

LAPORAN PRAKTIKUM

“Pertemuan ke-2: Post Test – Array Statis”

Diajukan untuk memenuhi salah satu praktikum Mata Struktur Data Informatika yang di
ampu oleh:

Dr., Ardiansyah, S.T., M.Cs.



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

A / Selasa 09.00 – 10.30 Lab. Komputasi Dasar

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023**

- Gunakan struct untuk data KTP dan class untuk operasi input dan output (Sesuaikan dengan urutan KTP) data yang dimaksudkan dalam array. Gunakan data anggota kelompok untuk program anda.

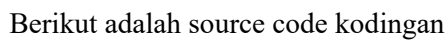
Jawab:

Berikut KTP kelompok saya:



Dan 2 KTP sebagai contoh lainnya





Pertemuan ke-2: Post Test – Array Statis



1960

```
Banyak KTP: 4
Menu:
1. Input Data
2. Tampilkan Data
3. Keluar
Pilihan: █
```

```

Provinsi: PROVINSI DKI JAKARTA
Kota: JAKARTA BARAT
NIK: 3171234567890123
Nama: MIRA SETIAWAN
Tempat Lahir: JAKARTA
Tanggal Lahir: 18
Bulan Lahir: 2
Tahun Lahir: 1986
Jenis kelamin (0=laki-laki, 1=perempuan): 1
Golongan Darah: B
Alamat: JL. PASTI CEPAT A7/66
RT: 7
RW: 8
Kel/Desa: PEGADUNGAN
Kecamatan: KALIDERES
Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konghucu): Islam
Status Perkawinan (Kawin, Belum Kawin): Kawin
Pekerjaan: PEGAWAI SWASTA
Data berhasil diinput
Press any key to continue . . .
    
```



Akan langsung tersimpan di database di data.txt

```
data.txt
File Edit View
NIK : 3203012503770011
Nama : GUOHUI CHEN
Tempat/Tgl Lahir : FUJIAN, 25-03-1977
Jenis Kelamin : LAKI-LAKI
Gol.Darah : A
Alamat : JL. SELAMET PERUMAHAN RANCABALI NO.40
RT/RW : 002/004
Kel/Desa : MUKA
Kecamatan : CIANJUR
Agama : KRISTEN
Status Perkawinan : KAWIN
Pekerjaan : OTHERS
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat : 11-10-2023
=====
Provinsi : PROVINSI DKI JAKARTA
Kota : JAKARTA BARAT
NIK : 3171234567890123
Nama : MIRA SETIAWAN
Tempat/Tgl Lahir : JAKARTA, 18-02-1986
Jenis Kelamin : PEREMPUAN
Gol.Darah : B
Alamat : JL. PASTI CEPAT A7/66
RT/RW : 007/008
Kel/Desa : PEGADUNGAN
Kecamatan : KALIDERES
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : PEGAWAI SWASTA
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat : 11-10-2023
=====
```

