TREE

CICI HANDISHA 185100019

Fakultas Komputer cicihandisha.student@umitra.ac.id

Abstract

Tree merupakan salah satu bentuk struktur data tidak linear yang menggambarkan hubungan yang bersifat hirarkis (hubungan one to many) antara elemenelemen. Tree bisa didefinisikan sebagai kumpulan simpul/node dengan satu elemen khusus yangdisebut Root dan node lainnya terbagi menjadi himpunan-himpeunan yang saling tak berhubungan satu sama lainnya (disebut subtree).

Tree juga adalah suatu graph yangacyclic, simple, connected yang tidak mengandung loop. Sebuah binary search tree (bst) adalah sebuah pohon biner yang boleh kosong, dan setiap nodenya harus memiliki identifier/value. value pada semua node subpohon sebelah kiri adalah selalu lebih kecil dari value dari root, sedangkan value subpohon disebelah kanan adalah sama atau lebih besar dari value pada root, masing—masing subpohon tersebut (kiri&kanan) itu sendiri adalah juga bst.

Struktur data bst sangat penting dalam struktur pencarian, misalkan, dalam kasus pencarian dalam sebuah list, jika list sudah dalam keadaan terurut maka proses pencarian akan sangat cepat, jika kita menggunanan list contigue dan melakukan pencarian biner. akan tetapi, jika kita ingin melakukan perubahan isi list (insert ataudelete), menggunakan list contigue akan sangat lambat, karena proses insert dan delete dalam list contigue butuh memindahkan banyak elemen setiap saat. mungkin kita bisa juga menggunakan linked-list, yang untuk operasi insert atau delete tinggal mengatur—atur pointer, akan tetapi pada n-linked list, kita tidak bisa melakukan pointer sembarangan setiap saat, kecuali hanya satu kali dengan kata lain hanya secara sequential

Kata Kunci: Tree

A. PENDAHULUAN

Struktur Data Tree

Dalam bab ini kita akan mempelajari satu bentuk struktur data tak linier yang mempunyai sifat-sifat khusus, yang dinamakan pohon (tree). Struktur ini biasanya digunakan untuk menggambarkan hubungan yang bersifat hirarkis antara elemenelemen yang ada. Silsilah keluarga, hasil pertandingan yang berbentuk turnamen, atau struktur organisasi dari sebuah perusahaan adalah contoh dalam organisasi tree. Dalam pemrograman, sebuah pohon terdiri dari elemen-elemen yang dinamakan node(simpul) yang mana hubungan antar simpul bersifat hirarki. Simpul yang paling atas dari hirarki dinamakan root. Simpul yang berada di bawah root secara langsung, dinamakan anak dari root, yang mana biasanya juga mempunyai anak di bawahnya. Sehingga biasa disimpulkan, kecuali root, masing-masing simpul dalam hirarki mempunyai satu induk (parent).

Jumlah anak sebuah simpul induk sangat bergantung pada jenis dari pohon. Jumlah anak dari simpul induk ini dinamakan faktor percabangan. Pada bab ini pembahasan difokuskan pada binary tree, yaitu pohon yang mempunyai factor percabangan.

Istilah-Istilah Dasar

Simpul juga mempunyai sibling, descendants, dan ancestors. Sibling dari sebuah simpul adalah anak lain dari induk simpul tersebut. Descendants dari sebuah simpul adalah semua simpul-simpul merupakan cabang (berada di bawah) simpul tersebut. Anchestors dari sebuah simpul adalah semua simpul yang berada di atas antara simpul tersebut dengan root.

Penampilan dari sebuah tree akan ditampilkan dengan berat dari tree tersebut, angka yang menunjukkan jumlah level yang ada di dalamnya.

Tingkat suatu simpul ditentukan dengan pertama kali menentukan akar sebagai bertingkat 1. Jika suatu simpul dinyatakan sebagai tingkat N, maka simpul-simpul yang merupakan anaknya akan berada pada tingkatan N+1.

