Nama: Azka Zafran Andiani

Kelas: IF-03-02

NIM: 1203230021

LAPORAN PRAKTIKUM ASD

A. Soal 1
   1. Source code dan penjelasan

```c
#include <stdio.h>
#define MAX_LENGTH 20

struct Node
{
    struct Node *link;
    char alphabet;
};
typedef struct Node Node;

int main()
{
    // inisialisasi struct yang akan dikaitkan
    Node l1, l2, l3, l4, l5, l6, l7, l8, l9;

    l1.link = NULL;
    l1.alphabet = 'F';
    l2.link = NULL;
    l2.alphabet = 'M';
    l3.link = NULL;
    l3.alphabet = 'A';
    l4.link = NULL;
    l4.alphabet = 'I';
    l5.link = NULL;
    l5.alphabet = 'K';
```

```
    l6.link = NULL;
    l6.alphabet = 'T';
    l7.link = NULL;
    l7.alphabet = 'N';
    l8.link = NULL;
    l8.alphabet = 'O';
    l9.link = NULL;
    l9.alphabet = 'R';

    // mengurutkan posisi struct di dalam array
    Node *arrayNode[9] = {&l7, &l1, &l8, &l2, &l5, &l3, &l6, &l9, &l4};

    // mengaitkan semua struct
    for(int i = 0; i < 9 - 1; i++)
    {
        arrayNode[i]->link = arrayNode[i+1];
    }
    arrayNode[8]->link = arrayNode[0];

    // insialisasi variabel pondasi yang mencakup semua struct yang
berkaitan
    Node *current = arrayNode[0];
    // variabel yang menyimpan rangkaian kata yang dibuat user
    char kalimat[MAX_LENGTH];

    int state = 1;
    int count = 0, choice, posChar = -1;

    // main program
    printf("Rangkai sebuah kata!\n");
    while(state)
    {
        printf("Rangkaian kata (%d/%d):", posChar+1, MAX_LENGTH);
        for(int i = 0; i < posChar+1; i++)
```

```
        {
            printf("%c", kalimat[i]);
        }
        printf("\nRute: ");
        for(int i = 0; i <= count; i++)
        {
            printf("Link%d->", i+1);
        }
        printf("%c\n", current->alphabet);
        printf("1. Insert character\n2. Next node\n3. Exit
program\nPilih menu: ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            posChar++;
            if(posChar < MAX_LENGTH)
            {
                kalimat[posChar] = current->alphabet;
            }
            else
            {
                printf("ERROR: Jumlah karakter mencapai batas
maksimal.\n");
                state = 0;
            }
            break;
        case 2:
            count++;
            current = current->link;
            if(current->alphabet == 'N')
            {
                count = 0;
            }
```

```
            break;
        case 3:
            state = 0;
            break;
        default:
            break;
        }
    }
    printf("\nHasil akhir rangkaian kata: ");
    for(int i = 0; i < posChar+1; i++)
    {
        printf("%c", kalimat[i]);
    }
    return 0;
}
```

2. Output

```
2. Next node
3. Exit program
Pilih menu: 2
Rangkaian kata (9/20):INFORMATI
Rute: Link1->Link2->Link3->Link4->M
1. Insert character
2. Next node
3. Exit program
Pilih menu: 2
Rangkaian kata (9/20):INFORMATI
Rute: Link1->Link2->Link3->Link4->Link5->K
1. Insert character
2. Next node
3. Exit program
Pilih menu: 1
Rangkaian kata (10/20):INFORMATIK
Rute: Link1->Link2->Link3->Link4->Link5->K
1. Insert character
2. Next node
3. Exit program
Pilih menu: 2
Rangkaian kata (10/20):INFORMATIK
Rute: Link1->Link2->Link3->Link4->Link5->Link6->A
Rute: Link1->Link2->Link3->Link4->Link5->Link6->A
1. Insert character
2. Next node
3. Exit program
Pilih menu: 1
Rangkaian kata (11/20):INFORMATIKA
Rute: Link1->Link2->Link3->Link4->Link5->Link6->A
1. Insert character
2. Next node
3. Exit program
Pilih menu: 3

Hasil akhir rangkaian kata: INFORMATIKA·
PS C:\Users\Sudarmadji\Documents\C files\tugas alpro\pertemuan#6>
```

