

```
1  """
2  Program    : Tipe Pecahan Campuran
3  Deskripsi  : Menentukan tipe pecahan campuran
4  NIM>Nama   : 24060124110142/Muchammad Yuda Tri Ananda
5  Tanggal    : 29/09/2024
6
7  ****
8  DEFINISI DAN SPESIFIKASI TYPE
9  type pecahanc : < bil: integer, n: integer >= 0, d: integer > 0 >
10     <bil, n, d> adalah sebuah pecahan campuran yang terdiri atas bil yang merupakan bilangan
    bulat positif, nol, atau negatif, n yaitu pembilang atau nominator yang merupakan bilangan non-
    negatif, dan d yaitu denominator yang merupakan bilangan positif, lalu nilai n pasti lebih
    kecil daripada nilai d
11
12  type pecahan: < pemb: integer, peny: integer > 0 >
13     <pemb, peny> adalah sebuah pecahan yang terdiri atas pembilang dan penyebut
14
15  ****
16  DEFINISI DAN SPESIFIKASI SELEKTOR
17
18  Bilangan : pecahanc ---> integer
19     Bilangan(P) mengembalikan nilai bil dari tipe <bil, n, d>
20  Nominator : pecahanc ---> integer >= 0
21     Nominator(P) mengembalikan nilai n yaitu nominator atau pembilang dari tipe <bil, n, d>
22  Denominator : pecahanc ---> integer > 0
23     Denominator(P) mengembalikan nilai d yaitu denominator atau penyebut dari tipe <bil, n, d>
24  Pembilang : pecahan ---> integer
25     Pembilang(P) mengembalikan nilai pemb atau pembilang dari tipe <pemb, peny>
26  Penyebut : pecahan ---> integer > 0
27     Penyebut(P) mengembalikan nilai peny atau penyebut dari tipe <pemb, peny>
28
29  ****
30  DEFINISI DAN SPESIFIKASI KONSTRUKTOR
31
32  MakePecahanC : integer, integer >= 0, integer > 0 ---> pecahanc
33     MakePecahanC(bil, n, d) membuat tipe pecahan campuran dengan bilangan bil, nominator n, dan
    denominator d
34  MakePecahan : integer, integer > 0 ---> pecahan
35     MakePecahan(pemb, peny) membuat tipe pecahan dengan pembilang pemb dan penyebut peny
36
37  ****
38  DEFINISI DAN SPESIFIKASI OPERATOR
39
40  KonversiPecahan : pecahanc ---> pecahan
41     KonversiPecahan(P) mengonversi pecahan campuran menjadi pecahan biasa
42  KonversiReal : pecahanc ---> real
43     KonversiReal(P) mengonversi pecahan campuran menjadi bilangan real
44  AddP : 2 pecahanc ---> pecahanc
```

```

45     AddP(P1, P2) menambah P1 dengan P2
46 SubP : 2 pecahanc ---> pecahanc
47     SubP(P1, P2) mengurangi P1 dengan P2
48 DivP : 2 pecahanc ---> pecahanc
49     DivP(P1, P2) membagi P1 dengan P2
50 MulP : 2 pecahanc ---> pecahanc
51     MulP(P1, P2) mengalikan P1 dengan P2
52
53 *****
54 DEFINISI DAN SPESIFIKASI FUNGSI PREDIKAT
55
56 IsEqP : 2 pecahanc ---> boolean
57     IsEqP(P1, P2) memeriksa apakah P1 sama dengan P2
58 IsLtP : 2 pecahanc ---> boolean
59     IsLtP(P1, P2) memeriksa apakah P1 lebih kecil daripada P2
60 IsGtP : 2 pecahanc ---> boolean
61     IsGtP(P1, P2) memeriksa apakah P1 lebih besar daripada P2
62
63 *****
64 DEFINISI DAN SPESIFIKASI FUNGSI ANTARA
65
66 KaliDenominator : 2 pecahanc ---> integer
67     KaliDenominator(P1, P2) mengalikan nilai denominator P1 dengan P2
68 KaliNoDe : 2 pecahanc ---> integer
69     KaliNoDe(P1, P2) mengalikan nilai nominator P1 dengan denominator P2
70 KaliPembPeny : 2 pecahanc ---> integer
71     KaliPembPeny(P1, P2) mengalikan nilai pembilang P1 dengan penyebut P2 setelah keduanya
    dikonversi menjadi pecahan biasa
72 KaliPemb : 2 pecahanc ---> integer
73     KaliPemb(P1, P2) mengalikan nilai pembilang P1 dengan pembilang P2 setelah keduanya
    dikonversi menjadi pecahan biasa
74 SimpNom : integer, 2 pecahanc ---> integer
75     SimpNom(hasilOp, P1, P2) memberikan nilai nominator yang bergantung pada nilai perkalian
    denominator
76 SimpPemb : integer, 2 pecahanc ---> integer
77     SimpPemb(hasilOp, P1, P2) memberikan nilai nominator yang bergantung pada nilai perkalian
    antara pembilang dan penyebut
78
79 *****
80
81 REALISASI
82 *****
83 ""
84
85
86 def Bilangan(P):
87     return P[0]
88
89
90 def Nominator(P):
91     return P[1]