Tinggi atau kedalaman dari suatu pohon adalah tingkat maksimum dari simpul dalam pohon dikurangi dengan 1. Selain tingkat, juga dikenal istilah derajad (degree) dari suatu simpul. Derajad suatu simpul dinyatakan sebagai banyaknya anak atau turunan dari simpul tersebut

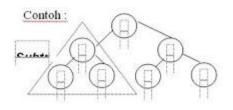
B. PEMBAHASAN / STUDI KASUS

Dalam materi Struktur data kita sudah belajar mengenai Lingked list seperti Stuck, Queue dimana struktur data tersebut termasuk dalam data yang bersifat linier. Sedangkan TREE adalah sebaliknya, agar lebih paham berikut penjelasaanya.

Tree merupakan salah satu bentuk struktur data tidak linear yang menggambarkan hubungan yang bersifat hirarkis (hubungan one to many) antara elemen-elemen. Tree bisa didefinisikan sebagai kumpulan simpul/node dengan satu elemen khusus yang disebut Root dan node lainnya terbagi menjadi himpunan-himpunan yang saling tak berhubungan satu sama lainnya (disebut subtree). Untuk jelasnya, di bawah akan diuraikan istilah-istilah umum dalam tree:

- a) Prodecessor: node yang berada diatas node tertentu.
- b) Successor: node yang berada di bawah node tertentu.
- c) Ancestor: seluruh node yang terletak sebelum node tertentu dan terletak pada jalur yang sama.
- d) Descendant : seluruh node yang terletak sesudah node tertentu dan terletak pada jalur yang sama.
- e) Parent: predecssor satu level di atas suatu node.
- f) Child: successor satu level di bawah suatu node.
- g) Sibling: node-node yang memiliki parent yang sama dengan suatu node.
- h) Subtree: bagian dari tree yang berupa suatu node beserta descendantnya dan memiliki semua karakteristik dari tree tersebut.
- i) Size: banyaknya node dalam suatu tree.

- j) Height: banyaknya tingkatan/level dalam suatu tree.
- k) Root: satu-satunya node khusus dalam tree yang tak punya predecssor.
- l) Leaf: node-node dalam tree yang tak memiliki seccessor.
- m) Degree: banyaknya child yang dimiliki suatu node.

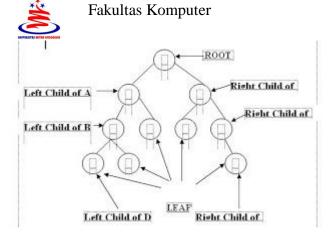


Ascestor (F)	= C.A	Y
Descendant (C) = F.G	ŀ
Parent (D)	= B	ŀ
Child (A)	$= B_{\bullet}C$	ŀ
Sibling (F)	= G	Ī
Size	= 7	1
Height	= 3	ľ
Root	= A	L
Leaf	= D.E.F.G	ŀ
Degree (C)	= 2	J

Beberapa jenis Tree yang memiliki sifat khusus:

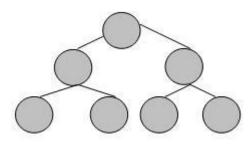
1) Binary Tree

Binary Tree adalah tree dengan syarat bahwa tiap node hanya boleh memiliki maksimal dua subtree dan kedua subtree tersebut harus terpisah. Sesuai dengan definisi tersebut, maka tiap node dalam binary tree hanya boleh memiliki paling banyak dua child.



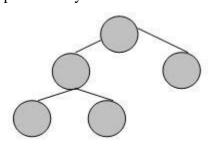
Jenis-jenis Binary Tree:

a) Full Binary Tree



Binary Tree yang tiap nodenya (kecuali leaf) memiliki dua child dan tiap subtree harus mempunyai panjang path yang sama.

b) Complete Binary Tree

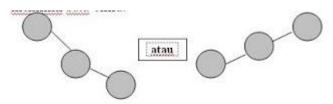


Mirip dengan Full Binary Tree, namun tiap subtree boleh memiliki panjang path yang berbed Create kecuali leaf memiliki 0 atau 2 child.



