

Paper_16.pdf

by

Submission date: 15-Aug-2019 06:29PM (UTC+0700)

Submission ID: 1160314346

File name: Paper_16.pdf (276.29K)

Word count: 2040

Character count: 12134

PENGGUNAAN STRUKTUR DATA UNTUK PEMBUATAN APLIKASI PERMAINAN TRADISIONAL INDONESIA

7

Yulia¹, Liliana²

Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra, Surabaya^{1,2}
Siwalankerto 121-131 Surabaya

yulia@petra.ac.id¹, lilian@petra.ac.id²

Abstrak

Struktur Data merupakan hal yang sangat penting karena menentukan jenis operasi yang akan dilakukan pada data dan seberapa efisiennya solusi yang dihasilkan. Hal ini juga menentukan seberapa dinamis dalam berhubungan dengan data, misalnya dalam menambahkan, menghapus atau mencari data. Penentuan bentuk struktur data yang digunakan untuk menyimpan data hanya bisa dilakukan setelah masalah diteliti atau paling tidak tahu apa yang akan dilakukan terhadap data. Pada penelitian ini akan dilakukan pembuatan aplikasi permainan tradisional Indonesia. Pada perancangan struktur data untuk menyimpan data digunakan, *double circular linked list* dan *binary tree*.

3

Kata Kunci: *binary tree, double circular linked list, permainan tradisional, struktur data*

1. Pendahuluan

Dewasa ini permainan tradisional sudah banyak ditinggalkan. Generasi muda saat ini tidak lagi mengetahui jenis-jenis permainan tradisional yang dimiliki oleh bangsa Indonesia. Hal ini dikarenakan adanya pergeseran budaya dimana teknologi berkembang sangat pesat. Pola pikir, kebiasaan, *trend* masa kini serta peralatan semakin bervariasi. Salah satunya adalah permainan, saat ini sudah mencapai titik yang sangat maju dimana mulai dari console game, jenis dan alirannya sangat banyak dan terus meningkat dari tahun ke tahun mulai dari ide-ide kreatifnya hingga tampilannya yang semakin modern.

Untuk membangkitkan kembali kecintaan masyarakat Indonesia terhadap budayanya khususnya permainan tradisional, maka perlu adanya peremajaan kembali pada jenis-jenis permainan tradisional yang ada. Salah satunya adalah dengan memanfaatkan kemajuan teknologi *video game* dalam permainan tradisional. Peneliti telah melakukan penelitian terhadap beberapa permainan tradisional Indonesia, antara lain Gobak Sodor, Dakon, Petualangan Timun Mas, Induk Ayam Melindungi Anaknya, Bentengan, Panjat Pinang dan Gasing. Setelah dilakukan penelitian tentang cara kerja permainan maka ditentukan struktur data yang tepat untuk penyimpanan dan manipulasi data dalam pembuatan aplikasinya.

Pada *paper* ini dibatasi pada permainan Dakon atau yang juga dikenal dengan nama Congklak dan permainan petualangan Timun Mas. Setelah dilakukan penelitian terhadap cara kerja permainan dan jenis data yang digunakan maka

dipilihlah bentuk struktur data *circular double linked list*[1] dan *tree*[2] sebagai struktur data dalam pembuatan aplikasi.

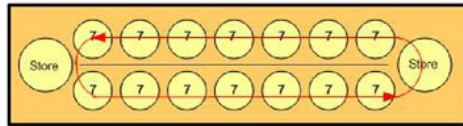
2. Teori Dasar

2.1 Dakon

Dakon atau Congklak adalah suatu permainan tradisional yang dikenal dengan berbagai macam nama di seluruh Indonesia. Permainan congklak dilakukan oleh dua orang. Dalam permainan digunakan papan yang dinamakan papan congklak dan 98 (14 x 7) buah biji yang dinamakan biji congklak atau buah congklak. Umumnya papan congklak terbuat dari kayu dan plastik, sedangkan bijinya terbuat dari cangkang kerang, biji-bijian, batu-batuan, kelereng atau plastik. Pada papan congklak terdapat 16 buah lubang yang terdapat 14 lubang kecil yang saling berhadapan dan 2 lubang besar di kedua sisinya. Setiap 7 lubang kecil di sisi pemain dan lubang besar di sisi kanannya dianggap sebagai milik sang pemain.

