

Searching

1.1 Kompetensi

1. Mahasiswa mampu menjelaskan mengenai algoritma *Searching*.
2. Mahasiswa mampu membuat dan mendeklarasikan struktur algoritma *Searching*.
3. Mahasiswa mampu menerapkan dan mengimplementasikan algoritma *Searching*.

1.2 Alat Dan Bahan:

- PC/Leptop
- Netbeans

1.3 Ulasan Teori:

Searching merupakan proses untuk menemukan suatu data atau informasi dari sekumpulan data/informasi yang ada. Algoritma pencarian/*searching* algorithm merupakan algoritma yang menerima suatu kata kunci sebagai kriteria pencarian, dan dengan langkah-langkah tertentu akan mencari rekaman yang sesuai dengan kata kunci tersebut. Di dalam jobsheet ini akan diperkenalkan teknik pencarian data yaitu pencarian beruntun/sequential searching

A. Pencarian Beruntun/Sequential Search

Sequential Search adalah teknik pencarian data dimana data dicari secara urut dari depan ke belakang atau dari awal sampai akhir. Kelebihan dari proses pencarian secara sequential ini jika data yang dicari terletak didepan, maka data akan ditemukan dengan cepat. Tetapi dibalik kelebihannya ini, teknik ini juga memiliki kekurangan. Pertama, jika data yang dicari terletak dibelakang atau paling akhir, maka akan membutuhkan waktu yang lama dalam proses pencariannya. Kedua, beban komputer akan semakin bertambah jika jumlah data dalam array sangat banyak.

Teknik pencarian data dari array yang paling mudah adalah dengan cara *sequential search*, dimana data dalam array dibaca 1 demi satu, diurutkan dari index terkecil ke index terbesar, maupun sebaliknya.

B. Algoritma Pencarian Beruntun/Sequential Search

Algoritma pencarian berurutan dapat dituliskan sebagai berikut :

1. $i \leftarrow 0$
2. ditemukan \leftarrow false
3. Selama (tidak ditemukan) dan $(i \leq N)$ kerjakan baris 4

4. Jika ($\text{Data}[i] = x$) maka ditemukan $\leftarrow \text{true}$, jika tidak $i \leftarrow i + 1$
5. Jika (ditemukan) maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

Contoh :

int a[5] = {0,3,6,10,1} (index array dimulai dari index ke 0) jika kita ingin mencari bilangan 6 dalam array tersebut, maka proses yang terjadi kita mencari

- a. dari array index ke-0, yaitu 0, dicocokkan dengan bilangan yang akan dicari, jika tidak sama, maka mencari ke index berikutnya
- b. pada array index ke-1, juga bukan bilangan yang dicari, maka kita mencari lagi pada index berikutnya
- c. pada array index ke-2, ternyata bilangan yang kita cari ada ditemukan, maka kita keluar dari looping pencarian.

1.4 Langkah Praktikum:

1. Praktikum 1 (Pencarian Beruntun/Sequential Search)

1. Buatlah project baru pada Netbeans dengan nama **“TestSequentialSearch”**
2. Buat class **“SequentialSearch”** , kemudian deklarasikan variabel berikut ini:

```
12 public class SequentialSearch {
13
14     public int[] data;
15     public int jumData;
16
17
18 }
```

3. Buatlah konstruktor dengan parameter Data[] dan jumData

```
public SequentialSearch(int Data[], int jumData){
}
}
```

4. Isi konstruktor tersebut dengan kode berikut!

```
jumData=jumData;
data=new int[jumData];
for(int i=0; i<jumData; i++){
    data[i]=Data[i];
}
```

5. Buatlah method **“Find”** bertipe integer dengan parameter **“dicari”** bertipe integer.

```
public int Find(int dicari){  
  
}
```

6. Deklarasikan isi method Find dengan algoritma pencarian data menggunakan teknik sequential searching

```
public int Find(int dicari){  
    int posisi=-1;  
    for (int j=0; j<jumData; j++){  
        if(data[j]==dicari){  
            posisi=j;  
            break;  
        }  
    }  
    return posisi;  
}
```

7. Buatlah method “**Tampilkan Data**” bertipe void

```
public void TampilkanData(){  
  
}
```

8. Deklarasikan isi dari method TampilData

```
for(int i=0; i<jumData;i++){  
    System.out.print(data[i]+" ");  
}  
System.out.println();
```

9. Buatlah array pada class “**TestSequentialSearch**” dengan nama **data** kemudian isi array tersebut!

```
int data[]={10,30,42,12,17,24,50,85};
```