 $\square \square$ Empty :

c) Skewed Binary Tree



akni Binary Tree yang semua nodenya (kecuali leaf) hanya memiliki satu child.

Implementasi Binary Tree

Binary Tree dapat diimplemntasikan dalam Pascal dengan menggunakan double Linked List. Untuk nodenya, bisa dideklarasikan sbb:

Type Tree = node ;

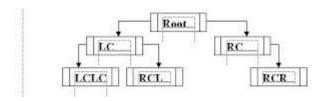
Node = record

Isi: TipeData;

Left, Right: Tree;

end;

Contoh ilustrasi Tree yang disusun dengan double linked list :



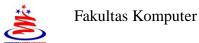
(Ket: LC=Left Child; RC=Right Child)

Operasi-operasi pada Binary Tree:

Membentuk binary tree baru yang masih kosong.

Mengosongkan binary tree yang sudah ada.

Function untuk memeriksa apakah binary tree masih kosong.



Insert: Memasukkan sebuah node ke dalam tree. Ada tiga pilihan insert: sebagai root, left child, atau right child. Khusus insert sebagai root, tree harus dalam keadaan kosong.

Find: Mencari root, parent, left child, atau right child dari suatu node. (Tree tak boleh kosong)

Update: Mengubah isi dari node yang ditunjuk oleh pointer current. (Tree tidak boleh kosong)

Retrieve: Mengetahui isi dari node yang ditunjuk pointer current. (Tree tidak boleh kosong)

DeleteSub: Menghapus sebuah subtree (node beserta seluruh descendantnya) yang ditunjuk current. Tree tak boleh kosong. Setelah itu pointer current akan berpindah ke parent dari node yang dihapus.

Characteristic: Mengetahui karakteristik dari suatu tree, yakni: size, height, serta average lengthnya. Tree tidak boleh kosong. (Average Length = [jumlahNodeLvl1*1+jmlNodeLvl2 *2+...+jmlNodeLvln*n]/Size)

Traverse: Mengunjungi seluruh node-node pada tree, masing-masing sekali.
Hasilnya adalah urutan informasi secara linier yang tersimpan dalam tree. Adatiga cara traverse: Pre Order, In Order, dan Post Order.

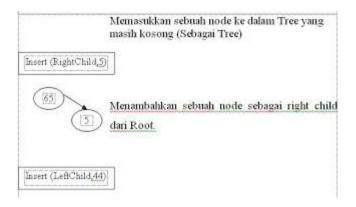
Langkah-Langkahnya Traverse:

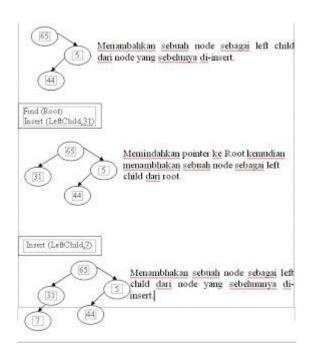
PreOrder : Cetak isi node yang dikunjungi, kunjungi Left Child, kunjungi Right Child.

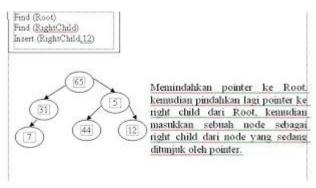
InOrder: Kunjungi Left Child, Cetak isi node yang dikunjungi, kunjungi Right Child.

PostOrder: Kunjungi Left Child, Kunjungi Right Child, cetak isi node yang dikunjungi.

Untuk lebih jelasnya perhatikan contoh operasioperasi pada Binary Tree berikut ini :



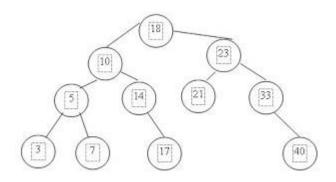




2) Binary search Tree

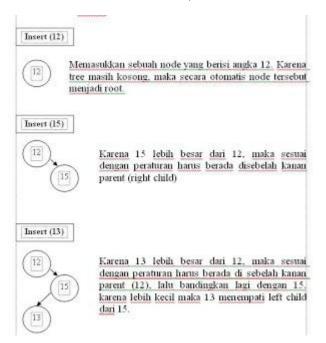
Adalah Binary Tree dengan sifat bahwa semua left child harus lebih kecil daripada right child dan parentnya. Juga semua right child harus lebih besar dari left child serta parentnya. Binary seach tree dibuat untuk mengatasi kelemahan pada binary tree

