**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**TEKNIK INFORMATIKA**

**(Contoh)**



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

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

-

**Algoritma/Deskripsi Function FatirClear()**

for(int Fatiri=0;Fatiri<FatirHASHSIZE;Fatiri++)

FatirhashTable[Fatiri]=NULL

endfor

**Kamus/Deklarasi Variabel Function Fatirinsert(FatirHashData \*Fatirarray)**

Fatirrec=int

**Algoritma/Deskripsi Function Fatirinsert(FatirHashData \*Fatirarray)**

Fatirrec=FatirhashFunction(Fatirarray)

FatirhashTable[Fatirrec]=Fatirarray

**Kamus/Deklarasi Variabel Function Fatirsearching(FatirHashData \*Fatircari)**

Fatirrec=int

**Algoritma/Deskripsi Function Fatirsearching(FatirHashData \*Fatircari)**

Fatirrec=FatirhashFunction(Fatircari)

Fatirrec=FatirsearchingProcess(Fatircari,Fatirrec)

if(Fatirrec>=0)

print(FatirhashTable[Fatirrec]->Fatirkode)

print(FatirhashTable[Fatirrec]->Fatirnama)

else

print(“Maaf! Data tidak ditemukan”)

endif

**Kamus/Deklarasi Variabel Function FatirsearchingProcess(FatirHashData \*Fatircari, int Fatirrec)**

Fatirm, Fatirn, Fatirj=int

**Algoritma/Deskripsi Function FatirsearchingProcess(FatirHashData \*Fatircari, int Fatirrec)**

bool Fatirequal=true

if(FatirisIndexEmpty(Fatirrec)==false)

Fatirn=FatircharacterAmount(Fatircari->Fatirkode)

Fatirm=FatircharacterAmount(FatirhashTable[Fatirrec]->Fatirkode)

if(Fatirn==Fatirm)

while((Fatirequal==true)&&(Fatircari->Fatirkode[Fatirj]!=’\0’))

if(Fatirtolower(Fatircari->Fatirkode[Fatirj])!=Fatirtolower(FatirhashTable[Fatirrec]->Fatirkode[Fatirj]))

Fatirequal=false

endif

Fatirj++

endwhile

if(Fatirequal==true)

return Fatirrec

else

return -1

endif

else

return -1

endif

**Kamus/Deklarasi Variabel Function Fatirdeleting(FatirHashData \*Fatirhapus)**

Fatirrec=int

**Algoritma/Deskripsi Function Fatirdeleting(FatirHashData \*Fatirhapus)**

Fatirrec=FatirhashFunction(Fatirhapus)

Fatirrec=FatirsearchingProcess(Fatirhapus, Fatirrec)

if(Fatirrec>=0)

print(FatirhashTable[Fatirrec]->Fatirkode)

print(FatirhashTable[Fatirrec]->Fatirnama)

FatirhashTable[Fatirrec]=NULL

else

print(“Maaf! Data tidak ditemukan”)

endif

**Kamus/Deklarasi Variabel Function FatirhashFunction(FatirhashData \*Fatirarray)**

Fatirvalue, Fatirrec, Fatirn=int

**Algoritma/Deskripsi Function FatirhashFunction(FatirhashData \*Fatirarray)**

Fatirn=FatircharacterAmount(Fatirarray->Fatirkode)

for(int Fatiri=0;Fatiri<Fatirn;Fatiri++)

Fatirvalue+=Fatirarray->Fatirkode[Fatiri]

Fatirrec=Fatirvalue&FatirHASHSIZE

return Fatirrec

endfor

**Kamus/Deklarasi Variabel Function FatirisIndexEmpty(int Fatirrec)**

**-**

**Algoritma/Deskripsi Function FatirisIndexEmpty(int Fatirrec)**

if(FatirhashTable[Fatirrec]==NULL)

return true

else

return false

endif

**Kamus/Deklarasi Variabel Function FatircharacterAmount(char Fatirarray[])**

Fatirjumlah=int

**Algoritma/Deskripsi Function FatircharacterAmount(char Fatirarray[])**

for(int Fatiri=0;Fatirarray[Fatiri]!=’\0’;Fatiri++)

