PERTEMUAN 6

DESIGN BAHASA PEMROGRAMAN

A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi ini diharapkan mahasiswa mampu memahami tujuan dan philosophi rancangan bahasa pemrograman.

B. URAIAN MATERI

1. Linguistik Pemrograman

Bahasa pemrograman tingkat tinggi pertama dirancang selama tahun 1950-an. Sejak saat itu, bahasa pemrograman telah menjadi bidang studi yang menarik dan produktif. Pada tingkat yang lebih akademis, ilmuwan komputer mencari cara untuk merancang bahasa pemrograman yang menggabungkan kekuatan ekspresif dengan kesederhanaan dan efisiensi.

Istilah linguistik pemrograman terkadang digunakan untuk mengartikan studi bahasa pemrograman. Ini adalah analogi dengan disiplin linguistik yang lebih tua, yang merupakan studi tentang bahasa alami. Baik bahasa pemrograman maupun bahasa alami yang memiliki sintaks (bentuk) dan semantik (makna). Namun, kita tidak bisa mengambil *analoginya* terlalu jauh. Bahasa alami jauh lebih luas, lebih ekspresif, dan lebih halus daripada bahasa pemrograman.

Bahasa alami adalah bahasa yang dibicarakan dan ditulis oleh populasi manusia, jadi ahli bahasa dibatasi untuk menganalisis bahasa alami yang ada (dan sudah mati). Di sisi lain, ahli bahasa pemrograman tidak hanya dapat menganalisis bahasa pemrograman yang ada, mereka juga dapat merancang dan menentukan bahasa pemrograman baru, dan mereka dapat menerapkan bahasa ini di komputer.

2. Sejarah Bahasa Pemrograman

Awal terciptanya bahasa pemrograman dimulai dari Antikythera yang berasal dari yunani kuno. Pada tahun 1200 seorang ilmuan islam yaitu Ismail

Al-Jazari membangun sebuah mesin bernama Automata, yaitu sebuah robot burung merak yang digerakkan oleh hydropower atau aliran air.

Pada tahun 1822 mahasiswa dari universitas Cambridge Inggris yang bernama Charles Babbage menciptakan sebuah mesin bernama Difference Engine. Namun mesin itu hanya bisa mengeluarkan satu jenis output. Pada tahun 1849 Charles Babbage mengembangkan mesin pengolah data buatannya hingga menjadi versi kedua. Pembuatan mesin yang dilakukan oleh Babbage dilanjutkan oleh Henry Prevost yaitu anak dari Charles Babbage. Henry pun berhasil membuat kopian dari perhitungan algoritma mesinn tersebut dan mengirimnya ke berbagai institusi di dunia.

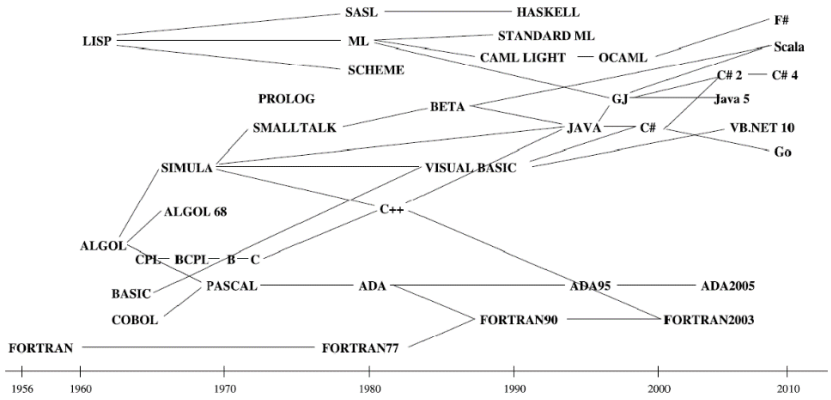
Lalu pada tahun 1854, George Boole menemukan sistem logika yang dikenal dengan logika Boole. Logika ini menyatakan hubungan-hubungan lebih besar, lebih kecil, sama dengan dan tidak sama dengan. Tahun demi tahun para ilmuan terus mengembangkan penemuan ini, hingga pada tahun 1935 sebuah mesin kalkulator biner dengan nama Z-1 berhasil diciptakan oleh seorang ilmuan asal jerman bernama Konrad Zuse. Kemudian pada tahun 1939, Zuse mendapatkan tawaran untuk mengabdi kepada militer dengan membuat Z-2, Z-3 dan Z-4.

