

# Perancangan permainan dakon menggunakan C++ dan GLUT (*OpenGL Utility Toolkit*)

## *Designing dakon game using C++ and GLUT (OpenGL Utility Toolkit)*

Tenia Wahyuningrum<sup>#</sup>

<sup>#</sup>SI-Teknik Informatika

STT Telematika Telkom Purwokerto

tenia@stttelematikatelkom.ac.id

**Abstract** — Dakon is one of Indonesian traditional game which is now rarely played by children. After long time, this game will be extinct. Whereas traditional game has several advantages including speed and accuracy. Hence came the idea to develop dakon game into a more modern form of the media through the computer. Game designed and built using the language C++ and GLUT as computer graphics. From the results of testing by a team of product developers, game dakon been going well, and have met the criteria of verification testing, validation and exploration.

**Key words**— dakon, GLUT, C++.

### I. PENDAHULUAN

Pemandangan anak-anak Indonesia bermain permainan tradisional seperti dakon, kini sudah tidak mudah lagi ditemui di kota-kota besar. Tak dapat dipungkiri lagi bahwa keberadaan permainan tradisional perlahan menghilang setelah teknologi ada dan semakin berkembang maju. Permainan tradisional yang turun-temurun sejak nenek moyang terdesak modernisasi karena secara temporal lebih menarik dari sisi tertentu. Saat ini hanya sekitar 20% permainan tradisional yang bertahan, di antaranya engklek maupun jangkungan. Permainan tradisional lebih cepat terkikis terutama di kawasan perkotaan [1]. Jika keadaan ini dibiarkan terus menerus, maka tidak mengherankan jika suatu saat nanti permainan tradisional akan punah. Dengan menggabungkan permainan tradisional yang dikemas secara modern, diharapkan anak-anak akan lebih menyukai dan mengenal permainan dari negeri sendiri, sehingga tidak kehilangan jati diri bangsa.

Salah satu permainan tradisional Indonesia adalah congklak atau dakon. Dakon memiliki asal-usul dari Afrika atau Arab. Beberapa bukti tertua ditemukan di National Old board Geographic disponsori penggalian arkeologi dakon datang kembali ke 7.000 menjadi 5.000 SM di zaman sekarang Yordania. Penggalian sebuah rumah menemukan sebuah lempengan batu kapur

dengan dua baris paralel melingkar. Banyak ahli menduga bahwa dakon mungkin sebenarnya papan permainan tertua yang pernah ada. Dakon dikenal dengan nama yang berbeda dari daerah ke daerah. Di Malaysia, permainan ini dikenal sebagai Congkak, nama yang banyak digunakan di provinsi Sumatera. Di Jawa, permainan ini dikenal sebagai Congklak, dakon, dhakon atau dhakonon. Di Lampung, permainan ini disebut dentuman lamban. Di Sulawesi, permainan ini disebut sebagai Mokaotan, Maggaleceng, Aggalacang dan Nogarata [2].

Papan dakon terbuat dari kayu dengan variasi jumlah lubang kecil di setiap sisi, yaitu 5, 6, 7 atau 9 lubang. Semua papan diakhiri dengan sebuah lubang besar pada masing-masing sisi. Desain papan dakon di daerah Jawa Tengah umumnya memiliki naga di ujung masing-masing papan, dengan wajah saling membelakangi, ekornya di dekorasi sebagai papan permainan dan lengannya menyangga papan ke lantai. Warna papan dakon yang populer yaitu merah dan emas. Anak-anak desa di Lampung seringkali memainkan dakon tanpa papan, tetapi mereka membuat area bermain dengan membuat lingkaran-lingkaran di tanah. Sejenis cangkang kerang digunakan sebagai biji dakon dan jika tidak ada, kadangkala digunakan biji-bijian dari tumbuh-tumbuhan, biji buah sawo yang disebut kecil atau biji buah asam jawa yang disebut klungsu, juga batu-batuan kecil [3].

