**PSEUDOCODE TUBES**

**kamus**

**type** DataMahasiswa <

Nama: **string**

NIM: **string**

>

**type** DataDosen <

Nama: **string**

NIDN: **string**

>

**type** DataKelas <

    NamaKelas: **string**

    Ruangan: **string**

>

Mahasiswa: **array** [0..99] **of DataMahasiswa**

Dosen: **array** [0..99] **of** **DataDosen**

Kelas: **array** [0..99] **of** **DataKelas**

**procedure** selectionSortingMahasiswa(**in/out** data: **Mahasiswa**, **in** n: **integer**)

**kamus**

i, j, minIndex: **integer**

**algoritma**

**for** i ← 0 **to** n-1 **do**

minIndex ← i

**for** j ← i+1 **to** n **do**

**if** data[j].NIM < data[minIndex].NIM **then**

minIndex ← j

**endif**

**endfor**

data[i] ← data[minIndex]

data[minIndex] ← data[i]

**endfor**

**endprocedure**

**procedure** insertionSortMahasiswa(**in/out** data: **Mahasiswa**, **in** n: **integer**)

**kamus**

i, j: **integer**

key: **Mahasiswa**

**algoritma**

**for** i ← 1 **to** n **do**

key ← data[i]

j ← i - 1

**while** j ≥ 0 **and** data[j].Nama > key.Nama **do**

data[j+1] ← data[j]

j ← j - 1

**endwhile**

data[j+1] ← key

**endfor**

**endprocedure**

**function** linearSearchMahasiswa(**in** data: **Mahasiswa**, **in** n: **integer**, **in** target: **string**) → **integer**

**kamus**

i: **integer**

**algoritma**

**for** i ← 0 **to** n **do**

**if** data[i].Nama = target **then**

**return** i

**endif**

**endfor**

**return** -1

**endfunction**

**function** binarySearchMahasiswa(**in** data: **Mahasiswa**, **in** n: **integer**, **in** target: **string**) → **integer**

**kamus**

temp: **array** of **DataMahasiswa**

bawah, atas, tengah: **integer**

**algoritma**

temp ← **new** **array** of **DataMahasiswa** **with** size n

**copy** data[0:n] **to** temp

**sort**(temp) by Nama **in** ascending order

bawah ← 0

atas ← n - 1

**while** bawah ≤ atas **do**

tengah ← (bawah + atas) / 2

**if** temp[tengah].Nama = target **then**

**return** tengah

**else** **if** temp[tengah].Nama < target **then**

bawah ← tengah + 1

**else**

atas ← tengah - 1

**endif**

**endwhile**

**return** -1

**endfunction**

**procedure** selectionSortingDosen(**in/out** data: **Dosen**, **in** n: **integer**)

**kamus**

i, j, minIndex: **integer**

**algoritma**

**for** i ← 0 **to** n-1 **do**

minIndex ← i

**for** j ← i+1 **to** n **do**

**if** data[j].NIDN < data[minIndex].NIDN **then**

minIndex ← j

**endif**

**endfor**

data[i] ← data[minIndex]

data[minIndex] ← data[i]

**endfor**

**endprocedure**

**procedure** insertionSortDosen(**in/out** data: **Dosen**, **in** n: **integer**)

**kamus**

i, j: **integer**

key: **Dosen**

**algoritma**

**for** i ← 1 **to** n **do**

key ← data[i]

j ← i - 1

**while** j ≥ 0 **and** data[j].Nama > key.Nama **do**

data[j+1] ← data[j]

j ← j - 1

**endwhile**

data[j+1] ← key

**endfor**

**endprocedure**

**function** linearSearchDosen(**in** data: **Dosen**, **in** n: **integer**, **in** target: **string**) → **integer**

**kamus**

i: **integer**

**algoritma**

**for** i ← 0 **to** n **do**

**if** data[i].Nama = target **then**

**return** i

**endif**

**endfor**

**return** -1

**endfunction**

**function** binarySearchDosen(**in** data: **Dosen**, **in** n: **integer**, **in** target: **string**) → **integer**

**kamus**

temp: **array** of **DataDosen**

bawah, atas, tengah: **integer**

**algoritma**

temp ← **new** **array** of **DataDosen** **with** size n

**copy** data[0:n] **to** temp

**sort**(temp) by Nama **in** ascending order

bawah ← 0

atas ← n - 1

**while** bawah ≤ atas **do**

tengah ← (bawah + atas) / 2

**if** temp[tengah].Nama = target **then**

**return** tengah

**else** **if** temp[tengah].Nama < target **then**

bawah ← tengah + 1

**else**

atas ← tengah - 1

**endif**

**endwhile**

**return** -1

**endfunction**

**procedure** selectionSortingKelas(**in/out** data: **Kelas**, **in** n: **integer**)

**kamus**

i, j, minIndex: **integer**

**algoritma**

**for** i ← 0 **to** n-1 **do**

minIndex ← i

**for** j ← i+1 **to** n **do**

**if** data[j].Ruangan < data[minIndex].Ruangan **then**

minIndex ← j

**endif**

**endfor**

data[i] ← data[minIndex]

data[minIndex] ← data[i]

