

Nama : Rafif Muhammad

NIM : 1203230018

Kelas : IF 03-03

## Laporan Praktikum

### a. 1. SourceCode

```
1  #include <stdio.h>
2
3  struct kata{
4      char alphabet;
5      struct kata *link;
6  };
7
8  int main(){
9
10     struct kata l1, l2, l3, l4, l5, l6, l7, l8, l9;
11
12     l1.link = NULL;
13     l1.alphabet = 'F';
14
15     l2.link = NULL;
16     l2.alphabet = 'M';
17
18     l3.link = NULL;
19     l3.alphabet = 'A';
20
21     l4.link = NULL;
22     l4.alphabet = 'I';
23
24     l5.link = NULL;
25     l5.alphabet = 'K';
26
27     l6.link = NULL;
28     l6.alphabet = 'T';
29
30     l7.link = NULL;
31     l7.alphabet = 'N';
32
33     l8.link = NULL;
34     l8.alphabet = 'O';
35
36     l9.link = NULL;
37     l9.alphabet = 'R';
```

```
1  l7.link = &l1;
2  l1.link = &l8;
3  l8.link = &l2;
4  l2.link = &l5;
5  l5.link = &l3;
6  l3.link = &l6;
7  l6.link = &l9;
8  l9.link = &l4;
9  l4.link = &l7;
10
11
12     printf("%c", l3.link->link->link->alphabet);
13     printf("%c", l3.link->link->link->link->alphabet);
14     printf("%c", l3.link->link->link->link->link->alphabet);
15     printf("%c", l3.link->link->link->link->link->link->alphabet);
16     printf("%c", l3.link->link->alphabet);
17     printf("%c", l3.link->link->link->link->link->link->link->alphabet);
18     printf("%c", l3.alphabet);
19     printf("%c", l3.link->alphabet);
20     printf("%c", l3.link->link->link->alphabet);
21     printf("%c", l3.link->link->link->link->link->link->link->link->alphabet);
22     printf("%c", l3.alphabet);
23     return 0;
24 }
25
```

### 2. Penjelasan

**#include <stdio.h>**

Ini adalah library standar stdio.h, yang diperlukan untuk fungsi input/output standar.

**struct kata{**

**char alphabet;**

**struct kata \*link;**

**};**

Mendefinisikan sebuah struktur kata yang terdiri dari dua anggota: alphabet yang menyimpan sebuah huruf dalam bentuk karakter, dan link yang merupakan pointer ke struktur kata berikutnya dalam linked list.

**int main(){**

**struct kata l1, l2, l3, l4, l5, l6, l7, l8, l9;**

mendeklarasikan variabel bertipe struct kata yang berisi l1 sampai 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';**

alphabet: Menyimpan huruf yang akan direpresentasikan oleh variabel tersebut. Misalnya, l1 akan merepresentasikan huruf 'F', l2 akan merepresentasikan huruf 'M', dan seterusnya.  
link: Menunjukkan ke variabel struct kata berikutnya dalam linked list. Awalnya, semua variabel diatur untuk menunjuk ke NULL, karena belum ada hubungan antar variabel yang terbentuk.

**l7.link = &l1;**  
**l1.link = &l8;**  
**l8.link = &l2;**  
**l2.link = &l5;**  
**l5.link = &l3;**  
**l3.link = &l6;**  
**l6.link = &l9;**  
**l9.link = &l4;**  
**l4.link = &l7;**

menghubungkan setiap variabel dengan variabel berikutnya untuk membuat linked list yang merepresentasikan kata 'INFORMATIKA'

**printf("%c", l3.link->link->link->alphabet);** output untuk huruf I  
**printf("%c", l3.link->link->link->link->alphabet);** output untuk huruf N  
**printf("%c", l3.link->link->link->link->link->alphabet);** output untuk huruf F  
**printf("%c", l3.link->link->link->link->link->link->alphabet);** output untuk huruf

O

**printf("%c", l3.link->link->alphabet);** output untuk huruf R

**printf("%c", l3.link->link->link->link->link->link->alphabet);** output untuk huruf M

**printf("%c", l3.alphabet);** output untuk huruf A

**printf("%c", l3.link->alphabet);** output untuk huruf T

**printf("%c", l3.link->link->link->alphabet);** output untuk huruf I

**printf("%c", l3.link->link->link->link->link->link->link->alphabet);** output untuk huruf K

**printf("%c", l3.alphabet);** output untuk huruf A

Mencetak karakter yang dimulai dari variabel l3 dalam linked list dan menelusuri linked list dengan jumlah langkah yang berbeda untuk mencetak kata "INFORMATIKA".

