

Minggu, 30 Juni 2019

LAPORAN KEGIATAN (BINARY SEARCH)

A. DASAR TEORI

1. Binary Search (Pencarian Bagi Dua) adalah algoritma pencarian untuk data yang terurut. Pencarian dilakukan dengan cara menebak apakah data yang dicari berada ditengah-tengah data, kemudian membandingkan data yang dicari dengan data yang ada ditengah. Bila data yang ditengah sama dengan data yang dicari, berarti data ditemukan. Namun, bila data yang dicari kemungkinan berada disebelah kiri dari data tengah dan data disebelah kanan data tengah dapat diabai. *Upper bound* dari bagian data kiri yang baru adalah indeks dari data tengah itu sendiri. Sebaliknya, bila data yang ditengah lebih kecil dari data yang dicari, maka dapat dipastikan bahwa data yang dicari kemungkinan besar berada disebelah kanan dari data tengah. *Lower bound* dari data disebelah kanan dari data tengah adalah indeks dari data tengah itu sendiri ditambah 1. Demikian seterusnya.

Berikut adalah ilustrasi algoritma binary search:

Level 0:	1	2	3	4	5	5	6	7	8	8	9	10	11	13	14	17	20
Level 1:	1	2	3	4	5	5	6	7	8	8	9	10	11	13	14	17	20
Level 2:	1	2	3	4	5	5	6	7	8	8	9	10	11	13	14	17	20
Level 3:	1	2	3	4	5	5	6	7	8	8	9	10	11	13	14	17	20
Level 4:	1	2	3	4	5	5	6	7	8	8	9	10	11	13	14	17	20

Found, position=4
Searching for element 4
Middle element is red.

B. PEMBAHASAN

1. Pada praktikum [pertama](#) ini akan mencoba membuat program untuk mencari angka 5 pada suatu array dimana indeks dari angka tersebut akan disimpan menggunakan algoritma binary search.

Sudah ada array **A** satu dimensi, insialisasi isinya dengan ilustrasi sebagai berikut:

0	1	2	3	4	5	6	7	8	9
12	15	7	10	25	2	17	25	5	20

Algoritma

Pertama membuat List untuk menyimpan angka-angka-nya. Kemudian membuat suatu perulangan selama angka paling ujung sebelah kanan kurang dari sama dengan angka paling ujung sebelah kanan dan angka yang dicari belum ditemukan maka akan melakukan pencarian angka. Di dalam perulangan tersebut menambahkan suatu percabangan dimana jika angka-angka pada List tersebut sama dengan angka yang ingin dicari maka simpan indeksny. Atau jika angka yang dicari kurang dari angka tengah maka posisi indeks angka tengah akan berpindah ke sebelah kiri sehingga angka yang di sebelah kiri tersebut akan menjadi indeks tengah. Atau jika angka yang dicari lebih dari angka tengah maka posisi indeks angka tengah akan berpindah ke sebelah kanan sehingga akan yang di sebelah kanan tersebut akan menjadi indeks tengah.

```

< > RezkyYustisioHadiPratama_Jobsheet8_Soal1.py x
1  data = sorted([12, 15, 7, 10, 25, 2, 17, 25, 5, 20])
2  key = 5
3  l = 0
4  r = len(data) - 1
5  ditemukan = False
6  perbandingan = 0
7  while l <= r and (not ditemukan):
8      m = int((l + r) / 2)
9      perbandingan += 1
10     if data[m] == key:
11         ditemukan = True
12     elif key < data[m]:
13         r = m - 1
14     elif key > data[m]:
15         l = m + 1
16 if not ditemukan:
17     print("Data tidak ditemukan")
18 else:
19     print("Sorted: ", end="")
20     for d in data:
21         print(d, end=" ")
22     print()
23     print(key, "berada di indeks", m)

```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk menampung angka-angka dengan menggunakan List. Kemudian angka-angka yang ada di dalam List tersebut diurutkan menggunakan perintah `sorted`.
- Pada baris nomor 2, berfungsi untuk tempat menyimpan angka yang akan dicari.
- Pada baris nomor 3, berfungsi untuk menentukan indeks pertama yang berada di sebelah kanan pada List yaitu indeks ke-0.

- Pada baris nomor 4, berfungsi untuk menentukan jumlah keluruhan indeks angka pada List
- Pada baris nomor 7, berfungsi untuk melakukan suatu perulangan selama angka dengan indeks pertama sebelah kiri kurang dari sama dengan angka indeks terakhir di sebelah kanan pada List dan angka belum ditemukan maka akan melakukan pencarian angka.
- Pada baris nomor 8, berfungsi untuk mencari angka indeks tengah kemudian di konversi menjadi bilangan integer dengan menggunakan perintah `int`.
- Pada baris nomor 10 sampai 15, berfungsi untuk melakukan suatu percabangan dimana jika angka indeks tengah sama dengan angka yang dicari maka angka telah ditemukan. Atau jika angka yang dicari kurang dari angka indeks tengah, maka indeks terakhir akan pindah posisi ke sebelah kiri. Atau jika angka yang dicari lebih dari angka indeks tengah, maka indeks pertama akan pindah posisi ke sebelah kanan.
- Pada baris nomor 16 sampai 17, berfungsi untuk menampilkan pesan "Data tidak ditemukan".
- Pada baris nomor 18 sampai 23, berfungsi untuk menampilkan angka terurut dan indeks angka yang dicari.

