**LAPORAN PRAKTIKUM**

**STRUKTUR DATA DAN ALGORITMA**

**MODUL 6**

**STACK**

Sebuah gambar berisi simbol, logo, Grafis, deasin

Deskripsi dibuat secara otomatis

Disusun oleh :

Faqih Abdullah

2311102048

Dosen Pengampu

Wahyu Andri Saputra S.Pd., m.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**ISNTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

1. **DASAR TEORI**

Stack adalah struktur data linear yang bekerja berdasarkan prinsip LIFO (Last In First Out) atau yang artinya data yang terakhit dimasukan ke dalam stack akan menjadi data yang pertama kali diambil atau dikeluarkan dari stack. Stack biasanya digunakan dalam implementasi algoritma dan fungsi-fungsi matematis seperti konversi infix ke postfix, evaluasi ekspresi postfix, dan validasi tanda kurung pada ekspresi matematis. Stack juga sering digunakan dalam aplikasi yang membutuhkan manajemen tumpukan data seperti pengembangan sistem operasi, kompilator, dan perangkat lunak grafis.

Stack sering digunakan dalam implementasi algoritma dan fungsi matematis seperti evaluasi ekspresi matematis dan konversi infix ke postfix. Selain itu stack juga digunakan dalam aplikasi yang membutuhkan manajemen tumpukan daa seperti pengembangan sistem operasi, kompilator, dan perangkat lunak grafis.

Pemahaman tetang struktur data stack sangat penting bagi para programmer untuk memecahkan masalah dan merancang algoritma yang efisien.

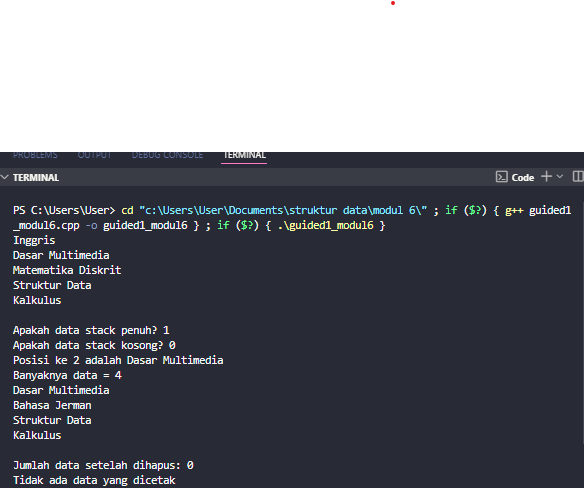
1. GUIDED

Guided 1

Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  string arrayBuku[5];  int maksimal = 5, top = 0;  bool isFull()  {      return (top == maksimal);  }  bool isEmpty()  {      return (top == 0);  }  void pushArrayBuku(*string* *data*)  {      if (isFull())      {          cout << "Data telah penuh" << endl;      }      else      {          arrayBuku[top] = data;          top++;      }  }  void popArrayBuku()  {      if (isEmpty())      {          cout << "Tidak ada data yang dihapus" << endl;      }      else      {          arrayBuku[top - 1] = "";          top--;      }  }  void peekArrayBuku(int *posisi*)  {      if (isEmpty())      {          cout << "Tidak ada data yang bisa dilihat" << endl;      }      else      {          int index = top;          for (int i = 1; i <= posisi; i++)          {              index--;          }          cout << "Posisi ke " << posisi << " adalah " << arrayBuku[index] << endl;      }  }  int countStack()  {      return top;  }  void changeArrayBuku(int *posisi*, *string* *data*)  {      if (posisi > top)      {          cout << "Posisi melebihi data yang ada" << endl;      }      else      {          int index = top;          for (int i = 1; i <= posisi; i++)          {              index--;          }          arrayBuku[index] = data;      }  }  void destroyArraybuku()  {      for (int i = top; i >= 0; i--)      {          arrayBuku[i] = "";      }      top = 0;  }  void cetakArrayBuku()  {      if (isEmpty())      {          cout << "Tidak ada data yang dicetak" << endl;      }      else      {          for (int i = top - 1; i >= 0; i--)          {              cout << arrayBuku[i] << endl;          }      }  }  int main()  {      pushArrayBuku("Kalkulus");      pushArrayBuku("Struktur Data");      pushArrayBuku("Matematika Diskrit");      pushArrayBuku("Dasar Multimedia");      pushArrayBuku("Inggris");      cetakArrayBuku();      cout << "\n";      cout << "Apakah data stack penuh? " << isFull() << endl;      cout << "Apakah data stack kosong? " << isEmpty() << endl;      peekArrayBuku(2);      popArrayBuku();      cout << "Banyaknya data = " << countStack() << endl;      changeArrayBuku(2, "Bahasa Jerman");      cetakArrayBuku();      cout << "\n";      destroyArraybuku();      cout << "Jumlah data setelah dihapus: " << top << endl;      cetakArrayBuku();      return 0;  } |

