

```
1  """
2  Program    : Tipe bentukan PECAHAN
3  Deskripsi  : Didefinisikan suatu type bernama Pecahan, yang terdiri dari pembilang dan penyebut.
              Berikut ini adalah teks dalam notasi fungsional untuk type pecahan tersebut. Perhatikanlah
              bahwa realisasi fungsi hanya dilakukan untuk operator aritmatika dan relasional terhadap
              pecahan. Realisasi selektor hanya diberikan secara konseptual, karena nantinya akan diserahkan
              implementasinya ke bahasa pemrograman
4  NIM>Nama   : 24060124110142/Muchammad Yuda Tri Ananda
5  Tanggal   : 29/09/2024
6  """
7
8  """
9  ****
10 DEFINISI DAN SPESIFIKASI
11     type pecahan : <n: integer >=0 ,d: integer >0>
12         {<n:integer >=0, d:integer >0> n adalah pembilang (numerator) dan d adalah penyebut
13         (denominator). Penyebut sebuah pecahan tidak boleh nol }
14     type pecahancampuran : <bil: integer, n: integer >= 0, d: integer > 0 >
15         {<bil, n, d> adalah sebuah pecahan campuran yang terdiri atas bilangan bil,
16
17 DEFINISI DAN SPESIFIKASI SELEKTOR DENGAN FUNGSI
18     PembPecahan : pecahan → integer >=0
19         { Pemb(p) memberikan numerator pembilang n dari pecahan tsb }
20     PenyPecahan : pecahan → integer > 0
21         { Peny(p) memberikan denominator penyebut d dari pecahan tsb }
22     BilanganPecahanCampuran : pecahancampuran → integer >=0
23         { BilanganPecahanCampuran(Pc) memberikan bilangan bil dari pecahan campuran tsb }
24     PembPecahanCampuran : pecahancampuran → integer >=0
25         { PembPecahanCampuran(Pc) memberikan numerator pembilang n dari pecahan campuran tsb }
26     PenyPecahanCampuran : pecahancampuran → integer > 0
27         { PenyPecahanCampuran(Pc) memberikan denominator penyebut d dari pecahan campuran tsb }
28
29 DEFINISI DAN SPESIFIKASI KONSTRUKTOR
30     MakePecahan : integer >=0, integer > 0 → pecahan
31         { MakeP(pemb,peny) membentuk sebuah pecahan dari pembilang pemb dan penyebut peny,
32         dengan pemb dan peny integer}
33     MakePecahanCampuran : integer, integer >= 0, integer > 0 → pecahancampuran
34         { MakePecahanCampuran(bilangan,pemb,peny) membentuk sebuah pecahan campuran dari
35         pembilang bilangan, pembilang pemb dan penyebut peny, dengan pemb, peny dan bilangan integer }
36
37 DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP PECAHAN { Operator aritmatika Pecahan }
38     AddPecahan : 2 pecahan → pecahan
39         { AddPecahan(P1,P2) : Menambahkan dua buah pecahan P1 dan P2 }
40     SubPecahan : 2 pecahan → pecahan
41         { SubPecahan(P1,P2) : Mengurangkan dua buah pecahan P1 dan P2 }
42     MulPecahan : 2 pecahan → pecahan
43         { MulPecahan(P1,P2) : Mengalikan dua buah pecahan P1 dan P2 }
44     DivPecahan : 2 pecahan → pecahan
45         { DivPecahan(P1,P2) : Membagi dua buah pecahan P1 dan P2 }
```

```

44     RealPecahan : pecahan → real
45         {Menuliskan bilangan pecahan dalam notasi desimal }