<Hasil Program di jalankan menggunakan IDLE Python>

```
Sorted: 2 5 7 10 12 15 17 20 25 25
5 berada di indeks 1
>>> |
```

Output Program

2. Pada praktikum kedua ini akan mengembangkan program pada praktikum pertama, program kali ini adalah membuat suatu inputan yang berguna untuk memasukan data. Kemudian membuat sebuah fungsi yang mana fungsi tersebut akan berisikan algoritma binary search.

Algoritma

Pertama membuat List untuk menyimpan angka-angka-nya. Kemudian membuat suatu perulangan selama angka paling ujung sebelah kanan kurang dari sama dengan angka paling ujung sebelah kanan dan angka yang dicari belum ditemukan maka akan melakukan pencarian angka. Di dalam perulangan tersebut menambahkan suatu percabangan dimana jika angka-angka pada List tersebut sama dengan angka yang ingin dicari maka simpan indeksnyanya. Atau jika angka yang dicari kurang dari angka tengah maka posisi indeks angka tengah akan berpindah ke sebelah kiri sehingga angka yang di sebelah kiri tersebut akan menjadi indeks tengah. Atau jika angka yang dicari lebih dari angka tengah maka posisi indeks angka tengah akan berpindah ke sebelah kanan sehingga akan yang di sebelah kanan tersebut akan menjadi indeks tengah.

```

< > RezkyYustisioHadiPratama_Jobsheet8_Soal2.py x
1  def binary_search(data, key):
2      l = 0
3      r = len(data) - 1
4      ditemukan = False
5      perbandingan = 0
6      while l <= r and (not ditemukan):
7          m = int((l + r) / 2)
8          perbandingan += 1
9          if data[m] == key:
10             ditemukan = True
11         elif key < data[m]:
12             r = m - 1
13         elif key > data[m]:
14             l = m + 1
15     if not ditemukan:
16         print("Data tidak ditemukan")
17     else:
18         print("Sorted: ", end="")
19         for d in data:
20             print(d, end=" ")
21         print()
22         print(key, "berada di indeks", m)
23
24 data = sorted([int(s) for s in input("Masukan angka: ").split()]])
25 key = 5
26 binary_search(data, key)

```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk membuat sebuah fungsi dengan nama **binary_search** dengan menerima 2 parameter yaitu List angka dan angka yang dicari.
- Pada baris nomor 2, berfungsi untuk menentukan indeks pertama yang berada di sebelah kanan pada List yaitu indeks ke-0.

- Pada baris nomor 3, berfungsi untuk menentukan jumlah keluruhan indeks angka pada List
- Pada baris nomor 6, berfungsi untuk melakukan suatu perulangan selama angka dengan indeks pertama sebelah kiri kurang dari sama dengan angka indeks terakhir di sebelah kanan pada List dan angka belum ditemukan maka akan melakukan pencarian angka.
- Pada baris nomor 7, berfungsi untuk mencari angka indeks tengah kemudian di konversi menjadi bilangan integer dengan menggunakan perintah `int`.
- Pada baris nomor 9 sampai 14, berfungsi untuk melakukan suatu percabangan dimana jika angka indeks tengah sama dengan angka yang dicari maka angka telah ditemukan. Atau jika angka yang dicari kurang dari angka indeks tengah, maka indeks terakhir akan pindah posisi ke sebelah kiri. Atau jika angka yang dicari lebih dari angka indeks tengah, maka indeks pertama akan pindah posisi ke sebelah kanan.
- Pada baris nomor 15 sampai 16, berfungsi untuk menampilkan pesan "Data tidak ditemukan".
- Pada baris nomor 17 sampai 22, berfungsi untuk menampilkan angka terurut dan indeks angka yang dicari.
- Pada baris nomor 24, berfungsi untuk menampung angka-angka dengan menggunakan List. Kemudian angka-angka yang ada di dalam List tersebut diurutkan menggunakan perintah `sorted`.
- Pada baris nomor 25, berfungsi untuk tempat menyimpan angka yang akan dicari.
- Pada baris nomor 26, berfungsi untuk memanggil nama fungsi **binary_search** sekaligus memasukkan 2 buah parameter, yaitu List angka dan angka yang dicari.

<Hasil Program di jalankan menggunakan IDLE Python>

```
Masukan angka: 12 15 7 10 25 2 17 25 5 20  
Sorted: 2 5 7 10 12 15 17 20 25 25  
5 berada di indeks 1  
>>> |
```

Output Program

C. KESIMPULAN

Binary Search (pencarian bagi dua) adalah salah satu teknik pencarian data dimana data dicari secara urut dengan menghilangkan setengah data pada setiap langkah.

D. DAFTAR PUSTAKA

Wibowo,Fitri. __, *Binary Search*, [pdf]