**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Dalam suatu proses belajar untuk mengukur tingkat keberhasilan adalah dengan adanya suatu tugas / Project. Dalam mata kuliah pemrograman dasar 2 ini diberikan berbagai macam fungsi bahasa C untuk memecahkan permasalah matematika ataupun permasalah yang lain. Beberapa contoh fungsi nya seperti : Tree, Hashing, Rekursif, Searching, Sorting dan Fungsi lainya. Terdapat pula metode dalam pendataan seperti Linked List, dan Graph.

Dalam mengukur keberhasilan mata kuliah Pemrograman Dasar 2 ini diberikan sebuah Tugas / Project untuk membuat sebuah game sederhana dengan memanfaatkan fungsi – fungsi yang telah dipelajari sebelumnya dalam mata kuliah pemrograman dasar 2. Dengan tugas membuat game ini diharapkan mahasiswa dapat mengimplementasikan mata kuliah yang telah dipelajari. Pada tugas ini kami membuat sebuah game *Rush & Defence* yang memanfaatkan fungsi stack dan double linked list sebagai jalannya program. Permainan ini bergenre Turn Based Strategy, dimana pemain dapat memberikan perintah pada pasukan untuk menyerang musuh.

* 1. **Rumusan masalah :**

1. Bagaimana mengimplementasikan fungsi – fungsi bahasa C untuk membuat sebuah program ?
2. Bagaimana membangun program game *Rush & Defence* dengan memanfaatkan fungsi stack dan double linked list?
   1. **Tujuan**
3. Mahasiswa mampu memahami dan mengerti fungsi untuk membangun program dengan bahasa C secara standart.
4. Mahasiswa Mampu mengimplementasikan fungsi Stack dan double link list untuk membangun sebuah game sederhana.

**BAB II**

**PEMBAHASAN PROGRAM**

* 1. **Dasar Teori**

**2.1.1 Stack**

Salah satu konsep yang efektif untuk menyimpan dan mengambil data adalah "terakhir masuk sebagai yang pertama keluar" (Last In First Out / LIFO). Dengan konsep ini, pengambilan data akan berkebalikan urutannya dengan penyimpanan data. Stack adalah sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO. Dengan demikian, elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas dari stack, maka kita melakukan push. Dan untuk memindahkan dari tempat yang atas tersebut, kita melakukan pop.

Ada beberapa cara untuk menyajikan sebuah stack tergantung pada permasalahan yang akan kita selesaikan. Dalam bab ini kita akan menggunakan cara yang paling sederhana, tipe data yang sudah kita kenal, yaitu array. Kita dapat menggunakan array untuk menyajikan sebuah stack, dengan anggapan bahwa banyaknya elemen maksimum dari stack tersebut tidak akan melebihi batas maksimum banyaknya elemen dalam array. Pada saat ukuran stack, kalau kita teruskan menambah data lagi, akan terjadi overflow. Dengan demikian perlu data tambahan untuk mencatat posisi ujung stack

**2.1.2 Relasional Operator**

Relasional operator adalah operator yang dapat digunakan untuk membandingkan antara dua operand, misalkan lebih besar mana antara a dan b, manakah yang lebih kecil antara c dan 12. Dari perbandingan diatas maka akan didapatkan suatu hasil yaitu benar atau salah (true or false) yang berupa nilai 1 (selain nol) untuk benar dan 0 untuk salah.

Statemnet if-else digunakan untuk menentukan pilihandari suatu kondisi yang diberikan. Cara kerjanya adalah menggunakan perbandingan dengan menggunakan Relational Operator dan/atau Logical Operator, apabila suatu kondisi terpenuhi (benar) maka akan mengerjakan proses A dan apabila kondisi tidak terpenuhi (salah) maka tidak akan mengerjakan proses Aatau dapat juga mengerjakan proses lain misalkan proses B. proses A dan B dapat berupa satu statement atau beberapa baris yang dikelompokkan (compound statement). Statement if-else bersarang merupakan kombinasi dari beberapa if-else, dinama bila ada beberapa kondisisi yang harus diuji kebenarannya secara bersama (if dalam if). Statement if-else adalah cara lain dari bentuk if-else, yang digunakan untuk menentukan suatu kondisi yang benar dari beberapa kondisi yang tersedia. Statement swith-case adalah bentuk lain dari statement else-if, dengan mendaftar kondisi secara vertical dalam suatu kolom sehingga memudahkan dalam hal evalusi program.

