

LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 4 : PENCARIAN



Disusun Oleh :
MHD. FARHAN LUBIS
L200220277
F

TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2024

Daftar Isi

Daftar Isi.....	2
1.11 Soal – Soal Mahasiswa.....	3
5. Buatlah suatu program untuk mencari suatu item di sebuah linked list.	3
• Kode Program.....	3
• Screenshot hasil praktikum.....	4
6. Binary search. Ubahlah fungsi binSe di halaman 43 agar mengembalikan index lokasi elemen yang ditemukan.	4
• Kode Program.....	4
• Screenshot hasil praktikum.....	5
7. Binary search. Ubahlah fungsi binSe itu agar mengembalikan semua index lokasi elemen yang ditemukan.....	5
• Kode Program.....	5
• Screenshot hasil praktikum.....	6
8. Pada permainan tebak angka yang sudah kamu buat di Modul 1 (soal nomer 12, halaman 15),6	
• Kode Program.....	7
• Screenshot hasil praktikum.....	7

1.11 Soal – Soal Mahasiswa

5. Buatlah suatu program untuk mencari suatu item di sebuah linked list.

- **Kode Program**

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None
    def add(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            return
        last_node = self.head
        while last_node.next:
            last_node = last_node.next
        last_node.next = new_node
    def search(self, target):
        current = self.head
        while current:
            if current.data == target:
                return f"Angka {target} ditemukan"
            current = current.next
        return f"Angka {target} tidak ditemukan"
    def display(self):
        current = self.head
        while current:
            print(current.data, end=" ")
            current = current.next
        print()

ll = LinkedList()
ll.add(9)
ll.add(4)
ll.add(3.5)
ll.add(8.2)
ll.add(2)
ll.display()
print(ll.search(4))
print(ll.search(8.3))

print("\nProgram Completed!\n\n--- By L200220277 ---")
```

Kode 4.5 program pencarian item linked list

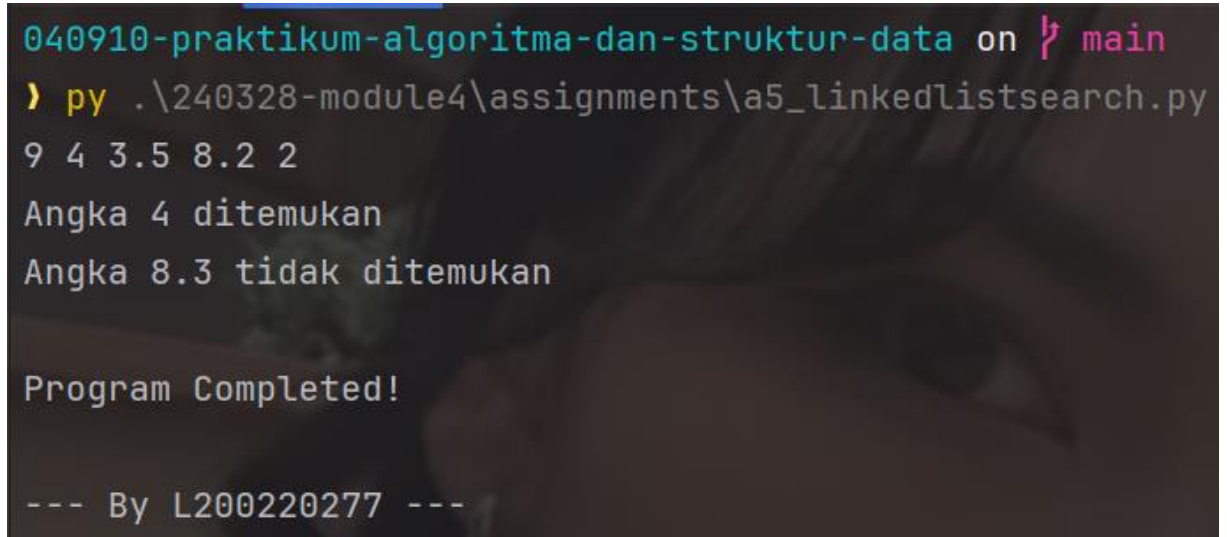
PENJELASAN:

Method add dimulai dengan membuat node baru dengan data yang diambil dari argumen. Jika linked list kosong, node baru ini akan menjadi head linked list. Jika tidak, method ini mencari node terakhir dalam linked list dengan melakukan iterasi untuk setiap node sampai node terakhir. Setelah node terakhir ditemukan, node baru akan ditambahkan setelahnya.

Method search digunakan untuk mencari angka tertentu dalam linked list. Method ini dimulai dari head linked list, dan melakukan iterasi setiap node. Jika angka yang dicari ditemukan, kembalikan pesan ditemukan. Jika tidak, kembalikan pesan tidak ditemukan.

Method display digunakan untuk menampilkan semua node dalam linked list. Method ini melakukan iterasi setiap node dimulai dari head, dan setiap node dicetak.

- **Screenshot hasil praktikum**