SCREENSHOT OUTPUT



DESKRIPSI PROGRAM

*Program diatas adalah penggunaan sederhana dari stack menggunakan array.*

***Deklarasi dan Inisialisasi variable***

* *String arrayBuku[5]; : mendeklarasikan array dengan ukuran 5 untuk menyimpan string.*
* *Int maskimal = 5, top = 0; : mendeklarasikan ‘maksimal’ sebagai kapasitas maksimum stack dan ‘top’ sebagai penunjuk posisi elemen teratas di stack, diinisialisasikan dengan nilai 0.*

***Fungsi-fungsi stack***

*1. isFull()*

*Mengembalikan ‘true’ jika stack penuh, yaitu jika’top’ sama dengan ‘maksimal’.*

*2. isEmpty()*

*Mengembalikan ‘true’ jika stack kosong, yaitu jika ‘top’ bernilai 0.*

*3. pushArrayBuku(string data)*

*Menambahkan elemen baru ke stack. Jika stack penuh, menampilkan pesan “Data telah Penuh”. Jika tidak penuh, menambahkan data ke posisi ‘top’ dan meningkatkan ‘top’ sebesar 1.*

*4. popArrayBuku()*

*Menghapus elemen teratas dari stack. Jika stack kosong, menampilkan pesan “Tidak ada data yang dihapus”. Jika kosong, mengosongkan elemen teratas dan mengurangi ‘top’ sebesar 1.*

*5. peekArrayBuku (int posisi)*

*Melihat elemen pada posisi tertentu dari atas stack. Jika stack kosong, menampilkan pesan “Tidak ada data yang bisa dilihat”. Jika tidak kosong, menghitung indeks posisi yang diminta dan menampilkan elemen pada posisi tersebut.*

*6. counStack()*

*Mengembalikan jumlah elemen dalam stack.*

*7 changeArrayBuku(int posisi, string data)*

*Mengubah nilai elemen pada posisi tertentu dalam stack, jika posisi melebihi jumlah elemen dalam stack, menampilkan pesan “Posisi melebihi data yang ada”. Jika tidak, menghitung indeks dan mengubah nilai elemen pada posisi tersebut.*

*8. destroyArrayBuku()*

*Menghapus semua elemen dalam stack dan mengatur ‘top’ Kembali ke 0.*

*9. cetakArrayBuku()*

*Mencetak semua elemen dalam stack dari atas ke bawah. Jika stack kosong, menampilkan pesan “Tidak ada data yang dicetak”.*

***Fungsi ‘main()’***

*Fungsi ‘main ()’ menjalankan beberapa operasi pada stack:*

* *Menambahkan beberapa buku ke stack menggunakan ‘ pushArrayBuku’.*
* *Mencetak isi stack dengan ‘cetakArrayBuku’.*
* *Mengecek apakah stack penuh atau kosong dengan ‘isFull’ dan ‘isEmpty’.*
* *Melihat elemen pada posisi tertentu dengan ‘peekArrayBuku’.*
* *Menghapus elemen teratas dengan ‘popArrayBuku’.*
* *Menghitung jumlah elemen dalam stack dengan ‘countStack’.*
* *Mengubah elemen pada posisi tertentu dengan ‘chabgeArrayBuku’.*
* *Menghapus semua elemen dalam stack dengan ‘destroyArrayBuku’.*
* *Mencetak ulang isi stack setelah dihapus.*

1. **UNGUIDED**

Unguided 1

1. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

Source code

|  |
| --- |
| #include <iostream>  #include <stack>  #include <string>  using namespace std;  bool isPalindrome(const string &word) {  stack<char> charStack;    for (char ch : word) {  charStack.push(ch);  }    string reversedWord;  while (!charStack.empty()) {  reversedWord += charStack.top();  charStack.pop();  }    return word == reversedWord;  }  int main() {  string word;  cout << "Masukkan kata: ";  getline(cin,word);    if (isPalindrome(word)) {  cout << word << " adalah palindrom." << endl;  } else {  cout << word << " bukan palindrom." << endl;  }    return 0;  } |

Screenshot output

Sebuah gambar berisi teks, cuplikan layar, software, Sistem operasi

Deskripsi dibuat secara otomatis

Deskripsi Program

*Program ini berfungsi untuk memeriksa apakah suatu kata yang dimasukan oleh pengguna adalah polindrom atau tidak. Proses pemeriksaan dilakukan dengan menggunakan stack untuk membalikan urutan karakter dalam kata, kemudian membandingkan kata asli dengan kata yang dibalik. Jika kedua kata tersebut sama, maka kata tersebut adalah polindrom.*

***Fungsi ‘isPolindrome’***

*1. parameter : fungsi ini menerima satu parameter bertipe ‘const string &’ yang Bernama ‘word’.*

*2. stack : mendeklarasikan stack ‘charStack’ untuk menyimpan karakter-karakter dari kata.*

