|  |
| --- |
| **IK141 Struktur Data**  **Struktur Data**  **STACK (TUMPUKAN)** |



**Di Susun Oleh :**

**Wendi Kardian, 2100016**

**PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER**

**FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PENDIDIKAN INDONESIA**

**26 Maret 2022**

1. **Implementasi dan Hasil**

|  |
| --- |
| **Implementasi dan Hasil** |
| 1. **Studi Kasus**    * + 1. Buatlah sebuah program untuk memeriksa apakah sebuah string adalah polindrom atau bukan. Diberikan sebuah string n dengan panjang string k dengan 0<k<1000. Contoh input:   katak  output :  polindrom  input :  jumat  output :  bukan polindrom  **Source Code :**   * + - * 1. Proses inisialisasi awal untuk melakukan import file header serta membuat structure yang nantinya akan di gunakan untuk membuat stack untuk mencek apakah polindrom atau bukan.      * 1. Prosedur untuk menambahkan data di akhir berupa char yang nantinya akan berupa tumpukan char yang nanti kelak akan di reverse dan di cek apakah kata tersebut polindrom atau tidak      * 1. Fungsi untuk melakukan proses pop atau menghapus data dari stack yang berada diujung stack tersebut, dan setelah di ambil datanya akan direturn char tersebut yang nantinya akan disimpen dalam sebuah char untuk di check apakah polindrom atau bukan.      * 1. Fungsi utama dimana terdapat beberapa proses deklarasi variabel, serta terdapat proses input data string yang nantinya akan di cek apakah polindrom atau tidak. Setelah itu, terdapat juga proses iterasi untuk melakukan push kedalam stack. Setelah data di push, langkah selanjutnya data di pop yang di simpan kedalam sebuah array of string. Untuk mengecek apakah kata tersebut string itu polindrom atau tidak menggunakan strcmp untuk melakukan pengecekan apakah array of string tersebut sama atau tidak.     **Test Case** |
| 1. Diberikan sebuah bilngan biner a dengan n digit, 0<n<1000. Lakukan konversi bilangan biner tersebut menjadi bilangan desimal dengan memanfaatkan penggunaan penyimpanan data stack dengan ukuran n.    1. Import library yang diperlukan, selanjutnya buat structure untuk stack yang akan menampung tumpukan string berupa bilangan biner dan top untuk menentukan index teratas dari stack tersebut. Kemudian, deklarasi sebuah stack dengan tipe data struct stack yang diberi nama biner.      * 1. Prosedur untuk menambahkan data di akhir stack tersebut yang di isi dengan bilangan biner o dan 1.      * 1. Fungsi pop untuk menghapus data yang berada paling akhir atau top. Dalam fungsi ini nanti akan dilakukan proses pengecekan apakah char tersebut 1 atau 0. Apabila 1 berarti basis tersebut akan mengembalikan sebuah integer yang bernilai 2 pangkat index dari basis tersebut. Apabila 0 berarti akan mengembalikan 0 juga (bilangan apapun yang dikalikan dengan 0 akan menghasilkan bilangan 0 juga)      * 1. Fungsi utama yang berisikan inisialisasi variable yang akan diperlukan, kemudian terdapat proses untuk memasukan data biner tersebut kedalam string yang nantinya tiap string tersebut akan dimasukan kedalam stack secara decrement. Setelah data dimasukan (di push) maka selanjutnya program akan melakukan proses pop (menghapus data) serta nilai returnnya akan disimpan dan terus ditambahkan kedalam variable yang diberi nama desimal, dimana variable tersebut memiliki fungsi untuk menampung nilai dari hasil convert biner ke desimal tersebut.     **TEST CASE**     1. Silahkan cari lagi dari berbagai referensi, minimal 2 operasi yang mungkin dapat diselesaikan dengan lebih mudah menggunakan penyimpanan data STACK. 2. **Fibonnaci Sequence**   Fibonnaci sequence adalah serangkaian deret angka sederhana yang susunan angkanya merupakan penjumlahan dari dua angka sebelumnya (0,1,1,2,3,5,8,13,21,...dst) rumus deret Fibonacci bisa ditulis sebagai berikut Un = Un-2 + Un-1. Adapun source code untuk membuat program fibonnaci menggunakan stack adalah sebagai berikut.   * 1. Import library yang diperlukan untuk program tersebut. Kemudian, terdapat pembuatan structure yang nantinya akan menampung stack untuk menghitung deret fibonnaci. Setelah itu, terdapat 2 variabel yang bertipe structure stack tersebut untuk menampung deret fibonnaci.      * 1. Prosedur untuk menambahkan data kedalam deret fibonnaci tersebut, ada beberapa aturan diantaranya (1) untuk baris pertama dan kedua selalu bernilai 1. (2) untuk baris selanjutnya merupakan hasil penjumlahan dari 2 baris sebelumnya. Angka tersebut akan dimasukan kedalam stack tersebut.      * 1. Fungsi untuk menghapus data (pop) serta akan mengembalikan nilai baris fibonnaci paling ujung yang nantinya akan disimpan kedalam variable reverse fibonanci.      * 1. Fungsi utama program tersebut yang dimana didalamnya terdapat deklarasi variable untuk fibonnaci (normal fibonnaci) dan fibonnaci2 (reverse fibonnaci), dimana data akan di push berdasarkan jumlah angka yang di input oleh user (apabila 9 maka program akan membuatkan deret fibonnaci sepanjang 9 baris). Setelah data ditambahkan, program juga akan melakukan proses pop yang hasil return value nya akan ditampung kedalam variable fibonnaci2 (reverse fibonnaci). Lalu dari fibonnaci 2 akan di push dan di pop Kembali untuk mendapatkan nilai dari deret fibonnaci tersebut.s     **TEST CASE**     1. **Pharanthesis Checker**   Pharanthesis checker merupakan sebuah program untuk melakukan proses pengecekan terhadap operasi aritmatika apakah penggunaan ( // { // [ sudah benar atau tidak. Ada beberapa rules yang diterapkan dalam program ini, (1) apabila pharanthesis sudah dibuka maka harus ditutup Kembali contoh, (A+(A\*B) -> salah (2) apabila pharanthesis diawali oleh ‘(‘ maka harus di akhiri oleh ‘)’, tidak boleh ‘}’ maupun ‘]’, contoh (A\*B+C] -> salah. Adapun source code untuk program Pharanthesis Checker adalah sebagai berikut.   * + - * 1. Import library, kemudian pembuatan structure untuk stack yang akan menampung expression yang akan di check pharenthesisnya. Kemudian, deklarasi sebuah variable berdasarkan structure yang sudah dibuat.      * + - * 1. Prosedur untuk menambahkan char expression kedalam stack.      * + - * 1. Fungsi untuk melakukan proses pop (menghapus data) dari stack yang nantinya akan mengembalikan sebuah char dari expression yang berada paling atas array of string tersebut.      * + - * 1. Fungsi utama yang di dalamnya terdapat proses deklarasi variable yang diperlukan oleh program. Kemudian, terjadi proses input expression yang dilakukan oleh user, selanjutnya berdasarkan input yang user berikan akan dilakukan proses iterasi sebanyak panjangnya string expression tersebut dan akan di cek apakah expression tersebut valid atau tidak berdasarkan rules proses pembuatan expression.      * + - * 1. Pada tahap sebelumnya, diinisialisasikan variable yang Bernama flag untuk menyimpan status benar atau tidak nya sebuah expression. Apabila flag bernilai 1 maka akan dicetakan valid expression, selain itu akan dicetakan invalid expression.     **TEST CASE** |

1. **Kesimpulan**

|  |
| --- |
| **Kesimpulan** |
| **STACK atau tumpukan** adalah sebuah bentuk penyimpanan data dengan cara ditumpuk, data disimpan diatas data lainnya. Struktur ini memungkinkan operasi akses data hanya dapat dilakukan pada data yang berada pada posisi puncak tumpukan saja yaitu data yang terakhir disimpan sehingga pada struktur data ini dikenal dengan istilah Last In First Out (LIFO). Proses dalam stack diantaranya adalah Push (untuk menambahkan data), Pop (untuk menghapus sebuah data), isEmpty (untuk mengecek apakah sebuah stack tersebut kosong atau tidak), dan isFull (untuk mengecek apakah sebuah stack tersebut sudah mencapai kapasitas maksimalnya atau belum).  Stack Data Structure In C++ With Illustration  Dalam pengimplementasiannya, stack dapat digunakan untuk berbagai jenis hal untuk melakukan berbagai jenis kalkulasi, seperti mengecek kata polindrom atau bukan, melakukan conversi dari biner ke desimal, membuat deret fibonnaci, serta mengecek apakah sebuah expression sudah benar atau tidak. |