B. Soal 2

1. Source code dan penjelasan

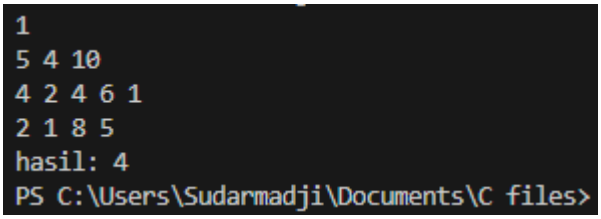
```c
#include <stdio.h>


int pop(int Data[], int *top)
{
    int newTop = *top;
    int temp = Data[newTop];
    Data[newTop] = 0;
    (*top)--;
```

```
    return temp;
}

int peek(int Data[], int top)
{
    return Data[top];
}

int maxSelect(int Data1[], int topIndex1, int Data2[], int topIndex2,
int maxSum)
{
    // total nilai yang telah dikeluarkan
    int sum = 0;
    // jumlah nilai yang telah dikeluarkan
    int removed = 0;

    // while loop akan terus berjalan jika belum ada stack yang kosong
    while(topIndex1 >= 0 && topIndex2 >= 0)
    {
        // ada pengecualian jika sum ditambah data teratas dari salah
satu stack
        // melebihi maxSum maka while loop akan berhenti
        if(sum + peek(Data1, topIndex1) > maxSum && sum + peek(Data2,
topIndex2) > maxSum)
        {
            break;
        }
        // jika kondisi tidak sesuai maka akan lanjut ke pengecekan
utama
        // if else ini akan mengecek apabila kondisi antara data
teratas stack 1 dengan stack dua
        // 'lebih dari', 'kurang dari', atau 'sama'.
        // jika 'kurang dari' maka program akan mengambil data teratas
dari stack 1 dan ditambahkan
```

```
        // ke dalam sum
        if(Data1[topIndex1] < Data2[topIndex2])
        {
            sum += pop(Data1, &topIndex1);
            removed++;
        }
        // jika 'lebih dari' maka yang diambil adalah data teratas dari
stack 2
        else if(Data2[topIndex2] < Data1[topIndex1])
        {
            sum += pop(Data2, &topIndex2);
            removed++;
        }
        // jika 'sama' maka akan melakukan pengecekan lagi yang hampir
sama dengan if else diatas
        // hanya yang dicek adalah data teratas kedua
        else
        {
            if(Data1[topIndex1-1] < Data2[topIndex2-1])
            {
                sum += pop(Data1, &topIndex1);
                removed++;
            }
            else if(Data1[topIndex1-1] > Data2[topIndex2-1])
            {
                sum += pop(Data2, &topIndex2);
                removed++;
            }
        }
    }


    return removed;
}
```

```c
int main()
{
    int round;
    int hasil;

    // user input berapa ronde game
    scanf("%d", &round);
    // main program
    for(int i = 0; i < round; i++)
    {
        int n, m, maxSum;
        scanf("%d %d %d", &n, &m, &maxSum);

        // stack 1 dan 2
        int stack1[n];
        int stack2[m];

        // input n dan m data berupa integer ke dalam stack 1 dan stack
2
        for(int i = n-1; i >= 0; i--)
        {
            scanf("%d", &stack1[i]);
        }
        for(int i = m-1; i >= 0; i--)
        {
            scanf("%d", &stack2[i]);
        }
        hasil = maxSelect(stack1, n-1, stack2, m-1, maxSum);
        printf("hasil: %d\n", hasil);
    }

    return 0;
}
```

2. Output


1
5 4 10
4 2 4 6 1
2 1 8 5
hasil: 4
PS C:\Users\Sudarmadji\Documents\C files>