**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**TEKNIK INFORMATIKA**

**(Contoh)**



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

**Pseudocode (Contoh 1)**

**Kamus/Deklarasi Function**

FatirBUAT\_SIMPUL(int FatirX), FatirAWAL(), FatirCETAK(), FatirINSERTKANAN() = void

**Kamus/Deklarasi Variabel Function FatirBUAT\_SIMPUL(int FatirX)**

**-**

**Algoritma/Deskripsi Function FatirBUAT\_SIMPUL(int FatirX)**

FatirP=(FatirSimpul\*) malloc(sizeof(FatirSimpul))

if(FatirP != NULL)

FatirP -> FatirINFO = FatirX

else

print("Pembuatan FatirSimpul tidak berhasil")

exit(1)

endif

**Kamus/Deklarasi Variabel Function FatirAWAL()**

-

**Algoritma/Deskripsi Function FatirAWAL()**

FatirFIRST = FatirP

FatirLAST = FatirP

FatirP 🡪 FatirLINK = NULL

**Kamus/Deklarasi Variabel Function FatirCETAK()**

FatirX = int

**Algoritma/Deskripsi Function FatirCETAK()**

FatirQ = FatirFIRST

while(FatirQ != NULL)

FatirX = FatirQ 🡪 FatirINFO

print(FatirX)

FatirQ = FatirQ 🡪 FatirLINK

endwhile

**Kamus/Deklarasi Variabel**

FatirX, FatirI, FatirA[7] = int

**Algoritma/Deskripsi**

struct Node (FatirINFO, struct Node \*FatirLINK)

typedef struct Node FatirSimpul

FatirSimpul \*FatirP, \*FatirFIRST, \*FatirLAST, \*FatirQ

FatirA[7] = {22,28,7,11,66,63,10}

FatirFIRST = NULL

FatirI=0

FatirX=FatirA[FatirI]

FatirBUAT\_SIMPUL(FatirX)

FatirAWAL()

for(FatirI=1; FatirI<=6; FatirI++)

FatirX=FatirA[FatirI]

FatirBUAT\_SIMPUL(FatirX)

FatirINSERTKANAN()

endfor

FatirCETAK()

FatirX=100

FatirBUAT\_SIMPUL(FatirX)

