

## Wordament-0



Link video : <http://www.youtube.com/watch?v=AMgEHC2Ljvo>

### Deskripsi Wordament :

Wordament merupakan permainan kata yang dimainkan dalam matriks/board 4x4 yang hanya berisi huruf 'A' sampai 'Z'. Pemain dapat membentuk suatu kata dengan cara memulai dari salah satu petak, kemudian meneruskannya ke sekeliling petak tersebut. Setiap petak dalam matriks hanya dapat dipakai maksimal satu kali untuk membentuk satu kata. Kemudian, kata yang dibentuk akan dicek ke dalam kamus kata. Jika kata tersebut ditemukan dalam kamus kata, maka pemain akan mendapatkan skor yang bergantung pada huruf penyusunnya dan panjang kata tersebut. Kata yang pernah dibentuk sebelumnya pada permainan yang sama tidak akan mendapatkan skor.

### Deskripsi Soal :

Dalam tugas besar ini, Anda diminta untuk membuat program untuk mensimulasikan Wordament yang dinamakan Wordament-0 (dibaca : Wordament-Zero). Wordament-0 disimulasikan dalam bahasa C dengan menggunakan command line dan input dari keyboard ditambah dengan sebuah kursor/penunjuk petak matriks. Kursor dapat digerakkan ke delapan arah dengan menekan tombol q, w, e, a, d, z, x, atau c pada keyboard. Tombol s digunakan untuk memulai atau mengakhiri pembentukan kata. Tombol r digunakan untuk memutar board 90 derajat CW (clock wise).

Saat memulai pembentukan kata, maka huruf pada petak yang sedang ditunjuk kursor dimasukkan ke dalam stack. Jika petak tersebut sudah pernah dikunjungi, maka pop stack hingga top stack tersebut sama dengan petak tersebut. Pada proses pembentukan kata, program juga harus menampilkan kata yang sedang diproses.

**Contoh input program :**

Keterangan	Board																
Inisialisasi board dan kursor (pada petak kiri atas). Isi stack : kosong Kata yang sedang diproses : tidak ada	<table><tr><td>&gt;A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	>A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
>A	B	C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol d, maka kursor akan digerakkan ke kanan. Isi stack : kosong Kata yang sedang diproses : tidak ada	<table><tr><td>A</td><td>&gt;B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	A	>B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	>B	C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol s, maka pembentukan kata pun dimulai dari petak kursor berada. Isi stack : B Kata yang sedang diproses : B	<table><tr><td>A</td><td>&gt;B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	A	>B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	>B	C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol d, maka kursor digerakkan ke kanan dan memroses huruf pada petak tersebut. Isi stack : B C Kata yang sedang diproses : BC	<table><tr><td>A</td><td>*B</td><td>&gt;C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	A	*B	>C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	*B	>C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol x, maka kursor digerakkan ke bawah dan memroses huruf pada petak tersebut. Isi stack : B C G Kata yang sedang diproses : BCG	<table><tr><td>A</td><td>*B</td><td>*C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>&gt;G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	A	*B	*C	D	E	F	>G	H	I	J	K	L	M	N	O	P
A	*B	*C	D														
E	F	>G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol q, maka kursor digerakkan ke kiri atas dan memroses huruf pada petak tersebut. Karena petak tersebut sudah pernah dikunjungi sebelumnya (yang ditandai dengan * pada contoh), maka stack di pop hingga top stack sama dengan huruf pada petak tersebut. Isi stack : B Kata yang sedang diproses : B	<table><tr><td>A</td><td>&gt;B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	A	>B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	>B	C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol a, maka kursor digerakkan ke kiri dan memroses huruf pada petak tersebut. Isi stack : B A Kata yang sedang diproses : BA	<table><tr><td>&gt;A</td><td>*B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	>A	*B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
>A	*B	C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														
Pemain menekan tombol s, maka pembentukan kata pun selesai. Kata yang dibentuk adalah BA.	<table><tr><td>&gt;A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>P</td></tr></table>	>A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
>A	B	C	D														
E	F	G	H														
I	J	K	L														
M	N	O	P														

Pemain menekan tombol r, maka board diputar 90 derajat CW. Kursor akan mengikuti perputaran sehingga tetap menunjuk petak yang sama.

