1. **DEFINISI MASALAH**
2. Kembangkan program di atas dengan menambahkan method berikut:
3. Mencari posisi bilangan yang merupakan kelipatan dari x0 yang terletak diantara index x1 dan index x2 dengan header sebagai berikut:

Int findPosKelipatan (int x0, int x1, int x2)

Contoh:

Sebuah set {1,2,3,4,5,6,7} dan dengan memanggil method findPosKelipatan(3,1,6) maka akan menghasilkan 2 5 karena kelipatan dari nilai 3 adalah 3 dan 6, dimana keduanya ada diantara index 1 – 6 dan memiliki indeks 2 dan indeks 5.

1. Pengurutan dengan menggunakan algoritma pengurutan yang lain selain Selection Sort.
2. Pada latihan kedua ini anda diminta untuk melengkapi bagian dari program ADT\_Larik sehingga jika diberikan program utama pada gambar 1 akan menghasilkan keluaran sebagaimana gambar 2.
3. **SOURCE CODE**

|  |  |
| --- | --- |
| Larik.java | |
| 1 | package asd; |
| 2 | public class Larik { |
| 3 | //data (strukur data) |
| 4 | private int size; |
| 5 | private int[] itemDt; |
| 6 | //method |
| 7 | public void buatLarik(int n) { |
| 8 | this.size = n; |
| 9 | this.itemDt = new int[this.size]; |
| 10 | } |
| 11 | public Larik(int n) { |
| 12 | buatLarik(n); |
| 13 | } |
| 14 | public int getSize() { |
| 15 | return this.size; |
| 16 | } |
| 17 | public Larik(int[] dt) { |
| 18 | buatLarik(dt.length); |
| 19 | for (int i = 0; i < dt.length; i++) { |
| 20 | isiItem(i, dt[i]); |
| 21 | } |
| 22 | } |
| 23 | public void isiItem(int id, int dt) { |
| 24 | this.itemDt[id] = dt; |
| 25 | } |
| 26 | public void cetak(String komentar) { |
| 27 | System.out.println(komentar); |
| 28 | for (int i = 0; i < this.size; i++) { |
| 29 | System.out.print(this.itemDt[i] + " "); |
| 30 | } |
| 31 | System.out.println(); |
| 32 | } |
| 33 | public int findBesar() { |
| 34 | int besar = this.itemDt[0]; |
| 35 | for (int i = 1; i < this.size; i++) { |
| 36 | if (besar < this.itemDt[i]) { |
| 37 | besar = this.itemDt[i]; |
| 38 | } |
| 39 | } |
| 40 | return besar; |
| 41 | } |
| 42 | /\*\* |
| 43 | \* program ini mencari posisi suatu data tertentu di larik |
| 44 | \*/ |
| 45 | public int getPosisi(int dtCari) { |
| 46 | int pos = -99; |
| 47 | boolean ketemu = false; |
| 48 | int i = 0; |
| 49 | while (!ketemu && i < this.size) { |
| 50 | if (dtCari == this.itemDt[i]) { |
| 51 | ketemu = true; |
| 52 | pos = i; |
| 53 | } |
| 54 | i++; |
| 55 | } |
| 56 | return pos; |
| 57 | } |
| 58 | private int getPosMax(int id) { |
| 59 | int max = this.itemDt[id]; |
| 60 | int posMax = id; |
| 61 | for (int i = id + 1; i < size; i++) { |
| 62 | if (max <= this.itemDt[i]) { |
| 63 | max = this.itemDt[i]; |
| 64 | posMax = i; |
| 65 | } |
| 66 | } |
| 67 | return posMax; |
| 68 | } |
| 69 | private int getPosMin(int id) { |
| 70 | int min = this.itemDt[id]; |
| 71 | int posMin = id; |
| 72 | for (int i = id + 1; i < size; i++) { |
| 73 | if (min >= this.itemDt[i]) { |
| 74 | min = this.itemDt[i]; |
| 75 | posMin = i; |
| 76 | } |
| 77 | } |
| 78 | return posMin; |
| 79 | } |
| 80 | public int PencarianBiner(int dtCari, int awal, int akhir) { |
| 81 | int pos = -99; |
| 82 | int tengah = (awal + akhir) / 2; |
| 83 | if (dtCari < this.itemDt[tengah]) { |
| 84 | //nilai awal -99 tengah 2 |
| 85 | return PencarianBiner(dtCari, awal, tengah); |
| 86 | } else if (dtCari > this.itemDt[tengah]) { |
| 87 | //34<5 |
| 88 | return PencarianBiner(dtCari, tengah + 1, akhir); |
| 89 | } else if (dtCari == this.itemDt[tengah]) { |
| 90 | return tengah; |
| 91 | } else { |
| 92 | return pos; |
| 93 | } |
| 94 | } |
| 95 | /\*\* |
| 96 | \* program untuk mencopy isi suatu Larik mulai dari posisi k sebanyak n item |
| 97 | \* hasilnya dikeluarkan sebagai array baru |
| 98 | \* @return |
| 99 | \*/ |
| 100 | public Larik copyLarik(int k, int n) { |
| 101 | Larik lHasil = null; |
| 102 | if (n <= this.size - k) { |
| 103 | lHasil = new Larik(n); |
| 104 | int j = 0; |
| 105 | for (int i = k; i < k + n; i++) { |
| 106 | lHasil.isiItem(j++, this.itemDt[i]); |
| 107 | } |
| 108 | } |
| 109 | return lHasil; |
| 110 | } |
| 111 | /\*\* |
| 112 | \* pilihan 0 : urutkan dari kecil ke besar lainya : urutkan dari besar ke |
| 113 | \* kecil Algoritma pengurutan ini menggunakan selection sort |
| 114 | \*/ |
| 115 | public Larik SelectionSort(int pilihan) { |
| 116 | Larik lsort = copyLarik(0, size); |
| 117 | for (int i = 0; i < lsort.getSize(); i++) { |
| 118 | int posData; |
| 119 | if (pilihan == 0) { |
| 120 | posData = lsort.getPosMin(i); |
| 121 | } else { |
| 122 | posData = lsort.getPosMax(i); |
| 123 | } |
| 124 | int dt1 = lsort.itemDt[i]; |
| 125 | int dt2 = lsort.itemDt[posData]; |
| 126 | lsort.itemDt[i] = dt2; |
| 127 | lsort.itemDt[posData] = dt1; |
| 128 | } |
| 129 | return lsort; |
| 130 | } |
| 131 | public int findPosKelipatan(int cari, int awal, int akhir) { |
| 132 | Larik salin = copyLarik(0, size); |
| 133 | int search = 0, index = 0; |
| 134 | for (int i = 0; i < akhir; i++) { |
| 135 | if (salin.itemDt[i] % cari == 0) { |
| 136 | System.out.print(i + " "); |
| 137 | } |
| 138 | // } |
| 139 | // Larik Hasil = new Larik(search); |
| 140 | // for (int i = 0; i < akhir; i++) { |
| 141 | // if (salin.itemDt[i] % cari == 0) { |
| 142 | // Hasil.itemDt[index] = salin.itemDt[i]; |
| 143 | // System.out.println(i+""); |
| 144 | // } |
| 145 | // } |
| 146 | } |
| 147 | return 0; |
| 148 | } |
| 149 | public void bubbleSort() { |
| 150 | int temp = 0; |
| 151 | for (int i = 0; i < (itemDt.length - 1); i++) { |
| 152 | for (int j = 0; j < (itemDt.length - 1); j++) { |
| 153 | if (itemDt[j] < itemDt[j + 1]) { |
| 154 | temp = itemDt[j]; |
| 155 | itemDt[j] = itemDt[j + 1]; |
| 156 | itemDt[j + 1] = temp; |
| 157 | } |
| 158 | } |
| 159 | } |
| 160 | for (int i = 0; i < itemDt.length; i++) { |
| 161 | System.out.print(itemDt[i] + " "); |
| 162 | } |
| 163 | } |
| 164 | public static void MergeSort(int[] dtArray) { |
| 165 | mergeSort\_1(dtArray, 0, dtArray.length - 1); |
| 166 | for (int i = 0; i < dtArray.length; i++) { |
| 167 | System.out.print(dtArray[i] + " "); |
| 168 | } |
| 169 | } |
| 170 | public static void mergeSort\_1(int array[], int lo, int n) { |
| 171 | int low = lo; |
| 172 | int high = n; |
| 173 | if (low >= high) { |
| 174 | return; |
| 175 | } |
| 176 | int middle = (low + high) / 2; |
| 177 | mergeSort\_1(array, low, middle); |
| 178 | mergeSort\_1(array, middle + 1, high); |
| 179 | int end\_low = middle; |
| 180 | int start\_high = middle + 1; |
| 181 | while ((lo <= end\_low) && (start\_high <= high)) { |
| 182 | if (array[low] < array[start\_high]) { |
| 183 | low++; |
| 184 | } else { |
| 185 | int Temp = array[start\_high]; |
| 186 | for (int k = start\_high - 1; k >= low; k--) { |
| 187 | array[k + 1] = array[k]; |
| 188 | } |
| 189 | array[low] = Temp; |
| 190 | low++; |
| 191 | end\_low++; |
| 192 | start\_high++; |
| 193 | } |
| 194 | } |
| 195 | } |
| 196 | public static void main(String[] args) { |
| 197 | int[] A = {1, 2, 3, 4, 5, 6, 7}; |
| 198 | Larik lA = new Larik(A); |
| 199 | lA.cetak("Sebelum"); |
| 200 | lA.SelectionSort(0).cetak("Sesudah di sort"); |
| 201 | // int k = lA.PencarianBiner(34,0,5); |
| 202 | // 34 adalah data ug dicari, 0 adalah indeks awal, |
| 203 | // 5 adalah akhir |
| 204 | // System.out.println(k); |
| 205 | Larik lB = lA.SelectionSort(0); |
| 206 | // int p = lB.PencarianBiner(3, 0, 5); |
| 207 | // System.out.println(p); |
| 208 | System.out.println("Kelipatan terdapat pada index: "); |
| 209 | lB.findPosKelipatan(3, 1, 6); |
| 210 | System.out.println(""); |
| 211 | System.out.println("Hasil bubble sort: "); |
| 212 | lB.bubbleSort(); |
| 213 | System.out.println(""); |
| 214 | } |
| 215 | int getItem(int i) { |
| 216 | return this.getItem(i); |
| 217 | } |
| 218 | int getPosKecil(int i, int n) { |
| 219 | int kecil = this.itemDt[0]; |
| 220 | for (i = 1; i > this.size; i++) { |
| 221 | if (kecil > this.itemDt[i]) { |
| 222 | kecil = this.itemDt[i]; |
| 223 | } |
| 224 | } |
| 225 | return kecil; |
| 226 | } |
| 227 | int getPosBesar(int i, int n) { |
| 228 | int besar = this.itemDt[0]; |
| 229 | for (i = 1; i < this.size; i++) { |
| 230 | if (besar < this.itemDt[i]) { |
| 231 | besar = this.itemDt[i]; |
| 232 | } |
| 233 | } |
| 234 | return besar; |
| 235 | } |
| 236 | } |