```
040910-praktikum-algoritma-dan-struktur-data on main
> py .\240328-module4\assignments\a5_linkedlistsearch.py
9 4 3.5 8.2 2
Angka 4 ditemukan
Angka 8.3 tidak ditemukan

Program Completed!

--- By L200220277 ---
```

Gambar 4.5 output a5_linkedlistsearch.py

6. **Binary search.** Ubahlah fungsi binSe di halaman 43 agar mengembalikan index lokasi elemen yang ditemukan.

Kalau tidak ketemu, akan mengembalikan False

- **Kode Program**

```
def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1
    while low <= high:
        mid = (high + low) // 2
        if kumpulan[mid] == target:
            return mid
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False

data = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
angka = 9
hasil = binSe(data, angka)
if hasil:
    print(f"Angka {angka} ditemukan pada indeks ke-{hasil} data ini")
else:
    print(f"Angka {angka} tidak ditemukan pada data ini")
```

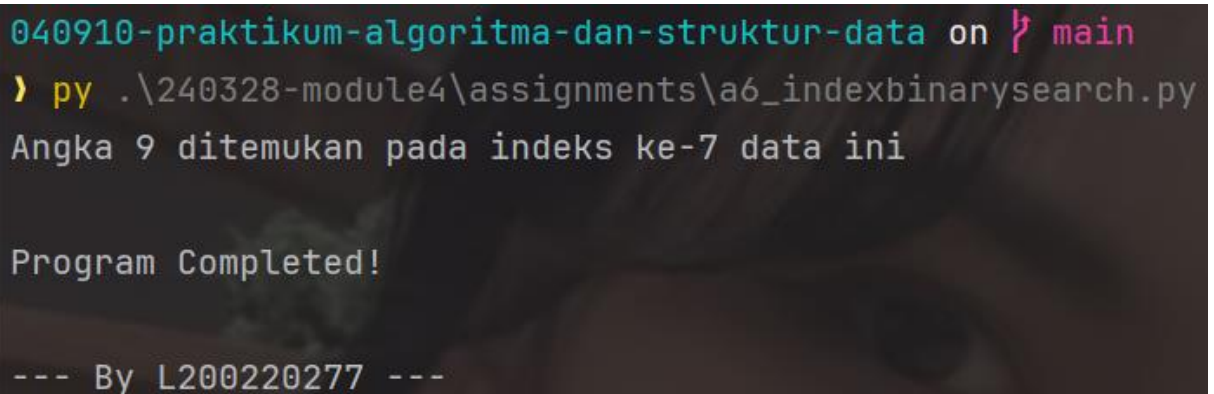
```
print("\nProgram Completed!\n\n--- By L200220277 ---")
```

Kode 4.6 program pencarian biner indeks

PENJELASAN:

Function binSe dimulai dengan menginisialisasi variable low dan high, yang menjadi rentang pencarian dalam list data yangurut. Selama low tidak melebihi high, variable mid akan dideklarasikan sebagai indeks tengah yang merupakan jumlah high dan low di bagi dan di bulatkan dengan 2. Jika angka pada indeks mid sama dengan target, function mengembalikan indeks mid sebagai tanda target di temukan. Jika target lebih kecil dari angka di indeks mid, high dideklarasikan ulang dengan setengah kiri dari rentang sebelumnya. Jika target lebih besar, low dideklarasikan ulang dengan setengah kanan rentang dari rentang sebelumnya. Proses ini berulang sampai rentang selesai. Jika tidak ada yang cocok dengan target, akan dikembalikan False.

- **Screenshot hasil praktikum**



```
040910-praktikum-algoritma-dan-struktur-data on main
> py .\240328-module4\assignments\a6_indexbinarysearch.py
Angka 9 ditemukan pada indeks ke-7 data ini

Program Completed!

