

NOTASI FUNGSIONAL
DASAR PEMROGRAMAN



Dosen Pengampu :

Bu Khadijah, S.Kom., M.Cs.

Nama :

- 1. Muchammad Yuda Tri Ananda (24060124110142)**
- 2. Muhammad Nauval Fadli (24060124120027)**
- 3. Novelya Cherina (24060124140174)**
- 4. Olivia Oktaviani (24060124120050)**
- 5. Nayla Husna (24060124140158)**

Universitas Diponegoro Informatika 2024 Kelas D

NOTASI FUNGSIONAL KOLEKSI OBJEK MAHASISWA

TYPE MAHASISWA (MHS)

DEFINISI DAN SPESIFIKASI TYPE

type Mhs: <nim: string, nama: string, kelas: character, nilai: list of integer>

{type Mhs terdiri atas nim, nama, dan kelas mahasiswa, serta kumpulan nilai kuis yang pernah dikerjakan, dengan maksimal jumlah mengerjakan adalah 10 kali. Nilai mahasiswa memiliki rentang antara 0-100}

type SetMhs: [] atau [e o List]

{type SetMhs adalah List dari type Mhs yang di dalamnya tidak ada Mhs yang mempunyai NIM yang sama}

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

SPESIFIKASI

MakeMhs: <string, string, character, list of integer> → Mhs

{MakeMhs(nim,nama, kelas, nilai) membentuk sebuah mahasiswa dengan dengan nim, nama, kelas dan nilai berbentuk list of integer.

Contoh:

MakeMhs('234', 'Andi', 'C', []) membentuk mahasiswa dengan nim '234', nama 'Andi' dari kelas C, dan belum pernah mengerjakan kuis (nilainya berupa list kosong).

MakeMhs('123', 'Caca', 'C', [90,80,100]) membentuk mahasiswa dengan nim '123', nama 'CC' dari kelas C, dan telah mengerjakan kuis sebanyak tiga kali dengan nilai masing-masing adalah 90, 80, dan 100. }

MakeSetMhs: Mhs → SetMhs

{MakeSetMhs(mhs) membentuk SetMhs dengan NIM yang harus unik (tidak boleh sama)}

DEFINISI DAN SPESIFIKASI SELEKTOR

SelectNimMhs(Mhs): Mhs \rightarrow string

{SelectNimMhs(mhs) Mengambil NIM mahasiswa dari elemen pertama dalam daftar mahasiswa.}

SelectNamaMhs(Mhs): Mhs \rightarrow string

{SelectNamaMhs(Mhs) Mengambil Nama mahasiswa dari elemen pertama dalam daftar mahasiswa.}

SelectKelasMhs(Mhs): Mhs \rightarrow string

{SelectKelasMhs(Mhs) Mengambil kelas mahasiswa dari elemen pertama dalam daftar mahasiswa.}

SelectNilaiMhs(Mhs): Mhs \rightarrow list

{SelectNilaiMhs(Mhs) Mengambil nilai mahasiswa dari elemen pertama dalam daftar mahasiswa.}

DEFINISI DAN SPESIFIKASI OPERATOR

HitungRataRataNilaiMhs(nilaiMhs): list \rightarrow integer

{HitungRataRataNilaiMhs(nilaiMhs) Menghitung rata-rata nilai mahasiswa jika tidak kosong}

MakeSetMhs(Mhs): Set of Mhs \rightarrow list

{MakeSetMhs(Mhs) Menghasilkan himpunan mahasiswa yang memiliki NIM unik.}

IsNimMemberSetMhs(nim, SetMhs): string, list → boolean

{IsNimMemberSetMhs(nim, SetMhs) Memeriksa apakah NIM ada dalam himpunan mahasiswa.}

SetMhsLulus(SetMhs): list → list

{SetMhsLulus(SetMhs) Menghasilkan himpunan mahasiswa yang lulus (rata-rata nilai ≥ 70).}

BanyakSetMhsLulus(SetMhs): list → integer

{BanyakSetMhsLulus(SetMhs) Menghitung jumlah mahasiswa yang lulus (rata-rata nilai ≥ 70).}

