

LAPORAN PRAKTIKUM
MATA KULIAH TEORI ALGORITMA STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN 14 : TREE



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

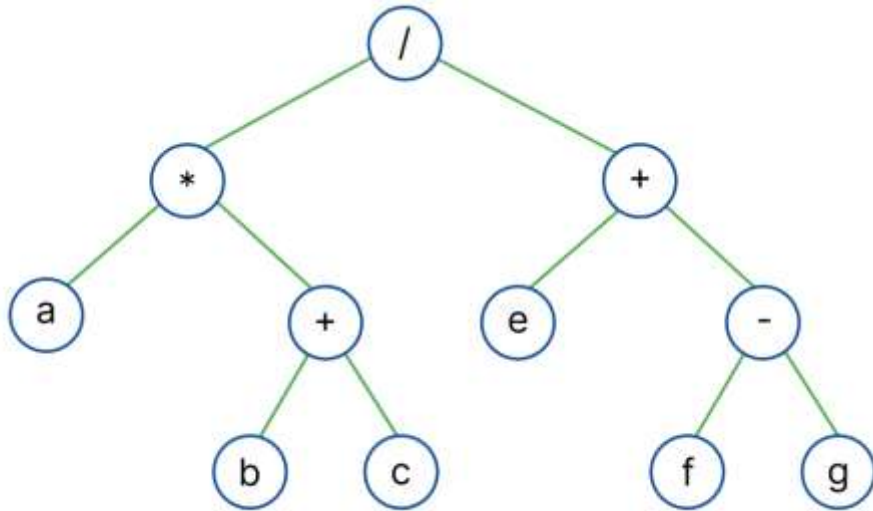
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

LATIHAN 1

Buatlah binary tree dari expresi aritmatik berikut :

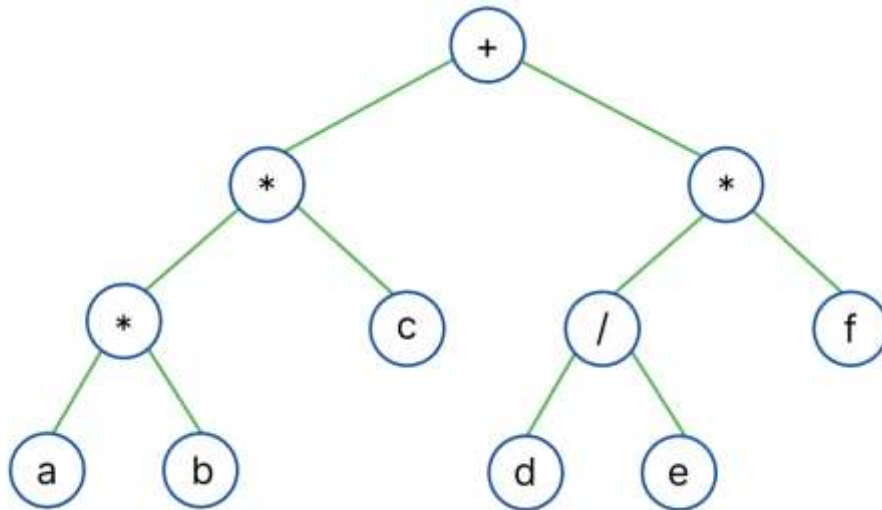
1. $a * (b + c) / (e + (f - g))$

Jawab :



2. $((a * b) * c) + (d / e) * f$

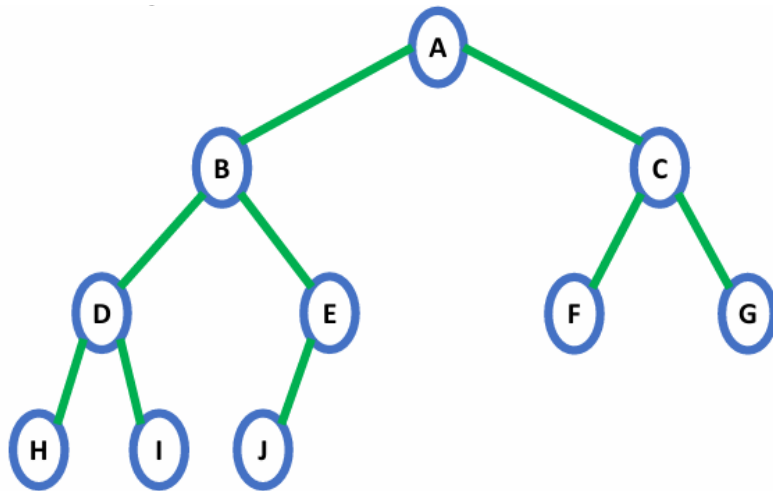
Jawab :



LATIHAN 2

Representasikan tree berikut dengan ilustrasi array dan linked list :

1.



