

PEMBAHASAN SOAL UAS DASPRO 2019/2020

1. Membuat fungsi dalam bahasa pemrograman Python

a. max2()

```
#max2: 2 integer --> integer
# max2(a,b) menentukan bilangan maksimum dari 2 bilangan integer
#Realisasi

def max2(a,b):
    if(a > b):
        return a
    else:
        return b
```

b. min2()

```
#min2: 2 integer --> integer
# min2 (a,b) menghitung rata-rata dari dua buah bilangan integer
#Realisasi

def min2 (a,b) :
    if(a < b):
        return a
    else:
        return b
```

c. max_list()

```
#max_list: list --> integer
# max_list(L) menentukan bilangan maksimum dari sebuah list
#Realisasi

def max_list(L):
    if(IsEmpty(L)):
        return 0
    elif(IsOneElmt(L)):
        return FirstElmt(L)
    else:
        return max2(FirstElmt(L), max_list(Tail(L)))
```

d. min_list()

```
#min_list: list --> integer
# min_list(L) menentukan bilangan minimum dari sebuah list
#Realisasi

def min_list(L):
    if(IsEmpty(L)):
        return 0
    elif(IsOneElmt(L)):
        return FirstElmt(L)
    else:
        return min2(FirstElmt(L), min_list(Tail(L)))
```

2. Membuat fungsi dalam bahasa pemrograman Python

a. total_elemen_daun()

```
#total_elemen_daun: PohonBiner --> integer
# total_elemen_daun(P) menghitung jumlah daun pada pohon biner
#Realisasi

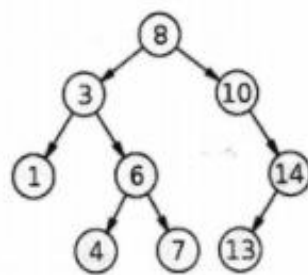
def total_elemen_daun(P):
    if(is_one_element(P)):
        return Akar(P)
    else:
        if(is_biner(P)):
            return total_elemen_daun(left(P)) + total_elemen_daun(right(P))
        elif(is_uner_left(P)):
            return total_elemen_daun(left(P))
        elif(is_uner_right(P)):
            return total_elemen_daun(right(P))
```

b. total_elemen_node()

```
#total_elemen_node: PohonBiner --> integer
# total_elemen_node(P) menghitung jumlah semua elemen pada pohon biner
#Realisasi

def total_elemen_node(P):
    if(is_one_element(P)):
        return Akar(P)
    else:
        if(is_biner(P)):
            return total_elemen_node(left(P)) + akar(P) + total_elemen_node(right(P))
        elif(is_uner_left(P)):
            return total_elemen_node(left(P)) + akar(P)
        elif(is_uner_right(P)):
            return akar(P) + total_elemen_node(Right(P))
```

c. Kunjungan secara Pre-Order, In-Order, dan Post-Order dari Pohon Biner P di bawah :



- Pre-Order = [8, 3, 1, 6, 4, 7, 10, 14, 13]
- In-Order = [1, 3, 4, 6, 7, 8, 13, 14, 10]
- Post-Order = [1, 4, 7, 6, 3, 13, 14, 10, 8]

3. Ekspresi lambda

a. Menulis fungsi dalam bahasa Python

```
#is_genap: integer --> boolean
# is_genap(x) bernilai true jika integer tersebut bernilai genap
#Realisasi
def is_genap(x):
    if x % 2 == 0
        return True

#is_ganjil: integer --> boolean
# is_ganjil(L) bernilai true jika integer tersebut bernilai ganjil
#Realisasi
def is_ganjil(x):
    if x % 2 != 0
        return True

#Filter_list: list --> list
# Filter_list(L) mem-filter elemen list yang bernilai ganjil atau genap
#Realisasi
def filter_list(L,f):
    if is_empty(L):
        return[]
    if not (f(first_elmt(L))):
        return filter_list(tail(L),f)
    else
        return konso(first_elmt(L),filter_list(tail),f))
```

b. Menulis ekspresi lambda untuk menghasilkan L2 dan L3

```
L2 = filter_list(L1, lambda x : x%2 == 0)
L3 = filter_list(L1, lambda x : x%2 !=0)
```

4. Himpunan

```
#is_member: elemen, list --> boolean
# is_member(x,L) bernilai benar jika x adalah elemen dari list L
#Realisasi
def is_member(x,L):
    if(IsEmpty(L)):
        return False
    else:
        if(FirstElmt(L) == x):
            return True
        else:
            return is_member(x, Tail(L))

#is_sub_set: 2 list --> boolean
# is_sub_set(H1,H2) bernilai True jika semua elemen H1 adalah elemen H2
#Realisasi
def is_sub_set(H1,H2):
    if IsEmpty(H1):
        return True
    elif not is_member(FirstElmt(H1), H2):
        return False
    else:
        return is_sub_set(Tail(H1), H2)
```

```
# Minus(H1,H2) menampilkan himpunan hasil selisih antara himpunan H1 dan H2
#Realisasi
def Minus(H1,H2):
    if is_sub_set(H1,H2):
        return []
    elif is_member(FirstElmt(H1),H2):
        return Minus(Tail(H1),H2)
    else :
        return Konso(FirstElmt(H1),Minus(Tail(H1),H2))
```