*3. pengisian stack : Loop ‘for’ digunakan untuk mendorong (push) setiap karakter dari ‘word’ kedalam ‘charStack’.*

*4. membalikkan kata : mendeklarasikan string ‘reverseWord’ untuk menyimpan kata yang dibalik. Loop ‘while’ digunakan untuk mengeluarkan (pop) setiap karakter dari “charStack’ dan menambahkannya ke ‘reverseWord’ hingga stack kosong.*

*5. mengembalikan hasil perbandingan antara ‘word dan ‘reverseWord’. Jika kedua sama, maka kata tersebut adalah polindrom.*

***Fungsi ‘main’***

*1. mendeklarasikan variable ‘word’ unuk menyimpan input dari pengguna.*

*2. menampilkan pesan “Masukkan kata: “ dan membaca seluruh baris input menggunakan ‘getline(cin, word)’ sehingga pengguna dapat memasukkan kata yang ingin dipersiksa.*

*3. memanggil fugnsi ‘isPolindrome’ dengan parameter ‘word’. Jika ‘true’ , menampilkan pesan bahwa kata tersebut adalah palindrom. Jika ‘false’ , menampilkan pesan bahwa kata tersebut bukan palindrom.*

Unguided 2

2. buatlah program untuk melakukan pembalikkan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya besrta operasi/fungsi yang dibuat?

Source Code

|  |
| --- |
| #include <iostream>  #include <stack>  #include <string>  #include <sstream>  #include <vector>  using namespace std;  // Fungsi untuk membalikkan kata menggunakan stack  *string* reverseWord(const *string* &*word*) {      stack<char> charStack;        // Push setiap karakter kata ke stack      for (char ch : *word*) {          charStack.push(ch);      }        // Pop setiap karakter dari stack untuk membentuk kata yang dibalik  *string* reversedWord;      while (!charStack.empty()) {          reversedWord += charStack.top();          charStack.pop();      }        return reversedWord;  }  int main() {  *string* input;      cout << "Masukkan kalimat dengan minimal 3 kata: ";      getline(cin, input);      // Split input string menjadi kata-kata  *stringstream* ss(input);  *string* word;      vector<*string*> words;      while (ss >> word) {          words.push\_back(word);      }        // Periksa jika jumlah kata minimal 3      if (words.size() < 3) {          cout << "Input harus mengandung minimal 3 kata." << endl;          return 1;      }        // Balikkan setiap kata dan simpan dalam vector baru      vector<*string*> reversedWords;      for (const *string* &word : words) {          reversedWords.push\_back(reverseWord(word));      }        // Cetak setiap kata yang sudah dibalik      cout << "Hasil :" << endl;      for (const *string* &reversedWord : reversedWords) {          cout << reversedWord << " ";      }      cout << endl;        return 0;  } |

Screenshot Output

Sebuah gambar berisi teks, elektronik, cuplikan layar, software

Deskripsi dibuat secara otomatis

Deskripsi Program

*Terdapat 3 ouputan program yang dibuat oleh user yang pertama kalimat “sayang ibu” Dimana program akan mengeluarkan pesan “input harus mengandung minimal 3 kata” karena inputan user kurang dari 3 kata. Outputan yang kedua kalimat “ aku sayang ibu” program membalikkan kalimat menjadi “uka gnayas ubi” program dapat berjalan karena inputan user memenuhi syarat program berjalan yaitu inputan harus 3 kata atau lebih. Output yang ketiga terdapat 4 kata “uka gnayas ubi llop” program membalikkan inputan user yang aslinya adalah “aku sayang ibu poll”. Jadi kesimpulan dari output diatas adalah dapat berjalan jika inputan user 3 kata atau lebih dan akan menampilkan pesan “input harus mengandung minimal 3 kata” jika inputan user kurang dari 3.*

***Fungsi ‘reverseWord’***

* *Menerima sebuah kata (string) dan mengembalikan kata tersebut dalam urutan terbalik.*
* *Menggunakan stack untuk membalikkan urutan karakter dalam kata.*

***Fungsi ‘main’***

* *Menerima input dari pengguna berupa datu baris kalimat.*
* *Menggunakan ‘stringsteam’ untuk memisahkan input menjadi kata-kata*
* *Memeriksa apakah jumlah kata minimal tiga. Jika tidak, program akan menampilkan pesan kesalahan dan keluar.*
* *Membalikkan setiap kata menggunakan dungsi ‘reverseWord’ dan menyimpan hasilnya dalam vector ‘reverseWord’.*
* *Mencetak setiap kata yang sudah dibalik.*

**DAFTAR PUSTAKA**

Praktikum, A. (2024). Stack. *Modul 6 Stack*, 1-6.

Sutiono S.Kom., M. M. (2023, maret 9). *Struktur Data Stack: Pengertian, Jenis dan Contoh: DosenIT.com*. Diambil kembali dari DosenIT.com: https://dosenit.com/ilmu-komputer/struktur-data-stack