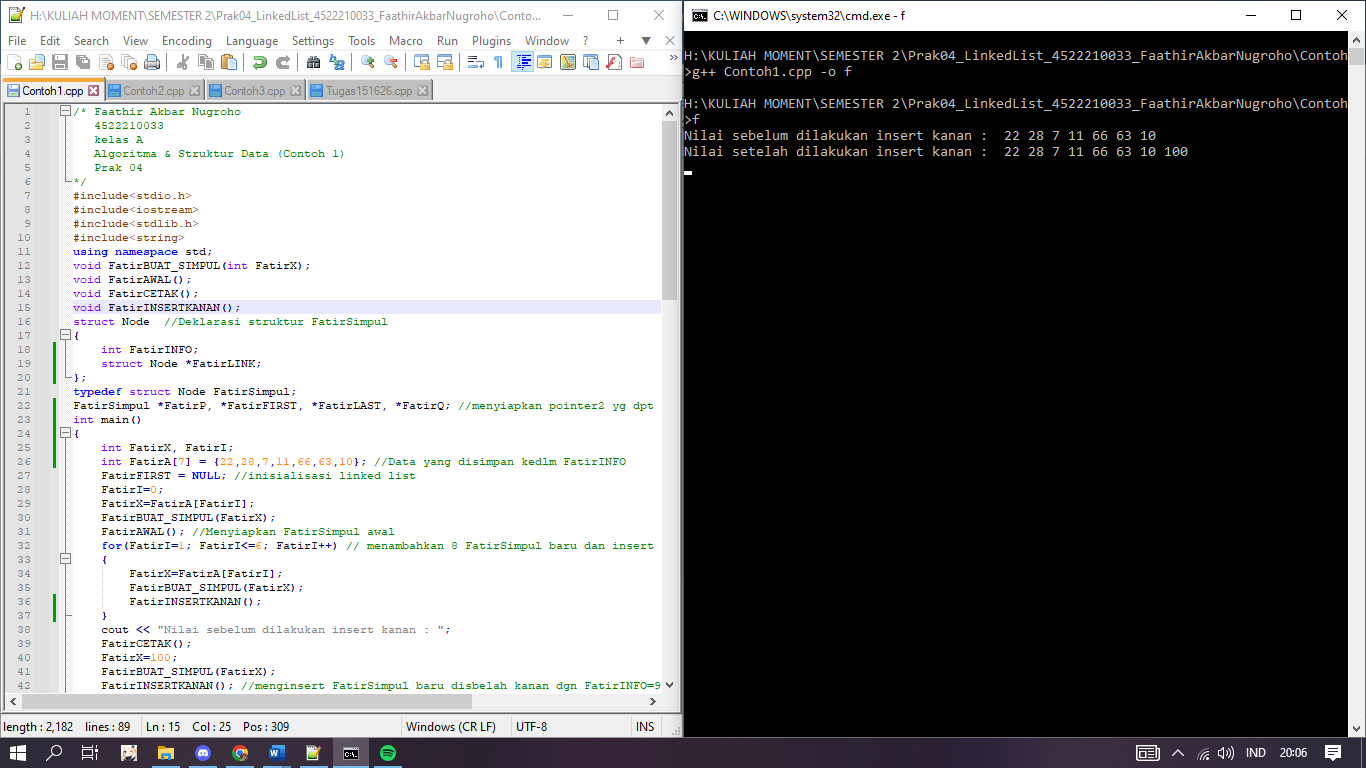
FatirINSERTKANAN()

FatirCETAK()

**Algoritma/Bahasa Natural (Contoh 1)**

1. Mendeklarasikan struktur (struct Node (FatirINFO, struct Node \*FatirLINK)
2. Membuat alias pada (struct Node FatirSimpul)
3. Membuat objek pada FatirSimpul menjadi \*FatirP
4. Membuat objek pada FatirSimpul menjadi \*FatirFIRST
5. Membuat objek pada FatirSimpul menjadi \*FatirLAST
6. Membuat objek pada FatirSimpul menjadi \*FatirQ
7. Membuat function FatirBUAT\_SIMPUL(int FatirX)
8. FatirP=(FatirSimpul\*) malloc(sizeof(FatirSimpul))
9. Jika (FatirP != NULL), maka kerjakan baris 4, jika tidak kerjakan baris 5
10. FatirP -> FatirINFO = FatirX
11. Mencetak ("Pembuatan FatirSimpul tidak berhasil")
12. Membuat function FatirAWAL
13. FatirFIRST = FatirP
14. FatirLAST = FatirP
15. FatirP -> FatirLINK = NULL
16. Membuat function FatirINSERTKANAN
17. FatirLAST -> FatirLINK = FatirP
18. FatirLAST = FatirP
19. FatirP -> FatirLINK = NULL
20. Membuat function FatirCETAK
21. FatirQ = FatirFIRST
22. Selama (FatirQ != NULL), maka kerjakan baris 23 s.d 25, jika tidak kerjakan baris 26
23. FatirX = FatirQ -> FatirINFO
24. Mencetak/menampilkan isi nilai variabel FatirX
25. FatirQ = FatirQ -> FatirLINK
26. FatirA[7] = {22,28,7,11,66,63,10}
27. FatirFIRST = NULL
28. FatirI = 0
29. FatirX = FatirA[FatirI]
30. Memanggil function FatirBUAT\_SIMPUL(FatirX)
31. Memanggil function FatirAWAL()
32. FatirI = 1
33. Selama (FatirI<=6), maka kerjakan baris ? s.d ?, jika tidak kerjakan baris ?
34. FatirX = FatirA[FatirI]
35. Memanggil function FatirBUAT\_SIMPUL(FatirX)
36. Memanggil function FatirINSERTKANAN()
37. FatirI++
38. Memanggil function FatirCETAK()
39. FatirX = 100
40. Memanggil function FatirBUAT\_SIMPUL(FatirX)
41. Memanggil function FatirINSERTKANAN()
42. Memanggil function FatirCETAK()
43. Selesai