Pada awal permainan setiap lubang kecil diisi dengan tujuh buah biji. Dua orang pemain yang berhadapan, salah seorang yang memulai dapat memilih lubang yang akan diambil dan meletakkan satu ke lubang di sebelah kanannya dan seterusnya. Bila biji habis di lubang kecil yang berisi biji lainnya, akan dapat mengambil biji-biji tersebut dan melanjutkan mengisi, bila habis di lubang besar miliknya maka akan dapat melanjutkan dengan memilih lubang kecil di sisinya. Bila habis di lubang kecil di sisinya maka akan berhenti dan mengambil seluruh biji di sisi yang berhadapan. Tetapi bila

berhenti di lubang kosong di sisi lawan maka akan berhenti dan tidak mendapatkan apa-apa. Permainan dianggap selesai bila sudah tidak ada biji lagi yang dapat diambil (seluruh biji ada di lubang besar kedua pemain). Pemenangnya adalah yang mendapatkan biji terbanyak. Bentuk papan dakon dapat dilihat pada Gambar 1.



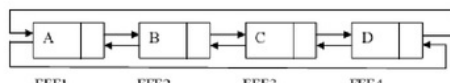
Gambar 1. Papan Dakon

2.2 Cerita Rakyat Timun Mas

Cerita rakyat Timun Mas berasal dari Jawa Tengah. Cerita rakyat Timun Mas ini mengisahkan tentang seorang janda yang mengidam-idamkan seorang anak. Suatu hari, keinginannya tersebut dikabulkan, kala ia didatangi oleh raksasa yang ingin memberi seorang anak dengan sebuah perjanjian. Apabila, anak itu genap berusia enam tahun harus diserahkan kembali kepada si raksasa itu untuk dijadikan makan malam.

2.3 Circular Double Linked List

Linked list adalah bentuk struktur data yang paling banyak digunakan[4]. *Circular Double Linked List* adalah *linked list* dengan menggunakan *pointer*, dimana setiap *node* memiliki tiga *field*, yaitu 1 *field pointer* yang menunjuk *pointer* berikutnya (*next*), 1 *field* menunjuk *pointer* sebelumnya (*prev*), serta sebuah *field* yang berisi data untuk *node* tersebut[2,3,4,5]. *Pointer next* dan *previous* pada *Circular Double Linked List* menunjuk ke dirinya sendiri secara circular. Gambar 2 berikut adalah ilustrasi dari *Circular Double Linked List*.



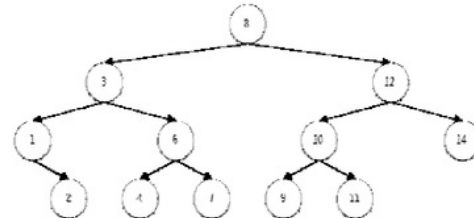
Gambar 2. *Circular Double Linked List*

2.4 Tree

Tree adalah salah satu bentuk struktur data yang digunakan untuk menyelesaikan masalah-masalah tertentu, misalnya menggambarkan struktur organisasi dan struktur direktori, *state solution tree*, *heuristic search*, dsb. Pada *tree* terdapat 2 bagian utama. Yang pertama adalah *root* dan *sub root*. *Root* merupakan kepala dari suatu *tree*, sedangkan *sub root* merupakan 'anak' dari *root*.

