

Array Of Record

Modul 12

Asisten Dasar Pemrograman

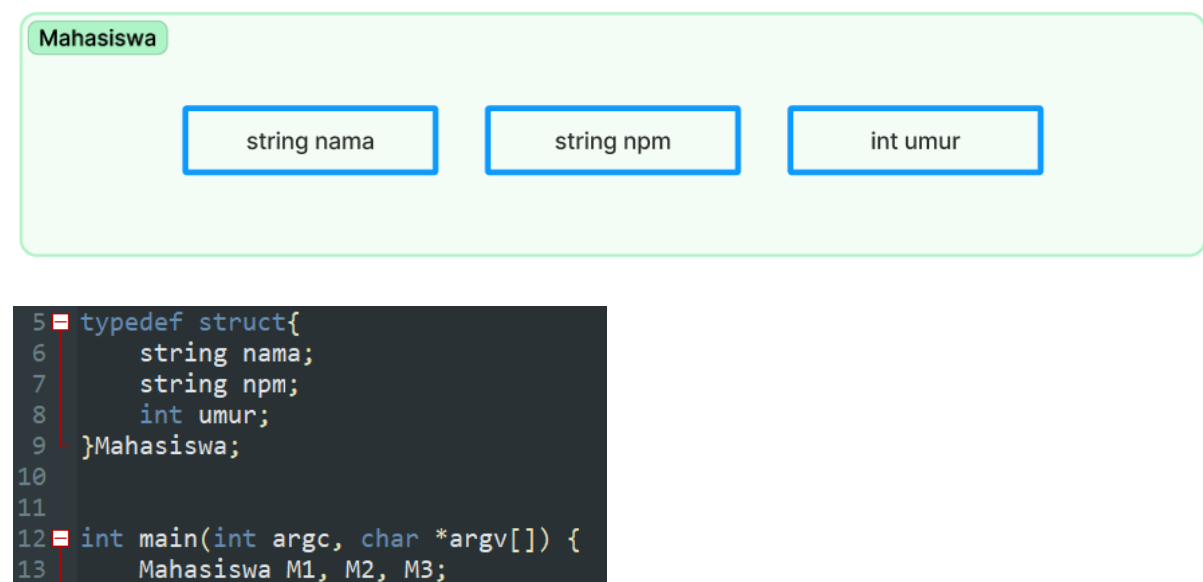
2021/2022 Pemegang Modul : Dandy Candra Oktaviano

TUJUAN

1. Praktikan dapat memahami konsep array of record
2. Praktikan dapat mengimplementasikan array of record pada sebuah studi kasus

Pengantar

Sebelumnya kita sudah mempelajari konsep array dan record pada dasar pemrograman. Array sendiri merupakan sebuah konsep dimana didalam sebuah variabel dapat menyimpan beberapa data / elemen yang mempunyai penunjuk data berupa index. Sedangkan record merupakan sebuah tipe data bentukan yang berisi kumpulan item yang dapat memiliki tipe data yang berbeda – beda. Sebagai contoh, mari kita buat sebuah record dengan tipe data Mahasiswa untuk menyimpan data mahasiswa.



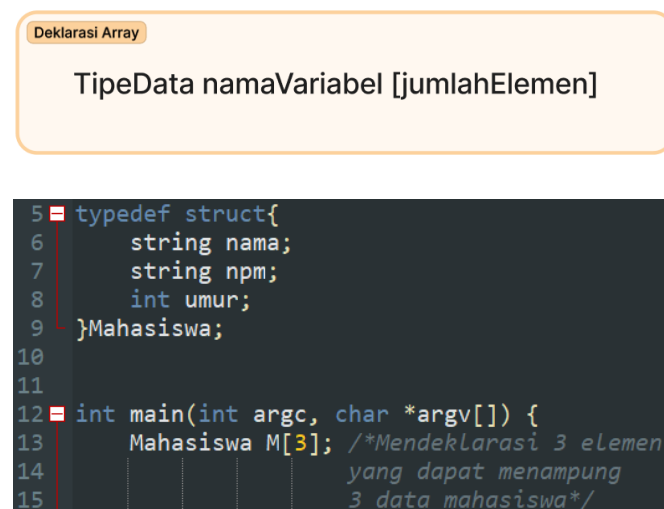
Gambar 1.0

Pada **Gambar 1.0** diperlihatkan bahwa kita membuat sebuah tipe data bentukan yaitu Mahasiswa dimana mempunyai field nama, npm, dan umur. Selanjutnya kita membuat 3 variabel M1, M2 dan M3 dengan tipe data Mahasiswa. Dapat dilihat bahwa semakin banyak jumlah mahasiswa yang ditampung maka semakin banyak juga variabel mahasiswa yang dibuat. Contoh jika kita ingin membuat 5 buah mahasiswa, maka kita akan menciptakan 5 buah variabel (M1, M2, M3, M4 dan M5), hal ini akan menyebabkan code yang dibuat tidak rapi dan memerlukan banyak baris code untuk melakukan pengolahan data pada

masing – masing mahasiswa. Maka solusi yang tepat untuk menyelesaikan studi kasus diatas adalah dengan menggunakan konsep **Array Of Record**.

Pengertian Array Of Record

Array of record merupakan gabungan dari konsep record dan konsep array, dimana penggunaan tipe data bentukan digunakan sebagai elemen dari sebuah array. Jika kita menerapkan studi kasus mahasiswa diatas dengan menggunakan konsep array of record, maka akan menjadi seperti berikut.



Gambar 1.1

Dapat dilihat pada gambar 1.1, dengan kita menggunakan array pada record maka kita dapat menyederhanakan penyederhanaan kode. Untuk konsep deklarasi tetap sama dengan konsep array pada tipe data primer, karena baik tipe data primer maupun tipe data bentukan, keduanya sama sama merupakan sebuah tipe data.

Manipulasi data pada Array Of Record

Setter

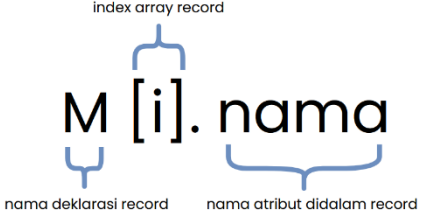
Berikut cara untuk melakukan *set* data pada array of record mahasiswa

```
Mahasiswa M[3]; /*Mendeklarasi 3 elemen
                 yang dapat menampung
                 3 data mahasiswa*/

//Setter
strcpy(M[0].nama, "Dandy");
strcpy(M[0].npm, "200710670");
M[0].umur = 20;

strcpy(M[1].nama, "Nessa");
strcpy(M[1].npm, "200710686");
M[1].umur = 20;

strcpy(M[2].nama, "Vincent");
strcpy(M[2].npm, "200710874");
M[2].umur = 20;
```



Gambar 1.2

Dapat dilihat pada gambar 1.2, bahwa untuk melakukan *set* data pada sebuah array record memiliki cara yang sama dengan array biasa.

