**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**TEKNIK INFORMATIKA**

**Prak-07**



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

**Pseudocode (Nomor 02)**

**Kamus/Deklarasi Variabel FatirSearch::FatirSearch01(int FatirA[], int FatirJmlElemenArray, int FatirElemen)**

-

**Algoritma/Deskripsi FatirSearch::FatirSearch01(int FatirA[], int FatirJmlElemenArray, int FatirElemen)**

int Fatirflag = -1

for (int Fatircount = 0; Fatircount < FatirJmlElemenArray; Fatircount++)

if (FatirElemen== FatirA[Fatircount])

Fatirflag = Fatircount

break

endif

endfor

return Fatirflag

**Kamus/Deklarasi Variabel**

-

**Algoritma/Deskripsi**

class FatirSearch

public:

static int FatirSearch01(int FatirA[], int FatirJmlElemenArray, int FatirElemen)

const int FatirJmlElemenArray = 11

int Fatirtika[FatirJmlElemenArray] = { 22, 61, 15, 66, 18, 25, 34, 87, 55, 45, 10 }

for (int Fatircount = 0; Fatircount < FatirJmlElemenArray; Fatircount++)

print((Fatircount+1),( Fatirtika[Fatircount]))

endfor

int FatirSearchElemen = 15

int Fatirflag = 0

print(FatirSearchElemen)

Fatirflag = FatirSearch::FatirSearch01(Fatirtika, FatirJmlElemenArray, FatirSearchElemen)

if (Fatirflag != -1)

print(Fatirflag+1)

else

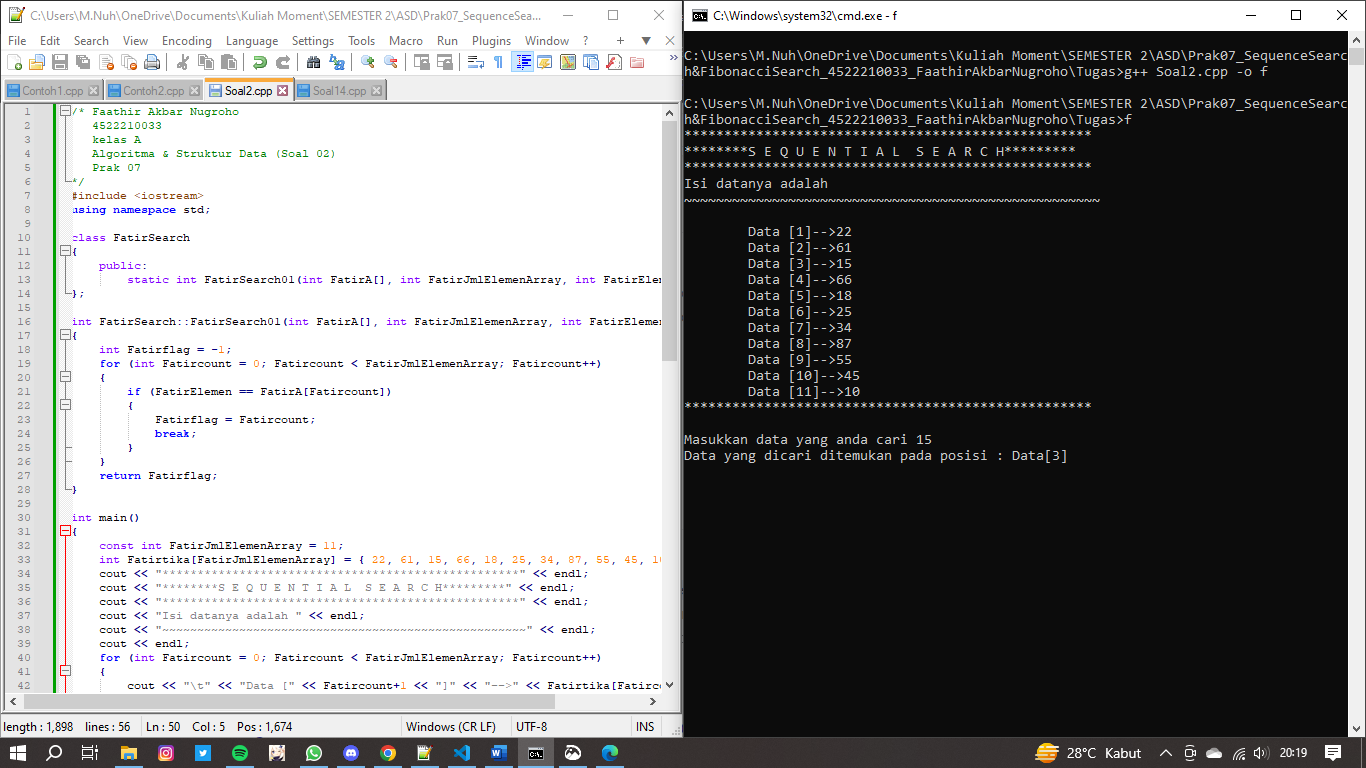
print(“Data yang anda cari tidak ditemukan”)

