

PRAKTIKUM 13
PRAKTEK SEARCHING

Disusun untuk Memenuhi Matakuliah Algoritma dan Struktur Data

Dosen Pengampu : Dr Arna Fariza S.Kom., M.Kom.



Oleh :

Nama : Fauzan Abderrasheed

NRP : 3123500020

Kelas : D3 IT A

Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer
Program Studi Teknik Informatika

Mei 2024

A. SOAL

1. Sequential search
2. Binary search
3. Sequential sort
4. Sequential rekursif
5. Binary rekursif

B. LAPORAN RESMI

Berikut merupakan **Source Code** dari soal diatas dalam bentuk terpisah-pisah sesuai fungsinya

1. Sequential Search Source Code

```
1  int Sequential(int key, int data[]) {
2      int i=0, ketemu=0;
3
4      while(!ketemu && i < N) {
5          if (key == data[i]) {
6              ketemu = 1;
7          } else {
8              i++;
9          }
10     }
11     if (ketemu) {
12         return i;
13     } else {
14         return -1;
15     };
16 }
```

2. Binary Search Source Code

```
1  int Binary(int key, int data[]) {
2      int ketemu = 0, L = 0, R = N - 1, m;
3
4      while (!ketemu && L <= R) {
5          m = (L + R) / 2;
6          if (key == data[m]) {
7              ketemu = 1;
8          } else if (key < data[m]) {
9              R = m - 1;
10         } else {
11             L = m + 1;
12         }
13     }
14     if (ketemu) {
15         return m;
16     } else {
17         return -1;
18     }
19 }
```

3. Sequential Rekursif Source Code

```
1  int SequentialRekursif(int i, int key, int data[]) {
2      if (i == N) {
3          return -1;
4      }
5
6      if (data[i] == key) {
7          return i;
8      }
9
10     return SequentialRekursif(i + 1, key, data);
11 }
```

4. Binary Rekursif Source Code

```
1
2  int BinaryRekursif(int key, int data[], int L, int R) {
3      if (L > R) {
4          return -1;
5      }
6
7      int m = (L + R) / 2;
8      if (data[m] == key) {
9          return m;
10     } else if (key < data[m]) {
11         return BinaryRekursif(key, data, L, m - 1);
12     } else {
13         return BinaryRekursif(key, data, m + 1, R);
14     }
15 }
```

5. Sequential Sort Source Code

```
1  int SequentialSorted(int key, int data[])
2  {
3      int i = 0;
4      bool ketemu = false;
5
6      while (!ketemu && i <= N) {
7          if (data[i] == key) {
8              ketemu = true;
9          } else {
10             i++;
11         }
12     }
13
14     if (ketemu) {
15         return i;
16     } else {
17         return -1;
18     }
19 }
```

Berikut merupakan **Output** dari soal diatas dalam bentuk terpisah-pisah sesuai fungsinya

1. Sequential Search Output

```
33 58 84 9 35 61 54 80 5 31
masukkan key = 9
1. Sequential
2. Sequential Rekursif
3. Binary
4. Binary Rekursif
5. Quicksort Sequential
Pilihan: 1
Data ditemukan di index 3
Lagi? n
```

2. Binary Search Output

```
60 13 66 19 72 25 46 99 52 4
masukkan key = 72
1. Sequential
2. Sequential Rekursif
3. Binary
4. Binary Rekursif
5. Quicksort Sequential
Pilihan: 3
Data ditemukan di index 4
Lagi? █
```

3. Sequential Sort Output

```
4 100 97 94 90 87 51 48 45 41
masukkan key = 41
1. Sequential
2. Sequential Rekursif
3. Binary
4. Binary Rekursif
5. Quicksort Sequential
Pilihan: 5
4 41 45 48 51 87 90 94 97 100
Data ditemukan di index 1
Lagi? n
```

4. Sequential Rekursif Output

```
44 81 18 55 92 29 66 71 8 46
masukkan key = 92
1. Sequential
2. Sequential Rekursif
3. Binary
4. Binary Rekursif
5. Quicksort Sequential
Pilihan: 2
Data ditemukan di index 4
Lagi? n
```

5. Binary Rekursif Output

```
49 90 32 73 15 56 98 7 49 90
masukkan key = 15
1. Sequential
2. Sequential Rekursif
3. Binary
4. Binary Rekursif
5. Quicksort Sequential
Pilihan: 4
Data ditemukan di index 4
Lagi? n
```

Berikut merupakan bentuk program secara keseluruhan

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <time.h>
5  #define N 10
6
7  int Sequential(int key, int data[]) {
8      int i=0, ketemu=0;
9
10     while(!ketemu && i < N) {
11         if (key == data[i]) {
12             ketemu = 1;
13         } else {
14             i++;
15         }
16     }
17     if (ketemu) {
18         return i;
19     } else {
20         return -1;
21     };
22 }
23
24 int SequentialRekursif(int i, int key, int data[]) {
25     if (i == N) {
26         return -1;
27     }
28
29     if (data[i] == key) {
30         return i;
31     }
32
33     return SequentialRekursif(i + 1, key, data);
34 }
35
36 int Binary(int key, int data[]) {
37     int ketemu = 0, L = 0, R = N - 1, m;
38
39     while (!ketemu && L <= R) {
40         m = (L + R) / 2;
41         if (key == data[m]) {
42             ketemu = 1;
43         } else if (key < data[m]) {
44             R = m - 1;
45         } else {
46             L = m + 1;
47         }
48     }
49     if (ketemu) {
50         return m;
51     } else {
52         return -1;
53     }
54 }
55
```

