**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**TEKNIK INFORMATIKA**

**Prak-12**



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

**Pseudocode (Nomor 01)**

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

-

**Algoritma/Deskripsi Function Fatirinithashtab()**

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

Fatirhashtab[Fatiri] = NULL

endfor

**Kamus/Deklarasi Variabel Function FatirmyHash(char\* Fatirs)**

**-**

**Algoritma/Deskripsi Function FatirmyHash(char\* Fatirs)**

unsigned int Fatirh = 0

for (; \*Fatirs; Fatirs++)

Fatirh = \*Fatirs + Fatirh \* 31

endfor

return Fatirh % FatirHASHSIZE

**Kamus/Deklarasi Variabel Function Fatirnode\* Fatirlookup(char\* Fatirn)**

**-**

**Algoritma/Deskripsi Function Fatirnode\* Fatirlookup(char\* Fatirn)**

unsigned int Fatirhi = FatirmyHash(Fatirn)

Fatirnode\* Fatirnp = Fatirhashtab[Fatirhi]

for (; Fatirnp != NULL; Fatirnp = Fatirnp->Fatirnext)

if (!strcmp(Fatirnp->Fatirnama, Fatirn))

return Fatirnp

endif

endfor

return NULL

**Kamus/Deklarasi Variabel Function** **Fatirm\_strdup(const char\* Fatiro)**

**-**

**Algoritma/Deskripsi Function Fatirm\_strdup(const char\* Fatiro)**

int Fatirl = strlen(Fatiro) + 1

char\* Fatirns = (char\*)malloc(Fatirl \* sizeof(char))

strcpy(Fatirns, Fatiro)

if (Fatirns == NULL)

return NULL

else

return Fatirns

endif

**Kamus/Deklarasi Variabel Function get(const char\* Fatirnama)**

**-**

**Algoritma/Deskripsi Function get(const char\* Fatirnama)**

Fatirnode\* Fatirn = Fatirlookup(const\_cast<char\*>(Fatirnama))

if (Fatirn == NULL)

return NULL

else

return Fatirn->Fatirdesc

endif

**Kamus/Deklarasi Variabel Function Fatirinstall(char\* Fatirnama, char\* Fatirdesc)**

-

**Algoritma/Deskripsi Function Fatirinstall(char\* Fatirnama, char\* Fatirdesc)**

unsigned int Fatirhi = FatirmyHash(Fatirnama)

Fatirnode\* Fatirnp

if ((Fatirnp = Fatirlookup(Fatirnama)) == NULL)

Fatirnp = (Fatirnode\*)malloc(sizeof(Fatirnode))

if (Fatirnp == NULL)

return 0

endif

Fatirnp->Fatirnama = Fatirm\_strdup(Fatirnama)

if (Fatirnp->Fatirnama == NULL)

return 0

endif

Fatirnp->Fatirnext = Fatirhashtab[Fatirhi]

Fatirhashtab[Fatirhi] = Fatirnp

else

free(Fatirnp->Fatirdesc)

endif

Fatirnp->Fatirdesc = Fatirm\_strdup(Fatirdesc)

if (Fatirnp->Fatirdesc == NULL)

return 0

endif

return 1

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

**-**

**Algoritma/Deskripsi Function Fatirdisplaytable()**

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

Fatirnode\* Fatirt = Fatirhashtab[Fatiri]

while (Fatirt != NULL)

print(Fatirt->Fatirnama)

print(Fatirt->Fatirdesc)

Fatirt = Fatirt->Fatirnext

endwhile

endfor

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

**-**

**Algoritma/Deskripsi Function Fatircleanup()**

Fatirnode\* Fatirnp, \* Fatirt

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

if (Fatirhashtab[Fatiri] != NULL)

Fatirnp = Fatirhashtab[Fatiri]

while (Fatirnp != NULL)

Fatirt = Fatirnp->Fatirnext

free(Fatirnp->Fatirnama)

free(Fatirnp->Fatirdesc)

free(Fatirnp)