Binary tree adalah salah satu bentuk *tree* yang terdiri dari tiga bagian, yaitu *root*, dan dua *sub root* yang masing-masing adalah *binary tree* juga. *Binary*

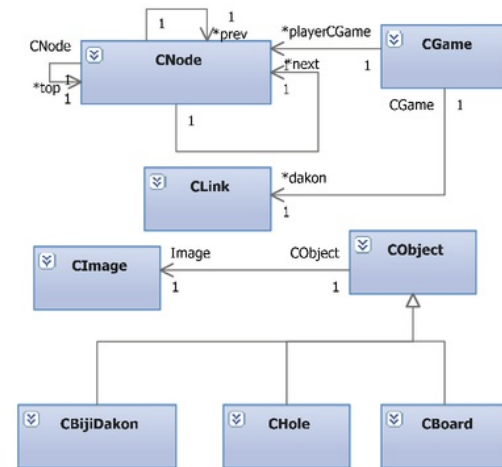
tree juga memungkinkan mempunyai himpunan elemen yang kosong. *Binary search tree* adalah *binary tree* dimana elemen yang bernilai lebih kecil dari *root* diletakkan di sebelah kiri *root*, sedang yang lebih besar di sebelah kanan. [3,4] Gambar 3 berikut ini merupakan bentuk *binary search tree* dengan urutan input data 8, 3, 6, 12, 1, 4, 7, 10, 9, 14, 11, 2