Dengan latar belakang tersebut, maka muncul ide untuk mengembangkan sebuah permainan tradisional menjadi permainan modern dengan mengubah kemasannya, untuk melatih kecepatan dan ketepatan berpikir penggunanya. Permainan yang akan dirancang merupakan jenis permainan papan (*board game*) menggunakan sistem *windowing* yang independen dari sistem operasinya. Ada banyak cara untuk membuat dan memperlihatkan suatu *window* dalam berbagai sistem

window. Salah satunya adalah dengan menggunakan OpenGL Utility Toolkit (GLUT). Hal ini berarti program yang dibuat dengan GLUT dapat beroperasi pada sistem *windowing* yang berbeda tanpa mengubah kode secara manual.

## II. METODE

### A. Konsep Game

Aturan dalam bermain dakon di setiap daerah berbeda-beda namun memiliki kesamaan konsep, yaitu bertujuan mengumpulkan biji dalam lubang besar sebanyak-banyaknya. Aturan yang digunakan dalam *game* ini yaitu dimainkan oleh dua orang pemain. Sasaran pemain *game* dakon adalah pemain berusia 6 tahun keatas. Peralatan yang dibutuhkan adalah sebuah papan dengan 16 lubang yang terdiri dari 14 lubang kecil dan dua lubang besar (tujuh lubang kecil dan satu lubang besar untuk masing-masing pemain), serta 98 (14 x 7) biji dakon.

Setelah setiap lubang kecil diisi dengan tujuh biji, pemain dengan giliran pertama mengambil seluruh biji yang terdapat pada salah satu lubang kecil dan membagikan biji tersebut pada lubang-lubang (kecuali lubang besar milik lawan) searah jarum jam, sampai habis. Jika biji habis di lubang yang berisi biji, maka pemain tersebut dapat terus bermain sampai ia mati sehingga pemain kedua mendapat giliran bermain. Keadaan mati yaitu ketika biji habis di lubang kecil yang kosong atau di lubang besar miliknya.

Jika pemain mati di lubang kecil miliknya, sementara pada lubang kecil lawan yang berhadapan dengan lubang kecil tempat ia mati terdapat biji, maka ia berhak mengambil seluruh biji yang terdapat pada lubang kecil lawan yang berhadapan itu, keadaan tersebut dinamakan menembak. Tapi jika pemain mati di lubang kecil milik lawan, pemain tersebut tidak berhak melakukan apapun. Jika pemain mati di lubang besar miliknya, maka pemain dapat memilih lubang kecil lagi untuk melanjutkan permainan. Permainan ini akan terus berlanjut sampai tak ada lagi biji pada lubang kecil salah satu atau kedua pemain. Pemain yang dikatakan memenangkan permainan adalah pemain yang berhasil memasukkan biji paling banyak ke dalam lubang besar miliknya [4].

### B. Desain Game

1) *Desain User Interface*. Papan dakon dibuat dengan menggunakan *software* aplikasi model 3D. Background tampilan layar dan papan dakon kemudian diolah menggunakan *software* pengolah citra Corel Draw 11 dan Adobe Photoshop.

2) *Desain Interaksi*. Interaksi *player* dan komputer dilakukan melalui *mouse* dan *keyboard*. Biji dakon disimulasikan dengan menggunakan angka yang menunjukkan jumlah biji pada setiap lubang. *User* memilih biji dengan cara mengeklik lubang kecil miliknya. Angka pada lubang-lubang akan bertambah sesuai dengan aliran sebaran biji.

3) *Desain teknis*. Pembuatan *image* untuk desain papan dakon menggunakan *software* pengolah citra Corel Draw 11 dan Adobe Photoshop. Proses rendering visual menggunakan GLUT dengan library yang digunakan antara lain *Glaux.lib*, *Glut32.lib*, *OpenGlut.lib*. Pengkodean program dilakukan dengan menggunakan bahasa pemrograman C++. *Image* yang di render yaitu *image* papan dakon, *pointer*, dan *background*.