10. Buatlah objek baru dengan nama **pencarian** yang merupakan instansiasi dari class SequentialSearch, kemudian isi parameternya!

```
SequentialSearch pencarian=new SequentialSearch(data, 8);
```

11. Lakukan pemanggilan method **TampilkanData** pada class SequentialSearch.

```
System.out.println("Isi Array adalah: ");  
pencarian.TampilkanData();
```

12. Deklarasikan variabel dicari bertipe integer dan isi nilainya dengan nilai yang ingin dicari

```
int dicari=10;
```

13. Deklarasikan variabel posisi bertipe integer yang memanggil method Find pada class SequentialSearch.

```
int posisi=pencarian.Find(dicari);
```

14. Lakukan pengecekan untuk kondisi data yang dicari.

```
if (posisi !=-1){  
    System.out.println("Data : " + dicari+" ditemukan pada indeks "+ posisi );  
}  
else {  
    System.out.println("Data : " + dicari+" Tidak ditemukan" );  
}
```

15. Jalankan program, maka hasilnya adalah sebagai berikut:

```
run:  
Isi Array adalah:  
10 30 42 12 17 24 50 85  
Data : 10 ditemukan pada indeks 0  
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Pertanyaan

1. Apa yang dimaksud dengan kode berikut ini:

```
SequentialSearch pencarian=new SequentialSearch(data, 8);
```

2. Jelaskan fungsi kode program berikut ini!

```
public SequentialSearch(int Data[], int jmlData){  
    jumData=jmlData;  
    data=new int[jmlData];  
    for(int i=0; i<jumData; i++){  
        data[i]=Data[i];  
    }  
}
```

3. Jelaskan fungsi **break** pada kode program dibawah ini!

```
if(data[j]==dicari){  
    posisi=j;  
    break;  
}
```

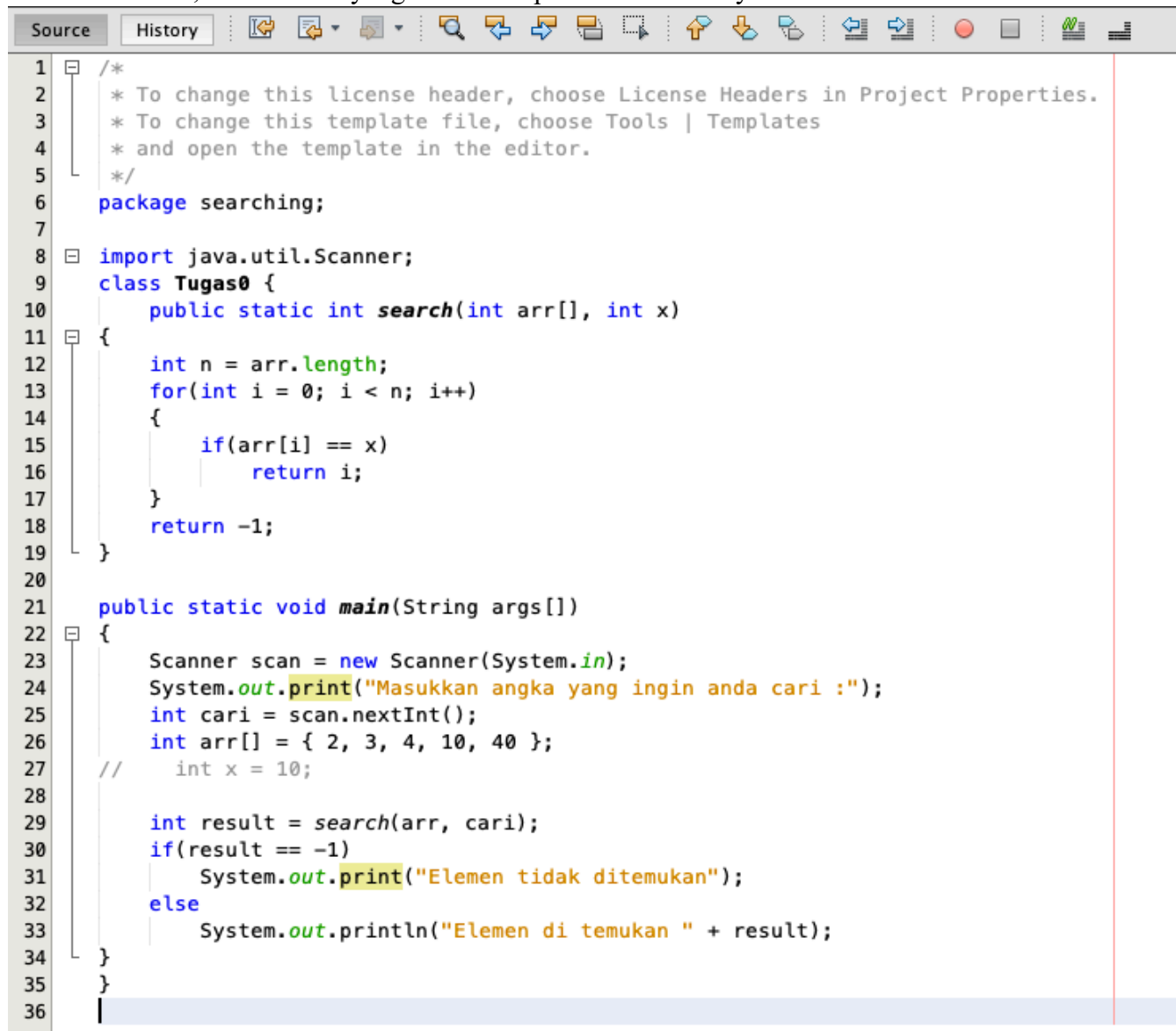
4. Modifikasilah program diatas yang mana data array dapat di inputkan atau bersifat dinamis!

1.5 Tugas

1. Terdapat sebuah data array 2 dimensi sebagai berikut:

indeks	0	1	2	3	4
0	45	78	7	200	80
1	90	1	17	100	50
2	21	2	40	18	65

Berdasarkan data di atas buatlah program untuk melakukan pencarian data pada array dua dimensi tersebut, dimana data yang dicari di inputkan melalui keyboard!



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package searching;
7
8  import java.util.Scanner;
9  class Tugas0 {
10     public static int search(int arr[], int x)
11     {
12         int n = arr.length;
13         for(int i = 0; i < n; i++)
14         {
15             if(arr[i] == x)
16                 return i;
17         }
18         return -1;
19     }
20
21     public static void main(String args[])
22     {
23         Scanner scan = new Scanner(System.in);
24         System.out.print("Masukkan angka yang ingin anda cari :");
25         int cari = scan.nextInt();
26         int arr[] = { 2, 3, 4, 10, 40 };
27         // int x = 10;
28
29         int result = search(arr, cari);
30         if(result == -1)
31             System.out.print("Elemen tidak ditemukan");
32         else
33             System.out.println("Elemen di temukan " + result);
34     }
35 }
36
```

