**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**TEKNIK INFORMATIKA**

**Prak-10**



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

**Pseudocode (Nomor 07)**

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

-

**Algoritma/Deskripsi Function FatirInisialisasi()**

FatirRoot = NULL

FatirP = NULL

**Kamus/Deklarasi Variabel Function FatirBuatSimpul(char FatirX)**

-

**Algoritma/Deskripsi Function FatirBuatSimpul(char FatirX)**

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

if (FatirP != NULL)

FatirP->FatirINFO = FatirX

FatirP->FatirLeft = NULL

FatirP->FatirRight = NULL

else

print(“Memory Heap Full”)

exit(1)

endif  
**Kamus/Deklarasi Variabel Function** **FatirBuatSimpulAkar()**

**-**

**Algoritma/Deskripsi Function FatirBuatSimpulAkar()**

if (FatirRoot == NULL)

if (FatirP != NULL)

FatirRoot = FatirP

FatirRoot->FatirLeft = NULL

FatirRoot->FatirRight = NULL

else

print(“FatirSimpul Belum Dibuat”)

endif

else

print(“Pohon Sudah Ada!!!!”)

endif

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

Fatiri, Fatirj, FatirFlag=int

FatirX=char

**Algoritma/Deskripsi Function** **FatirInsertUrutNomor()**

FatirFlag = 0

Fatiri = 1

Fatirj = 1

FatirQ[Fatiri] = FatirRoot

while (FatirFlag == 0 && Fatirj < 127)

input(FatirX)

if (FatirX != '0')

FatirBuatSimpul(FatirX)

FatirCurrent = FatirQ[Fatiri]

FatirCurrent->FatirLeft = FatirP

Fatirj++

FatirQ[Fatirj] = FatirP

else

FatirFlag = 1

Fatirj++

FatirQ[Fatirj] = NULL

endif

if (FatirFlag == 0)

if (FatirX != '0')

FatirBuatSimpul(FatirX)

FatirCurrent->FatirRight = FatirP

Fatirj++

FatirQ[Fatirj] = FatirP

else

FatirFlag = 1

Fatirj++

FatirQ[Fatirj] = NULL

endif

endif

Fatiri++

endwhile

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

Fatiri, Fatirj, Fatirn, FatirCounter=int

FatirX=char

**Algoritma/Deskripsi Function FatirBacaUrutNomor()**

Fatiri = 1

Fatirj = 1

Fatirn = 1

FatirCounter = 0

while (FatirQ[Fatiri] != NULL)

FatirCurrent = FatirQ[Fatiri]

print(FatirCurrent->FatirINFO)

FatirCounter++

if (FatirCounter == Fatirn)

FatirCounter = 0

Fatirn = Fatirn \* 2

endif

if (FatirCurrent->FatirLeft != NULL)

Fatirj++

FatirQ[Fatirj] = FatirCurrent->FatirLeft

endif

if (FatirCurrent->FatirRight != NULL)

Fatirj++

FatirQ[Fatirj] = FatirCurrent->FatirRight

endif

Fatiri++

**Kamus/Deklarasi Variabel**

FatirX=char

**Algoritma/Deskripsi**

struct FatirNode

struct FatirNode \*FatirLeft

char FatirINFO

struct FatirNode \*FatirRight

FatirInisialisasi()

input(FatirX)

FatirBuatSimpul(FatirX)

FatirBuatSimpulAkar()

FatirInsertUrutNomor()

FatirBacaUrutNomor()