### 2.1.3 Control Statement

Control Statement adalah statement yang digunakan untuk mengarahkan jalannya program, sehingga urutan pengerjaan program dapat melompat – lompat atau berulang. Ada beberapa statement yang termasuk control statement yaitu statement if-else, else-if,goto ( ),switch ( ),break, continue, for( ),while ( ) dan do-while. Statement for ( ) digunakan untuk membuat perulangan yang dikontrol oleh nilai – nilai tertentu yaitu initial ( nilai awal), condition (kondisi), increment/decrement (penambahan/penurunan). Perulangan dengan statement for ( ), tidak memeriksa kondisi terlebih dulu, kemudian baru memeriksa kondisi. Statement while ( ) digunakan juga untuk membuat perulangan seperti pada statement for ( ), perbedaannya adalah statement while ( ) memeriksa kondisi dulu, bila kondisi memenuhi (benar), maka perulangan dilaksanakan. Statement do-while adalah juga digunakan untuk membuat perulangan seperti halnya dua statement diatas, dan kemungkinan merupakan kombinasi antara kedua statement diatas, cara kerja dari perulangan do-while adalah langsung mengerjakan badan loop, kemudian kondisi diperiksa bila kondisi memenuhi perulangan dilanjutkan bila tidak perulangan selesai. Dari semua perulangan dapat dibuat perulangan bersarang dan tak hingga.

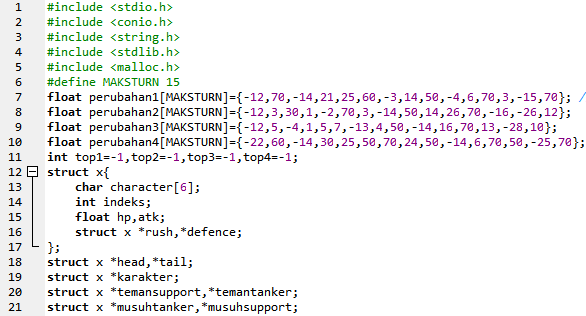
**2.1.4 Double Linked List**

Secara konseptual, linked list merupakan deretan elemen yang berdampingan. Akan tetapi, karena elemen-elemen tersebut dialokasikan secara dinamis (menggunakan malloc), sangat penting untuk diingat bahwa kenyataannya, linked list akan terpencarpencar di memori. Pointer dari elemen ke elemen berarti sebagai penjamin bahwa semua elemen dapat diakses. Single linked list atau biasa disebut linked list terdiri dari elemen-elemen individu, dimana masing-masing dihubungkan dengan pointer tunggal. Masing-masing elemen terdiri dari dua bagian, yaitu sebuah data dan sebuah pointer yang disebut dengan pointer next. Dengan menggunakan struktur two-member seperti ini, linked list dibentuk dengan cara menunjuk pointer next suatu elemen ke elemen yang mengikutinya. Pointer next pada elemen terakhir merupakan NULL, yang menunjukkan akhir dari suatu list. Elemen pada awal suatu list disebut head, dan elemen terakhir dari suatu list disebut tail.

**2.2 Alur Program *Rush & Defence***

Rush & Defencemerupakan sebuah program permainan yang mana dalam game ini memanfaatkan fungsi dan banyaknya variable-variable yang ada didalam struct yang didata dengan konsep linked list.

1. Peraturan permainan
2. Menggunakan num key dalam navigasi menu.
3. Nama karakter yang diinputkan maksimal sebanyak 7
4. Critical chance bergantung pada tipe class yang dipilih.
5. Tanker tidak bisa menyerang support, tetapi sebaliknya untuk support.
6. Semi real time, musuh dapat mengcounter serangan.
7. Bila salah satu nyawa dari karakter kurang atau sama dengan nol, maka player akan kalah.
8. Algoritma program
9. Struct diinisialisasikan, didalam struct ini terdapat variable hitpoint, attack point, name dari karakter.
10. Variable-variable ini dapat diisikan dengan fungsi pemilihan karakter.
11. Setelah proses input data selesai. Program menawarkan menu dalam battle
12. Dalam opsi serang terdapat animasi pertempuran dalam arena perang yang disebut dengan battlefield.
13. Setelah dilakukan penyerangan, program akan mengakumulasi damage berdasarkan pengolahan yang diperoleh dari fungsi pop() yang diberikan dan diterima oleh player.
14. Source Code
15. Deklarasi