2. Terdapat sebuah data array 1 dimensi sebagai berikut

0	1	2	3	4	5	6	7	8	9
12	17	2	1	70	50	90	17	2	90

Buatlah program untuk mencari dan mencetak isi array yang nilainya terbesar, dan mencetak ada berapa buah nilai terbesar tersebut serta berada dilokasi mana saja nilai terbesar tersebut!

```

Source History
1 package searching;
2 public class Tugas1 {
3     int[] data;
4
5     void Print(){
6         for(int i = 0; i < data.length; i++){
7             System.out.print(""+data[i]+" - ");
8         }
9     }
10
11     int max(){
12         int mx = data[0];
13         for(int i=1; i<data.length; i++){
14             if(data[i]>mx){
15                 mx = data[i];
16                 // System.out.println(""+mx);
17             }
18         }
19         return mx;
20     }
21
22     void Posisi(int x){
23         for(int i=0; i<data.length; i++){
24             if(data[i]==x)
25                 System.out.println("Posisi"+i+"");
26         }
27     }
28
29
30     public static void main(String[] args) {
31         int data []={12,17,2,1,70,50,90,17,2,90};
32         Tugas1 dt1 = new Tugas1();
33         dt1.data= data;
34         dt1.Print();
35         System.out.println("Max =" +dt1.max());
36
37         dt1.Posisi(dt1.max());
38
39     }
40
41 }

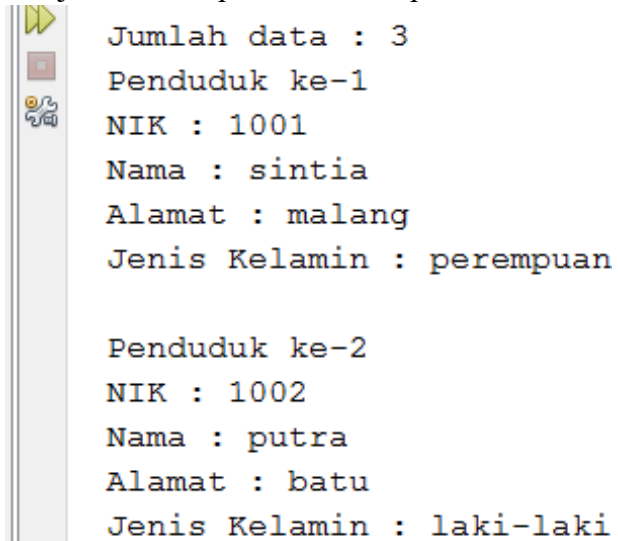
```

3. Buatlah program untuk pencarian data nama yang mana jumlah panjang data berupa inputan dan data nama tersebut di inputkan menggunakan keyboard!