SetMhsTidakMengerjakanKuisKelas(kelas, SetMhs): string, list → list

{SetMhsTidakMengerjakanKuisKelas(kelas, SetMhs) Menghasilkan himpunan mahasiswa yang tidak mengerjakan kuis di kelas tertentu.}

BanyakSetMhsTidakMengerjakanKuis(SetMhs): list → integer

{BanyakSetMhsTidakMengerjakanKuis(SetMhs) Menghitung jumlah mahasiswa yang tidak mengerjakan kuis.}

NilaiTertinggi(SetMhs): list → integer

{NilaiTertinggi(SetMhs) Menghasilkan nilai tertinggi di antara semua mahasiswa.}

NilaiTertinggiKelas(kelas, SetMhs): string, list → integer

{NilaiTertinggiKelas(kelas, SetMhs) Menghasilkan nilai tertinggi di kelas tertentu.}

MhsNilaiTertinggiKelas(kelas, SetMhs, MaxNilai): string, list, integer → list

{MhsNilaiTertinggiKelas(kelas, SetMhs, MaxNilai) Menghasilkan mahasiswa dengan nilai tertinggi di kelas tertentu.}

NilaiMahasiswaTertinggiKelas(Kelas, SetMhs): string, list → list

{NilaiMahasiswaTertinggiKelas(Kelas, SetMhs) Menghasilkan mahasiswa dengan nilai tertinggi di kelas tertentu dengan menggunakan nilai tertinggi.}

REALISASI

MakeMhs(nim, nama, kelas, nilai): then

[nim, nama, kelas, nilai]

SelectNimMhs(Mhs): then

Mhs[0]

SelectNamaMhs(Mhs): then

Mhs[1]

SelectKelasMhs(Mhs): then

Mhs[2]

SelectNilaiMhs(Mhs): then

Mhs[3]

HitungRataRataNilaiMhs(nilaiMhs):

if IsEmpty(nilaiMhs): then

0 else

SumElmt(nilaiMhs) / NbElmt(nilaiMhs)

MakeSetMhs(Mhs):

If IsEmpty(Mhs): then

[]

If IsNimMemberSetMhs(SelectNimMhs(FirstElmt(Mhs)), Tail(Mhs)): then

MakeSetMhs(Tail(Mhs)) else

Konso(FirstElmt(Mhs), MakeSetMhs(Tail(Mhs)))

IsNimMemberSetMhs(nim, SetMhs):

If IsEmpty(SetMhs): then

False

if nim = SelectNimMhs(FirstElmt(SetMhs)): then

True else

IsNimMemberSetMhs(nim, Tail(SetMhs))

SetMhsLulus(SetMhs):

if IsEmpty(SetMhs): then

[]

if HitungRataRataNilaiMhs(SelectNilaiMhs(FirstElmt(SetMhs))) >= 70: then

Konso(FirstElmt(SetMhs), SetMhsLulus(Tail(SetMhs))) else

SetMhsLulus(Tail(SetMhs))

BanyakSetMhsLulus(SetMhs):

if IsEmpty(SetMhs): then

0

if HitungRataRataNilaiMhs(SelectNilaiMhs(FirstElmt(SetMhs))) >= 70: then

1 + BanyakSetMhsLulus(Tail(SetMhs)) else

BanyakSetMhsLulus(Tail(SetMhs))

SetMhsTidakMengerjakanKuisKelas(kelas, SetMhs):

if IsEmpty(SetMhs): then

[]

if kelas = SelectKelasMhs(FirstElmt(SetMhs)) and IsEmpty(

SelectNilaiMhs(FirstElmt(SetMhs))

): then

Konso(

FirstElmt(SetMhs), SetMhsTidakMengerjakanKuisKelas(kelas, Tail(SetMhs))

) else

SetMhsTidakMengerjakanKuisKelas(kelas, Tail(SetMhs))

BanyakSetMhsTidakMengerjakanKuis(SetMhs):

if IsEmpty(SetMhs): then

0

if IsEmpty(SelectNilaiMhs(FirstElmt(SetMhs))): then

1+ BanyakSetMhsTidakMengerjakanKuis(Tail(SetMhs)) else

BanyakSetMhsTidakMengerjakanKuis(Tail(SetMhs))

