

## Soal Latihan Praktikum Algoritma Struktur Data - Array, Pointer, dan Fungsi

Nama : I Gede Fender Adrea Veda

NIM : 1203230009

Kelas : IF 03-01

### Soal Pertama

#### 1. Source Code

```
1. #include <stdio.h>
2. #include <string.h>
3.
4. void swap(int* a, int* b) {
5.     int temp = *a;
6.     *a = *b;
7.     *b = temp;
8. }
9.
10. void printCard(int number) {
11.     if (number == 1) printf("1 ");
12.     else if (number == 2) printf("2 ");
13.     else if (number == 3) printf("3 ");
14.     else if (number == 4) printf("4 ");
15.     else if (number == 5) printf("5 ");
16.     else if (number == 6) printf("6 ");
17.     else if (number == 7) printf("7 ");
18.     else if (number == 8) printf("8 ");
19.     else if (number == 9) printf("9 ");
20.     else if (number == 10) printf("10 ");
21.     else if (number == 11) printf("J ");
22.     else if (number == 12) printf("Q ");
23.     else if (number == 13) printf("K ");
24.     else printf("%d ", number);
25. }
26.
27. void sortCards(int n, int *arr) {
28.     int swaps = 0;
29.
30.     for (int i = 0; i < n - 1; i++) {
31.         for (int j = 0; j < n - i - 1; j++) {
32.             if (arr[j] > arr[j + 1]) {
33.                 swap(&arr[j], &arr[j + 1]);
34.                 swaps++;
35.
36.                 printf("Langkah %d: Pertukaran antara kartu %d (%c) dan
kartu %d (%c)\n", swaps, j+1, arr[j], j+2, arr[j+1]);
37.
38.                 printf("Kartu setelah pertukaran %d: ", swaps);
```

```
39.         for (int k = 0; k < n; k++) {
40.             printCard(arr[k]);
41.         }
42.         printf("\n");
43.     }
44. }
45. }
46.
47.     printf("Jumlah pertukaran yang dilakukan : %d\n", swaps);
48. }
49.
50. int main() {
51.     int n;
52.     printf("Masukkan jumlah kartu : ");
53.     scanf("%d", &n);
54.
55.     int arr[n];
56.     char temp[3];
57.     printf("Masukkan angka kartu (1-10, J/Q/K) : ");
58.     for (int i = 0; i < n; i++) {
59.         scanf("%s", temp);
60.         if (strcmp(temp, "J") == 0) {
61.             arr[i] = 11;
62.         } else if (strcmp(temp, "Q") == 0) {
63.             arr[i] = 12;
64.         } else if (strcmp(temp, "K") == 0) {
65.             arr[i] = 13;
66.         } else {
67.             sscanf(temp, "%d", &arr[i]);
68.         }
69.     }
70.
71.     sortCards(n, arr);
72.
73.     return 0;
74. }
```

## 2. Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
cd "/Users/igedefenderadreaaveda/codingan pertama alpro/popok/" && gcc berhasil.c -o berhasil && "/Users/igedefenderadreaaveda/codingan pertama alpro/popok/"berhasil
igedefenderadreaaveda@Is-MacBook-Pro popok % cd "/Users/igedefenderadreaaveda/codingan pertama alpro/popok/" && gcc berhasil.c -o berhasil && "/Users/igedefenderadreaaveda/codingan pertama alpro/popok/"berhasil
Masukkan jumlah kartu : 5
Masukkan angka kartu (1-10, J/Q/K) : J 10 K 9 Q
Langkah 1: Pertukaran antara kartu 1 (
) dan kartu 2 (
)
Kartu setelah pertukaran 1: 10 J K 9 Q
Langkah 2: Pertukaran antara kartu 3 (
) dan kartu 4 (
)
Kartu setelah pertukaran 2: 10 J 9 K Q
Langkah 3: Pertukaran antara kartu 4 (
) dan kartu 5 (
)
Kartu setelah pertukaran 3: 10 J 9 Q K
Langkah 4: Pertukaran antara kartu 2 (
) dan kartu 3 (
)
Kartu setelah pertukaran 4: 10 9 J Q K
Langkah 5: Pertukaran antara Kartu 1 (
) dan kartu 2 (
)
Kartu setelah pertukaran 5: 9 10 J Q K
Jumlah pertukaran yang dilakukan : 5
igedefenderadreaaveda@Is-MacBook-Pro popok %
```

## 3. Penjelasan