Kita juga dapat menggunakan konsep perulangan pada array of record untuk mempercepat penulisan kode serta membuat kode lebih rapi, sebagai contoh kita akan menginisialisasi setiap data pada array record mahasiswa.

```
Mahasiswa M[3]; /*Mendeklarasi 3 elemen
                 yang dapat menampung
                 3 data mahasiswa*/

//Setter
for (i=0; i<3; i++){
    strcpy(M[i].nama, "-");
    strcpy(M[i].npm, "-");
    M[i].umur = 0;
}
```

Gambar 1.3

Getter

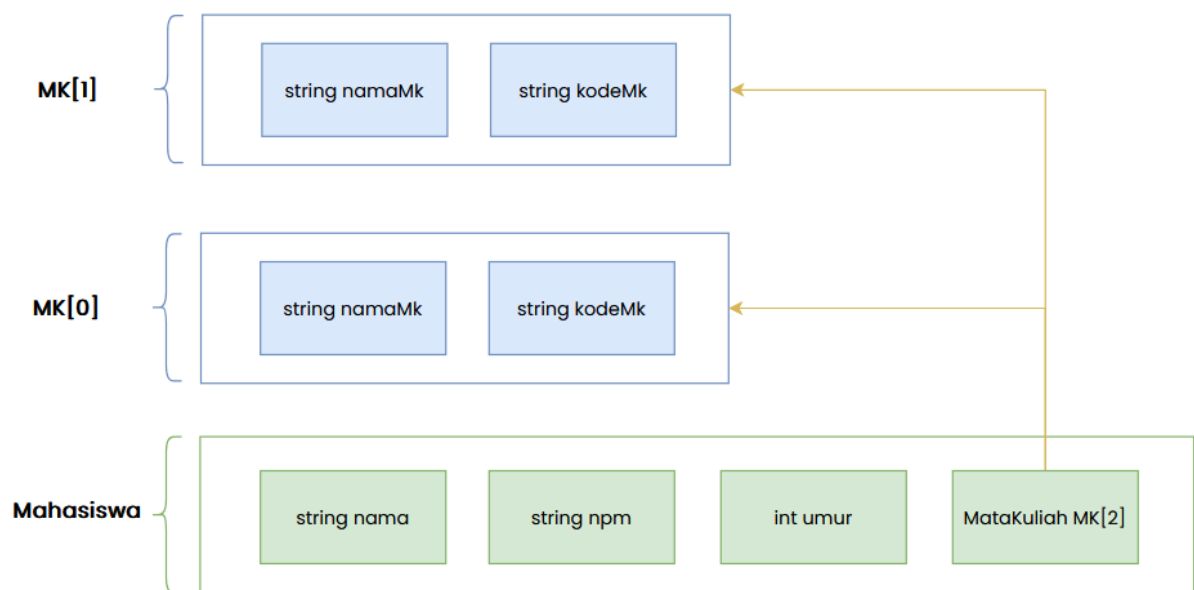
Berikut cara untuk melakukan *get* data pada array of record mahasiswa untuk menampilkan setiap atribut yang disimpan.

```
//Getter
for (i=0; i<3; i++){
    printf("\n Nama : %s",M[i].nama);
    printf("\n NPM : %s",M[i].npm);
    printf("\n Umur : %d",M[i].umur);
}
```

Gambar 1.4

Array Of Record didalam Record

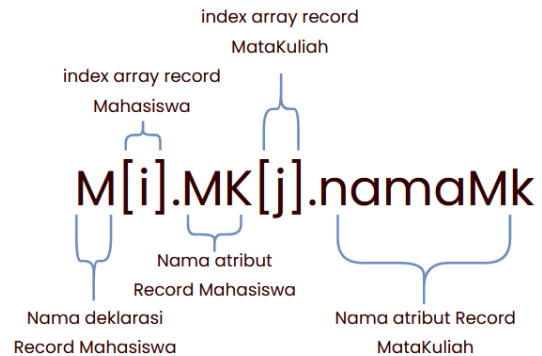
Kondisi dimana dalam suatu record memiliki sebuah atribut berupa array of record. Sebagai contoh didalam record data **Mahasiswa** yang dibuat, kita tambahkan array of record **MataKuliah**, sehingga masing masing **Mahasiswa** dapat menampung beberapa **MataKuliah** yang mereka ampu.



```

8 typedef struct{
9
10     string namaMk;
11     string kodeMk;
12 }MataKuliah;
13
14
15 typedef struct{
16
17     string nama;
18     string npm;
19     int umur;
20     MataKuliah MK[2]; /*Mendeklarasi 2 elemen
21                        yang dapat menampung
22                        2 data MataKuliah*/
23 }Mahasiswa;
24
25
26
27 int main(int argc, char *argv[]) {
28     Mahasiswa M[3]; /*Mendeklarasi 3 elemen
29                     yang dapat menampung
30                     3 data mahasiswa*/

```



Gambar 1.5

Dapat dilihat pada **Gambar 1.5**, ditunjukkan bahwa kita menambahkan atribut baru berupa tipe data bentukan yaitu **MataKuliah**. Selanjutnya kita mendeklarasikan variabel array **MK** yang dapat menampung 2 elemen data. Sehingga dalam 1 elemen **Mahasiswa**, dapat menampung 2 buah data **MataKuliah**.

