

Nama : Gede Satyamahinsa Prastita Utama

NIM : 1203220054

Kelas : IF-02-03

Latihan 1 Praktikum Algoritma Struktur Data – Array

1. Source Code:

```
1  #include <stdio.h>
2
3  int main() {
4      // Menginisialisasi array 2 dimensi bertipe data integer yang terdiri dari 3 baris dan 2 kolom.
5      int x[3][2] = {{3, 6}, {9, 8}, {15, 10}};
6
7      // Melakukan perulangan bersarang untuk menampilkan elemen pada indeks ke-i dan ke-j.
8      // Perulangan pertama berfungsi untuk melakukan pemanggilan terhadap baris.
9      for(int i = 0; i < 3; i++) {
10         // Perulangan kedua berfungsi untuk melakukan pemanggilan terhadap kolom.
11         for(int j = 0; j < 2; j++) {
12             // Menampilkan elemen di dalam array dengan i sebagai baris dan j sebagai kolom.
13             printf("Nilai array x[%d][%d] adalah : %d\n", i, j, x[i][j]);
14         }
15     }
16
17     return 0;
18 }
```

Output:

```
Nilai array x[0][0] adalah : 3
Nilai array x[0][1] adalah : 6
Nilai array x[1][0] adalah : 9
Nilai array x[1][1] adalah : 8
Nilai array x[2][0] adalah : 15
Nilai array x[2][1] adalah : 10
```

2. Source Code:

```
1  #include <stdio.h>
2
3  // sorting_array() ⇒ digunakan untuk melakukan pengurutan elemen array dari yang terkecil ke terbesar dengan menggunakan metode bubble sort.
4  void sorting_array(int *arr, int size) {
5      // Perulangan pertama digunakan untuk mengakses seluruh elemen secara satu per satu.
6      for (int i = 0; i < size - 1; i++) {
7          // Membuat variabel temp yang digunakan sebagai variabel sementara.
8          int temp;
9          // Perulangan kedua digunakan untuk membandingkan satu per satu elemen dalam menentukan urutan dengan benar.
10         for(int j = 0; j < size - i - 1; j++) {
11             // Melakukan pengecekan apakah elemen saat ini lebih besar daripada elemen setelahnya.
12             if(arr[j] > arr[j + 1]) {
13                 // Jika kondisi di atas terpenuhi, maka proses pertukaran elemen dilakukan dengan menggunakan bantuan dari variabel temp.
14                 temp = arr[j];
15                 arr[j] = arr[j + 1];
16                 arr[j + 1] = temp;
17             }
18         }
19     }
20 }
21
22 // print_array() ⇒ digunakan untuk menampilkan seluruh elemen array.
23 void print_array(int *arr, int size) {
24     // Perulangan ini digunakan untuk mengakses setiap elemen berdasarkan indeks.
25     for (int i = 0; i < size; i++) {
26         // Menampilkan elemen di dalam array.
27         printf("%d, ", arr[i]);
28     }
29 }
30
31 int main() {
32     // Menginisialisasi array 1 dimensi bertipe data integer dengan elemen sebanyak 5 buah bilangan bulat.
33     int x[5] = {15, 21, 4, 6, 2};
34     // Membuat variabel size untuk mengetahui panjang dari array dengan menggunakan sizeof dari array dan sizeof dari integer.
35     int size = sizeof(x)/sizeof(int);
36
37     // Menampilkan seluruh elemen array yang belum terurut.
38     printf("Array awal :\n");
39     // Memanggil fungsi print_array dengan parameter x merupakan array dan size merupakan panjang array.
40     print_array(x, size);
41
42     // Melakukan pengurutan elemen di dalam array menggunakan sorting_array() dengan parameter x merupakan array dan size merupakan panjang array.
43     sorting_array(x, size);
44
45     // Menampilkan seluruh array yang sudah terurut.
46     printf("\nArray yang telah diurutkan :\n");
47     // Memanggil fungsi print_array dengan parameter x merupakan array dan size merupakan panjang array.
48     print_array(x, size);
49
50     return 0;
51 }
```

Output:

```
Array awal :
15, 21, 4, 6, 2,
Array yang telah diurutkan :
2, 4, 6, 15, 21,
```

