**Anggota Kelompok :**

- Muhammad Kemal Faza (24060124120013)

- Muhammad Zaidaan Ardiyansyah (24060124140200)

- Muhammad Dimas Arya Putra (24060124130062)

- Muhammad Hafidh Al-Ghifari (24060124140133)

- Muhammad Akmal Fazli Riyadi (24060124130123)

**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** Set of Mhs: [] atau [e o List]  *{type Set of Mhs adalah List dari type Mhs yang di dalamnya tidak ada Mhs yang mempunyai NIM yang sama}* |
| **DEFINISI DAN SPESIFIKASI KONSTRUKTOR**  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. }*  SetMhs: Mhs, Set of Mhs → Set of Mhs  *{SetMhs(mhs, L) menambahkan elemen mahasiswa baru ke dalam List L dengan NIM yang harus unik (tidak boleh sama}* |
| **DEFINISI DAN SPESIFIKASI SELEKTOR**  Nim: Mhs → string  *{Nim(mhs) memberikan nim mahasiswa mhs}*  Nama: Mhs → string  *{Nama(mhs) memberikan nama mahasiswa mhs}*  Kelas: Mhs → string  *{Kelas(mhs) memberikan asal kelas mahasiswa mhs}*  Nilai: Mhs → List  *{Nilai(mhs) mengembalikan nilai-nilai dari mahasiswa mhs}* |
| **DEFINISI DAN SPESIFIKASI OPERATOR**  MhsLulus: Set of Mhs → Set of Mhs  *{MhsLulus(S) mengembalikan himpunan mahasiswa yang lulus, yaitu yang memiliki nilai rata-rata lebih dari sama dengan 70}*  MhsEmptyQuiz: character, Set of Mhs → Set of Mhs  *{MhsEmptyQuiz(C, S) mengembalikan himpunan mahasiswa yang tidak mengerjakan kuis sama sekali di suatu kelas tertentu sesuai dengan nama kelas di-input-kan sebagai parameter}*  NilaiTertinggi: Set of Mhs → integer  *{NilaiTertinggi(S) mengembalikan nilai tertinggi dari semua kelas}*  MhsNilaiTertinggi: character, Set of Mhs → Mhs  *{MhsNilaiTertinggi(C, S) mengembalikan mahasiswa yang mendapatkan nilai tertinggi dari suatu kelas tertentu sesuai dengan nama kelas di-input-kan sebagai parameter}*  BanyakMhsEmptyQuiz: Set of Mhs → integer  *{BanyakMhsEmptyQuiz(S) mengembalikan banyaknya mahasiswa yang tidak mengerjakan kuis dari semua kelas}*  BanyakMhsLulus: Set of Mhs → integer  *{BanyakMhsLulus(S) mengembalikan banyaknya mahasiswa yang lulus dari semua kelas}* |
| **REALISASI**  MakeSetMhs(mhs, L):  if IsEmpty(L) then  []  else  if Nim(mhs) = Nim(FirstElmnt(L)) then  MakeSetMhs(mhs, Tail(L))  else  Konsi(mhs, L)  MhsLulus(S):  if IsEmpty(S) then  []  else  if AvgElmnt(Nilai(FirstElmnt(S))) ≥ 70 then  Konso(FirstElmnt(S), MhsLulus(Tail(S)))  else  MhsLulus(Tail(S))  MhsEmptyQuiz(C, S):  if IsEmpty(S) then  []  else  if Kelas(FirstElmnt(S)) = C and Nilai(FirstElmnt(S)) = [] then  Konso(FirstElmnt(S), MhsEmptyQuiz(C, Tail(S)))  else  MhsEmptyQuiz(C, Tail(S))  NilaiTertinggiSemua(S):  if IsOneElmnt(S) then  MaxElmnt(Nilai(FirstElmnt(S)))  else  if IsEmpty(Nilai(FirstElmnt(S))) then  NilaiTertinggiSemua(Tail(S))  else  max2(AvgElmnt(Nilai(FirstElmnt(S))),  NilaiTertinggiSemua(Tail(S)))  NilaiTertinggiPerKelas(C, S):  if IsOneElmnt(S) then  if not IsEmpty(Nilai(FirstElmnt(S))) then  AvgElmnt(Nilai(FirstElmnt(S)))  else  0  else  if IsEmpty(Nilai(FirstElmnt(S))) or Kelas(FirstElmnt(S)) ≠ C then  NilaiTertinggiPerKelas(C, Tail(S))  else  max2(AvgElmnt(Nilai(FirstElmnt(S))),NilaiTertinggiPerKelas(C, Tail(S)))  MhsNilaiTertinggi(C, S, HighScore):  if IsEmpty(S) then  []  else  if IsEmpty(Nilai(FirstElmnt(S))) or C ≠ Kelas(FirstElmnt(S)) then  MhsNilaiTertinggi(C, Tail(S), HighScore)  else  if AvgElmnt(Nilai(FirstElmnt(S))) = HighScore then  Konso(FirstElmnt(S),MhsNilaiTertinggi(C,Tail(S),HighScore))  else  MhsNilaiTertinggi(C, Tail(S), HighScore)  BanyakMhsEmptyQuiz(S):  if IsEmpty(S) then  0  else  if Nilai(FirstElmnt(S)) = [] then  1 + BanyakMhsEmptyQuiz(Tail(S))  else  BanyakMhsEmptyQuiz(Tail(S))  BanyakMhsLulus(S):  if IsEmpty(S) then  0  else  if AvgElmnt(Nilai(FirstElmnt(S))) ≠ 70 then  1 + BanyakMhsLulus(Tail(S))  else  BanyakMhsLulus(Tail(S)) |
| **APLIKASI**   * MakeSetMhs(<"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <<"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>) * MhsLulus(<"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>) * MhsEmptyQuiz("A", <"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>) * NilaiTertinggiSemua(<"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>) * NilaiTertinggiPerKelas("A", <"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>) * MhsNilaiTertinggi("A", <"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>, NilaiTertinggiPerKelas("A", <"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>)) * BanyakMhsEmptyQuiz(<"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>) * BanyakMhsLulus(<"24060123130061", "Ananda Rachmawati Purwanto", "A", <90, 90, 90>>, <"24060123130090", "Bramantyo Kunni Nurrisqi", "D", <>>, <"24060123140152", "Kayis Hilmi Farih", "A",<90, 70, 100>,>, <"24060123140162", "Lalu Gilang Wirapati", "D", <100, 80, 90>>, <"24060124120013", "Muhamad Kemal Faza", "D", <100, 90, 80>>)   **OUTPUT**   * <<'24060123130061', 'Ananda Rachmawati Purwanto', 'A', <90, 90, 90>>, <'24060123140152', 'Kayis Hilmi Farih', 'A', <90, 70, 100>>, <'24060123140162', 'Lalu Gilang Wirapati', 'D', <100, 80, 90>>, <'24060124120013', 'Muhamad Kemal Faza', 'D', <100, 90, 80>>> * <<'24060124110142', 'Muchammad Yuda Tri Ananda', 'A', <>>> * 90.0 * 90.0 * 2 * 4 * <<'24060123130061', 'Ananda Rachmawati Purwanto', 'A', <90, 90, 90>>> |