**LAPORAN PRAKTIKUM**

**STRUKTUR DATA**

**PERTEMUAN KE – 13**

**Disusun Oleh :**

**NAMA : TARISA DWI SEPTIA**

**NIM : 205410126**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

**UNIVERSITAS TEKNOLOGI DIGITAL INDONSIA**

**YOGYAKARTA**

**2020**

**POHON BINER**

1. **Tujuan**

* Mahasiswa dapat mengimplementasikan penggunaan Simpul milik Double Linked List untuk pembuatan pohon

1. **Praktik**
2. Tuliskan program berikut ini menggunakan TextPad

|  |
| --- |
| class simpul{  String elemen;  simpul kiri;  simpul kanan;  }  class pohon{  public static simpul akar;  public static void deklarasiPohon(){  akar = null;  }  public static simpul tambahSimpul(simpul Penunjuk, String ELEMEN){  if (Penunjuk == null){  simpul baru = new simpul();  baru.elemen = ELEMEN;  baru.kiri = null;  baru.kanan = null;  Penunjuk = baru;  return(Penunjuk);  }else{  if (ELEMEN.compareTo(Penunjuk.elemen) < 0 ){  Penunjuk.kiri = tambahSimpul(Penunjuk.kiri, ELEMEN);  return(Penunjuk);  }else{  Penunjuk.kanan= tambahSimpul(Penunjuk.kanan, ELEMEN);  return(Penunjuk);  }  }  }  public static void preOrder(simpul Penunjuk){  if(Penunjuk != null){  System.out.print(Penunjuk.elemen + ",");  preOrder(Penunjuk.kiri);  preOrder(Penunjuk.kanan);  }  }  public static void inOrder(simpul Penunjuk){  if(Penunjuk != null){  inOrder(Penunjuk.kiri);  System.out.print(Penunjuk.elemen + ",");  inOrder(Penunjuk.kanan);  }  }  public static void postOrder(simpul Penunjuk){  if(Penunjuk != null){  postOrder(Penunjuk.kiri);  postOrder(Penunjuk.kanan);  System.out.print(Penunjuk.elemen + ",");  }  }  public static void main(String[] args){  deklarasiPohon();  akar =tambahSimpul(akar,"M");  akar =tambahSimpul(akar,"P");  akar =tambahSimpul(akar,"D");  akar =tambahSimpul(akar,"A");  akar =tambahSimpul(akar,"S");  akar =tambahSimpul(akar,"K");  akar =tambahSimpul(akar,"N");  akar =tambahSimpul(akar,"G");  akar =tambahSimpul(akar,"O");  akar =tambahSimpul(akar,"L");  akar =tambahSimpul(akar,"W");  akar =tambahSimpul(akar,"F");  akar =tambahSimpul(akar,"J");  akar =tambahSimpul(akar,"T");  akar =tambahSimpul(akar,"H");  akar =tambahSimpul(akar,"U");  preOrder(akar);  }  } |

Output :

1. Lakukan juga eksperimen yang sama untuk cetak inOrder dan cetak postOrder berikut :

|  |
| --- |
| public static void main(String[] args){  deklarasiPohon();  akar =tambahSimpul(akar,"M");  akar =tambahSimpul(akar,"P");  akar =tambahSimpul(akar,"D");  akar =tambahSimpul(akar,"A");  akar =tambahSimpul(akar,"S");  akar =tambahSimpul(akar,"K");  akar =tambahSimpul(akar,"N");  akar =tambahSimpul(akar,"G");  akar =tambahSimpul(akar,"O");  akar =tambahSimpul(akar,"L");  akar =tambahSimpul(akar,"W");  akar =tambahSimpul(akar,"F");  akar =tambahSimpul(akar,"J");  akar =tambahSimpul(akar,"T");  akar =tambahSimpul(akar,"H");  akar =tambahSimpul(akar,"U");  // preOrder(akar);  **inOrder(akar);**  } |

Kemudian eksekusilah program di atas. Hasil apakah yang didapat? Catatlah dalam laporan anda dan jelaskan mengapa bisa demikian.



1. Lakukan hal yang sama sebagaimana pelaksanaan praktikum 2 untuk cetak postOrder.

|  |
| --- |
| public static void main(String[] args){  deklarasiPohon();  akar =tambahSimpul(akar,"M");  akar =tambahSimpul(akar,"P");  akar =tambahSimpul(akar,"D");  akar =tambahSimpul(akar,"A");  akar =tambahSimpul(akar,"S");  akar =tambahSimpul(akar,"K");  akar =tambahSimpul(akar,"N");  akar =tambahSimpul(akar,"G");  akar =tambahSimpul(akar,"O");  akar =tambahSimpul(akar,"L");  akar =tambahSimpul(akar,"W");  akar =tambahSimpul(akar,"F");  akar =tambahSimpul(akar,"J");  akar =tambahSimpul(akar,"T");  akar =tambahSimpul(akar,"H");  akar =tambahSimpul(akar,"U");  // preOrder(akar);  // inOrder(akar);  **postOrder(akar);**  } |

Kemudian eksekusilah program di atas. Hasil apakah yang didapat? Catatlah dalam laporan anda dan jelaskan mengapa bisa demikian.



1. **Tugas**

Modifikasilah program pohon biner pada Modul 13 ini dari yang semula berelemen Karakter menjadi angka. Kemudian lakukan proses cetak PreOrder, InOrder dan PosOrder terhadapnya

|  |
| --- |
| public static void main(String[] args){  deklarasiPohon();  akar =tambahSimpul(akar,"1");  akar =tambahSimpul(akar,"2");  akar =tambahSimpul(akar,"3");  akar =tambahSimpul(akar,"4");  akar =tambahSimpul(akar,"5");  akar =tambahSimpul(akar,"6");  akar =tambahSimpul(akar,"7");  akar =tambahSimpul(akar,"8");  akar =tambahSimpul(akar,"9");  akar =tambahSimpul(akar,"10");  akar =tambahSimpul(akar,"11");  akar =tambahSimpul(akar,"12");  akar =tambahSimpul(akar,"13");  akar =tambahSimpul(akar,"14");  akar =tambahSimpul(akar,"15");  akar =tambahSimpul(akar,"16");  //preOrder(akar);  //inOrder(akar);  //postOrder(akar);  } |

* preOrder(akar);



* inOrder(akar);
* postOrder(akar);



1. **Kesimpulan**

Setelah melakukan praktik diatas, dapat disimpulkan bahwa mahasiswa dapat mengimplementasikan penggunaan Simpul milik Double Linked List untuk pembuatan pohon.