Laporan Program ADT Stack

Permainan Blackjack

*Dokumen Spesifikasi Program untuk Tugas Praktikum 8 MK Struktur Data dan Algoritma (Praktik)*



Disusun Oleh :

Kelompok VIII

Fidelis Yugita TB 141511008

M. Muttabi Hudaya 141511022

Senny Hapiffah 141511028

Kelas 1 – A

**Jurusan Teknik Komputer dan Informatika**

**Program Studi D3 Teknik Informatika**

**Politeknik Negeri Bandung**

**2015**

**DAFTAR ISI**

DAFTAR ISI

BAB I SPESIFIKASI PROGRAM

1.1 Deskripsi Program

1.2 Identifikasi Proses

1.3 Identifikasi Data

BAB II PERANCANGAN PROGRAM

2.1 Perancangan Modul

2.2 Perancangan tampilan

BAB III KESIMPULAN

**BAB I**

**SPESIFIKASI PROGRAM**

**1.1 Deskripsi Program**

Program yang kami buat merupakan implementasi dari STACK (tumpukan). Dengan pembuatannya menggunakan ADT. Dimana dalam hal ini kami membuat program permainan Blackjack, dengan aturan sebagai berikut,

1. Banyaknya kartu yang kami gunakan adalah 1 set yang terdiri dari 13 kartu ♦, 13 kartu ♥, 13 kartu ♣ dan 13 kartu ♠ sebanyak 52 kartu.
2. Dalam permainan BlackJack ini, jenis kartu seperti ♦ ♥ ♣ ♠ tidak berpengaruh pada peraimainan. Dalam permainan Black Jack ini, hanya akan memperhitungkan nilai dari kartunya dengan nilai kartu sebagai berikut,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kartu ♦ | | Kartu ♥ | | Kartu ♣ | | Kartu ♠ | |
| Jenis | Nilai | Jenis | Nilai | Jenis | Nilai | Jenis | Nilai |
| AS♦  2♦  3♦  4♦  5♦  6♦  7♦  8♦  9♦  10♦  11♦  12♦  13♦ | = 1 atau 11  = 2  = 3  = 4  = 5  = 6  = 7  = 8  = 9  = 10  = 11  = 12  = 13 | AS♥  2♥  3♥  4♥  5♥  6♥  7♥  8♥  9♥  10♥  11♥  12♥  13♥ | = 1 atau 11  = 2  = 3  = 4  = 5  = 6  = 7  = 8  = 9  = 10  = 11  = 12  = 13 | AS♣  2♣  3♣  4♣  5♣  6♣  7♣  8♣  9♣  10♣  11♣  12♣  13♣ | = 1 atau 11  = 2  = 3  = 4  = 5  = 6  = 7  = 8  = 9  = 10  = 11  = 12  = 13 | AS♠  2♠  3♠  4♠  5♠  6♠  7♠  8♠  9♠  10♠  11♠  12♠  13♠ | = 1 atau 11  = 2  = 3  = 4  = 5  = 6  = 7  = 8  = 9  = 10  = 11  = 12  = 13 |

1. Banyaknya pemain yang dapat bermain adalah dari 1 pemain sampai 3 pemain, dengan pemain tambahan banker sebagai titik pusat permainan.
2. Pada awal permainan, masing-masing pemain beserta banker akan diberikan 2 kartu secara acak dan sejumlah uang untuk dipertaruhkan.
3. Masing-masing pemain akan diberi kesempatan untuk menambah kartu yang akan ditaruhkan.
4. Masing-masing pemain, dapat mempertaruhkan sejumlah uang yang akan ia taruhkan untuk mendapatkan uang sebanyak 2 kali lipat dari yang ia taruhkan jika menang. Jika pemain kalah, maka sejumlah uang yang telah ditaruhkan tidak dapat dikembalikan kembali.
5. Untuk pemain yang memiliki jumlah nilai kartu paling besar dan jumlah nilai kartu tidak lebih dari 21, akan menjadi pemenangnya. Sedangkan untuk pemain yang jumlah nilai kartunya lebih dari 21 maka kartu pemain akan terbakar dan kalah.
6. Semua pemain dinyatakan kalah jika banker memiliki jumlah nilai kartu paling besar dan jumlah nilai kartu tidak labih dari 21.