**return 0;**

}

mengembalikan nilai 0 menandakan bahwa program telah selesai

### 3. Output

```
PS C:\Users\rafif\Documents> cd "c:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2\"; if ($?) { gcc praktikum6a.c -o prakti
($?) { .\praktikum6a }
INFORMATIKA
PS C:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2> 
```

#### b. 1. SourceCode

```

1 #include <stdio.h>
2
3 int twoStacks(int maxJumlah, int array_a[], int banyak_array_a, int array_b[], int banyak_array_b) {
4     int jumlah = 0, count = 0, i = 0, j = 0;
5
6     printf("step 1: mengambil elemen A hingga max jumlah:\n");
7     while (i < banyak_array_a && jumlah + array_a[i] <= maxJumlah) {
8         jumlah += array_a[i];
9         printf("%d ", array_a[i]);
10        i++;
11    }
12    count = i;
13
14    printf("\nstack A: ");
15    for (int a = 0; a < i; a++) {
16        printf("%d ", array_a[a]);
17    }
18    printf("\nstack B: ");
19    for (int b = 0; b < j; b++) {
20        printf("%d ", array_b[b]);
21    }
22    printf("\n\n");
23
24    while (j < banyak_array_b && i >= 0) {
25        jumlah += array_b[j];
26        j++;
27        while (jumlah > maxJumlah && i > 0) {
28            i--;
29            jumlah -= array_a[i];
30        }
31        if (jumlah <= maxJumlah && i + j > count) {
32            count = i + j;
33        }
34        printf("step %d: mengambil elemen dari B hingga max jumlah\n", j + 1);
35        printf("stack A: ");
36        for (int a = 0; a < i; a++) {
37            printf("%d ", array_a[a]);
38        }
39        printf("\nstack B: ");
40        for (int b = 0; b < j; b++) {
41            printf("%d ", array_b[b]);
42        }
43        printf("\n\n");
44    }
45    return count;
46 }
47

```

```

1 int main() {
2     int g;
3     printf("Masukkan jumlah kasus uji:\n");
4     scanf("%d", &g);
5
6     while (g--) {
7         int banyak_array_a;
8         int banyak_array_b;
9         int maxJumlah;
10
11        scanf("%d %d %d", &banyak_array_a, &banyak_array_b, &maxJumlah);
12
13        int array_a[banyak_array_a];
14        int array_b[banyak_array_b];
15
16        for (int i = 0; i < banyak_array_a; i++)
17            scanf("%d", &array_a[i]);
18
19        for (int i = 0; i < banyak_array_b; i++)
20            scanf("%d", &array_b[i]);
21
22        printf("Maksimum elemen yang dapat diambil: %d\n", twoStacks(maxJumlah, array_a, banyak_array_a, array_b, banyak_array_b));
23    }
24    return 0;
25 }
26
27
28

```

## 2. Penjelasan

### #include <stdio.h>

Ini adalah library standar stdio.h, yang diperlukan untuk fungsi input/output standar.

### int twoStacks(int maxJumlah, int array\_a[], int banyak\_array\_a, int array\_b[], int banyak\_array\_b) {

Mendefinisikan fungsi twoStacks dengan parameter input sebagai jumlah maksimum (maxJumlah), dua array integer (array\_a dan array\_b), dan ukuran masing-masing array (banyak\_array\_a dan banyak\_array\_b).

**int jumlah = 0, count = 0, i = 0, j = 0;**

mendeklarasikan variabel jumlah, count i, dan j

**printf("step 1: mengambil elemen A hingga max jumlah:\n");**

mencetak output step 1: mengambil elemen A hingga max jumlah

**while (i < banyak\_array\_a && jumlah + array\_a[i] <= maxJumlah) {**

Memulai loop while yang akan berjalan selama masih ada elemen dalam tumpukan A ( $i < \text{banyak\_array\_a}$ ) dan jumlah elemen yang telah diambil dari tumpukan A belum melebihi batas maksimum ( $\text{jumlah} + \text{array\_a}[i] \leq \text{maxJumlah}$ ).

```
jumlah += array_a[i];
```

menambah nilai elemen yang diambil dari stack a kedalam jumlah

```
printf("%d ", array_a[i]);
```

mencetak nilai elemen yang diambil dari stack a

```
i++;
```

menambah indeks i

```
}
```

```
count = i;
```

Menyimpan jumlah elemen yang telah diambil dari tumpukan A ke dalam variabel count.

```
printf("\nstack A: ");
```

mencetak stack a :

```
for (int a = 0; a < i; a++) {
```

```
    printf("%d ", array_a[a]);
```

mencetak isi stack a dari elemen pertama sampai elemen terakhir yang telah di ambil

```
    printf("\nstack B: ");
```

mencetak stack b:

```
    for (int b = 0; b < j; b++) {
```

```
        printf("%d ", array_b[b]);
```

Mencetak isi tumpukan B dari elemen pertama hingga elemen terakhir yang telah diambil.

```
    printf("\n\n");
```

```
while (j < banyak_array_b && i >= 0) {
```