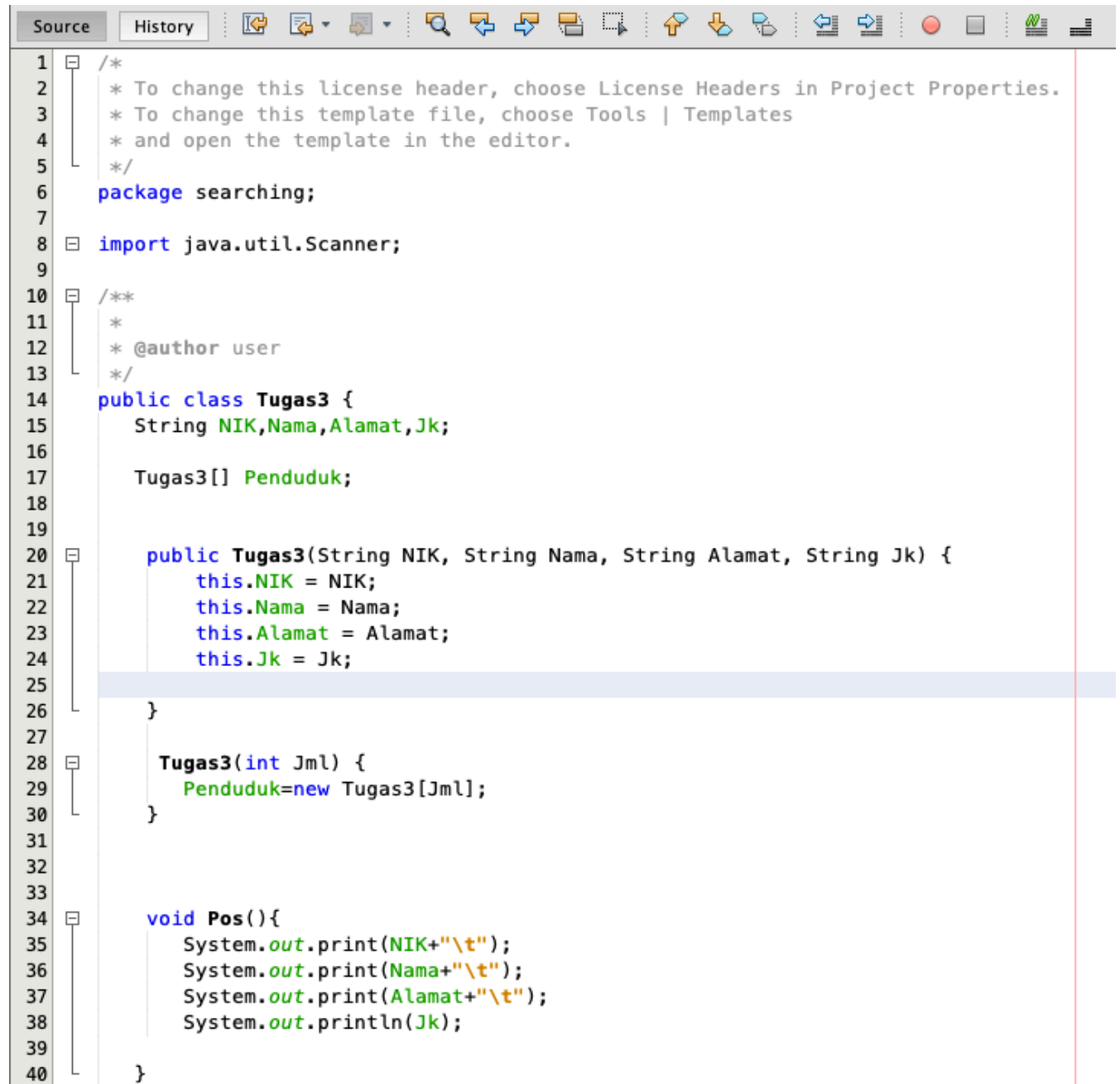
```
run:
Jumlah data : 4
Masukkan nama ke-0
ilham
Masukkan nama ke-1
sinta
Masukkan nama ke-2
putri
Masukkan nama ke-3
putra
Masukkan Nama yang dicari:
sinta
Nama : sinta Ditemukan Pada Index :1.
BUILD SUCCESSFUL (total time: 17 seconds)
```

```
Source History
1 package searching;
2 import java.util.Scanner;
3 public class Tugas2 {
4     public String nama;
5     String Penonton[][] ;
6
7     void Posisi (String x){
8         for (int i=0;i<Penonton.length;i++){
9             for(int j=0;j<Penonton.length;j++){
10                 if(Penonton[i][j].equalsIgnoreCase(x)){
11                     System.out.println("Posisi"+" "+i+" "+j);
12                 }
13             }
14         }
15
16     }
17 }
18
19
20 public static void main(String[] args) {
21     Scanner sc = new Scanner (System.in);
22     String cari;
23     Tugas2 a = new Tugas2();
24
25     String[][] b={
26         {"Santi","Lukman","Tiara","iren","Umar"},
27         {"Fani","Janet","Eva","Winda","Yona"},
28         {"Tio","Ronald","Nami","Dila","bobi"}};
29     a.Penonton=b;
30
31     System.out.println("cari elemnt Array");
32     cari= sc.nextLine();
33     a.Posisi(cari);
34 }
35 }
36 }
37 }
```