```
void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Fungsi swap bekerja dengan memindahkan nilai dari a ke variabel lokal temp, lalu memindahkan nilai dari b ke a, dan akhirnya memindahkan nilai dari temp ke b. Dengan cara ini, nilai dari a dan b akan berpindah tempat, sehingga melakukan pemutaran antara kedua variabel.

```
void printCard(int number) {
    if (number == 1) printf("1 ");
    else if (number == 2) printf("2 ");
    else if (number == 3) printf("3 ");
    else if (number == 4) printf("4 ");
    else if (number == 5) printf("5 ");
    else if (number == 6) printf("6 ");
    else if (number == 7) printf("7 ");
    else if (number == 8) printf("8 ");
    else if (number == 9) printf("9 ");
    else if (number == 10) printf("10 ");
    else if (number == 11) printf("J ");
    else if (number == 12) printf("Q ");
    else if (number == 13) printf("K ");
    else printf("%d ", number);
}
```

Fungsi *printCard* di atas adalah sebuah fungsi yang digunakan untuk menampilkan angka dari 1 sampai 13 dengan menggunakan notasi kartu poker. Fungsi *printCard* terdiri dari 13 baris kode, yang masing-masing memiliki pernyataan *else if* untuk setiap angka dari 1 sampai 13.

```
void sortCards(int n, int *arr) {
    int swaps = 0;
```

Artinya : Fungsi *sortCards* di atas adalah sebuah fungsi yang digunakan untuk memutar kartu dari 1 sampai 13 dengan menggunakan notasi kartu poker. Fungsi ini memiliki dua parameter bernama *n* dan *arr*, yang merupakan *int* dan pointer ke array yang akan diubah menjadi urutan kunci dari 1 sampai 13 sesuai notasi kartu poker.

```
for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - i - 1; j++) {
        if (arr[j] > arr[j + 1]) {
            swap(&arr[j], &arr[j + 1]);
            swaps++;
        }
    }
}
```

Algoritma pemutaran (bubblesort) yang digunakan dalam fungsi *sortCards* memiliki dua loop for. Pertama, loop for dari 0 sampai *n*-1 digunakan untuk memutar setiap elemen dari array *arr*. Kedua, loop for dari 0 sampai *n*-i-1 digunakan untuk memutar setiap elemen dari array *arr* kecuali yang terakhir. Pada loop pertama kita memeriksa apakah elemen dalam array lebih kecil dari elemen setelahnya. Jika tidak, pemutaran antara kedua elemen tersebut dilakukan. Fungsi *swap* digunakan untuk melakukan pemutaran antara kedua elemen tersebut. Selanjutnya meningkatkan variabel *swaps* untuk mengukur berapa banyak pemutaran yang telah dilakukan.

```
printf("Langkah %d: Pertukaran antara kartu %d (%c) dan kartu %d (%c)\n",
swaps, j+1, arr[j], j+2, arr[j+1]);
```

Untuk mencetak informasi tentang pertukaran kartu yang terjadi selama proses pengurutan. Indeks dan nilai dari kartu pertama yang pertukaran nilai dengan kartu kedua.

```
printf("Kartu setelah pertukaran %d: ", swaps);
for (int k = 0; k < n; k++) {
    printCard(arr[k]);
}
```

Untuk mencetak nilai kartu setelah dilakukan pertukaran nilai. Melakukan iterasi melalui array kartu (*arr*) untuk mencetak nilai setiap kartu setelah proses pertukaran. *printCard* untuk mencetak nilai kartu yang berada pada indeks ke-*k* dalam array setelah proses pertukaran dilakukan dan konversi nilai numerik kartu menjadi representasi karakter sesuai dengan aturan.

```
printf("Jumlah pertukaran yang dilakukan : %d\n", swaps);
```