M	I	E	>A
N	J	F	B
O	K	G	C
P	L	H	D

Setelah selesai membentuk kata, kata tersebut akan dicek pada list kata yang sudah dibentuk. Jika kata tersebut belum pernah dibentuk, maka kata tersebut selanjutnya akan dicek pada kamus kata. Jika kata tersebut ada di dalam kamus kata, maka pemain akan mendapatkan skor yang dapat dihitung dengan rumus :

skor = total poin seluruh huruf pada kata tersebut \* panjang kata + bonus kata panjang

Setiap huruf memiliki poin yang berbeda-beda seperti pada tabel berikut :

Huruf	Poin
E	1
A,I,N,O,R,S,T	2
C,D,L	3
G,H,M,P,U	4
B,F,Y	5
K,V,W	6
Z	8
X	9
J,Q	10

Bonus kata panjang diberikan khusus untuk kata yang lebih atau sama dengan 10 huruf. Besar bonus adalah 100 poin.

Durasi satu kali permainan Wordament-0 adalah dua menit. Jika permainan sudah melewati durasi, maka permainan akan diakhiri. Karena itu, selama permainan berlangsung, program harus dapat menerima input secara non-blocking. Berikut dilampirkan contoh source code untuk menerima non-blocking input untuk platform Linux:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/select.h>
#include <termios.h>
#include <time.h>

static struct termios old_termios, new_termios;

/* restore new terminal i/o settings */
void resetTermios(){
    tcsetattr(0,TCSANOW,&old_termios);
}

/* initialize new terminal i/o settings */
void initTermios(){
```

```
tcgetattr(0,&old_termios); // store old terminal
new_termios = old_termios; // assign to new setting
new_termios.c_lflag &= ~ICANON; // disable buffer i/o
new_termios.c_lflag &= ~ECHO; // disable echo mode
tcsetattr(0,TCSANOW,&new_termios); // use new terminal setting
}

/* detect keyboard press */
int kbhit(){
    struct timeval tv = {0L,0L};
    fd_set fds;
    FD_ZERO(&fds);
    FD_SET(0,&fds);
    return select(1,&fds,NULL,NULL,&tv);
}

/* read 1 character */
char getch(){
    char ch;
    ch = getchar();
    return ch;
}

/* skeleton program for play */
void Play(double seconds){
    initTermios(); // initialize new terminal setting to make kbhit() and getch() work
    char cc;
    const double TIME_LIMIT = seconds * CLOCKS_PER_SEC;
    clock_t startTime = clock();
    while ((clock() - startTime) <= TIME_LIMIT){
        if (kbhit()){
            cc = getch();
            resetTermios(); // reset terminal setting to enable buffer i/o and echo (printf)
            printf("Pressed %c\n",cc); // process character
            initTermios(); // use new terminal setting again to make kbhit() and getch() work
        }
    }
    printf("\nTime Up\n");
    resetTermios(); // restore default terminal setting
}