**Program (Contoh 1)**



**Pseudocode (Contoh 2)**

**Kamus/Deklarasi Variabel Function FatirLLIST :: FatirCETAK1()**

-

**Algoritma/Deskripsi Function FatirLLIST :: FatirCETAK1()**

Node \*FatirQ

FatirQ = FatirFIRST

while (FatirQ != NULL)

print(FatirQ -> FatirData)

FatirQ = FatirQ -> FatirLINK

endwhile

**Kamus/Deklarasi Variabel Function FatirLLIST :: FatirCETAK2()**

-

**Algoritma/Deskripsi Function FatirLLIST :: FatirCETAK2()**

Node \*FatirQ

FatirQ = FatirFIRST

while (FatirQ != NULL)

print(FatirQ -> FatirData)

FatirQ = FatirQ -> FatirLINK

endwhile

**Kamus/Deklarasi Variabel int FatirLLIST :: FatirINSERTKIRI(int Fatirdt)**

Fatirdt = int

**Algoritma/Deskripsi int FatirLLIST :: FatirINSERTKIRI(int Fatirdt)**

Node \*FatirP

FatirP = new Node

FatirP -> FatirLINK = FatirFIRST

FatirP -> FatirData = Fatirdt

FatirFIRST = FatirP

**Kamus/Deklarasi Variabel**

FatirINSERTKIRI, FatirData = int

**Algoritma/Deskripsi**

Private: class FatirLLIST (Node \*FatirFIRST, \*FatirLAST, \*FatirQ)

Public: class FatirLLIST(FatirLLIST(), FatirINSERTKIRI(FatirData), FatirCETAK1(), FatirCETAK2())

FatirLLIST FatirXx

FatirXx.FatirINSERTKIRI(22)

FatirXx.FatirINSERTKIRI(66)

FatirXx.FatirINSERTKIRI(28)

FatirXx.FatirINSERTKIRI(63)

FatirXx.FatirINSERTKIRI(10)