### C. Deskripsi sistem

#### 1) Deskripsi Perangkat Keras

- CPU, minimal Pentium 3 atau yang setara
- RAM, minimal sebesar 128 Mbyte
- Mouse, Keyboard

2) *Deskripsi Perangkat Lunak*. Secara garis besar, arsitektur perangkat lunak dari *gamedakon* terdiri dari komponen-komponen sebagai berikut :

- Modul *game* : berfungsi untuk mendefinisikan properti dari data-data yang terkait pada *engine*, menyusun logika, dan menyusun alur permainan.
- Modul *gameengine* : menggabungkan modul grafis dan input.
- Modul visual : mengolah semua data yang berhubungan dengan visual.
- Modul *artificial intelligence* : mengolah semua data yang berhubungan dengan *artificial intelligence*.
- Modul *input* : menerima masukan dari perangkat input *mouse* dan *keyboard*.
- Modul *output* : menampilkan keluaran ke perangkat output *monitor*.

3) *Interface*. Application Programming Interface (API) yang digunakan untuk berkomunikasi dengan hardware yaitu OpenGL / DirectX, *interface* untuk berkomunikasi dengan hardware pengolah grafik 3 dimensi.

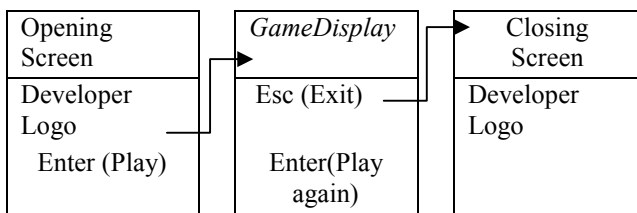
4) *Deskripsi Data Grafis*. *Geometric primitive* digambar dengan warna solid atau warna hasil interpolasi warna-warna *vertex*-nya. *Texture mapping* memungkinkan untuk menaruh gambar pada *geometric primitive* tersebut dan sekaligus mengikuti transformasi yang diterapkan kepada polygon tersebut [5]. Data grafis yang digunakan adalah data texture yang merupakan

*data image* (citra) yang digunakan dalam tampilan papan dakon, *pointer* dan *background*.

#### D. Desain Produk

Desain interaksi produk digunakan untuk mendukung orang berkomunikasi dan berinteraksi pada kehidupan sehari-hari. Desain interaksi terkait dengan bagaimana membuat pengalaman pengguna lebih berkualitas, hal ini memerlukan sejumlah faktor ketergantungan, termasuk konteks penggunaan, tipe aktivitas, perbedaan budaya, dan kelompok user. Desain interaksi dibuat berdasarkan berbagai disiplin ilmu [6].

1) *Desain User Interface*. Desain User Interface *gamedakon* memiliki diagram alir seperti pada gambar diagram alir *user interface game dakon* berikut. Ketika pemain mengklik *icon game dakon*, maka akan tampil layar pembuka/*opening screen* yang berisi logo *developer* dan instruksi tekan tombol enter untuk memulai permainan. Setelah pemain menekan tombol enter, maka akan muncul tampilan *game dakon*, terdapat dua menu yang dapat dipilih yaitu Esc untuk keluar dari permainan, dan Enter untuk bermain kembali setelah permainan selesai. Jika user menekan tombol Esc maka akan tampil layar penutup/*closing screen* yang berisi logo *developer*. *Closing Screen* berisi keterangan yang dihasilkan pada permainan (menang/kalah) (Gambar 1).