|  |  |
| --- | --- |
| Larik2.java | |
| 1 | public class Larik2 { |
| 2 | private int size; |
| 3 | private double[] itemDt; |
| 4 | public Larik2(int n) { |
| 5 | buatLarik(n); |
| 6 | } |
| 7 | public void buatLarik(int n) { |
| 8 | this.size = n; |
| 9 | this.itemDt = new double[this.size]; |
| 10 | } |
| 11 | public Larik2(double[] A) { |
| 12 | this.size = A.length; |
| 13 | this.itemDt = new double[this.size]; |
| 14 | for (int i = 0; i < this.size; i++) |
| 15 | this.itemDt[i] = A[i]; |
| 16 | } |
| 17 | public int getSize() { |
| 18 | return this.size; |
| 19 | } |
| 20 | public double getItem(int i) { |
| 21 | return this.itemDt[i]; |
| 22 | } |
| 23 | public static Larik2 sambung(Larik2 l1, Larik2 l2) { |
| 24 | Larik2 lHasil = null; |
| 25 | int temp; |
| 26 | if (0 <= l1.getSize()+l2.getSize()) { |
| 27 | lHasil = new Larik2(l1.getSize()+l2.getSize()); |
| 28 | for (temp = 0; temp < l1.getSize(); temp++) |
| 29 | lHasil.isiItem(temp, l1.getItem(temp)); |
| 30 | for (int i=0; i < l2.getSize(); temp++,i++) |
| 31 | lHasil.isiItem(temp, l2.getItem(i)); |
| 32 | }return lHasil; |
| 33 | } |
| 34 | public void isiItem(int id, double dt) { |
| 35 | this.itemDt[id] = dt; |
| 36 | } |
| 37 | public void cetak(String komentar) { |
| 38 | System.out.println(komentar); |
| 39 | for (int i = 0; i < this.size; i++) { |
| 40 | System.out.printf("%.2f ", this.itemDt[i]); |
| 41 | } |
| 42 | System.out.println(); |
| 43 | } |
| 44 | public double findBesar() { |
| 45 | double besar = this.itemDt[0]; |
| 46 | for (int i = 1; i < this.size; i++) |
| 47 | if (besar < this.itemDt[i]) |
| 48 | besar = this.itemDt[i]; |
| 49 | return besar; |
| 50 | } |
| 51 | public int FindPosPos(int x){ |
| 52 | boolean a=false; |
| 53 | int b=0; |
| 54 | for (int i = 0; i <itemDt.length; i++) |
| 55 | if (itemDt[i]==x){ |
| 56 | a=true; |
| 57 | b=i; |
| 58 | } |
| 59 | if(a) |
| 60 | return b; |
| 61 | else return 0; |
| 62 | } |
| 63 | public static int LarikKaliLarik(Larik2 l1,Larik2 l2){ |
| 64 | int hasil=0; |
| 65 | for (int i = 0; i < l1.getSize(); i++) |
| 66 | hasil+=l1.getItem(i)\*l2.getItem(i); |
| 67 | return hasil; |
| 68 | } |
| 69 | public int getPosisi(double dtCari) { |
| 70 | int pos = -99; |
| 71 | boolean ketemu = false; |
| 72 | int i = 0; |
| 73 | while (!ketemu && i < this.size) { |
| 74 | if (dtCari == this.itemDt[i]) { |
| 75 | ketemu = true; |
| 76 | pos = i; |
| 77 | } |
| 78 | i++; |
| 79 | }return pos; |
| 80 | } |
| 81 | /\*\* |
| 82 | \* fungsi static untuk mencopy isi suatu larik l |
| 83 | \* |
| 84 | \* @param k : posisi awal |
| 85 | \* @param n : jumlah item yang akan dicopy |
| 86 | \* @param l : larik asal |
| 87 | \* @return Larik hasil copy |
| 88 | \*/ |
| 89 | public static Larik2 copyLarik(int k, int n, Larik2 l) { |
| 90 | // lenkapi bagian ini |
| 91 | Larik2 lHasil = null; |
| 92 | if (k <= n ) { |
| 93 | lHasil = new Larik2(n); |
| 94 | for (int i = k; i < n; i++) |
| 95 | lHasil.isiItem(i, l.getItem(i)); |
| 96 | }return lHasil; |
| 97 | } |
| 98 | /\*\* |
| 99 | \* fungsi untuk mencari posisi terbesar suatu data suatu posisi awal sampai |
| 100 | \* akhir |
| 101 | \* |
| 102 | \* @param awal : posisi awal |
| 103 | \* @param akhir : posisi akhir |
| 104 | \* @return posisi data terbesar |
| 105 | \*/ |
| 106 | public int getPosBesar(int awal, int akhir) { |
| 107 | int posBesar = -1; |
| 108 | double itemBesar; |
| 109 | if (awal <= akhir) { |
| 110 | posBesar = awal; |
| 111 | itemBesar = this.getItem(awal); |
| 112 | for (int i = awal + 1; i < akhir; i++) { |
| 113 | double nilaiItem = this.getItem(i); |
| 114 | if (itemBesar < nilaiItem) { |
| 115 | itemBesar = nilaiItem; |
| 116 | posBesar = i; |
| 117 | } |
| 118 | } |
| 119 | }return posBesar; |
| 120 | } |
| 121 |  |
| 122 | /\*\* |
| 123 | \* fungsi untuk mencari posisi data terkecil suatu array mulai dari posisi |
| 124 | \* awal sampai posisi akhir |
| 125 | \* |
| 126 | \* @param awal : posisi awal |
| 127 | \* @param akhir : posisi akhir |
| 128 | \* @return posisi data terkecil |
| 129 | \*/ |
| 130 | public int getPosKecil(int awal, int akhir) { |
| 131 | // lenkapi bagian ini |
| 132 | int poskecil = -1; |
| 133 | double itemkecil; |
| 134 | if (awal <= akhir) { |
| 135 | poskecil = awal; |
| 136 | itemkecil = this.getItem(awal); |
| 137 | for (int i = awal + 1; i < akhir; i++) { |
| 138 | double nilaiItem = this.getItem(i); |
| 139 | if (itemkecil > nilaiItem) { |
| 140 | itemkecil = nilaiItem; |
| 141 | poskecil = i; |
| 142 | } |
| 143 | } |
| 144 | }return poskecil; |
| 145 | } |
| 146 | /\*\* |
| 147 | \* fungsi pengurutan suatu larik lAsal dimana kondisi lAsal akan tetap |
| 148 | \* setelah proses pengurutan |
| 149 | \* |
| 150 | \* @param lAsal : Array asal yang akan diurutkan |
| 151 | \* @param status : 0-> urut dari kecil ke besar 1-> urut dari besar ke kecil |
| 152 | \* \* @return Array baru hasil pengurutan |
| 153 | \*/ |
| 154 | public static Larik2 SelectionSort(Larik2 lAsal, int status) { |
| 155 | int n = lAsal.getSize(); |
| 156 | Larik2 lhasil = Larik2.copyLarik(0, n, lAsal); |
| 157 | if (status == 0) { |
| 158 | for (int i = 0; i < n; i++) { |
| 159 | int posKecil = lhasil.getPosKecil(i,n); |
| 160 | double itemKecil= lhasil.getItem(posKecil); |
| 161 | double itemI = lhasil.getItem(i); |
| 162 | lhasil.isiItem(i, itemKecil); |
| 163 | lhasil.isiItem(posKecil, itemI); |
| 164 | } |
| 165 | } else { |
| 166 | for (int i = 0; i < n; i++) { |
| 167 | int posBesar = lhasil.getPosBesar(i,n); |
| 168 | double itemBesar = lhasil.getItem(posBesar); |
| 169 | double itemI = lhasil.getItem(i); |
| 170 | lhasil.isiItem(i, itemBesar); |
| 171 | lhasil.isiItem(posBesar, itemI); |
| 172 | } |
| 173 | }return lhasil; |
| 174 | } |
| 175 | public static void main(String[] args) { |
| 176 | // implementasi untuk ADT\_Larik |
| 177 | double[] A = {3, 4, 1, 10, 5, 2, 10, 20, 16}; |
| 178 | double[] B = {4, 3, 1, 11, 7}; |
| 179 | Larik2 L1 = new Larik2(A); |
| 180 | Larik2 L2 = new Larik2(B); |
| 181 | L1.cetak("L1"); |
| 182 | L2.cetak("L2"); |
| 183 | Larik2 L3 = Larik2.sambung(L1, L2); |
| 184 | L3.cetak("L3"); |
| 185 | Larik2 L4 = Larik2.copyLarik(0, L1.getSize(), L1); |
| 186 | L1.cetak("L1"); |
| 187 | L4.cetak("L4"); |
| 188 | Larik2 L5 = Larik2.SelectionSort(L1, 0); |
| 189 | L5.cetak("L5 Hasil pengurutan dari L1 besar -> kecil"); |
| 190 | Larik2 L6=Larik2.SelectionSort(L1, 1); |
| 191 | L6.cetak("L6 Hasil pengurutan dari L1 kecil -> besar"); |
| 192 | //int[] posisi = L1.FindPosPos(10); |
| 193 | L1.cetak("Isi Larik L1"); |
| 194 | double hasil = Larik2.LarikKaliLarik(L1, L4); |
| 195 | System.out.printf("HASIL KALI Larik L1\*L4 = %.3f\n", hasil); |
| 196 | } |
| 197 | } |

