

MODUL 2

MATRIKS

Tujuan :

- Praktikan mampu memahami konsep dari matriks 2 dimensi
- Praktikan mampu mengimplementasikan konsep Matriks untuk menyelesaikan suatu permasalahan
- Praktikan mampu mengembangkan konsep Matriks untuk menyelesaikan permasalahan yang lebih kompleks

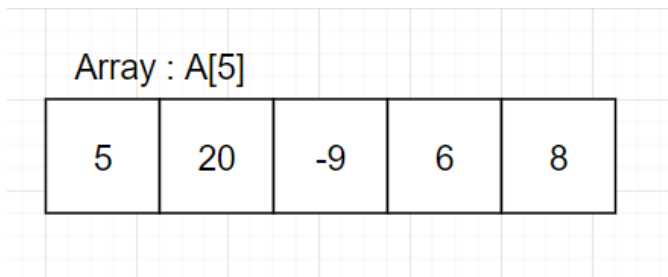
Teori :

Di dunia matematika, Matriks merupakan sekumpulan bilangan yang disusun berdasarkan baris dan kolom, serta ditempatkan di dalam tanda kurung [ruangguru.com]. Matriks kerap kali digunakan untuk memudahkan menyelesaikan permasalahan yang cukup rumit. Contohnya adalah permasalahan yang berhubungan dengan sistem sumbu Cartesian.

Di dunia Informatika, Matriks dapat diimplementasikan menggunakan konsep Array. Konsep Array yang dimaksud adalah Array 2 Dimensi. Array 2 dimensi memungkinkan kita untuk membuat suatu sistem penyimpanan data yang membutuhkan 2 identitas (nantinya akan berupa index) layaknya Matriks. Matriks dapat digunakan untuk menyelesaikan masalah seperti sistem pemesanan kursi bioskop, sistem pemesanan kamar Hotel, *check-in* tempat duduk pesawat, sistem komputer warnet, membuat mekanisme peta *game*, dan masih banyak lagi.

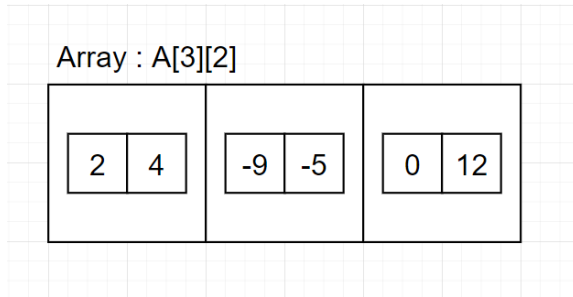
Praktek :

Kita *review* kecil mengenai konsep Array terlebih dahulu. Array yang selama ini teman-teman ketahui dan pelajari adalah Array 1 Dimensi. Array 1 Dimensi memiliki penyimpanan seperti berikut.

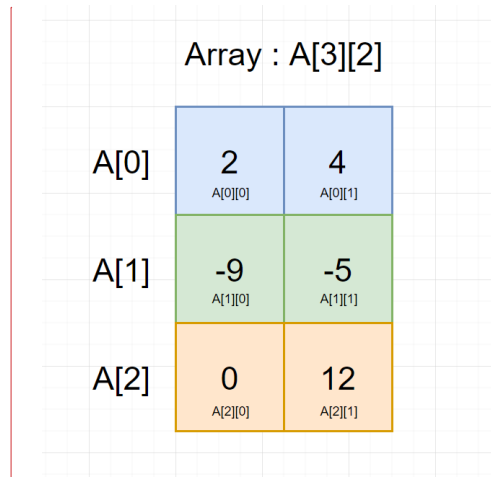


Pada contoh, Array A ditujukan untuk menyimpan 5 buah data seperti yang telah teman-teman pelajari di semester sebelumnya. A[0] menyimpan data 5, A[3] menyimpan data 6, dan A[4] menyimpan data 8. Disini terlihat bahwa Array A hanya memiliki 1 index saja untuk mengakses sebuah Data.

Sedangkan pada Array 2 Dimensi, Suatu Array akan menyimpan Array lainnya di dalam setiap elemen-nya. jika diilustrasikan akan menjadi seperti berikut



Ilustrasi di atas menggambarkan bagaimana penyimpanan Array 2 Dimensi. Terlihat bahwa pada A[0] menyimpan sebuah Array {2,4} dan A[2] menyimpan Array {0,12}. Untuk lebih memudahkan, kita akan sedikit mengubah ilustrasi penyimpanan Array 2 Dimensi tersebut.



Commented [SPASM1]: Gambar ini lebih baik dibuat seperti gambar yang di bawah?
Supaya indeks nya lebih jelas kelihatan?

Bagaimana, Lebih mudah untuk dibayangkan bukan?. Demikianlah cara Array 2 Dimensi berubah bentuk menjadi Matriks. Untuk mengakses tiap elemen-nya, maka teman-teman harus memberikan 2 index yang diperlukan, yakni index untuk Array induk (selanjutnya akan disebut index i) dan index untuk Array yang disimpan (selanjutnya akan disebut index j).

