

Nama : Muhammad Naufal Ramadhani

Nim : 1203230036

Kels : IF 03-03

1. Source Code

```
2. #include <stdio.h>
3. #include <stdlib.h>
4.
5. // Definisi struktur node untuk representasi huruf
6. struct Node {
7.     char alphabet; // Huruf pada node
8.     struct Node* link; // Pointer ke node berikutnya
9. };
10.
11. int main() {
12.     // Inisialisasi node-node yang telah diberikan
13.     struct Node node1 = {'I', NULL};
14.     struct Node node2 = {'N', NULL};
15.     struct Node node3 = {'F', NULL};
16.     struct Node node4 = {'O', NULL};
17.     struct Node node5 = {'R', NULL};
18.     struct Node node6 = {'M', NULL};
19.     struct Node node7 = {'A', NULL};
20.     struct Node node8 = {'T', NULL};
21.     struct Node node9 = {'I', NULL};
22.     struct Node node10 = {'K', NULL};
23.     struct Node node11 = {'A', NULL};
24.
25.     // Membuat link-node sesuai dengan urutan yang diinginkan
26.     node1.link = &node2;
27.     node2.link = &node3;
28.     node3.link = &node4;
29.     node4.link = &node5;
30.     node5.link = &node6;
31.     node6.link = &node7;
32.     node7.link = &node8;
33.     node8.link = &node9;
34.     node9.link = &node10;
35.     node10.link = &node11;
36.     node11.link = NULL;
37.
38.     // Akses data menggunakan node 1 sebagai starting point
39.     printf("Output: ");
```

```

40.     struct Node* current = &node1;
41.     while (current != NULL) {
42.         printf("%c", current->alphabet);
43.         current = current->link;
44.     }
45.     printf("\n");
46.
47.     return 0;
48. }

```

2. Penjelasan

```
include <stdio.h>
```

```
#include <stdlib.h>
```

Baris 1 dan 2: Mendefinisikan dua header file standar dalam bahasa C, yaitu <stdio.h> untuk fungsi input-output standar, dan <stdlib.h> untuk fungsi-fungsi standar seperti alokasi memori dinamis.

```
// Definisi struktur node untuk representasi batu
```

```
struct Node {
```

```
    char alphabet; // Huruf pada batu
```

```
    struct Node* link; // Pointer ke batu berikutnya
```

```
};
```

Baris 5-9: Mendefinisikan sebuah struktur Node yang memiliki dua anggota, yaitu alphabet untuk menyimpan huruf pada batu dan link yang merupakan pointer ke batu berikutnya.

```
int main() {
```

```
    // Inisialisasi node-node yang telah diberikan
```

```
    struct Node l1 = {'N', NULL};
```

```
    struct Node l2 = {'I', NULL};
```

```
    struct Node l3 = {'F', NULL};
```

```
    struct Node l4 = {'O', NULL};
```

```

struct Node l5 = {'R', NULL};
struct Node l6 = {'M', NULL};
struct Node l7 = {'A', NULL};
struct Node l8 = {'T', NULL};
struct Node l9 = {'I', NULL};
struct Node l10 = {'K', NULL};

```

Baris 13-22: Di dalam fungsi main(), dilakukan inisialisasi sejumlah node-node l1 hingga l10 yang masing-masing menyimpan sebuah huruf dan memiliki pointer link yang diatur menjadi NULL terlebih dahulu.

```

// Membuat link sesuai dengan urutan yang telah ditentukan
l1.link = &l2;
l2.link = &l3;
l3.link = &l4;
l4.link = &l5;
l5.link = &l6;
l6.link = &l7;
l7.link = &l8;
l8.link = &l9;
l9.link = &l10;
l10.link = NULL;

```

Baris 26-35: Menyusun koneksi antara node-node tersebut sesuai dengan urutan yang diinginkan, dimana setiap node terhubung dengan node berikutnya dalam urutan yang telah ditentukan, dan node terakhir ditandai dengan NULL untuk menunjukkan akhir dari urutan.

```

// Akses data menggunakan l1 sebagai starting point
printf("Output: ");
struct Node* current = &l1;

```

```

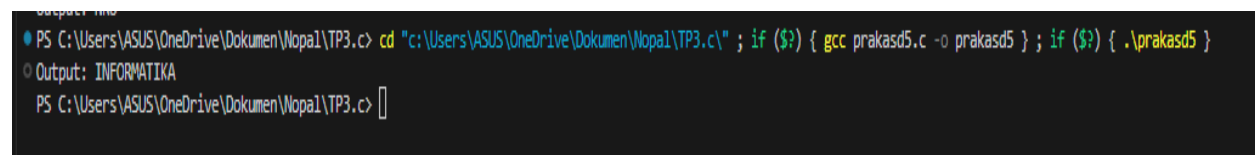
while (current != NULL) {
    printf("%c", current->alphabet);
    current = current->link;
}
printf("\n");

return 0;
}

```

Baris 39-50: Mulai dari node pertama l1, program menelusuri urutan node-node yang telah dibuat dan mencetak huruf-huruf pada masing-masing node sampai mencapai akhir urutan, ditandai dengan NULL

3. SS Output



```

Output: INFO
PS C:\Users\ASUS\OneDrive\Dokumen\Nopa1\TP3.c> cd "c:\Users\ASUS\OneDrive\Dokumen\Nopa1\TP3.c\" ; if ($?) { gcc prakasd5.c -o prakasd5 } ; if ($?) { .\prakasd5 }
Output: INFORMATIKA
PS C:\Users\ASUS\OneDrive\Dokumen\Nopa1\TP3.c>

```