Untuk mencetak jumlah total pertukaran yang telah dilakukan selama proses pengurutan kartu menggunakan algoritma bubble sort.

```
int main() {
    int n;
    printf("Masukkan jumlah kartu : ");
    scanf("%d", &n);
```

Fungsi *main* yang digunakan dalam kode di atas terdiri dari beberapa baris kode, yang masing-masing memiliki pernyataan *else if* untuk setiap angka dari 1 sampai 13.

```
int arr[n];
char temp[3];
printf("Masukkan angka kartu (1-10, J/Q/K) : ");
```

```
for (int i = 0; i < n; i++) {
    scanf("%s", temp);
```

variabel `arr` yang merupakan sebuah array dengan tipe data `int` dan panjang `n`. Variabel `temp` merupakan sebuah array dengan tipe data `char` dan panjang 3. Perulangan pertama untuk memasukkan angka kunci dengan menggunakan notasi kartu poker. Perulangan ini akan berjalan `n` kali, `n` merupakan jumlah kunci yang akan dibaca. Perulangan kedua untuk mengurutkan array `arr` yang terdiri dari angka kunci yang telah dibaca. Perulangan ini akan berjalan `n-1` kali, `n` merupakan jumlah kunci yang akan dibaca. Jika terdapat perbedaan antara dua angka kunci yang terletak di posisi terdekat, maka akan dilakukan pertukaran. Setiap pertukaran ditampilkan ke layar, serta jumlah pertukaran yang dilakukan selama proses pengurutan.

```
if (strcmp(temp, "J") == 0) {
    arr[i] = 11;
} else if (strcmp(temp, "Q") == 0) {
    arr[i] = 12;
} else if (strcmp(temp, "K") == 0) {
    arr[i] = 13;
} else {
    sscanf(temp, "%d", &arr[i]);
}
}
```

Program memeriksa apakah nilai yang dimasukkan untuk kartu adalah "J" menggunakan fungsi `strcmp(temp, "J")`. Jika nilai tersebut adalah "J", maka program menginisialisasi nilai array `arr` pada indeks `i` dengan nilai 11. Begitupula jika dimasukkan "Q" dan "K". Jika nilai yang dimasukkan tidak sama dengan "J", "Q", atau "K", maka program akan mendapatkan nilai integer menggunakan `sscanf(temp, "%d", &arr[i])` dari string `temp` dan menyimpannya ke dalam array `arr` pada indeks `i`.

```
sortCards(n, arr);
```

Dengan memanggil `sortCards(n, arr);`, kita memberikan informasi tentang jumlah elemen dan isi array yang akan diurutkan kepada fungsi `sortCards`.

```
return 0;
```

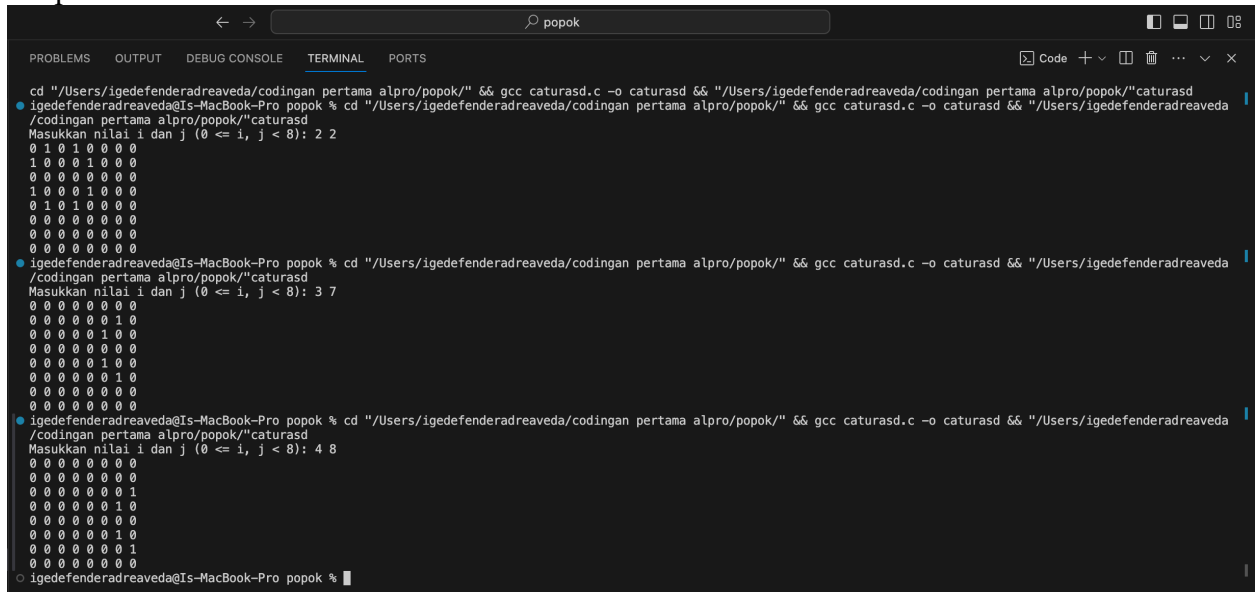
Program selesai dijalankan.