Setter

```

Mahasiswa M[3]; /*Mendeklarasi 3 elemen
yang dapat menampung
3 data mahasiswa*/

//Set Data
strcpy(M[0].nama, "Dandy");
strcpy(M[0].npm, "200710670");
M[0].umur = 20;
strcpy(M[0].MK[0].namaMk, "Dasar Pemrograman");
strcpy(M[0].MK[0].kodeMk, "INFT06204P");

strcpy(M[0].MK[1].namaMk, "Kewirausahaan");
strcpy(M[0].MK[1].kodeMk, "MKUN06002");

strcpy(M[1].nama, "Nessa");
strcpy(M[1].npm, "200710686");
M[1].umur = 20;
strcpy(M[1].MK[0].namaMk, "Dasar Pemrograman");
strcpy(M[1].MK[0].kodeMk, "INFT06204P");

strcpy(M[1].MK[1].namaMk, "Kewirausahaan");
strcpy(M[1].MK[1].kodeMk, "MKUN06002");

strcpy(M[2].nama, "Vincent");
strcpy(M[2].npm, "200710874");
M[2].umur = 20;
strcpy(M[2].MK[0].namaMk, "Dasar Pemrograman");
strcpy(M[2].MK[0].kodeMk, "INFT06204P");

strcpy(M[2].MK[1].namaMk, "Kewirausahaan");
strcpy(M[2].MK[1].kodeMk, "MKUN06002");

```

```

Mahasiswa M[3]; /*Mendeklarasi 3 elemen
yang dapat menampung
3 data mahasiswa*/

//Set Data Inisialisasi
for(i = 0; i<3; i++){
    strcpy(M[i].nama, "-");
    strcpy(M[i].npm, "-");
    M[i].umur = 0;

    for(j = 0; j<2; j++){
        strcpy(M[i].MK[j].namaMk, "-");
        strcpy(M[i].MK[j].kodeMk, "-");
    }
}

```

Gambar 1.6

Getter

```
//Getter
for(i = 0; i<3; i++){
    printf("\n Nama : %s",M[i].nama);
    printf("\n NPM : %s",M[i].npm);
    printf("\n Umur : %d",M[i].umur);

    for(j = 0; j<2; j++){
        printf("\n Nama Matkul : %s",M[i].MK[j].namaMk);
        printf("\n Kode Matkul : %s",M[i].MK[j].kodeMk);
    }
}
```

Gambar 1.7

Penerapan Array Of Record dengan Prosedur dan Fungsi

Kita telah mengetahui bahwa penggunaan array pada prosedur tidak menggunakan tanda (*) pada variabel yang ingin dilakukan perubahan data. Sehingga pada Array Of Record kita juga tidak perlu memberikan tanda (*) dan (&). Berikut contoh penggunaan prosedur dan fungsi pada Array Of Record, kita akan menggunakan contoh studi kasus yang sama seperti diatas.

```
string nama,npm,namaMk1,kodeMk1,namaMk2,kodeMk2;
int i,j,umur;
Mahasiswa M[3]; /*Mendeklarasi 3 elemen
                 yang dapat menampung
                 3 data mahasiswa*/

//Set Data (Inisialisasi)
inisialisasiData(M);

//User menginputkan data yang akan disimpan pada mahasiswa
printf("\nNama : ");fflush(stdin);gets(nama);
printf("\nNPM : ");fflush(stdin);gets(npm);
printf("\nUmur : ");scanf("%d",&umur);
printf("\nNama Matkul 1 : ");fflush(stdin);gets(namaMk1);
printf("\nKode Matkul 1 : ");fflush(stdin);gets(kodeMk1);
printf("\nNama Matkul 2 : ");fflush(stdin);gets(namaMk2);
printf("\nKode Matkul 2 : ");fflush(stdin);gets(kodeMk2);

//Set Data (Mahasiswa dengan indeks 0 diisi dengan variabel yang tadi sudah diinputkan oleh user)
M[0] = isiData(nama, npm, umur, namaMk1, kodeMk1, namaMk2, kodeMk2);

//Getter
tampilMahasiswa(M);
```

```

void inisialisasiData(Mahasiswa M[]){
    int i,j;
    for(i = 0; i<3; i++){
        strcpy(M[i].nama,"-");
        strcpy(M[i].npm,"-");
        M[i].umur = 0;

        for(j = 0; j<2; j++){
            strcpy(M[i].MK[j].namaMk,"-");
            strcpy(M[i].MK[j].kodeMk,"-");
        }
    }
}

Mahasiswa isiData(string nama, string npm, int umur, string namaMk1, string kodeMk1, string namaMk2, string kodeMk2 ){
    Mahasiswa M;
    strcpy(M.nama,nama);
    strcpy(M.npm,nama);
    M.umur = umur;

    strcpy(M.MK[0].namaMk,namaMk1);
    strcpy(M.MK[0].kodeMk,kodeMk1);

    strcpy(M.MK[1].namaMk,namaMk2);
    strcpy(M.MK[1].kodeMk,kodeMk2);

    return M;
}

void tampilMahasiswa (Mahasiswa M[]){
    int i,j;
    for(i = 0; i<3; i++){
        printf("\n\n Nama : %s",M[i].nama);
        printf("\n NPM : %s",M[i].npm);
        printf("\n Umur : %d\n",M[i].umur);

        for(j = 0; j<2; j++){
            printf("\n Nama Matkul : %s",M[i].MK[j].namaMk);
            printf("\n Kode Matkul : %s",M[i].MK[j].kodeMk);
        }
    }
}

```

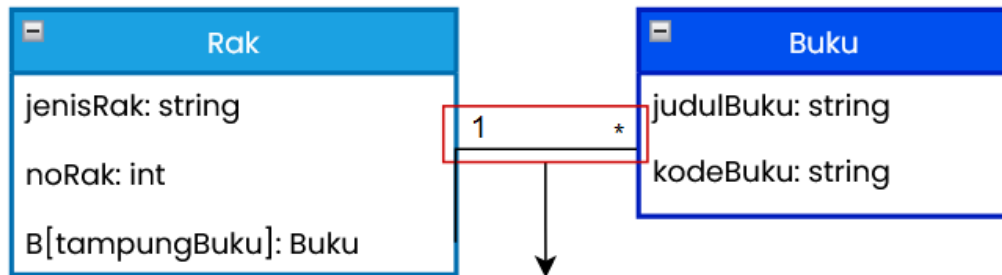
Gambar 1.8

Dapat terlihat bahwa penerapan prosedur dan fungsi pada Array Of Record sama seperti penggunaan array pada umumnya.