1. **PEMBAHASAN**

**No. 1**

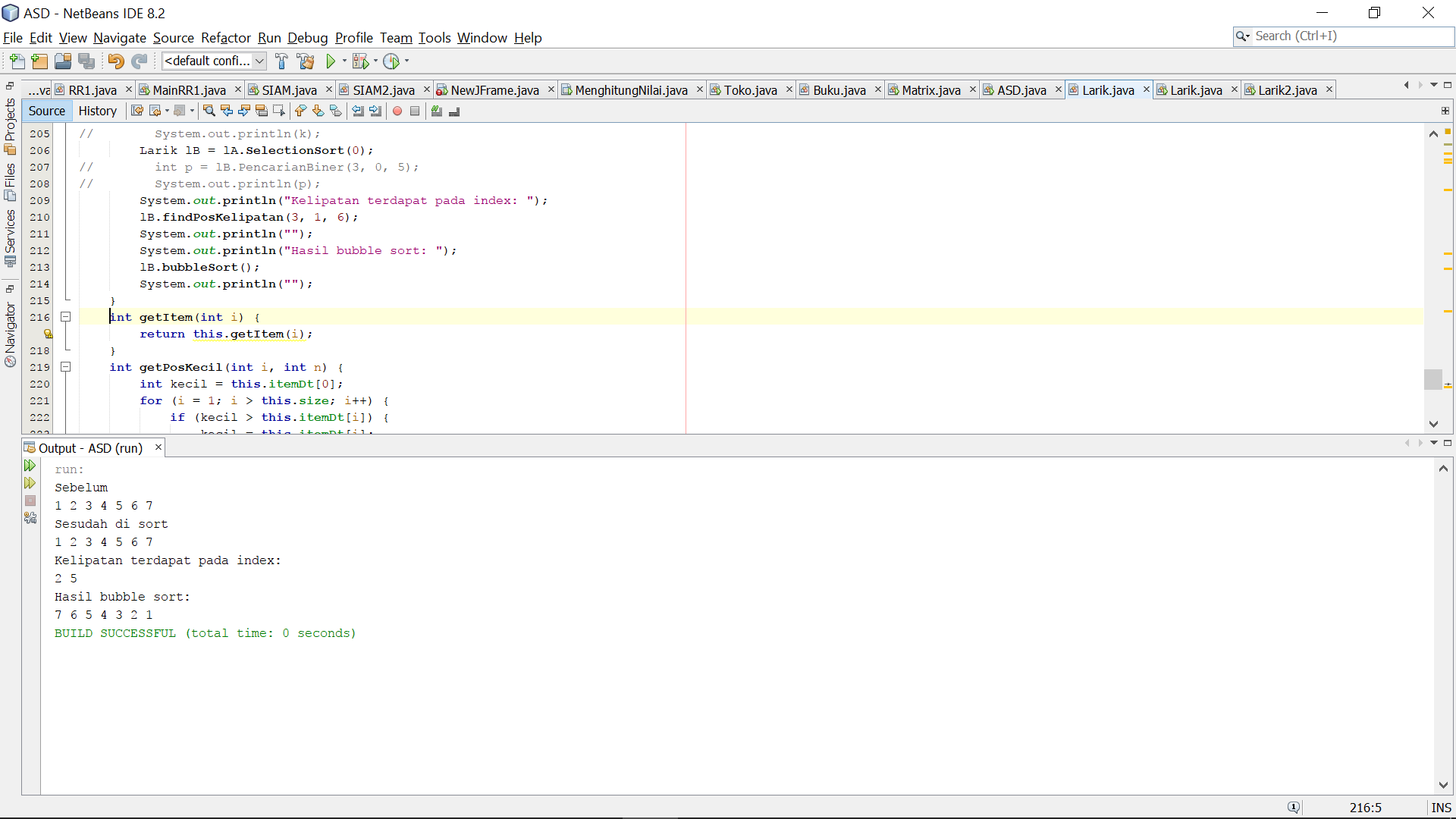
|  |  |
| --- | --- |
| Larik.java | |
| 1 | membuat package berna asd |
| 2 | membuat class bernama Larik |
| 3 | comment |
| 4 | deklarasi variable bernama size dengan tipe data intger dan memiliki hak akses private |
| 5 | deklarasi variable berjenis aray dengan nama itemDt dengan tipe data integer dan hak akses private |
| 6 | comment |
| 7 | membuat method berjenis void dengan nama buatLarik dengan parameter variable n bertipe data integer |
| 8 | nilai variable n disimpan pada variable size |
| 9 | nilai variabel itemDt dengan data pada array size |
| 10 | menutup method buatLarik |
| 11 | membuat konstruktor dengan nama yang sama dengan nama class dengan parameter bernama n bertipe data integer |
| 12 | memanggil method buatLarik dengan parameter n |
| 13 | menutup konstruktor |
| 14 | membuat method bernama getSize betipe data integer |
| 15 | mengembalikan nilai pada variable size |
| 16 | menutup method getSize |
| 17 | membuat konstruktor dengan nama yang sama dengan nama class dengan parameter berjenis array dengan nama dt bertipedata integer |
| 18 | memanggil method buatLarik dengan parameter banyak nilai darivariable dt |
| 19 | proses perulangan |
| 20 | memanggil method isiItem dengan parameter variable I dan variable dt dalam bentuk array dengan indeks ke i |
| 21 | menutup proses perulangan |
| 22 | menutup konstruktor |
| 23 | membuat method void bernama isiItem dengan parameter variable bernama id dan dt yang masing-masing bertipedata integer |
| 24 | variable berbentu array bernama itemDT berindeks id menyimpan nilai dari variable dt |
| 25 | menutup method |
| 26 | membuat method void bernama cetak dengan parameter variable komentar bertipedata String |
| 27 | mencetak keluaran dari nilai variable komentar |
| 28 | proses perulangan |
| 29 | mencetak hasil dari memanggil nilai dari variable itemDt berindeks I |
| 30 | menutup proses perulangan |
| 31 | mencetak enter |
| 32 | menutup method |
| 33 | membuat method bernama findBesar dengan tipedata integer |
| 34 | deklarasi variable bernama besar yang menimpan nilai variable berbentuk array bernama itemDt dengan indeks ke 0 |
| 35 | melakukan proses perulangan |
| 36 | seleksi kondisi jika variable besar lebih kecil dari nilai arrat variable itemDt indeks ke i |
| 37 | maka variable besar = variabel array itemDt |
| 38 | menutup proses seleksi kondisi |
| 39 | menutup proses perulangan |
| 40 | mengembalikan nilai dan disimpan pada variable besar |
| 41 | menutup method |
| 42 | comment |
| 43 | comment |
| 44 | comment |
| 45 | membuat method bernama getPosisi bertpe data integer dengan parameter variabel dtCari bertipedata integer |
| 46 | deklarasi dan inisialisasi variable bernama pos bertipe data integer dengan nilai -99 |
| 47 | deklarasi nilai boolean pada variable ketemu yang bernilai false |
| 48 | inisialisasi variable i bertipedata integer yaitu i = 0 |
| 49 | proses perulangan |
| 50 | proses seleksi kondisi |
| 51 | dimana variable ketemu = true |
| 52 | dimana pos = i |
| 53 | menutup proses seleksi kondisi |
| 54 | increment variable i |
| 55 | menutup proses perulangan |
| 56 | mengembalikan nilai yang disimpan pada variabel pos |
| 57 | menutup method |
| 58 | Deklarasi method bernama getposMax berparameter id bertipe data integer |
| 59 | Menambahkan variabel max dengan memanggil methd itemDt dengan array id |
| 60 | Menambahkan variabel poMax dengan data i |
| 61 | Melakukan perulangan for dengan syarat int i = id + 1; i < size; i++ |
| 62 | Melakukan seleksi if dengan syarat max <= this.itemDt[i] |
| 63 | Melakukan seleksi if dengan syarat max = this.itemDt[i]; |
| 64 | Memanggil variabel posMax dengan data i |
| 65 | Menutup badan seleksi |
| 66 | Menutup badan perulangan |
| 67 | Mengambalikan nilai ke variabel posMax |
| 68 | Menutup badan method |
| 69 | Deklarasi method bernama getposMin berparameter id bertipe data integer |
| 70 | Menambahkan variabel min dengan memanggil methd itemDt dengan array id |
| 71 | Menambahkan variabel posMin dengan data id |