Fatirnp = Fatirt

endwhile

endif

endfor

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

Fatiri = int

**Algoritma/Deskripsi Function** **Fatirdata()**

const char\* Fatirnamas[] = {"Fatirnama ", "alamat ", "telpon ", "cita-cita ", "sekolah "}

const char\* Fatirdescs[] = {"Ichan", "Depok", "089999991", "Dosen", "Mahasiswa"}

Fatirinithashtab()

for (Fatiri = 0; Fatiri < 5; Fatiri++)

Fatirinstall(const\_cast<char\*>(Fatirnamas[Fatiri]), const\_cast<char\*>(Fatirdescs[Fatiri]))

endfor

print(("nama "), ("alamat "), ("telpon "), ("cita-cita "), ("sekolah "))

**Kamus/Deklarasi Variabel**

Fatirpilihan = int

**Algoritma/Deskripsi**

typedef struct Fatir\_node

char\* Fatirnama

char\* Fatirdesc

struct Fatir\_node\* Fatirnext

Fatirnode

static Fatirnode\* Fatirhashtab[FatirHASHSIZE]

do

input(Fatirpilihan)

switch (Fatirpilihan)

case 1:

Fatirdata()

case 2:

Fatirdisplaytable();

case 3:

Fatircleanup();

case 4:

print("Terima Kasih! ")

while (Fatirpilihan != 4)

endwhile

**Algoritma/Bahasa Natural (Nomor 01)**

1. Membuat function Fatirinithashtab()
2. int Fatiri = 0
3. Selama (Fatiri < FatirHASHSIZE), maka kerjakan baris 4 s.d 5, kalau tidak kerjakan baris 6
4. Fatirhashtab[Fatiri] = NULL
5. Fatiri++
6. Membuat function FatirmyHash(char\* Fatirs)
7. unsigned int Fatirh = 0
8. Selama (\*Fatirs), maka kerjakan baris 9 s.d 10, kalau tidak kerjakan baris 11
9. Fatirh = \*Fatirs + Fatirh \* 31
10. Fatirs++
11. return Fatirh % FatirHASHSIZE
12. Membuat function Fatirnode\* Fatirlookup(char\* Fatirn)
13. unsigned int Fatirhi = FatirmyHash(Fatirn)
14. Fatirnode\* Fatirnp = Fatirhashtab[Fatirhi]
15. Selama (Fatirnp != NULL), maka kerjakan baris 16 s.d 18, kalau tidak kerjakan baris 19
16. Jika (!strcmp(Fatirnp->Fatirnama, Fatirn)), maka kerjakan baris 17, kalau tidak kerjakan baris 18
17. return Fatirnp
18. Fatirnp = Fatirnp->Fatirnext
19. return NULL
20. Membuat function Fatirm\_strdup(const char\* Fatiro)
21. int Fatirl = strlen(Fatiro) + 1
22. char\* Fatirns = (char\*)malloc(Fatirl \* sizeof(char))
23. strcpy(Fatirns, Fatiro)
24. Jika (Fatirns == NULL), maka kerjakan baris 25, kalau tidak kerjakan baris 26
25. return NULL
26. return Fatirns
27. Membuat function get(const char\* Fatirnama)
28. Fatirnode\* Fatirn = Fatirlookup(const\_cast<char\*>(Fatirnama))
29. Jika (Fatirn == NULL), maka kerjakan baris 30, kalau tidak kerjakan baris 31
30. return NULL
31. return Fatirn->Fatirdesc
32. Membuat function Fatirinstall(char\* Fatirnama, char\* Fatirdesc)
33. unsigned int Fatirhi = FatirmyHash(Fatirnama)
34. Mendefinisikan struct (Fatirnode\* Fatirnp)
35. Jika ((Fatirnp = Fatirlookup(Fatirnama)) == NULL), maka kerjakan baris 36 s.d 44, kalau tidak kerjakan baris 45
36. Fatirnp = (Fatirnode\*)malloc(sizeof(Fatirnode))
37. Jika (Fatirnp == NULL), maka kerjakan baris 38, kalau tidak kerjakan baris 39