Jawab :

a) Array :

1. Asumsi root dimulai dari indeks-0 :

- Anak kiri dari node i berada pada indeks : $2*i+1$
- Anak kanan dari node i berada pada indeks : $2*i+2$

A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9

Berikut adalah prosesnya :

- 1) $B = 2*0+1 = 1$
- 2) $C = 2*0+2 = 2$
- 3) $D = 2*1+1 = 3$
- 4) $E = 2*1+2 = 4$
- 5) $F = 2*2+1 = 5$
- 6) $G = 2*2+2 = 6$
- 7) $H = 2*3+1 = 7$
- 8) $I = 2*3+2 = 8$
- 9) $J = 2*4+1 = 9$

2. Asumsi root dimulai dari indeks ke-1 :

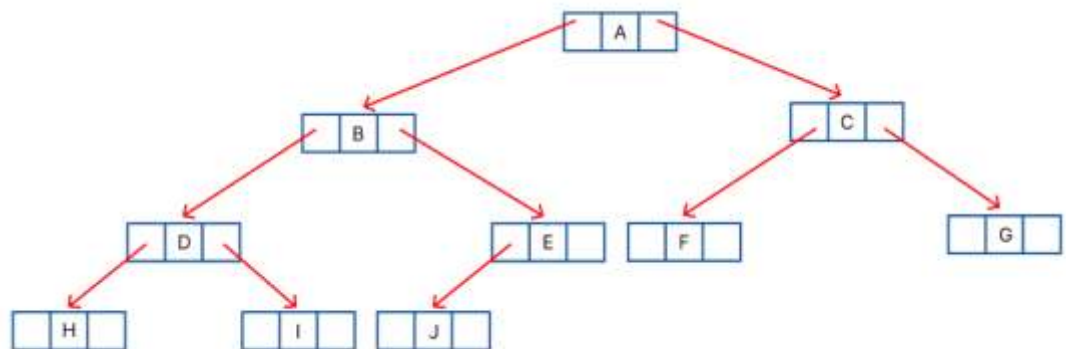
- Anak kiri dari node i berada pada indeks : $2*i$
- Anak kanan dari node i berada pada indeks : $2*i+1$

	A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9	10

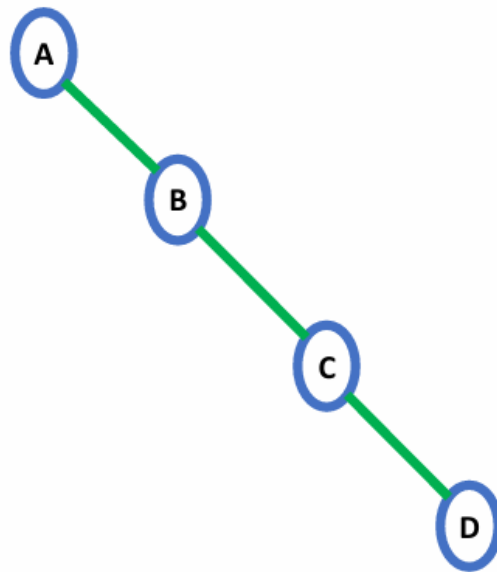
Berikut adalah prosesnya :

- 1) $B = 2 * 1 = 2$
- 2) $C = 2 * 1 + 1 = 3$
- 3) $D = 2 * 2 = 4$
- 4) $E = 2 * 2 + 1 = 5$
- 5) $F = 2 * 3 = 6$
- 6) $G = 2 * 3 + 1 = 7$
- 7) $H = 2 * 4 = 8$
- 8) $I = 2 * 4 + 1 = 9$
- 9) $J = 2 * 5 = 10$

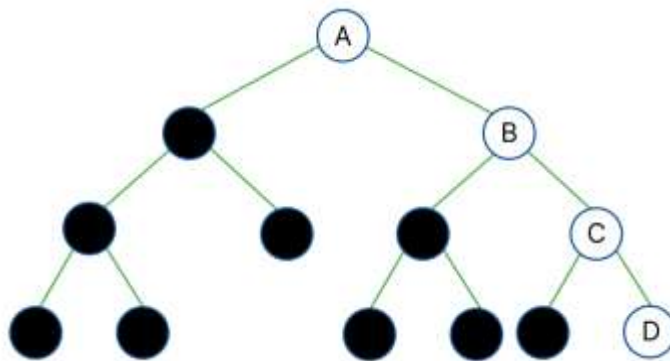
b) Linked List : Masing-masing node terdiri dari 3 bagian, yaitu pointer kiri, data, dan pointer kanan



2.