Fatirjumlah++

return Fatirjumlah

endfor

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

**-**

**Algoritma/Deskripsi Function FatirDisplayTable()**

for(int Fatiri=0;Fatiri<FatirHASHSIZE;Fatiri++)

if(FatirisIndexEmpty[Fatiri]==false)

print((FatirhashTable[Fatiri]->Fatirkode)&( FatirhashTable[Fatiri]->Fatirnama))

endif

endfor

**Kamus/Deklarasi Variabel**

Fatirulang, Fatirmenu=char

FatirClear(), Fatirinsert(FatirHashData \*), FatirSearching(FatirHashData \*), Fatirdeleting(FatirHashData \*), FatirDisplaytable()=void

FatirsearchingProcess(FatirHashData \*, int)=int

FatirisIndexEmpty=bool

**Algoritma/Deskripsi**

static FatirHashData \*FatirhashTable[FatirHASHSIZE]

struct FatirHashData

char Fatirkode[FatirKODEMK]

char Fatirnama[FatirNAMAMK]

do

input(Fatirmenu)  
 switch(Fatirmenu)

case ’1’:

FatirHashData \*Fatirarray

Fatirarray=new FatirHashData

input(Fatirarray->Fatirkode)

input(Fatirarray->Fatirnama)

Fatirinsert(Fatirarray) break

case ‘2’:

FatirHashData \*Fatirhapus

Fatirhapus=new FatirHashData

input(Fatirhapus->Fatirkode)

Fatirdeleting(Fatirhapus) break

case ‘3’:

FatirhashData \*Fatircari

Fatircari=new FatirHashData

input(Fatircari->Fatirkode)

Fatirsearching(Fatircari)

case ‘4’:

FatirDisplayTable() break

case ‘5’:

FatirClear()

break

case ‘x’:

case ‘X’:

Fatirulang=’T’ break

default:

print(“Pilihan diluar Menu yang tersedia”) break

while(Fatirtoupper(Fatirulang)==’Y’)