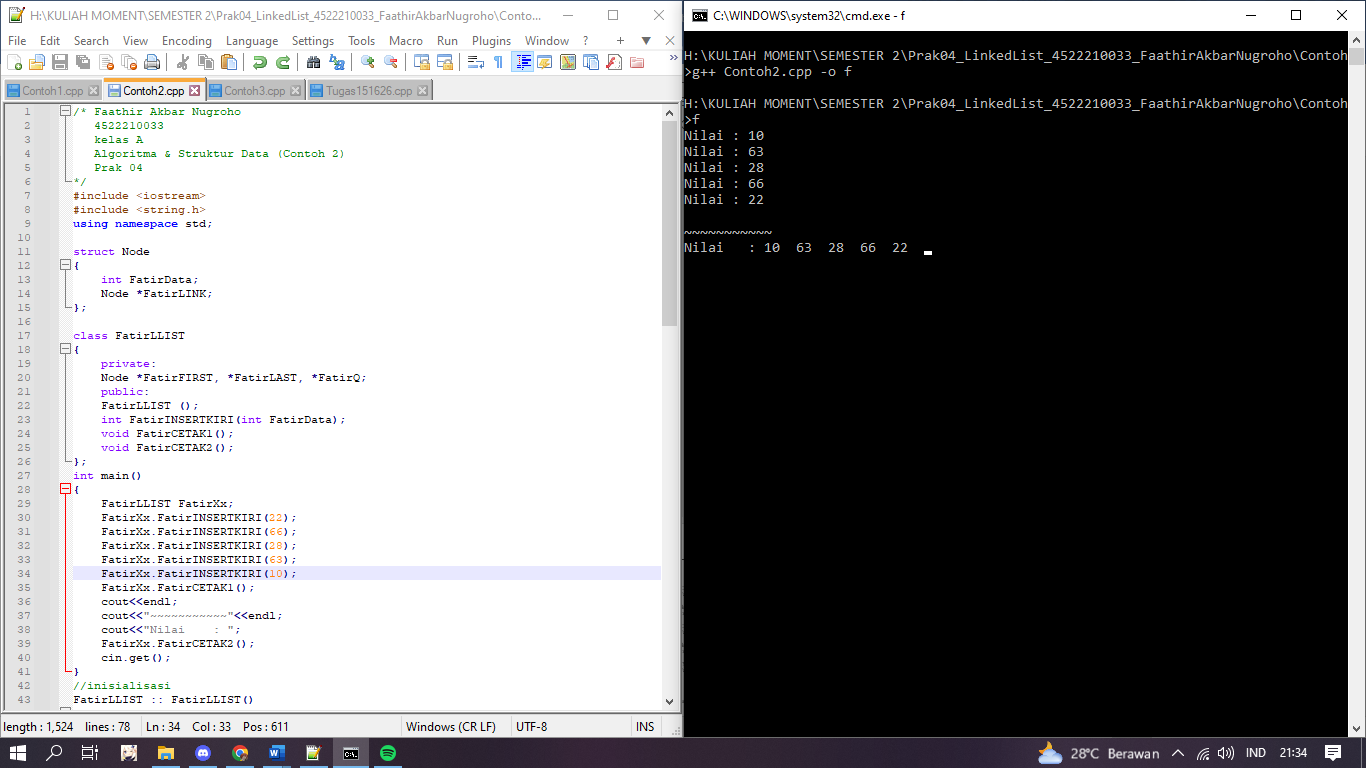
FatirXx.FatirCETAK1()

FatirXx.FatirCETAK2()

**Algoritma/Bahasa Natural (Contoh 2)**

1. Membuat function FatirLLIST :: FatirCETAK1()
2. Membuat objek pada Node menjadi \*FatirQ
3. FatirQ = FatirFIRST
4. Selama (FatirQ != NULL), maka kerjakan baris 5 s.d 6, jika tidak kerjakan baris 7
5. Mencetak/menampilkan anggota struktur (FatirQ -> FatirData)
6. FatirQ = FatirQ -> FatirLINK
7. Membuat function FatirLLIST :: FatirCETAK2()
8. Membuat objek pada Node menjadi \*FatirQ
9. FatirQ = FatirFIRST
10. Selama (FatirQ != NULL), maka kerjakan baris 11 s.d 12, jika tidak kerjakan baris 13
11. Mencetak/menampilkan anggota struktur (FatirQ -> FatirData)
12. FatirQ = FatirQ -> FatirLINK
13. Membuat function FatirLLIST :: FatirINSERTKIRI(int Fatirdt)
14. Membuat objek pada Node menjadi \*FatirQ
15. FatirP = new Node
16. FatirP -> FatirLINK = FatirFIRST
17. FatirP -> FatirData = Fatirdt
18. FatirFIRST = FatirP
19. Mendefinisikan FatirLLIST menjadi FatirXx
20. FatirXx.FatirINSERTKIRI(22)
21. FatirXx.FatirINSERTKIRI(66)
22. FatirXx.FatirINSERTKIRI(28)
23. FatirXx.FatirINSERTKIRI(63)
24. FatirXx.FatirINSERTKIRI(10)
25. Memanggil function FatirXx.FatirCETAK1()
26. Memanggil function FatirXx.FatirCETAK2()
27. Selesai

