

LAPORAN PRAKTIKUM STRUKTUR DATA

Jobsheet 7 Queue



Disusun Oleh:

Fitri Walidi: 22343021

Dosen Pengampu:

Randi Proska Sandra S.Pd., M.Sc.

PROGRAM STUDI S1 INFORMATIKA (NK)

JURUSAN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2023

Tugas

Buatlah program tentang algoritma Algoritma Breadth First Search (BFS) dalam bahasa C. Lalu, jelaskan terkait algoritma tersebut dan bagaimana prinsip queue digunakan dalam algoritma tersebut!

Kode:

```
//Dibuat_oleh_Fitri-Waldi_NIM-22343021

#include <stdio.h>
#include <stdlib.h>

// Struktur simpul (node)
struct Node {
    int data;
    struct Node* left;
    struct Node* right;
};

// Membuat simpul baru
struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct
Node));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}

// Menampilkan simpul dalam urutan Breadth First Search (BFS)
void breadthFirstSearch(struct Node* root) {
    if (root == NULL)
        return;

    // Inisialisasi queue untuk menyimpan simpul yang akan diproses
    struct Node** queue = (struct Node**)malloc(sizeof(struct Node*)
* 100);
    int front = 0;
    int rear = 0;

    // Masukkan root ke dalam queue
    queue[rear++] = root;
```

```

while (front < rear) {
    // Keluarkan simpul dari queue dan tampilkan datanya
    struct Node* currentNode = queue[front++];
    printf("%d ", currentNode->data);

    // Masukkan anak-anak simpul ke dalam queue jika ada
    if (currentNode->left != NULL)
        queue[rear++] = currentNode->left;
    if (currentNode->right != NULL)
        queue[rear++] = currentNode->right;
}

// Bebaskan memori yang dialokasikan untuk queue
free(queue);
}

int main() {
    // Membuat pohon biner
    struct Node* root = createNode(1);
    root->left = createNode(2);
    root->right = createNode(3);
    root->left->left = createNode(4);
    root->left->right = createNode(5);
    root->right->left = createNode(6);
    root->right->right = createNode(7);

    printf("Breadth First Search (BFS) traversal: ");
    breadthFirstSearch(root);

    return 0;
}

```

Output
Clear

```

/tmp/C5cZnmfGQb.o
Breadth First Search (BFS) traversal: 1 2 3 4 5 6 7

```

Penjelasan:

Program di atas adalah implementasi Algoritma Breadth First Search (BFS) dalam bahasa C. Berikut ini penjelasan singkat mengenai program tersebut untuk laporan:

1. Pada awal program, terdapat inklusi pustaka header **stdio.h** dan **stdlib.h** yang diperlukan untuk fungsi input/output dan alokasi memori dinamis.
2. Selanjutnya, didefinisikan struktur **Node** yang merepresentasikan simpul dalam pohon. Struktur ini memiliki tiga anggota yaitu **data** untuk menyimpan nilai data, **left** untuk menyimpan alamat simpul anak kiri, dan **right** untuk menyimpan alamat simpul anak kanan.
3. Di dalam fungsi **createNode**, simpul baru dibuat dengan menggunakan alokasi memori dinamis. Data dimasukkan ke dalam simpul, dan pointer anak kiri dan anak kanan diatur menjadi NULL. Fungsi ini mengembalikan alamat simpul yang baru dibuat.
4. Fungsi **breadthFirstSearch** digunakan untuk menampilkan data dari simpul-simpul dalam urutan Breadth First Search (BFS). Fungsi ini menerima parameter simpul **root** yang merupakan akar pohon. Pada awalnya, fungsi ini melakukan inisialisasi queue untuk menyimpan simpul yang akan diproses. Kemudian, simpul akar dimasukkan ke dalam queue.
5. Selama queue tidak kosong, loop while akan berjalan. Pada setiap iterasi, simpul yang sedang diproses diambil dari depan queue. Data dari simpul tersebut ditampilkan menggunakan fungsi **printf**. Anak-anak simpul (jika ada) dimasukkan ke dalam queue untuk diproses selanjutnya.
6. Setelah semua simpul telah diproses, memori yang dialokasikan untuk queue dibebaskan menggunakan fungsi **free**.
7. Di dalam fungsi **main**, sebuah pohon biner dibangun menggunakan fungsi **createNode**. Simpul-simpul dan hubungan antar simpul ditetapkan. Pada contoh ini, pohon biner memiliki kedalaman maksimum 2.
8. Terakhir, hasil BFS traversal dicetak menggunakan fungsi **printf** dengan format "Breadth First Search (BFS) traversal: ".

Program ini mengimplementasikan algoritma BFS untuk melintasi dan menampilkan simpul-simpul dalam pohon secara melintang (horizontal) dari simpul awal ke simpul-simpul tetangga sebelum melanjutkan ke simpul-simpul selanjutnya.