

LAPORAN PRAKTIKUM UJIAN TENGAH SEMESTER STRUKTUR DATA DAN ALGORITMA

Kelas B



Disusun Oleh:

Fadhlurrahman Alaudin (2408107010053)

M. Ilham (2408107010043)

Asisten Lab Pengampu:

Athar Rayyan Muhammad

Rafli Afriza Nugraha

Muhammad Syahidal Akbar Zas

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SYIAH KUALA

2025

A. Pendahuluan

Dalam pemrograman dan ilmu komputer, ekspresi matematika dapat direpresentasikan dalam berbagai bentuk notasi, yaitu infix, prefix, dan postfix. Notasi ini digunakan untuk menuliskan operasi matematika dengan cara yang mempermudah evaluasi ekspresi, terutama dalam komputasi yang melibatkan algoritma dan struktur data seperti *stack*.

- Notasi infix adalah bentuk penulisan ekspresi yang paling umum digunakan dalam matematika sehari-hari, di mana operator ditempatkan di antara dua operand, seperti $A + B$.
- Notasi prefix (Polish Notation) menempatkan operator sebelum operand, misalnya $+ A B$. Notasi ini sering digunakan dalam pemrosesan bahasa pemrograman karena tidak memerlukan tanda kurung untuk menunjukkan urutan operasi.
- Notasi postfix (Reverse Polish Notation) menempatkan operator setelah operand, misalnya $A B +$. Notasi ini sangat efisien dalam evaluasi menggunakan *stack*, karena tidak memerlukan aturan prioritas operator atau tanda kurung.

Pada praktikum ini, dilakukan implementasi berbagai algoritma untuk mengonversi ekspresi dari satu bentuk ke bentuk lainnya, seperti konversi dari infix ke postfix, infix ke prefix, serta evaluasi ekspresi postfix dan prefix menggunakan struktur *stack*. Implementasi ini bertujuan untuk memahami bagaimana komputer memproses ekspresi aritmatika serta bagaimana struktur data *stack* digunakan dalam manipulasi ekspresi matematika.

B. Nama Fungsi yang Digunakan

- `charTanpaBuffer()`: fungsi untuk menginput char tanpa harus menekan enter.
- `clearTerminal()`: fungsi untuk membersihkan terminal.
- `createStack()`: fungsi untuk membuat stack baru.
- `isEmpty()`: fungsi untuk mengecek apakah stack kosong.
- `isFull()`: fungsi untuk mengecek apakah stack penuh.
- `push()`: fungsi untuk menambah elemen ke stack.
- `pop()`: fungsi untuk mengeluarkan elemen teratas dari stack.
- `precedence()`: fungsi untuk menentukan prioritas operator.
- `isOperator()`: fungsi untuk mengecek apakah char merupakan operator.
- `infixToPostfix()`: fungsi untuk mengonversi ekspresi infix ke postfix.
- `postfixToInfix()`: fungsi untuk mengonversi ekspresi postfix ke infix.

- reverseString(): fungsi untuk membalikkan string.
- replaceBrackets(): fungsi untuk membalikkan tanda kurung.
- infixToPrefix(): fungsi untuk mengonversi ekspresi infix ke prefix.
- prefixToInfix(): fungsi untuk mengonversi ekspresi prefix ke infix.
- postfixToPrefix(): fungsi untuk mengonversi ekspresi postfix ke prefix.
- prefixToPostfix(): fungsi untuk mengonversi ekspresi prefix ke postfix.

C. Metode Struktur Data yang Digunakan

Struktur data yang digunakan dalam kode program adalah stack. Stack digunakan dalam berbagai operasi konversi ekspresi, seperti dari infix ke postfix, postfix ke infix, infix ke prefix, dan sebagainya.

Struktur stack didefinisikan sebagai berikut:

```
typedef struct
{
    int top;
    char *items[MAX_SIZE];
} Stack;
```

Stack diimplementasikan dengan metode berikut:

- createStack(): Membuat stack baru.
- isEmpty(): Mengecek apakah stack kosong.
- isFull(): Mengecek apakah stack penuh.
- push(): Menambahkan elemen ke dalam stack.
- pop(): Mengeluarkan elemen dari stack.

D. Jumlah Fungsi yang Terdapat dalam Kode Program (Tidak Termasuk main)

Dari daftar nama fungsi di bagian (a), terdapat 17 fungsi selain main().