endif

**Algoritma/Bahasa Natural (Nomor 02)**

1. Membuat function FatirSearch::FatirSearch01(int FatirA[], int FatirJmlElemenArray, int FatirElemen)
2. Fatirflag = -1
3. Fatircount = 0
4. Selama (Fatircount < FatirJmlElemenArray), maka kerjakan baris 5 s.d 7, kalau tidak kerjakan baris 8
5. Jika ((FatirElemen == FatirA[Fatircount])), maka kerjakan baris 6, kalau tidak kerjakan baris 7
6. Fatirflag = Fatircount
7. Fatircount++
8. Kembali ke Fatirflag
9. Mendeklarasikan class (class FatirSearch(static int FatirSearch01(int FatirA[], int FatirJmlElemenArray, int FatirElemen))) dengan tipe akses secara public
10. FatirJmlElemenArray = 11
11. Fatirtika[FatirJmlElemenArray] = { 22, 61, 15, 66, 18, 25, 34, 87, 55, 45, 10 }
12. Fatircount = 0
13. Selama (Fatircount < FatirJmlElemenArray), maka kerjakan baris 14 s.d 15, kalau tidak kerjakan baris 16
14. Menampilkan isi/nilai variabel ((Fatircount+1),( Fatirtika[Fatircount]))
15. Fatircount++
16. FatirSearchElemen = 15
17. Fatirflag = 0
18. Menampilkan isi/nilai variabel FatirSearchElemen
19. Fatirflag = FatirSearch::FatirSearch01(Fatirtika, FatirJmlElemenArray, FatirSearchElemen)
20. Jika (Fatirflag != -1), maka kerjakan baris 21, kalau tidak kerjakan baris 22
21. Menampilkan isi/nilai variabel (Fatirflag+1)
22. Menampilkan (“Data yang anda cari tidak ditemukan”)
23. Selesai

**Program (Nomor 02)**



**Pseudocode (Nomor 14)**

**Kamus/Deklarasi Variabel Function FatirFibonacciSearch::Fatirfibonacci(int \*FatirF)**

-

**Algoritma/Deskripsi** **Function FatirFibonacciSearch::Fatirfibonacci(int \*FatirF)**

FatirF[0] = 1

FatirF[1] = 1

for (int Fatiri = 2; Fatiri < FatirMAXSIZE; Fatiri++)

FatirF[Fatiri] = FatirF[Fatiri - 2] + FatirF[Fatiri - 1]

endfor

**Kamus/Deklarasi Variabel FatirFibonacciSearch::Fatirfibonacci\_search(int \*Fatira, int Fatirkey, int Fatirn)**

FatirF[] = int

**Algoritma/Deskripsi FatirFibonacciSearch::Fatirfibonacci\_search(int \*Fatira, int Fatirkey, int Fatirn)**

int FatirRendah = 0, FatirTinggi = Fatirn - 1

int Fatirmid = 0

int Fatirk = 0

Fatirfibonacci(FatirF)

while (Fatirn > FatirF[Fatirk] - 1)

++Fatirk

endwhile

for (int Fatiri = Fatirn; Fatiri < FatirF[Fatirk] - 1; ++Fatiri)

Fatira[Fatiri] = Fatira[FatirTinggi]

endfor

while (FatirRendah <= FatirTinggi)

Fatirmid = FatirRendah + FatirF[Fatirk - 1] - 1

if (Fatira[Fatirmid] > Fatirkey)