**1.2 Identifikasi Proses**

Dalam permainan Blackjack ini, memiliki beberapa proses utama, diantaranya,

1. Proses user memilih jumlah pemain dengan memakai modul **void player(int \*n);**. Dimana user akan memilih banyaknya pemain yang akan bermain dalam permainan Blackjack ini.
2. Proses mengocok kartu dalam deck dengan memakai modul **void shuffle(address \*deck, int jmlKartu);**. Pada proses ini, modul akan melakukan proses pengocokan kartu dalam deck sejumlah banyaknya kartu yang dibutuhkan. Dalam permainnan ini kami membutuhkan 52 kartu.
3. Proses membagi kartu dengan memakai modul **void bagiKartu(address \*deck, address pemain);**. Dimana dalam proses ini, modul akan digunakan untuk membagikan kartu yang terdapat dalam deck kepada setiap pemain dan tentu saja banker.
4. Proses bermain bergiliran dengan memakai modul **void turn(address \*deck, address \*pemain, bool \*tambah, int pos);** untuk giliran bermain pada pemain.
5. Proses tambah kartu dengan memakai modul **void lagi(bool \*tambah, int pos);**. Dalam proses ini, segara bergiliran pemain akan diberikan kolom pertanyaan apakah pemain akan menambah kartu ataupun tidak, untuk memperbesar kesempatan menang agar mendekati jumlah nilai kartu 21.
6. Proses perhitungan dengan memakai modul **void Count(address Player);**. Dalam proses ini, modul akan melakukan perhitungan jumlah nilai kartu pada banker dan setiap pemain.
7. Proses mencari pemenang dengan memakai modul **void Winner(address Player, address Banker, int i);**. Dalam proses ini modul akan digunakan untuk mencari pemenang dengan memanggil modul **void Count(address Player);** dan mencari jumlah nilai kartu yang paling besar sebagai pemenangnya. Jika banker yang menang, maka semua pemain dinyatakan kalah.

**1.3 Identifikasi Data**

Dalam permainan Blackjack ini, kami banyak menggunakan variabel lokal dengan passing parameter, tetapi kamu juga menggunakan data utama, sebagai berikut

1. Array komposit digunakan sebagai penampung utama kartu yang terdiri dari subvar isi bertype integer dan next bertype pointer, seperti,

typedef struct kartu \*address;

typedef struct kartu{

int isi;

address next;

}KARTU;

1. Deck bertype array komposit seperti data yang diatas. Menampung data dari Deck seperti jumlah kartu, nilai kartu dan jenis kartu.
2. Pemain bertype array komposit seperti data yang diatas. Menampung data dari Pemain yaitu kartu yang dimiliki pemain.
3. Banker bertype array komposit seperti data yang diatas. Menampung data dari Banker yaitu kartu yang dimiliki banker.

**BAB II**

**PERANCANGAN PROGRAM**

**2.1 Perancangan Modul**

Dalam membuat program permainan Blackjack ini, kami menggunakan beberapa modul, diantaranya