| 72 | Melakukan perulangan for dengan syarat int i = id + 1; i < size; i++ |
| 73 | Melakukan seleksi if dengan syarat min <= this.itemDt[i] |
| 74 | Melakukan seleksi if dengan syarat min = this.itemDt[i]; |
| 75 | Memanggil variabel posMin dengan data i |
| 76 | Menutup badan seleksi |
| 77 | Menutup badan perulangan |
| 78 | Mengambalikan nilai ke variabel posMin |
| 79 | Menutup badan method |
| 80 | membuat method PencarianBinner dengan tipedata integer dengan parameter variable bernama dtCari, awal, akhir yang masing-masing bertipedata integer |
| 81 | instansiasi variable bernama pos bertipedata integer dengan nilai -99 |
| 82 | deklarasi variable tengah bertipedata integer yang menyimpan nilai/hasil dari proses perhitungan |
| 83 | proses seleksi kondisi |
| 84 | comment |
| 85 | mengembalikan nilai pada method PencarianBinner dengan 3 parameternya |
| 86 | kondisi lain |
| 87 | comment |
| 88 | mengembalikan nilai pada method PencarianBinner dengan 3 parameternya |
| 89 | kondisi lain |
| 90 | mengembalikan nilai ke variable tengah |
| 91 | selain itu |
| 92 | mengembalikan nilai ke variable pos |
| 93 | menutup proses perulangan |
| 94 | menutip method |
| 95 | comment |
| 96 | comment |
| 97 | comment |
| 98 | comment |
| 99 | comment |
| 100 | Deklarasi method bernama copyLarik dengan parameter int k dan int n |
| 101 | Menambahkan variabel Larik dan lHasil dengan data null |
| 102 | Melakukan seleksi if dengan syarat n <= this.size – k |
| 103 | Menampilkan variabel lHasil dengan data variabel Larik |
| 104 | Menambahkan variabel j dengan data 0 |
| 105 | Melakukan perulangan for dengan syarat int i = k; i < k + n; i++ |
| 106 | Memanggil variabel isiItem dengan variabel lHasil dengan data arrayItemDt |
| 107 | Menutup badan perulangan |
| 108 | Menutup badan seleksi |
| 109 | Mengambalikan nilai pada variabel 1hasil |
| 110 | Menutup badan method |
| 111 | Argument |
| 112 | Argument |
| 113 | Argument |
| 114 | Argument |
| 115 | Deklarasi method Larik selectionSort dengan parameter pilihan dengan tipe data integer |
| 116 | Memanggil variabel Larik dan 1sort dengan data copyLarik |
| 117 | Melakukan perulangan for dengan syarat int i = 0; i < lsort.getSize(); i++ |
| 118 | Menambahkan variabel posData bertipe data integer |
| 119 | Melakukan seleksi if dengan syarat pilihan == 0 |
| 120 | Menampilkan posData dengan memanggil getPosMin menggunakan variabel 1sort |
| 121 | Seleksi if else |
| 122 | Menampilkan posData dengan memanggil getPosMax menggunakan variabel 1sort |
| 123 | Menutup badan seleksi if else |
| 124 | Menambahkan variabel dt1 dengan memanggil variabel itemDt menggunakan 1sort dengan indeks i |
| 125 | Menambahkan variabel dt2 dengan memanggil variabel itemDt menggunakan 1sort dengan indeks posData |
| 126 | Memanggil variabel itemDt array I menggunakan 1sort dengan data dt2 |
| 127 | Memanggil variabel itemDt array posData menggunakan 1sort dengan data dt1 |
| 128 | Menutup badan perulangan |
| 129 | Mengembalikan nilai pada variabel 1sort |
| 130 | Menutup badan method |
| 131 | deklarasi method bertipe integer bernama findPosKelipatan dengan 3 parameter |
| 132 | memanggil method copyLarik dengan 2 parameter dengan objek salin pada class Larik |
| 133 | inisialisasi variable bertipe data integer berna,a search dan index |
| 134 | proses perulangan |
| 135 | proses seleksi kondisi |
| 136 | mencetak output |
| 137 | menutup proses seleksi kondisi |
| 138 | comment |
| 139 | comment |
| 140 | comment |
| 141 | comment |
| 142 | comment |
| 143 | comment |
| 144 | comment |
| 145 | comment |
| 146 | menutup proses perulangan |
| 147 | mengembalika nilai |
| 148 | menutup method |
| 149 | Deklarasi method bernama bubbleSort |
| 150 | Menambahkan variabel temp dengan data 0 |
| 151 | Melakukan perulangan for dengan syarat int i = 0; i < (itemDt.length - 1); i++ |
| 152 | Melakukan perulangan for dengan syarat int j = 0; j < (itemDt.length - 1); j++ |
| 153 | Melakukan seleksi if dengan syarat itemDt[j] < itemDt[j + 1] |
| 154 | Melakukan seleksi dengan syarat temp = itemDt[j]; |
| 155 | Melakukan seleksi dengan syarat itemDt[j] = itemDt[j + 1]; |
