

Anggota Kelompok :

1. Muhammad Fikri - 24060124130069
2. Muhammad Firdaus Argifari - 24060124130107
3. Muhammad Izzat Fauzan Putra Arya - 24060124130096
4. Nawaal Hanif Mumtaz Arriye - 24060124120041
5. Rafa Azlan - 24060124140126

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: <mhs: list of Mhs>

{type SetMhs terdiri atas kumpulan Mhs}

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

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

{MakeMhs(nim, nama, kelas, nilai) membentuk sebuah mahasiswa 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 : <list of mhs> → SetMhs

{MakeSetMhs(mhs) membentuk sebuah set yang berisi kumpulan mahasiswa dengan nim unik (jika nim sudah ada maka tidak ditambahkan) }

DEFINISI DAN SPESIFIKASI SELEKTOR

{Akan menggunakan selektor terhadap list yang sudah pernah didefinisikan sebelumnya yaitu FirstElmt() untuk mengakses elemen pertama dalam list dan Tail() untuk mengembalikan semua elemen list kecuali elemen pertama}

getNIM: Mhs → string

{getNIM(mhs) mengembalikan nim pada type Mhs}

getNama: Mhs → string

{getNama(mhs) mengembalikan nama pada type Mhs}

getKelas: Mhs → character

{getKelas(mhs) mengembalikan kelas pada type Mhs}

getNilai: Mhs \rightarrow list of integer
{getNilai(mhs) mengembalikan list berisi nilai pada type Mhs}

DEFINISI DAN SPESIFIKASI PREDIKAT

IsEmpty : list \rightarrow boolean
{IsEmpty(L) bernilai benar jika list L merupakan list kosong.}

DEFINISI DAN SPESIFIKASI FUNGSI ANTARA

nilaiTertinggiKelas : character, SetMhs \rightarrow integer
{nilaiTertinggiKelas(kelas, SetOfMhs) mencari nilai tertinggi pada kelas tertentu.}

mahasiswaTertinggiKelas : character, SetMhs, integer \rightarrow SetMhs
{mahasiswaTertinggiKelas(kelas, SetOfMhs, nilaitinggi) menghasilkan sebuah set himpunan mahasiswa dengan nilai tertinggi.}

DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP SETMHS

SetMhs: Mhs, SetMhs \rightarrow SetMhs
{SetMhs(mhs, SetOfMhs) menambahkan elemen mahasiswa baru harus menggunakan nim yang unik (tidak boleh sama dengan nim yang sudah ada).}

lulus: SetMhs \rightarrow SetMhs
{lulus(SetOfMhs) mengembalikan himpunan mahasiswa yang lulus, yaitu yang memiliki nilai rata-rata lebih dari sama dengan 70.}

tidakMengerjakanKuisKelas: character, SetOfMhs \rightarrow SetOfMhs
{tidakMengerjakanKuisKelas(kelas, SetOfMhs) mengembalikan himpunan mahasiswa yang tidak mengerjakan kuis sama sekali di suatu kelas tertentu sesuai dengan nama kelas di-input-kan sebagai parameter.}

nilaiTertinggi: SetMhs \rightarrow real
{nilaiTertinggi(SetOfMhs) mengembalikan nilai tertinggi dari semua kelas.}

himpunanMahasiswaTertinggi: character, SetMhs \rightarrow SetMhs
{himpunanMahasiswaTertinggi(kls, SetOfMhs) mengembalikan mahasiswa yang mendapatkan nilai tertinggi dari suatu kelas tertentu sesuai dengan nama kelas di-input-kan sebagai parameter.}

jumlahTidakMengerjakanKuis: SetMhs \rightarrow integer
{jumlahTidakMengerjakanKuis(SetOfMhs) mengembalikan banyaknya mahasiswa yang tidak mengerjakan kuis dari semua kelas.}

jumlahLulus: SetMhs \rightarrow integer
{jumlahLulus(SetOfMhs) mengembalikan banyaknya mahasiswa yang lulus dari semua kelas.}