```

```

92
93
94 def Denominator(P):
95     return P[2]
96
97
98 def Pembilang(P):
99     return P[0]
100
101
102 def Penyebut(P):
103     return P[1]
104
105
106 # Konstruktor
107 def MakePecahanC(bil, n, d):
108     return [bil, n, d]
109
110
111 def MakePecahan(n, d):
112     return [n, d]
113
114
115 # Fungsi Antara
116 def KaliDenominator(P1, P2):
117     return (
118         Denominator(P1) * Denominator(P2)
119         if Denominator(P1) != Denominator(P2)
120         else Denominator(P1)
121     )
122
123
124 def KaliNoDe(P1, P2):
125     return Nominator(P1) * Denominator(P2)
126
127
128 def KaliPembPeny(P1, P2):
129     return Pembilang(KonversiPecahan(P1)) * Penyebut(KonversiPecahan(P2))
130
131
132 def KaliPemb(P1, P2):
133     return Pembilang(KonversiPecahan(P1)) * Pembilang(KonversiPecahan(P2))
134
135
136 def SimpNom(hasilOp, P1, P2):
137     return (
138         hasilOp
139         if abs(hasilOp) < KaliDenominator(P1, P2)
140         else hasilOp % KaliDenominator(P1, P2)
141     )

```

```

142
143
144 def SimpPemb(hasilOp, P1, P2):
145     return (
146         hasilOp * -1
147         if hasilOp < abs(KaliPembPeny(P2, P1))
148         else hasilOp % abs(KaliPembPeny(P2, P1))
149     )
150
151
152 # Predikat
153 def IsEqP(P1, P2):
154     return (KaliPembPeny(P1, P2)) == (KaliPembPeny(P2, P1))
155
156
157 def IsLtP(P1, P2):
158     return (KaliPembPeny(P1, P2)) < (KaliPembPeny(P2, P1))
159
160
161 def IsGtP(P1, P2):
162     return (KaliPembPeny(P1, P2)) > (KaliPembPeny(P2, P1))
163
164
165 # Operator
166 def KonversiPecahan(P):
167     if Bilangan(P) < 0:
168         return MakePecahan(
169             (Bilangan(P) * Denominator(P)) - Nominator(P), Denominator(P)
170         )
171     else:
172         return MakePecahan(
173             (Bilangan(P) * Denominator(P)) + Nominator(P), Denominator(P)
174         )
175
176
177 def KonversiReal(P):
178     return Pembilang(KonversiPecahan(P)) / Penyebut(KonversiPecahan(P))
179
180
181 def AddP(P1, P2):
182     return MakePecahanC(
183         Bilangan(P1)
184         + Bilangan(P2)
185         + ((KaliNoDe(P1, P2) + KaliNoDe(P2, P1)) // KaliDenominator(P1, P2)),
186         SimpNom((KaliNoDe(P1, P2) + KaliNoDe(P2, P1)), P1, P2),
187         KaliDenominator(P1, P2),
188     )
189
190
191 def SubP(P1, P2):

```

```

192     return MakePecahanC(
193         Bilangan(P1)
194         - Bilangan(P2)
195         - (abs(KaliNoDe(P1, P2) - KaliNoDe(P2, P1)) // KaliDenominator(P1, P2)),
196         SimpNom((KaliNoDe(P1, P2) - KaliNoDe(P2, P1)), P1, P2),
197         KaliDenominator(P1, P2),
198     )
199
200
201 def DivP(P1, P2):
202     return MakePecahanC(
203         (KaliPembPeny(P1, P2) // KaliPembPeny(P2, P1))
204         + (1 if KaliPembPeny(P2, P1) < 0 else 0),
205         SimpPemb(KaliPembPeny(P1, P2), P1, P2),
206         abs(KaliPembPeny(P2, P1)),
207     )
208
209
210 def MulP(P1, P2):
211     return MakePecahanC(
212         (KaliPemb(P1, P2) // KaliDenominator(P1, P2))
213         + (1 if KaliPemb(P1, P2) < 0 else 0),
214         (KaliPemb(P1, P2)) % KaliDenominator(P1, P2),
215         KaliDenominator(P1, P2),
216     )
217
218
219 """
220 APLIKASI
221 *****
222 """
223
224 print(AddP(MakePecahanC(1, 1, 2), MakePecahanC(1, 2, 4)))
225 print(SubP(MakePecahanC(1, 1, 2), MakePecahanC(1, 3, 4)))
226 print(DivP(MakePecahanC(1, 1, 2), MakePecahanC(-1, 1, 4)))
227 print(MulP(MakePecahanC(1, 1, 2), MakePecahanC(1, 1, 4)))
228 print(IsEqP(MakePecahanC(1, 1, 2), MakePecahanC(1, 2, 4)))
229 print(IsLtP(MakePecahanC(1, 1, 2), MakePecahanC(1, 1, 4)))
230 print(IsGtP(MakePecahanC(1, 1, 2), MakePecahanC(1, 1, 4)))
231

```