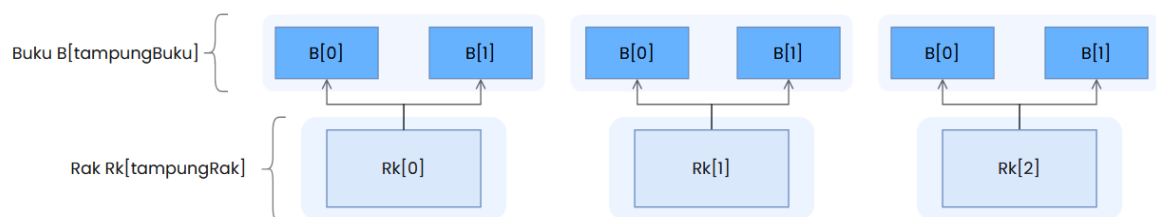
Guided

(Pembahasan lengkap guided terdapat pada akhir halaman modul)

Membuat sebuah program untuk mengolah data buku pada perpustakaan. Terdapat 3 Rak, dimana masing masing rak dapat menampung 2 Buku. Atribut Rak terdiri dari jenisRak (string), noRak (int) dan B[] (Buku). Sedangkan atribut Buku terdiri dari judulBuku (string) dan kodeBuku(string). Berikut diagram kelas nya:



Icon * memiliki arti **banyak data**, jika diartikan maka didalam 1 rak terdapat banyak buku



Dikarenakan setiap rak hanya dapat menampung maksimal 2 buku saja, maka tampungBuku akan diisi dengan 2.

Source code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <stdbool.h>
5
6 #define tampungBuku 2 // jumlah elemen yang dapat ditampung oleh array Buku
7 #define tampungRak 3 // jumlah elemen yang dapat ditampung oleh array Rak
8 typedef char string [50];
9
10 typedef struct{
11     string judulBuku;
12     string kodeBuku;
13 }Buku;
14
15 typedef struct{
16     string jenisRak;
17     int noRak;
18     Buku B[tampungBuku]; // tampung buku berasal dari define pada line 6
19 }Rak;
20
21 void inisialisasiRak(Rak Rk[]); // Inisialisasi Data Array Rak
22 Buku isiDataBuku(string judulBuku, string kodeBuku); // Mengisi data kedalam record Buku
23 Rak isiDataRak(string jenisRak, int noRak, Buku temp[]); // Mengisi data kedalam record Rak
24 int cekEmpty(Rak Rk[]); // Mengecek apakah array Rak mempunyai data yang kosong atau tidak, jika kosong maka akan return i (index yang kosong)
25 int cekIsi(Rak Rk[]); // Return i akan menunjukan elemen rak mana yang sudah diisi, return -1 jika semua elemen pada array Rak masih kosong
26 int searchRak(Rak Rk[], int noRak); // Mencari elemen pada Array Rak berdasarkan noRak, return i jika data yang dicari ada, return -1 sebaliknya
27 void tampilData(Rak Rk[]); // Menampilkan semua elemen data pada array Rak
```

```

27 void tampilData(Rak Rk[]); // Menampilkan semua elemen data pada array Rak
28
29
30 int main(int argc, char *argv[]) {
31
32     Buku temp[tampungBuku]; //tampung buku berasal dari define pada line 6
33     Rak Rk[tampungRak]; //tampung Rak berasal dari define pada line 7
34     string jenisRak, judulBuku, kodeBuku;
35     int pil, i, noRak;
36     bool menu = true;
37
38     inisialisasiRak(Rk);
39
40     while(menu){
41         system("cls");
42         printf("\n === MENU RAK ===");
43         printf("\n[1]. Isi Data Rak");
44         printf("\n[2]. Tampil Data Rak");
45         printf("\n[3]. Delete Data Rak");
46         printf("\n[0]. Exit");
47         printf("\n>> "); scanf("%d",&pil);
48
49         switch(pil){
50             case 1:
51                 if(cekEmpty(Rk)!=-1){
52                     printf("\n[Isi Data Rak]");
53                     printf("\nNo Rak      : "); scanf("%d",&noRak);
54                     printf("Jenis Rak : "); fflush(stdin); gets(jenisRak);
55
56
57                     printf("\n\t[Isi Data Buku]");
58
59                     for(i = 0; i < tampungBuku; i++){
60                         printf("\n\tData Buku Ke - %d", (i+1));
61                         printf("\n\tJudul Buku : "); fflush(stdin); gets(judulBuku);
62                         printf("\tKode Buku   : "); fflush(stdin); gets(kodeBuku);
63                         temp[i] = isiDataBuku(judulBuku, kodeBuku);
64                     }
65
66                     Rk[cekEmpty(Rk)] = isiDataRak(jenisRak, noRak, temp);
67
68                     printf("\n\tBerhasil Tambah Data");
69                 }else{
70                     printf("\nData Full [!]");
71                 }
72
73                 break;

```

```

72
73         break;
74
75     case 2:
76         if(cekIsi(Rk)!=-1){
77
78             tampilData(Rk);
79
80         }else{
81             printf("\nSemua Rak Masih Kosong [!]);
82         }
83
84         break;
85
86     case 3:
87         if(cekIsi(Rk)!=-1){
88
89             printf("\nNoRak yang ingin dihapus : ");scanf("%d",&noRak);
90             if(searchRak(Rk, noRak)!=-1){
91                 for(i=0; i < tampungBuku; i++){
92                     strcpy(temp[i].judulBuku,"-");
93                     strcpy(temp[i].kodeBuku,"-");
94                 }
95                 Rk[searchRak(Rk,noRak)] = isiDataRak("-", 0, temp);
96
97                 printf("\nBerhasil Delete Data [!]);
98             }else{
99                 printf("\nData Tidak Ditemukan [!]);
100             }
101
102
103         }else{
104             printf("\nSemua Rak Masih Kosong [!]);
105         }
106
107         break;
108
109     case 0:
110         printf(" Have a nice day [!]);
111         printf("\n NAMA / NPM / KELAS");
112         menu = false;
113         break;
114
115
116     default:
117         break;
118
119 }getch();

```

