**TUGAS MODUL PRAKTIKUM**

**MATA KULIAH ANALISIS ALGORITMA**



Disusun oleh :

Fahmi Izzuddin Abdurrahman

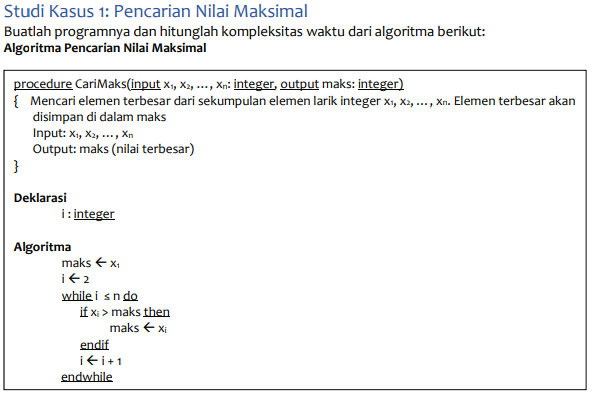
140810170062

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

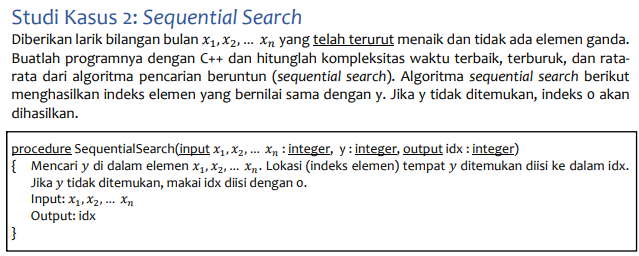
**2019**

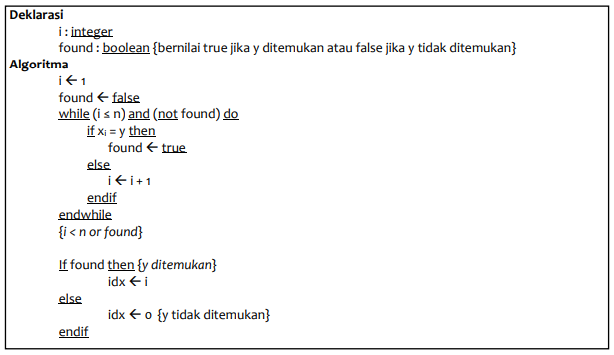


Jawaban :

Tn = t1 + t2 + t3 + t4 + t5 + t6

= 1 + 1 + n + n + n + n = 4n + 2





Jawaban :

Kasus terbaik terjadi jika x1 = y

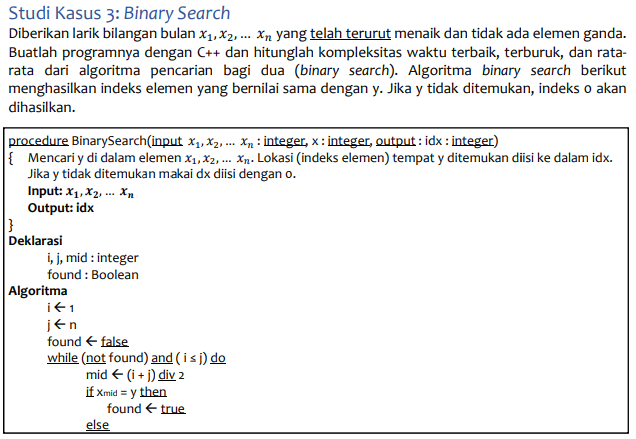
Tmin(n) = 1

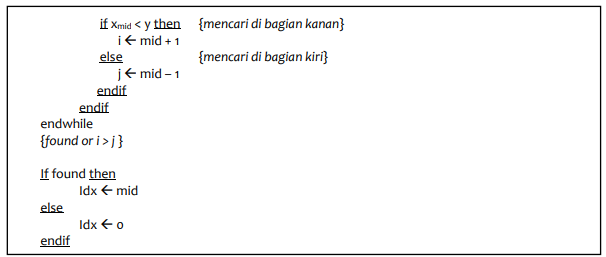
Kasus terburuk terjadi jika xn = y atau y tidak ditemukan