NilaiTertinggi(SetMhs):

```
if IsEmpty(SelectNilaiMhs(FirstElmt(SetMhs))): then  
    NilaiTertinggi(Tail(SetMhs))  
  
if IsOneElmt(SetMhs): then  
    HitungRataRataNilaiMhs(SelectNilaiMhs(FirstElmt(SetMhs))) else  
  
max2(  
    HitungRataRataNilaiMhs(SelectNilaiMhs(FirstElmt(SetMhs))),  
    NilaiTertinggi(Tail(SetMhs)),  
)
```

NilaiTertinggiKelas(kelas, SetMhs):

```
if IsEmpty(SetMhs): then  
    0  
  
if IsEmpty(SelectNilaiMhs(FirstElmt(SetMhs))) or not kelas = SelectKelasMhs(  
    FirstElmt(SetMhs)  
): then  
    NilaiTertinggiKelas(kelas, Tail(SetMhs)) else  
  
max2(  
    HitungRataRataNilaiMhs(SelectNilaiMhs(FirstElmt(SetMhs))),  
    NilaiTertinggiKelas(kelas, Tail(SetMhs)),  
)
```

MhsNilaiTertinggiKelas(kelas, SetMhs, MaxNilai):

```
if IsEmpty(SetMhs): then
```


[]

if IsEmpty(SelectNilaiMhs(FirstElmt(SetMhs))): then

MhsNilaiTertinggiKelas(kelas, Tail(SetMhs), MaxNilai)

if kelas = SelectKelasMhs(

FirstElmt(SetMhs)

) and MaxNilai = HitungRataRataNilaiMhs(SelectNilaiMhs(FirstElmt(SetMhs))): then

Konso(

FirstElmt(SetMhs), MhsNilaiTertinggiKelas(kelas, Tail(SetMhs), MaxNilai)

) else

MhsNilaiTertinggiKelas(kelas, Tail(SetMhs), MaxNilai)

NilaiMahasiswaTertinggiKelas(Kelas, SetMhs): then

MhsNilaiTertinggiKelas(Kelas, SetMhs, NilaiTertinggiKelas(Kelas, SetMhs))

APLIKASI

a. Konstruktor khusus untuk SetMhs dengan syarat saat menambahkan elemen mahasiswa baru harus menggunakan nim yang unik (tidak boleh sama dengan nim yang sudah ada).

(

SetMhs(

[

MakeMhs("001", "Budi", "A", [88, 75, 92]),

MakeMhs("002", "Siti", "B", [60, 58]),

MakeMhs("003", "Agus", "A", []),

```

MakeMhs("004", "Rina", "C", [85, 90, 78]),
MakeMhs("005", "Dewi", "B", [72, 65, 80]),
MakeMhs("006", "Toni", "A", [50, 60, 40]),
MakeMhs("007", "Mira", "D", []),
MakeMhs("008", "Rudi", "C", [100, 95, 90]),
MakeMhs("009", "Linda", "D", [55, 65]),
MakeMhs("010", "Rahmat", "A", [95, 98, 99]),
MakeMhs("011", "Fahmi", "B", []),
MakeMhs("123", "Kemal", "A", [90, 69, 90]),
MakeMhs("2001", "Zaki", "E", [85, 80, 92]),
MakeMhs("3001", "Faris", "F", [45, 55, 60]),
MakeMhs("5001", "Alvin", "G", [92, 94, 96]),
MakeMhs("9999", "Nina", "H", [72, 80, 88]),
MakeMhs("69", "Ambasing", "C", [69, 42, 100]),

```

```

]

```

```

)

```

```

)

```

b. Fungsi yang mengembalikan himpunan mahasiswa yang lulus, yaitu yang memiliki nilai rata-rata lebih dari sama dengan 70.

```

(

