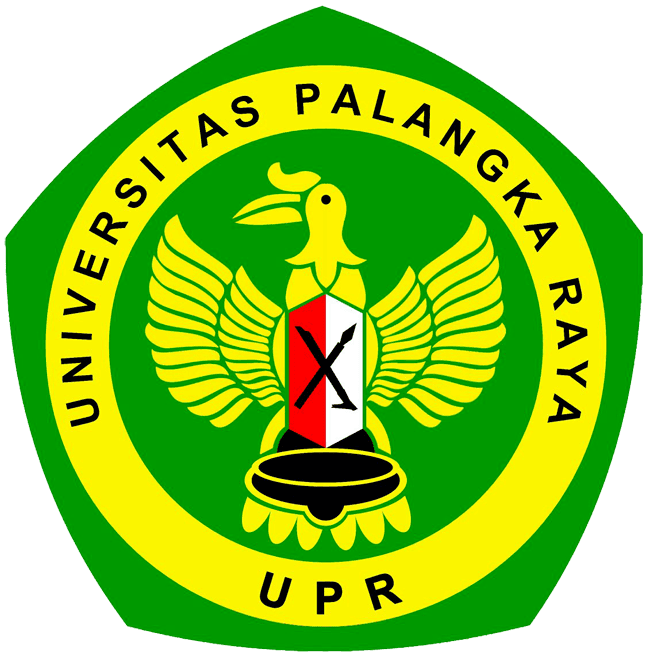
**LAPORAN RENCANA PRAKTIKUM**

**STRUKTUR DATA**



|  |  |  |
| --- | --- | --- |
| **NAMA** | **:** | **INDRA FIQI RIPANI** |
| **NIM** | **:** | **213010503002** |
| **KELAS** | **:** | **F** |
| **MODUL** | **:** | **V (PENGURUTAN (*SORTING*))** |

**Program Studi S1 Teknik Informatika**

**Fakultas Teknik**

**Universitas Palangka Raya**

**Palangka Raya, Kalimantan Tengah**

**2022**

**MODUL V**

**PENGURUTAN (*SORTING*)**

1. **TUJUAN PRAKTIKUM**
2. Mahasiswa memahami bagaimana logika yang digunakan dalam setiap jenis-jenis *sorting*.
3. Mahasiswa mampu menerapkan algoritma *sorting* ke dalam program.
4. **LANDASAN TEORI**

Algoritma Pengurutan adalah suatu metode yang dibutuhkan untuk mengurutkan data. Data yang diurutkan dapat berupa tipe dasar atau bertipe tersturktur (*struct*). Jika data bertipe terstruktur, maka harus dispesifikasikan berdasarkan *field* apa data tersebut diurutkan. Terdapat banyak algoritma yang digunakan untuk mengurutkan data. Setiap algoritma mempunyai kekuatan dan kelemahan. Pada banyak cara pemecahan masalah, teknik terbaik untuk mengg-unakan algoritma pengurutan adalah dengan membangun algoritma pengurutan dalam bentuk fungsi. Jenis algoritma pengurutan yang dibahas pada modul ini adalah algoritma *buble sort*, *selection sort*, *shell short*, dan *insertion sort*.

1. *Buble Sort*

Algoritma pengurutan apung akan membandingkan elemen yang sekarang dengan elemen berikutnya, pada pengurutan *ascending* jika nilai elemen sekarang > elemen berikutnya, maka dilakukan proses penukaran. Misalkan terdapat L larik yang tiap indeksnya k berisi data belum terurut dengan jumlah elemen n, maka untuk mendapatkan larik yang terurut *ascending*, algoritmanya secara umum seperti berikut.

|  |
| --- |
| Procedure Buble\_sort ( Larik[] array of integer, n > integer )  Begin  For i=1 to n do  For j=1+i to n do  if data[i]>data[j] swap(data[i],data[j])  end. |

1. *Selection Sort*

Ide dasar pengurutan seleksi adalah memilih elemen maksimum /minimum dari larik lalu menempatkan elemen maksimum/minimum itu pada awal atau pada akhir larik. Selanjutnya elemen awal/akhir tersebut “diisolasi” dan tidak disertakan dalam proses perbandingan selanjutnya. Seperti pada algoritma pengurutan apung, proses memilih nilai maksimum /minimum dilakukan pada setiap pass. Jika larik berukuran n, maka jumlah pass adalah n-1. Algoritma pengurutan seleksi untuk mendapatkan larik yang terurut *ascending*, secara umum seperti berikut.

|  |
| --- |
| Procedure Selection\_Sort (Larik[] array of integer, n > integer)  Begin  For i=1 to n do  Temp = i;  For j=i+1 to n do  If data[j]<data[Temp]  Temp;  Swap(data[j], data[Temp]  End. |

1. *Insertion Sort*

Inspirasi dari algoritma pengurutan sisip adalah seperti ketika mengurutkan kartu pada permainan kartu. Metode pengurutan dengan cara menyisipkan elemen larik pada posisi yang tepat. Pencarian posisi yang tepat dilakukan dengan menyisir lariknya. Selama penyisiran dilakukan pergeseran elemen larik. Algoritma pengurutan sisip untuk mendapatkan larik yang terurut *ascending*, secara umum seperti berikut.

Untuk setiap pass i = 2, ..., n lakukan:

1. y = L[i]
2. Sisipkan y pada tempat yang sesuai di antara L[1]...L[i].
3. *Shell Sort*

Pada algoritma pengurutan *shell*, jarak di antara dua elemen yang dibandingkan dan ditukarkan. Langkah pertama, ambil elemen pertama dan bandingkan dengan elemen pada jarak tertentu dari suatu elemen pertama tersebut. Kemudian elemen kedua dibandingkan dengan elemen lain dengan jarak yang sama seperti di atasnya. Demikian seterusnya sampai seluruh elemen dibandingkan. Pada *pass* berikutnya proses diulang dengan langkah yang lebih kecil, hingga proses dihentikan jika jarak sudah sama dengan satu. Secara umum, algoritma pengurutan Shell dituliskan sebagai berikut.

1. step = n {n = ukuran larik}
2. while step > 1 do
3. step = step div 3 +1
4. for I = 1 to step do

*insertion sort* setiap elemen ke - step mulai dari elemen ke - i.

1. **TUGAS**

Gunakanlah keempat jenis algoritma pengurutan di atas untuk mengurutkan data integer sebagai berikut.

15, 8, 0, -2, 19, 42, 31, 81, -72, 1, 2.