| 156 | Melakukan seleksi dengan syarat itemDt[j + 1] = temp; |
| 157 | Menutup badan seleksi |
| 158 | Menutup badan perulangan for |
| 159 | Menutup badan perulangan for |
| 160 | Melakukan perulangan for dengan syarat int i = 0; i < itemDt.length; i++ |
| 161 | Menampilkan output menggunakan kelas system dengan data pada array I variabel itemDt |
| 162 | Menutup badan perulangan |
| 163 | Menutup badan method |
| 164 | Deklarasi method bernama MergeSort dengan parameter array dtArray |
| 165 | Menampilkan variabel mergeSort dengan dtArray |
| 166 | Melakukan perulangan for dengan syarat int i = 0; i < dtArray.length; i++ |
| 167 | Menampilkan output menggunakan kelas system dengan keluaran data array dtArry |
| 168 | Menutup badan perulangan |
| 169 | Menutup badan method |
| 170 | Deklarasi method bernama mergeSort\_1 dengan parameter array[] dan Io dan n |
| 171 | Menambahkan variabel low |
| 172 | Menambahkan variabel hight |
| 173 | Melakukan seleksi if dengan syarat low >= high |
| 174 | Sintaks return value |
| 175 | Menutup badan seleksi |
| 176 | Melakukan operasi matematika pada variabel middle dengan operasi(low+ high)/ 2 |
| 177 | Menampilkan variabel mergeSort\_1 dengan parameter array, low, middle |
| 178 | Menampilkan variabel mergeSort\_1 dengan parameter array, middle + 1, high |
| 179 | Menambahkan variabel end\_low dengan sata middle |
| 180 | Menambahkan variabel start\_hight dengan data middle + 1 |
| 181 | Melakukan perulangan while dengan syarat lo <= end\_low) &&(start\_high <= high |
| 182 | Melakukan seleksi if dengan syarat array[low] < array[start\_high] |
| 183 | increment nilai low |
| 184 | proses lain seleksi kondisi |
| 185 | variable integer bernama Temp = array[start\_high] |
| 186 | proses perulangan |
| 187 | memanggil nilai pada array[k + 1] = array[k]; |
| 188 | menutup proses perulangan |
| 189 | nilai array[low] disimpan pada vaiable Temp |
| 190 | increment nilai low |
| 191 | increment nilai end\_low |
| 192 | increment nilai start\_high |
| 193 | Menutup badan seleksi if else |
| 194 | Menutup badan perulangan |
| 195 | Menutup badan method |
| 196 | Deklarasi method main |
| 197 | Menambahkkan array a dengan data 1, 2, 3, 4, 5, 6, 7 |
| 198 | Inisialisasi Larik dengan 1A |
| 199 | Memanggil keluaran sebelum |
| 200 | Memanggil SelectionSort dengan mencetak “Sesudah di Sort” |
| 201 | comment |
| 202 | comment |
| 203 | comment |
| 204 | comment |
| 205 | Menampilkan output hasil sesudah disort |
| 206 | comment |
| 207 | comment |
| 208 | mencetak ouput " Kelipatan terdapat pada index: " |
| 209 | memanggil method findPosKelipatan dengan objek lB dengan parameter 3, 0, 5 |
| 210 | mencetak enter |
| 211 | mencetak output " hasil bubble sort:" |
| 212 | memanggil method bubbleSort dengan objek lB |
| 213 | mencetak enter |
| 214 | menutup main method |
| 215 | deklarasi method getItem bertipe integer dengan parameter variable integer bernama i |
| 216 | mengembalikan nilai ke method getItem dengan 1 parameter |
| 217 | menutup method |
| 218 | deklarasi method getPosKecil bertipe integer dengan parameter variable masing-masing berjenis integer bernama i dan n |
| 219 | nilai array itemDt indeks ke 0 disimpan pada variable integer bernama kecil |
| 220 | proses perulangan |
| 221 | proses seleksi kondisi |
| 222 | nilai dari array itemDt indeks ke - I akan disimpan pada variable kecil jika kondisi memenuhi |
| 223 | menutup seleksi kondisi |
| 224 | menutup perulangan |
| 225 | mengembalikan nilai ke variable kecil |
| 226 | menutup method |
| 227 | deklarasi method getPosBesar bertipe integer dengan parameter variable masing-masing berjenis integer bernama i dan n |
| 228 | nilai array itemDt indeks ke 0 disimpan pada variable integer bernama kecil |
| 229 | proses perulangan |
| 230 | proses seleksi kondisi |
| 231 | nilai dari array itemDt indeks ke - i akan disimpan pada variable besar jika kondisi memenuhi |
| 232 | menutup seleksi kondisi |
| 233 | menutup perulangan |
| 234 | mengembalikan nilai ke variable besar |
| 235 | menutup method |
| 236 | mengakhiri class |