```

120     }
121
122     return 0;
123 }
124
125 void inisialisasiRak(Rak Rk[]){
126     int i,j;
127
128     for(i=0 ; i < tampungRak; i++){
129         strcpy(Rk[i].jenisRak,"-");
130         Rk[i].noRak = 0;
131
132         for(j=0; j < tampungBuku; j++){
133             strcpy(Rk[i].B[j].judulBuku,"-");
134             strcpy(Rk[i].B[j].kodeBuku,"-");
135         }
136     }
137 }
138
139 Buku isiDataBuku(string judulBuku, string kodeBuku){
140     Buku B;
141     strcpy(B.judulBuku, judulBuku);
142     strcpy(B.kodeBuku, kodeBuku);
143     return B;
144 }
145
146 Rak isiDataRak(string jenisRak, int noRak, Buku temp[]){
147     Rak Rk;
148     int i;
149
150     Rk.noRak = noRak;
151     strcpy(Rk.jenisRak,jenisRak);
152     for(i = 0 ; i < tampungBuku; i++){
153         Rk.B[i] = temp[i];
154     }
155
156     return Rk;
157 }
158
159 int cekEmpty(Rak Rk[]){
160     int i;
161     for(i = 0; i < tampungRak; i++){
162         if(strcmp(Rk[i].jenisRak,"-")==0){
163             return i;
164         }
165     }
166     return -1;
167 }

```

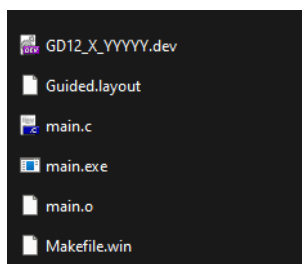
```

168
169 int cekIsi(Rak Rk[]){
170     int i;
171     for(i = 0; i < tampungRak; i++){
172         if(strcmp(Rk[i].jenisRak,"-")!=0){
173             return i;
174         }
175     }
176     return -1;
177 }
178
179 void tampilData(Rak Rk[]){
180     int i,j;
181
182     printf("\n[DATA RAK]");
183     for(i = 0; i < tampungRak; i++){
184         if(strcmp(Rk[i].jenisRak,"-")!=0){
185             printf("\nNo Rak      : %d",Rk[i].noRak);
186             printf("\nJenis Rak   : %s\n",Rk[i].jenisRak);
187
188             printf("\n\n\t[DATA BUKU]");
189             for(j = 0; j < tampungBuku; j++){
190                 printf("\n\n\tData Buku Ke - %d", (j+1));
191                 printf("\n\tJudul Buku  : %s", Rk[i].B[j].judulBuku);
192                 printf("\n\tKode Buku   : %s\n", Rk[i].B[j].kodeBuku);
193             }
194         }
195     }
196 }
197
198
199 int searchRak(Rak Rk[], int noRak){
200     int i;
201     for(i = 0; i < tampungRak; i++){
202         if(Rk[i].noRak == noRak)
203             return i;
204     }
205     return -1;
206 }

```

Ketentuan Pengerjaan :

Jangan Hanya Mengumpulkan File .c nya saja.



Semua file tersebut dijadikan 1 folder lalu di zip dengan ketentuan penamaan dibawah.

Format penamaan guided :

- **GD12_X_YYYYY.zip**

Keterangan format penamaan :

X = kelas

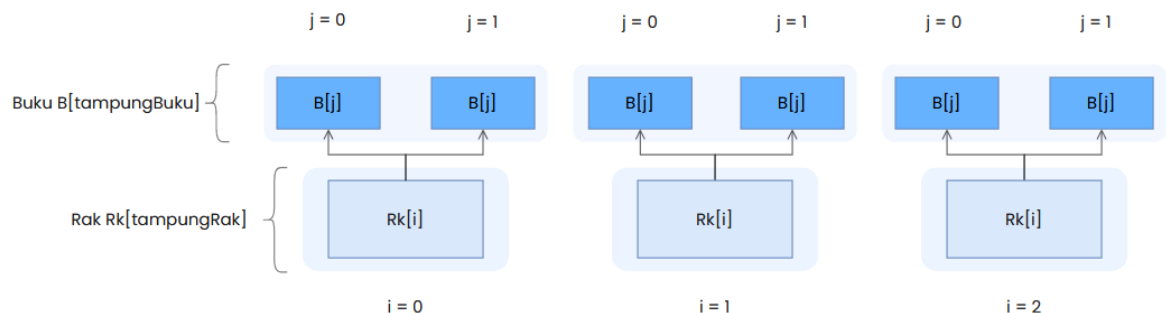
Y = 5 digit NPM terakhir

Pembahasan Guided

1. Inisialisasi Data

```
125 void inisialisasiRak(Rak Rk[]){
126     int i,j;
127
128     for(i=0 ; i < tampungRak; i++){
129         strcpy(Rk[i].jenisRak,"-");
130         Rk[i].noRak = 0;
131
132         for(j=0; j < tampungBuku; j++){
133             strcpy(Rk[i].B[j].judulBuku,"-");
134             strcpy(Rk[i].B[j].kodeBuku,"-");
135         }
136     }
137 }
```

Dapat dilihat pada prosedur diatas, digunakan untuk melakukan inisialisasi data pada array Rak, dimana string akan diisi dengan default value “-” dan integer akan diisi dengan ‘0’. Inisialisasi pada Array Of Record Rak dapat menggunakan konsep nested loop, dimana variabel i akan menjadi penunjuk pada array Rak, sedangkan variabel j akan menjadi penunjuk pada array Buku.



Dapat dilihat pada gambar diatas bahwa value kondisi perulangan i dan j dapat dijadikan sebagai index pada Array Of Record Rak yang digunakan untuk mengunjungi setiap atribut yang ada untuk melakukan inisialisasi setiap elemen.

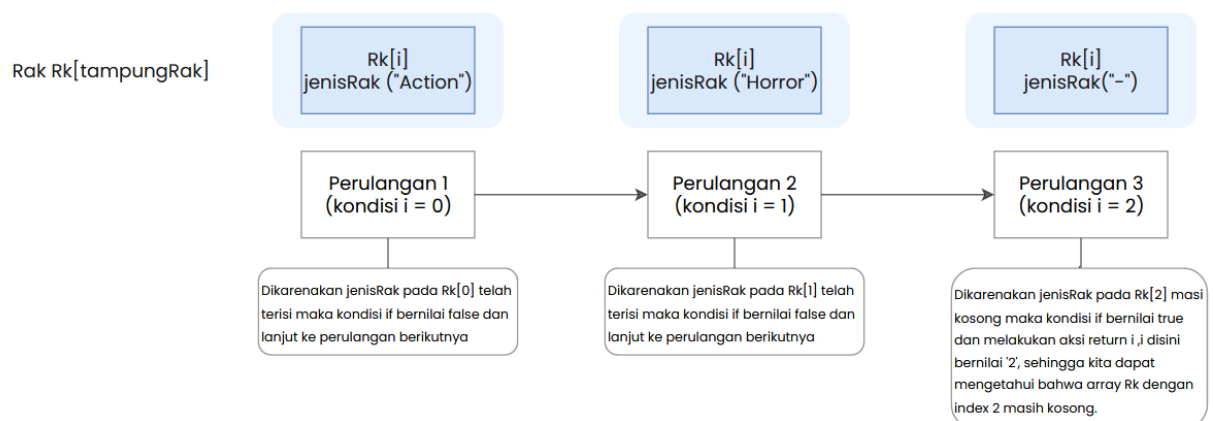
2. CekEmpty

Fungsi cekEmpty yang mereturnkan integer digunakan untuk mengecek index array Rak manakah yang kosong.