Gambar 3. Class Diagram Dakon

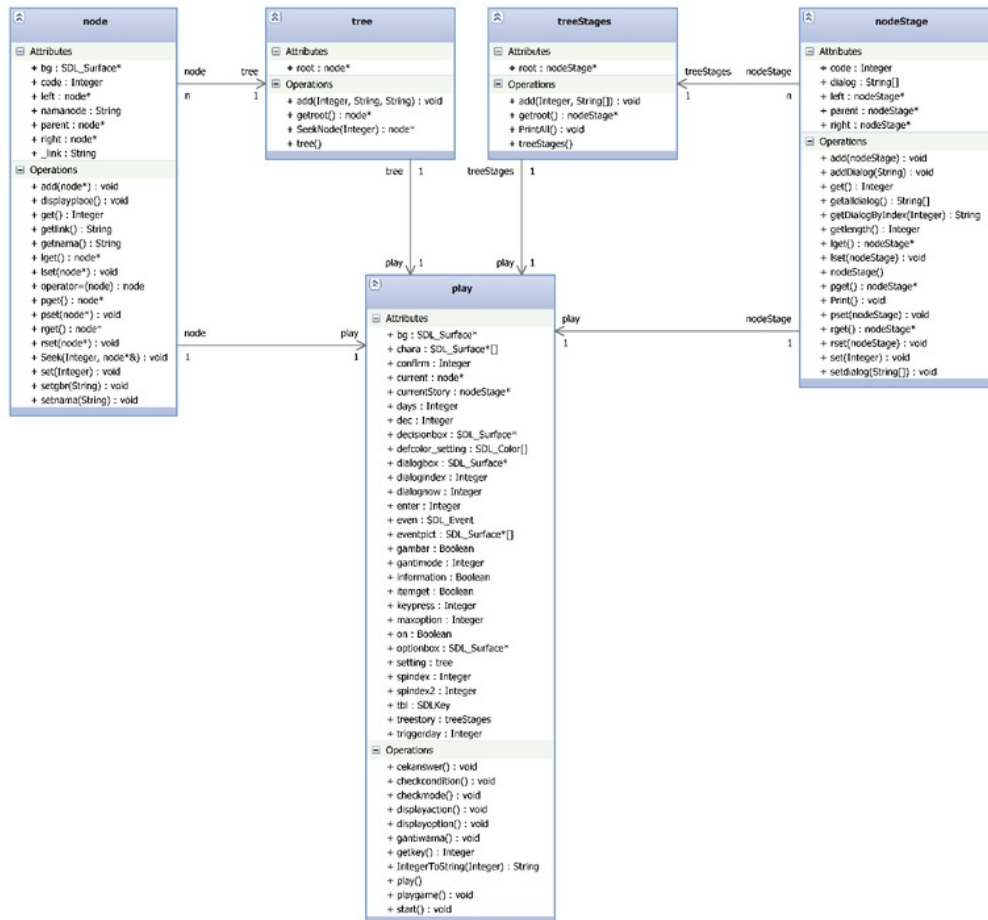
3. Perancangan Sistem

Sistem yang dikembangkan ini menggunakan bantuan SDL [6,7] sebagai *library* untuk menampilkan animasi tiga dimensi pada layar monitor dan menggunakan bahasa pemrograman C++ [8]. Secara garis besar, sistem pembuatan permainan Dakon menggunakan struktur data *circular double linked list* dan terdiri beberapa modul. Class Diagram Dakon dapat dilihat dalam Gambar 4.



Gambar 4. Class Diagram Dakon

Sedangkan untuk sistem pembuatan permainan petualangan Timun Mas menggunakan struktur data *binary tree* dan terdiri beberapa modul. Class Diagram permainan petualangan Timun Mas dapat dilihat dalam Gambar 5.



Gambar 5. *Class Diagram* Petualangan Timun Mas

Berikut keterangan dari semua *class* pada *class Diagram* Petualangan Timun Mas

1. *Class play*

Class play digunakan untuk memproses permainan. Semua *class* terhubung dengan *class* ini. Dalam *class* ini teratur semua *setting* dari permainan seperti besarnya *font* yang digunakan, gambar yang digunakan untuk *home screen*, perintah apa yang dilakukan jika tombol keyboard ditekan. Proses dari cerita dan pengaturan perpindahan *tree* juga diatur dalam *class* ini. *Class play* ini yang dipanggil oleh fungsi main.

2. *Class node*

Class node digunakan sebagai *node* pada *class tree*. Berisi tentang nama tempat dan *link image* untuk menampilkan tempat dan nomor/*id node*.

3. *Class tree*

Class tree digunakan sebagai *tree* pada tempat/lokasi permainan. Pada *constructor tree* akan membaca file yang berisi tentang informasi tempat. *Tree* akan melakukan *generate node* yang untuk tempat yang saling terhubung.

4. *Class nodeStage*

Class nodeStage digunakan sebagai *node* pada *class treeStage*. Berisi tentang dialog yang terdapat pada 1 stage dan nomor/*id nodeStage*.

5. *Class treeStage*

Class treeStage digunakan sebagai *tree* untuk mengontrol *storyline* dari permainan. Pada *constructor treeStage* akan membaca file yang berisi tentang informasi seperti dialog dari setiap *stage*. *TreeStage* akan melakukan *generate nodeStage* yang untuk tempat yang saling terhubung.

Proses perancangan Dakon meliputi:

- Proses inialisasi (*window*, dan obyek *SDL*).
- Pembuatan *class node* untuk menyimpan *data* dakon
- Pembuatan *class list*
- Modul permainan (*game*).

3.1. Proses Inialisasi Permainan Dakon

Pada bagian ini dibuat *class* untuk tampilan antarmuka beserta dengan obyek-obyek *SDL*.

Berikut ini adalah struktur class CImage (Segmen 1) dan class CObject (Segmen 2) untuk semua gambar *object* yang digunakan pada tampilan aplikasi ini seperti papan, lubang dan biji dakon.

Segmen 1. Class CImage

```
class CImage{
private:
    SDL_Texture * _texture;
    SDL_Rect * _bound;
public:
    CImage(const char *path);
    CImage(SDL_Texture *texture);
    SDL_Texture *GetTexture();
    SDL_Rect *GetBound();
    ~CImage();
};
```

Segmen 2. Class CObject

```
class CObject {
protected:
    CImage * _image;
    SDL_Rect * _transform;
    std::vector<CObject*> _children;
public:
    CObject();
    SDL_Rect *GetTransform();
    CImage *GetImage();
    virtual void Update();
    void Render();
    void AddChild(CObject *object);
    CObject *GetChildAt(int index);
    void RemoveChild(CObject *object);
    ~CObject();
};
```

Kemudian dibuat pula beberapa class turunan dari class CObject di atas antara lain class *CBoard* untuk papan dakon, class *CHole* untuk lubang dakon, dan class *CBijiDakon* untuk menyimpan biji dakon.