Tampilan menu lihat Data

```

PROBLEMS  TERMINAL  OUTPUT  DEBUG CONSOLE  PORTS  GITLENS
=====
                                KARTU TANDA PENDUDUK
=====
                                PROVINSI KALIMANTAN TENGAH
                                KABUPATEN KOTAWARINGIN BARAT
NIK          : 6201052907040001
Nama         : RENDIE ABDI SAPUTRA
Tempat/Tgl Lahir : PANGKALAN BUN, 29-07-2004
Jenis Kelamin  : PEREMPUAN                      Gol.Darah : B
Alamat       : JL. A. YANI
RT/RW        : 006/003
Kel/Desa     : SUMBER AGUNG
Kecamatan    : PANGKALAN LADA
Agama       : ISLAM
Status Perkawinan : BELUM KAWIN                      Foto.jpg
Pekerjaan    : PELAJAR/MAHASISWA
Kewarganegaraan : WNI                      02-08-2021
Berlaku Hingga : SEUMUR HIDUP                      TTD
=====
                                PROVINSI BANTEN
                                KOTA TANGERANG SELATAN
NIK          : 3674072703040003
Nama         : MOHAMMAD FARID HENDIANTO
Tempat/Tgl Lahir : TANGERANG, 27-03-2004
Jenis Kelamin  : PEREMPUAN                      Gol.Darah : B
Alamat       : BATAN INDAH BLOK J-14
RT/RW        : 003/004
Kel/Desa     : KADEMANGAN
Kecamatan    : SETU
Agama       : Islam
Status Perkawinan : Belum Kawin                      Foto.jpg
Pekerjaan    : PELAJAR/MAHASISWA
Kewarganegaraan : WNI                      27-05-2021
Berlaku Hingga : SEUMUR HIDUP                      TTD
=====
                                PROVINSI DAERAH ISTIMEWA YOGYAKARTA
                                KOTA YOGYAKARTA
NIK          : 3471135407040001
Nama         : AISYAH SYAFI'I NURJANNAH
Tempat/Tgl Lahir : YOGYAKARTA, 14-07-2004
Jenis Kelamin  : LAKI-LAKI                      Gol.Darah : B
Alamat       : GLAGAH UH 4/67
RT/RW        : 012/003
Kel/Desa     : WARUNGBOTO
Kecamatan    : UMBULHARJO
Agama       : ISLAM
Status Perkawinan : BELUM KAWIN                      Foto.jpg
Pekerjaan    : PELAJAR/MAHASISWA
Kewarganegaraan : WNI                      24-06-2021
Berlaku Hingga : SEUMUR HIDUP                      TTD
=====
                                PROVINSI DAERAH ISTIMEWA YOGYAKARTA
                                KOTA YOGYAKARTA
NIK          : 3203012503770011
Nama         : GUOHUI CHEN
Tempat/Tgl Lahir : FUJIAN, 25-03-1977
Jenis Kelamin  : PEREMPUAN                      Gol.Darah : A

```



```

data.txt
File Edit View

Provinsi      : PROVINSI KALIMANTAN TENGAH
Kota/Kabupaten : KABUPATEN KOTAWARINGIN BARAT
NIK           : 6201052907040001
Nama          : RENDIE ABDI SAPUTRA
Tempat/Tgl Lahir : PANGKALAN BUN, 29-07-2004
Jenis Kelamin  : LAKI-LAKI
Gol.Darah     : B
Alamat        : JL. A. YANI
               RT/RW      : 006/003
               Kel/Desa    : SUMBER AGUNG
               Kecamatan   : PANGKALAN LADA
Agama         : ISLAM
Status Perkawinan : BELUM KAWIN
Pekerjaan     : PELAJAR/MAHASISWA
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat        : 02-08-2021
=====
Provinsi      : PROVINSI BANTEN
Kota/Kabupaten : KOTA TANGERANG SELATAN
NIK           : 3674072703040003
Nama          : MOHAMMAD FARID HENDIANTO
Tempat/Tgl Lahir : TANGERANG, 27-03-2004
Jenis Kelamin  : LAKI-LAKI
Gol.Darah     : B
Alamat        : BATAN INDAH BLOK J-14
               RT/RW      : 003/004
               Kel/Desa    : KADEMANGAN
               Kecamatan   : SETU
Agama         : Islam
Status Perkawinan : Belum Kawin
Pekerjaan     : PELAJAR/MAHASISWA
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat        : 27-05-2021

Ln 78, Col 40