Memulai loop while untuk mengambil elemen dari tumpukan B dan memastikan jumlah total elemen dari kedua tumpukan tidak melebihi batas maksimum.

```
    jumlah += array_b[j];
```

meanmbah nilai eleen yang ada pada b kedalam jumlah

```
    j++;
```

menambah indeks j

```
    while (jumlah > maxJumlah && i > 0) {
```

```
        i--;
```

```
        jumlah -= array_a[i];
```

Melakukan pengurangan elemen dari tumpukan A selama jumlah elemen yang diambil melebihi batas maksimum.

```
    if (jumlah <= maxJumlah && i + j > count) {
```

```
        count = i + j;
```

Memperbarui nilai count jika jumlah elemen yang telah diambil dari tumpukan A dan B lebih besar dari nilai sebelumnya.

```
    printf("step %d: mengambil elemen dari B hingga max jumlah\n", j + 1);
```

mencetak langkaah dari elemen stack b

```
    printf("stack A: ");
```

```
    for (int a = 0; a < i; a++) {
```

```
        printf("%d ", array_a[a]);
```

Mencetak isi tumpukan A setelah proses pengambilan elemen dari tumpukan B.

```
printf("\nstack B: ");  
for (int b = 0; b < j; b++) {  
    printf("%d ", array_b[b]);
```

Mencetak isi tumpukan B setelah proses pengambilan elemen dari tumpukan B.

```
printf("\n\n");  
}
```

```
return count;
```

mengembalikan nilai count

```
int main() {
```

fungsi utama dalam sebuah program

```
int g;
```

variabel g untuk menyimpan jumlah kasus uji

```
printf("Masukkan jumlah kasus uji:\n");  
scanf("%d", &g);
```

fungsi untuk menginputkan kasus uji

```
while (g--) {
```

Memulai loop while yang akan berjalan sebanyak g kali, yaitu jumlah kasus uji.

```
int banyak_array_a;  
int banyak_array_b;  
int maxJumlah;
```

Deklarasi variabel lokal untuk menyimpan jumlah elemen tumpukan A, jumlah elemen tumpukan B, dan nilai maksimum.

```
scanf("%d %d %d", &banyak_array_a, &banyak_array_b, &maxJumlah);
```

Menerima input jumlah elemen tumpukan A, jumlah elemen tumpukan B, dan nilai maksimum.

```
int array_a[banyak_array_a];  
int array_b[banyak_array_b];
```

Deklarasi array untuk menampung elemen-elemen tumpukan A dan B.

```
for (int i = 0; i < banyak_array_a; i++)  
    scanf("%d", &array_a[i]);
```

Mengisi array `array_a` dengan elemen-elemen tumpukan A.

```
for (int i = 0; i < banyak_array_b; i++)
```

```
    scanf("%d", &array_b[i]);
```

Mengisi array `array_b` dengan elemen-elemen tumpukan B.

```
printf("Maksimum elemen yang dapat diambil: %d\n", twoStacks(maxJumlah,  
array_a, banyak_array_a, array_b, banyak_array_b));
```

Mencetak jumlah maksimum elemen yang dapat diambil dari kedua tumpukan dengan memanggil fungsi `twoStacks` dan menyediakan parameter yang sesuai.

```
return 0;
```

mengembalikan nilai 0 menandakan bahwa program telah selesai dijalankan

### 3. Output

```
1
5 4 11
4 5 2 1 1
3 1 1 2
step 1: mengambil elemen A hingga max jumlah:
4 5 2
stack A: 4 5 2
stack B:

step 2: mengambil elemen dari B hingga max jumlah
stack A: 4
stack B: 3

step 3: mengambil elemen dari B hingga max jumlah
stack A: 4
stack B: 3 1

step 4: mengambil elemen dari B hingga max jumlah
stack A: 4
stack B: 3 1 1
stack B: 3 1 1

step 5: mengambil elemen dari B hingga max jumlah
stack A: 4
stack B: 3 1 1 2

Maksimum elemen yang dapat diambil: 5
PS C:\Users\rafif\Documents\Templat Office K> cd "c:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2\" ; if ($?) { gcc praktikumb.c -o

Masukkan jumlah kasus uji:
1
5 4 10
4 2 4 6 1
2 1 8 5
step 1: mengambil elemen A hingga max jumlah:
4 2 4
stack A: 4 2 4
stack B:

step 2: mengambil elemen dari B hingga max jumlah
stack A: 4 2
stack B: 2

step 3: mengambil elemen dari B hingga max jumlah
stack A: 4 2
stack B: 2 1

step 4: mengambil elemen dari B hingga max jumlah
stack A:
stack B: 2 1 8

step 5: mengambil elemen dari B hingga max jumlah
stack A:
stack B: 2 1 8 5

Maksimum elemen yang dapat diambil: 4
PS C:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2> 
```