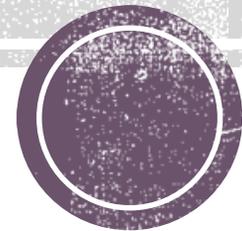


Perancangan Bahasa Pemrograman

Kuliah Online : TEKKOM [2013/VI]



Sumber Perancangan Bahasa Pemrograman

- Bahasa alami (natural language)

Konstruksi yang diturunkan dari bahasa alami, karena bahasa alami dapat digunakan sebagai panduan untuk perancangan sintaks

- Ilmu Matematika

Banyak dipakai untuk aturan-aturan yang terdapat pada bahasa pemrograman, misalnya ekspresi matematika

- Bahasa pemrograman yang telah ada

Bisa menjadi sumber yang bagus untuk perancangan bahasa pemrograman, tetapi perlu ketelitian saat menggunakannya, karena bahasa tersebut mungkin mengandung masalah yang serius, contoh GO TO



Tujuan Perancangan Bahasa Pemrograman

- Komunikasi dengan manusia
- Pencegahan dan deteksi kesalahan
- Usability(mudah dipelajari)
- Efektifitas pemrograman
- Compilability(tidak terlalu kompleks)
- Efisiensi
- Machine independent(tidak bergantung pada satu mesin)
- Kesederhanaan (simplicity)
- Uniformity(mengerjakan hal yang sama dengan cara yang sama)
- Orthogonality(setiap fungsi tidak bergantung pada struktur lain)
- Generalisasi dan Spesialisasi



Detail Rancangan

- Microstructure
- Struktur Ekspresi
- Struktur Data
- Struktur Kontrol
- Struktur Kompilasi
- Struktur I/O



Microstructure

- Mencakup masalah-masalah dalam perancangan bahasa yang mempengaruhi penampilan bahasa tapi tidak mengubah semantiknya
- Aspek terendah adalah set karakter yang dipergunakan.
- Contoh

Keyword : SIZE

Penggunaan Komentar : //, *, {}



Struktur Ekspresi

- Salah satu topik berhubungan dengan ekspresi adalah urutan dari evaluasinya. Metode yang biasa dipergunakan untuk menentukannya berdasar pada dua tingkatan sebagai berikut :
 - 1. Explicit bracketing: menggunakan karakter seperti ‘(,)’, ‘[,]’ untuk membatasi ekspresi.
 - 2. Operator binding: beberapa penggunaannya dengan binding:
 - a) Kiri ke kanan
 - b) Kanan ke kiri dan
 - c) Berdasar prioritas.



Struktur Data

- Deklarasi Data
- Tipe – tipe data yang disediakan dalam bahasa pemrograman
- Strategi alokasi storage
- Lingkup dari variabel



Struktur Data [Deklarasi Data]

- Konstanta

Berguna untuk menuliskan konstanta untuk berbagai tipe data

Contoh : `CONST PI = 22/7`

- Type

Pendekalrasian type atau dalam konstruksi struktur yang lebih kompleks, akan mempermudah penulisan dan readability

Contoh : `TYPE MATRIKS = ARRAY [1..100, 1..100] OF INTEGERE`

- Variabel

Contoh : `VAR a, b, c : integer`



Struktur Data [Tipe Data]

- Sederhana -> Tipe Data Dasara
- Gabungan -> Array, Record
- Kompleks -> Tree, Stack, Quee



Struktur Data [Strategi Alokasi Storage]

- Alokasi static
- Alokasi local, dinamis dan otomatis : Store Procedure
- Retention : tidak banyak digunakan



Struktur Data [Lingkup dari Variabel]

- Batasan untuk mengurangi kompleksitas system
- Biasanya dilakukan pembatasan interaksi antara segmen yang berbeda
- Misal : penggunaan BEGIN – END untuk memulai suatu prosedur atau suatu algoritma



Struktur Kontrol

- Begin – End
- If – Then – Else
- Case – Of
- Repeat – Until
- For – End



Struktur Kompilasi

- Tahapan yang berhubungan dengan proses kompilasi, sebagai contoh dalam bahasa C pada saat kompilasi ada beberapa file yang terpisah sehingga membutuhkan menyisipkan file tersebut dengan menggunakan perintah *Include*.



Struktur I/O

- Merupakan fasilitas untuk menangani masukan dan keluaran
- Tingkatan :
 - Bentuk *Format Free*,

Berguna untuk komunikasi sederhana bagi para programmer untuk memeriksa kebenaran program. Programmer bisa menampilkan dengan mudah nilai dari variable dan untuk memeriksa secara cepat logik dari program pada sekumpulan input data



Struktur I/O

➤ Bentuk *Formatted*

Nilai setiap variable dalam masukan atau keluaran ditulis atau dibaca sesuatu dengan format yang ditentukan, misalnya panjang dari field dan tipe data.

Contoh :

Jika $x=4$ dan $s="nama"$, maka

```
printf ("nilai x adalah %d \n nilai s adalah %s", x,s)
```

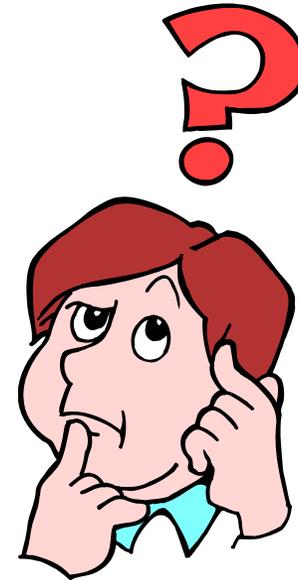
Maka output :

```
nilai x adalah 4
```

```
nilai s adalah nama
```



ANDAIKAN.....



Anda akan menciptakan sebuah bahasa program, coba sebutkan urutan proses yang harus ditentukan/skenario yang dijalani untuk menghasilkan bahasa pemrograman impian Anda tersebut !



Skenario Perancangan

1. Tentukan apa yang diinginkan.
2. Tentukan feature yang mungkin
3. Tentukan desain dan sesuaikan dengan featurenya
4. Tentukan rincian, parsing, dan error checking.



Skenario Perancangan

5. Tuliskan user manual dan help.
6. Evaluasilah, jika salah mulai lagi dari langkah 3.
7. Jika sudah benar, optimisasilah dan uji segala kemungkinan.
8. Cobakan kepada pengguna, tunggu reaksinya.
9. Perbaiki bug dan mulai versi baru.