```

Berikut adalah Output fullnya:

```

KARTU TANDA PENDUDUK

PROVINSI KALIMANTAN TENGAH
KABUPATEN KOTAWARINGIN BARAT
NIK : 6280182987840001
Nama : BENDIE ABDI SAPUTRA
Tempat/Tgl Lahir : PANGKALAN BUN, 29-07-2004
Jenis Kelamin : PEREMPUAN Gol.Darah : B
Alamat : 3L. A. YANI
RT/RW : 006/003
Kel/Desa : SIKBER AGUNG
Kecamatan : PANGKALAN LADA
Agama : ISLAM
Status Perkawinan : BELUM KAWIN Foto.jpg
Pekerjaan : PELAJAR/MAHASISWA
Kewarganegaraan : MNI 02-08-2021
Berlaku Hingga : SELUMUR HIDUP TTD

PROVINSI BANTEN
KOTA TANGERANG SELATAN
NIK : 3674872703840003
Nama : MOHAMMAD FARID HENDIANTO
Tempat/Tgl Lahir : TANGERANG, 29-03-2004
Jenis Kelamin : PEREMPUAN Gol.Darah : B
Alamat : BATAN INDAH BLOK 3-14
RT/RW : 003/004
Kel/Desa : KADEMANGAN
Kecamatan : SETU
Agama : Islam
Status Perkawinan : Belum Kawin Foto.jpg
Pekerjaan : PELAJAR/MAHASISWA
Kewarganegaraan : MNI 27-05-2021
Berlaku Hingga : SELUMUR HIDUP TTD

PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KOTA YOGYAKARTA
NIK : 3471135407840001
Nama : AISYAH SYAFI'I NURJANNAH
Tempat/Tgl Lahir : YOGYAKARTA, 14-07-2004
Jenis Kelamin : LAKI-LAKI Gol.Darah : B
Alamat : GLAGAH UH 4/67
RT/RW : 012/003
Kel/Desa : WARUNGOTO
Kecamatan : UMBULHARJO
Agama : ISLAM
Status Perkawinan : BELUM KAWIN Foto.jpg
Pekerjaan : PELAJAR/MAHASISWA
Kewarganegaraan : MNI 24-06-2021
Berlaku Hingga : SELUMUR HIDUP TTD

PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KOTA YOGYAKARTA
NIK : 3203012503770011
Nama : GUSALU QIN
Tempat/Tgl Lahir : FUDJIAN, 25-03-1977
Jenis Kelamin : PEREMPUAN Gol.Darah : A
Alamat : 3L. SELAMET PERUMAHAN RANCARALI NO.40
RT/RW : 002/004
Kel/Desa : MUKA
Kecamatan : CIANJUR
Agama : KRISTEN
Status Perkawinan : KAWIN Foto.jpg
Pekerjaan : OTHERS
Kewarganegaraan : MNI 11-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

PROVINSI DKI JAKARTA
JAKARTA BARAT
NIK : 3171234567890123
Nama : MIRA SETIAWAN
Tempat/Tgl Lahir : JAKARTA, 18-02-1986
Jenis Kelamin : LAKI-LAKI Gol.Darah : B
Alamat : 3L. PASTI CEPAT A7/66
RT/RW : 007/008
Kel/Desa : PESALINGAN
Kecamatan : KALDERES
Agama : ISLAM
Status Perkawinan : KAWIN Foto.jpg
Pekerjaan : PEKAWAI SWASTA
Kewarganegaraan : MNI 11-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

Press any key to continue . . .

```

Input Data dari File

Sudah terlihat bahwa program saya memiliki data.txt sebagai database. Sehingga apabila program dijalankan, maka data yang berada di data.txt akan tersimpan di program. Lihat pada class pada prosedur initialize().

Struct

Program ini menggunakan struct bernama data sebagai penyimpanan data KTP berbentuk Array.

```
12 struct Data {
13     std::string province;
14     std::string city;
15     std::string nik;
16     std::string name;
17     struct Born {
18         std::string place;
19         int date;
20         int month;
21         int year;
22     } born;
23
24     bool gender;
25     std::string bloodType;
26
27     struct Address {
28         std::string address;
29         int neighborhoodAssociation;
30         int communityAssociation;
31         std::string district;
32         std::string subDistrict;
33     } address;
34     std::string religion;
35     std::string isMarried;
36     std::string job;
37     std::string citizenship;
38     std::string validUntil;
39     struct Created {
40         int date;
41         int month;
42         int year;
43     } created;
44 };
```