Pada tahun 1945 terciptalah bahasa pemrograman tingkat tinggi pertama didunia, yaitu Plankalkus (Plan Kalkulus). Dengan itu terciptalah mesin catur komputer pertama didunia. Diawal tahun 1950an Alick Glennie dari Universitas Manchester mengembangkan bahasa pemrograman Autocode. Bahasa ini menggunakan kompiler yang mengkonversi bahasanya secara otomatis ke bahasa mesin. Tahun 1952 adalah awal digunakannya bahasa pemrograman yang dipergunakan untuk komputer Mark 1 di Universitas Manchester.

Sejak 1956, ribuan bahasa pemrograman telah diusulkan dan diimplementasikan, beberapa ratus di antaranya telah digunakan secara luas. John W. Backus membuat proposal untuk atasannya di IBM (International Business Machines Corporation) untuk mengembangkan sebuat bahasa alternatif yang tentunya lebih praktis untuk memprogram IBM 704 mainframe computer dengan nama Formula Translation atau yang kita kenal dengan FORTRAN. Kompiler FORTRAN berhasil diselesaikan pada tahun 1957.

Sebagian besar bahasa pemrograman baru muncul sebagai reaksi terhadap beberapa bahasa yang telah diketahui (dan disukai atau tidak disukai

oleh desainer), sehingga seseorang dapat mengusulkan pohon keluarga atau silsilah untuk bahasa pemrograman, seperti untuk organisme hidup.



Gambar 5.1 Silsilah bahasa pemrograman

Tentu saja ada lebih banyak bahasa daripada yang ditampilkan, khususnya jika seseorang menghitung juga lebih banyak bahasa khusus domain seperti Matlab, SAS dan R, dan bahasa pemograman aneh lainnya.

3. Bahasa Pemrograman

Merancang bahasa pemrograman baru adalah jenis aktivitas pemrograman metalevel yang sama menyenangkannya dengan pemrograman dalam bahasa biasa. Anda akan menemukan bahwa kejelasan dan kesederhanaan bahkan lebih penting dalam desain bahasa daripada dalam pemrograman biasa. Saat ini ratusan bahasa pemrograman digunakan, baik itu bahasa skrip untuk perdagangan Internet, alat pemrograman antarmuka pengguna, makro spreadsheet, maupun bahasa spesifikasi format halaman yang bila dijalankan dapat menghasilkan dokumen yang diformat. Desain aplikasi yang diinginkan seringkali membutuhkan programmer untuk menyediakan bahasa pemrograman baru atau untuk mengembangkan yang sudah ada. Ini karena aplikasi yang fleksibel dan dapat diperluas perlu menyediakan semacam kemampuan pemrograman kepada pengguna akhir mereka.

Sebelum kita mempelajari tentang perancangan bahasa pemrograman hal yang utama harus kita ketahui adalah apa itu bahasa pemrograman?

Bahasa pemrograman atau sering diistilahkan bahasa pemrograman komputer adalah intruksi standar yang berfungsi untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks, dan sematik yang dipakai untuk mendefinisikan program komputer, dengan bahasa ini seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, disimpan atau diteruskan dan jenis langkah apa yang akan diambil dalam berbagai situasi.

Bahasa pemrograman berbeda dengan bahasa alami. Bahasa alami umumnya digunakan untuk berinteraksi antar manusia, sedangkan bahasa pemrograman digunakan oleh manusia untuk berinteraksi dengan komputer. Manusia dapat mengendalikan atau memerintahkan komputer dengan bahasa pemrograman, dan orang yang dapat memerintah komputer itu disebut ebagai seorang programmer.

4. Tujuan Perancangan Bahasa Pemrograman

Setiap perancangan pasti memiliki tujuan untuk mempermudah, baik itu penggunaannya maupun membacanya. Berikut adalah tujuan dari perancangan bahasa pemrograman:

a. Komunikasi dengan manusia

Hal ini adalah tujuan utama dari terciptanya perancangan bahasa program, karena sebuah program yang baik adalah suatu program yang mudah dimengerti dan dipahami.