Jawab :



a) Array :

1. Asumsi root dimulai dari indeks-0 :

- Anak kiri dari node i berada pada indeks : $2*i+1$
- Anak kanan dari node i berada pada indeks : $2*i+2$

A		B				C						D
0	1	2	3	4	5	6	7	8	9	10	11	12

Berikut adalah prosesnya :

- 1) $B = 2*0+2 = 2$
- 2) $C = 2*2+2 = 6$
- 3) $D = 2*5+2 = 12$

2. Asumsi root dimulai dari indeks ke-1 :

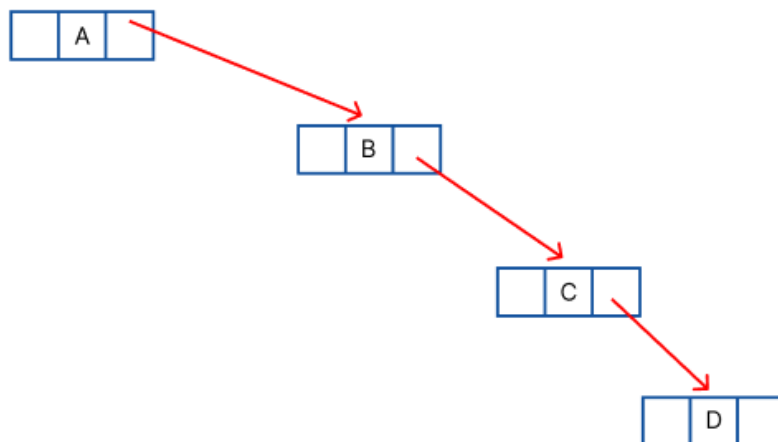
- Anak kiri dari node i berada pada indeks : $2*i$
- Anak kanan dari node i berada pada indeks : $2*i+1$

	A		B				C						D
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Berikut adalah prosesnya :

- 1) $B = 2 \cdot 1 + 1 = 3$
- 2) $C = 2 \cdot 3 + 1 = 5$
- 3) $D = 2 \cdot 6 + 1 = 13$

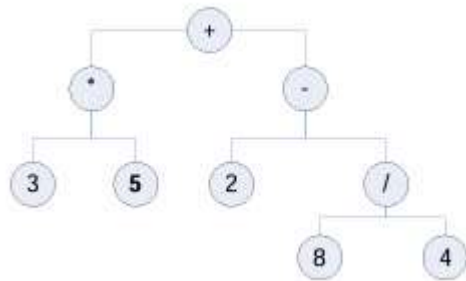
b) **Linked List** : Masing-masing node terdiri dari 3 bagian, yaitu pointer kiri, data, dan pointer kanan



LATIHAN 3

Telusuri pohon biner berikut dengan menggunakan metode preorder, inorder, postorder, dan level order traversal.

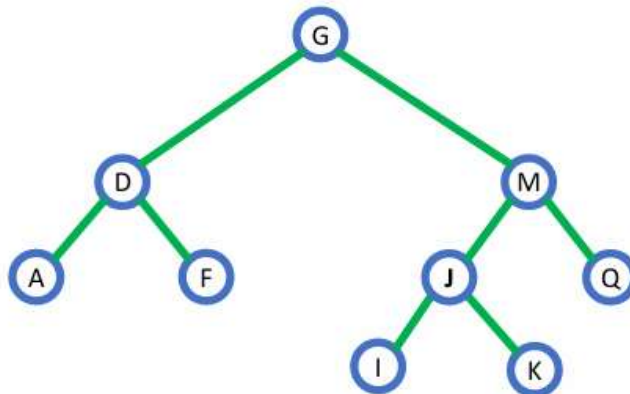
1.