3.2. Pembuatan Class Node Permainan Dakon

Class Node digunakan untuk menyimpan data dakon antara lain jumlah biji yang ada pada tiap lubang (*_data*), pemain, yang berada pada suatu node (*_player*), nomer lubang (*_nomer*) serta beberapa pointer seperti *next* dan *prev* untuk menghubungkan antar lubang pada dakon, *top* untuk melakukan proses “memakan” biji dakon dari lawan serta *lambung/store* dimana untuk setiap lubang akan ditentukan ketika mendapat biji dakon, akan diletakkan ke *lambung* yang mana.

Class Node dapat dilihat pada segmen 3 berikut ini.

Segmen 3. Class CNode

```
class CNode
{
private:
    int _data;
    CNode *next;
    CNode *prev;
    CNode *top;
    CNode *lambung;
    int _player;
    int _nomer;
    int _tipe;
public:
    CNode();
    //untuk inialisasi pointer pada node
    void setLambung(CNode * _lambung);
    CNode *getLambung();
    void setType(int tipe);
```

```
int getType();
void setNomer(int nomer);
int getNomer();
void setPlayer(int player);
int getPlayer();
void setData(int nilai);
int getData();
void setNext(CNode * _next);
void setPrev(CNode * _prev);
void setTop(CNode * _top);
CNode *getNext();
CNode *getPrev();
CNode *getTop();
};
```

3.3. Pembuatan Class List Permainan Dakon

Pada class ini suatu *object node* akan mengarah ke *object node* lain sehingga banyak *node* dapat dirangkai bersama hanya menggunakan satu *variable* untuk mengakses keseluruhan rangkaian *nodes*. Class ini menyimpan tiga *data member* antara lain pointer *head* untuk menyimpan alamat *object node* yang paling awal, pointer *tail* untuk menyimpan alamat *object node* paling akhir dan pointer posisi untuk menyimpan pointer pada *object node* yang sedang aktif. Segmen 4 berikut untuk pengisian *lambung* dari suatu lubang atau *lambung* pada dakon.

Segmen 4. Fungsi Pengisian Lambung

```
void setLambung()
{
    CNode *temp=head;
    CNode *lambung1;
    9 node *lambung2;
    for(int i=0;i<7;i++)
    {
        temp=temp->getNext();
    }
    11 lambung1=temp;
    lambung2=head->getPrev();
    g=head;
    for(int i=0;i<16;i++)
    {
        if(i<8)
            temp->setLambung(lambung1);
        else
            temp->setLambung(lambung2);
        temp=temp->getNext();
    }
}
```

3.4. Pembuatan Modul Permainan Dakon

Pada bagian ini dibuat modul untuk mengatur jalannya permainan. Class ini dapat dilihat pada Segmen 5. *Contractor* *CGame* akan melakukan inialisasi tiap lubang pada dakon dengan 7 biji dakon. Fungsi *play* akan mengatur jalannya permainan seperti aturan “memakan lawan”, bagaimana menjalankan biji dakon serta pengecekan kapan permainan berakhir.

Segmen 5. Class CGame

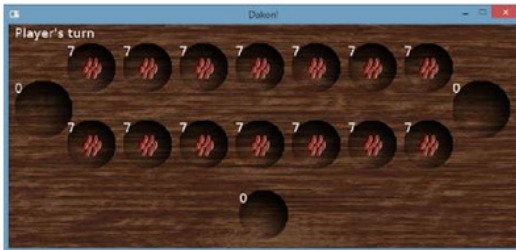
```
class CGame
{
private:
    int _player;
    CLink *dakon;
    CNode *player;
    int _biji;
public:
```

```

CGame();
CLink *GetDakon();
int play();
};

```

13
4. IMPLEMENTASI DAN HASIL PENGUJIAN
 Aplikasi ini dikembangkan menggunakan bahasa C++ dan menggunakan library SDL. Untuk jenis permainannya pada Dakon ada dua, yaitu user melawan user serta user melawan komputer. Gambar 6 berikut ini merupakan tampilan awal dari aplikasi Dakon



Gambar 6. Tampilan Awal Dakon
 Pada Gambar 7 berikut adalah kondisi akhir yaitu *Player menang*.



