**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**TEKNIK INFORMATIKA**

**(Contoh)**



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

**Pseudocode (Contoh 1)**

**Kamus/Deklarasi Variabel Function FatirCanBTree\* Fatirinit()**

Fatiri=int

**Algoritma/Deskripsi Function FatirCanBTree\* Fatirinit()**

Fatirnp = new FatirCanBTree

Fatirnp->Fatird = new int[6]

Fatirnp->FatirCanAnakPointer = new FatirCanBTree \*[7]

Fatirnp->Fatirl = true

Fatirnp->Fatirn = 0

for (Fatiri = 0; Fatiri < 7; Fatiri++)

Fatirnp->FatirCanAnakPointer[Fatiri] = NULL

endfor

return Fatirnp

**Kamus/Deklarasi Variabel Function FatirpohonB(FatirCanBTree \*Fatirp)**

Fatiri=int

**Algoritma/Deskripsi Function FatirpohonB(FatirCanBTree \*Fatirp)**

for (Fatiri = 0; Fatiri < Fatirp->Fatirn; Fatiri++)

if (Fatirp->Fatirl == false)

FatirpohonB(Fatirp->FatirCanAnakPointer[Fatiri])

endif

print(Fatirp->Fatird[Fatiri])

endfor

if (Fatirp->Fatirl == false)

FatirpohonB(Fatirp->FatirCanAnakPointer[Fatiri])

endif

**Kamus/Deklarasi Variabel Function FatirUrut(int \*Fatirp, int Fatirn)**

Fatiri, Fatirj, Fatirt=int

**Algoritma/Deskripsi Function FatirUrut(int \*Fatirp, int Fatirn)**

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

for (Fatirj = Fatiri; Fatirj <= Fatirn; Fatirj++)

if (Fatirp[Fatiri] > Fatirp[Fatirj])

Fatirt = Fatirp[Fatiri]

Fatirp[Fatiri] = Fatirp[Fatirj]

Fatirp[Fatirj] = Fatirt

endif

endfor

endfor

**Kamus/Deklarasi Variabel Function FatirPecahAnak(FatirCanBTree \*Fatirx, int Fatiri)**

Fatirj, Fatirmid=int

**Algoritma/Deskripsi Function FatirPecahAnak(FatirCanBTree \*Fatirx, int Fatiri)**

FatirCanBTree \*Fatiry

Fatirnp3 = Fatirinit()

Fatirnp3->Fatirl = true

if (Fatiri == -1)

Fatirmid = Fatirx->Fatird[2]

Fatirx->Fatird[2] = 0

Fatirx->Fatirn--

endif

return Fatirmid

**Kamus/Deklarasi Variabel Function FatirSisip(int Fatira)**

Fatiri, Fatirj, Fatirt, Fatirmid=int

**Algoritma/Deskripsi Function FatirSisip(int Fatira)**

Fatirx = Fatirr

if (Fatirx == NULL)

Fatirr = Fatirinit()

Fatirx = Fatirr

else

if (Fatirx->Fatirl == true && Fatirx->Fatirn == 6)

Fatirt = FatirPecahAnak(Fatirx, -1)

Fatirx = Fatirr

for (Fatiri = 0; Fatiri < (Fatirx->Fatirn); Fatiri++)

if ((Fatira > Fatirx->Fatird[Fatiri]) && (Fatira < Fatirx->Fatird[Fatiri + 1]))

Fatiri++

break

else

if (Fatira < Fatirx->Fatird[0])

break

else

continue

endif

endif

endfor

Fatirx = Fatirx->FatirCanAnakPointer[Fatiri]

else

while (Fatirx->Fatirl == false)

for (Fatiri = 0; Fatiri < (Fatirx->Fatirn); Fatiri++)

if ((Fatira > Fatirx->Fatird[Fatiri]) && (Fatira < Fatirx->Fatird[Fatiri + 1]))

Fatiri++

break

else

if (Fatira < Fatirx->Fatird[0])

Fatirnpl = Fatirinit()

Fatirnpl->Fatirl = false

Fatirx->Fatirl = true

for (Fatirj = 3; Fatirj < 6; Fatirj++)

Fatirnp3->Fatird[Fatirj - 3] = Fatirx->Fatird[Fatirj]

Fatirnp3->FatirCanAnakPointer[Fatirj - 3] = Fatirx->FatirCanAnakPointer[Fatirj]