Jawab :

- 1) **Preorder : root – left child – right chil**
: +, *, 3, 5, -, 2, /, 8, 4
- 2) **Inorder : left child – root – right child**
: 3, *, 5, +, 2, -, 8, /, 4
- 3) **Postorder : left child – right child – root**
: 3, 5, *, 2, 8, 4, /, -, +
- 4) **Level Order**
: +, *, -, 3, 5, 2, /, 8, 4

2.

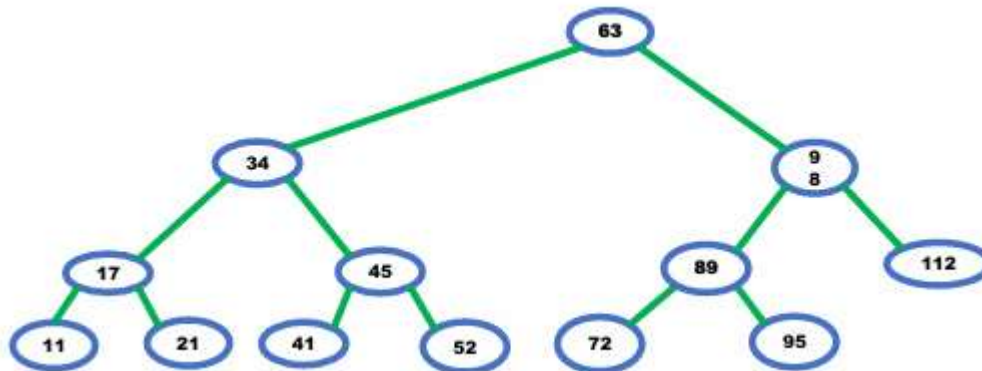


Jawab :

- 1) **Preorder : root – left child – right chil**
: G, D, A, F, M, J, I, K, Q
- 2) **Inorder : left child – root – right child**
: A, D, F, G, I, J, K, M, Q
- 3) **Postorder : left child – right child – root**
: A, F, D, I, K, J, Q, M, G
- 4) **Level Order**
: G, D, M, A, F, J, Q, I, K

LATIHAN 4

Terdapat sebuah tree seperti gambar disamping.



Terdapat data baru (40) yang akan ditambahkan dan data lama (98) yang akan dihapus. Ilustrasikan operasi (find, insert, delete, display) yang akan dilakukan untuk mengatasi penambahan dan penghapusan data tersebut!

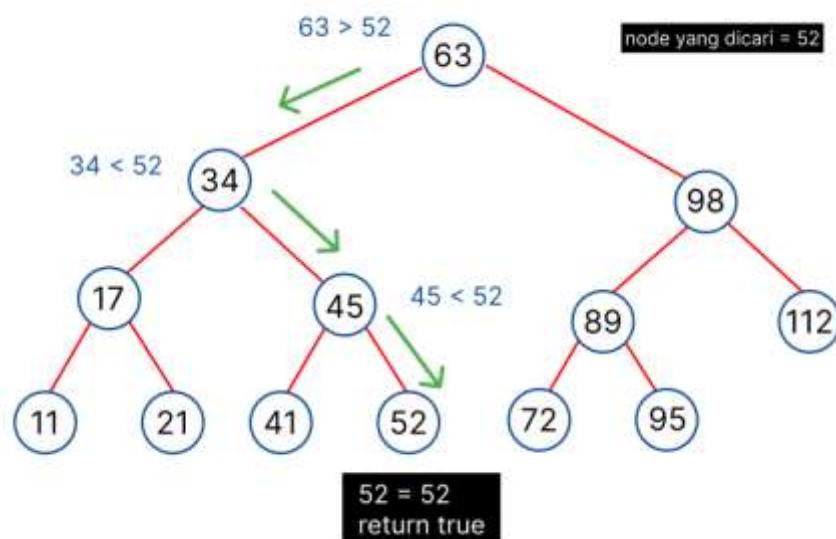
Jawab :

1. Mencari data 52 (find)

Langkah – langkahnya :

- Dimulai dari root dan bandingkan root dengan 52
- Jika root > 52, dilakukan pencarian pada subtree kiri
- Jika root < 52, dilakukan pencarian pada subtree kanan
- Ulangi langkah b dan c hingga 52 ketemu, dan return true
- Return false, jika sampai pada leaf tidak ketemu

Berikut adalah ilustrasi nya :