Gambar 7. Setelah Salah Satu Pemain Menang

Berikut ini adalah pengujian untuk permainan petualangan Timun Mas. Di sini pemain akan memainkan peran sebagai Timun Mas yang harus memilih jalan mana yang harus ditempuh dalam mengalahkan Buto Ijo. Gambar 8 dan 9 adalah contoh tampilan dalam permainan ini



Gambar 8. Tampilan Permainan Timun Mas – Buto Ijo



Gambar 9. Tampilan Permainan Timun Mas – Pilihan Perjalanan

8
4. KESIMPULAN DAN SARAN

Dari hasil penelitian yang dilakukan dapat diketahui bahwa penentuan struktur data yang tepat dalam pembuatan program akan menghasilkan algoritma yang jelas dan tepat sehingga akan membuat program secara keseluruhan lebih efisien dan sederhana. *Circular linked list* dan *binary tree* telah berhasil diimplementasikan dengan baik dalam pembuatan aplikasi pada penelitian ini.

6
DAFTAR PUSTAKA

- [1] Sinha, Prokash. *A Memory-Efficient Doubly Linked List*. Linux Journal Volume 2005 Issue 129. January 2005
- [2] Timnat, Shahr., Anastasia Braginsky, Alex Kogan, Erez Petrank. *Wait-Free Linked Lists*. Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming. February 2012
- [3] Drozdek, Adam. *Data Structures and Algorithms in C++*. USA. Course Technology. 2005
- [4] Harrington, Jan L. *Object Oriented C++ Data Structures for Real Programmers*. Morgan Kaufmann. 1st Edition. 2001
- [5] Nobile, Carl. *Doubly Linked Lists and the Abstract Data Type*. Linux Journal volume 1998 Issue 49. 1998
- [6] <http://www.lazyfoo.net/tutorials/SDL/index.php>
 Akses terakhir 1 Sep 2013
- [7] <http://www.wiki.libsdl.org/SDL1.2Tutorials>
 Akses terakhir 1 Sep 2013
- [8] Chien, Chao C. *Professional Software Development with Visual C++ 6.0 & MFC*. Hingham, Massachusetts : Charles River Media Inc. 2002.

ORIGINALITY REPORT

10%

SIMILARITY INDEX

9%

INTERNET SOURCES

3%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1	www.fraksipkb.com Internet Source	1%
2	www.mnm-team.net Internet Source	1%
3	ebooktake.in Internet Source	1%
4	Submitted to The Hong Kong Polytechnic University Student Paper	1%
5	www.umoncton.ca Internet Source	1%
6	Yao-Tung Tsou, Chun-Shien Lu, Sy-Yen Kuo. "Privacy- and integrity-preserving range query in wireless sensor networks", 2012 IEEE Global Communications Conference (GLOBECOM), 2012 Publication	1%
7	edoc.pub Internet Source	1%

8	docobook.com Internet Source	1%
9	Submitted to Columbia Basin College Student Paper	1%
10	www.linuxjournal.com Internet Source	<1%
11	itfap.ru Internet Source	<1%
12	Submitted to BINUS Interanational School (Simprug) Student Paper	<1%
13	haniknurlaili.wordpress.com Internet Source	<1%
14	ar.scribd.com Internet Source	<1%
15	ja.wikipedia.org Internet Source	<1%
16	Submitted to Laureate Higher Education Group Student Paper	<1%
17	englishinfocusversionwakamadkurikulum.blogspot.com Internet Source	<1%

Exclude quotes On

Exclude bibliography On

Exclude matches < 5 words