



Pendalaman String (Pascal)

Tim Olimpiade Komputer Indonesia

Pendahuluan

Melalui dokumen ini, kalian akan:

- Memahami lebih dalam soal string.
- Mengenal fungsi-fungsi dasar untuk pengolahan string.

Seluruh pembahasan string pada dokumen ini mengacu pada tipe data **string** pada **Pascal**.



Bagian 1

Pengolahan String



Pengolahan String

Pascal menyediakan berbagai fungsi dan prosedur dasar pengolahan string, beberapa di antaranya adalah:

- length
- pos
- copy
- delete
- insert
- str
- val



Pengolahan String: length

length(S)

Merupakan fungsi yang mengembalikan panjang dari string S.

Contoh:

```
s1 := 'bebek';
```

```
s2 := '';
```

```
s3 := 'Pak Dengklek';
```

```
writeln(length(s1)); (* mencetak 5 *)
```

```
writeln(length(s2)); (* mencetak 0 *)
```

```
writeln(length(s3)); (* mencetak 12 *)
```



Pengolahan String:

pos

pos(T , S)

Merupakan fungsi yang mencari dan mengembalikan posisi terawal substring T dari suatu string S .

Jika tidak ditemukan, dikembalikan nilai 0.

Contoh:

```
s := 'Pak Dengklek berternak';
```

```
t1 := 'Dengklek';
```

```
t2 := 'pak';
```

```
t3 := 'klek';
```

```
writeln(pos(t1, s)); (* mencetak 4 *)
```

```
writeln(pos(t2, s)); (* mencetak 0, tidak ditemukan *)
```

```
writeln(pos(t3, s)); (* mencetak 9 *)
```



Pengolahan String: copy

`copy(S, pos, cnt)`

Merupakan fungsi yang mengembalikan substring dari indeks *pos* sebanyak *cnt* karakter dari string *S*.

Contoh:

```
s := 'Pak Dengklek berternak';
```

```
writeln(copy(s, 1, 6)); (* mencetak 'Pak De' *)
```

```
writeln(copy(s, 3, 1)); (* mencetak 'k' *)
```



Pengolahan String: delete

`delete(S, pos, cnt)`

Merupakan prosedur yang menghapus substring dari indeks *pos* sebanyak *cnt* karakter dari string *S*.

Parameter *S* dipanggil dengan *by reference*.

Contoh:

```
s := 'Pak Dengklek berternak';
```

```
delete(s, 2, 3);
```

```
writeln(s); (* mencetak 'PDengklek berternak' *)
```



Pengolahan String:

insert

`insert(T, S, pos)`

Merupakan prosedur yang menyisipkan string T ke dalam string S mulai dari indeks pos .

Parameter S dipanggil dengan *by reference*.

Contoh:

```
s := 'Pak Dengklek berternak';  
t := 'dan Bu ';
```

```
insert(t, s, 5);  
writeln(s); (* mencetak 'Pak dan Bu Dengklek berternak' *)
```



Pengolahan String:

str

str(v, S)

Merupakan prosedur yang mengkonversi suatu data numerik v menjadi string, dan ditampung ke dalam string S .
Parameter S dipanggil dengan *by reference*.

Contoh:

```
s := ' ';  
nilai := 781;  
  
str(nilai, s);  
writeln(s);           (* mencetak '781' *)  
writeln(length(s));   (* mencetak 3 *)
```



Pengolahan String: val

val(S, v, e)

Merupakan prosedur yang mengkonversi suatu string *S* menjadi data numerik, dan ditampung ke dalam variabel *v*.

Jika terjadi **error**, variabel *e* akan berisi nilai yang tidak nol.

Parameter *v* dan *e* dipanggil dengan *by reference*.

Contoh:

```
s := '123';  
nilai := 0;  
e := 0;
```

```
val(s, nilai, e);  
writeln(nilai + 5); (* mencetak 128 *)
```

Error bisa terjadi misalnya ketika *s* bernilai '1a23';



Pengolahan String: val (lanj.)

Parameter e sebenarnya tidak harus ada, sehingga bisa saja ditulis:

```
s := '123';  
nilai := 0;
```

```
val(s, nilai);  
writeln(nilai + 5); (* mencetak 128 *)
```



Operasi Tambahan: konkatenasi

- Konkatenasi adalah penggabungan string.
- Pada Pascal, hal ini dapat dilakukan cukup dengan operasi '+', layaknya operasi numerik.
- Contoh:

```
s := 'Pak';  
t := 'Dengklek';
```

```
gabung := s + t;  
writeln(gabung); (* mencetak 'PakDengklek' *)
```



Bagian 2

Wujud Asli String



Fakta Tentang String...

Sebenarnya, **string** adalah **array of char**!



Fakta Tentang String... (lanj.)

- Pascal membungkus **array[0..255] of char** menjadi **string**, kemudian menambahkan fungsi dan prosedur dasar untuk pengolahan string.
- Elemen ke-0 dari string tidak digunakan untuk menampung karakter, melainkan untuk menyimpan **panjang dari string tersebut**.
- Artinya, tipe data **string** hanya bisa menampung maksimal 255 karakter yang menyusunnya.
- Gunakan tipe data **ansistring** untuk menampung karakter yang lebih banyak dari itu, atau buat **array of char** sendiri :)



Bukti

- Coba jalankan potongan kode berikut!

```
s := 'tes';  
writeln(byte(s[0]));  
s := 'tes lagi'  
writeln(byte(s[0]));
```



Selanjutnya...

- Pembelajaran kalian tentang Bahasa Pascal sudah cukup untuk bisa menuliskan algoritma-algoritma kompleks.
- Berikutnya kita akan mempelajari hal-hal yang lebih berkaitan dengan **algoritma**, bukan sekedar belajar bahasa.