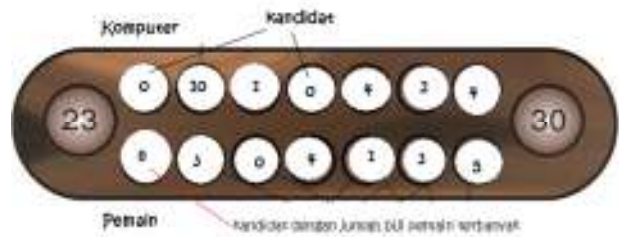


Gambar 1. Diagram alir *user interface game dakon*

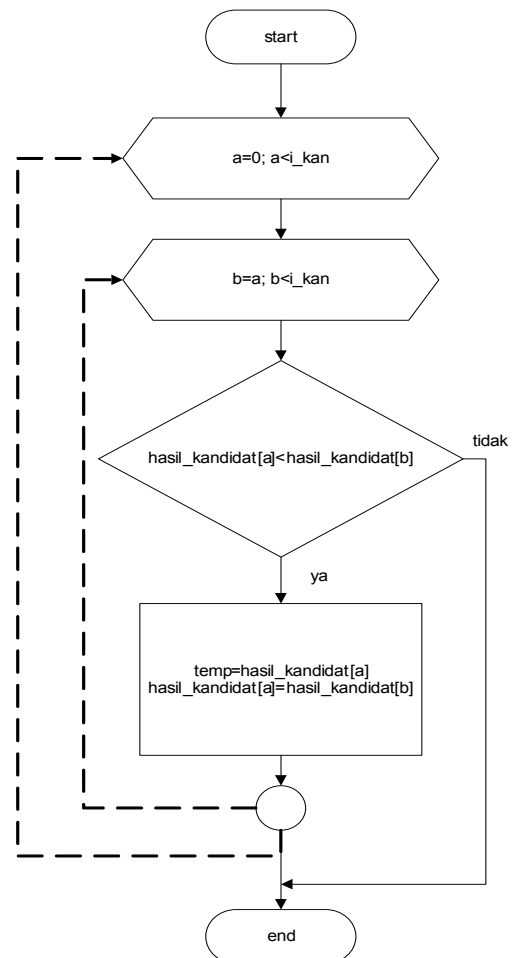
2) *Artificial Intelligence*. *Artificial intelligence* pada *game "dakon"* ditulis dalam fungsi AI(). Fungsi tersebut melakukan perulangan untuk memilih kandidat lubang kecil yang akan dimainkan. Pemilihan kandidat didasarkan pada lubang kecil yang kosong di sisi komputer. Hasil pemilihan kandidat tersebut kemudian digunakan sebagai parameter untuk memanggil fungsi *jalan\_invisible*. Fungsi *jalan\_invisible* digunakan untuk memberikan skor pada setiap kandidat. Score ini merupakan nilai *heuristic* untuk mengevaluasi nilai yang merepresentasikan hasil permainan yang akan terjadi jika langkah tersebut dipilih.

Dari lubang-lubang kecil yang dijadikan kandidat, diurutkan berdasarkan jumlah biji di lubang kecil milik pemain secara menaik menggunakan algoritma *exchange sort*. Algoritma tersebut sangat mirip dengan *Bubble Sort*, dan banyak yang mengatakan *Bubble Sort*

sama dengan *Exchange Sort*. Perbedaan ada dalam hal bagaimana membandingkan antar elemen-elemennya. *Exchange sort* membandingkan suatu elemen dengan elemenelemenlainnya dalam array tersebut, dan melakukan pertukaran elemen jika perlu. Jadi ada elemen yang selalu menjadi elemen pusat (*pivot*). Setelah diurutkan secara menaik, selanjutnya memilih kandidat lubang kecil yang layak. Lubang kecil yang layak yaitu lubang kecil dengan jumlah biji dalam lubang kecil milik pemain yang terbanyak. Proses tersebut diilustrasikan dengan gambar ilustrasi pemilihan kandidat (Gambar 2 dan Gambar 3).



Gambar 2. Ilustrasi pemilihan kandidat



Gambar 3. Diagram alir *artificial intelligence game "dakon"*

Berdasarkan kedua gambar tersebut dapat dikodekan sebagai berikut:

```

for(a=0;a<7;a++){
    if(inv_player2[a]!=0){
        //lubang yang ada isinya adalah kandidat
        hasil =
        jalan_invisible(a+1,inv_player2,inv_player1);
        kandidat[i_kan]=a+1;
        hasil_kandidat[i_kan]=hasil;
        i_kan++;0
    }
}

for(a=0;a<i_kan-1;a++){
    for(b=a;b<i_kan;b++){
        if(hasil_kandidat[a]<hasil_kandidat[b]){
            temp = hasil_kandidat[a];
            hasil_kandidat[a] = hasil_kandidat[b];
            hasil_kandidat[b] = temp;
            temp = kandidat[a];
            kandidat[a] = kandidat[b];
            kandidat[b] = temp;
        }
    }
}

```

### III. HASIL DAN PEMBAHASAN

Desain produk *game* dakon kemudian diimplementasikan menjadi sebuah prototipe produk. Prototipe produk tersebut kemudian diuji oleh tim pengembang untuk mengetahui kekurangan dalam desain dan kesalahan dalam penulisan program. Prototipe dalam desain interaksi dapat berupa seri sketsa layar, *storyboard* (seperti kartun), *powerpoint slide show*, simulasi video penggunaan system, kardus bekas untuk simulasi ruangan, atau suatu *software* yang memiliki fungsi yang masih terbatas .