Class

Program ini menggunakan Class bernama KTP.

```
class KTP {
private:
    Data data[MAX];
    int count = 0;

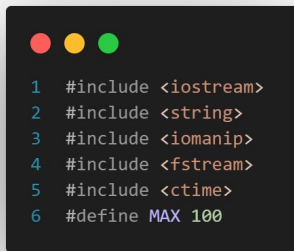
public:
    bool validateProvince(std::string province) {
        return province.length() <= 50;
    }

    bool validateCity(std::string city) {
        return city.length() <= 50;
    }

    bool validateNik(std::string nik) {
        return nik.length() == 16;
    }
    bool validateName(std::string name) {
        return name.length() <= 50;
    }
    bool validateBornPlace(std::string bornPlace) {
        return bornPlace.length() <= 20;
    }

    bool validateDate(int date) {
        return date >= 1 && date <= 31;
    }
}
```

Penjelasan lengkap kodingan

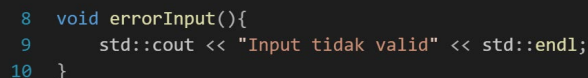


```
1 #include <iostream>
2 #include <string>
3 #include <iomanip>
4 #include <fstream>
5 #include <ctime>
6 #define MAX 100
```

Kodingan tersebut merupakan sebuah header file pada bahasa pemrograman C++. Header file tersebut memuat beberapa library yang akan digunakan dalam program, yaitu:

- `iostream`: untuk input dan output stream
- `string`: untuk manipulasi string
- `iomanip`: untuk manipulasi input dan output
- `fstream`: untuk manipulasi file
- `ctime`: untuk manipulasi waktu

Selain itu, terdapat juga sebuah preprocessor directive yang didefinisikan dengan `#define MAX 100`. Preprocessor directive ini digunakan untuk mendefinisikan sebuah konstanta bernama `MAX` dengan nilai 100. Konstanta ini nantinya dapat digunakan dalam program untuk membatasi ukuran array atau variabel lainnya.



```
8 void errorInput(){
9     std::cout << "Input tidak valid" << std::endl;
10 }
```

Kode tersebut merupakan sebuah fungsi bernama `errorInput` yang tidak memiliki parameter dan tidak mengembalikan nilai apapun (`void`). Fungsi ini bertujuan untuk menampilkan pesan "Input tidak valid" ke dalam output stream menggunakan `std::cout`. Pesan tersebut diakhiri dengan karakter newline (`\n`) menggunakan `std::endl`. Fungsi ini dapat dipanggil dari bagian lain dalam program untuk menampilkan pesan error jika terjadi kesalahan input.

```

12 struct Data {
13     std::string province;
14     std::string city;
15     std::string nik;
16     std::string name;
17     struct Born {
18         std::string place;
19         int date;
20         int month;
21         int year;
22     } born;
23
24     bool gender;
25     std::string bloodType;
26
27     struct Address {
28         std::string address;
29         int neighborhoodAssociation;
30         int communityAssociation;
31         std::string district;
32         std::string subDistrict;
33     } address;
34     std::string religion;
35     std::string isMarried;
36     std::string job;
37     std::string citizenship;
38     std::string validUntil;
39     struct Created {
40         int date;
41         int month;
42         int year;
43     } created;
44 };

```

GitHub Copilot: Kode tersebut merupakan sebuah definisi dari sebuah struct bernama 'Data'. Struct ini memiliki beberapa anggota data yang terdiri dari tipe data string, int, bool, dan struct lainnya. Berikut adalah penjelasan dari setiap anggota data pada struct 'Data':

- 'province': string yang menyimpan nama provinsi
- 'city': string yang menyimpan nama kota
- 'nik': string yang menyimpan nomor induk kependudukan
- 'name': string yang menyimpan nama lengkap
- 'born': struct yang menyimpan tempat dan tanggal lahir
 - 'place': string yang menyimpan tempat lahir
 - 'date': int yang menyimpan tanggal lahir
 - 'month': int yang menyimpan bulan lahir