3. Source Code:

```
1  #include <stdio.h>
2
3  // find_index() ⇒ digunakan untuk melakukan pengecekan terhadap semua elemen yang ada di dalam array untuk mencari indeks dari bilangan yang diinputkan oleh user
4  void find_index(int arr[3][2], int input) {
5      // Menginisialisasi variabel found yang digunakan sebagai tanda apakah bilangan sudah ditemukan atau belum.
6      // Nilai awal dari found adalah 0 sebagai tanda bahwa bilangan belum ditemukan.
7      int found = 0;
8      // Perulangan pertama berfungsi untuk melakukan pemanggilan terhadap baris.
9      for (int i = 0; i < 3; i++) {
10         // Perulangan kedua berfungsi untuk melakukan pemanggilan terhadap kolom.
11         for(int j = 0; j < 2; j++) {
12             // Melakukan pengecekan apakah array pada baris ke-i dan kolom ke-j sama dengan bilangan yang dicari oleh user.
13             if(arr[i][j] == input) {
14                 // Jika elemen pada baris ke-i dan kolom ke-j sama dengan bilangan yang dicari oleh user, maka tampilkan indeks dari bilangan tersebut.
15                 printf("Elemen %d berada di indeks[%d][%d]\n", input, i, j);
16                 // Mengubah variabel found menjadi 1 sebagai tanda bahwa bilangan yang dicari oleh user telah ditemukan.
17                 found = 1;
18                 // Lakukan break untuk menghentikan perulangan saat ini.
19                 break;
20             }
21         }
22     }
23
24     // Jika nilai dari found tetap 0, maka bilangan yang dicari oleh user tidak ditemukan di dalam array tersebut.
25     if(found == 0) {
26         printf("Tidak ada elemen tersebut di dalam array\n");
27     }
28 }
29
30 int main() {
31     // Menginisialisasi array 2 dimensi bertipe data integer yang terdiri dari 3 baris dan 2 kolom.
32     int arr[3][2] = {{3, 6}, {9, 8}, {15, 10}};
33
34     // Menampilkan elemen yang ada di dalam array.
35     printf("Array: ");
36     // Perulangan pertama berfungsi untuk melakukan pemanggilan terhadap baris.
37     for(int i = 0; i < 3; i++) {
38         // Perulangan kedua berfungsi untuk melakukan pemanggilan terhadap kolom.
39         for(int j = 0; j < 2; j++) {
40             // Menampilkan elemen di dalam array dengan i sebagai baris dan j sebagai kolom.
41             printf("%d, ", arr[i][j]);
42         }
43     }
44
45     // Meminta user untuk memasukkan sebuah bilangan yang ingin dicari indeksny di dalam array.
46     int user_input;
47     printf("\nMasukkan elemen yang ingin dicari indeksny: ");
48     scanf("%d", &user_input);
49
50     // Melakukan pengecekan terhadap bilangan dari user.
51     find_index(arr, user_input);
52
53     return 0;
54 }
```

Output:

```
Array: 3, 6, 9, 8, 15, 10,
Masukkan elemen yang ingin dicari indeksny: 8
Elemen 8 berada di indeks[1][1]
```

```
Array: 3, 6, 9, 8, 15, 10,
Masukkan elemen yang ingin dicari indeksny: 1
Tidak ada elemen tersebut di dalam array
```