Fatirnp3->Fatirn++

Fatirx->Fatird[Fatirj] = 0

Fatirx->Fatirn--

endfor

Fatirmid = Fatirx->Fatird[2]

Fatirx->Fatird[2] = 0

Fatirx->Fatirn--

for (Fatirj = 0; Fatirj < 6; Fatirj++)

Fatirx->FatirCanAnakPointer[Fatirj] = NULL

endfor

Fatirnpl->Fatird[0] = Fatirmid

Fatirnpl->FatirCanAnakPointer[Fatirnpl->Fatirn] = Fatirx

Fatirnpl->FatirCanAnakPointer[Fatirnpl->Fatirn + 1] = Fatirnp3

Fatirnpl->Fatirn++

Fatirr = Fatirnpl

else

Fatiry = Fatirx->FatirCanAnakPointer[Fatiri]

Fatirmid = Fatiry->Fatird[2]

Fatiry->Fatird[2] = 0

Fatiry->Fatirn--

Fatirx->FatirCanAnakPointer[Fatiri + 1] = Fatiry

Fatirx->FatirCanAnakPointer[Fatiri + 1] = Fatirnp3

break

endif

endif

endfor

if (Fatiri == Fatirx->Fatirn)

Fatirx = Fatirx->FatirCanAnakPointer[Fatiri]

endif

if ((Fatirx->FatirCanAnakPointer[Fatiri])->Fatirn == 6)

Fatirt = FatirPecahAnak(Fatirx, 1)

Fatirx->Fatird[Fatirx->Fatirn] = Fatirt

Fatirx->Fatirn++

continue

else

Fatirx = Fatirx->FatirCanAnakPointer[Fatiri]

endif

endwhile

endif

endif

Fatirx->Fatird[Fatirx->Fatirn] = Fatira

FatirUrut(Fatirx->Fatird, Fatirx->Fatirn)

Fatirx->Fatirn++

**Kamus/Deklarasi Variabel**

Fatiri, Fatirn, Fatirt=int

**Algoritma/Deskripsi**

struct FatirCanBTree

int \*Fatird

FatirCanBTree \*\*FatirCanAnakPointer

bool Fatirl

int Fatirn

\*Fatirr = NULL, \*Fatirnp = NULL, \*Fatirx = NULL, \*Fatirnpl = NULL, \*Fatirnp3 = NULL, \*Fatiry = NULL

input(Fatirn)

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

input(Fatirt)

FatirSisip(Fatirt)

endfor

FatirpohonB(Fatirr)

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

1. Mendeklarasikan struktur FatirBTree (\*Fatird, FatirBTree \*\*FatirAnakPointer, Fatiri, Fatirn)

2. Membuat simpul FatirBTree\* Fatirinit()

3. Fatirnp = new FatirBTree

4. Fatirnp->Fatird = new int[6]

5. Fatirnp->FatirAnakPointer = new FatirBTree \*[7]

6. Fatirnp->1 = true

7. Fatirnp->Fatirn = 0

8. Fatiri = 0

9. selama (Fatiri < 7), maka kerjakan baris 10 s.d. 12

10. Fatirnp->FatirAnakPointer[Fatiri] = NULL

11. kembalikan Fatirnp

12. Fatiri++

13. membuat fungsi FatirPohonB(FatirBTree \*Fatirp)

14. Fatiri = 0

15. selama (Fatiri < Fatirp->Fatirn), maka kerjakan baris 16 s.d. 18

16. jika (Fatirp->Fatiri == false)

17. FatirPohonB(Fatirp->FatirAnakPointer[Fatiri])

18. Fatiri++

19. mencetak nilai Fatirp->Fatird[Fatiri]

20. jika (Fatirp->Fatiri == false), maka kerjakan baris 21 s.d. 22

21. FatirPohonB(Fatirp->FatirAnakPointer[Fatiri])

22. memanggil fungsi FatirUrut

23. Fatiri = 0

24. selama (Fatiri < Fatirn), maka kerjakan baris 25 s.d. 26

25. Fatiri++

26. Fatirj = Fatiri

27. selama (Fatirj <= Fatirn), maka kerjakan baris 28

28. Fatirj++

29. jika (Fatirp[Fatiri] > Fatirp[Fatirj]), maka kerjakan baris 30 s.d. 32

30. Fatirt = Fatirp[Fatiri]

31. Fatirp[Fatiri] = Fatirp[Fatirj]

32. Fatirp[Fatirj] = Fatirt

33. memanggil fungsi FatirpecahAnak(FatirBTree \*Fatirx)

