LAPORAN UTS PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

PROGRAM STUDI INFORMATIKA 2025

KELAS B

KELOMPOK 13:

FATHURRAHMAN PAHMIPUTRA (2408107010069)

KHALISH AUFA (2408107010068)

ASISTEN LAB:

RAFLI AFRIZA NUGRAHA

ATHAR RAYYAN MUHAMMAD

M. SYAHIDAL AKBAR ZAS

Laporan Analisis Kode Program:

a. Nama Fungsi yang Digunakan

Berikut adalah daftar fungsi yang digunakan dalam program (total 20 fungsi):

1. Operasi Stack untuk Karakter (CharStack)

1. createCharStack(): Inisialisasi stack karakter.
2. pushChar(CharStack *stack, char data): Menambahkan karakter ke stack.
3. popChar(CharStack *stack): Mengambil dan menghapus karakter dari stack.
4. isEmptyChar(CharStack *stack): Memeriksa apakah stack karakter kosong.
5. topChar(CharStack *stack): Mengembalikan karakter di puncak stack.

2. Operasi Stack untuk String (StringStack)

6. createStringStack(): Inisialisasi stack string.
7. pushString(StringStack *stack, const char *data): Menambahkan string ke stack.
8. popString(StringStack *stack): Mengambil dan menghapus string dari stack.
9. isEmptyString(StringStack *stack): Memeriksa apakah stack string kosong.
10. freeStringStack(StringStack *stack): Membebaskan memori stack string.

3. Fungsi Bantu

11. isOperator(char c): Memeriksa apakah karakter adalah operator (+, -, *, /).
12. getPrecedence(char op): Mengembalikan prioritas operator.
13. reverseString(const char *str): Membalikkan string.
14. swapParentheses(char *str): Menukar tanda kurung (↔).

4. Fungsi Konversi

15. infixToPostfix(const char *infix): Infix → Postfix.
16. postfixToInfix(const char *postfix): Postfix → Infix.
17. infixToPrefix(const char *infix): Infix → Prefix.
18. prefixToInfix(const char *prefix): Prefix → Infix.
19. prefixToPostfix(const char *prefix): Prefix → Postfix.
20. postfixToPrefix(const char *postfix): Postfix → Prefix.

b. Metode Struktur Data yang Digunakan

Program menggunakan dua struktur data utama:

1. Stack (Tumpukan)
    - CharStack:
        - Implementasi: *Linked list* untuk menyimpan operator selama konversi Infix ↔ Postfix/Prefix.
        - Digunakan untuk:
            - Menyimpan operator sementara.
            - Mengatur prioritas operator (*precedence*).
            - Menangani tanda kurung.
    - StringStack:
        - Implementasi: *Linked list* untuk menyimpan string (operand atau ekspresi sementara).
        - Digunakan untuk:
            - Membangun ekspresi Infix dari Postfix/Prefix.
            - Menyimpan kombinasi operand dan operator selama konversi.
2. Linked List
    - Digunakan sebagai dasar implementasi stack:
        - Setiap node pada CharStack dan StringStack menggunakan linked list untuk menyimpan data dan pointer ke node berikutnya.

c. Jumlah Fungsi dalam Kode

Total fungsi yang digunakan: 20 fungsi, tidak termasuk fungsi main().

Kategori Fungsi:

- Stack Operations: 10 fungsi (5 untuk CharStack, 5 untuk StringStack).
- Helper Functions: 4 fungsi.
- Conversion Functions: 6 fungsi.