--- By L200220277 ---
```

Gambar 4.6 output a6_indexbinarysearch.py

7. Binary search. Ubahlah fungsi binSe itu agar mengembalikan semua index lokasi elemen yang ditemukan.

Contoh: mencari angka 6 pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14] akan mengembalikan [3, 4, 5]. Karena sudah urut, “tinggal melihat kiri dan kanannya”.

- **Kode Program**

```
def binSe(kumpulan, target):
    temp = []
    low = 0
    high = len(kumpulan) - 1
    while low <= high:
        mid = (high + low) // 2
        if kumpulan[mid] == target:
            temp.append(mid)
            left = mid - 1
            while left >= 0 and kumpulan[left] == target:
                temp.append(left)
                left -= 1
            right = mid + 1
            while right < len(kumpulan) and
kumpulan[right] == target:
```

```

        temp.append(right)
        right += 1
    return temp
elif kumpulan[mid] > target:
    high = mid - 1
else:
    low = mid + 1
return False

data = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
angka = 6
hasil = binSe(data, angka)
if hasil:
    print(f"Angka {angka} ditemukan pada indeks {hasil} data ini")
else:
    print(f"Angka {angka} tidak ditemukan pada data ini")

print("\nProgram Completed!\n\n--- By L200220277 ---")

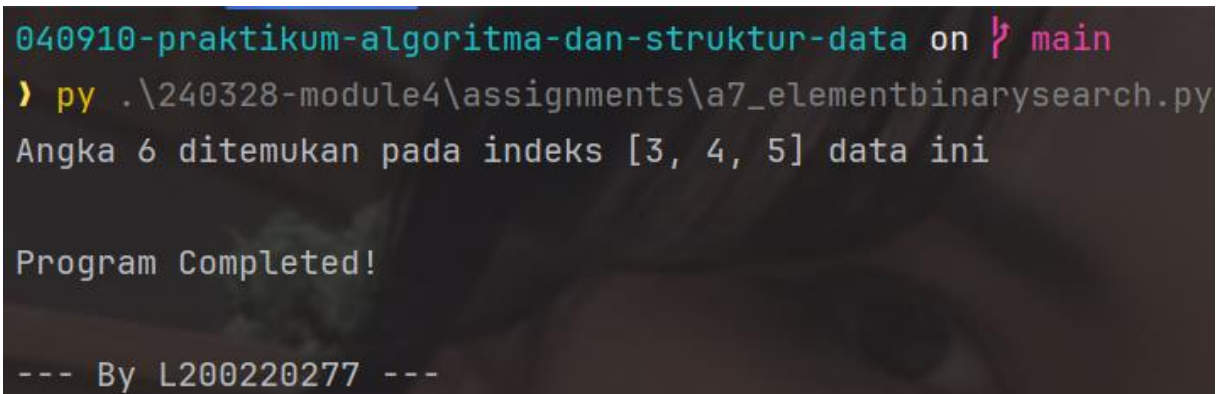
```

Kode 4.7 program pencarian biner elemen

PENJELASAN:

Function binSe dimulai dengan menetapkan variable low dan high sebagai rentang pencarian dalam list. Kemudian, list kosong temp dibuat untuk menyimpan indeks di mana target ditemukan. Melalui sebuah loop while, function ini membagi rentang pencarian menjadi setengah-setengah secara berulang sampai target atau rentang tidak bisa dibagi lagi. Pada setiap iterasi, indeks mid akan di hitung dan mengecek apakah angka pada indeks sama dengan target. Jika ya, indeks mid ditambahkan ke temp, dan kemudian pencarian dilanjutkan ke kiri dan kanan dari mid untuk mencari semua indeks di mana target ditemukan. Setelah semua indeksnya ditemukan, temp dikembalikan. Jika target tidak ditemukan dalam list, function ini mengembalikan False.

- **Screenshot hasil praktikum**



```

040910-praktikum-algoritma-dan-struktur-data on main
> py .\240328-module4\assignments\a7_elementbinarysearch.py
Angka 6 ditemukan pada indeks [3, 4, 5] data ini

Program Completed!