46
47 DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP PECAHAN CAMPURAN { Operator aritmatika
PecahanCampuran }
48     AddPecahanCampuran : 2 pecahancampuran → pecahancampuran
49         { AddPecahanCampuran(Pc1,Pc2) : Menambahkan dua buah pecahan campuran Pc1 dan Pc2 }
50     SubPecahanCampuran : 2 pecahancampuran → pecahancampuran
51         { SubPecahanCampuran(Pc1,Pc2) : Mengurangkan dua buah pecahan campuran Pc1 dan Pc2 }
52     DivPecahanCampuran : 2 pecahancampuran → pecahancampuran
53         { DivPecahanCampuran(Pc1,Pc2) : Membagi dua buah pecahan campuran Pc1 dan Pc2 }
54     MulPecahanCampuran : 2 pecahancampuran → pecahancampuran
55         {MulPecahanCampuran(Pc1,Pc2) : Mengalikan dua buah pecahan campuran Pc1 dan Pc2 }
56     RealPecahanCampuran : pecahancampuran → real
57         {Menuliskan bilangan pecahan campuran dalam notasi desimal }
58
59 DEFINISI DAN SPESIFIKASI PREDIKAT
60     IsEqPecahan?: 2 pecahan → boolean
61         {IsEqP?(p1,p2) true jika p1 = p2
62         Membandingkan apakah dua buah pecahan saman ilainya:  $n1/d1 = n2/d2$  jika dan
63         hanya jika  $n1d2=n2d1$  }
64     IsLtPecahan?: 2 pecahan → boolean
65         {IsLtP?(p1,p2) true jika p1 < p2
66         Membandingkan dua buah pecahan, apakah p1 lebih kecil nilainya dari p2:  $n1/d1 <$ 
67          $n2/d2$  jika dan hanya jika  $n1d2 < n2d1$  }
68     IsGtPecahan?: 2 pecahan → boolean
69         {IsGtP?(p1,p2) tue jika p1 > p2
70         Membandingkan nilai dua buah pecahan,, apakah p1 lebih besar nilainya dari p2:
71          $n1/d1 > n2/d2$  jika dan hanya jika  $n1d2 > n2d1$  }
72     IsEqPecahanCampuran?: 2 pecahancampuran → boolean
73         {IsEqP?(p1,p2) true jika p1 = p2
74         Membandingkan apakah dua buah pecahan campuran saman ilainya:  $n1/d1 = n2/d2$  jika dan
75         hanya jika  $n1d2=n2d1$  }
76     IsLtPecahanCampuran?: 2 pecahancampuran → boolean
77         {IsLtP?(p1,p2) true jika p1 < p2
78         Membandingkan dua buah pecahan campuran, apakah p1 lebih kecil nilainya dari p2:  $n1/d1$ 
79         <
80          $n2/d2$  jika dan hanya jika  $n1d2 < n2d1$  }
81     IsGtPecahanCampuran?: 2 pecahancampuran → boolean
82         {IsGtP?(p1,p2) tue jika p1 > p2
83         Membandingkan nilai dua buah pecahan campuran,, apakah p1 lebih besar nilainya dari p2:
84          $n1/d1 > n2/d2$  jika dan hanya jika  $n1d2 > n2d1$  }
85
86 DEFINISI DAN SPESIFIKASI FUNGSI ANTARA
87     PecahanCampuranKePecahan : pecahancampuran → pecahan
88         { PecahanCampuranKePecahan(Pc) mengonversi pecahan campuran menjadi pecahan biasa }
89     PecahanKePecahanCampuran : pecahan → pecahancampuran
90         { PecahanKePecahanCampuran(P) mengonversi pecahan biasa menjadi pecahan campuran }
91     *****
92     """"

```

```

92
93 """
94 *****
95 REALISASI
96 *****
97 """
98
99
100 # Konstruktor Pecahan
101 def MakePecahan(Pemb, Peny):
102     return [Pemb, Peny]
103
104
105 # Selektor Pecahan