```
159 int cekEmpty(Rak Rk[]){
160     int i;
161     for(i = 0; i < tampungRak; i++){
162         if(strcmp(Rk[i].jenisRak,"-")==0){
163             return i;
164         }
165     }
166     return -1;
167 }
```

Dapat dilihat untuk mengunjungi array Rak, dapat menggunakan perulangan, kondisi perulangan tertulis $i < \text{tampungRak}$, yang berarti index value i akan mengunjungi setiap elemen array Rak, lalu terdapat pengkodisian $\text{if}(\text{strcmp}(\text{Rk}[i].\text{jenisRak}, "-") == 0)$, hal ini memiliki arti yaitu jika elemen yang ditunjuk oleh value i memiliki jenisRak yang masih "-" atau 'kosong', maka fungsi akan langsung mereturnkan i , atau dengan kata lain mereturnkan index elemen array Rak yang masih kosong. Jika fungsi mereturnkan -1 maka setiap elemen data pada array Rak sudah terisi atau dengan kata lain array Rak sudah full.

Kita permisalkan index 0 dan index 1 pada arrayRak bagian atribut jenisRak telah terisi, sedangkan pada index 2, jenisRak masih kosong



Kesimpulanya, dengan kita membuat fungsi cekEmpty, kita dapat mengetahui index mana yang masih kosong, sehingga ketika kita nanti akan melakukan isi data pada array Rak, kita tinggal mengisi data pada index hasil return *i* dari fungsi cekEmpty.

3. CekIsi

```
int cekIsi(Rak Rk[]){
    int i;
    for(i = 0; i < tampungRak; i++){
        if(strcmp(Rk[i].jenisRak,"-")!=0){
            return i;
        }
    }
    return -1;
}
```

Cara kerja dari fungsi CekIsi sama seperti fungsi cekEmpty, yang berbeda hanyalah pada pengkodisianya, dimana jika pada cekEmpty pengkodisian akan bernilai *true* jika atribut jenisRak yang ditunjuk masih kosong atau "-". Sedangkan pada cekIsi, pengkodisian akan bernilai *true* jika atribut jenisRak yang ditunjuk telah terisi. Dengan adanya fungsi cekIsi, kita dapat mengetahui apakah data pada array Rak masih kosong semua atau sudah terdapat data yang terisi, karena jika fungsi cekIsi mereturnkan nilai -1, itu berarti semua data pada arrayRak masih kosong.

4. Isi data array

Berikut langkah – langkah untuk melakukan set data / isi data pada array Rak

a. Melakukan inputan user di main program

```
printf("\n[Isi Data Rak]");
printf("\nNo Rak      : ");scanf("%d",&noRak);
printf("Jenis Rak : ");fflush(stdin);gets(jenisRak);

printf("\n\t[Isi Data Buku]");

for(i = 0; i < tampungBuku; i++){
    printf("\n\tData Buku Ke - %d", (i+1));
    printf("\n\tJudul Buku : ");fflush(stdin);gets(judulBuku);
    printf("\tKode Buku   : ");fflush(stdin);gets(kodeBuku);
    temp[i] = isiDataBuku(judulBuku, kodeBuku);
}

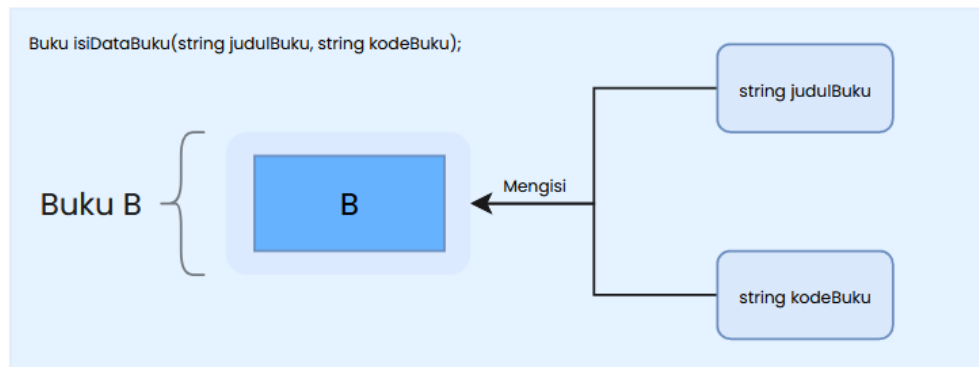
Rk[cekEmpty(Rk)] = isiDataRak(jenisRak, noRak, temp);