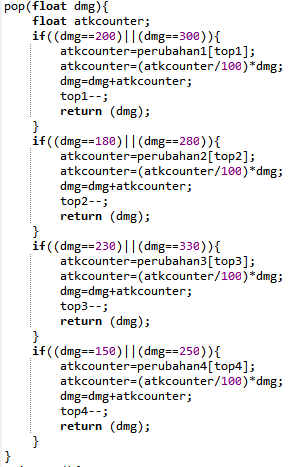


Pendeklarasian dilakukan agar memudahkan dalam pembuatan program. Pemberian damage pada array-array diatas disesuaikan dengan pemilihan damage.

Tipe data yang dipakai adalah float, hal ini digunakan karena tipe variable ini memiliki kelebihan, yaiut dapat menghitung nilai pecahan, jadi data yang dihasilkan akan lebih presisi.

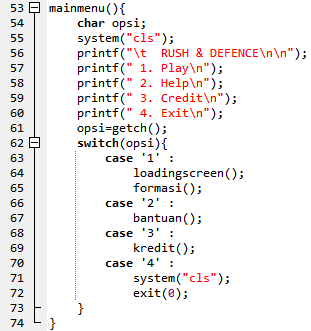
Pointer seperti \*rush dan \*defence digunakan untuk merangkai data menjadi layaknya double linked list. \*head dan \*tail untuk posisi akhir. Sedangkan pointer lainya digunakan untuk pencaraian karakter.

1. Fungsi pop



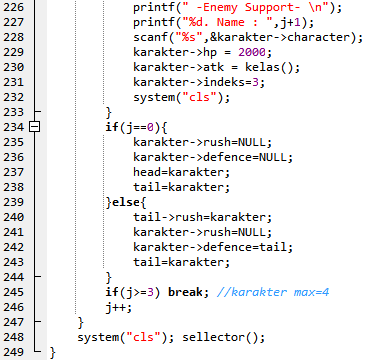
Fungsi ini merupakan fungsi yang digunakan utuk mengeluarkan data yang telah ada pada array perubahan[]. Disini dilakukan pengkondisian terhadap input damage yang dihitung secara presisi untuk mengeluarkan nilai damage yang unik.

1. Main Menu



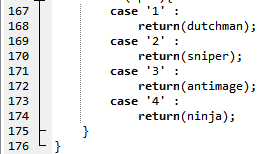
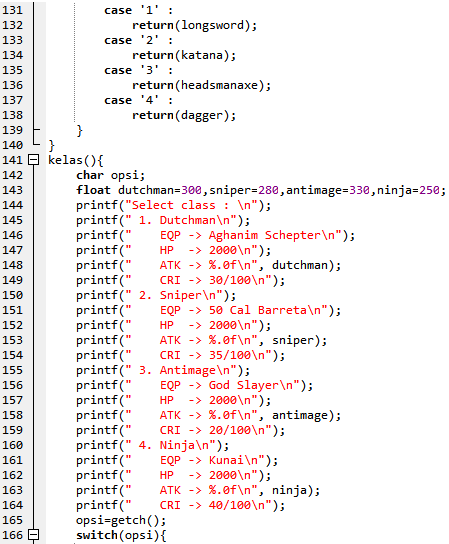
Fungsi ini merupakan fungsi menu yang ditampilkan pertama kali game dimulai. Dimana menu yang terdapat disini dapat melakukan *switching* ke fungsi lain.

1. Fungsi inisialisasi data



Fungsi ini merupakan fungsi untuk menginiliasisaikan data dalam struct. Dimana data diinputkaan satu-persatu lalu kemudian dirangkai membentuk double linked list.

1. Fungsi Input Class



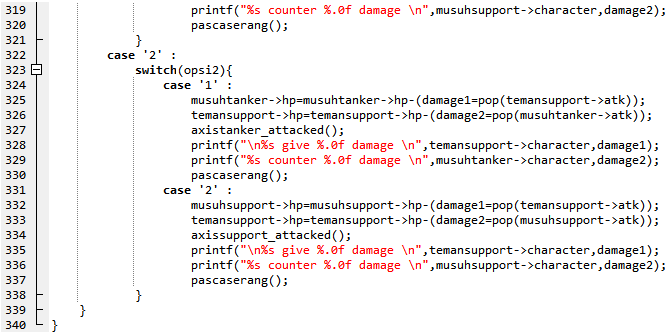
Fungsi-fungsi ini merupakan fungsi yang digunakan untuk menginputkan data untuk selanjutnya disimpan dalam variable struct.