Tahapan implementasi produk *game* dakon

1. Menginstal *software* yang menjadi tool pengembangan;
2. Membuat citra untuk *user interface* dengan *software* pengolah citra ;
3. Menulis kode program;
4. Menjalankan aplikasi.

*Software* yang diinstall sebagai tool pengembangan yaitu *software* pengolah citra Adobe Photoshop, Corel Draw, *softwarecompiler* CodeBlocks, *software* pengolah kata OpenOffice.org, Microsoft Word 2007 untuk dokumentasi pengembangan produk, dan sistem operasi Windows XP.

Desain *user interface game* dakon mengalami beberapa perubahan pada tahap implementasi. Pada *game* dakon v.1.0 tampilan masih dalam bentuk *console*

(Gambar 4), kemudian perubahan tampilan diimplementasikan pada *gamedakon* v.1.1 (Gambar 5), tetapi masih menemui kesalahan dalam pembangunan tampilan 3D. Tampilan *user interface* kemudian diperbaiki kembali pada *game* dakon v.1.2 (Gambar 6, 7 dan 8). Metamorfosa tampilan *user interface* dapat dilihat sebagai berikut.



Gambar 4. *User interface* dakon v.1.0



Gambar 5. *User interface* dakon v.1.1



Gambar 6. *User interface opening screen* dakon v.1.2



Gambar 7. User interface game display dakon v.1.2



Gambar 8. User interface closing screen dakon v.1.2

Penulisan kode program diimplementasikan menjadi modul-modul yang saling berkaitan satu sama lain. Modul-modul tersebut dapat dibagi menjadi beberapa modul sebagai berikut.

- Modul yang terkait dengan user interface *game* dakon,
- Modul yang terkait dengan atribut dan perilaku objek,
- Modul yang terkait dengan AI,
- Modul yang terkait dengan penilaian,
- Modul yang terkait dengan integrasi permainan.

Modul yang terkait dengan *user interface* dakon yaitu modul `display()` untuk menampilkan *background* tampilan *opening screen* saat *game* di load, *gamedisplay* saat *game* dimainkan, dan tampilan *closing screen* saat *game* berakhir.

#### A. Modul `init()` //untuk inisialisasi gambar

```
glClearColor(0,0,0,0);
selected = 0;
```

```
//panjang koordinat
gluOrtho2D(0,800,600,0);
// Panggil gambar disini
loadbmp(textures, "dakon\\back.bmp", 0); //background
loadbmp(textures, "dakon\\lubang_glut2_01.bmp", 1);// 1
loadbmp(textures, "dakon\\lubang_glut2_39.bmp", 2);// 39
loadbmp(textures, "dakon\\com01.bmp", 3); // 3 / lubang
loadbmp(textures, "dakon\\lubang_glut2_25.bmp", 4);
// 25 / blank
loadbmp(textures, "dakon\\lambung_com.bmp", 5);
// 18 / lambung
loadbmp(textures, "dakon\\lubang_glut_besar_02.bmp", 6);
// 2
loadbmp(textures, "dakon\\lubang_glut2_17.bmp", 7);// 17
loadbmp(textures, "dakon\\lubang_glut2_37.bmp", 8);// 37
loadbmp(textures, "dakon\\lubang_glut2_16.bmp", 9);// 16
loadbmp(textures, "dakon\\lubang_glut2_22.bmp", 10);// 22
loadbmp(textures, "dakon\\lubang_glut2_38.bmp", 11);// 38
loadbmp(textures, "dakon\\pointer.bmp", 12); // 38
loadbmp(textures, "dakon\\sprite.bmp", 13); // 38
loadbmp(textures, "dakon\\game_over.bmp", 14); // 38
loadbmp(textures, "dakon\\pointer2.bmp", 15); // 38
```

