

TIPE DATA BOOLEAN

Cici Handisha 185100019

Fakultas Komputer cicihandisha.student@umitra.ac.id

Abstract

Tipe data boolean sebenarnya sangat serderhana. Tipe data ini hanya bisa diisi dengan salah satu dari 2 nilai: TRUE atau FALSE. Tipe data boolean banyak digunakan untuk percabangan kode program, atau untuk memutuskan apa yang mesti dijalankan ketika sebuah kondisi terjadi.

Sebagai contoh, kita bisa membuat kode program untuk menentukan apakah sebuah angka genap atau ganjil berdasarkan input dari pengguna. Untuk keperluan ini kita harus mengecek terlebih dahulu apakah angka itu bisa dibagi 2 (untuk angka genap), atau tidak bisa dibagi 2 (untuk angka ganjil). Tipe data boolean bisa digunakan untuk menampung kondisi seperti ini, benar atau salah (*True* atau *False*).

Contoh penggunaan tipe data boolean akan lebih jelas jika anda telah mempelajari kondisi percabangan program seperti IF (yang akan saya jelaskan pada tutorial terpisah). Untuk sementara anda hanya perlu memahami bahwa tipe data boolean adalah tipe data yang hanya memiliki 2 isi: TRUE atau FALSE.

Kata Kunci : Tipe Data Boolean

A. PENDAHULUAN

Salah satu bahasa yang paling awal untuk memberikan tipe data Boolean yang eksplisit adalah ALGOL 60 (1960) dengan operator nilai benar dan palsu dan logis dinotasikan dengan simbol '\ wedge' (dan), '\ vee' (atau), '\ supset' (berarti), '\ equiv' (kesetaraan), dan '\ neg' (tidak). Karena keterbatasan perangkat input waktu, namun, kompiler paling banyak digunakan representasi alternatif untuk yang terakhir, seperti DAN atau 'DAN'. Pendekatan ("Boolean adalah terpisah built-in tipe data primitif") diadopsi oleh bahasa kemudian, banyak seperti ALGOL 68 (1970) [3], Jawa, dan C#.

Versi pertama dari FORTRAN (1957)penggantinya dan FORTRAN II (1958) tidak memiliki nilai logis atau operasi, bahkan pernyataan kondisional JIKA mengambil sebuah ekspresi aritmatika dan bercabang ke salah satu dari sesuai tiga lokasi dengan tandanya. **FORTRAN** IV (1962),namun, mengikuti ALGOL 60 misalnya dengan menyediakan tipe data Boolean literal (LOGIS), kebenaran (TRUE.. Dan. FALSE.), operator Bernilai Boolean perbandingan numerik (. EQ., GT.., dll), dan operator logika (TIDAK.., DAN.., ATAU.). Dalam laporan FORMAT, karakter kontrol tertentu ('L')

diberikan untuk parsing atau format nilai logis. [4]

B. PEMBAHASAN / STUDI KASUS

Bahasa Lisp (1958) tidak pernah memiliki built-in tipe data Boolean. Sebaliknya, bersyarat konstruksi seperti COND berasumsi bahwa nilai logis "palsu" yang diwakili oleh daftar kosong (), yang didefinisikan untuk menjadi sama dengan atom khusus nihil atau NIL; sedangkan setiap s ekspresi-lain ditafsirkan "benar" sebagai Untuk kenyamanan, sebagian dialek modern Lisp predefine t atom memiliki nilai t, sehingga kita dapat menggunakan t sebagai notasi mnemonik untuk "benar". Pendekatan ("nilai digunakan apapun dapat sebagai nilai Boolean") dipertahankan dalam dialek Lisp sebagian besar (Common Lisp, Scheme, Emacs Lisp), dan model serupa diadopsi oleh scripting bahasa banyak, bahkan yang memiliki tipe Boolean yang berbeda atau nilai-nilai Boolean; walaupun nilai-nilai yang diinterpretasikan sebagai "palsu" "benar" dan yang bervariasi dari bahasa ke bahasa. Dalam Skema, misalnya, "false" nilai adalah berbeda atom dari daftar kosong, sehingga yang terakhir ini ditafsirkan sebagai "benar". Pada Python, nilai numerik dari nol (integer atau pecahan), nilai

null (Tidak ada), dan kontainer kosong (string yaitu, daftar, set, dll) dianggap salah Boolean; semua nilai-nilai lain dianggap Boolean benar secara default. Di Ruby, di sisi lain, hanya objek null dan benda palsu khusus adalah "false", segala sesuatu yang lain (termasuk integer 0 dan array kosong) "benar". adalah Dalam JavaScript, string "kosong", null, undefined, NaN, +0, -0, dan false [5] kadang-kadang disebut "falsy", dan mereka melengkapi, "truthy", untuk membedakan tipeantara diperiksa secara ketat dan memaksa boolean. [6] Bahasa seperti PHP juga menggunakan pendekatan ini.