**Algoritma/Bahasa Natural (Nomor 07)**

1. Membuat function FatirInisialisasi()
2. FatirRoot = NULL
3. FatirP = NULL
4. Membuat function FatirBuatSimpul(char FatirX)
5. FatirP = (FatirSimpul\*)malloc(sizeof(FatirSimpul))
6. Jika (FatirP != NULL), maka kerjakan baris 7 s.d 9, kalau tidak kerjakan baris 10 s.d 11
7. FatirP->FatirINFO = FatirX
8. FatirP->FatirLeft = NULL
9. FatirP->FatirRight = NULL
10. Menampilkan (“Memory Heap Full”)
11. exit(1)
12. Membuat function FatirBuatSimpulAkar()
13. Jika (FatirRoot == NULL), maka kerjakan baris 14 s.d 18, kalau tidak kerjakan baris 19
14. Jika (FatirP != NULL), maka kerjakan baris 15 s.d 17, kalau tidak kerjakan baris 18
15. FatirRoot = FatirP
16. FatirRoot->FatirLeft = NULL
17. FatirRoot->FatirRight = NULL
18. Menampilkan (“FatirSimpul Belum Dibuat”)
19. Menampilkan (“Pohon Sudah Ada!!!!”)
20. Membuat function FatirInsertUrutNomor()
21. FatirFlag = 0
22. Fatiri = 1
23. Fatirj = 1
24. FatirQ[Fatiri] = FatirRoot
25. Selama (FatirFlag == 0 && Fatirj < 127), maka kerjakan baris 26 s.d 45, kalau tidak kerjakan baris 46
26. Memasukkan isi/nilai variabel FatirX
27. Jika (FatirX != '0'), maka kerjakan baris 28 s.d 32, kalau tidak kerjakan baris 33 s.d 35
28. FatirBuatSimpul(FatirX)
29. FatirCurrent = FatirQ[Fatiri]
30. FatirCurrent->FatirLeft = FatirP
31. Fatirj++
32. FatirQ[Fatirj] = FatirP
33. FatirFlag = 1
34. Fatirj++
35. FatirQ[Fatirj] = NULL
36. Jika (FatirFlag == 0), maka kerjakan baris 37 s.d 44, kalau tidak kerjakan baris 45
37. Jika (FatirX != '0'), maka kerjakan baris 38 s.d 41, kalau tidak kerjakan baris 42 s.d 44
38. FatirBuatSimpul(FatirX)
39. FatirCurrent->FatirRight = FatirP
40. Fatirj++
41. FatirQ[Fatirj] = FatirP
42. FatirFlag = 1
43. Fatirj++
44. FatirQ[Fatirj] = NULL
45. Fatiri++
46. Membuat function FatirBacaUrutNomor()
47. Fatiri = 1
48. Fatirj = 1
49. Fatirn = 1
50. FatirCounter = 0
51. Selama (FatirQ[Fatiri] != NULL), maka kerjakan baris 52 s.d 64, kalau tidak kerjakan baris 65
52. FatirCurrent = FatirQ[Fatiri]
53. Menampilkan isi/nilai variabel (FatirCurrent->FatirINFO)
54. FatirCounter++
55. Jika (FatirCounter == Fatirn), maka kerjakan baris 56 s.d 57, kalau tidak kerjakan baris 58
56. FatirCounter = 0
57. Fatirn = Fatirn \* 2
58. Jika (FatirCurrent->FatirLeft != NULL), maka kerjakan baris 59 s.d 60, kalau tidak kerjakan baris 61
59. Fatirj++
60. FatirQ[Fatirj] = FatirCurrent->FatirLeft
61. Jika (FatirCurrent->FatirRight != NULL), maka kerjakan baris 62 s.d 63, kalau tidak kerjakan baris 64
62. Fatirj++
63. FatirQ[Fatirj] = FatirCurrent->FatirRight
64. Fatiri++
65. Mendeklarasikan struct (struct FatirNode(struct FatirNode \*FatirLeft, char FatirINFO, struct FatirNode \*FatirRight))
66. Mendefinisikan struct (typedef struct FatirNode FatirSimpul, FatirSimpul \*FatirRoot, \*FatirP, \*FatirCurrent, FatirSimpul \*FatirQ[129])
67. Memanggil function FatirInisialisasi()
68. Memasukkan isi/nilai variabel FatirX
69. Memanggil function FatirBuatSimpul(FatirX)
70. Memanggil function FatirBuatSimpulAkar()
71. Memanggil function FatirInsertUrutNomor()
72. Memanggil function FatirBacaUrutNomor()
73. Selesai

**Program (Nomor 07)**