**Program (Contoh 2)**



**Pseudocode (Contoh 3)**

**Kamus/Deklarasi Function Fatirinit(void)**

-

**Algoritma/Deskripsi Function Fatirinit(void)**

FatirAwal = -1

FatirAkhir = -1

**Kamus/Deklarasi Function FatirEnqueue(struct FatirBiodata FatirMhs)**

-

**Algoritma/Deskripsi Function FatirEnqueue(struct FatirBiodata FatirMhs)**

if(FatirEmpty() == true)

FatirAwal = 0

FatirAkhir = 0

FatirQueue[FatirAwal] = FatirMhs

else

if(FatirFull() !=true)

FatirAkhir++

FatirQueue[FatirAkhir] = FatirMhs

else

print("Queue sudah penuh... \n")

endif

**Kamus/Deklarasi Function struct FatirBiodata FatirDequeue(void)**

FatirI = int

**Algoritma/Deskripsi Function struct FatirBiodata FatirDequeue(void)**

if(FatirEmpty() != true)

FatirMhs = FatirQueue[FatirAwal]

for(FatirI=FatirAwal; FatirI<=FatirAkhir; FatirI++)

FatirQueue[FatirI] = FatirQueue[FatirI+1]

FatirAkhir--

return(FatirMhs)

endfor

else

print("Queue kosong...\n")

endif

**Kamus/Deklarasi Function int FatirFull(void)**

-

**Algoritma/Deskripsi Function int FatirFull(void)**

if(FatirAkhir==FatirMAX-1) return(true)

else return(false)

**Kamus/Deklarasi Function int FatirEmpty(void)**

-

**Algoritma/Deskripsi Function** **int FatirEmpty(void)**

if(FatirAkhir== -1) return(true)

else return(false)

**Kamus/Deklarasi Variabel Function void FatirBaca(void)**

FatirI = int

**Algoritma/Deskripsi Function void FatirBaca(void)**

if(FatirEmpty() != true)

for(FatirI=FatirAwal; FatirI<=FatirAkhir; FatirI++)

FatirDisplay(FatirQueue[FatirI])

endfor

else

print("Data Kosong, ", "\n")

endif

**Kamus/Deklarasi Variabel Function struct FatirBiodata FatirEntriData(void)**

-

**Algoritma/Deskripsi Function struct FatirBiodata FatirEntriData(void)**

FatirBiodata FatirMhs

input(FatirMhs.FatirNama)

input(FatirMhs.FatirUsia)

return(FatirMhs)

**Kamus/Deklarasi Variabel Function void FatirDisplay(struct FatirBiodata FatirMhs)**

-

**Algoritma/Deskripsi Function void FatirDisplay(struct FatirBiodata FatirMhs)**

print(FatirMhs.FatirNama)

print(FatirMhs.FatirUsia)

**Kamus/Deklarasi Variabel**

FatirPilih = char

**Algoritma/Deskripsi**

Fatirinit()

do

input(FatirPilih)

switch(FatirPilih)

case '1' : FatirMhs = FatirEntriData();FatirEnqueue(FatirMhs) ;break;

case '2' : FatirMhs = FatirDequeue (); FatirDisplay(FatirMhs) ;break;

case '3' : FatirBaca() ;break;

case '4' : Fatirinit() ;break;

case '5' : ;break;

default : print("Pilih 1 sd 5 ! ")

while (FatirPilih != '5')