REALISASI

nilaiTertinggiKelas(kelas, setOfMhs) :
 if IsEmpty(setOfMhs) then

```

0
else
  if not IsEmpty(getNilai(FirstElmt(setOfMhs))) and getKelas(FirstElmt(setOfMhs)) =
kelas
    max2(AvgElmt(getNilai(FirstElmt(setOfMhs))), nilaiTertinggiKelas(kelas,
Tail(setOfMhs)))
  else
    nilaiTertinggiKelas(kelas, Tail(setOfMhs))

mahasiswaTertinggiKelas(kelas, setOfMhs, nilaitinggi):
  if IsEmpty(setOfMhs):
    [ ]
  else
    if getKelas(FirstElmt(setOfMhs)) = kelas and not
IsEmpty(getNilai(FirstElmt(setOfMhs))):
      if AvgElmt(getNilai(FirstElmt(setOfMhs))) = nilaitinggi:
        Konso(FirstElmt(setOfMhs), mahasiswaTertinggiKelas(kelas, Tail(setOfMhs),
nilaitinggi))
      else
        mahasiswaTertinggiKelas(kelas, Tail(setOfMhs), nilaitinggi)
    else
      mahasiswaTertinggiKelas(kelas, Tail(setOfMhs), nilaitinggi)

SetMhs(mhs, SetOfMhs):
  if IsEmpty(SetOfMhs) then
    Konso(mhs, SetOfMhs)
  else
    if getNIM(mhs) = getNim(FirstElmt(SetOfMhs)) then
      SetOfMhs
    else
      Konso(FirstElmt(SetOfMhs), SetMhs(mhs, Tail(SetOfMhs)))

lulus(SetOfMhs):
  if IsEmpty(SetOfMhs) then
    SetOfMhs
  else
    if AvgElmt(getNilai(FirstElmt(SetOfMhs))) ≥ 70 then
      Konso(FirstElmt(SetOfMhs), lulus(Tail(SetOfMhs)))
    else
      lulus(Tail(SetOfMhs))

tidakMengerjakanKuisKelas(kelas, SetOfMhs) :
  if IsEmpty(SetOfMhs) then
    SetOfMhs
  else
    if (getKelas(FirstElmt(SetOfMhs)) = kelas
and IsEmpty(getNilai(FirstElmt(SetOfMhs)))) then

```

```

        Konso(FirstElmt(SetOfMhs),
        tidakMengerjakanKuisKelas(kelas, Tail(SetOfMhs)))
    else
        tidakMengerjakanKuisKelas(kelas, Tail(SetOfMhs))

nilaiTertinggi(setOfMhs):
    if IsEmpty(setOfMhs)
        then 0
    else
        if not IsEmpty(getNilai(FirstElmt(setOfMhs))) then
            max2(AvgElmt(getNilai(FirstElmt(setOfMhs))),
            nilaiTertinggi(Tail(setOfMhs)))
        else
            nilaiTertinggi(Tail(setOfMhs))

himpunanMahasiswaTertinggi(kelas, setOfMhs):
    mahasiswaTertinggiKelas(kelas, setOfMhs, nilaiTertinggiKelas(kelas, setOfMhs))

jumlahTidakMengerjakanKuis(setOfMhs) :
    if IsEmpty(setOfMhs) then
        0
    else
        if IsEmpty(getNilai(FirstElmt(setOfMhs))) then
            1 + FirstElmt(setOfMhs), jumlahTidakMengerjakanKuis(Tail(setOfMhs))
        else
            jumlahTidakMengerjakanKuis(Tail(setOfMhs))

def jumlahLulus(setOfMhs):
    NbElmt(lulus(setOfMhs))

```

APLIKASI

```

→ IsEmpty([[1], ['andi'], ['C'], []]) : → false
→
→ SetMhs(
→ ['4', 'Tom', 'D', [100, 80],
→ [
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]) → [
→   ['4', 'Tom', 'D', [100, 80],
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]

```

```

→
→ lulus([
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]) → [['6', 'Doe', 'A', [80, 80]]]
→
→ tidakMengerjakanKuisKelas('A', [
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ])
→ ) → [['7', 'Tania', 'A', []]]
→
→ nilaiTertinggi([
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]) → [['6', 'Doe', 'A', [80, 80]]]
→
→ himpunanMahasiswaTertinggi('A', [
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]) → [['7', 'Tania', 'A', []]]
→
→ jumlahTidakMengerjakanKuis([
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]) → 1
→
→ jumlahLulus([
→   ['5', 'John', 'A', [70, 20]],
→   ['6', 'Doe', 'A', [80, 80]],
→   ['6', 'Adam', 'D', [80, 45]],
→   ['7', 'Tania', 'A', []],
→ ]) → 1

```