## Soal Kedua

### 1. Source Code

```
1. #include <stdio.h>
2.
3. void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
4.
5.     for (int x = 0; x < size; x++) {
6.         for (int y = 0; y < size; y++) {
7.             *(chessBoard + x*size + y) = 0;
8.         }
9.     }
10.
11.     int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1,
12.         2}, {1, -2}, {1, 2}};
13.
14.     for (int k = 0; k < 8; k++) {
15.         int new_i = i + moves[k][0];
16.         int new_j = j + moves[k][1];
17.
18.         if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
19.             *(chessBoard + new_i*size + new_j) = 1;
20.         }
21.     }
22.
23. int main() {
24.     int i, j;
25.     int chessBoard[8][8];
26.
27.     printf("Masukkan nilai i dan j (0 <= i, j < 8): ");
28.     scanf("%d %d", &i, &j);
29.
30.     koboImaginaryChess(i, j, 8, &chessBoard[0][0]);
31.
32.     for (int x = 0; x < 8; x++) {
33.         for (int y = 0; y < 8; y++) {
34.             printf("%d ", chessBoard[x][y]);
35.         }
36.         printf("\n");
37.     }
38.
39.     return 0;
40. }
```

## 2. Output



```
cd "/Users/igedefenderdreaveda/codingan pertama alpro/popok/" && gcc caturasd.c -o caturasd && "/Users/igedefenderdreaveda/codingan pertama alpro/popok/"caturasd
igedefenderdreaveda@Is-MacBook-Pro popok % cd "/Users/igedefenderdreaveda/codingan pertama alpro/popok/" && gcc caturasd.c -o caturasd && "/Users/igedefenderdreaveda/codingan pertama alpro/popok/"caturasd
Masukkan nilai i dan j (0 <= i, j < 8): 2 2
0 1 0 1 0 0 0 0
1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0
0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
igedefenderdreaveda@Is-MacBook-Pro popok % cd "/Users/igedefenderdreaveda/codingan pertama alpro/popok/" && gcc caturasd.c -o caturasd && "/Users/igedefenderdreaveda/codingan pertama alpro/popok/"caturasd
Masukkan nilai i dan j (0 <= i, j < 8): 3 7
0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
igedefenderdreaveda@Is-MacBook-Pro popok % cd "/Users/igedefenderdreaveda/codingan pertama alpro/popok/" && gcc caturasd.c -o caturasd && "/Users/igedefenderdreaveda/codingan pertama alpro/popok/"caturasd
Masukkan nilai i dan j (0 <= i, j < 8): 4 8
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0
igedefenderdreaveda@Is-MacBook-Pro popok %
```

