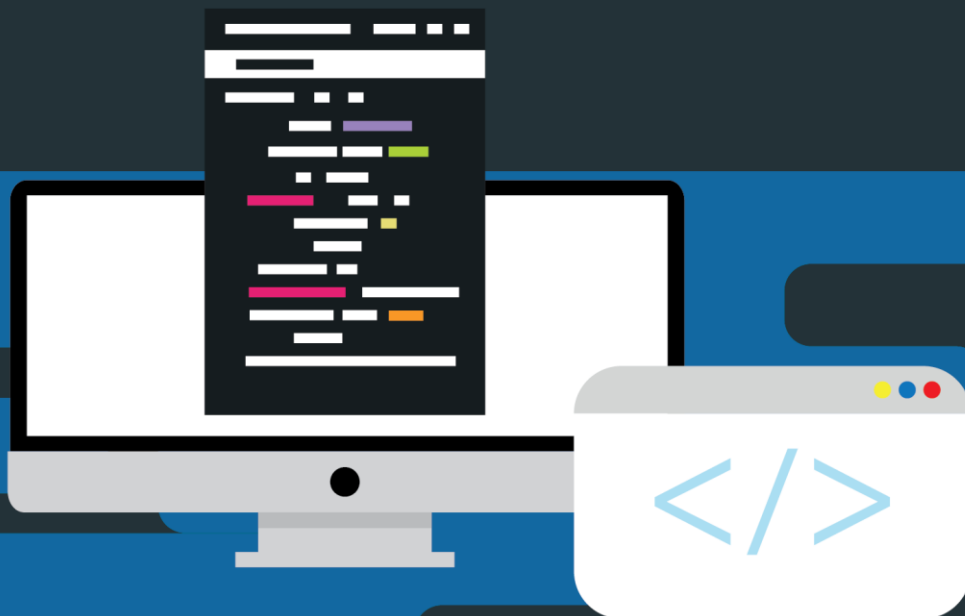


MODUL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN I

ARRAY OF STRING



TIM ASISTEN PEMROGRAMAN
ANGKATAN 11
ILMU KOMPUTER FPMIPA UPI

APA ITU ARRAY OF STRING?

Singkatnya, **array of string** merupakan **kumpulan dari beberapa array of char**, yang bisa diartikan kalau **array of char** adalah sebuah **kata**, dan **array of string** merupakan **kumpulan kata** atau bisa disebut juga **kalimat**.

CARA MENGGUNAKAN ARRAY OF STRING

Sebenarnya, ada banyak cara untuk menggunakan array of string ini, namun disini kita akan mempelajari dua cara saja, yaitu :

- Menggunakan metode larik (array)
- Menggunakan metode tipe data terstruktur

MENGGUNAKAN LARIK (ARRAY)

```
char nama_variabel[jumlah][panjang_karakter];
```

CONTOH PROGRAM MENGGUNAKAN LARIK

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      int n, i; //deklarasi variabel
7
8      scanf("%d", &n); //input banyak
9
10     char input[n][100]; //deklarasi
11
12     /*input nilai array of string*/
13     for (i = 0; i < n; ++i)
14     {
15         scanf("%s", &input[i]);
16     }
17
18     /*tampilkan semua string*/
19     printf("=====\n");
20     for (i = 0; i < n; ++i)
21     {
22         printf("%s\n", input[i]);
23     }
24
25     return 0;
26 }
```

METODE TIPE DATA BUNGKUSAN

```
1  #include <stdio.h>
2  #include <string.h>
3
4  //deklarasi bungkusannya
5  typedef struct{
6      char variabel_string[panjang_karakter]
7  }nama_bungkusannya;
8
9  int main()
10 {
11     //deklarasi variabel bungkusannya
12     nama_bungkusannya nama_variabel_bungkusannya[banyak];
13
14     return 0;
15 }
```

CONTOH PROGRAM DENGAN TIPE DATA TERSTRUKTUR

```
1  #include <stdio.h>
2  #include <string.h>
3
4  //deklarasi bungkusannya
5  typedef struct{
6      char kata[100];
7  }kalimat;
8
9  int main()
10 {
11     //deklarasi
12     int n, i;
13     n = 0;
14     //deklarasi variabel bungkusannya
15     kalimat input[50];
16     //input kata untuk memancing penggunaan while
17     scanf("%s", &input[n].kata);
18     //lakukan selama bukan kata selesai
19     while(strcmp(input[n].kata, "selesai") != 0) {
20         //jumlahkan
21         n++;
22         //masukkan kata yang baru
23         scanf("%s", &input[n].kata);
24     }
25
26     /*tampilkan semua string jika n bernilai lebih dari 0*/
27     if (n > 0)
28     {
29         printf("=====\n");
30         for (i = 0; i < n; ++i)
31         {
32             printf("%s\n", input[i].kata);
33         }
34     }
35
36     return 0;
37 }
```