- `year`: int yang menyimpan tahun lahir
- `gender`: bool yang menyimpan jenis kelamin (true untuk laki-laki, false untuk perempuan)
- `bloodType`: string yang menyimpan golongan darah
- `address`: struct yang menyimpan alamat
 - `address`: string yang menyimpan alamat lengkap
 - `neighborhoodAssociation`: int yang menyimpan nomor RT
 - `communityAssociation`: int yang menyimpan nomor RW
 - `district`: string yang menyimpan nama kecamatan
 - `subDistrict`: string yang menyimpan nama kelurahan
- `religion`: string yang menyimpan agama
- `isMarried`: string yang menyimpan status pernikahan
- `job`: string yang menyimpan pekerjaan
- `citizenship`: string yang menyimpan kewarganegaraan
- `validUntil`: string yang menyimpan tanggal berakhirnya KTP
- `created`: struct yang menyimpan tanggal pembuatan KTP
 - `date`: int yang menyimpan tanggal pembuatan
 - `month`: int yang menyimpan bulan pembuatan
 - `year`: int yang menyimpan tahun pembuatan

Struct `Data` ini dapat digunakan untuk merepresentasikan data pada sebuah KTP (Kartu Tanda Penduduk).

```

46 class KTP {
47 private:
48     Data data[MAX];
49     int count = 0;

```

Kode tersebut merupakan definisi dari sebuah class bernama KTP. Class ini memiliki dua anggota data private, yaitu:

data: array dari struct Data dengan ukuran MAX yang menyimpan data KTP

count: int yang menyimpan jumlah data KTP yang sudah disimpan dalam array data

Class KTP ini memiliki akses modifier private, sehingga anggota data dan fungsi yang ada di dalam class ini hanya dapat diakses oleh class itu sendiri. Class KTP ini dapat digunakan untuk merepresentasikan kumpulan data KTP dan menyimpannya dalam sebuah array.

```

51 public:
52     bool validateProvince(std::string province) {
53         return province.length() <= 50;
54     }
55
56     bool validateCity(std::string city) {
57         return city.length() <= 50;
58     }
59
60     bool validateNik(std::string nik) {
61         return nik.length() == 16;
62     }
63     bool validateName(std::string name) {
64         return name.length() <= 50;
65     }
66     bool validateBornPlace(std::string bornPlace) {
67         return bornPlace.length() <= 20;
68     }
69
70     bool validateDate(int date) {
71         return date >= 1 && date <= 31;
72     }
73
74     bool validateMonth(int month) {
75         return month >= 1 && month <= 12;
76     }
77
78     bool validateYear(int year) {
79         return year >= 1900 && year <= 2020;
80     }
81
82     bool validateGender(int gender) {
83         return gender == 0 || gender == 1;
84     }
85
86     bool validateBloodType(std::string bloodType) {
87         return bloodType == "A" || bloodType == "B" || bloodType == "AB" || bloodType == "O";
88     }
89     bool validateAddress(std::string address) {
90         return address.length() <= 100;
91     }
92
93     bool validateNeighborhoodAssociation(int neighborhoodAssociation) {
94         return neighborhoodAssociation >= 0 && neighborhoodAssociation <= 999;
95     }
96
97     bool validateCommunityAssociation(int communityAssociation) {
98         return communityAssociation >= 0 && communityAssociation <= 999;
99     }
100
101     bool validateSubDistrict(std::string subDistrict) {
102         return subDistrict.length() <= 20;
103     }
104
105     bool validateDistrict(std::string district) {
106         return district.length() <= 20;
107     }
108
109     bool validateReligion(std::string religion) {
110         return religion == "Islam" || religion == "Kristen" || religion == "Katolik" || religion == "Hindu" || religion == "Buddha" || religion == "Konghucu";
111     }
112
113     bool validateIsMarried(std::string isMarried) {
114         return isMarried == "Kawin" || isMarried == "Belum Kawin";
115     }
116     bool validateJob(std::string job) {
117         return job.length() <= 30;
118     }
119
120

```

GitHub Copilot: Kode tersebut merupakan sebuah bagian dari class `KTP` yang berisi beberapa fungsi validasi untuk memeriksa apakah data yang dimasukkan sesuai dengan format yang diinginkan. Fungsi-fungsi validasi tersebut adalah:

- `validateProvince`: memeriksa apakah panjang string `province` kurang dari atau sama dengan 50 karakter.
- `validateCity`: memeriksa apakah panjang string `city` kurang dari atau sama dengan 50 karakter.
- `validateNik`: memeriksa apakah panjang string `nik` sama dengan 16 karakter.
- `validateName`: memeriksa apakah panjang string `name` kurang dari atau sama dengan 50 karakter.
- `validateBornPlace`: memeriksa apakah panjang string `bornPlace` kurang dari atau sama dengan 20 karakter.
- `validateDate`: memeriksa apakah nilai `date` berada di antara 1 dan 31.
- `validateMonth`: memeriksa apakah nilai `month` berada di antara 1 dan 12.
- `validateYear`: memeriksa apakah nilai `year` berada di antara 1900 dan 2020.
- `validateGender`: memeriksa apakah nilai `gender` sama dengan 0 atau 1.
- `validateBloodType`: memeriksa apakah string `bloodType` sama dengan "A", "B", "AB", atau "O".
- `validateAddress`: memeriksa apakah panjang string `address` kurang dari atau sama dengan 100 karakter.
- `validateNeighborhoodAssociation`: memeriksa apakah nilai `neighborhoodAssociation` berada di antara 0 dan 999.
- `validateCommunityAssociation`: memeriksa apakah nilai `communityAssociation` berada di antara 0 dan 999.
- `validateSubDistrict`: memeriksa apakah panjang string `subDistrict` kurang dari atau sama dengan 20 karakter.
- `validateDistrict`: memeriksa apakah panjang string `district` kurang dari atau sama dengan 20 karakter.
- `validateReligion`: memeriksa apakah string `religion` sama dengan "Islam", "Kristen", "Katolik", "Hindu", "Buddha", atau "Konghucu".
- `validateIsMarried`: memeriksa apakah string `isMarried` sama dengan "Kawin" atau "Belum Kawin".

- `validateJob`: memeriksa apakah panjang string `job` kurang dari atau sama dengan 30 karakter.

