TUGAS PRAKTIKUM
MODUL XI
PENGGUNAAN DAN APLIKASI STACK

Versi A.1

Prepared by: Gun n Rikses

Notasi Infix dan Postfix pada Operasi Aritmatika

Penjelasan

Dalam kegiatan sehari-hari, tentunya kita sudah sering melakukan operasi aritmatika. Biasanya, operasi aritmatika dilakukan dengan memberikan prioritas berupa tanda kurung pada operasi yang lebih diprioritaskan. Kemudian, setelah operasi dalam kurung selesai dilakukan, operasi berjalan dengan urutan prioritas perkalian/pembagian dan penjumlahan/pengurangan.

Operasi yang disebutkan di atas disebut dengan operasi infix (sisipan). Operasi di atas disebut operasi infix karena operator dari operasi tersebut disisipkan di antara dua buah operand. Karena operasi ini memiliki banyak prioritas yang harus ditangani sebelum melakukan proses operasi, terdapat banyak waktu yang hilang apabila operasi ini diaplikasikan dalam pemrograman.

Akan tetapi, operasi aritmatika tidak terbatas hanya berupa operasi infix saja. Terdapat satu operasi lain yang dapat digunaan dalam operasi aritmatika ini. Operasi ini disebut operasi postfix (akhiran). Operasi ini menempatkan operator di akhir input pengguna sehingga operasi aritmatika dapat dilakukan secara berurutan. Hal ini dapat menghemat waktu operasi karena program tidak perlu mengurutkan operasi berdasarkan prioritas terlebih dahulu.

Contoh operasi postfix adalah seperti berikut :

- 1 3 + 4 * 2 /
- 4 2 * 3 – 1 +

Enkripsi

Penjelasan

Teknologi informasi merupakan suatu hal yang tidak dapat dipisahkan lagi dari kehidupan sehari-hari. Hampir semua kegiatan sehari-hari menggunakan teknologi informasi untuk pelaksanaannya. Akan tetapi, dibalik kemudahan dari penggunaan teknologi informasi ini, terdapat bahaya yang mengintai karena seluruh data pribadi kita disebar luas ke dunia maya.

Oleh karena itu, dibutuhkan suatu sistem yang dapat mengamankan data pribadi yang kita sebar ke dunia maya agar tidak dapat diketahui orang yang tidak bertanggung jawab. Cara yang digunakan adalah melalui enkripsi. Enkripsi bertujuan untuk mengubah suatu data menjadi suatu bentuk data lain yang hanya dapat diakses oleh orang-orang yang berwenang. Terdapat banyak cara untuk mengenkripsi data, salah satunya adalah dengan memanfaatkan stack.

Problem 1 : Membuat kalkulator dengan notasi postfix

Definisi Masalah

Buatlah sebuah program yang menghitung operasi aritmatika yang diberikan oleh user dengan format masukan berupa notasi postfix. Gunakan stack untuk menyimpan masukan berupa operand. Gunakan fungsi push() dan pop() untuk melakukan operasi pada stack. Input dibatasi hanya satu digit (0-9). Tampilkan hasil dalam 2 angka di belakang koma. Untuk memudahkan, penjelasan notasi postfix dapat dilihat pada ilustrasi berikut :

Misalkan kita mempunyai operasi "4 2 * 3 – 1 +", maka hal yang dilakukan adalah seperti berikut:

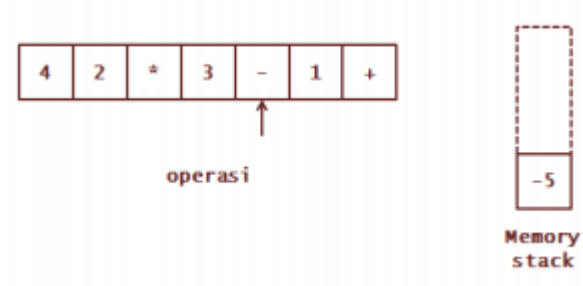
1. Bila bertemu dengan operand, masukkan operand ke dalam memory stack.

Gambar 1 Bila bertemu operand, masukkan operand ke dalam stack

2. Bila bertemu dengan operator, lakukan operasi dan hasilnya disimpan ke dalam memory stack.

Gambar 2 Bila bertemu operator, lakukan operasi dan simpan hasilnya di stack

3. Urutan operasi adalah dari angka kedua ke angka pertama.



Gambar 3 Operasi yang dilakukan adalah "3-8", bukan "8-3"

Contoh Input dan Output

Input ke STDIN

```
34*2-9+
```

Output ke STDOUT

```
INPUT :
-1.00
```

Deliverable

Simpan program utama problem 1 dengan nama `problem1.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini.

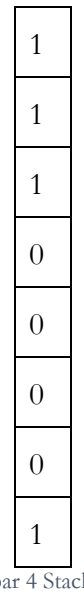
Problem 2 : Melakukan enkripsi dengan menggunakan stack

Definisi Masalah

Buatlah sebuah program menerima 8 buah input berupa bilangan biner. Program kemudian mengenkripsi input tersebut dan menampilkan hasil enkripsi berupa bilangan heksadesimal. Gunakan fungsi push() dan pop() untuk melakukan operasi pada stack.