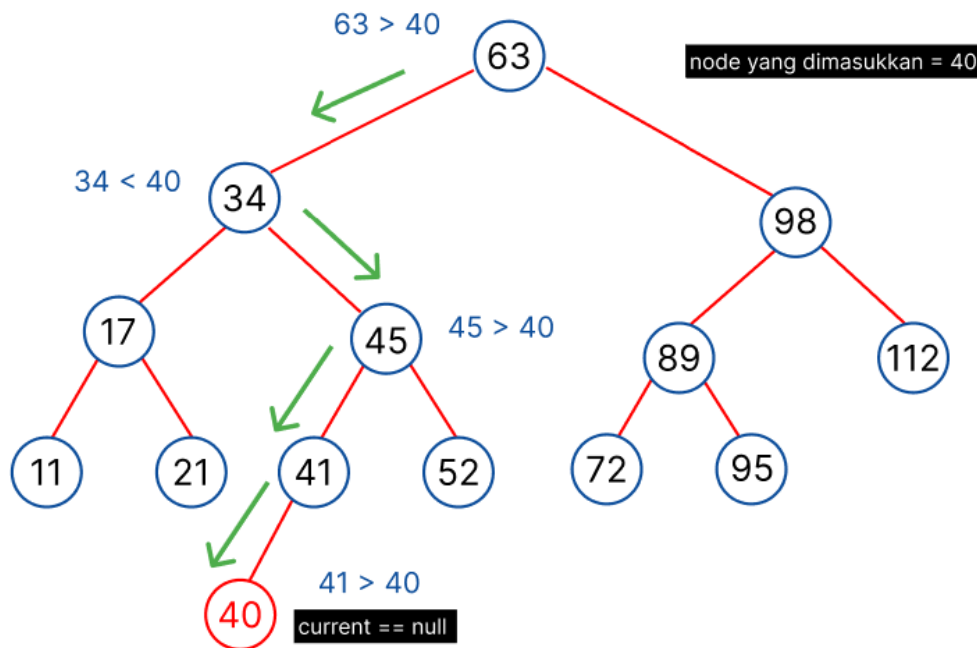


2. Menambahkan data 40 (insert)

Langkah – langkahnya :

- Lakukan operasi find dulu untuk mencari tempat meletakkan node 40.
- Membuat variable current bertipe node, set current dengan root. Current yaitu node yang sedang ditelusuri
- Jika $\text{current} > 40$, maka pencarian di sebelah kiri root
- Jika $\text{current} < 40$, maka pencarian di sebelah kanan root
- Jika ketemu, nilai $\text{current} = \text{null}$. Letakkan node yang baru berisikan 40 di posisi current

Berikut adalah ilustrasinya :

