

# **LAPORAN TUGAS PRAKTIKUM 5**

## **STRUKTUR DATA DAN ALGORITMA**



**Disusun oleh:**

Nama	: Muhammad Daffa (221534049)
	Banteng Harisantoso (221524036)
Program Studi	: D4 Teknik Informatika
Kelas	: 1B-TIF4

**POLITEKNIK NEGERI BANDUNG**  
JL. GEGERKALONG HILIR, CIWARUGA, KECAMATAN. PARONGPONG,  
KABUPATEN BANDUNG BARAT, JAWA BARAT  
Telp. (022) 2013789  
2022/2023

## Daftar Isi

I.	Struktur Data.....	2
II.	Modul .....	2
1)	Entry .....	2
2)	Delete.....	3
3)	Program Utama.....	5
III.	Lesson Learn.....	6
IV.	Kendala .....	6
V.	Sumber dan teman yang membantu penyelesaian.....	6

## [LINK GITHUB](#)

### I. Struktur Data

```
//STRUKTUR DATA UNTUK NODE
struct Node{
    char* nama;
    char* kota;
    struct Node* next;
};

//STRUKTUR DATA UNTUK LIST ARRAY
struct ArrayList{
    int jumlah;
    struct Node** array;
};
```

### II. Modul

#### 1) Entry

```
struct ArrayList* buatArrayList(int jumlah)
/*  DESKRIPSI    : Prosedur untuk membuat array list
    I.S          : Tidak ada array list
    F.S          : Sudah teralokasi array list sebanyak variabel jumlah dengan nilai
kosong

*/
{
    struct ArrayList* list = (struct ArrayList*) malloc(sizeof(struct ArrayList));
    list->jumlah = jumlah;
    list->array = (struct Node**) malloc(jumlah * sizeof(struct Node*));
    for(int i = 0; i < jumlah; i++){
        list->array[i]=NULL;
    }

    return list;
}

struct Node* buatNode(char* kota, char* nama)
/*  DESKRIPSI    : Prosedur untuk membuat node dengan data kota dan nama
    I.S          : Node masih kosong
    F.S          : Sudah dibentuk beberapa node dengan data kota dan nama

*/
{
    struct Node* node = (struct Node*) malloc(sizeof(struct Node));

    node->kota = kota;
    node->nama = nama;
```

```

    node->next = NULL;

    return node;
}

void tambahElemen(struct ArrayList* list, int index, char* kota, char* nama)
/* DESKRIPSI    : Menambahkan dan menghubungkan antar node
   I.S          : Elemen belum dibuat
   F.S          : Elemen sudah terbentuk menggunakan prosedur buatNode dan sudah
dihubungkan dengan node lain
*/
{
    struct Node* node = buatNode(kota, nama);
    node->next = list->array[index];
    list->array[index] = node;
}

```

## 2) Delete

Prosedur untuk menghapus salah satu nama di dalam kota.

```

void hapusElemenNama(struct ArrayList* list, char* nama)
/* DESKRIPSI    : Menghapus salah satu elemen di indeks yang sama
   I.S          : Elemen satu nama masih ada dan terhubung
   F.S          : Elemen sudah terhapus dan elemen sebelumnya sudah terhubung dengan
elemen selanjutnya
*/
{
    bool ada = false;
    for(int i = 0; i < list->jumlah ; i++){
        struct Node* first = list->array[i];
        while(first != NULL){
            if(first->nama == nama) {
                if(first == list->array[i]) {
                    // hapus node awal
                    list->array[i] = first->next;
                    first->next = NULL;
                    free(first);
                    ada = true;
                    break;
                }
            }
            else {
                // hapus node tengah atau akhir
                struct Node* temp = list->array[i];
                while(temp->next != first) {
                    temp = temp->next;
                }
            }
        }
    }
}

```

```

        }
        temp->next = first->next;
        first->next = NULL;
        free(first);
        ada = true;
        break;
    }
}
first = first->next;
}

if(ada == false){
    printf("Tidak ada nama %s yang terdata.\n", nama);
}
}