#### B. Modul `display`

```
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, textures[0]);
glPushMatrix();
glBegin(GL_QUADS);
glTexCoord2f(0,1);
glVertex2f(0, 0);
glTexCoord2f(1,1);
glVertex2f(800, 0);
glTexCoord2f(1,0);
glVertex2f(800, 600);
glTexCoord2f(0,0);
glVertex2f(0, 600);
glEnd();
glPopMatrix();
glDisable(GL_TEXTURE_2D);
```

Modul `renderBitmapCharacher(float x, float y, float z, void *font,char *string)` untuk menampilkan jumlah biji dakon di setiap lubang dengan karakter angka. Modul papan (`GLenum mode`) untuk menampilkan papan dakon, modul `pointer(int x, int y, int _player)` digunakan untuk menampilkan pointer yang menunjuk setiap lubang yang diisi biji dakon. Modul `display_status()` untuk menampilkan umpan balik kepada pemain, yaitu keterangan siapa yang sedang bermain, jumlah biji dalam genggamannya.

Modul yang terkait dengan atribut dan perilaku objek yaitu modul `jalan_pemain()` untuk menjalankan permainan pemain, dan modul `jalan_comp()` untuk menjalankan permainan komputer. Modul `resize( int w, int h )` dibuat untuk ukuran windows yang dapat diubah-ubah. Modul `mouse(int button, int _state, int x, int y)`

digunakan untuk menjalankan fungsi perangkat input *mouse*. Modul *myKeyboard*(unsigned char key,int x,int y) digunakan untuk menjalankan fungsi perangkat input *keyboard*.

Modul yang terkait dengan AI yaitu modul *AI()* digunakan untuk memilih kandidat lubang kecil yang akan dimainkan komputer dan modul *jalan\_invisible()* untuk menentukan lubang kecil yang akan dimainkan berdasarkan kandidat yang terpilih.

Modul yang terkait dengan penilaian yaitu modul *nilai()* untuk memberikan nilai pada pemain dan komputer. Modul *skor\_akhir()* merupakan modul untuk menentukan pemenang dari permainan.

#### C. Modul nilai()

```
char jml_kecik_p1[8][5];
char jml_kecik_p2[8][5];
char pemain[2];
char kecil_di_tangan1[5];
char kecil_di_tangan2[5];
char status[5];
char posisi[5];

sprintf(pemain,"%d",player);
sprintf(status,"%c",stat);
sprintf(kecil_di_tangan1,"%d",genggam);
sprintf(kecil_di_tangan2,"%d",tmp_genggam);
if(stat=='P') sprintf(posisi,"%d",x+1);
else if(stat=='L') sprintf(posisi,"%d",y+1);

/*renderBitmapCharacher(10,80,0,(void *)font,pemain);
renderBitmapCharacher(10,100,0,(void *)font,status);
renderBitmapCharacher(10,120,0,(void *)font,posisi);
renderBitmapCharacher(10,140,0,(void
*)font,kecil_di_tangan1);
renderBitmapCharacher(10,160,0,(void
*)font,kecil_di_tangan2);
sprintf(status,"%d",hole);
renderBitmapCharacher(10,180,0,(void *)font,status);*/

sprintf(jml_kecik_p1[7],"%d",player1[7]);
sprintf(jml_kecik_p2[7],"%d",player2[7]);

renderBitmapCharacher(108,303,0,(void
*)font,jml_kecik_p1[7]);
renderBitmapCharacher(670,303,0,(void
*)font,jml_kecik_p2[7]);
for(int x=0;x<7;x++){
    sprintf(jml_kecik_p1[x],"%d",player1[x]);
    sprintf(jml_kecik_p2[x],"%d",player2[x]);
}
int y=6;
for(int x=0;x<7;x++){
    renderBitmapCharacher(180+(x*70),265,0,(void
*)font,jml_kecik_p2[x]);
```

```
renderBitmapCharacher(180+(x*70),343,0,(void
*)font,jml_kecik_p1[y]);
    y--;
}
```

Modul yang terkait dengan integrasi permainan yaitu modul *main* berfungsi untuk inisialisasi *world* dan pengaturan konfigurasi sistem, yaitu inisialisasi *window*, *load* tampilan awal, memanggil fungsi *mouse*, *keyboard*, dan *display*.