Contoh : a/b/c

Bisa berarti a dibagi dengan b, hasilnya dibagi dengan c atau a dibagi dengan hasil pembagian b dengan c

b. Pencegahan dan deteksi kesalahan

Hal ini bertujuan agar program dapat mengidentifikasi error yang mungkin terjadi, mempermudah deteksi kesalahan dan menghilangkan kesalahan.

c. Usability(mudah dipelajari)

Berkaitan dengan aspek kenyamanan seorang programmer, karena bahasa pemrograman yang baik harus dapat dipelajari dan mudah diingat karena

berkaitan dengan kenyamanan programmer dalam menggunakan bahasa pemrograman.

d. Efektifitas pemrograman

Merupakan bagian dari rekayasa perangkat lunak yaitu bagaimana mancatat keputusan yang dibuat selama mengembangkan program.

e. Compilability (Kesesuaian)

Perancangan diperlukan agar terdapat kesesuaian antara pemilihan bahasa pemrograman, tipe data hingga hardware yg diperlukan.

f. Efisiensi

Peningkatan yang signifikan dalam efisiensi bisa didapat dengan membuat bahasa lebih sederhana untuk menghasilkan kode yang efisien.

g. Machine independent (tidak bergantung pada satu mesin)

Sebuah program yang telah sukses dikompilasi dan dieksekusi pada suatu mesin, saat dipindahkan ke mesin lainnya akan berjalan dengan input dan output yang sama.

h. Kesederhanaan (Simplicity)

Kesederhanaan dapat dicapai melalui:

1) Pembatasan-pembatasan tujuan
2) Perhatian pada keterbacaan
3) Pendefinisian yang baik, dan
4) Konsep yang sederhana.

i. Uniformity

Uniformity bisa didefinisikan sebagai mengerjakan hal yang sama dengan cara yang sama. Bila diadopsi sebagai prinsip bahasa pemrograman, ini bisa menolong mengurangi hal-hal yang harus diingat pemrogram, karena akan lebih mudah memahami bagaimana suatu feature akan dilakukan.

j. Orthogenality

Gagasan mendasar dari orthogenality adalah setiap fungsi bekerja tanpa perlu mengetahui struktur dari yang lain.

5. Prinsip Dasar Bahasa Pemrograman

Suatu bahasa pemrograman dibuat dengan tujuan untuk mempermudah dibaca dan ditulis oleh seorang programmer yang berisikan sebuah intruksi untuk memerintah komputer supaya bisa menjalankan fungsi tertentu.

Suatu bahasa pemrograman dapat dibaca dan ditulis dimudahkan oleh prinsip-prinsip sebagai berikut:

a. Sintaks

Sintaks menjelaskan struktur program yang benar. Sintaks merupakan kumpulan aturan yang mendefinisikan suatu bentuk bahasa. Bagaimana suatu kalimat dibentuk sebagai barisan atau urutan dari pemilihan suatu kata dasar. Dengan menggunakan aturan ini maka suatu kalimat dapat dikatakan legal atau tidak legal. Sebagai contoh, dalam keyword bahasa C (seperti while, do,if, dan else), identifier, angka, operator, dan seterusnya, merupakan kata dalam suatu bahasa.Sintaks dalam bahasa C mengatur cara mengombinasikan kata-kata tersebut ke dalam suatu statement dengan bentuk yang benar sehingga dapat disusun suatu program yang dapat berjalan dengan benar.

Sintaks berfungsi menyediakan bentuk-bentuk notasi untuk komunikasi antar programmer dan pemroses bahasa pemrograman sehingga dapat mempermudah pembuatan suatu program.

Contoh : pada pembuatan program Pascal antara dua statement dipisahkan oleh titik koma (;). X := 1 ; X := X + 1;

b. Tipe sistem dan semantik

Semantik mendefinisikan arti dari program yang benar secara sintak dari bahasa tersebut. Semantik suatu bahasa membutuhkan semacam ekspresi untuk mengirimkan suatu nilai kebenaran (TRUE, FALSE, NOT atau nilai integer).

c. Manajemen memori

Tindakan mengelola memori komputer. Kebutuhan utama manajemen memori adalah untuk menyediakan cara untuk secara dinamis mengalokasikan bagian-bagian dari memori untuk program atas permintaan

mereka, dan membebaskan untuk digunakan kembali ketika tidak lagi diperlukan.

d. Exception handling

Exception handling adalah penanganan atau proses menanggapi suatu kejadian terhadap pelaksanaan suatu program jika terdapat kesalahan dalam runtime sehingga aliran aplikasi normal dapat terjaga. Keuntungan dari exception handling adalah untuk menjaga aliran normal suatu aplikasi.