**endfor**

**endprocedure**

**procedure** insertionSortKelas(**in/out** data: **Kelas**, **in** n: **integer**)

**kamus**

i, j: **integer**

key: **Kelas**

**algoritma**

**for** i ← 1 **to** n **do**

key ← data[i]

j ← i - 1

**while** j ≥ 0 **and** data[j].NamaKelas > key.NamaKelas **do**

data[j+1] ← data[j]

j ← j - 1

**endwhile**

data[j+1] ← key

**endfor**

**endprocedure**

**function** linearSearchKelas(**in** data: **Kelas**, **in** n: **integer**, **in** target: **string**) → **integer**

**kamus**

i: **integer**

**algoritma**

**for** i ← 0 **to** n **do**

**if** data[i].NamaKelas = target **then**

**return** i

**endif**

**endfor**

**return** -1

**endfunction**

**function** binarySearchKelas(**in** data: **Kelas**, **in** n: **integer**, **in** target: **string**) → **integer**

**kamus**

temp: **array** of **DataKelas**

bawah, atas, tengah: **integer**

**algoritma**

temp ← **new** **array** of **DataKelas** **with** size n

**copy** data[0:n] **to** temp

**sort**(temp) by Nama **in** ascending order

bawah ← 0

atas ← n - 1

**while** bawah ≤ atas **do**

tengah ← (bawah + atas) / 2

**if** temp[tengah].Nama = target **then**

**return** tengah

**else** **if** temp[tengah].Nama < target **then**

bawah ← tengah + 1

**else**

atas ← tengah - 1

**endif**

**endwhile**

**return** -1

**endfunction**

**procedure** inputDataMahasiswa(**out** data: **Mahasiswa**, **out** n: **integer**)

**kamus**

i: **integer**

**algoritma**

**output**("Masukkan jumlah mahasiswa: ")

**input** (n)

**for** i ← 0 **to** n **do**

**output**("Nama mahasiswa ke-", i+1, ": ")

**input**(data[i].Nama)

**output**("NIM mahasiswa ke-", i+1, ": ")

**input**(data[i].NIM)

**endfor**

**endprocedure**

**procedure** inputDataDosen**(out** data**: Dosen, out** n**: integer)**

**kamus**

i: **integer**

**algoritma**

**output**("Masukkan jumlah dosen: ")

**input**(n)

**for** i ← 0 **to** n **do**

**output**("Nama dosen ke-", i+1, ": ")

**input**(data[i].Nama)

**output**("NIDN dosen ke-", i+1, ": ")

**input**(data[i].NIDN)

**endfor**

**endprocedure**

**procedure** inputDataKelas(**out** data: **Kelas**, **out** n: **integer**)

**kamus**

i: **integer**

**algoritma**

**output**("Masukkan jumlah kelas: ")

**input**(n)

**for** i ← 0 **to** n **do**

**output**("Nama kelas ke-", i+1, ": ")

**input**(data[i].NamaKelas)

**output**("Ruangan kelas ke-", i+1, ": ")

**input**(data[i].Ruangan)

**endfor**

**endprocedure**

**procedure** menuSortMahasiswa(**in** data: **Mahasiswa**, **in** n: **integer**)

**kamus**

pilih: **integer**

**algoritma**

**output**("1. Selection Sort Mahasiswa")

**output**("2. Insertion Sort Mahasiswa")

**output**("Pilih: ")

**input**(pilih)

**if** pilih = 1 **then**

selectionSortingMahasiswa(data, n)

**output** ("Setelah selection sort:")

**for** i ← 0 **to** n **do**

**output**(data[i].Nama, data[i].NIM)

**endfor**

**else** **if** pilih = 2 **then**

insertionSortingMahasiswa(data, n)

**output**("Setelah insertion sort:")

**for** i ← 0 **to** n **do**

**output**(data[i].Nama, data[i].NIM)

**endfor**

**endif**

**endprocedure**

**procedure** menuSortDosen(**in** data: **Dosen**, **in** n: **integer**)

**kamus**

pilih: **integer**

**algoritma**

**output**("1. Selection Sort Dosen")

**output**("2. Insertion Sort Dosen")

**output**("Pilih: ")

**input**(pilih)

**if** pilih = 1 **then**

selectionSortingDosen(data, n)

**output** ("Setelah selection sort:")

**for** i ← 0 **to** n **do**

**output**(data[i].Nama, data[i].NIDN)

**endfor**

**else** **if** pilih = 2 **then**

insertionSortingDosen(data, n)

**output**("Setelah insertion sort:")

**for** i ← 0 **to** n **do**

**output**(data[i].Nama, data[i].NIDN)

**endfor**

**endif**

**endprocedure**

**procedure** menuSortKelas(**in** data: **Kelas**, **in** n: **integer**)

**kamus**

pilih: **integer**

**algoritma**

**output**("1. Selection Sort Kelas")

**output**("2. Insertion Sort Kelas")

**output**("Pilih: ")

**input**(pilih)

**if** pilih = 1 **then**

selectionSortingKelas(data, n)

