1. **Melengkapi tree.cpp operasi insert dan delete berjalan dengan benar.**

Algoritma insert ditunjukkan pada gambar berikut.

Algorithm addBST (root, newNode){

|  |
| --- |
| void insert (node \*\*root, node \*newnode){  if (isEmpty(\*root)) {  \*root = newnode;  } else {  if (newnode->data < (\*root)->data) {  insert (&(\*root)->left, newnode);  } else if (newnode->data > (\*root)->data) {  insert (&(\*root)->right, newnode);  } else {  cout << "Data sudah ditambahkan..." << endl;  }  }  } |

Algoritma DeleteBST(root, dltKey)

|  |
| --- |
| bool deleteNode (node \*\*root, int key){  if (isEmpty(\*root)) {  return false;  } else if (key < (\*root)->data) {  deleteNode (&(\*root)->left, key);  } else if (key > (\*root)->data) {  deleteNode (&(\*root)->right, key);  } else {  if (isEmpty((\*root)->left)) {  (\*root) = (\*root)->right;  return true;  } else if (isEmpty((\*root)->right)) {  (\*root) = (\*root)->left;  } else {  node \*hapus=\*root, \*terbesar=findLargestBST((\*root)->left);  hapus->data=terbesar->data;  return deleteNode((&(hapus->left)),terbesar->data);  }  }  }  } |

1. **Menggambarkan perubahan pada tree ketika data data berikut ditambahkan ke tree : 2, 7, 1, 89, 5, 23, 1.**
2. **Gambarkan perubahan pada tree ketika data-data berikut dari tree : 89, 5, 23. (Dihapus)**