106 def PembPecahan(P):
107     return P[0]
108
109
110 def PenyPecahan(P):
111     return P[1]
112
113
114 # Konstruktor PecahanCampuran
115 def MakePecahanCampuran(bilangan, pemb, peny):
116     return [bilangan, pemb, peny]
117
118
119 # Selektor PecahanCampuran
120 def BilanganPecahanCampuran(Pc):
121     return Pc[0]
122
123
124 def PembPecahanCampuran(Pc):
125     return Pc[1]
126
127
128 def PenyPecahanCampuran(Pc):
129     return Pc[2]
130
131
132 # Operator Pecahan
133 def AddPecahan(P1, P2):
134     return MakePecahan(
135         PembPecahan(P1) * PenyPecahan(P2) + PembPecahan(P2) * PenyPecahan(P1),
136         PenyPecahan(P1) * PenyPecahan(P2),
137     )
138
139
140 def SubPecahan(P1, P2):
141     return MakePecahan(

```

```

142         PembPecahan(P1) * PenyPecahan(P2) - PembPecahan(P2) * PenyPecahan(P1),
143         PenyPecahan(P1) * PenyPecahan(P2),
144     )
145
146
147 def MulPecahan(P1, P2):
148     return MakePecahan(
149         PembPecahan(P1) * PembPecahan(P2), PenyPecahan(P1) * PenyPecahan(P2)
150     )
151
152
153 def DivPecahan(P1, P2):
154     return MakePecahan(
155         PembPecahan(P1) * PenyPecahan(P2), PenyPecahan(P1) * PembPecahan(P2)
156     )
157
158
159 def RealPecahan(P):
160     return PembPecahan(P) / PenyPecahan(P)
161
162
163 # Predikat
164 def IsEqPecahan(P1, P2):
165     return RealPecahan(SubPecahan(P1, P2)) == 0
166
167
168 def IsLtPecahan(P1, P2):
169     return RealPecahan(SubPecahan(P1, P2)) < 0
170
171
172 def IsGtPecahan(P1, P2):
173     return RealPecahan(SubPecahan(P1, P2)) > 0
174
175
176 # Fungsi Antara
177 def PecahanCampuranKePecahan(Pc):
178     return (
179         MakePecahan(
180             PenyPecahanCampuran(Pc) * BilanganPecahanCampuran(Pc)
181             + PembPecahanCampuran(Pc),
182             PenyPecahanCampuran(Pc),
183         )
184         if BilanganPecahanCampuran(Pc) >= 0
185         else MakePecahan(
186             PenyPecahanCampuran(Pc) * BilanganPecahanCampuran - PembPecahanCampuran(Pc),
187             PenyPecahanCampuran(Pc),
188         )
189     )
190
191

```

```

192 def PecahanKePecahanCampuran(P):
193     return (
194         MakePecahanCampuran(
195             PembPecahan(P) // PenyPecahan(P),
196             PembPecahan(P),
197             PenyPecahan(P),
198         )
199     if PembPecahan(P) // PenyPecahan(P) == 0
200     else MakePecahanCampuran(
201         PembPecahan(P) // PenyPecahan(P),
202         PembPecahan(P) % PenyPecahan(P),
203         PenyPecahan(P),
204     )
205 )
206
207 # Operator PecahanCampuran
208 def AddPecahanCampuran(Pc1, Pc2):
209     return PecahanKePecahanCampuran(
210         AddPecahan(
211             PecahanCampuranKePecahan(Pc1),
212             PecahanCampuranKePecahan(Pc2),
213         )
214     )
215
216 def SubPecahanCampuran(Pc1, Pc2):
217     return PecahanKePecahanCampuran(
218         SubPecahan(
219             PecahanCampuranKePecahan(Pc1),
220             PecahanCampuranKePecahan(Pc2),