6. Detail Perancangan Bahasa Pemrograman

a. Microstructure

Pada dasarnya mencakup masalah-masalah dalam perancangan bahasa yang mempengaruhi penampilan bahasa tanpa mengubah semantiknya. Prinsip mendasar micro-struture adalah arti dari suatu konstruksi, seperti operator, harus jelas dari wujudnya.

Aspek terendah dari microstruture adalah set karakter yang dipergunakan harus menjadi suatu standar untuk menghindari perubahan program saat dipindah antar mesin. (ASCCI 7 bit merupakan set karakter terbaik).

Aspek lain dari mikrostruktur adalah pengaturan komentar, Suatu komentar adalah diawali dari suatu tanda simbol awal komentar sampai dengan ditemukannya tanda simbol tertentu sebagai akhir komentar, walaupun ada karakter apapun didalamnya termasuk spasi.
Simbol komentar idealnya

1) Terdiri dari 2 karakter, lebih baik barupa karakter yang sama
2) Simbolnya jarang digunakan.
3) Terdiri dari karakter yang berlokasi sama pada keyboard

b. Struktur Ekspresi

Struktur ekspresi berhubungan dengan ekspresi adalah urutan dan 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 penggunanya dengan binding:
   a) Kiri ke kanan
   b) Kanan ke kiri dan
   c) Berdasar prioritas.

c. Struktur Data

Struktur data memiliki empat aspek:

1) Deklarasi Data

   Kebutuhan bahasa pemrograman berkaitan dengan deklarasi:

   a) Konstanta

      Berguna untuk menuliskan konstanta untuk berbagai tipe data.

      Contoh : CONST PI = 22/7

   b) Type

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

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

   c) Variabel

      Variabel berguna untuk memilih jenis dari variabel itu sendiri

      Contoh : VAR a, b, c : integer atau varchar

2) Tipe-tipe data yang disediakan dalam bahasa pemrograman
   a) Tidak ada sama sekali (*Assembly*)
   b) Soft Typing (Variabel boleh memuat nilai apapun)
   c) Hard Typing (Variabel hanya boleh memuat nilai yang menjadi domain dari tipe tersebut)

3) Strategi Alokasi Storage

   Alokasi storage untuk variabel dalam suatu program merupakan pekerjaan kompilator dan sistem pada saat run time.

   Terdapat empat bentuk alokasi variabel :

   a) Alokasi statik (kegunaan : pemakainya tidak dalam prosedure tapi secara global)
   b) Alokasi Lokal, dinamik dan otomatis (kegunaan: untuk alokasi variable dalam prosedure)
   c) Retention (Kegunaan: pada implementasi algoritma back tracking)
   d) Alokasi Heap

4) Lingkup Variable

Tujuannya adalah agar tidak terdapat dua nama untuk interprestasi yang sama.

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

d. Struktur Kontrol

Struktur kontrol paling sederhana adalah kombinasi dari beberapa statment ke dalam statement tunggal.

Contoh :

1) Begin – End
2) If – Then – Else
3) Case – Of
4) Repeat – Until
5) For – End

e. Struktur Kompilasi

Stuktur kompilasi mencakup aspek dari bahasa yang berkaitan dengan proses kompilasi, berhubungan dengan operasi pada bahasa yang dikerjakan saat kompilasi dan bagaimana kompilasi modul berbeda dan terpisah dari program.

f. Struktur I/O

Struktur I/O merupakan fasilitas untuk menangani masukan dan keluaran. Fasilitas ini bisa disediakan pada berbagai tingkatan berikut dalam suatu bahasa :

1) 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 lojik dari program pada sekumpulan input data.

2) 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.

