

Nama : Gede Satyamahinsa Prastita Utama

NIM : 1203220054

Kelas : IF-02-03

## TUGAS REKURSIF

### 1. Source Code

```
1  #include <stdio.h>
2  #include <string.h>
3
4  // cetakSubseken() ⇒ digunakan untuk menampilkan subseken yang ada di dalam str.
5  void cetakSubseken(char str[], int index, char subseken[]) {
6      // Basis: Jika mencapai index akhir pada str, maka cetak subseken yang telah terbentuk.
7      if(str[index] == '\0') {
8          printf("%s\n", subseken);
9          return;
10     }
11
12     // Rekurens: Memanggil fungsi cetakSubseken() dengan index selanjutnya dan subseken di awal.
13     cetakSubseken(str, index + 1, subseken);
14     // Masukkan str pada nilai index ke dalam subseken dengan index setelah index akhir dari subseken.
15     subseken[strlen(subseken)] = str[index];
16     // Rekurens: Memanggil fungsi cetakSubseken() dengan index selanjutnya dan subseken yang telah diupdate.
17     cetakSubseken(str, index + 1, subseken);
18     // Mengembalikan subseken pada index terakhir dari subseken menjadi char kosong.
19     subseken[strlen(subseken) - 1] = '\0';
20 }
21
22 int main() {
23     // Deklarasi variabel str bertipe data char untuk menyimpan input dari user.
24     // Inisialisasi variabel subseken bertipe data char dengan nilai awal, yaitu "".
25     char str[100], subseken[100] = "";
26     // Meminta user untuk memasukkan input string yang ingin dicari subsekuennya.
27     scanf("%s", &str);
28
29     // Memanggil fungsi cetakSubseken() untuk menampilkan semua subseken yang ada di dalam str.
30     cetakSubseken(str, 0, subseken);
31 }
32
```

### Output

```
abc
"" "c" "b" "bc" "a" "ac" "ab" "abc"
```

```
abcd
"" "d" "c" "cd" "b" "bd" "bc" "bcd" "a" "ad" "ac" "acd" "ab" "abd" "abc" "abcd"
```

## 2. Source Code

```
1  #include <stdio.h>
2  #include <string.h>
3
4  // palindrom() ⇒ digunakan untuk mengecek apakah sebuah kata yang sama
5  // jika dibaca dari depan atau belakang.
6  int palindrom(char str[], int start, int end) {
7      // Basis: Jika terdapat karakter pada index start dan index end yang tidak sama,
8      // maka dapat dipastikan bahwa kata tersebut bukan palindrom.
9      if(str[start] != str[end]) {
10         return 0;
11     }
12     // Rekurens: Jika index start dan index end sudah sama atau bertemu,
13     // maka hal tersebut menandakan bahwa seluruh karakter sama dari depan maupun belakang.
14     } else if(start == end) {
15         return 1;
16     }
17     // Rekurens: Memanggil fungsi palindrom() dengan nilai parameter start bertambah 1 dan end berkurang 1.
18     // Hal ini digunakan untuk melakukan pengecekan karakter
19     // dari depan maupun dari belakang secara bersamaan.
20     } else {
21         return palindrom(str, start + 1, end - 1);
22     }
23 }
24
25 int main() {
26     // Deklarasi variabel str bertipe data char.
27     char str[100];
28     // Meminta user untuk memasukkan input string yang ingin dicek.
29     scanf("%s", &str);
30
31     // Memanggil fungsi palindrom()
32     // Jika nilai kembalian = 1, maka string dari user merupakan palindrom.
33     if (palindrom(str, 0, strlen(str) - 1) == 1) {
34         printf("%s merupakan palindrom.", str);
35     }
36     // Jika nilai kembalian = 0, maka string dari user bukan merupakan palindrom.
37     } else {
38         printf("%s bukan merupakan palindrom.", str);
39     }
40     return 0;
41 }
```

### Output

```
katak
katak merupakan palindrom.
```

```
satya
satya bukan merupakan palindrom.
```

### 3. Source Code

```
1  #include <stdio.h>
2
3  // nilaiTerbesar() ⇒ digunakan untuk mencari nilai terbesar di dalam array.
4  int nilaiTerbesar(int arr[], int index, int ukuran, int max) {
5      // Basis: Jika nilai index sama dengan ukuran dari array (mencapai nilai terakhir di array),
6      // maka kembalikan nilai max yang sudah didapatkan.
7      if(index == ukuran) {
8          return max;
9      }
10     // Jika belum mencapai nilai terakhir di array,
11     // maka lakukan pengecekan nilai max terhadap setiap elemen di dalam array.
12     } else {
13         // Rekurens: Jika nilai elemen di dalam array lebih besar dari max,
14         // maka ubah nilai max menjadi nilai elemen tersebut.
15         if(arr[index] > max) {
16             max = arr[index];
17         }
18         // Kembalikan fungsi nilaiTerbesar() sebagai rekursif dengan menambahkan 1 pada nilai index
19         // dan nilai max terbaru.
20         return nilaiTerbesar(arr, index + 1, ukuran, max);
21     }
22 }
23
24 int main() {
25     // Meminta user untuk memasukkan input ukuran dari array.
26     int ukuran;
27     printf("Ukuran Array: ");
28     scanf("%d", &ukuran);
29
30     // Meminta user untuk memasukkan input setiap elemen di dalam array.
31     int num_arr[ukuran];
32     printf("Masukkan bilangan sebanyak %d: ", ukuran);
33     for(int i = 0; i < ukuran; i++) {
34         scanf("%d", &num_arr[i]);
35     }
36
37     // Menampilkan hasil pencarian nilai terbesar di dalam array.
38     printf("%d", nilaiTerbesar(num_arr, 0, ukuran, num_arr[0]));
39     return 0;
40 }
```

### Output

```
Ukuran Array: 5
Masukkan bilangan sebanyak 5: 45 21 39 5 60
60
```

#### 4. Source Code

```
1  #include <stdio.h>
2
3  // hitungDigit() ⇒ digunakan untuk menghitung jumlah digit dari suatu bilangan.
4  int hitungDigit(int input) {
5      // Basis: Jika bilangan kurang dari 10, maka kembalikan 1 (artinya terdapat 1 digit).
6      if(input < 10) {
7          return 1;
8      }
9      // Rekurens: Jika bilangan lebih dari 10, maka kembalikan fungsi hitungDigit() sebagai rekursif
10     // dengan membagi bilangan dengan 10 untuk menghapus digit terakhir dan tambahkan 1
11     // karena digit terakhir telah terhitung.
12     } else {
13         return hitungDigit(input / 10) + 1;
14     }
15 }
16
17 int main() {
18     // Meminta user untuk memasukkan input bilangan bulat yang ingin dihitung jumlah digitnya.
19     int input;
20     scanf("%d", &input);
21
22     // Menampilkan hasil perhitungan jumlah digit dari fungsi hitungDigit().
23     printf("%d", hitungDigit(input));
24     return 0;
25 }
```

#### Output

```
1234
4
```

```
1902134
7
```