Berikut adalah *list*-nya:

- A[0][0] memberikan nilai 2
- A[0][1] memberikan nilai 4
- A[1][0] memberikan nilai -9
- A[1][1] memberikan nilai -5
- A[2][0] memberikan nilai 0
- A[2][1] memberikan nilai 12.

Jika teman-teman perhatikan, maka index i dan j sebenarnya dapat merepresentasikan sebuah sumbu cartesian. Jika di-ilustrasikan akan menjadi seperti berikut

Array : A[4][4]

	0	1	2	3	Index J (sumbu X)
0	2	4	15	22	
1	-9	101	67	-2	
2	14	-98	54	38	
3	29	25	11	7	
Index i (sumbu y)					

Data Array diganti untuk memudahkan ilustrasi Matriks

Demikianlah penjelasan mengenai Matriks. Silahkan cermati implementasi Array 2 Dimensi pada Guided yaaa... :)

GUIDED

header.h

```
1 #ifndef HEADER
2 #define HEADER
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #include <stdbool.h>
8 #include <conio.h> //laptopku gamau getch kalau gaada conio.. :v
9
10
11 //definisikan kolom dan baris pada matriks yang kita hendak
12 #define ROW 3
13 #define COL 5
14
15 typedef char string[50];
16
17
18 //ini akan menjadi data utama kita
19 typedef struct Data{
20     int value;
21     string status;
22 }Data;
23
24
25
26
27 /*
28  dibawah ini merupakan struct Index.
29  struct ini akan menjadi navigasi untuk
30  index i dan index j yang akan dibutuhkan nanti
31
32  kelebihan menggunakan struct ini adalah teman-teman
33  dapat dengan mudah menemukan index dari sebuah Array N Dimensi
34  Tenang saja, UGD hanya akan menggunakan Array 2D :)
35 */
36 typedef struct Index{
37     int i;
38     int j;
39 }Index;
40
41 void initMatriks(Data data[ROW][COL]);
42 void printData(Data data[ROW][COL]);
43
44 Data buatData(int value, string status);
45 Index cariData(Data data[ROW][COL], int value);
46
47 bool sudahIsi(Data data[ROW][COL], Index index);
48
49
50
51 #endif
```

source.c

```
1 #include "header.h"
2
3 //digunakan untuk menginit Matriks;
4 void initMatriks(Data data[ROW][COL]){
5     int i;
6     int j;
7
8     for(i=0 ; i<ROW ; i++){
9         for(j=0 ; j<COL ; j++){
10             data[i][j].value = -1;
11             strcpy(data[i][j].status, "-");
12         }
13     }
14 }
15
16
17 //digunakan untuk memprint data
18 void printData(Data data[ROW][COL]){
19     int i;
20     int j;
21
22     for(i=0 ; i<ROW ; i++){
23         printf("\t");
24         for(j=0 ; j<COL ; j++){
25             printf("%d\t", data[i][j].value);
26         }
27         printf("\n");
28     }
29 }
30
31
32 //digunakan untuk membuat Data
33 Data buatData(int value, string status){
34     Data temp;
35
36     temp.value = value;
37     strcpy(temp.status, status);
38
39     return temp;
40 }
41
42
43 //untuk mencari data Matriks berada di index mana
44 Index cariData(Data data[ROW][COL], int cari){
45     int i;
46     int j;
47     Index index;
48
49     //set nilai i menjadi -1 sebagai default data tidak ditemukan
50     index.i = -1;
51
52     for(i=0 ; i<ROW ; i++){
53         for(j=0 ; j<COL ; j++){
54             if(data[i][j].value == cari){
55                 index.i = i;
56                 index.j = j;
57                 return index;
58             }
59         }
60     }
61
62     //jika tidak ditemukan, maka i dan j milik index akan bernilai -1;
63     return index;
64 }
65
66
67 //digunakan untuk memeriksa apakah data sudah terisi atau belum
68 bool sudahIsi(Data data[ROW][COL], Index index){
69     if(strcmpi(data[index.i][index.j].status, "-")){
70         return true;
71     }else{
72         return false;
73     }
74 }
75
```