endwhile

**Algoritma/Bahasa Natural (Contoh 3)**

1. Membuat function Fatirinit(void)
2. FatirAwal = -1
3. FatirAkhir = -1
4. Membuat function FatirEnqueue(struct FatirBiodata FatirMhs)
5. Selama (FatirEmpty() == true), maka kerjakan baris 6 s.d 8, jika tidak kerjakan baris 9
6. FatirAwal = 0
7. FatirAkhir = 0
8. FatirQueue[FatirAwal] = FatirMhs
9. Selama (FatirFull() !=true), maka kerjakan baris 10 s.d 11, jika tidak kerjakan baris 12
10. FatirAkhir++
11. FatirQueue[FatirAkhir] = FatirMhs
12. Mencetak/menampilkan ("Queue sudah penuh... \n"”)
13. Membuat function struct FatirBiodata FatirDequeue(void)
14. Selama (FatirEmpty() != true), maka kerjakan baris 15 s.d 20 , jika tidak kerjakan baris 22
15. FatirMhs = FatirQueue[FatirAwal]
16. FatirI=FatirAwal
17. Selama (FatirI<=FatirAkhir), maka kerjakan baris 18 s.d 20, jika tidak kerjakan baris 21
18. FatirQueue[FatirI] = FatirQueue[FatirI+1]
19. FatirAkhir--
20. return(FatirMhs)
21. FatirI++
22. Mencetak/menampilkan (“Queue kosong...\n")
23. Membuat function int FatirFull(void)
24. Selama (FatirAkhir==FatirMAX-1), maka kerjakan baris 25, jika tidak kerjakan baris 26
25. return(true)
26. return(false)
27. Membuat function int FatirEmpty(void)
28. Selama (FatirAkhir== -1), maka kerjakan baris 29, jika tidak kerjakan baris 30
29. return(true)
30. return(false)
31. Membuat function void FatirBaca(void)
32. Selama (FatirEmpty() != true), maka kerjakan baris 33 s.d 35, jika tidak kerjakan baris 37
33. FatirI=FatirAwal
34. Selama (FatirI<=FatirAkhir), maka kerjakan baris 35, jika tidak kerjakan baris 36
35. FatirDisplay(FatirQueue[FatirI])
36. FatirI++
37. Mencetak/menampilkan (“"Data Kosong, ", "\n")
38. Membuat function struct FatirBiodata FatirEntriData(void)
39. Mendefinisikan struct FatirBiodata ke FatirMhs
40. Memasukkan isi/nilai pengaksesan anggota struktur (FatirMhs.FatirNama)
41. Memasukkan isi/nilai pengaksesan anggota struktur (FatirMhs.FatirUsia)
42. return(FatirMhs)
43. Membuat function void FatirDisplay(struct FatirBiodata FatirMhs)
44. Menampilkan isi/nilai pengaksesan anggota struktur (FatirMhs.FatirNama)
45. Menampilkan isi/nilai pengaksesan anggota struktur (FatirMhs.FatirUsia).

**Algoritma/Bahasa Natural (Contoh 3)**

1. Memanggil function Fatirinit()
2. Memasukkan isi/nilai variabel FatirPilih
3. Jika (FatirPilih = 1) maka kerjakan baris 49 s.d 50, kalau tidak kerjakan baris 51
4. FatirMhs = FatirEntriData()
5. Memanggil function FatirEnqueue(FatirMhs)
6. Jika (FatirPilih = 2) maka kerjakan baris 52 s.d 53, kalau tidak kerjakan baris 54
7. FatirMhs = FatirDequeue ()
8. Memanggil function FatirDisplay(FatirMhs)
9. Jika (FatirPilih = 3) maka kerjakan baris 55, kalau tidak kerjakan baris 56
10. Memanggil function FatirBaca()
11. Jika (FatirPilih = 4) maka kerjakan baris 57, kalau tidak kerjakan baris 58
12. Memanggil function Fatirinit()
13. Jika (FatirPilih = 5) maka kerjakan baris 61, kalau tidak kerjakan baris 59
14. Mencetak/menampilkan ("Pilih 1 sd 5 ! ")
15. Selama (FatirPilih != '5'), maka kerjakan baris 48 s.d 59, kalau tidak kerjakan baris 61
16. Selesai

**Program (Contoh 3)**