**output** ("Setelah selection sort:")

**for** i ← 0 **to** n **do**

**output**(data[i].NamaKelas, data[i].Ruangan)

**endfor**

**else** **if** pilih = 2 **then**

insertionSortingKelas(data, n)

**output**("Setelah insertion sort:")

**for** i ← 0 **to** n **do**

**output**(data[i].NamaKelas, data[i].Ruangan)

**endfor**

**endif**

**endprocedure**

**procedure** menuSearchMahasiswa(**in** data: **Mahasiswa**, **in** n: **integer**)

**kamus**

pilih, index: **integer**

target: **string**

**algoritma**

**output**("1. Linear Search Mahasiswa")

**output**("2. Binary Search Mahasiswa")

**output**("Pilih: ")

**input**(pilih)

**output**("Masukkan nama yang dicari: ")

**input**(target)

**if** pilih = 1 **then**

index ← linearSearchMahasiswa(data, n, target)

**else** **if** pilih = 2 **then**

index ← binarySearchMahasiswa(data, n, target)

**endif**

**if** index != -1 **then**

**output**("Ditemukan pada indeks:", index)

**output**(data[index].Nama, data[index].NIM)

**else**

**output**("Tidak ditemukan")

**endif**

**endprocedure**

**procedure** menuSearchDosen(**in** data: **Dosen**, **in** n: **integer**)

**kamus**

pilih, index: **integer**

target: **string**

**algoritma**

**output**("1. Linear Search Dosen")

**output**("2. Binary Search Dosen")

**output**("Pilih: ")

**input**(pilih)

**output**("Masukkan nama yang dicari: ")

**input**(target)

**if** pilih = 1 **then**

index ← linearSearchDosen(data, n, target)

**else** **if** pilih = 2 **then**

index ← binarySearchDosen(data, n, target)

**endif**

**if** index != -1 **then**

**output**("Ditemukan pada indeks:", index)

**output**(data[index].Nama, data[index].NIDN)

**else**

**output**("Tidak ditemukan")

**endif**

**endprocedure**

**procedure** menuSearchKelas(**in** data: **Kelas**, **in** n: **integer**)

**kamus**

pilih, index: **integer**

target: **string**

**algoritma**

**output**("1. Linear Search Kelas")

**output**("2. Binary Search Kelas")

**output**("Pilih: ")

**input**(pilih)

**output**("Masukkan nama kelas yang dicari: ")

**input**(target)

**if** pilih = 1 **then**

index ← linearSearchKelas(data, n, target)

**else** **if** pilih = 2 **then**

index ← binarySearchKelas(data, n, target)

**endif**

**if** index != -1 **then**

**output**("Ditemukan pada indeks:", index)

**output**(data[index].NamaKelas, data[index].Ruangan)

**else**

**output**("Tidak ditemukan")

**endif**

**endprocedure**

**program** main

**kamus**

mahasiswa: **Mahasiswa**

dosen: **Dosen**

kelas: **Kelas**

nMahasiswa, nDosen, nKelas: **integer**

pilih, subpilih: **integer**

**algoritma**

**while** true **do**

**output**("-----Kemahasiswaan-----")

**output**("1. Input data Mahasiswa")

**output**("2. Input data Dosen")

**output**("3. Input data Kelas")

**output**("4. Sort Data")

**output**("5. Search Data")

**output**("0. Keluar")

**output**("Pilih menu: ")

**input**(pilih)

**if** pilih = 0 **then**

**break**

**endif**

**switch** pilih **do**

**case 1**:

inputDataMahasiswa(mahasiswa, nMahasiswa)

**case 2**:

inputDataDosen(dosen, nDosen)

**case 3**:

inputDataKelas(kelas, nKelas)

**case 4**:

**output**("Sort untuk: 1. Mahasiswa 2. Dosen 3. Kelas")

**output**("Pilih: ")

**input**(subpilih)

**if** subpilih = 1 **then**

menuSortMahasiswa(mahasiswa, nMahasiswa)

**else** **if** subpilih = 2 **then**

menuSortDosen(dosen, nDosen)

**else** **if** subpilih = 3 **then**

menuSortKelas(kelas, nKelas)

**endif**

**case 5**:

**output**("Search untuk: 1. Mahasiswa 2. Dosen 3. Kelas")

**output**("Pilih: ")

**input**(subpilih)

**if** subpilih = 1 **then**

menuSearchMahasiswa(mahasiswa, nMahasiswa)

**else** **if** subpilih = 2 **then**

menuSearchDosen(dosen, nDosen)

**else** **if** subpilih = 3 **then**

menuSearchKelas(kelas, nKelas)

**endif**

**endswitch**

**output**("")

**endwhile**

**endprogram**