printf("\n\tBerhasil Tambah Data");
```


Setiap data yang kita inputkan akan disimpan terlebih dahulu pada masing masing variabel yang berbeda (perhatikan garis biru pada gambar). Dikarenakan dalam 1 Rak terdapat 2 buku, maka kita menggunakan perulangan untuk mengisi data pada array Buku.

b. Fungsi isiDataBuku

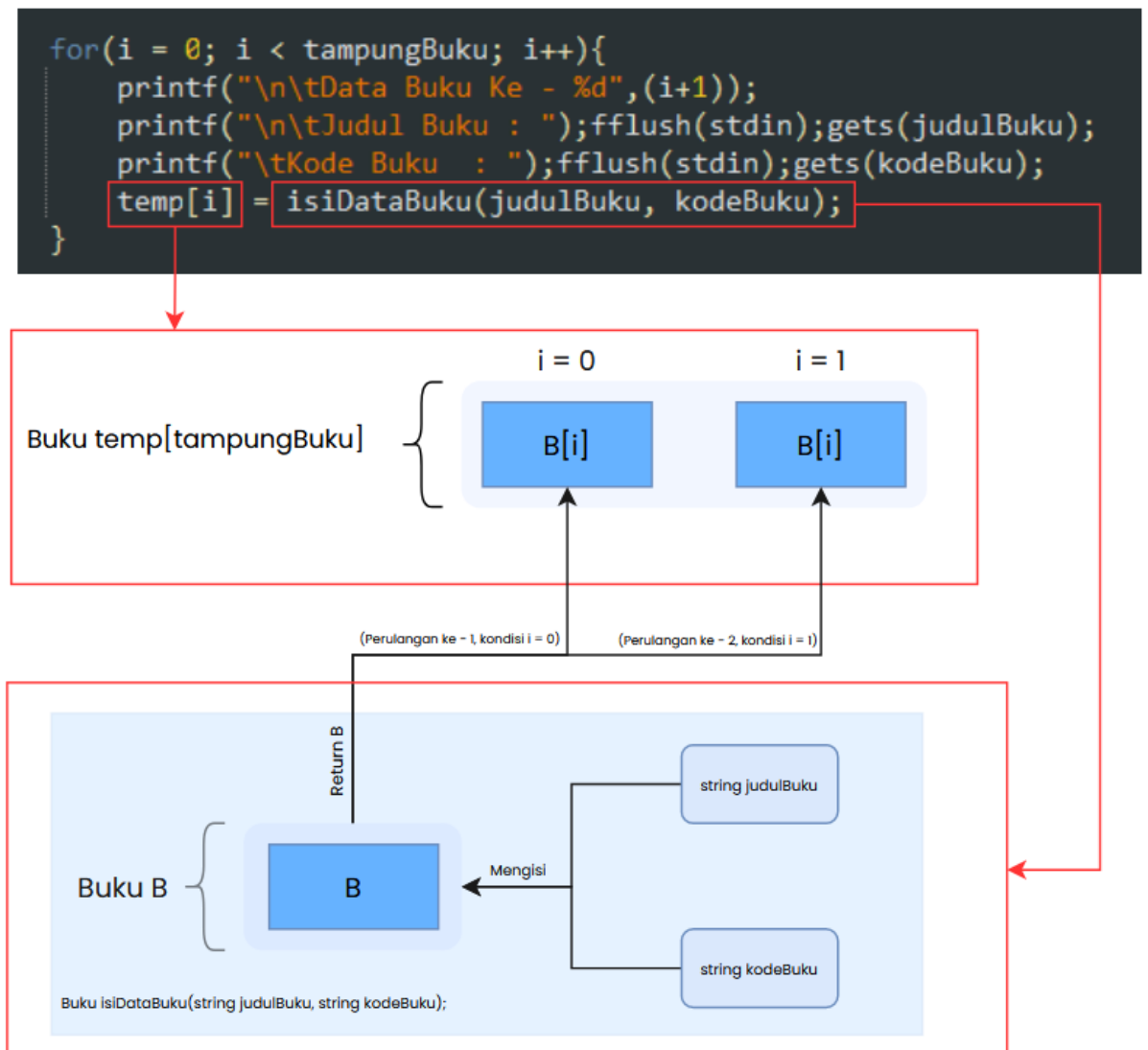
```
139 Buku isiDataBuku(string judulBuku, string kodeBuku){  
140     Buku B;  
141     strcpy(B.judulBuku, judulBuku);  
142     strcpy(B.kodeBuku, kodeBuku);  
143     return B;  
144 }
```



`isiDataBuku` merupakan fungsi yang mereturnkan tipe data `Buku`, dimana didalam fungsi tersebut akan dibuat sebuah variabel bernama `B` dengan tipe data `Buku`. Selanjutnya variabel `B` tersebut akan diisi dengan data yang berasal dari variabel global yaitu `judulBuku` dan `kodeBuku`. Terakhir fungsi akan mereturnkan `B`.

c. Mengisi array temp[i]

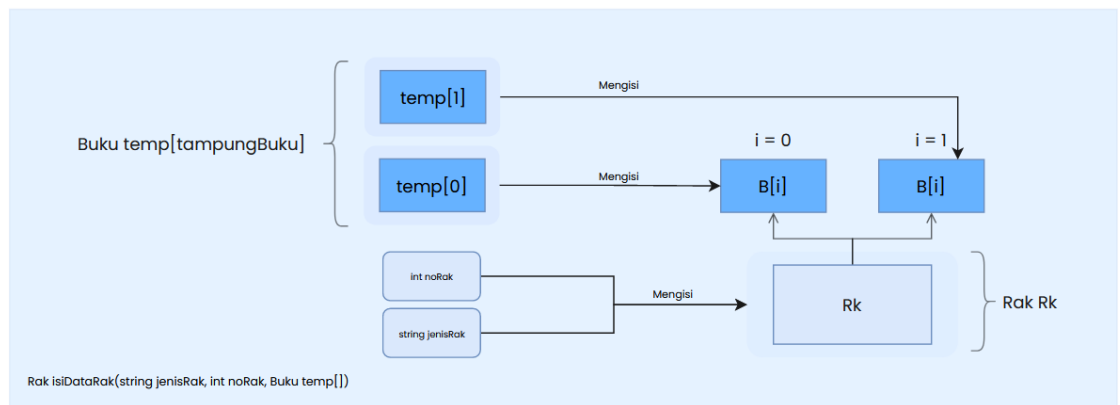
Selanjutnya hasil return B dari fungsi isiDataBuku akan dimasukan datanya pada array Buku temp[i]



d. Fungsi isiDataRak

Kita telah mengisi noRak, jenisRak dan semua data Buku. Selanjutnya kita memasukan semua variabel tersebut kedalam fungsi isiDataRak.

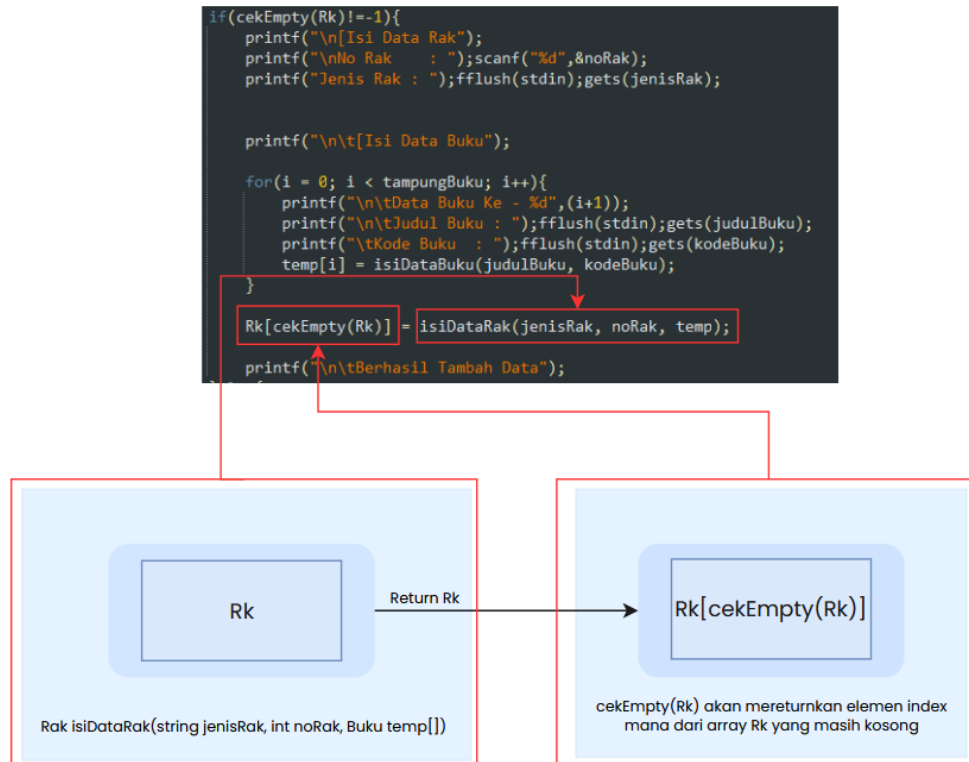
```
Rak isiDataRak(string jenisRak, int noRak, Buku temp[]){  
    Rak Rk;  
    int i;  
  