```

1  #include <stdio.h>
2  #include <string.h>
3
4  //deklarasi bungkusan
5  typedef struct{
6      char kata[100];
7  }kalimat;
8
9  int main()
10 {
11     //deklarasi
12     int n, i, j;
13     n = 0;
14     //deklarasi variabel bungkusan
15     kalimat input[50];
16     //lakukan do-while selama bukan kata selesai
17     do{
18         //input kata
19         scanf("%s", &input[n].kata);
20         //jika bukan selesai, jumlahkan
21         if (strcmp(input[n].kata, "selesai") != 0){
22             n++;
23         }
24     }while(strcmp(input[n].kata, "selesai") != 0);
25
26
27     /*tampilkan semua string jika n bernilai lebih dari 0*/
28     if (n > 0)
29     {
30         printf("=====\n");
31         for (i = 0; i < n; ++i)
32         {
33             for (j = 0; j < strlen(input[i].kata); ++j)
34             {
35                 printf("%c", input[i].kata[j]);
36                 if(j < strlen(input[i].kata)-1) printf(" ");
37             }
38             printf("\n");
39         }
40     }
41
42     return 0;
43 }

```

```

halo
kenalkan
nama
saya
aldi
selesai
=====
h a l o
k e n a l k a n
n a m a
s a y a
a l d i

```

DOMINO STRING

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      //deklarasi
7      int n, i, j, k;
8      scanf("%d", &n); //input banyak
9
10     //deklarasi variabel
11     char input[n][100];
12     //deklarasi variabel untuk mengecek panjang string
13     int panjang[n];
14     //deklarasi variabel maks untuk mengecek panjang string paling panjang
15     int maxPanjang = 0;
16
17     /*input nilai array of string*/
18     for (i = 0; i < n; ++i)
19     {
20         scanf("%s", &input[i]);
21         //input panjangnya
22         panjang[i] = strlen(input[i]);
23         //cek panjangnya
24         if(maxPanjang < panjang[i]) {
25             maxPanjang = panjang[i];
26         }
27     }
28
29     /*deklarasi array of string baru untuk menampung
30     array of string domino dengan banyak diinisialisasi oleh
31     nilai string paling panjang dan panjang karakternya sebanyak string inputan*/
32     char tampung[maxPanjang][n];
33
34     //pindahkan per karakter array of string inputan kedalam tampungan
35     /*gunakan perulangan*/
36     for (i = 0; i < maxPanjang; ++i)
37     {
38         for (j = 0; j < n; ++j)
39         {
40             //jika nilai i kurang dari panjang string inputan, pindahkan karakter stringnya
41             if (i < strlen(input[j]))
42             {
43                 tampung[i][j] = input[j][i];
44             }
45         }
46     }
```

```

45     //jika tidak, masukkan spasi pada tampung
46     else{
47         tampung[i][j] = ' ';
48     }
49 }
50 }
51
52 printf("\n");
53 //tampilkan string secara per karakter
54 for (i = 0; i < maxPanjang; ++i)
55 {
56     for (j = 0; j < n; ++j)
57     {
58         //jika bukan spasi tampilkan karakter stringnya
59         if (tampung[i][j] != ' ')
60         {
61             printf("%c", tampung[i][j]);
62         }
63         //jika tidak
64         else{
65             /*cek karakter string berikutnya di baris itu
66             jika menemukan huruf didepannya, tampilkan spasi*/
67             int ada = 0;
68
69             k = j;
70             while((k < n) && (ada == 0)) {
71                 if ((tampung[i][k] >= 'a') && (tampung[i][k] <= 'z'))
72                 {
73                     ada = 1;
74                 }
75                 k++;
76             }
77             if (ada == 1)
78             {
79                 printf(" ");
80             }
81         }
82     }
83     printf("\n");
84 }
85
86 return 0;
87 }

```

HASIL PROGRAM DOMINO STRING

```
5  
halo  
semuanya  
kenalkan  
aku  
dudu
```

```
hskad  
aeeku  
lmnud  
oua u  
al  
nk  
ya  
an
```


LATIHAN

- Buatlah sebuah program array of string yang meminta masukan sampai menemukan kata “mantul”. Lalu tampilkan hasilnya seperti berikut!

```
welcome  
to  
the  
jungle  
everybody  
mantul  
=====  
7  
2  
3  
6  
9
```

- Buatlah array of string dengan isi masukan sebanyak n. Lalu tampilkan string nya dalam bentuk domino sebagai berikut!

```
5  
halo  
semuanya  
kenalkan  
aku  
dudu  
  
sk  
ee  
mn  
ua  
hal d  
ankau  
lyakd  
oanuu
```

AKHIR KATA

Penulis mengucapkan terimakasih yang sebesar-besarnya kepada semua elemen yang telah mendukung berjalannya sesi praktikum pada mata kuliah Algoritma dan Pemrograman I kali ini. Semoga apa yang kita dapatkan kali ini bisa menjadi berkah bagi kita semua.

DAFTAR PUSTAKA

Tim Asisten Pemrograman Algoritma dan Pemrograman 1 Angkatan 10. (2019). Array Of String. Modul Praktikum Algoritma dan Pemrograman 1. Bandung, Jawa Barat, Indonesia