main.c

```
1 #include "header.h"
2
3 int main() {
4
5     //membuat sebuah Matriks Data 3x5
6     /*
7     [ ] [ ] [ ] [ ] [ ]
8     [ ] [ ] [ ] [ ] [ ]
9     [ ] [ ] [ ] [ ] [ ]
10    */
11    Data data[ROW][COL];
12
13    //pendeklarasian variabel-variabel lainnya;
14    int i;
15    int j;
16    int menu;
17    int angka;
18    Index index;
19
20    initMatriks(data);
21
22
23
24
25    do{
26        system("cls");
27        printf("\n\t=== Guided Matriks ===");
28        printf("\nNama : <Nama Praktikkan> ");
29        printf("\nNPM : <NPM Praktikkan> ");
30        printf("\n[1] Isi Data ");
31        printf("\n[2] Cari Data ");
32        printf("\n[3] Print Data ");
33        printf("\n[0] Keluar ");
34
35        printf("\n>>> "); scanf("%d", &menu);
36
37        switch(menu){
38
39            case 1 :
40                printf("\n\t=== Input Data ===\n\n");
41
42                do{
43                    printf("\tIndex I : "); scanf("%d", &index.i);
44                    if(index.i < 0){
45                        printf("\t\t[!] Index tidak boleh < 0\n\n");
46                    }else if(index.i >= ROW){
47                        printf("\t\t[!] Batas ROW adalah %d\n\n", ROW-1);
48                    }else{
49                        break;
50                    }
51                }while(true);
52
53                do{
54                    printf("\tIndex J : "); scanf("%d", &index.j);
55                    if(index.j < 0){
56                        printf("\t\t[!] Index tidak boleh < 0\n\n");
57                    }else if(index.j >= COL){
58                        printf("\t\t[!] Batas COL adalah %d\n\n", COL-1);
59                    }else{
60                        break;
61                    }
62                }while(true);
63
64                //memeriksa apakah sudah terisi untuk index yang diinput;
65                if(sudahIsi(data, index)){
66                    printf("\n\t\t[!] Data Sudah Terisi!");
67                }else{
68                    printf("\n\t\tInput Angka : "); scanf("%d", &angka);
69                    data[index.i][index.j] = buatData(angka, "Sudah Input");
70                    printf("\n\t\t[*] Berhasil Input Data");
71                }
72                break;
73            }
74        }
75    }
76
77    break;
78
79 }
```

```

80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
}

case 2 :
    printf("\n\t=== Cari Data ===\n\n");
    printf("\tInput Angka yang ingin dicari : "); scanf("%d", &angka);

    //disini kita menggunakan fungsi cariData()
    //untuk mengambil index apabila ada;
    index = cariData(data, angka);

    //jika salah satu index negatif, dipastikan data tidak ditemukan
    //dikarenakan Array selalu dimulai dari nol, dan tidak pernah negatif.
    if(index.i == -1){
        printf("\n\t\t[] Data Tidak Ditemukan!");
    }else{
        printf("\n\t\t[] Data Pertama Kali ditemukan pada index");
        printf("\n\t\t\t\t\t i : %d", index.i);
        printf("\n\t\t\t\t\t j : %d", index.j);
    }
    break;

case 3 :
    printf("\n\t=== Print Data ===\n\n");
    printData(data);
    break;

case 0 :
    printf("\n\t\t[] Keluar");
    printf("\n\t\t\t < Nama Praktikkan >");
    printf("\n\t\t\t < NPM yooooooooo >");
    break;

default :
    printf("\n\t\t[] Menu Tidak Ditemukan");
    break;
}
getch();
}while(menu != 0);
return 0;
}

```

Ketentuan & Format Pengumpulan Guided:

1. Untuk comment tidak menjadi masalah jika tidak ditulis di Guided
2. Paling tidak coba compile 1x lah yaa, cuman copas loh.. :v
konsekuensi Guided yang tidak bisa di compile bisa berupa **pengurangan nilai** dan konsekuensi paling parah adalah **tidak boleh mengikuti** dan **mengerjakan** Pretest, UGD, Tugas, dan Laporan.
3. Tindakan kecurangan tidak perlu dibahas lebih lanjut lah yaa.. :)
Skip semua komponen penilaian
4. Teman-teman boleh berkreasi dengan Guided seperti mengganti nama variabel, membuat prosedur lain, mengganti logika program, dll. Dengan ketentuan :
 - Poin 2 dan 3 tetap harus dipatuhi yaa...
 - Fungsionalitas Program tetap terjaga (menu 1, 2, dan 3 jangan diubah-ubah)
 - Pastikan tidak ada error pada penambahan/perubahan code yang teman-teman lakukan ya, atau tetap akan dikenakan pengurangan nilai :(
 - Keuntungannya mungkin saja saya akan memberikan saran atau tips dari program tambahan yang teman-teman ketikkan.. :)Tidak diperkenankan untuk menggunakan modul-modul selanjutnya yaaa...
[Kecuali Recursive Function, Sorting, dan Searching bagi yang sudah mempelajari. selalu ada *Privilege* bagi teman-teman yang *push* belajar]
5. Jangan ragu-ragu untuk bertanya kepada saya ataupun asisten lain yoo... :)
6. Semua kode dimasukkan ke dalam sebuah folder dengan format penamaan: **GD2_X_YYYY**
7. Folder tadi kemudian di Zip dengan format penamaan: **GD2_X_YYYY.zip**
8. Keterangan:
 - X = Kelas
 - YYYY = 5 digit terakhir NPM Praktikkan

HINT UGD & Tugas

1. Silahkan pelajari untuk menu-menu CRUD (Create, Read, Update, Delete)
2. Menu CRUD dapat dibuat hanya dengan menggunakan prosedur dan fungsi pada Guided, tinggal bagaimana teman-teman menggunakannya. Tentu saja jika ingin membuat prosedur dan fungsi lain tetap diperbolehkan