```

1  int BinaryRekursif(int key, int data[], int L, int R) {
2      if (L > R) {
3          return -1;
4      }
5
6      int m = (L + R) / 2;
7      if (data[m] == key) {
8          return m;
9      } else if (key < data[m]) {
10         return BinaryRekursif(key, data, L, m - 1);
11     } else {
12         return BinaryRekursif(key, data, m + 1, R);
13     }
14 }
15
16 int Partition(int p, int r, int A[]) {
17     int i, j, pivot, temp;
18     pivot = A[p];
19     i = p;
20     j = r;
21
22     while (i <= j) {
23
24         while(A[i] < pivot) {i++;};
25         while(A[j] > pivot) {j--};
26
27         if(i < j){
28             temp = A[i];
29             A[i] = A[j];
30             A[j] = temp;
31             i++;
32             j--;
33         }
34         else {
35             return j;
36         }
37     }
38     return j;
39 }
40
41 void Quicksort(int p, int r, int A[]) {
42     int q;
43     if (p < r) {
44         q = Partition(p, r, A);
45         Quicksort(p, q, A);
46         Quicksort(q+1, r, A);
47     }
48 }
49
50 int SequentialSorted(int key, int data[])
51 {
52     int i = 0;
53     bool ketemu = false;
54
55     while (!ketemu && i <= N) {
56         if (data[i] == key) {
57             ketemu = true;
58         } else {
59             i++;
60         }
61     }
62
63     if (ketemu) {
64         return i;
65     } else {
66         return -1;
67     }
68 }

```

```

1  main() {
2      int data[N], i, pilih, key, index;
3      char lagi = 'y';
4      for (i = 0; i < N; i++) {
5          srand(time(NULL) * (i+1));
6          data[i] = rand() % 100+1;
7          printf("%d ", data[i]);
8      }
9      printf("\n");
10
11     while (lagi=='y') {
12         // key = rand() % 100+1;
13         // printf("Key = %d\n", key);
14
15         printf("masukkan key = ");
16         scanf("%d", &key);
17
18         printf("1. Sequential\n");
19         printf("2. Sequential Rekursif\n");
20         printf("3. Binary\n");
21         printf("4. Binary Rekursif\n");
22         printf("5. Quicksort Sequential\n");
23
24         printf("Pilihan: ");
25         fflush(stdin);
26         scanf("%d", &pilih);
27
28         switch(pilih) {
29             case 1:
30                 index = Sequential(key, data);
31                 break;
32             case 2:
33                 index = SequentialRekursif(0, key, data);
34                 break;
35             case 3:
36                 index = Binary(key, data);
37                 break;
38             case 4:
39                 index = BinaryRekursif(key, data, 0, N - 1);
40                 break;
41             case 5:
42                 Quicksort(0, N-1, data);
43                 index = SequentialSorted(key, data);
44                 for (int i = 0; i < N; i++)
45                 {
46                     printf("%d ", data[i]);
47                 }
48                 printf("\n");
49                 break;
50         }
51         if (index >= 0) {
52             printf("Data ditemukan di index %d\n", index);
53         } else {
54             printf("Data tidak ditemukan\n");
55         }
56
57         printf("Lagi? ");
58         fflush(stdin);
59         scanf("%c", &lagi);
60     }
61 }

```

Analisa

Program diatas merupakan program yang memberikan opsi kepada user untuk memilih metode apa yang ingin digunakan untuk mencari sebuah data yang ada data random yang dibagikan oleh program. Pada program di atas diberikan 5 opsi metode yaitu sequential, binary, sequential rekursif, binary rekursif, dan sequential sorted. Dimana sequential sorted disini adalah sama seperti metode sequential hanya saja data diurutkan terlebih dahulu, pada program diatas data diurutkan menggunakan metode quick sort. Sequential rekursif dan binary rekursif disini sama dengan metode sequential dan binary biasa, hanya saja program berjalan secara rekursif. Sedangkan 2 metode biasa dalam algoritma searching yaitu adalah sequential dan binary, dimana sequential disini adalah metode dimana data dibaca dari depan ke belakang dan diperiksa apakah data sama dengan key yang dicari oleh user, sedangkan binary disini adalah data dibaca dengan cara membandingkan nilai tengah terlebih dahulu dengan key yang ingin dicari pengguna dan akan berjalan sampe nilai tengah atau nilai yang terakhir ada pada barisan data itu sama dengan key.

C. KESIMPULAN

Kesimpulan yang dapat diberikan adalah yaitu pada 2 metode yang ada, yaitu binary dan sequential, dapat dimodifikasi sesuai dengan kebutuhan dan perbandingan keefektifan dari kedua metode tersebut. Metode tersebut dapat dimodifikasi menjadi berjalan secara rekursif, atau dengan cara diurutkan terlebih dahulu lalu mencari datanya yang sama dengan key yang diberikan oleh user.