## 3. Penjelasan

```
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
```

Fungsi `koboImaginaryChess` menerima koordinat posisi awal bidak, ukuran papan catur, dan array pointer yang merepresentasikan papan catur bidak. `i` dan `j` adalah koordinat posisi awal bidak catur. `Size` adalah ukuran papan catur, berisi nilai 8 untuk papan catur 8x8. `*chessBoard` adalah parameter pointer yang menunjuk ke array 1 dimensi yang merepresentasikan papan catur bidak.

```
for (int x = 0; x < size; x++) {
    for (int y = 0; y < size; y++) {
        *(chessBoard + x*size + y) = 0;
    }
}
```

for pertama merupakan perulangan yang menjalankan kode di dalamnya dalam rentang nilai 0 hingga `size-1` untuk variabel `x`. for kedua dalam perulangan `x`, terdapat perulangan `y` yang akan menjalankan kode di dalamnya, nilai 0 hingga `size-1` untuk variabel `y`.

```
int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};
```

`int moves[8][2]` mendeklarasikan array `moves` berukuran 8x2. Array ini menyimpan 8 pasang nilai, di mana setiap pasangan terdiri dari dua nilai bulat. `{{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}}` memberi nilai awal pada array `moves`. Setiap pasangan nilai dalam array ini merepresentasikan langkah-langkah yang mungkin dilakukan oleh suatu bidak pada papan catur.

```
for (int k = 0; k < 8; k++) {
    int new_i = i + moves[k][0];
    int new_j = j + moves[k][1];
```

Perulangan ini akan menjalankan kode di dalamnya sebanyak 8 kali. `int new_i` pada setiap iterasi, kedua variabel `new_i` dan `new_j` dihitung berdasarkan nilai `i` dan `j` posisi awal bidak ditambah dengan pergeseran dari daftar langkah-langkah *moves*.

```
if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {  
    *(chessBoard + new_i*size + new_j) = 1;  
}
```

if memeriksa apakah posisi baru yang dihasilkan berada di dalam batas papan catur yang valid, yaitu harus `>= 0` dan `< size` (ukuran papan catur). Jika posisi baru bidak berada di dalam papan catur, sel tersebut diidentifikasi dalam array `chessBoard` dengan menggunakan aritmetika pointer dan diberikan nilai 1, yang menunjukkan bidak sudah melakukan gerakan ke posisi tersebut.

```
int main() {  
    int i, j;  
    int chessBoard[8][8];
```

Mendeklarasikan array dua dimensi *chessBoard* berukuran 8x8, yang akan merepresentasikan papan catur. Setiap elemen dalam array ini akan merepresentasikan status atau keadaan dari setiap sel papan catur.

```
printf("Masukkan nilai i dan j (0 <= i, j < 8): ");  
scanf("%d %d", &i, &j);
```

Mencetak pesan yang meminta pengguna untuk memasukkan nilai `i` dan `j` dengan batasan bahwa nilai harus berada dalam rentang 0 hingga 7. Format string `"%d %d"` menunjukkan kedua input yang diharapkan adalah bilangan bulat, hasil input akan disimpan ke dalam variabel `i` dan `j` menggunakan operator `&`.

```
koboImaginaryChess(i, j, 8, &chessBoard[0][0]);
```

Pemanggilan fungsi `koboImaginaryChess` sesuai inputan yang dimasukkan. Tujuannya melakukan operasi atau transformasi tertentu pada papan catur yang direpresentasikan oleh array `chessboard`.

```
for (int x = 0; x < 8; x++) {  
    for (int y = 0; y < 8; y++) {  
        printf("%d ", chessBoard[x][y]);  
    }  
    printf("\n");  
}
```

Perulangan pertama (outer loop) ini akan mengiterasi baris dari `chessBoard`, dimulai dari baris pertama (indeks 0) hingga baris terakhir (indeks 7) dan perulangan kedua mengiterasi kolom. `Printf` pertama mencetak nilai yang terdapat dalam sel tertentu dari array `chessBoard`. Setiap elemen pada array `chessBoard` akan dicetak ke dalam konsol dengan format `%d` yang menampilkan nilai integer.

```
return 0;
```

Program selesai dijalankan.



Letakkan tabel di bawah ini pada halaman pertama laporan untuk mempermudah dalam proses pengecekan!

Komponen Penilaian	Ya	Tidak
Soal 1 sesuai dengan output yang diinginkan		
Soal 2 sesuai dengan output yang diinginkan		
Bonus soal 1 dikerjakan		