**No.2**

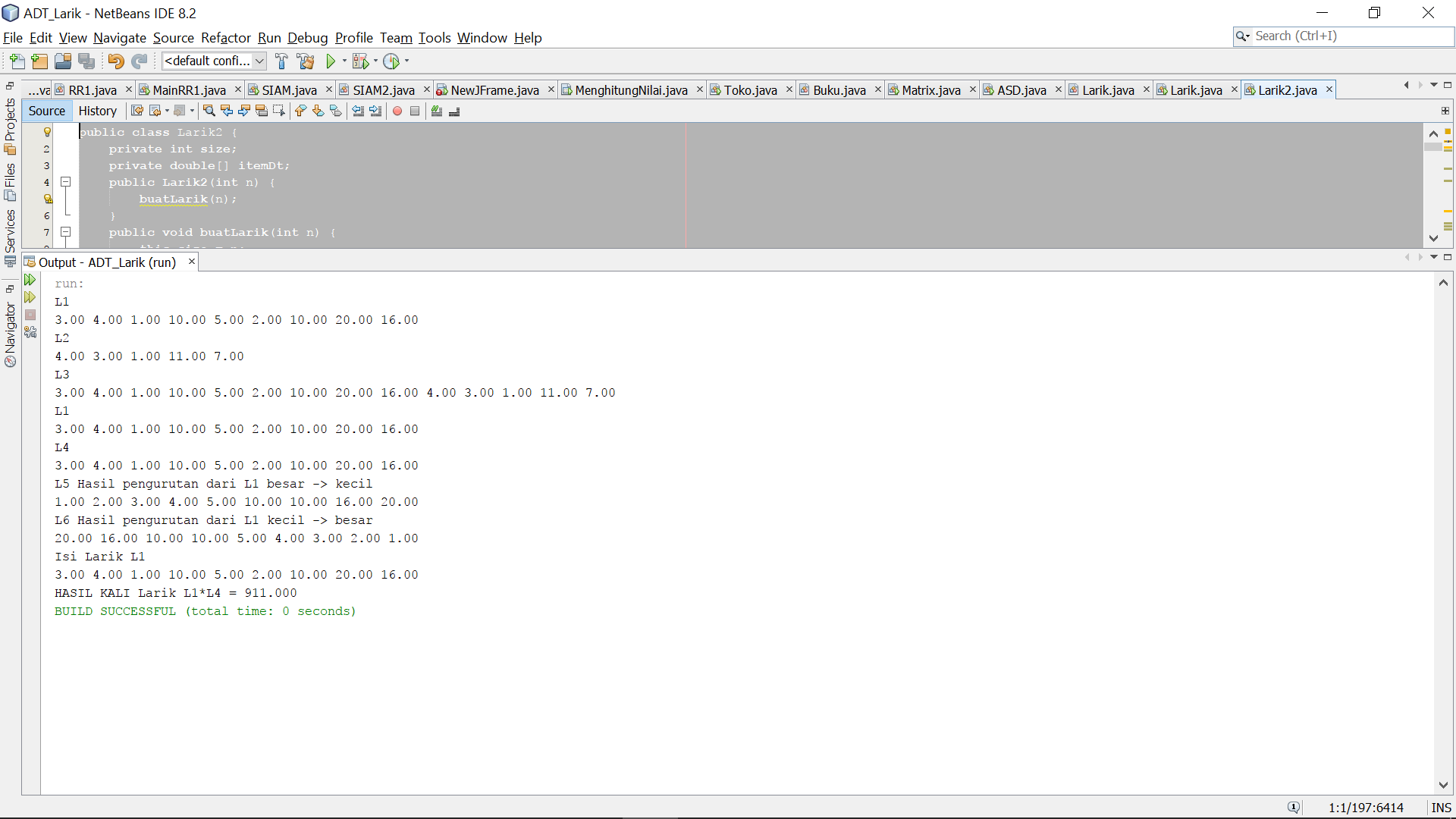
|  |  |
| --- | --- |
| Larik2.java | |
| 1 | Nama class larik2 |
| 2 | Inisiasi int variable size |
| 3 | Inisiasi array double bernama itemDt |
| 4 | Konstruktor larik2 dengan satu parameter |
| 5 | Memanggil method buatlarik |
| 6 | menutup method |
| 7 | Deklarasi Method buatLarik dengan satu parameter |
| 8 | menyimpan nilai pada size ke variable n |
| 9 | panjang array nilai size disimpan pada itemDt |
| 10 | menutup method |
| 11 | deklarasi kontrukor dengan parameter array 1 dimensi |
| 12 | panjang nilai a dsimpan pada variable size |
| 13 | panjang array nilai size disimpan pada itemDt |
| 14 | proses perulangan |
| 15 | nilai array A indeks ke - I disimpan pada array itemDt indeks ke i |
| 16 | menutup konstruktor |
| 17 | deklarasi method getSize bertipe integer |
| 18 | mengembalikan nilai size |
| 19 | menutup method |
| 20 | deklarasi method get item bertipedata double berparameter satu |
| 21 | Mengembalikan nilai itemdt |
| 22 | menutup method |
| 23 | deklarasi method static sambung bertipe Larik2 dengan 2 parameter |
| 24 | Object iHasil memiliki nilai null |
| 25 | Inisiasi temp |
| 26 | proses seleksi kondisi jika 0 kurang darisamadengan panjang l1 dan l2 |
| 27 | Maka lhasil samadengan larik dengan panajng l1danl2 |
| 28 | proses perulangan |
| 29 | Memanggil method isiItem dengan objek lHasil dengan 2 parameter |
| 30 | proses perulangan |
| 31 | Memanggil method isiItem dengan objek lHasil dengan 2 parameter |
| 32 | Mengembalikan lhasil dan menutup proses seleksi kondisi |
| 33 | menutup method |
| 34 | deklarasi method isiItem dengan 2 poarameter |
| 35 | menyimpan nilai array itemDt pada variable dt |
| 36 | menutup method |
| 37 | deklarasi method void cetak dengan 1 parameter string |
| 38 | mencetak method komentar |
| 39 | proses perulangan |
| 40 | mencetak nilai dari array itemBt indeks ke-i |
| 41 | menutup seleksei kondisi |
| 42 | mencetak enter |
| 43 | menutup method |
| 44 | deklarasi method findBesar |
| 45 | inisiasi variable bertipe double bernama besar yang menyimpan nilai itemDt indeks ke 0 |
| 46 | proses perulangan |
| 47 | proses seleksi kondisi jika variable besar kurang dari itemDt indeks ke i |
| 48 | atau variabel besar sama dengan itenDt indeks ke i |
| 49 | mengembalikan nilai ke variabel besar |
| 50 | menutup method |
| 51 | deklarasi Method findpospos dengan satu parameter integer |
| 52 | Inisiasi bolean a bernilai false |
| 53 | Inisasi int b |
| 54 | proses Perulangan for |
| 55 | seleksi kondisi Jika itemdt[i] sama dengan x |
| 56 | Maka a bernilai true |
| 57 | B bernilai i |
| 58 | menutup seleksi kondisi pertama |
| 59 | Jika variable a bernilai true |
| 60 | maka akan mengembalikan b |
| 61 | Selain itu mengembalikan 0 |
| 62 | menutup method |
| 63 | deklarasi Method larikkalilarik dengan 2 parameter bertipe Larik2 |
| 64 | Inisiasi hasil |
| 65 | Proses perulangan for |
| 66 | Hasil samadengan hasil ditambah hasi kali l1 da l2 |
| 67 | Mengembalikan hasil |
| 68 | menutup method |
| 69 | deklarasi Method getposisi dengan satu parameter |
| 70 | Inisiasi pos |
| 71 | Inisisasi ketemu = false |
| 72 | Inisiasi i |
| 73 | Perulangan while |
| 74 | seleksi kondisi jika dtcari sama dengan itemdt index ke i |
| 75 | Maka ketemeu menjadi true |
| 76 | pos menyimpan nilai i |
| 77 | menutup perulangan |
| 78 | increment variable i |
| 79 | menutup perulangan dan mengembalikan nilai ke variable pos |
| 80 | menutup method |
| 81 | comment |
| 82 | comment |
| 83 | comment |
| 84 | comment |
| 85 | comment |
| 86 | comment |
| 87 | comment |
| 88 | comment |
| 89 | deklarasi method static bertipe LArik2 dengan nama copyLArik dengan 3 parameter |
| 90 | comment |
| 91 | object lHasil bertipe Larik2 bernilai null |
| 92 | proses perulangan |
| 93 | penyimpanan nilai pada variable lHasil |
| 94 | proses perulangan |
| 95 | memanggil method isiItem dengan 2 parameter |
| 96 | menutup proses perulangan dan mengembalikan nilai ke lHasil |
| 97 | menutup method |
| 98 | comment |
| 99 | comment |