221         )
222     )
223
224 def DivPecahanCampuran(Pc1, Pc2):
225     return PecahanKePecahanCampuran(
226         DivPecahan(
227             PecahanCampuranKePecahan(Pc1),
228             PecahanCampuranKePecahan(Pc2),
229         )
230     )
231
232 def MulPecahanCampuran(Pc1, Pc2):
233     return PecahanKePecahanCampuran(
234         MulPecahan(
235             PecahanCampuranKePecahan(Pc1),
236             PecahanCampuranKePecahan(Pc2),
237         )
238     )
239
240
241

```

```

242     )
243
244
245 def RealPecahanCampuran(Pc):
246     return RealPecahan(PecahanCampuranKePecahan(Pc))
247
248
249 # Predikat
250 def IsEqPecahanCampuran(Pc1, Pc2):
251     return RealPecahanCampuran(SubPecahanCampuran(Pc1, Pc2)) == 0
252
253
254 def IsLtPecahanCampuran(Pc1, Pc2):
255     return RealPecahanCampuran(SubPecahanCampuran(Pc1, Pc2)) < 0
256
257
258 def IsGtPecahanCampuran(Pc1, Pc2):
259     return RealPecahanCampuran(SubPecahanCampuran(Pc1, Pc2)) > 0
260
261
262 """
263 *****
264 APLIKASI
265 *****
266 """
267
268 # Pecahan
269 print(
270     f"AddPecahan: <3,1>, <2,1> -> {AddPecahan(MakePecahan(3, 1), MakePecahan(2, 1))}",
271 )
272 print(
273     f"SubPecahan: <3,1>, <2,1> -> {SubPecahan(MakePecahan(3, 1), MakePecahan(2, 1))}",
274 )
275 print(
276     f"DivPecahan: <3,1>, <2,1> -> {DivPecahan(MakePecahan(3, 1), MakePecahan(2, 1))}",
277 )
278 print(
279     f"MulPecahan: <3,1>, <2,1> -> {MulPecahan(MakePecahan(3, 1), MakePecahan(2, 1))}",
280 )
281 print(
282     f"IsGtPecahan: <3,1>, <2,1> -> {IsGtPecahan(MakePecahan(3, 1), MakePecahan(2, 1))}",
283 )
284 print(
285     f"IsLtPecahan: <3,1>, <2,1> -> {IsLtPecahan(MakePecahan(3, 1), MakePecahan(2, 1))}",
286 )
287 print(
288     f"IsEqPecahan: <3,1>, <3,1> -> {IsEqPecahan(MakePecahan(3, 1), MakePecahan(3, 1))}",
289 )
290
291 # PecahanCampuran

```

```
292 print(  
293     f"AddPecahanCampuran: <1,4,5>, <1,2,5> -> {AddPecahanCampuran(MakePecahanCampuran(1, 4, 5),  
MakePecahanCampuran(1, 2, 5))}",  
294 )  
295 print(  
296     f"SubPecahanCampuran: <1,4,5>, <1,2,5> -> {SubPecahanCampuran(MakePecahanCampuran(1, 4, 5),  
MakePecahanCampuran(1, 2, 5))}",  
297 )  
298 print(  
299     f"DivPecahanCampuran: <1,4,5>, <1,2,5> -> {DivPecahanCampuran(MakePecahanCampuran(1, 4, 5),  
MakePecahanCampuran(1, 2, 5))}",  
300 )  
301 print(  
302     f"MulPecahanCampuran: <1,4,5>, <1,2,5> -> {MulPecahanCampuran(MakePecahanCampuran(1, 4, 5),  
MakePecahanCampuran(1, 2, 5))}",  
303 )  
304 print(  
305     f"IsGtPecahanCampuran: <1,4,5>, <1,2,5> -> {IsGtPecahanCampuran(MakePecahanCampuran(1, 4,  
5), MakePecahanCampuran(1, 2, 5))}",  
306 )  
307 print(  
308     f"IsLtPecahanCampuran: <1,4,5>, <1,2,5> -> {IsLtPecahanCampuran(MakePecahanCampuran(1, 4,  
5), MakePecahanCampuran(1, 2, 5))}",  
309 )  
310 print(  
311     f"IsEqPecahanCampuran: <1,4,5>, <1,2,5> -> {IsEqPecahanCampuran(MakePecahanCampuran(1, 4,  
5), MakePecahanCampuran(1, 2, 5))}",  
312 )  
313
```