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Modul** | **Deskripsi** | **Jenis** | **Parameter** | **Kamus Data (Lokal)** |
| 1. | **void push (address \*p,int isi);** | Modul untuk menambahkan isi stack. Dengan menggunakan prinsip insert akhir. | Prosedur | Address \*p menggunakan passing parameter by address  Int isi menggunakan passing parameter by value | pNew bertype pointer |
| 2. | **void pop (address \*p,int \*isi);** | Modul untuk menghapus isi stack. Dengan menggunakan prinsip delete akhir. | Prosedur | Address \*p menggunakan passing parameter by address  Int isi menggunakan passing parameter by value | pDel bertype pointer |
| 3. | **void clear();** | Modul untuk menghapus beberapa bagian dari tampilan | Prosedur |  |  |
| 4. | **address alokasi(int isi);** | Modul untuk mengalokasikan data pada memori | Fungsi | Passing parameter by address |  |
| 5. | **void insertAfter (address \*p,int isi);** | Modul untuk menginsertkan data diakhir. | Prosedur | Address \*p menggunakan passing parameter by address  Int isi menggunakan passing parameter by value | pNew bertype pointer |
| 6. | **void menu();** | Modul untuk menampilkan menu permainan | Prosedur |  |  |
| 7. | **void gotoxy(int x, int y);** | Modul untuk meletakkan suatu output sesuai koordinat (x,y) | Prosedur | Passing parameter by value |  |
| 8. | **void player(int \*n);** | Modul untuk menentukan jumlah pemain yang akan bermain. | Prosedur | Passing parameter by value | n bertype integer |
| 9. | **void shuffle(address \*deck,int jmlKartu);** | Modul untuk mengocok kartu pada deck | Prosedur | Address \*deck passing parameter by addess  Int jmlKartu passing parameter by value | i, j, pos bertype integer  p bertype pointer |
| 10. | **void Banker(address \*deck,address \*Banker);** | Modul untuk banker saat bermain yang menyimpan kartu yang telah dibagikan oleh deck | Prosedur | Address \*deck passing parameter by address  Address \*Banker passing parameter by address | j bertype integer |
| 11. | **void bagiKartu(address \*deck,address \*pemain);** | Modul untuk membagikan kartu yang sudah dikocok dari deck kepada masing-masing pemain | Prosedur | Passing parameter by address | Isi bertype integer |
| 12. | **void turn(address \*deck,address \*pemain,bool \*tambah,int pos);** | Modul untuk bergiliran menambah kartu | Prosedur | Address \*deck passing parameter by address  Address \*pemain passing parameter by address  Bool \*tambah passing parameter by address  Int pos passing parameter by value |  |
| 13. | **void lihat(address p,int pos,int kartuKe,bool tutup);** | Modul untuk menampilkan kartu-kartu | Prosedur | Address p passing parameter by address  Int pos dan int kartu passing parameter by value | i bertype integer |
| 14. | **void tampKartu(int posisi,int kartuKe,int nilai,bool tutup);** | Modul untuk menampilkan kartu | Prosedur | Passing parameter by value |  |
| 15. | **void tampIsi(int nilai);** | Modul untuk menampilkan nilai dari kartu | Prosedur | Passing parameter by value | jenis, isi bertype integer |
| 16. | **void lagi(bool \*tambah,int pos);** | Modul untuk menambah kartu jika | Prosedur | bool \*tambah passing parameter by address  int pos passing parameter by value | Kursor bertype integer  End bertype boolean |
| 17. | **int Count(address Player);** | Modul untuk menghitung jumlah nilai kartu pemain | Prosedur | Passing parameter by address | Temp dan cekAs bertype pointer  Jumlah bertype integer |
| 18. | **void Winner(address Player,address Banker,int i);** | Modul untuk menentukan pemenang | Prosedur | Address player dan address player passing parameter by address  Int i passing parameter by value |  |

**2.2 Perancangan tampilan**

Menu Utama

Keluar

Mulai

3 Player

2 Player

1 Player

Papan main

Mulai lagi

Keluar

Kalah

Menang

**BAB III**

**KESIMPULAN**

Dalam proses pembuatan permainan Blackjack menggunakan ADT Stack ini, kami sudah sejauh menyelesaikan permainan tanpa ada proses taruhan. Kami sudah dapat menentukan pemenang dari permainan Blackjack ini. Hal yang kami belum selesaikan adalah memakai proses taruhan dalam permainan Blackjack ini, agar penentuan permain dapat menurut jumlah uang yang dimiliki juga.