FatirTinggi = Fatirmid - 1

Fatirk = Fatirk - 1

else if (Fatira[Fatirmid] < Fatirkey)

FatirRendah = Fatirmid + 1

Fatirk = Fatirk - 2

else

if (Fatirmid <= FatirTinggi)

return Fatirmid

else

return -1

endif

endif

endwhile

return -1

**Kamus/Deklarasi Variabel**

**-**

**Algoritma/Deskripsi**

class FatirFibonacciSearch

public:

static void Fatirfibonacci(int \*FatirF)

static int Fatirfibonacci\_search(int \*Fatira, int Fatirkey, int Fatirn)

int Fatira[FatirMAXSIZE] = {1, 4, 6, 8, 9, 11, 23, 35, 47, 51, 55, 63, 64, 78, 88, 95, 99}

int Fatirk = 35

print(Fatirk)

int Fatirpos = FatirFibonacciSearch::Fatirfibonacci\_search(Fatira, Fatirk, 17)

if (Fatirpos != -1)

print((Fatirpos + 1),( Fatirk))

else

print("Data tidak ditemukan")

endif

**Algoritma/Bahasa Natural (Nomor 14)**

1. Membuat function FatirFibonacciSearch::Fatirfibonacci(int \*FatirF)
2. FatirF[0] = 1
3. FatirF[1] = 1
4. Fatiri = 2
5. Selama (Fatiri < FatirMAXSIZE), maka kerjakan baris 6 s.d 7, kalau tidak kerjakan baris 8
6. FatirF[Fatiri] = FatirF[Fatiri - 2] + FatirF[Fatiri - 1]
7. Fatiri++
8. Membuat function FatirFibonacciSearch::Fatirfibonacci\_search(int \*Fatira, int Fatirkey, int Fatirn)
9. FatirRendah = 0
10. FatirTinggi = Fatirn - 1
11. Fatirmid = 0
12. Fatirk = 0
13. FatirF[FatirMAXSIZE]
14. Memanggil function Fatirfibonacci(FatirF)
15. Selama (Fatirn > FatirF[Fatirk] - 1), maka kerjakan baris 16, kalau tidak kerjakan baris 17
16. ++Fatirk
17. Fatiri = Fatirn
18. Selama (Fatiri < FatirF[Fatirk] - 1), maka kerjakan baris 19 s.d 20, kalau tidak kerjakan baris 21
19. Fatira[Fatiri] = Fatira[FatirTinggi]
20. ++Fatiri
21. Selama (FatirRendah <= FatirTinggi), maka kerjakan baris 22 s.d 31, kalau tidak kerjakan baris 32
22. Fatirmid = FatirRendah + FatirF[Fatirk - 1] - 1
23. Jika (Fatira[Fatirmid] > Fatirkey), maka kerjakan baris 24 s.d 25, kalau tidak kerjakan baris 26
24. FatirTinggi = Fatirmid - 1
25. Fatirk = Fatirk - 1
26. Jika (Fatira[Fatirmid] < Fatirkey), maka kerjakan baris 27 s.d 28, kalau tidak kerjakan baris 29
27. FatirRendah = Fatirmid + 1
28. Fatirk = Fatirk - 2
29. Jika (Fatirmid <= FatirTinggi), maka kerjakan baris 30, kalau tidak kerjakan baris 31
30. Kembali ke Fatirmid
31. Kembali -1
32. Kembali -1
33. Mendeklarasikan class (class FatirFibonacciSearch(static void Fatirfibonacci(int \*FatirF)), static int Fatirfibonacci\_search(int \*Fatira, int Fatirkey, int Fatirn))
34. ))
35. Fatira[FatirMAXSIZE] = {1, 4, 6, 8, 9, 11, 23, 35, 47, 51, 55, 63, 64, 78, 88, 95, 99}
36. Fatirk = 35
37. Menampilkan isi/nilai variabel Fatirk
38. Memanggil function Fatirpos = FatirFibonacciSearch::Fatirfibonacci\_search(Fatira, Fatirk, 17)
39. Jika (Fatirpos != -1), maka kerjakan baris 40, kalau tidak kerjakan baris 41
40. Menampilkan isi/nilai variabel ((Fatirpos + 1),( Fatirk))
41. Menampilkan ("Data tidak ditemukan")
42. Selesai

**Program (Nomor 14)**