4. Source Code:

```
1  #include <stdio.h>
2
3  // hitung_rata_rata() ⇒ digunakan untuk menghitung rata-rata dari jumlah seluruh elemen array.
4  float hitung_rata_rata(int *arr) {
5      // Menginisialisasi variabel total dengan nilai awal, yaitu 0.
6      float total = 0;
7      // Perulangan ini digunakan untuk mengakses elemen array secara satu per satu, lalu dijumlahkan dengan variabel total.
8      for (int i = 0; i < 70; i++) {
9          total += arr[i];
10     }
11     // Mengembalikan nilai rata-rata dengan tipe data float.
12     return total / 70;
13 }
14
15 // print_array() ⇒ digunakan untuk menampilkan seluruh elemen array.
16 void print_array(int *arr) {
17     // Perulangan ini digunakan untuk mengakses setiap elemen berdasarkan indeks.
18     for (int i = 0; i < 70; i++) {
19         printf("%d, ", arr[i]);
20     }
21 }
22
23 int main() {
24     // Deklarasi sebuah array dengan ukuran sebanyak 70 elemen.
25     int arr[70];
26
27     // Memasukkan elemen dengan angka 1 - 70 ke dalam array.
28     for(int i = 0; i < 70; i++) {
29         // Memasukkan i + 1 ke dalam setiap elemen karena indeks dimulai dari 0.
30         arr[i] = i + 1;
31     }
32
33     // Menampilkan semua elemen yang ada di dalam array.
34     printf("Array: ");
35     print_array(arr);
36
37     // Melakukan perhitungan rata-rata dari jumlah seluruh elemen array.
38     printf("\nRata - rata dari array tersebut adalah: %.2f\n", hitung_rata_rata(arr));
39
40     return 0;
41 }
```

Output:

```
Array: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 6
8, 69, 70,
Rata - rata dari array tersebut adalah: 35.50
```

5. Source Code:

```
1 #include <stdio.h>
2
3 int main() {
4     // Menginisialisasi array lantai dengan elemen sebanyak lantai gedung.
5     int lantai[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
6     // Menginisialisasi variabel banyak_lantai untuk menampung banyak elemen yang ada di dalam array lantai.
7     int banyak_lantai = sizeof(lantai)/sizeof(int);
8     // Deklarasi variabel pilihan untuk menampung pilihan lantai dari user.
9     int pilihan;
10    printf("Pilih nomor lantai: ");
11    scanf("%d", &pilihan);
12
13    // Menginisialisasi variabel before dan start sebagai nilai awal banyak elemen yang berada di sebelum dan sesudah pilihan dari user.
14    int before = 2, after = 2;
15
16    // Melakukan perulangan sebanyak ukuran dari array untuk mengecek pilihan dari user.
17    for(int i = 0; i < banyak_lantai; i++) {
18        // Jika pilihan dari user terdapat di dalam array, maka lakukan pengecekan kondisi untuk menyesuaikan urutan lantai sebanyak 5 lantai.
19        if(lantai[i] == pilihan) {
20            if(i + after >= banyak_lantai) {
21                // Jika banyak lantai yang ada setelah lantai pilihan user adalah kurang dari 2, maka lakukan operasi berikut agar dapat menyesuaikan dengan kondisi yang terjadi.
22                // Untuk mendapatkan nilai before dan after secara dinamis yang bergantung pada banyak lantai, maka dapat menggunakan rumus di bawah ini.
23                // Misal, angka pilihan adalah 9 (i = 8), maka banyak angka yang tersedia sebelum angka pilihan user bertambah menjadi 3. Sedangkan, banyak angka yang tersedia setelah angka pilihan user berkurang menjadi 1.
24                before += i + after - (banyak_lantai - 1);
25                after -= i + after - (banyak_lantai - 1);
26            } else if (i - before < 0) {
27                // Jika banyak lantai yang ada sebelum lantai pilihan user adalah kurang dari 2, maka lakukan operasi berikut agar dapat menyesuaikan dengan kondisi yang terjadi.
28                // Untuk mendapatkan nilai before dan after secara dinamis tergantung dari banyak lantai, maka dapat menggunakan rumus di bawah ini.
29                // Misal, angka pilihan adalah 1 (i = 0), maka banyak angka yang tersedia sebelum angka pilihan user menjadi 0. Sedangkan, banyak angka yang tersedia setelah angka pilihan user bertambah menjadi 4.
30                before = i;
31                after += i - after;
32            }
33            // Menampilkan urutan 5 lantai sesuai dengan kondisi.
34            for(int j = i - before; j <= i + after; j++) {
35                printf("%d, ", lantai[j]);
36            }
37        }
38    }
39
40    return 0;
41 }
```

Output:

```
Pilih nomor lantai: 9
6, 7, 8, 9, 10,
```

```
Pilih nomor lantai: 1
1, 2, 3, 4, 5,
```

```
Pilih nomor lantai: 5
3, 4, 5, 6, 7,
```