34. FatirBTree \*Fatirnpl, \*Fatirnp3, \*Fatiry

35. Fatirnp3 = Fatirinit()

36. Fatirnp3-> = true

37. jika (Fatiri == -1)

38. Fatirmid = Fatirx->Fatird[2]

39. Fatirx->Fatird[2] = 0

40. Fatirx->Fatirn--

41. Fatirnpl=Fatirinit()

42. Fatirnpl->Fatiri=false

43. Fatirx->Fatiri=true

44. Fatirj=3

45. Selama (Fatirj<6), maka kerjakan baris 46 s.d 51

46. Fatirnp3->Fatird[Fatirj-3]= Fatirx->Fatird[Fatirj]

47. Fatirnp3->FatirAnakPointer[Fatirj-3]= Fatirx- >FatirAnakPointer[Fatirj]

48. Fatirnp3->Fatirn++

49. Fatirx->Fatird[Fatirj]=0

50. Fatirx->Fatirn--

51. Fatirj++

52. Fatirj=0

53. Selama (Fatirj<6), maka kerjakan baris 54 s.d 55

54. Fatirx->FatirAnakPointer[Fatirj]=NULL

55. Fatirj++

56. Fatirnpl->Fatird[0]= Fatirmid

57. Fatirnpl->FatirAnakPointer[Fatirnpl->Fatirn]= Fatirx

58. Fatirnpl->FatirAnakPointer[Fatirnpl- >Fatirn+1]= Fatirnp3

59. Fatirnpl->Fatirn++

60. Fatirr=Fatirnpl

61. Fatiry=Fatirx->FatirAnakPointer[Fatiri]

62. Fatirmid=Fatiry->Fatird[2]

63. Fatiry->d[2]=0

64. Fatiry->Fatirn--

65. Fatirj=3

66. Selama (Fatirj<6), maka kerjakan baris 67 s.d 71

67. Fatirnp3->Fatird[Fatirj-3]= Fatiry->Fatird[Fatirj]

68. Fatirnp3->Fatirn++

69. Fatiry->Fatird[Fatirj]=0

70. Fatiry->Fatirn--

71. Fatirj++

72. Fatirx->FatirAnakPointer[Fatiri+1]= Fatiry

73. Fatirx->FatirAnakPointer[Fatiri+1]= Fatirnp3

74. kembalikan Fatirmid

75. Membuat function FatirSisip(int Fatira)

76. Fatirx=Fatirr

77. Jika (Fatirx==NULL), maka kerjakan baris 78 s.d 79

78. Fatirr=Fatirinit()

79. Fatirx=Fatirr

80. Jika (Fatirx->Fatiri==true&&Fatirx->Fatirn==6), maka kerjakan baris 81 s.d. 82

81. Fatirt=FatirpecahAnak(Fatirx, -1)

82. Fatirx=Fatirr