| 100 | comment |
| 101 | comment |
| 102 | comment |
| 103 | comment |
| 104 | comment |
| 105 | comment |
| 106 | deklarasi method getposbesar |
| 107 | inisiasi posbesar |
| 108 | deklarasi itembesar |
| 109 | proses seleksi kondisi jika awal kurang dari akhir |
| 110 | nilai posbesar di awal |
| 111 | itembesar disimpan nilainya di itemdt indek awal |
| 112 | proses perulangan for |
| 113 | inisiadi nilai item |
| 114 | proses seleksi kondisi jika itembesar<nilaiitem |
| 115 | maka itembesar menjadi nilai item |
| 116 | dan posbesar disimpan nilainya di variable i |
| 117 | menutup seleksi kondisi |
| 118 | menutup perulangan |
| 119 | menutup seleksi kondisi dan mengembalikan nilai |
| 120 | menutup method |
| 121 | - |
| 122 | comment |
| 123 | comment |
| 124 | comment |
| 125 | comment |
| 126 | comment |
| 127 | comment |
| 128 | comment |
| 129 | comment |
| 130 | deklarasi method getposkecil |
| 131 | comment |
| 132 | inisiasi poskecil |
| 133 | deklarasi variable itemkecil |
| 134 | proses seleksi kondisi |
| 135 | poskecil menyimpan nilai pada variable awal |
| 136 | itemkecil menyimpan nilai di itemdt indek awal |
| 137 | proses perulangan |
| 138 | inisiasi variable nilaiItem yang disimpan nilainya pada variable getItem pada indeks ke - i |
| 139 | seleksi kondisi jika itemkecilr>nilaiitem |
| 140 | maka variable itemkecil menyimpan nilai pada variable nilaiItem |
| 141 | nilai variable pokecil disimpan pada variable i |
| 142 | menutup seleksi kondisi |
| 143 | menutup perulangan dan mengembalikan nilai ke variable pos |
| 144 | menutup seleksi kondisi dan mengembalikan nilai |
| 145 | menutup method |
| 146 | comment |
| 147 | comment |
| 148 | comment |
| 149 | comment |
| 150 | comment |
| 151 | comment |
| 152 | comment |
| 153 | comment |
| 154 | deklarasi method static bertipe Larik2 beranama Selectionsort |
| 155 | inisiasi n |
| 156 | inisiasi object lhasil dengan nilai yang dipanggil dari method copylarik |
| 157 | seleksi kondisi |
| 158 | proses perulangan |
| 159 | deklarasi variable poskecil yang menyimpan nilai dari getItem |
| 160 | deklarasi variable itemkecil yang menyimpan nilai dari getItem |
| 161 | deklarasi itemI dengan nilai dari getitem |
| 162 | memanggil mehtod isiItem dengan 2 parameter |
| 163 | menutup perulangan |
| 164 | menutup seleksi kondisi |
| 165 | jika ada kondisi lain |
| 166 | proses perulangan |
| 167 | deklarasi pos besar menyimpan nilai dari getPosBesar |
| 168 | deklarasi itemBesar dengan nilai dari getItem |
| 169 | deklarasi itemI dengan nilai dari getItem; |
| 170 | memanggil method isiItem dengan 2 parameter |
| 171 | memanggil method isiItem dengan 2 parameter |
| 172 | menutup perulangan |
| 173 | menutup seleksi kondisi dan mengembalikan nilai |
| 174 | menutup method |
| 175 | deklarasi main method |
| 176 | comment |
| 177 | deklarasi variable array bernama A bertipe double beserta isi dari arraynya |
| 178 | deklarasi variable array bernama B bertipe double beserta isi dari arraynya |
| 179 | membuat objek bernama L1 |
| 180 | membuat objek bernama L2 |
| 181 | objek L1 memanggil method cetak |
| 182 | objek L2 memanggil method cetak |
| 183 | membuat objek baru bernama L3 dimana merupakan gabungan dari L1 dan L2 |
| 184 | objek L3 memanggil method cetak |
| 185 | membuat objek baru bernama L4 dimana nilainya berasal dari method copyLarik beserta parameternya |
| 186 | objek L1 memanggil method cetak |
| 187 | objek L4 memanggil method cetak |
| 188 | membuat objek baru bernama L5 dimana nilainya berasal dari method SelectionSort beserta parameternya |
| 189 | objek L5 memanggil method cetak |
| 190 | membuat objek baru bernama L6 dimana nilainya berasal dari method SelectionSort beserta parameternya |
| 191 | objek L6 memanggil method cetak |
| 192 | comment |
| 193 | objek L1 memanggil method cetak |
| 194 | deklarasi variabel hasil bertipe double yang menyimpan nilai dari method LarikKaliLarik beserta parameternya |
| 195 | mencetak nilai variable hasil |
| 196 | menutup main method |
| 197 | menutup class |

