

**“Konversi Ekspresi Infix, Prefix, dan Postfix Menggunakan  
Struktur Data Stack dalam Bahasa C”**

Oleh:

**1.Muhammad Fazel Mawla / 2408107010074**

**2.Faris Fudhaili / 2408107010040**



**PROGRAM STUDI INFORMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS SYIAH KUALA**

**2025**

## A. Nama Fungsi yang Digunakan

Fungsi-fungsi yang ditemukan dalam kode program adalah:

- `init` → Menginisialisasi stack dengan menetapkan nilai awal.
- `isEmpty` → Mengecek apakah stack kosong.
- `isFull` → Mengecek apakah stack penuh.
- `push` → Menambahkan elemen ke dalam stack.
- `priority` → Menentukan prioritas operator dalam ekspresi matematika.
- `strrev` → Membalikkan string.
- `infixToPostfix` → Mengonversi ekspresi infix menjadi postfix.
- `postfixToInfix` → Mengonversi ekspresi postfix menjadi infix.
- `infixToPrefix` → Mengonversi ekspresi infix menjadi prefix.
- `prefixToInfix` → Mengonversi ekspresi prefix menjadi infix.
- `prefixToPostfix` → Mengonversi ekspresi prefix menjadi postfix.
- `postfixToPrefix` → Mengonversi ekspresi postfix menjadi prefix.
- `main` → Mengontrol alur program dengan menampilkan menu, menerima input pengguna, memanggil fungsi konversi ekspresi yang sesuai, dan menampilkan hasil.

## B. Metode Struktur Data yang Digunakan

Program ini menggunakan struktur data **Stack (tumpukan)**.

- Stack bekerja dengan prinsip **LIFO** (Last In, First Out), yaitu elemen yang terakhir dimasukkan adalah yang pertama dikeluarkan.
- Stack digunakan dalam banyak aplikasi pemrograman, termasuk konversi ekspresi matematika, pemrosesan rekursi, dan manajemen memori.

### Implementasi Stack dalam Program:

1. Stack diimplementasikan menggunakan array dengan kapasitas maksimum yang telah ditentukan (MAX).
2. **Variabel top** digunakan untuk melacak elemen terakhir dalam stack.
3. Operasi utama stack dalam program ini:
  - **push()** → Menambahkan elemen ke dalam stack.
  - **pop()** → Menghapus dan mengembalikan elemen teratas stack.
  - **peek()** → Melihat elemen teratas tanpa menghapusnya.
  - **isEmpty()** dan **isFull()** digunakan untuk memeriksa status stack.

### Penggunaan Stack dalam Konversi Ekspresi:

- Dalam konversi **infix ke postfix** dan **infix ke prefix**, stack digunakan untuk **menyimpan operator** sementara ekspresi diproses.
- Dalam konversi **postfix ke infix** dan **prefix ke infix**, stack digunakan untuk **menyimpan operand** dan membangun ulang ekspresi.

### C. Jumlah Fungsi dalam Kode Program (tidak termasuk main)

Terdapat **14 fungsi dalam kode program** yang mendukung operasi stack dan konversi ekspresi.