

UTS KELOMPOK 5

*disusun untuk memenuhi tugas praktikum
Struktur Data dan Algoritma*

Oleh:

Arifa Muthmainnah	(2308107010012)
Thahira Riska	(2308107010024)



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
2025**

A. Pendahuluan

Program ini ditulis untuk melakukan konversi aritmatika antara bentuk Infix, Postfix, dan Prefix. Program menggunakan struktur data stack untuk membantu dalam proses konversi. Program ini dapat menangani ekspresi dengan operator $-$, $+$, $*$, $/$, dan $^$, serta mendukung penggunaan tanda kurung. Program akan terus menampilkan menu setelah menampilkan output, dan program dapat berhenti jika pengguna memilih opsi **7. Keluar**.

B. Nama fungsi yang digunakan

Berikut adalah fungsi-fungsi yang digunakan dalam kode program:

- `initStack(Stack *s)`: Menginisialisasi stack.
- `isEmpty(Stack *s)`: Mengecek apakah stack kosong.
- `isFull(Stack *s)`: Mengecek apakah stack penuh.
- `push(Stack *s, char item[])`: Menambahkan elemen (string) ke dalam stack.
- `pop(Stack *s)`: Menghapus dan mengembalikan elemen teratas dari stack.
- `peek(Stack *s)`: Melihat elemen teratas stack tanpa menghapusnya.
- `isOperator(char ch)`: Mengecek apakah karakter adalah operator ($+$, $-$, $*$, $/$, $^$).
- `precedence(char ch)`: Menentukan prioritas operator.
- `infixToPostfix(char infix[], char postfix[])`: Konversi infix ke postfix.
- `reverseString(char str[])`: Membalik string.
- `infixToPrefix(char infix[], char prefix[])`: Konversi infix ke prefix.
- `postfixToInfix(char postfix[], char infix[])`: Konversi postfix ke infix.
- `prefixToInfix(char prefix[], char infix[])`: Konversi prefix ke infix.
- `prefixToPostfix(char prefix[], char postfix[])`: Konversi prefix ke postfix.
- `postfixToPrefix(char postfix[], char prefix[])`: Konversi postfix ke prefix.
- `main()`: Fungsi utama untuk menjalankan program.

C. Metode struktur data yang digunakan

Kode ini menggunakan struktur data stack untuk menyimpan dan memanipulasi ekspresi aritmatika selama proses konversi. Stack digunakan untuk:

- Menyimpan operator dalam konversi infix ke postfix/prefix.
- Menyimpan operan dan operator dalam konversi postfix/prefix ke infix.

D. Jumlah fungsi yang terdapat di dalam kode program

Kode program ini terdiri dari 15 fungsi tambahan di luar fungsi main(). Fungsi-fungsi ini bertanggung jawab atas berbagai operasi seperti manipulasi stack, pengecekan operator, prioritas operator, serta berbagai konversi ekspresi aritmatika.