endwhile

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

1. Membuat function FatirClear()
2. Fatiri = 0
3. Selama (Fatiri < FatirHASHSIZE), maka kerjakan baris 4 s.d 5, kalau tidak kerjakan baris 6
4. FatirhashTable[Fatiri] = NULL
5. Fatiri++
6. Membuat function Fatirinsert(FatirHashData \*Fatirarray)
7. Fatirrec = FatirhashFunction(Fatirarray)
8. FatirhashTable[Fatirrec] = Fatirarray
9. Membuat function Fatirsearching(FatirHashData \*Fatircari)
10. Fatirrec = FatirhashFunction(Fatircari)
11. Fatirrec = FatirsearchingProcess(Fatircari, Fatirrec)
12. Jika (Fatirrec >= 0), maka kerjakan baris 13 s.d 14, kalau tidak kerjakan baris 15
13. Menampilkan isi/nilai variabel (FatirhashTable[Fatirrec]->Fatirkode)
14. Menampilkan isi/nilai variabel (FatirhashTable[Fatirrec]->Fatirnama)
15. Menampilkan (“Maaf! Data tidak ditemukan.”)
16. Membuat function FatirsearchingProcess(FatirHashData \*Fatircari, int Fatirrec)
17. bool Fatirequal = true
18. Jika (FatirisIndexEmpty(Fatirrec), maka kerjakan baris 19 s.d 29, kalau tidak kerjakan baris 30
19. Fatirn = FatircharacterAmount(Fatircari->Fatirkode)
20. Fatirm = FatircharacterAmount(FatirhashTable[Fatirrec]->Fatirkode)
21. Jika (Fatirn == Fatirm), maka kerjakan baris 22 s.d 28, kalau tidak kerjakan baris 29
22. Selama ((Fatirequal == true) && (Fatircari->Fatirkode[Fatirj] != '\0')), maka kerjakan baris 23 s.d 25, kalau tidak kerjakan baris 26
23. Selama (tolower(Fatircari->Fatirkode[Fatirj]) != tolower(FatirhashTable[Fatirrec]->Fatirkode[Fatirj])), maka kerjakan baris 24 s.d 25, kalau tidak kerjakan baris 26
24. Fatirequal = false
25. Fatirj++
26. Jika (Fatirequal == true), maka kerjakan baris 27, kalau tidak kerjakan baris 28
27. return Fatirrec
28. return -1
29. return -1
30. return -1
31. Membuat function Fatirdeleting(FatirHashData \*Fatirhapus)
32. Fatirrec = FatirhashFunction(Fatirhapus)
33. Fatirrec = FatirsearchingProcess(Fatirhapus, Fatirrec)
34. Jika (Fatirrec >= 0), maka kerjakan baris 35 s.d 37, kalau tidak kerjakan baris 38
35. Menampilkan isi/nilai variabel (FatirhashTable[Fatirrec]->Fatirkode)
36. Menampilkan isi/nilai variabel (FatirhashTable[Fatirrec]->Fatirnama)
37. FatirhashTable[Fatirrec] = NULL
38. Menampilkan ("Maaf Data tidak ditemukan!")
39. Membuat function FatirhashFunction(FatirHashData \*Fatirarray)
40. Fatirn = FatircharacterAmount(Fatirarray->Fatirkode)
41. int Fatiri = 0
42. Selama (Fatiri < Fatirn), maka kerjakan baris 43 s.d 44, kalau tidak kerjakan baris 45
43. Fatirvalue += Fatirarray->Fatirkode[Fatiri]
44. Fatiri++
45. Fatirrec = Fatirvalue % FatirHASHSIZE
46. return Fatirrec
47. Membuat function FatirisIndexEmpty(int Fatirrec)
48. Jika (FatirhashTable[Fatirrec] == NULL), maka kerjakan baris 49, kalau tidak kerjakan baris 50
49. return true
50. return false
51. Membuat function FatircharacterAmount(char Fatirarray[])
52. int Fatirjumlah = 0
53. int Fatiri = 0
54. Selama (Fatirarray[Fatiri] != '\0'), maka kerjakan baris 55 s.d 56, kalau tidak kerjakan baris 57
55. Fatirjumlah++
56. Fatiri++
57. return Fatirjumlah
58. Membuat function FatirDisplayTable()
59. int Fatiri = 0
60. Selama (Fatiri < FatirHASHSIZE), maka kerjakan baris 61 s.d 63, kalau tidak kerjakan baris 64
61. Jika (FatirisIndexEmpty(Fatiri) == false), maka kerjakan baris 62, kalau tidak kerjakan baris 63
62. Menampilkan isi/nilai variabel (Fatiri,( FatirhashTable[Fatiri]->Fatirkode),( FatirhashTable[Fatiri]->Fatirnama))
63. Fatiri++
64. Mendeklarasikan struct (struct FatirHashData(char Fatirkode[FatirKODEMK], char Fatirnama[FatirNAMAMK]))
65. Mendefinisikan struct (static FatirHashData \*FatirhashTable[FatirHASHSIZE])
66. char Fatirulang = 'Y'
67. Memasukkan isi/nilai variabel Fatirmenu
68. Jika (Fatirmenu=1), maka kerjakan baris 69 s.d 73, kalau tidak kerjakan baris 74
69. Mendefinisikan struct (FatirHashData \*Fatirarray)
70. Fatirarray = new FatirHashData
71. Memasukkan isi/nilai variabel (Fatirarray->Fatirkode)
72. Memasukkan isi/nilai variabel (Fatirarray->Fatirnama)
73. Memanggil function Fatirinsert(Fatirarray)
74. Jika (Fatirmenu=2), maka kerjakan baris 75 s.d 78, kalau tidak kerjakan baris 79
75. Mendefinisikan struct (FatirHashData \*Fatirhapus)
76. Fatirhapus = new FatirHashData
77. Memasukkan isi/nilai variabel (Fatirhapus->Fatirkode)
78. Memanggil function Fatirdeleting(Fatirhapus)
79. Jika (Fatirmenu=3), maka kerjakan baris 80 s.d 83, kalau tidak kerjakan baris 84
80. Mendefinisikan struct (FatirHashData \*Fatircari)
81. Fatircari = new FatirHashData
82. Memasukkan isi/nilai variabel (Fatircari->Fatirkode)
83. Memanggil function Fatirsearching(Fatircari)
84. Jika (Fatirmenu=4), maka kerjakan baris 85, kalau tidak kerjakan baris 86
85. Memanggil function FatirDisplayTable()
86. Jika (Fatirmenu=’X’&&’x’), maka kerjakan baris 87, kalau tidak kerjakan baris 88
87. Fatirulang = 'T'
88. Jika (Fatirmenu=default), maka kerjakan baris 89, kalau tidak kerjakan baris 90
89. Menampilkan ("Pilihan diluar Fatirmenu yang tersedia")
90. Selama (toupper(Fatirulang) == 'Y'), maka kerjakan baris 67 s.d 89, kalau tidak kerjakan baris 91
91. Selesai

**Program (Contoh 1)**