1. Fungsi penampilan status



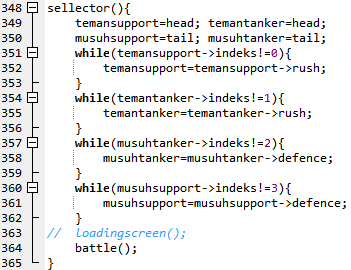
Fungsi rekursi ini merupakan fungsi menampilkan status setiap karakter berdasarkan poiter penghubung terdekatnya.

1. Fungsi serang



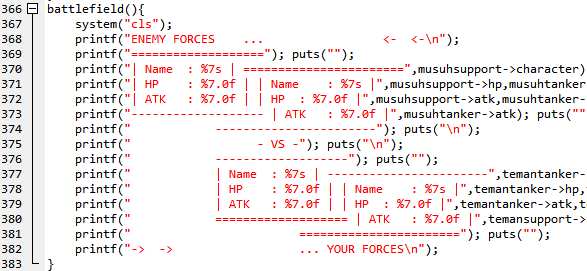
Fungsi ini merupakan *workhouse* dalam menentukan mana player akan dipilih, dan player mana yang akan diserang, begitu pula proses pengurangan *hitpoint* dalam masing-masing karakter

1. Fungsi selector



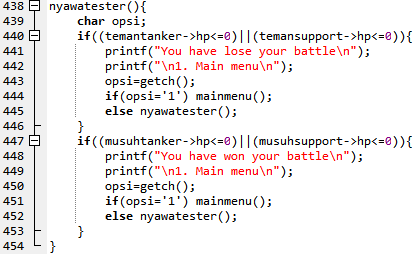
Fungsi ini digunakan untuk menandakan atau menempatkan pointer dalam suatu simpul, sehingga program tidak perlu mencari data lagi secara berulang-ulang.

1. Fungsi *battlefield*



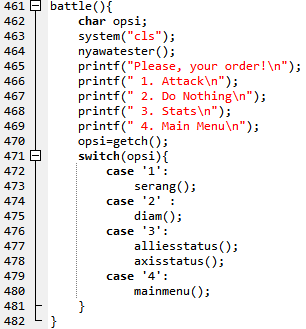
Fungsi ini berisikan atau menampilkan posisi dari setiap karakter sehingga terlihat animasi atau grafis dari game ini.

1. Fungsi kondisi menang



Pada fungsi ini, diadakan kondisi untuk menentukan apakah player menang atau tidak, konsepnya adalah bila *hitpoint* salah satu karakter kurang dari atau sama dengan nol, maka dapat ditentukan siapa yang memenagkan permainan.

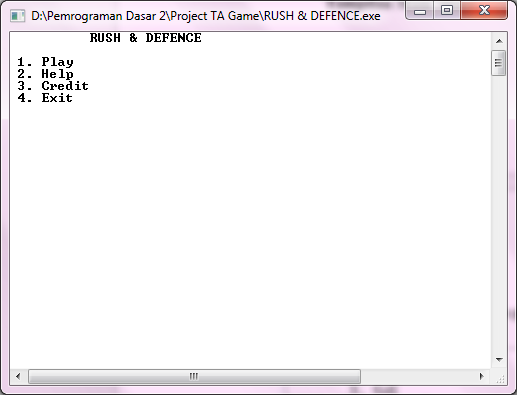
1. Fungsi strategi



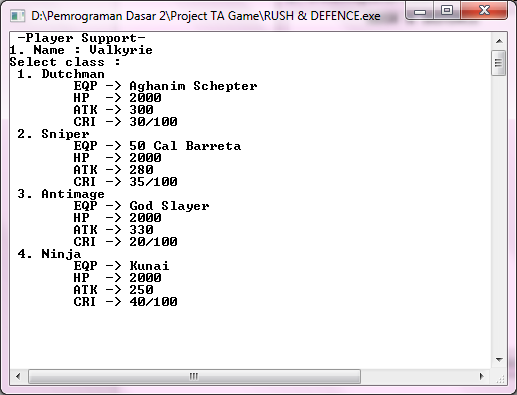
Fungsi ini berisikan pemilihan bagaimana player akan bertindak. Sehingga player dapat menyusun strategi terlebih dahulu.