    Rk.noRak = noRak;  
    strcpy(Rk.jenisRak, jenisRak);  
    for(i = 0 ; i < tampungBuku; i++){  
        Rk.B[i] = temp[i];  
    }  
  
    return Rk;  
}
```



Cara kerja dari fungsi ini sama seperti `isiDataBuku`, dimana fungsi ini akan mereturnkan tipeData `Rak`. Didalam fungsi tersebut akan dibuat sebuah variabel bernama `Rk` dengan tipe data `Rak`. Selanjutnya variabel `Rk` tersebut akan diisi dengan data yang berasal dari variabel global yaitu `jenisRak`, `noRak` dan `temp[]`. Terakhir fungsi akan mereturnkan `Rk`.

e. Mengisi array `Rk[cekEmpty(Rk)]`

Selanjutnya hasil return `Rk` dari fungsi `isiDataRak` akan dimasukan datanya pada array `Rak Rk[cekEmpty(Rk)]`. Index array `Rk` akan diisi dengan fungsi `cekEmpty(Rk)` sehingga data akan mengisi pada elemen array `Rk` yang kosong.



5. SearchRak

```
int searchRak(Rak Rk[], int noRak){
    int i;
    for(i = 0; i < tampungRak; i++){
        if(Rk[i].noRak == noRak)
            return i;
    }
    return -1;
}
```

Cara kerja dari fungsi `SearchRak` sama seperti fungsi `cekEmpty`, yang berbeda hanyalah pada pengkodisiannya, dimana sesuai namanya, fungsi `searchRak` akan mencari rak berdasarkan variabel global `noRak` yang berasal dari inputan user. Jika pengkodisian bernilai true maka akan mereturnkan `i` (index yang dicari), sedangkan return `-1` jika `noRak` yang dicari tidak ada yang sesuai dengan `noRak` didalam array `Rk`. Salah satu kegunaan dari `searchRak` adalah kita dapat menghapus salah satu data pada elemen array `Rk`, berdasarkan `noRak` yang ingin kita hapus.

6. Delete Data

```
case 3:
    if(cekIsi(Rk)!=-1){
        printf("\nNoRak yang ingin dihapus : ");scanf("%d",&noRak);
        if(searchRak(Rk, noRak)!=-1){
            for(i=0; i < tampungBuku; i++){
                strcpy(temp[i].judulBuku,"-");
                strcpy(temp[i].kodeBuku,"-");
            }
            Rk[searchRak(Rk,noRak)] = isiDataRak("-", 0, temp);

            printf("\nBerhasil Delete Data [!]);
        }else{
            printf("\nData Tidak Ditemukan [!]);
        }

    }else{
        printf("\nSemua Rak Masih Kosong [!]);
    }

    break;
```

Pertama akan dilakukan cekIsi terlebih dahulu, jika return -1 maka semua data pada array Rk masih kosong dan tidak dapat melakukan delete data. Jika sudah ada data yang terisi maka user akan menginputkan noRak yang akan dihapus, lalu akan masuk ke pengkodisian searchRak, jika return -1 maka data yang dicari tidak ada. Selanjutnya konsep hapus data pada array Rk memiliki konsep yang sama dengan isi data. Yang berbeda hanyalah jika isi data berdasarkan inputan user, maka delete data akan diisi dengan “-” untuk tipe data string dan ‘0’ untuk integer. Terakhir, Rk dengan index hasil return searchRak(Rk,noRak) akan dihapus datanya atau diset menjadi nilai default kembali.

7. Tampil data

```
void tampilData(Rak Rk[]){
    int i,j;

    printf("\n[DATA RAK]");
    for(i = 0; i < tampungRak; i++){
        if(strcmp(Rk[i].jenisRak,"-")!=0){
            printf("\nNo Rak      : %d",Rk[i].noRak);
            printf("\nJenis Rak   : %s\n",Rk[i].jenisRak);

            printf("\n\n\t[DATA BUKU]");
            for(j = 0; j < tampungBuku; j++){
                printf("\n\n\tData Buku Ke - %d", (j+1));
                printf("\n\tJudul Buku : %s",Rk[i].B[j].judulBuku);
                printf("\n\tKode Buku  : %s\n\n",Rk[i].B[j].kodeBuku);
            }
        }
    }
}
```

Cara kerja dari tampil data sama seperti init data, dimana setiap elemen dikunjungi untuk ditampilkan datanya. Namun setiap index i yang mengunjungi elemen array Rk, akan ada pengkodisian apakah index array Rk yang ditunjuk sudah terisi atau belum, jika belum maka data tidak akan ditampilkan.

Setelah praktikkan memahami modul Array Of Record, praktikkan diharapkan dapat memplajari guided, supaya tidak mengalami kesulitan saat mengerjakan unguided nanti. Jangan lupa perhatikan ketentuan pengerjaan dan format pengumpulan, jika ada yang ingin ditanyakan silahkan menghubungi asisten. Terima kasih!.