83. Fatiri=0

84. Selama (in), maka kerjakan baris 85

85. Jika ((Fatira>Fatirx->Fatird[Fatiri])&&( Fatirad[Fatiri+1])), maka kerjakan baris 86

86. Fatiri++

87. Jika (Fatirad[0]), maka kerjakan baris 88 s.d. 90

88. break 89. melanjutkan

90. Fatiri++

91. Fatirx=Fatirx->FatirAnakPointer[Fatiri]

92. Selama (Fatirx->Fatiri==false), maka kerjakan baris 93 s.d. 94

93. Fatiri=0

94. Selama (in)), maka kerjakan baris 95

95. Jika((Fatira>Fatirx->Fatird[Fatiri])&&(Fatirad[i+1])), maka kerjakan baris 96 s.d. 98

96. Fatiri++

97. break

98. continue

99. Jika ((Fatirx->FatirAnakPointer[Fatiri])-> Fatirn==6), maka kerjakan baris 100 s.d. 107

100. Fatirt=FatirPecahAnak(Fatirx, Fatiri)

101. Fatirx->Fatird[Fatirx->Fatirn]= Fatirt

102. Fatirx->Fatirn++

103. continue

104. Fatirx=Fatirx->FatirAnakPointer[Fatiri]

105. Fatirx->Fatird[Fatirx->Fatirn]= Fatira

106. FatirUrut(Fatirx->Fatird, Fatirx->Fatirn)

107. Fatirx->Fatirn++

108. memasukkan nilai Fatirn

109. Fatiri=0

110. Selama (Fatiri<Fatirn), maka kerjakan baris 111 s.d 112

111. Memasukkan isi/nilai variabel Fatirt

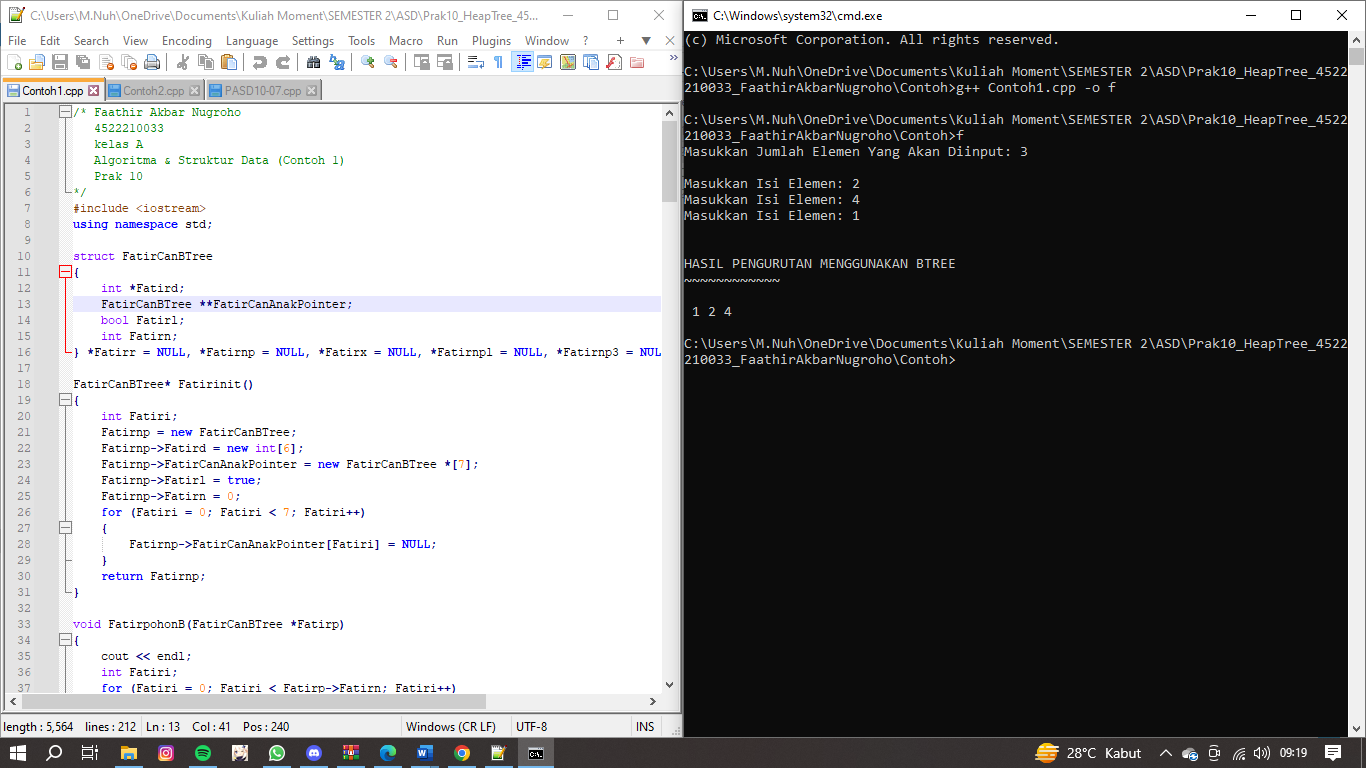
112. Memanggil function FatirSisip(Fatirt)

113. Fatiri++

114. Memanggil function FatirPohonB(Fatirr)

115. Selesai

**Program (Contoh 1)**



**Pseudocode (Contoh 2)**

**Kamus/Deklarasi Variabel Function FatirHEAP(int FatirCanArray[], int Fatirn, int Fatiri)**

Fatirtemp,=int

**Algoritma/Deskripsi Function FatirHEAP(int FatirCanArray[], int Fatirn, int Fatiri)**

int FatirCanBesar = Fatiri

int Fatirkiri = 2 \* Fatiri + 1

int Fatirkanan = 2 \* Fatiri + 2

if (Fatirkiri < Fatirn && FatirCanArray[Fatirkiri]>FatirCanArray[FatirCanBesar])