7. Tipe Data Dalam Bahasa Pemrograman

Tipe data menentukan kumpulan nilai data dan sekumpulan operasi yang telah ditentukan sebelumnya pada nilai tersebut. Program komputer menghasilkan hasil dengan memanipulasi data. Faktor penting dalam menentukan kemudahan mereka melakukan tugas ini adalah seberapa baik tipe data yang tersedia dalam bahasa yang digunakan cocok dengan objek di dunia nyata dari masalah yang sedang ditangani. Oleh karena itu, sangat penting bahwa suatu bahasa mendukung kumpulan tipe dan struktur data yang sesuai.

Konsep kontemporer pengetikan data telah berkembang selama 60 tahun terakhir. Dalam bahasa paling awal, semua struktur data ruang masalah harus dimodelkan dengan hanya beberapa struktur data dasar yang didukung bahasa. Misalnya, di Fortrans pra-90, daftar tertaut dan pohon biner diimplementasikan dengan array.

Mengambil konsep tipe yang ditentukan pengguna selangkah lebih maju, kita sampai pada tipe data abstrak, yang didukung oleh sebagian besar bahasa pemrograman yang dirancang sejak pertengahan 1980-an. Ide dasar dari tipe data abstrak adalah bahwa antarmuka suatu tipe, yang terlihat oleh pengguna, dipisahkan dari representasi dan rangkaian operasi pada nilai-nilai tipe itu, yang tersembunyi dari pengguna. Semua tipe yang disediakan oleh bahasa pemrograman tingkat tinggi adalah tipe data abstrak.

Ada beberapa penggunaan sistem tipe bahasa pemrograman. Yang paling praktis adalah deteksi kesalahan. Proses dan nilai pemeriksaan tipe, yang diarahkan oleh sistem tipe bahasa. Penggunaan kedua dari sistem tipe

adalah bantuan yang disediakannya untuk modularisasi program. Ini hasil dari pemeriksaan jenis modul silang yang memastikan konsistensi antarmuka antar modul. Penggunaan lain dari sistem tipe adalah dokumentasi. Deklarasi jenis dalam program mendokumentasikan informasi tentang datanya, yang memberikan petunjuk tentang perilaku program.

Sistem tipe dari bahasa pemrograman menentukan bagaimana tipe dikaitkan dengan setiap ekspresi dalam bahasa dan menyertakan aturannya untuk kesetaraan tipe dan kompatibilitas tipe. Tentu saja, salah satu bagian terpenting dalam memahami semantik bahasa pemrograman adalah memahami sistem tipenya.

8. Jenis Tipe Data

Tipe data yang tidak didefinisikan dalam tipe lain disebut tipe data primitif. Hampir semua bahasa pemrograman menyediakan sekumpulan tipe data primitif. Beberapa tipe primitif hanyalah refleksi dari perangkat keras. sebagai contoh, kebanyakan tipe integer, yang lain hanya membutuhkan sedikit dukungan nonperangkat keras untuk implementasinya.

Untuk menentukan tipe terstruktur, tipe data primitif dari suatu bahasa digunakan, bersama dengan satu atau lebih tipe konstruktor. Tipe data primive terbagi menjadi 3 tipe data yaitu tipe numerik, tipe boolean dan tipe karakter.

a. Tipe Numerik

Beberapa bahasa pemrograman awal hanya memiliki tipe primitif numerik. Jenis numerik masih memainkan peran sentral di antara koleksi jenis yang didukung oleh bahasa kontemporer.

Tipe data numerik adalah tipe data yang digunakan pada variabel atau konstanta untuk menyimpan nilai dalam bentuk angka. Jika bilangan desimal maka menggunakan float atau double dan jika bilangan bulat maka menggunakan byte, short, int dan long.