Fungsi-fungsi validasi ini dapat digunakan untuk memeriksa apakah data yang dimasukkan oleh pengguna sesuai dengan format yang diinginkan.

```

122 void input() {
123     do {
124         std::cout << "Provinsi: ";
125         std::getline(std::cin, data[count].province);
126         if (!validateProvince(data[count].province)) {
127             errorInput();
128         }
129     } while (!validateProvince(data[count].province));
130     do {
131         std::cout << "Kota: ";
132         std::getline(std::cin, data[count].city);
133         if (!validateCity(data[count].city)) {
134             errorInput();
135         }
136     } while (!validateCity(data[count].city));
137     do {
138         std::cout << "NIK: ";
139         std::getline(std::cin, data[count].nik);
140         if (!validateNIK(data[count].nik)) {
141             errorInput();
142         }
143     } while (!validateNIK(data[count].nik));
144     do {
145         std::cout << "Nama: ";
146         std::getline(std::cin, data[count].name);
147         if (!validateName(data[count].name)) {
148             errorInput();
149         }
150     } while (!validateName(data[count].name));
151     do {
152         std::cout << "Tempat lahir: ";
153         std::getline(std::cin, data[count].born_place);
154         if (!validateBornPlace(data[count].born_place)) {
155             errorInput();
156         }
157     } while (!validateBornPlace(data[count].born_place));
158     do {
159         std::cout << "Tanggal lahir: ";
160         std::cin >> data[count].born_date;
161         if (!validateDate(data[count].born_date)) {
162             errorInput();
163         }
164     } while (!validateDate(data[count].born_date));
165     do {
166         std::cout << "Bulan lahir: ";
167         std::cin >> data[count].born_month;
168         if (!validateMonth(data[count].born_month)) {
169             errorInput();
170         }
171     } while (!validateMonth(data[count].born_month));
172     do {
173         std::cout << "Tahun lahir: ";
174         std::cin >> data[count].born_year;
175         if (!validateYear(data[count].born_year)) {
176             errorInput();
177         }
178     } while (!validateYear(data[count].born_year));
179     // Jenis kelamin
180     do {
181         std::cout << "Jenis kelamin (0=laki-laki, 1=perempuan): ";
182         std::cin >> data[count].gender;
183         if (!validateGender(data[count].gender)) {
184             errorInput();
185         }
186     } while (!validateGender(data[count].gender));
187     std::cin.ignore();
188     do {
189         std::cout << "Golongan darah: ";
190         std::getline(std::cin, data[count].bloodtype);
191         if (!validateBloodType(data[count].bloodtype)) {
192             errorInput();
193         }
194     } while (!validateBloodType(data[count].bloodtype));
195     do {
196         std::cout << "Alamat: ";
197         std::getline(std::cin, data[count].address.address);
198         if (!validateAddress(data[count].address.address)) {
199             errorInput();
200         }
201     } while (!validateAddress(data[count].address.address));
202     do {
203         std::cout << "RT: ";
204         std::cin >> data[count].address.neighborhoodAssociation;
205         if (!validateRT(data[count].address.neighborhoodAssociation)) {
206             errorInput();
207         }
208     } while (!validateRT(data[count].address.neighborhoodAssociation));
209     do {
210         std::cout << "RW: ";
211         std::cin >> data[count].address.communityAssociation;
212         if (!validateRW(data[count].address.communityAssociation)) {
213             errorInput();
214         }
215     } while (!validateRW(data[count].address.communityAssociation));
216     std::cin.ignore();
217     do {
218         std::cout << "Kecamatan: ";
219         std::getline(std::cin, data[count].address.subDistrict);
220         if (!validateSubDistrict(data[count].address.subDistrict)) {
221             errorInput();
222         }
223     } while (!validateSubDistrict(data[count].address.subDistrict));
224     do {
225         std::cout << "Kabupaten: ";
226         std::getline(std::cin, data[count].address.district);
227         if (!validateDistrict(data[count].address.district)) {
228             errorInput();
229         }
230     } while (!validateDistrict(data[count].address.district));
231     do {
232         std::cout << "Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konghucu): ";
233         std::getline(std::cin, data[count].religion);
234         if (!validateReligion(data[count].religion)) {
235             errorInput();
236         }
237     } while (!validateReligion(data[count].religion));
238     do {
239         std::cout << "Status Perkawinan (Kawin, Belum Kawin): ";
240         std::getline(std::cin, data[count].isMarried);
241         if (!validateIsMarried(data[count].isMarried)) {
242             errorInput();
243         }
244     } while (!validateIsMarried(data[count].isMarried));
245     do {
246         std::cout << "Pekerjaan: ";
247         std::getline(std::cin, data[count].job);
248         if (!validateJob(data[count].job)) {
249             errorInput();
250         }
251     } while (!validateJob(data[count].job));
252     data[count].citizenship = "WNI";
253     data[count].validEmail = "statis@icloud.com";
254     time_t now = time(0);
255     tm *ltm = localtime(now);
256     data[count].created_date = 10000000;
257     data[count].created_month = 10000000;
258     data[count].created_year = 10000000;
259     for (int i = 0; i < data[count].religion.length(); i++) {
260         data[count].religion[i] = tolower(data[count].religion[i]);
261     }
262     for (int i = 0; i < data[count].isMarried.length(); i++) {
263         data[count].isMarried[i] = tolower(data[count].isMarried[i]);
264     }
265 }