FatirCanBesar = Fatirkiri

endif

if (Fatirkanan < Fatirn && FatirCanArray[Fatirkanan]>FatirCanArray[FatirCanBesar])

FatirCanBesar = Fatirkanan

endif

if (FatirCanBesar != Fatiri)

Fatirtemp = FatirCanArray[Fatiri]

FatirCanArray[Fatiri] = FatirCanArray[FatirCanBesar]

FatirCanArray[FatirCanBesar] = Fatirtemp

FatirHEAP(FatirCanArray, Fatirn, FatirCanBesar)

endif

**Kamus/Deklarasi Variabel Function sortHEAPmae(int FatirCanArray[], int Fatirn)**

Fatirtemp=int

**Algoritma/Deskripsi Function FatirsortHEAP(int FatirCanArray[], int Fatirn)**

for (int Fatiri = Fatirn / 2 - 1; Fatiri >= 0; Fatiri--)

FatirHEAP(FatirCanArray, Fatirn, Fatiri)

endfor

for (int Fatiri = Fatirn - 1; Fatiri >= 0; Fatiri--)

Fatirtemp = FatirCanArray[0]

FatirCanArray[0] = FatirCanArray[Fatiri]

FatirCanArray[Fatiri] = Fatirtemp

FatirHEAP(FatirCanArray, Fatiri, 0)

endfor

**Kamus/Deklarasi Variabel**

Fatiri=int

**Algoritma/Deskripsi**

int FatirCanArray[]={22, 7, 66, 28, 11, 63, 24, 12, 77, 99}

int Fatirn = 10

for (Fatiri=0; Fatiri < Fatirn; Fatiri++)  
 print(FatirCanArray[Fatiri])

endfor

FatirsortHEAP(FatirCanArray,Fatirn)

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

print(FatirCanArray[Fatiri])

endfor

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

1. Membuat variabel function FatirHEAP(int FatirArray[], int Fatirn, int Fatiri)

2. Fatirbesar=Fatiri

3. FatirKiri = 2\*Fatiri+1

4. FatirKanan = 2 \* Fatiri+2

5. Jika (Fatirkiri < Fatirn && FatirArray[Fatirkiri] > FatirArray[Fatirbesar]) maka kerjakan baris 6

6. Fatirbesar = Fatirkiri

7. Jika (Fatirkiri < Fatirn && FatirArray[Fatirkiri] > FatirArray[Fatirbesar]) maka kerjakan baris 8

8. Fatirbesar = Fatirkanan

9. Jika (Fatirbesar != Fatiri) maka kerjakan baris 10

10. Fatirtemp=FatirArray[0]

11. FatirArray[0] = FatirArray[Fatiri]

12. FatirArray[Fatiri] = temp

13. FatirHEAP(FatirArray, Fatiri, 0)

14. Membuat variabel function FatirsortHEAP (int FatirArray[], int Fatirn)

15. Fatiri >= 0

16. selama Fatiri = Fatirn/2-1 maka kerjakan baris 17-19 jika tidak maka kerjakan baris 20

17. HEAPFATIR(FatirArray, Fatirn, Fatiri)

18. Fatiri—

19. Fatiri>=0

20. selama Fatiri = Fatirn-1 maka kerjakan baris 21- 25

21. Fatirtemp = FatirArray[0]

22. FatirArray[0] = FatirArray[Fatiri]

23. FatirArray[Fatiri] = Fatirtemp

24. FatirHEAP (FatirArray, Fatiri, Fatiro)

25. Fatiri--

26. FatirArray[]={22, 7, 66, 28, 11, 63, 24, 12, 77, 99}

27. Fatirn=10

28. Fatiri<Fatirn

29. selama Fatiri = 0 maka kerjakan baris 30-32

30. Mencetak/menampilkan isi variabel (FatirArray[Fatiri])

31. FatirSortHEAP(FatirArray, Fatirn)

32. Fatiri++

33. Fatiri<Fatirn

34. selama Fatiri = 0 maka kerjakan baris 35-36

35. Mencetak/menampilkan isi variabel (FatirArray[Fatiri])

36. Selesai

**Program (Contoh 2)**