```

```

    SetMhsLulus(

```

```

        SetMhs(

```

```

            [

```

```

                MakeMhs("001", "Budi", "A", [88, 75, 92]),

```

```

                MakeMhs("002", "Siti", "B", [60, 58]),

```

```

MakeMhs("003", "Agus", "A", []),

MakeMhs("004", "Rina", "C", [85, 90, 78]),

MakeMhs("005", "Dewi", "B", [72, 65, 80]),

MakeMhs("006", "Toni", "A", [50, 60, 40]),

MakeMhs("007", "Mira", "D", []),

MakeMhs("008", "Rudi", "C", [100, 95, 90]),

MakeMhs("009", "Linda", "D", [55, 65]),

MakeMhs("010", "Rahmat", "A", [95, 98, 99]),

MakeMhs("011", "Fahmi", "B", []),

MakeMhs("123", "Kemal", "A", [90, 69, 90]),

MakeMhs("2001", "Zaki", "E", [85, 80, 92]),

MakeMhs("3001", "Faris", "F", [45, 55, 60]),

MakeMhs("5001", "Alvin", "G", [92, 94, 96]),

MakeMhs("9999", "Nina", "H", [72, 80, 88]),

MakeMhs("69", "Ambasing", "C", [69, 42, 100]),

```

```

]

```

```

)

```

```

)

```

```

)

```

c. Fungsi yang mengembalikan himpunan mahasiswa yang tidak mengerjakan kuis sama sekali di suatu kelas tertentu sesuai dengan nama kelas di-input-kan sebagai parameter.

```

(

```

```

    SetMhsTidakMengerjakanKuisKelas(

```

```

        "A",

```

```

        SetMhs(

```

```
[  
    MakeMhs("001", "Budi", "A", [88, 75, 92]),  
    MakeMhs("002", "Siti", "B", [60, 58]),  
    MakeMhs("003", "Agus", "A", []),  
    MakeMhs("004", "Rina", "C", [85, 90, 78]),  
    MakeMhs("005", "Dewi", "B", [72, 65, 80]),  
    MakeMhs("006", "Toni", "A", [50, 60, 40]),  
    MakeMhs("007", "Mira", "D", []),  
    MakeMhs("008", "Rudi", "C", [100, 95, 90]),  
    MakeMhs("009", "Linda", "D", [55, 65]),  
    MakeMhs("010", "Rahmat", "A", [95, 98, 99]),  
    MakeMhs("011", "Fahmi", "B", []),  
    MakeMhs("123", "Kemal", "A", [90, 69, 90]),  
    MakeMhs("2001", "Zaki", "E", [85, 80, 92]),  
    MakeMhs("3001", "Faris", "F", [45, 55, 60]),  
    MakeMhs("5001", "Alvin", "G", [92, 94, 96]),  
    MakeMhs("9999", "Nina", "H", [72, 80, 88]),  
]
```

),

)

)

d. Fungsi yang mengembalikan nilai tertinggi dari semua kelas.

("NILAI TERTINGGI DARI SEMUA KELAS :")

(

NilaiTertinggi(

SetMhs(

[

MakeMhs("001", "Budi", "A", [88, 75, 92]),

MakeMhs("002", "Siti", "B", [60, 58]),

MakeMhs("003", "Agus", "A", []),

MakeMhs("004", "Rina", "C", [85, 90, 78]),

MakeMhs("005", "Dewi", "B", [72, 65, 80]),

MakeMhs("006", "Toni", "A", [50, 60, 40]),

MakeMhs("007", "Mira", "D", []),

MakeMhs("008", "Rudi", "C", [100, 95, 90]),

MakeMhs("009", "Linda", "D", [55, 65]),

MakeMhs("010", "Rahmat", "A", [95, 98, 99]),

MakeMhs("011", "Fahmi", "B", []),

MakeMhs("123", "Kemal", "A", [90, 69, 90]),

MakeMhs("2001", "Zaki", "E", [85, 80, 92]),

MakeMhs("3001", "Faris", "F", [45, 55, 60]),

MakeMhs("5001", "Alvin", "G", [92, 94, 96]),

MakeMhs("9999", "Nina", "H", [72, 80, 88]),

]

)

)