* 1. **Tampilan Game**

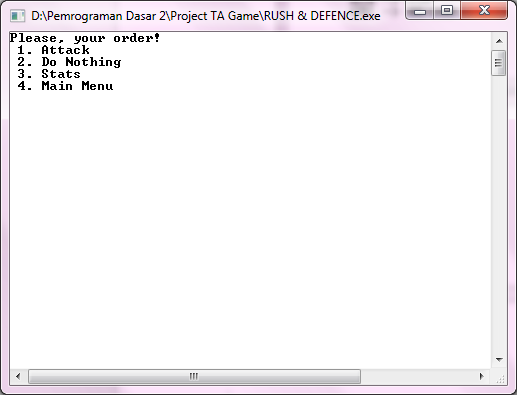
**Tampilan Main Menu**



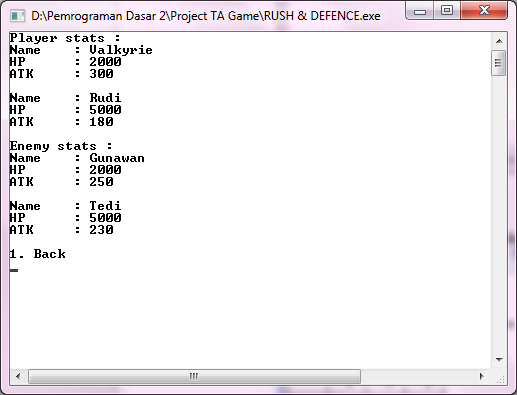
**Tampilan Input data**



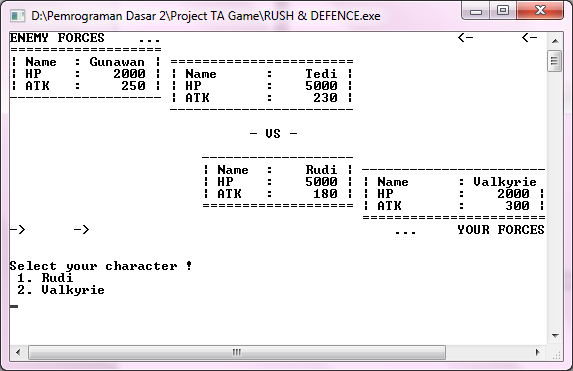
**Tampilan Strategi *Battle***



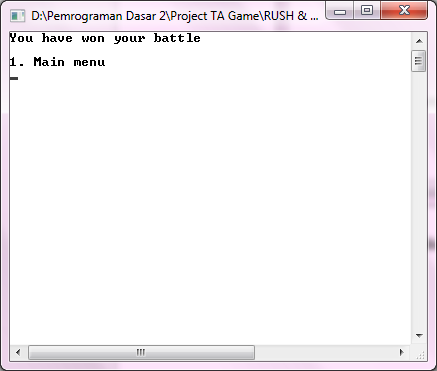
**Tampilan Status**



**Tampilan *Battlefield***



**Tampilan kondisi menang atau kalah**



**BAB III**

**PENUTUP**

* 1. **Kesimpulan**

Game Rush & Defence merupakan sebuah game turn based strategy yang mengasah otak. Dimana dalam game ini pemain diharuskan untuk menyusun strategi dalam menyelesaikan peperangan. Didalam program game ini diberikan fitur input data, yaiut player dapat memilih class yang diinginkan. Selanjutnya player dapat memeriksa kondisi setiap pemain dengan menu status. Player dapat melakukan penyerangan kepada musuh diikuti dengan counter berdasarkan base damage yang telah diolah terlebih dahulu. Hal ini membuat sistem strategi ini menjadi *semi real time*. Player sendiri bisa mengan dengan membuat *hitpoint* musuh menjadi nol. Semua hal ini didata dengan struktur data melalui konsep linked list, dimana kondisi logika dibutuhkan untuk membuat antar menu yang menghubungkan antar strategi.

DAFTAR PUSTAKA

1. *Materi Pemrograman Dasar I*
2. *Materi Pemrograman Dasar II*
3. *www.stackoverflow.com/make-a-clearscreen/*
4. *www.stackoverflow.com/make-a-loop-condition/*