1. **SCREENSHOT PROGRAM**

**No. 1**



**No. 2**



1. **KESIMPULAN**

Tipe data abstrak (ADT) dapat didefinisikan sebagai model matematika dari objek data yang menyempurnakan tipe data dengan cara mengaitkannya dengan fungsi-fungsi yang beroprasi pada data yang bersangkutan*.* Merupakan hal yang sangat penting untuk mengenali bahwa operasi-operasi yang akan dimanipulasi data pada objek yang bersangkutan termuat dalam spesifikasi ADT.

Cara membuat array pada Java. Kita mulai dengan pengertian array. Array adalah sekumpulan data yang memiliki tipe yang sama, variabel yang sama dan tiap data dibedakan dengan indeks dalam array tersebut. Indeks dari tiap array dimulai dengan indeks ke-0. Jadi jika suatu array memiliki berisi 5 buah data maka indeks maksimal dari array tersebut adalah 4. Manfaat dari array adalah untuk mengurangi jumlah variabel yang digunakan. Array dapat berbentuk dimensi satu, dua, tiga dan seterusnya tergantung kebutuhan dari pengguna.

- Pengaksesan Array

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public class array{  public static void main (String args []){  int nilai[] = new int[3];  nilai[0]=70;  nilai[1]=80;  nilai[2]=65;  for(int i=0; i<nilai.length;i++)  System.out.println("indeks ke-"+i+" = " +nilai[i]);  } } |

Array 1 dimensi adalah array yang deretan nilainya adalah linier dan squensial, atau sering disebut sebagai representasi data bentuk row atau 1 dimensi.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import java.io.\*;  public class ContohArray1{ public static void main(String[] args) { try{ int[] angka = new int[5]; System.out.println("Masukkan 5 Data"); System.out.println("==============="); BufferedReader in = new BufferedReader(new InputStreamReader (System.in)); for (int i=0;i<angka.length;i++) { System.out.print("Masukkan Data Ke-"+(i+1)+" : "); angka [i] = Integer.parseInt(in.readLine()); } System.out.println("\nData Yang Ada Di Array :"); System.out.println("==============="); for (int i=0;i<angka.length;i++) { System.out.println("Data Ke-"+(i+1)+" : "+angka[i]);  } } catch(Exception e) { System.out.println("Error");  }  } } |