)

e. Fungsi yang mengembalikan mahasiswa yang mendapatkan nilai tertinggi dari suatu kelas tertentu sesuai dengan nama kelas di-input-kan sebagai parameter.

```
(
    NilaiMahasiswaTertinggiKelas(
        "C",
        SetMhs(
            [
                MakeMhs("001", "Budi", "A", [88, 75, 92]),
                MakeMhs("002", "Siti", "B", [60, 58]),
                MakeMhs("003", "Agus", "A", []),
                MakeMhs("004", "Rina", "C", [85, 90, 78]),
                MakeMhs("005", "Dewi", "B", [72, 65, 80]),
                MakeMhs("006", "Toni", "A", [50, 60, 40]),
                MakeMhs("007", "Mira", "D", []),
                MakeMhs("008", "Rudi", "C", [100, 95, 90]),
                MakeMhs("009", "Linda", "D", [55, 65]),
                MakeMhs("010", "Rahmat", "A", [95, 98, 99]),
                MakeMhs("011", "Fahmi", "B", []),
                MakeMhs("123", "Kemal", "A", [90, 69, 90]),
                MakeMhs("2001", "Zaki", "E", [85, 80, 92]),
                MakeMhs("3001", "Faris", "F", [45, 55, 60]),
                MakeMhs("5001", "Alvin", "G", [92, 94, 96]),
                MakeMhs("9999", "Nina", "H", [72, 80, 88]),
                MakeMhs("69", "Ambasing", "C", [90, 100, 95]),
            ]
        ),
    )
)
```

)

f. Fungsi yang mengembalikan banyaknya mahasiswa yang tidak mengerjakan kuis dari semua kelas.

(

BanyakSetMhsTidakMengerjakanKuis(

SetMhs(

[

MakeMhs("001", "Budi", "A", [88, 75, 92]),

MakeMhs("002", "Siti", "B", [60, 58]),

MakeMhs("003", "Agus", "A", []),

MakeMhs("004", "Rina", "C", [85, 90, 78]),

MakeMhs("005", "Dewi", "B", [72, 65, 80]),

MakeMhs("006", "Toni", "A", [50, 60, 40]),

MakeMhs("007", "Mira", "D", []),

MakeMhs("008", "Rudi", "C", [100, 95, 90]),

MakeMhs("009", "Linda", "D", [55, 65]),

MakeMhs("010", "Rahmat", "A", [95, 98, 99]),

MakeMhs("011", "Fahmi", "B", []),

MakeMhs("123", "Kemal", "A", [90, 69, 90]),

MakeMhs("2001", "Zaki", "E", [85, 80, 92]),

MakeMhs("3001", "Faris", "F", [45, 55, 60]),

MakeMhs("5001", "Alvin", "G", [92, 94, 96]),

MakeMhs("9999", "Nina", "H", [72, 80, 88]),

MakeMhs("69", "Ambasing", "C", [69, 42, 100]),

]

),

)

)

g. Fungsi yang mengembalikan banyaknya mahasiswa yang lulus dari semua kelas.

(

BanyakSetMhsLulus(

SetMhs(

[

MakeMhs("001", "Budi", "A", [88, 75, 92]),

MakeMhs("002", "Siti", "B", [60, 58]),

MakeMhs("003", "Agus", "A", []),

MakeMhs("004", "Rina", "C", [85, 90, 78]),

MakeMhs("005", "Dewi", "B", [72, 65, 80]),

MakeMhs("006", "Toni", "A", [50, 60, 40]),

MakeMhs("007", "Mira", "D", []),

MakeMhs("008", "Rudi", "C", [100, 95, 90]),

MakeMhs("009", "Linda", "D", [55, 65]),

MakeMhs("010", "Rahmat", "A", [95, 98, 99]),

MakeMhs("011", "Fahmi", "B", []),

MakeMhs("123", "Kemal", "A", [90, 69, 90]),

MakeMhs("2001", "Zaki", "E", [85, 80, 92]),

MakeMhs("3001", "Faris", "F", [45, 55, 60]),

MakeMhs("5001", "Alvin", "G", [92, 94, 96]),

MakeMhs("9999", "Nina", "H", [72, 80, 88]),

MakeMhs("69", "Ambasing", "C", [69, 42, 100]),

1

),

)

)