```

**Prosedur untuk menghapus satu kota beserta elemen-elemen yang terkait (Mahasiswa)**

```

void hapusElemenKota(struct ArrayList* list, char* kota)
/* DESKRIPSI    : Menghapus salah satu elemen di indeks yang sama
   I.S          : Elemen kota masih ada dan terhubung
   F.S          : Elemen kota beserta sambungan-sambungannya (Mahasiswa) terhapus
*/
{
    bool ada = false;
    for(int i = 0; i < list->jumlah; i++){
        struct Node* first = list->array[i];
        struct Node* prev = NULL;
        while(first != NULL){
            if(first->kota == kota) {
                if(first == list->array[i]) {
                    // hapus node awal
                    list->array[i] = first->next;
                    first->next = NULL;
                    free(first);
                    first = list->array[i];
                }
                else {
                    // hapus node tengah atau akhir
                    prev->next = first->next;
                    first->next = NULL;
                    free(first);
                    first = prev->next;
                }
            }
            ada = true;
        }
    }
}

```

```

        else {
            prev = first;
            first = first->next;
        }
    }
}

if (ada == false){
    printf("Tidak ada kota %s yang terdata.\n", kota);
}
}

```

### 3) Program Utama

```

int main() {
    //Buatlah array list sejumlah 5
    struct ArrayList* list = buatArrayList(5);

    //Tambahkan elemen dengan parameter(list, indeks, kota, nama)
    tambahElemen(list, 0, "Bandung", "Banteng Harisantoso");
    tambahElemen(list, 0, "Bandung", "Muhammad Daffa");
    tambahElemen(list, 0, "Bandung", "Asep Sudrajat");
    tambahElemen(list, 1, "Jakarta", "Liem Tjaya");
    tambahElemen(list, 1, "Jakarta", "Ferdie Ahmad");
    tambahElemen(list, 2, "Tangerang", "Thoriq M Fadhli");
    tambahElemen(list, 2, "Tangerang", "Muhammad Naruto");
    tambahElemen(list, 3, "Palembang", "Faris Abulkhoir");
    tambahElemen(list, 3, "Palembang", "Raiden Shogun");
    tambahElemen(list, 4, "Cimahi", "Alisha Nara");
    tambahElemen(list, 4, "Cimahi", "Muhammad Fathur R");
    hapusElemenNama(list, "Muhammad Naruto");
    hapusElemenKota(list, "Cimahi");

    //Menampilkan ke layar
    for(int i = 0; i < list->jumlah; i++){
        if(list->array[i] != NULL){
            struct Node* first = list->array[i];
            printf("%s: ", first->kota);
            while(first != NULL){
                printf("%s", first->nama);
                if(first->next != NULL){
                    printf(", ");
                }
                else{
                    printf(".");
                }
            }
        }
    }
}

```

```

    }
    first = first->next;
}
printf("\n");
}

}
}

```

#### 4) Hasil Akhir Program



```

C:\Windows\system32\cmd.exe
Bandung: Asep Sudrajat, Muhammad Deffa, Banteng Harisantoso.
Jakarta: Ferdi Ahmad, Liem Tjaya.
Tangerang: Thoriq M Fadili.
Press any key to continue . . .

```

### III. Lesson Learn

Kita dapat membuat array of linked list secara mudah dengan menggunakan malloc dan pemberian nilai melalui indeks terlebih dahulu. Contoh pada program ini adalah pada perintah:

```

list->array = (struct Node**) malloc(jumlah * sizeof(struct Node*));
for(int i = 0; i < jumlah; i++){
    list->array[i]=NULL;
}

```

### IV. Kendala

Masih sulit memahami bagaimana cara pointer bekerja, bagaimana sebuah sub variabel next bisa menunjuk ke node selanjutnya.

### V. Sumber dan teman yang membantu penyelesaian

Sumber:

1. Geeks For Geeks (Konsep Array of Linked List)
2. Open AI Chat GPT (Contoh Pengaplikasian Konsep Array of Linked List)