#### D. Modul main()

```
glutInit(&argc, argv);
glutInitWindowSize(800,600);
glutInitWindowPosition(0,0);
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
glutCreateWindow("Dakon V.1.2");
init();
glutMouseFunc(mouse);
glutKeyboardFunc(myKeyboard);
glutDisplayFunc(display);
glutTimerFunc(100, myTimeOut, 0);
glutMainLoop();
return 0;
```

#### E. Pengujian

Pengujian produk *game* dakon dilakukan oleh tim pengembang sendiri. Pengujian dilakukan agar perangkat lunak benar-benar berjalan dengan baik. Dari berbagai aktifitas yang dilakukan pada bagian pengujian, ada tiga hal yang menjadi fokus perhatian. Tiga hal tersebut meliputi :

- verifikasi,
- validasi,
- eksplorasi.

1) *Verifikasi* menjawab pertanyaan apakah perangkat lunak dikembangkan sesuai dengan yang direncanakan atau tidak. Verifikasi adalah membandingkan perangkat lunak dengan hal-hal yang tercantum di dalam dokumen spesifikasi persyaratannya. Melalui verifikasi, dapat diketahui apakah semua spesifikasi yang diinginkan telah terfasilitasi di dalam perangkat lunak tersebut atau belum. Dari hasil pengujian oleh tim pengembang, *gamedakon* telah dikembangkan sesuai spesifikasi yang telah direncanakan pada konsep dan desain *user interface game*.

2) *Validasi* menjawab pertanyaan apakah perangkat lunak berperilaku sesuai dengan yang seharusnya atau tidak. Pada saat proses validasi, penguji menentukan keluaran yang diinginkan pada setiap fungsi di dalam perangkat lunak tersebut. Selanjutnya penguji menjalankan fungsi yang akan diuji. Jika keluaran

fungsi sesuai dengan yang diinginkan, maka fungsi sudah dianggap valid. Jika semua fungsi menghasilkan keluaran atau berperilaku sesuai dengan yang diinginkan, maka perangkat lunak dapat dianggap sudah valid. Dari hasil pengujian oleh tim pengembang, *game* dakon telah memenuhi keluaran yang diinginkan, yaitu skor akhir, nilai masing-masing pemain, dan pemenang permainan.

3) *Eksplorasi* menjawab pertanyaan apakah perangkat lunak bersih dari kesalahan penulisan kode atau tidak. Eksplorasi pada umumnya dilakukan pada saat *source code* di-*compile*. Jika proses kompilasi gagal, maka akan muncul peringatan letak kesalahan-kesalahan yang ada di *source code*. Kesalahan sintak dapat saja tidak terdeteksi pada saat proses kompilasi. Oleh karena itu, ada beberapa pendekatan dalam proses pembersihan *bug* yang ada di dalam *source code*. Dari hasil pengujian oleh tim pengembang, *game* dakon yang dijalankan tidak terdapat kesalahan baik dalam penulisan kode program maupun kesalahan *runtime*.

#### F. Tahap pengujian

*Unit Testing* adalah pengujian yang pada low-level structure dari kode program. Digunakan untuk menguji apakah logical work dari setiap kode bekerja dengan baik. Testing yang digunakan adalah menguji apakah setiap unit sudah bekerja sesuai dengan fungsi masing – masing unit kode program (Tabel I). *Module Testing* adalah pengujian yang pada *low-level structure* dari module program. Digunakan untuk menguji apakah *logical work* dari setiap module bekerja dengan baik. Testing yang digunakan adalah menguji apakah setiap module sebagai kesatuan dari beberapa unit kode sudah bekerja sesuai dengan fungsi masing – masing (Tabel II). Sedangkan *Functional Testing* adalah pengujian untuk mengetahui aplikasi sudah sesuai kebutuhan pengguna (Tabel III).