1) Tipe data float

Tipe ini digunakan untuk menandakan nilai-nilai yang mengandung presisi atau ketelitan tunggal (single-precision) yang menggunakan ruang penyimpanan 32-bit. Presisi tunggal biasanya lebih cepat untuk

processor-processor tertentu dan memakan ruang penyimpanan setengah kali lebih sedikit dibandingkan presisi ganda (double precision). Permasalahan yang timbul dari pemakaian tipe float untuk nilai-nilai yang terlalu kecil atau justru terlalu besar, karena nilai yang dihasilkan akan menjadi tidak akurat.

a) Ukuran memori 32-bit

b) Digunakan untuk bilangan desimal

c) Default value adalah 0.0

d) Contoh: float f1 = 234.5

2) Tipe data double

Tipe ini mengandung tingkat ketelitian ganda atau presisi ganda (double precisiondan menggunakan ruang penyimpanan 64-bit untuk menyimpan nilai. Tipe double tentu lebih cepat untuk melakukan perhitungan-perhitungan matematis daripada tipe float. Untuk perhitungan yang bersifat bilangan riil dan menghasilkan hasil yang lebih akurat, maka lebih baik menggunakan tipe double.

a) Ukuran memori 64-bit

b) Pilihan utama untuk bilangan desimal

c) Default value adalah 0.0

d) Contoh: double d1 = 123.4

3) Tipe data byte

Type byte umumnya digunakan pada saat kita bekerja dengan sebuah data stream dari suatu file maupun jaringan, yaitu untuk keperluan proses membaca/menulis. Selain itu, tipe ini juga digunakan saat bekerja dengan data biner yang tidak kompatibel dengan tipe-tipe lain yang didefiniskan di dalam bahasa pemrograman.

a) Ukuran memori 8-bit

b) Nilai miminimal -128 (-2^7)

c) Nilai maksimal 127 ($2^7 -1$)

d) Default value adalah 0

e) Contoh: byte a = 100, byte b = -50

4) Tipe data short

Pada umumnya diaplikasikan pada komputer-komputer 16-bit, yang saat ini semakin jarang keberadaanya.

a) Ukuran memori 16-bit

b) Nilai miminimal -32.768 (-2^{15})

c) Nilai maksimal 32.767 (2^{15} -1)

d) Default value adalah 0

e) Contoh: short s = 10000, short r = -20000

5) Tipe data int

Tipe ini juga merupakan tipe yang paling banyak dipakai dalam menggunakan angka pada bahasa pemrograman, dikarenakan dianggap paling efisien dibandingkan dengan tipe-tipe integer lainnya. Tipe int banyak digunakan untuk indeks dalam struktur pengulangan maupun dalam konstruksi sebuah array. Selain itu, secara teori setiap ekspresi yang melibatkan tipe integer (byte, short, int, long) semuanya akan dipromosikan ke int terlebih dahulu sebelum dilakukan proses perhitungan. Namun kelemahan dari tipe ini adalah range yang terbatas, sehingga untuk penggunaan sepeti nomor telepon tidak bisa menggunakan tipe ini.

a) Ukuran memori 32-bit

b) Nilai miminimal -2.147.483.648 (-2^{31})

c) Nilai maksimal 2.147.483.647(2^{31} -1)

d) Default value adalah 0

e) Contoh: int a = 100000, int b = -200000

6) Tipe data long

Tipe ini digunakan untuk kasus-kasus tertentu yang nilainya berada di luar rentang tipe int, karna tipe ini punya range paling tinggi dibanding Integer lainnya. Dengan kata lain, tipe long terpaksa digunakan jika data memiliki range diluar range int.

a) Ukuran memori 64-bit

b) Nilai miminimal -9.223.372.036.854.775.808 (-2^{63})

c) Nilai maksimal 9.223.372.036.854.775.807 (2^{63} -1)

d) Default value adalah 0

e) Contoh: long a = 100000, long b = -200000

7) Tipe data komplex

Beberapa bahasa pemrograman mendukung tipe data yang kompleks misalnya, Fortran dan Python. Nilai kompleks direpresentasikan sebagai pasangan terurut nilai floating-point. Bahasa yang mendukung tipe kompleks mencakup operasi aritmatika pada nilai kompleks.

Bilangan kompleks tersusun atas dua bilangan pecahan yaitu bagian riil ditambah dengan bagian imaginer yang nilainya diakhiri dengan huruf j atau dalam disiplin ilmu lain ditandai dengan huruf i dibelakannya.