Tmax(n) = n

Kasus rata-rata terjadi jika x ditemukan pada posisi ke-j, maka operasi perbandingan akan dilakukan sebanyak j kali

Tavg(n) = (1 + 2 + 3 + … + n)/ n = (1 / 2n(1 + n))/ n = (n + 1)/ 2

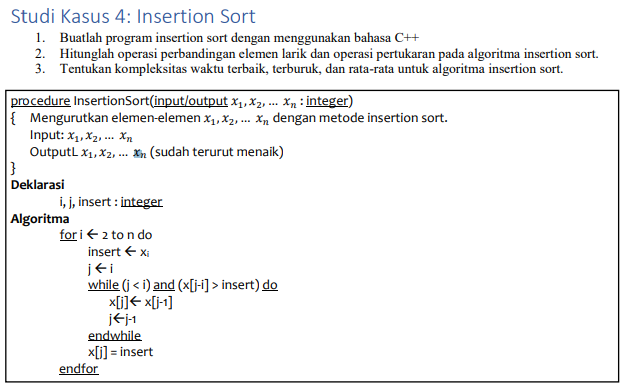




Jawaban :

Kasus Terbaik : Tmin(n) = 1

Kasus Terburuk : Tmax(n) = 2log n

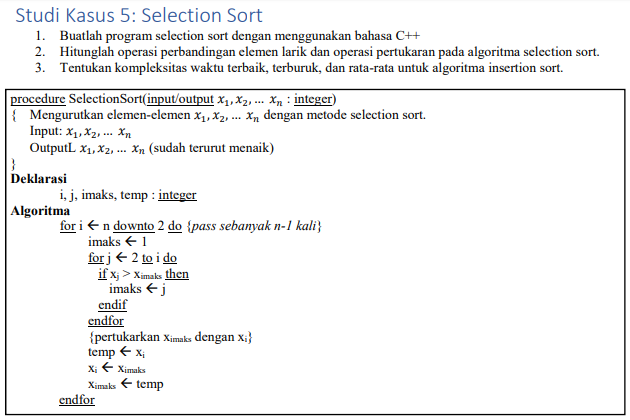


Jawaban :

Loop sementara dijalankan hanya jika I > j dan x[i] < x[j]. jumlah total iterasi loop sementara (untuk semua nilai i) sama dengan jumlah inversi.

Kompleksitas waktu keseluruhn dari jenis penyisipan adalah O(n + f(n)) di mana f(n) adalah jumlah inversi. Jika jumlah inversi adalah O(n), maka kompleksitas waktu dari jenis penyisipan adalah O(n).

Dalam kasus terburuk, bisa ada inversi n \* (n – 1)/ 2. Kasus terburuk terjadi ketika array diurutkan dalam urutan terbalik. Jadi kompleksitas waktu kasus terburuk dari jenis penyisipan adalah O (n2)



Jawaban :

Jumlah operasi perbandingan element untuk setiap pass ke-i

i = 1 🡪 jumlah perbandingan = n – 1

i = 2 🡪 jumlah perbandingan = n – 2

i = 3 🡪 jumlah perbandingan = n – 3

:

i = k 🡪 jumlah perbandingan = n – k

:

i = n – 1 🡪 jumlah perbandingan = 1

Jumlah seluruh operasi perbandingan elemen-elemen larik adalah T(n) = (n – 1) + (n – 2) + … + 1 ini adalah kompleksitas waktu untuk kasus terbaik dan terburuk, karena algoritma urut tidak bergantung pada batasan apakah data masukannya sudah terurut atau acak

Jumlah operasi pertukaran

Untuk setiap I daari 1 sampai n – 1, terjadi satu kali pertukaran elemen, sehingga jumlah operasi pertukaran seluruhnya adalah T(n) = n – 1.

Jadi, algoritma pengurutan maksimum membutuhkan n(n – 1)/ 2 buah operasi perbandingan elemen dan n – 1 operasi pertukaran