Nama : Farhan Adi Suripto
NIM : 222212596
Kelas : 2KS4
Struktur Data

TUGAS PERTEMUAN 3

1. Kerjakan latihan 3 sampai dengan 6 apda akhir bahan tayang (ppt) Pertemuan 3.
   - Latihan 3

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

int main()
{
    int *p, *q;
    p = (int*)malloc(sizeof(int));
    q = (int*)malloc(sizeof(int));

    *p = 3;

     free (q); //q di-free-kan dulu sebelum q = p
    q = p;

    printf("Nilai p = %d\n", *p);
    printf("Nilai q = %d\n", *q);

    printf("%d\n", p);
    printf("%d\n", q);

    free(p);
    return 0;
}
```

Apakah output dari program tersebut?

⇒

```
Nilai p = 3
Nilai q = 3
7870304
7870304
```

   - Latihan 4

```c
#include <stdlib.h>

int main()
{
    int ptr=(int)malloc(sizeof(int));
    return 0;
}
```

```c
#include <stdlib.h>

int main()
```

Mana yang lebih tepat?

⇒ Kode yang dibawah lebih tepat, karena memori yang telah dialokasikan dibebaskan kembali setelah selesai digunakan

```
{
    int ptr=(int)malloc(sizeof(int));
    free(ptr);
    return 0;
}
```

- Latihan 5

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

int main() {
    int *p, *q;
    p = (int*)malloc(sizeof(int));
    q = p;
    *q= 3;
    printf("%d %d", *p, *q);
    free (p);
    printf(" %d %d", *p, *q);
    return 0;
}
```

Apa output dari program tersebut?

⇒

```
3 3 7229968 7229968
```

- Latihan 6

Apa kegunaan fungsi calloc() dan realloc()? Apakah perbedaanya dengan fungsi malloc?

⇒ calloc() dan malloc() memiliki fungsi yang sama yaitu untuk mengalokasikan memori pada Heap. Akan tetapi, calloc() menggunakan beberapa blok memori untuk satu variabel, tidak seperti malloc() yang hanya mengalokasikan blok dengan ukuran tertentu untuk satu variabel. Sedangkan fungsi realloc() berguna untuk mengalokasikan ulang memori dari variabel yang telah dialokasikan oleh malloc() dan calloc(). Perbedaan calloc() dan realloc() dengan malloc() yang lain ialah parameter yang diterimanya ada 2.

2. Buatlah contoh program sederhana yang memuat sebuah structure dengan elemen pointer di dalamnya.

⇒

Source Code

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
typedef struct
{
    char *namep;
    int *agep;
} status;
int main()
{
```

Output

```
Umur = 21
Nama = Farhan Adi
```

```
    status Mystatus;
    Mystatus.agep =  malloc(sizeof(int));

    if((Mystatus.agep) == NULL)
    {
        printf("FAIL TO ALLOCATE MEMORY\n");
        return 0;
    }

    *Mystatus.agep = 21;
    printf("Umur = %d\n",*Mystatus.agep);

    Mystatus.namep = malloc(sizeof(char) *
12);
    if((Mystatus.namep) == NULL)
    {
        free(Mystatus.agep);
        printf("FAIL TO ALLOCATE MEMORY\n");
        return 0;
    }

    strncpy(Mystatus.namep,"Farhan Adi",
(*Mystatus.agep));
    printf("Nama = %s\n",Mystatus.namep);

    return 0;
}
```

3. Buatlah contoh program sederhana yang memuat sebuah pointer yang menunjuk sebuah structure.

⇒

Source Code

```
#include <stdio.h>

typedef struct {
    char name[30];
    char nim[10];
    int age;
    char class[4];
} student;
```

Output

```
Nama    : Farhan Adi
NIM     : 222212596
Umur    : 21
Kelas   : 2KS4

Data Mahasiswa
Nama    : Farhan Adi
NIM     : 222212596
Umur    : 21
Kelas   : 2KS4
```

```c
void main() {
    student mhs;
    student *mhsp=&mhs;

    printf("Nama\t: ");
    scanf(" %[^\n]s", &mhsp->name);
    printf("NIM\t: ");
    scanf("%s", &mhsp->nim);
    printf("Umur\t: ");
    scanf("%d", &mhsp->age);
    printf("Kelas\t: ");
    scanf("%s", &mhsp->class);

    printf("\nData Mahasiswa\n");
    printf("Nama\t: %s\n", mhsp->name);
    printf("NIM\t: %s\n", mhsp->nim);
    printf("Umur\t: %d\n", mhsp->age);
    printf("Kelas\t: %s", mhsp->class);
}
```

4. Buatlah contoh program untuk membuat array dinamis 2 dimensi menggunakan fungsi malloc().

⇒

Source Code

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

void main() {
    int row, column, *matrix;

    printf("Baris: ");
    scanf("%d", &row);
    printf("Kolom: ");
    scanf("%d", &column);

    matrix=malloc(row*column*sizeof(int));

    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            printf("[%d, %d]: ", i+1, j+1);
            scanf("%d", matrix+(i*column+j));
        }
    }
```

Output

```
Baris: 2
Kolom: 3
[1, 1]: 4
[1, 2]: 5
[1, 3]: 1
[2, 1]: 3
[2, 2]: 7
[2, 3]: 2

Tampilkan matriks:
4 5 1
3 7 2
```

```
    printf("\nTampilkan matriks:\n");
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            printf("%d ",
*(matrix+(i*column+j)));
        }
        printf("\n");
    }
}
```