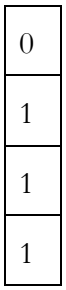
Berikut adalah ilustrasi proses enkripsi :

1. Program menerima 8 buah bilangan biner secara berurutan tidak secara langsung. Misalkan inputnya adalah "1 0 0 0 0 1 1 1".
2. Program kemudian memasukkan setiap input tersebut ke dalam suatu stack dengan menggunakan fungsi push().



Gambar 4 Stack bilangan biner

3. Program mengambil 4 data teratas stack dan mengubahnya menjadi heksadesimal dengan urutan terbalik.



Gambar 5 Urutan stack hasil enkripsi

4. Program mengambil 4 data berikutnya dari stack dan mengubahnya menjadi heksadesimal.
5. Program menampilkan 2 karakter heksadesimal hasilnya pada console.

Contoh Input dan Output

Input ke STDIN

```
1
0
0
0
0
1
1
1
```

Output ke STDOUT

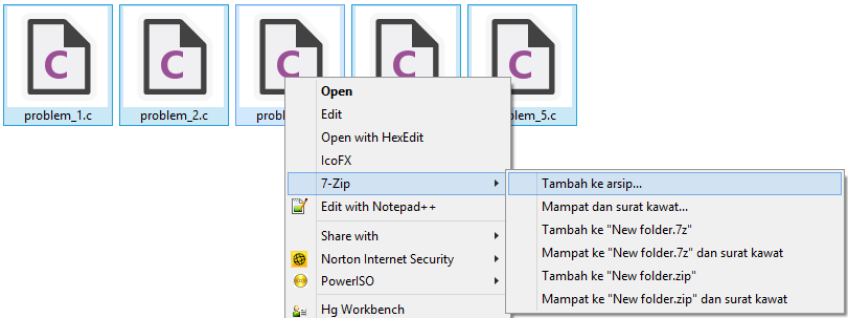
```
Input biner ke-0 :
Input biner ke-1 :
Input biner ke-2 :
Input biner ke-3 :
Input biner ke-4 :
Input biner ke-5 :
Input biner ke-6 :
Input biner ke-7 :
Hasil : e1
```

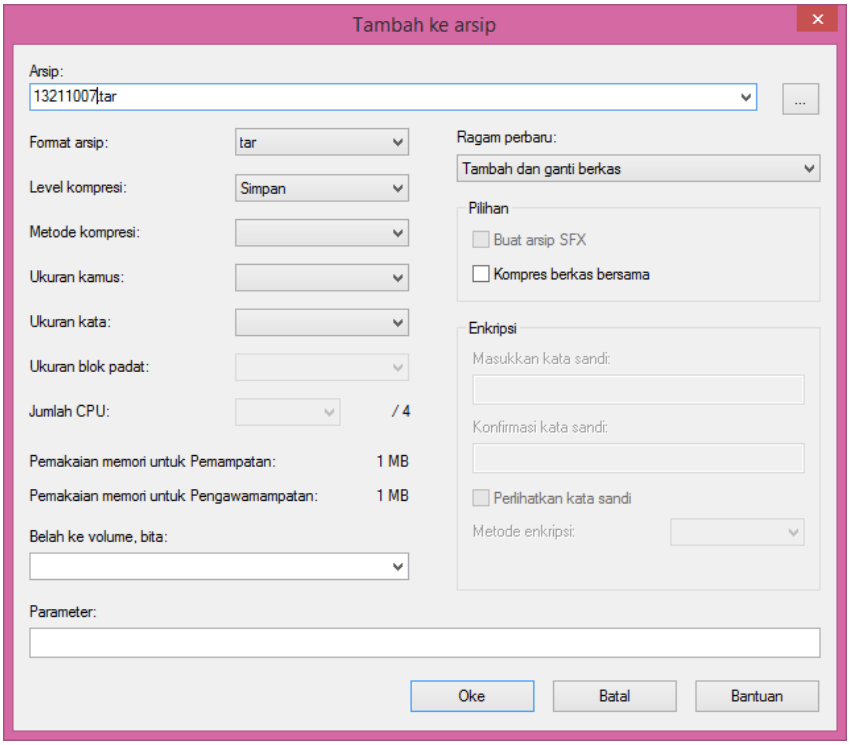
Deliverable

Simpan program utama problem 2 dengan nama `problem2.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini.

Petunjuk Penyerahan Tugas Praktikum Modul XI

Simpan file `problem1.c` dan `problem2.c`. Gunakan program 7-zip untuk mengkompresi menjadi arsip TAR (.tar). Penamaan file TAR bebas (disarankan menggunakan NIM). File TAR ini yang akan di-submit ke server MIKU. Hanya file kode saja yang dimasukkan ke dalam arsip TAR. File *executable* tidak perlu dimasukkan.





Selesai