Standar awal untuk bahasa C (1972) tidak memberikan tipe Boolean, dan, sampai hari ini, nilai Boolean biasanya diwakili oleh bilangan bulat (int) dalam C. Perbandingan program ('>','==', operator dll) didefinisikan untuk mengembalikan signed integer (int) hasil, baik nol (untuk false) atau 1 (untuk yang benar). Konvensi yang sama diasumsikan oleh operator logika ('&&','||',, '!', Dll) dan pernyataan kondisi-testing ('jika', 'sementara'). Dengan demikian nilai-nilai logika dapat disimpan dalam variabel integer, dan digunakan di mana bilangan saja bulat akan berlaku, termasuk dalam pengindeksan, aritmatika. parsing, format. dan

Pendekatan ("Boolean nilai hanya bilangan bulat") dipertahankan dalam semua versi C.

Beberapa dialek yang, seperti C99 dan Objective-C. memberikan definisi standar tipe Boolean sebagai sinonim dari int dan macro untuk "palsu" dan "benar" sebagai 0, dan 1 masing-masing. Visual Basic menggunakan pendekatan yang sama. C + + memiliki Boolean terpisah tipe data ('bool'), tetapi dengan konversi otomatis dari nilai skalar dan pointer yang sangat mirip dengan C. Pendekatan ini diadopsi juga oleh bahasa banyak kemudian, terutama oleh beberapa orang scripting seperti AWK dan Perl. Satu masalah dengan pendekatan ini adalah bahwa tes jika (t){...} == TRUE dan jika (t) tidak setara. Python memiliki situasi yang terkait, dimana Boolean, bool merupakan subtipe dari integer int jenis,, dan boolean bertindak benar dan salah sebagai 0 dan 1, masing-masing, dalam konteks aritmatika.

Pascal (1978)Bahasa memperkenalkan konsep tipe programmerenumerated didefinisikan. Built-in tipe data Boolean kemudian diberikan sebagai tipe enumerated yang telah ditetapkan dengan nilainilai **FALSE** dan TRUE. Menurut definisi, semua perbandingan, operasi logis, pernyataan bersyarat dan

diterapkan untuk dan / atau menghasilkan nilai Boolean. Jika tidak, tipe Boolean memiliki semua fasilitas yang tersedia untuk tipe enumerated secara umum seperti pemesanan dan digunakan sebagai indeks. Di sisi lain, konversi antara boolean dan bulat (atau jenis bilangan lainnya) masih diperlukan tes eksplisit atau pemanggilan fungsi, seperti pada ALGOL Pendekatan 60. ("Boolean adalah tipe enumerasi") diadopsi oleh bahasa yang paling kemudian yang telah disebutkan jenis, seperti Modula, Ada dan Haskell.

Setelah tipe enumerated (enum "s) ditambahkan ke versi ANSI C (1989), banyak programmer \mathbf{C} terbiasa untuk mendefinisikan jenis mereka sendiri Boolean seperti itu, karena alasan mudah dibaca. Namun, tipe enumerated yang setara dengan bilangan bulat sesuai dengan standar bahasa, sehingga identitas yang efektif antara boolean dan integer tetap berlaku bagi program C. Boolean dlm versi terbaru dari Python, benda yang ditetapkan pengguna dapat menentukan nilai kebenaran mereka dengan memberikan metode __bool__. [7]

Pada standar 1999, SQL yang ditetapkan tipe data Boolean dengan empat nilai yang mungkin: true, false, tidak diketahui atau null. Namun,

vendor dapat memilih untuk menyamakan dua nilai terakhir [8].. Karena inkonsistensi ini kebanyakan implementasi SQL (dengan pengecualian dari Postgresql) [9] menggunakan tipe data lain (seperti bit, byte, dan karakter) untuk mensimulasikan nilai Boolean.

C. KESIMPULAN

Tipe data Boolean mengandung nilai dua-negara seperti benar / salah, ya / tidak, atau on / off.

Nilai default dari Boolean adalah False. Jenis Konversi

Ketika Visual Basic mengkonversi nilai numerik ke tipe data Boolean, 0 menjadi Palsu dan semua nilai lain menjadi True. Ketika Visual Basic mengkonversi nilai Boolean untuk tipe numerik, Palsu dan Sejati menjadi 0 menjadi -1.

Bila Anda mengkonversi antara nilai-nilai boolean dan tipe data numerik, perlu diingat bahwa Framework. NET metode konversi tidak selalu menghasilkan hasil yang sama seperti kata kunci konversi Visual Basic. Hal ini karena konversi Visual Basic mempertahankan perilaku yang kompatibel dengan sebelumnya. Untuk informasi lebih lanjut, lihat Pemecahan Masalah Tipe Data. Programming Tips

- Bilangan Negatif. Boolean bukan tipe numerik dan tidak dapat mewakili nilai negatif. Dalam hal apapun, Anda tidak harus menggunakan Boolean untuk menyimpan nilai numerik.
- · Tipe Karakter. Boolean tidak memiliki tipe karakter harfiah atau pengidentifikasi tipe karakter.
- Jenis Framework. Jenis yang sesuai di NET Framework. adalah struktur System.Boolean.

