran 1)

V. PENUTUP

Pada bagian penutup ini membahas tentang kesimpulan yang dapat diambil dan saran-saran untuk pengembangan aplikasi permainan othello.

5.1 KESIMPULAN

1. Aplikasi permainan Othello yang dibuat sudah sesuai dengan perencanaan.
2. Aplikasi *server* hanya mengatur koneksi klien dan pertukaran data antar klien.
3. Tingkat kesulitan permainan hanya digunakan pada permainan lokal dengan komputer sebagai lawan. Jumlah perulangan fungsi rekursi digunakan sebagai penentu tingkat kesulitan permainan.
4. Banyaknya perhitungan yang dilakukan fungsi rekursi dapat menyebabkan beban proses yang besar pada prosesor.
5. Pada permainan secara on-line, seorang pemain yang bergabung dengan ruang permainan pemain lain diharuskan untuk menghapus ruang permainan miliknya jika pemain tersebut sudah memiliki ruang permainan sendiri.
6. Pesan dari satu klien akan diteruskan pada semua klien yang terhubung dengan server.

5.2 SARAN

1. Penambahan objek gambar, animasi, efek suara, video dapat digunakan untuk penyempurnaan aplikasi permainan Othello.
2. Penambahan menu untuk pengaturan ukuran layar, volume musik latar, dsb.
3. Pembuatan fungsi yang lebih efektif untuk dapat menanggapi langkah pemain.
4. Penerapan pemrograman komunikasi jaringan komputer dengan kompresi data dapat digunakan pada aplikasi bila aplikasi akan ditujukan untuk mendukung permainan internet.

3. Purbo, dkk., *Trik Pemrograman Java untuk Jaringan dan Internet*, PT. Elex Media Komputindo, Jakarta, 2000.
4. Rosidi, R.I., *Membuat Sendiri Game Server dan Multiplayer Game*, PT. Elex Media Komputindo, Jakarta, 2003.
5. Utami, E., dan Raharjo, S., *logika algoritma dan implementasinya dalam bahasa Python di GNU/Linux*, Penerbit ANDI, Yogyakarta, 2004.
6. <http://www.objectcentral.com/oobook/Chapter2...>
7. <http://www.pygame.org,...>
8. <http://www.python.org/psf,...>
9. http://www.sei.cmu.edu/intro/process/technqs/q_uc m...
10. <http://www.visualcase.com/tutorials/use-case-diagram...>

Riwayat Penulis



Nama : Rori Setiawan
NIM : L2F301470
Konsentrasi: Informatika dan
Komputer
Jurusan : Teknik Elektro
Universitas Diponegoro

Menyetujui

Pembimbing I

pembimbing II

DAFTAR PUSTAKA

1. Fowler, M., *UML Distilled, 3th Ed., A Brief Guide to the Standard Object Modeling Language*, Pearson Education, 2004.
2. Hendri, *Cepat Mahir Python*, <http://www.ilmukomputer.com>, 2003.

Aghus Sofwan, ST, MT
NIP. 132 163 757

Maman Somantri, ST, MT
NIP. 132 231 133