--- By L200220277 ---

```

Gambar 4.7 output a7_elementbinarysearch.py

8. Pada permainan tebak angka yang sudah kamu buat di Modul 1 (soal nomer 12, halaman 15),

kalau angka yang harus ditebak berada di antara 1 dan 100, seharusnya maksimal jumlah tebakan adalah 7. Kalau antara 1 dan 1000, maksimal jumlah tebakan adalah 10. Mengapa seperti itu? Bagaimanakah polanya?

- **Kode Program**

```
from random import randint
import math

def tebak_angka(awal, akhir):
    angka_acak = randint(awal, akhir)
    maks_tebakan = math.ceil(math.log2(akhir - awal + 1))
    tebakan = 0
    print("Permainan tebak angka.")
    print(f"Bilangan bulat ini terdapat diantara {awal} sampai {akhir}. Coba tebak.")
    while True:
        tebakan += 1
        angka_tebakan = int(input(f"Masukkan tebakan ke-{tebakan}:> "))
        if angka_tebakan < angka_acak:
            print("Itu terlalu kecil. Coba lagi.")
        elif angka_tebakan > angka_acak:
            print("Itu terlalu besar. Coba lagi.")
        else:
            print(f"Ya. Anda benar!")
            break
        if tebakan == maks_tebakan:
            print(f"Batas maksimum tebakan yaitu ({maks_tebakan}) tercapai. Jawabannya adalah {angka_acak}.")
            break

tebak_angka(1, 100)
print()
tebak_angka(1, 1000)

print("\nProgram Completed!\n\n--- By L200220277 ---")
```

Kode 4.8 program tebak angka


PENJELASAN:

Prinsip dalam permainan tebak angka adalah mempersempit rentang angka yang berisi jawabannya dengan memperhatikan hint nya di setiap tebakan. Dalam hal ini, pola yang digunakan untuk menentukan jumlah maksimal tebakan adalah menggunakan logaritma basis 2 dari jumlah angka dalam rentang tertentu yang dibulatkan ke atas agar jumlah maksimal tebakan tidak berupa pecahan. Misal:

Untuk rentang angka antara 1 dan 100, jumlah angkanya sebesar 100 maka jumlah tebakan didapat dari $\log_2(100) \approx 6.64$. Kemudian dibulatkan ke atas menjadi 7 sehingga angka ini menjadi jumlah tebakan maksimal.

Selanjutnya untuk rentang angka antara 1 dan 1000, jumlah angkanya sebesar 1000 maka jumlah tebakan didapat dari $\log_2(1000) \approx 9.96$. Kemudian dibulatkan ke atas menjadi 10 sehingga angka ini menjadi jumlah tebakan maksimal.

- **Screenshot hasil praktikum**

```
040910-praktikum-algoritma-dan-struktur-data on  main  
> py .\240328-module4\assignments\a8_guessnumber.py  
Permainan tebak angka.  
Bilangan bulat ini terdapat diantara 1 sampai 100. Coba tebak.  
Masukkan tebakan ke-1:> 1  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-2:> 3  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-3:> 5  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-4:> 7  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-5:> 9  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-6:> 11  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-7:> 13  
Itu terlalu kecil. Coba lagi.  
Batas maksimum tebakan yaitu (7) tercapai. Jawabannya adalah 25.
```



```
Permainan tebak angka.  
Bilangan bulat ini terdapat diantara 1 sampai 1000. Coba tebak.  
Masukkan tebakan ke-1:> 2  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-2:> 4  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-3:> 5  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-4:> 6  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-5:> 8  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-6:> 10  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-7:> 12  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-8:> 14  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-9:> 16  
Itu terlalu kecil. Coba lagi.  
Masukkan tebakan ke-10:> 18  
Itu terlalu kecil. Coba lagi.  
Batas maksimum tebakan yaitu (10) tercapai. Jawabannya adalah 250.  
  
Program Completed!  
  
--- By L200220277 ---
```

Gambar 4.8 output *a8_guessnumber.py*