Contoh x = 2.1+4j

print(x)

b. Tipe Boolean (Logika)

Jenis Boolean mungkin yang paling sederhana dari semua jenis. Rentang nilainya hanya memiliki dua elemen: satu untuk benar dan satu untuk salah. Mereka diperkenalkan di ALGOL 60 dan telah dimasukkan dalam sebagian besar bahasa tujuan umum yang dirancang sejak 1960. Satu pengecualian populer adalah C89, di mana ekspresi numerik digunakan sebagai kondisional. Dalam ekspresi seperti itu, semua operan dengan nilai bukan nol dianggap benar, dan nol dianggap salah. Meskipun C99 dan C ++ memiliki tipe Boolean, mereka juga memungkinkan ekspresi numerik digunakan seolah-olah mereka adalah Boolean. Ini tidak terjadi pada bahasa berikutnya, Java dan C #.

Tipe Boolean sering digunakan untuk merepresentasikan switch atau flag dalam program. Meskipun tipe lain, seperti integer, dapat digunakan untuk tujuan ini, penggunaan tipe Boolean lebih mudah dibaca.

Nilai Boolean dapat diwakili oleh satu bit, tetapi karena satu bit memori tidak dapat diakses secara efisien pada banyak mesin, mereka sering

disimpan dalam sel memori terkecil yang dapat dialamatkan secara efisien, biasanya dalam satu byte.

Tipe data boolean mempunyai dua nilai yaitu true (benar) dan false (salah). Nilai Boolean sangat penting digunakan untuk pengambilan keputusan dalam suatu kejadian tertentu.

1) Hanya memiliki dua nilai yaitu: *true* dan *false*)
2) tipe data boolean merepresentasikan satu bit informasi
3) Tipe data ini digunakan untuk menandai sebuah kondisi
4) Default value adalah false
5) Contoh:

```
x = True
y = False
print (x and y)
print (x or y)
```

c. Tipe Karakter (char)

Char adalah karakter tunggal yang didefinisikan dengan diawali dan diakhiri dengan tanda “ (petik tunggal). Char berbeda dengan String, karena String bukan merupakan tipe data primitif, tetapi sudah merupakan sebuah objek. Tipe char mengikuti aturan unicode, sehingga dapat menggunakan kode \u kemudian diikuti bilangan dari 0 sampai 65535, tetapi yang biasa digunakan adalah bilangan heksadesimal dari 0000 sampai FFFF. Misalnya : “\u123". Contoh bahasa pemrograman Java menggunakan karakter Unicode untuk merepresentasikan semua karakter yang ada. Unicode ialah sekumpulan karakter yang terdapat pada semua bahasa, seperti bahasa Latin, Arab, Yunani dan lain-lainnya.

1) Ukuran memori 16-bit
2) Untuk menyimpan karakter apapun
3) Contoh: char letterA = ‘A’

C. SOAL LATIHAN/TUGAS

1. Pada tahun berapa bahasa pemrograman tingkat tinggi diciptakan untuk pertama kalinya didunia?
2. Jelaskan apa tujuan dari perancangan bahasa pemrograman!
3. Tipe data apa yg cocok untuk membuat layout sebagai berikut!
   a. 890.869
   b. 879578
   c. 087894875947

D. REFERENSI

Robert W. Sebesta. (2019). *Concept of Pogramming Languages.* University of Colorado at Colorado Springs:New York. Edisi 12

Franklyn Turbak and David Gifford bersama Mark A. Sheldon. (2008). *Design Concepts in Programming Languages*. The MIT Press:Inggris

David A. Watt dan William Findlay. (2004). *Programming Language Design Concept.* University of Glasgow:Inggris

Blogspot. *Teknik Kompilasi.* (diakses pada 04 November 2020) Tersedia pada http://dewisuciawati.blogspot.com/2014/10/teknik-kompilasi.html

Dicoding. *Sejarah Bahasa Pemrograman dibuat.* (Diakses pada 06 November 2020). Tersedia pada https://www.dicoding.com/blog/sejarah-bahasa-pemrograman/

Kelas Programmer. *Tipe Data di Python (String, Numerik, Boolean).* (Diakses pada 07 November 2020). Tersedia pada https://kelasprogrammer.com/tipe-data-python/

GLOSARIUM

Bahasa pemrograman adalah kumpulan sintaks yang berfungsi untuk memerintahkan perangkat lunak pada komputer.

Tipe data adalah himpunan yang bisa user temui pada semua data.