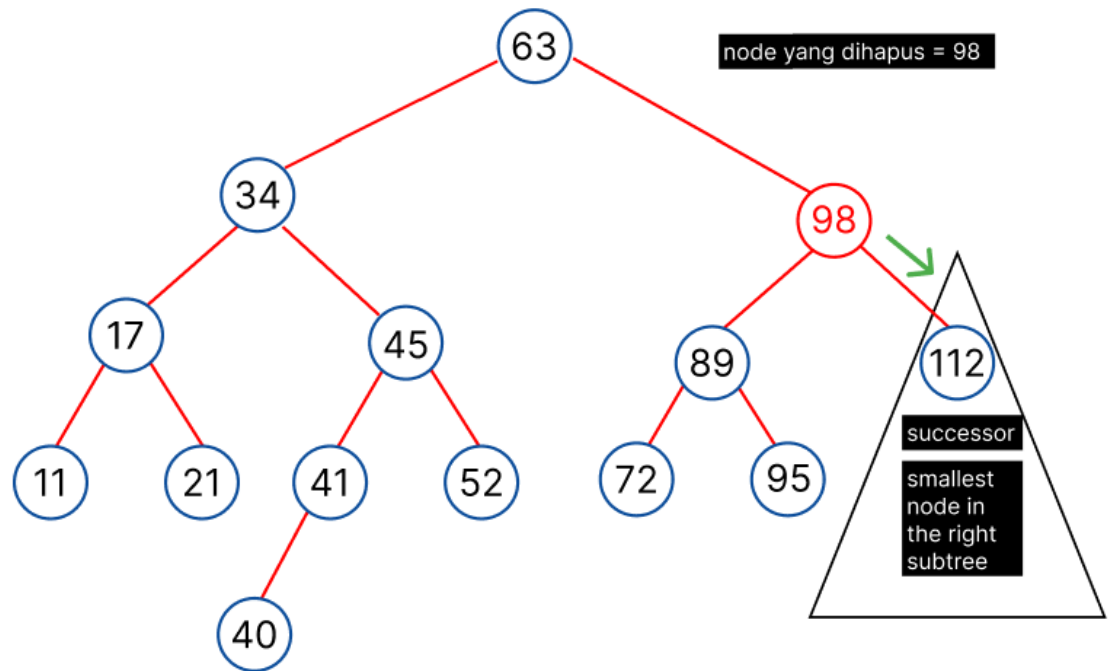


3. Menghapus data 98 (delete)

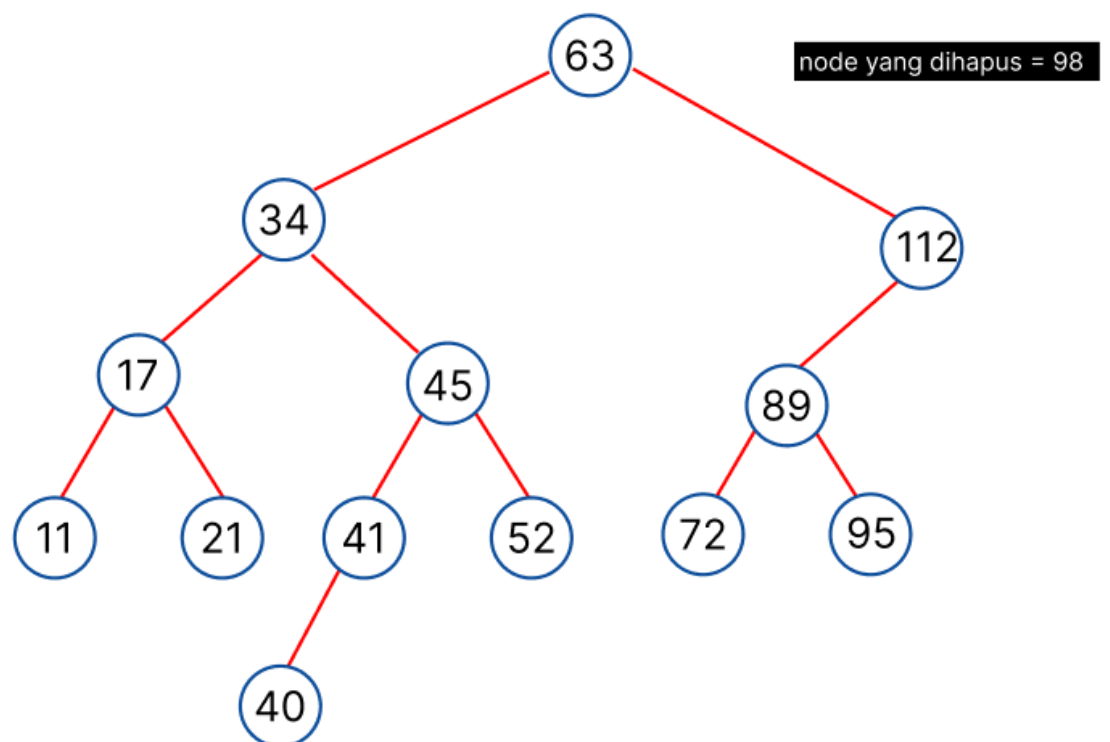
Skenario delete yang dilakukan yaitu pada node yang memiliki 2 child. Langkah – langkahnya :

- Lakukan traverse menuju node yang akan dihapus
- Mencari successor dari 98, untuk menggantikan 98.
- Successor adalah node paling kecil dari subtree sebelah kanan pada node yang akan dihapus.

Berikut adalah ilustrasinya :



Maka menjadi :



4. Display

- Preorder : 63, 34, 17, 11, 21, 45, 41, 40, 52, 112, 89, 72, 95
- Inorder : 11, 17, 21, 34, 40, 41, 45, 52, 63, 72, 89, 95, 112
- Postorder : 11, 21, 17, 40, 41, 52, 45, 34, 72, 95, 89, 112, 63
- Level Order : 63, 34, 112, 17, 45, 89, 11, 21, 41, 52, 72, 95, 40