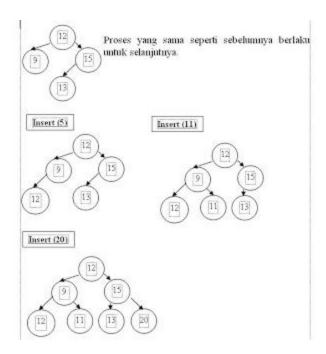
biasa, yaitu kesulitan dalam searching / pencarian node tertentu dalam binary tree. Contoh binary search tree umum:



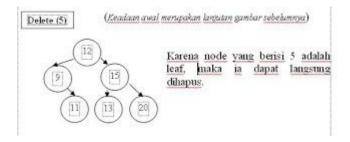
Pada dasarnya operasi dalam binary search tree sama dengan Binary tree biasa, kecuali pada operasi insert, update, dan delete.

1. Insert: Pada Binary Search Tree, dilakukan setelah ditemukan lokasi yang tepat. (Lokasi tidak ditentukan oleh user sendiri).



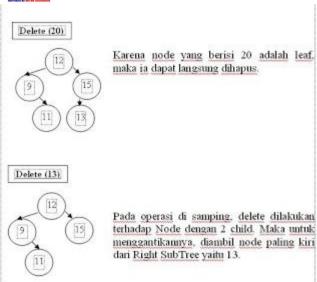


- 2. Update: Seperti pada Binary Tree biasa, namun disini uapte akan berpengaruh pada posisi node Bila tersebut selanjutnya. setelah diupdate mengakibatkan tree tersebut bukan Binary Search Tree lagi, maka harus dilakukan perubahan pada tree dengan melakukan perubahan pada tree dengan melakukan rotasi supaya tetap menjadi Binary Search Tree.
- 3. Delete: Seperti halnya update, delete dalam Binary Search Tree juga turut mempengaruhi struktur dari tree tersebut.

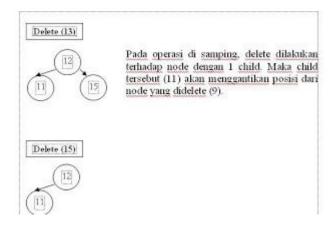


(Keadaan awal merupakan lanjutan gambar sebelumnya)





Pada operasi di samping, delete dilakukan terhadap Node dengan 2 child. Maka untuk menggantikannya, diambil node paling kiri dari Right SubTree yaitu 13.



C. **ID SECURITY** QWTD4452377-ASP-5244107

D. DISKUSI

Apakah artikel ini layak untuk dipulikasi?

BAGAS: Layak, karena menurut saya artikel ini membantu saya untuk memahami tentang struktur data tree

Apakah kelebihan artikel ini?

DWIMIFTAH: Kelibihan artikel ini adalah

mudah dipahami dengan kata-kata yang ringkas dan simple membuat saya lebih mudah dipahami.

E. KESIMPULAN

Tree merupakan salah satu bentuk struktur data tidak linear yang menggambarkan hubungan yang bersifat hirarkis (hubungan one to many) antara elemen-elemen. Tree bisa didefinisikan kumpulan sebagai elemen simpul/node dengan satu khusus yang disebut Root dan node lainnya. Tree juga adalah suatu graph yang acyclic, simple, connected yang tidak mengandung loop.



F. 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."
- A. S. Putra, D. R. Aryanti, and [24] "Metode I. Hartati, **SAW** (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya)," in Prosiding Seminar Nasional Darmaiava, 2018, vol. 1, no. 1, pp. 85–97.
- A. S. Putra and O. M. Febriani, [25] "Knowledge Management Online Application in PDAM Province," Lampung Prosiding **International** conference on *Information 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.