1. return 0
2. Fatirnp->Fatirnama = Fatirm\_strdup(Fatirnama)
3. Jika (Fatirnp->Fatirnama == NULL), maka kerjakan baris 40, kalau tidak kerjakan baris 41
4. return 0
5. Fatirnp->Fatirnext = Fatirhashtab[Fatirhi]
6. Fatirhashtab[Fatirhi] = Fatirnp
7. free(Fatirnp->Fatirdesc)
8. Fatirnp->Fatirdesc = Fatirm\_strdup(Fatirdesc)
9. Jika (Fatirnp->Fatirdesc == NULL), maka kerjakan baris 47, kalau tidak kerjakan baris 48
10. return 0
11. return 1
12. Membuat function Fatirdisplaytable()
13. int Fatiri = 0
14. Selama (Fatiri < FatirHASHSIZE), maka kerjakan baris 52 s.d 57, kalau tida kerjakan baris 58
15. Fatirnode\* Fatirt = Fatirhashtab[Fatiri]
16. Selama (Fatirt != NULL), maka kerjakan baris 54 s.d 56, kalau tidak kerjakan baris 57
17. Menampilkan isi/nilai variabel (Fatirt->Fatirnama)
18. Menampilkan isi/nilai variabel (Fatirt->Fatirdesc)
19. Fatirt = Fatirt->Fatirnext
20. Fatiri++
21. Membuat function Fatircleanup()
22. Mendefinisikan static (Fatirnode\* Fatirnp, \* Fatirt)
23. int Fatiri = 0
24. Selama (Fatiri < FatirHASHSIZE), maka kerjakan baris 62 s.d 70, kalau tidak kerjakan baris 71
25. Jika (Fatirhashtab[Fatiri] != NULL), maka kerjakan baris 63 s.d 69, kalau tidak kerjakan baris 70
26. Fatirnp = Fatirhashtab[Fatiri]
27. Selama (Fatirnp != NULL), maka kerjakan baris 65 s.d 69, kalau tidak kerjakan baris 70
28. Fatirt = Fatirnp->Fatirnext
29. free(Fatirnp->Fatirnama)
30. free(Fatirnp->Fatirdesc
31. free(Fatirnp)
32. Fatirnp = Fatirt
33. Fatiri++
34. Membuat function Fatirdata()
35. const char\* Fatirnamas[] = {"nama ", "alamat ", "telpon ", "cita-cita ", "sekolah "}
36. const char\* Fatirdescs[] = {"Ichan", "Depok", "089999991", "Dosen", "Mahasiswa"}
37. Memanggil function Fatirinithashtab()
38. Fatiri = 0
39. Selama (Fatiri < 5), maka kerjakan baris 77 s.d 78, kalau tidak kerjakan baris 79
40. Fatirinstall(const\_cast<char\*>(Fatirnamas[Fatiri]), const\_cast<char\*>(Fatirdescs[Fatiri]))
41. Fatiri++
42. Menampilkan isi/nilai (("nama "),("alamat "),("telpon "),("cita-cita "), ("sekolah "))
43. Mendeklarasikan struct (typedef struct Fatir\_node(char\* Fatirnama, char\* Fatirdesc, struct Fatir\_node\* Fatirnext))
44. Mendefinisikan struct (Fatirnode)
45. static Fatirnode\* Fatirhashtab[FatirHASHSIZE]
46. Memasukkan isi/nilai variabel Fatirpilihan
47. Jika (Fatirpilihan=1), maka kerjakan baris 85, kalau tidak kerjakan baris 86
48. Memanggil function Fatirdata()
49. Jika (Fatirpilihan=2), maka kerjakan baris 87, kalau tidak kerjakan baris 88
50. Memanggil function Fatirdisplaytable()
51. Jika (Fatirpilihan=3), maka kerjakan baris 89, kalau tidak kerjakan baris 90
52. Memanggil function Fatircleanup()
53. Jika (Fatirpilihan=4), maka kerjakan baris 91, kalau tidak kerjakan baris 92
54. Menampilkan ("Terima Kasih! ")
55. Selama (Fatirpilihan != 4), maka kerjakan baris 83 s.d 91, kalau tidak kerjakan baris 93
56. Selesai

**Program (Nomor 01)**