int main(){
    Play(120); // play with time limit 120 seconds (2 minutes)
    return 0;
}
```

Selama permainan berlangsung, program juga harus dapat menampilkan nama user yang sedang bermain, skor yang sudah didapatkan user pada permainan tersebut, sisa waktu permainan tersebut, dan suggestion kata. Suggestion kata terdiri atas kata-kata yang sudah pernah dibentuk oleh user pada board tersebut dalam permainan-permainan sebelumnya dan belum dibentuk dalam permainannya kali ini. Suggestion kata ditampilkan satu per satu dan hanya akan berubah saat user tersebut membentuk kata yang di-suggest.

Saat permainan selesai, program akan menampilkan skor akhir yang didapatkan dan semua kata yang berhasil mendapatkan skor (terurut mulai dari kata yang mendapatkan skor terbesar) beserta skornya.

**Fitur lainnya :**

*1. Main Menu*

Tampilan program saat pertama kali dijalankan atau sesudah user logout. Berisi judul program dan pilihan command yang dapat dilakukan user sebagai berikut :

- a. Register : untuk mendaftarkan user baru. Nama user harus unik dan hanya terdiri dari karakter alfanumerik.
- b. Login : untuk mulai bermain sebagai user tertentu. Nama user harus terdaftar.
- c. How to play : berisi cara memainkan program
- d. About : berisi identitas pembuat program
- e. Quit : keluar dari program dan menyimpan daftar user, highscore, dan hal-hal yang diperlukan ke dalam file eksternal. Saat program dijalankan kembali, file eksternal ini akan di-load.

*2. Preparation Menu*

Tampilan program sesudah user login. Berisi pilihan command yang dapat dilakukan user sebagai berikut :

- a. Play Game : mulai bermain Wordament-0 dengan board yang sudah dipilih.
- b. Select Board : untuk memilih nomor board yang digunakan dalam permainan. Default menggunakan board 1.
- c. View My Highscore : untuk melihat 10 skor tertinggi pada board yang sedang dipilih yang didapatkan user tersebut. Tampilan high score akan menampilkan skor secara terurut mengecil dan waktu(tanggal+jam) pencapaian skor tersebut. Jika ada lebih dari 2 skor yang sama, maka diurutkan berdasarkan waktu secara ascending.
- d. View All Highscore : untuk melihat 10 skor tertinggi pada board yang sedang dipilih yang didapatkan dari semua user yang terdaftar. Tampilan high score akan menampilkan skor secara terurut mengecil, nama user yang mendapatkan skor tersebut, dan waktu(tanggal+jam) pencapaian skor tersebut. Jika ada lebih dari 2 skor yang sama, maka diurutkan berdasarkan waktu secara ascending.
- e. Logout : untuk kembali ke Main Menu.

Kamus kata dan board yang tersedia dapat didapatkan di <http://milestone.if.itb.ac.id/kuliah/if2110/> yang dapat diakses dengan username dan password sebagai berikut :

Username : if2110

Password : typezero

**Bonus :**

1. Kamus kata disimpan dalam struktur data pohon n-er atau hash table untuk mempercepat pencarian kata dalam kamus kata.
2. Pada akhir permainan, ditampilkan seluruh kata dalam kamus kata yang dapat dibentuk dari board.
3. Menambahkan menu *View Board Statistic* pada *Preparation Menu*: untuk melihat statistik board yang sedang dipilih. Statistik board berisi informasi jumlah user berbeda yang pernah memainkan board tersebut, rata-rata skor dari seluruh user yang pernah memainkan board tersebut, dan ranking kesulitan board berdasarkan rata-rata skor user.
4. Duel Mode: dua user berbeda memainkan permainan bersamaan.  
Contoh: Board player 1 ditampilkan di sebelah kiri dan board player 2 di sebelah kanan. Input yang digunakan player 1 adalah qwerasdzc seperti yang terdefinisi dalam soal, sedangkan input player 2 adalah input player 1 yang digeser 4 karakter keyboard ke kanan, yaitu tyuighjbnm.  
Buatlah spesifikasi yang jelas terkait mode ini.
5. Tampilan program yang menarik.

**ADT yang digunakan (dapat pula menggunakan ADT lain, cantumkan analisis alasan penggunaan ADT tersebut dalam laporan):**

1. ADT Point

Digunakan untuk menyimpan posisi kursor saat permainan yang berisi posisi index Matriks yang ditunjuk.

2. ADT Tanggal + ADT Jam

Digunakan untuk menyimpan waktu pencapaian high score.

3. ADT Array Terurut Mengecil

Digunakan untuk menampilkan all high score.

4. ADT Matriks

Digunakan untuk merepresentasikan board permainan.

5. ADT Mesin Kata

Digunakan untuk membaca kamus data dan file eksternal.

6. ADT Stack atau ADT List Double Pointer

Digunakan untuk menyimpan huruf-huruf yang sedang diproses selama permainan.

7. ADT Queue

Digunakan untuk menampilkan suggestion kata dalam permainan.

8. ADT Priority Queue

Digunakan untuk menampilkan semua kata yang dibentuk selama permainan saat akhir permainan yang ditampilkan terurut mengecil mulai dari kata yang memiliki skor terbesar.

9. ADT Multi List

Digunakan untuk menyimpan List User dengan elemennya mengandung List Highscore user tersebut.

10. ADT Set

Digunakan untuk menyimpan kata-kata yang sudah pernah dibentuk sebelumnya.

11. ADT Map

Digunakan untuk menyimpan poin pada huruf dengan key = huruf, value = poin huruf tersebut.