4. Buatlah program untuk melakukan pencarian data penduduk berdasarkan NIK yang mana data penduduk tersebut terdiri dari NIK, nama, alamat, dan jenis kelamin. Data penduduk dan jumlah data penduduk di inputkan melalui keyboard.

A screenshot of a Java Swing window with a light gray title bar and a white content area. On the left side of the window, there is a vertical toolbar with four icons: a green double arrow pointing right, a red square, a yellow circle with a black dot, and a blue speech bubble. The text output is as follows:

```
Jumlah data : 3  
Penduduk ke-1  
NIK : 1001  
Nama : sintia  
Alamat : malang  
Jenis Kelamin : perempuan  
  
Penduduk ke-2  
NIK : 1002  
Nama : putra  
Alamat : batu  
Jenis Kelamin : laki-laki
```



```

41
42 void Cari(String a){
43     Boolean flag=false;
44     for(int i=0; i<Penduduk.length;i++){
45         if(a.equalsIgnoreCase(Penduduk[i].NIK)){
46             Penduduk[i].Pos();
47             flag=true;
48         }
49
50     if(flag==false) {
51
52         System.out.println("Maaf data tidak ada");
53     }
54 }
55
56
57
58 public static void main(String[] args) {
59     Scanner sc= new Scanner(System.in);
60     System.out.print("Masukan Jumlah Penduduk :");
61     int Jml=sc.nextInt();
62     Tugas3 t3=new Tugas3(Jml);
63     for(int i=0;i<Jml;i++){
64
65         System.out.print("NIK : ");
66         String NIK=sc.next();
67         System.out.print("Nama : ");
68         String Nama=sc.next();
69         System.out.print("Alamat : ");
70         String Alamat=sc.next();
71         System.out.print("Jenis Kelamin : ");
72         String Jk=sc.next();
73         t3.Penduduk[i]=new Tugas3(NIK, Nama, Alamat, Jk);
74     }
75     System.out.println("");
76
77     for(int i=0;i<Jml;i++){
78         t3.Penduduk[i].Pos();
79     }
80     System.out.print("Masukan NIK yang anda Cari :");
81     String cari=sc.next();
82     t3.Cari(cari);
83
84 }
85
86
87

```



Penduduk ke-3

NIK : 1003



Nama : budi

Alamat : malang

Jenis Kelamin : laki-laki

Data Kependudukan

```
+---+-----+-----+-----+
{ Nn { NIK { NRNR { RLRNR£ {IN IS KELRN IN
+---+-----+-----+-----+
1 1 1 1001 1 sintia malang lperempuan 1
+---+-----+-----+-----+
1 2 1 1002 1 putra batu llaki-laki 1
+---+-----+-----+-----+
1 3 1 1003 1 budi malang llaki-laki 1
+---+-----+-----+-----+n
```

t Cari Data Pendudukt

Masukkan NIK : 1002

ndata di temukan...

```
+---+-----+-----+-----+
{ Nn { NIK { NRNR { RLRNR£ {IN IS KELRN IN
+---+-----+-----+-----+
1 2 11002 1 putra 1 batu llaki-laki 1
+---+-----+-----+-----+n
```

BUILD SUCCESSFUL (total time: 1 minute 42 seconds)