D. DISKUSI

Mayang: apakah tujuan mempelajari tipe data booelan Saya: kita dapat supaya memahami nilai dua negara seperti ya/tidak benar/salah. Indah: jenis apa saja yang terdapat struktur di data boolean? Saya: Jenis Framework. Jenis sesuai **NET** yang di Framework. adalah struktur System.Boolean.

E. REFERENCE

- [1] O. M. Febriani and A. S. Putra, "Sistem Informasi Monitoring Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, vol. 13, no. 1, pp. 90–98, 2014.
- [2] A. S. Putra, "Paperplain: Execution Fundamental Create

- Application With Borland Delphi 7.0 University Of Mitra Indonesia," 2018.
- [3] A. S. Putra, "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer," 2018.
- [4] A. S. Putra, "ALIAS MANAGER USED IN DATABASE DESKTOP STUDI CASE DB DEMOS."
- [5] A. S. Putra,
 "COMPREHENSIVE SET OF
 PROFESSIONAL FOR
 DISTRIBUTE COMPUTING."
- [6] A. S. Putra, "DATA ORIENTED RECOGNITION IN BORLAND DELPHI 7.0."
- [7] A. S. Putra, "EMBARCADERO DELPHI XE 2 IN GPU-POWERED FIREMONKEY APPLICATION."
- [8] A. S. Putra, "HAK ATAS KEKAYAAN INTELEKTUAL DALAM DUNIA TEKNOLOGY BERBASIS REVOLUSI INDUSTRI 4.0."
- [9] A. S. Putra, "IMPLEMENTASI PERATURAN PERUNDANGAN UU. NO 31 TAHUN 2000 TENTANG DESAIN INDUSTRI BERBASIS INFORMATION TECHNOLOGY."
- [10] A. S. Putra,
 "IMPLEMENTATION OF
 PARADOX DBASE."
- [11] A. S. Putra,
 "IMPLEMENTATION OF
 TRADE SECRET CASE
 STUDY SAMSUNG MOBILE
 PHONE."
- [12] A. S. Putra,
 "IMPLEMENTATION
 PATENT FOR APPLICATION
 WEB BASED CASE STUDI



- WWW. PUBLIKLAMPUNG. COM."
- [13] A. S. Putra,
 "IMPLEMENTATION
 SYSTEM FIRST TO INVENT
 IN DIGITALLY INDUSTRY."
 - [14] A. S. Putra, "MANUAL REPORT & INTEGRATED DEVELOPMENT ENVIRONMENT BORLAND DELPHI 7.0."
- [15] A. S. Putra, "PATENT AS RELEVAN SUPPORT RESEARCH."
- [16] A. S. Putra, "PATENT FOR RESEARCH STUDY CASE OF APPLE. Inc."
- [17] A. S. Putra, "PATENT PROTECTION FOR APPLICATION INVENT."
- [18] A. S. Putra, "QUICK REPORT IN PROPERTY PROGRAMMING."
- [19] A. S. Putra, "REVIEW CIRCUIT LAYOUT COMPONENT REQUIREMENT ON ASUS NOTEBOOK."
- [20] A. S. Putra, "REVIEW TRADEMARK PATENT FOR INDUSTRIAL TECHNOLOGY BASED 4.0."
- [21] A. S. Putra, "TOOLBAR COMPONENT PALLETTE IN OBJECT ORIENTED PROGRAMMING."
- [22] A. S. Putra, "WORKING DIRECTORY SET FOR PARADOX 7."
- [23] A. S. Putra, "ZQUERY CONNECTION IMPLEMENTED PROGRAMMING STUDI CASE PT. BANK BCA Tbk."
- [24] A. S. Putra, D. R. Aryanti, and

- I. Hartati, "Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya)," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 85–97.
- A. S. Putra and O. M. Febriani, [25] "Knowledge Management Online Application in PDAM Lampung Province," **Prosiding International** conference Information on *Technology* and **Business** (ICITB), 2018, pp. 181–187.
- [26] A. S. Putra, O. M. Febriani, and B. Bachry, "Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung," *SIMADA (Jurnal Sist. Inf. dan Manaj. Basis Data)*, vol. 1, no. 1, pp. 21–30, 2018.
- [27] A. S. Putra, H. Sukri, and K. Zuhri, "Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, pp. 221–232.
- [28] D. P. Sari, O. M. Febriani, and A. S. Putra, "Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya," in *Prosiding Seminar Nasional Darmajaya*, 2018, vol. 1, no. 1, pp. 289–294.