A logo of a university

Description automatically generated

NAMA : RAKASYAEL HIZKIA KOLONDAM

NIM : 2023105490

MATA KULIAH : TI0091 – PRAKTIKUM DATA TERSTRUKTUR

PRAKTIKUM KE - : 1

|  |  |  |
| --- | --- | --- |
| **No.** | **Kode** | **Penjelasan** |
| 1 | public Array(int value)  {  Node newNode = new Node(value);  nodes[length] = newNode;  length++;  } | Untuk mendeklarasikan array yang dipakai. Setiap ada node baru yang dimasukkan, maka panjang array bertambah 1. |
| 2 | public void printArray()  {  System.out.print("Array: ");  for (int i = 0; nodes[i] != null; i++)  {  System.out.print(nodes[i].value + " ");  }  System.out.println();  } | Digunakkan untuk display array dengan cara membaca isi array dari kiri ke kanan satu persatu berdasarkan indeksnya. |
| 3 | public int getLength()  {  return length;  } | Mengembalikan ukuran array. Pada bagian insert, setiap ada node baru yang dimasukkan, maka panjang array bertambah. |
| 4 | public void append(int value)  {  Node newNode = new Node(value);  nodes[length] = newNode;  length++;  } | Menempelkan elemen atau node array ke belakang lalu menambah panjang array. |
| 5 | public void prepend(int value)  {  length++;  for (int i = length - 1; i > 0; i--)  {  nodes[i] = nodes[i - 1];  }  Node newNode = new Node(value);  nodes[0] = newNode;  } | Menempelkan elemen atau node array ke depan lalu menambah panjang array. Cara menambahkannya adalah dengan menggeser semua elemen ke belakang 1 indeks lalu menambahkan node baru di depan yang di isi oleh nilai tertentu |
| 6 | public Node removeLast()  {  if (length > 0)  {  nodes[length - 1] = null;  length--;  }  return null;  } | Menghapus bagian array paling belakang dengan cara membuat indeks array yang paling belakang menjadi null lalu mengurang panjang array tersebut. |
| 7 | public Node removeFirst()  {  if (length > 0)  {  Node node = nodes[0];  for (int i = 0; i < length; i++)  {  nodes[i] = nodes[i + 1];  }  nodes[length - 1] = null;  length--;  return node;  }  return null;  } | Menghapus bagian array paling depan dengan cara membuat elemen dengan indeks 0 menjadi null lalu menggeser setiap elemen dalam array ke depan 1 indeks dan mengurangi ukurannya. |
| 8 | ***Ganti sesuai dengan kode yang anda tulis***  public Node get(int *index*) {      if(index >= 0 && index <= length){        return nodes[index];      }      else{        return null;      }    } | Searching elemen array berdasarkan indeksnya dengan mengembalikan elemen yang berada di indeks yang dicari. |
| 9 | ***Ganti sesuai dengan kode yang anda tulis***  public void insert(int *index*, int *value*) {      for(int i = length; i >= 1; i--){        nodes[i] = nodes[i - 1];      }      nodes[index] = new Node(value);      length++;    } | Memasukkan elemen ke indeks tertentu dengan cara menggeser semua bagian array sesudah indeks yang ingin dimasukkan node yang baru lalu menambah panjang array nya. |
| 10 | ***Ganti sesuai dengan kode yang anda tulis***  public Node remove(int *index*) {      if(index >= 0 && index < length){        Node removedNode = nodes[index];        for(int i = index + 1; i < length; i++){          nodes[i - 1] = nodes[i];        }        nodes[length - 1] = null;        length--;        return removedNode;      }      else{        return null;      }    } | Menghapus bagian array sesuai dengan indeks yang dipilih dengan cara dibuat null terlebih dahulu lalu menggeser semua array yang ada di kanan indeks ke kiri 1 indeks lalu menghapus elemen paling akhir (karena akan tercopy demi menggantikan node kosong) lalu ukurannya di kurangi. |
| 11 | ***Ganti sesuai dengan kode yang anda tulis***  public void reverse() {      int left = 0;      int right = length - 1;        while(left < right){        Node temp = nodes[left];        nodes[left] = nodes[right];        nodes[right] = temp;        left++;        right--;      }    } | Fungsi membalikkan dengan cara menghitung dari kanan ke kiri dan swap elemen kanan dengan kiri (sesuai indeks yang di tunjuk) yang dilakukan berulang sehingga kanan = kiri atau kanan > kiri. |