TABEL I  
PENGUJIAN UNIT

No	Modul	Unit	Fungsi	Hasil
1	Game	init	Inisialisasi variabel, class, gambar	OK
2	Game	main	Fungsi utama untuk memanggil class lain	OK
3	Game engine	processselecti on	komputer memilih lubang pemain/komputer	OK
		processshits	memproses pemilihan lubang	OK
		jalan pemain	menggerakkan permainan pemain	OK
		jalan comp	menggerakkan permainan komputer	OK
		jalan invisible	menggerakkan permainan tapi tidak terlihat di monitor	OK
3	Visual	nilai	memberikan nilai pada pemain dan komputer	OK
		papan	mengatur papan dakon	OK
		pointer	mengatur pointer (penunjuk) lubang mana yang sedang berjalan	OK
4	Artificial Intelligence	resize	mengatur ukuran window game	OK
		ai	mengatur pemilihan kandidat lubang (pada giliran komputer bermain)	OK
5	Input	mouse keyboard	mengatur mouse mengatur penekanan tombol esc dan enter	OK
6	Output	display status	mengatur tampilan akhir di monitor	OK

TABEL II  
PENGUJIAN MODUL

No	Modul	Sub System	Fungsi	Hasil
1	Game (unit init)	Loadbmp.cpp Bmp.h	Inisialisasi gambar	OK
2	Game (unit main)		Menguji integrasi permainan	
3	engine (unit processhits)		yang akan ditampilkan di monitor	
4	Input (unit mouse)	Glaux.lib Glut32.lib OpenGLut.lib	mulai dari input,	OK
5	Visual (unit papan, class pointer)		proses, sampai output	
	Output (unit display)			

TABEL III  
PENGUJIAN FUNGSIONAL

No	Kebutuhan	Kesesuaian
1.	User Interface	
	a. Opening Screen	Sesuai
	b. Game display	Sesuai
	c. Closing Screen	Sesuai
2	Atribut dan perilaku Objek	
	a. Jalan Pemain	Sesuai
	b. Jalan Komputer	Sesuai
3	Artificial Intelligence (komputer dapat memilih kandidat lubang yang sesuai)	Sesuai
4	Penilaian (dapat menampilkan nilai masing-masing pemain dan dapat menampilkan skor akhir permainan, serta menentukan pemenangnya)	Sesuai
5	Integrasi Permainan	Sesuai

#### IV. PENUTUP

Dari hasil penelitian ini dapat disimpulkan :

1. Telah dirancang suatu prototipe *game* dakon yang dengan menggunakan bahasa C++ dan GLUT.
2. Dari hasil pengujian unit, modul dan fungsional, *game* dakon yang dikembangkan telah memenuhi kriteria pengujian verifikasi, validasi dan eksplorasi.

#### UCAPAN TERIMA KASIH

Tim pengembang *game* dakon, Rizal, Rizky, Boni alumni S2 Media Digital dan Game Teknologi, Institut Teknologi Bandung 2010.

#### DAFTAR PUSTAKA

- [1] Permainan Tradisional, Bertahan Melawan Modernisasi Teknologi. <http://disparbud.jabarprov.go.id/applications/frontend/index.php?mod=news&act=showdetail&id=659>. Diakses tanggal 20 Februari 2013, 12:41 WIB
- [2] Ristanti, D., 2011. Pembuatan dan Perancangan *Game* Dakon dengan Macromedia Flash 8 Sebagai usaha untuk tetap melestarikan permainan tradisional pada era digitalisasi. Publikasi. STMIK AMIKOM Yogyakarta. [http://repository.amikom.ac.id/files/Publikasi\\_07.12.2314.pdf](http://repository.amikom.ac.id/files/Publikasi_07.12.2314.pdf). Diakses tanggal 23 Februari 2013, pukul 10.20
- [3] Expat Web Site Association, "Dakon: Traditional *Game* of Indonesia", <http://www.expat.or.id/info/dakon.html>. Diakses tanggal 20 Februari 2013, pukul 10.51 WIB.
- [4] Expat Web Site Association, "Dakon: Instructions for Play", <http://www.expat.or.id/info/dakoninstructions.html>. Diakses tanggal 20 Februari 2013, pukul 10.51 WIB.
- [5] Chapter 9, Texture Mapping. <http://www.gprogramming.com/openglredbook/chapter09.html>. Diakses tanggal 21 Februari 2013, pukul 09.45 WIB.
- [6] Preece, Rogers, Sharp, *Interaction Design, Beyond Human Computer Interaction, Chapter 1*. Second Edition. 2002. ISBN 0471492787.