```

```

208 std::ofstream file;
209 file.open("data.txt", std::ios::app);
210 file << std::endl;
211 file << "Provinsi: ";
212 file << data[count].province << std::endl;
213 file << "Kota: ";
214 file << data[count].city << std::endl;
215 file << "Kab: ";
216 file << data[count].kab << std::endl;
217 file << "Nama: ";
218 file << data[count].name << std::endl;
219 file << "Tempat/Tgl Lahir: ";
220 file << data[count].born_place << " " << std::setfill('0') << std::setw(2) << data[count].born_date << " " << std::setfill('0') << std::setw(2) << data[count].born_month << " " << data[count].born_year << std::endl;
221 file << "Jenis Kelamin: ";
222 file << (data[count].gender == 0 ? "laki-laki" : "perempuan") << std::endl;
223 file << "Gol Darah: ";
224 file << data[count].bloodtype << std::endl;
225 file << "Alamat: ";
226 file << data[count].address.address << std::endl;
227 file << "RT/RW: ";
228 file << std::setfill('0') << std::setw(3) << data[count].address.neighborhoodAssociation << "/" << std::setfill('0') << std::setw(3) << data[count].address.communityAssociation << std::endl;
229 file << "Agama: ";
230 file << data[count].religion << std::endl;
231 file << "Status Perkawinan: ";
232 file << data[count].married << std::endl;
233 file << "Pekerjaan: ";
234 file << data[count].job << std::endl;
235 file << "Program: ";
236 file << data[count].stipendship << std::endl;
237 file << "Merkala Higgs: ";
238 file << data[count].validuntil << std::endl;
239 file << "Tgl Buat: ";
240 file << std::setfill('0') << std::setw(2) << data[count].created_date << " " << std::setfill('0') << std::setw(2) << data[count].created_month << " " << data[count].created_year << std::endl;
241 file << "-----" << std::endl;
242 count++;
243 }

```

Kode tersebut merupakan sebuah fungsi bernama input yang tidak memiliki parameter dan tidak mengembalikan nilai apapun (void). Fungsi ini bertujuan untuk meminta input dari pengguna untuk mengisi data KTP dan menyimpannya dalam array data. Setiap input yang dimasukkan akan dicek validitasnya menggunakan fungsi-fungsi validasi yang telah didefinisikan sebelumnya. Jika input tidak valid, maka akan dipanggil fungsi errorInput untuk menampilkan pesan error. Setelah semua data KTP diisi, data tersebut akan disimpan dalam file "data.txt" menggunakan std::ofstream. Fungsi ini juga akan menambahkan 1 ke variabel count setiap kali data KTP baru berhasil disimpan.

```

244 void output() {
245     std::cout << "-----" << std::endl;
246     std::cout << "KARTU TANPA NOMOR" << std::endl;
247     std::cout << "-----" << std::endl;
248     for (int i = 0; i < count; i++) {
249         int provincelength = data[i].province.length();
250         int citylength = data[i].city.length();
251         int maxlength = provincelength + citylength + provincelength + citylength;
252         for (int j = 0; j < (maxlength - provincelength) / 2; j++) {
253             std::cout << " ";
254         }
255         std::cout << " " << data[i].province << std::endl;
256         for (int j = 0; j < (maxlength - citylength) / 2; j++) {
257             std::cout << " ";
258         }
259         std::cout << " " << data[i].city << std::endl;
260         std::cout << " " << data[i].kab << std::endl;
261         std::cout << " " << data[i].name << std::endl;
262         std::cout << "Tempat/Tgl Lahir: ";
263         file << data[i].born_place << " " << std::setfill('0') << std::setw(2) << data[i].born_date << " " << std::setfill('0') << std::setw(2) << data[i].born_month << " " << data[i].born_year << std::endl;
264         std::cout << "Jenis Kelamin: ";
265         file << (data[i].gender == 0 ? "laki-laki" : "perempuan") << std::endl;
266         std::cout << "Gol Darah: ";
267         file << data[i].bloodtype << std::endl;
268         std::cout << "Alamat: ";
269         file << data[i].address.address << std::endl;
270         std::cout << "RT/RW: ";
271         file << std::setfill('0') << std::setw(3) << data[i].address.neighborhoodAssociation << "/" << std::setfill('0') << std::setw(3) << data[i].address.communityAssociation << std::endl;
272         std::cout << "Agama: ";
273         file << data[i].religion << std::endl;
274         std::cout << "Status Perkawinan: ";
275         file << data[i].married << std::endl;
276         std::cout << "Pekerjaan: ";
277         file << data[i].job << std::endl;
278         std::cout << "Program: ";
279         file << data[i].stipendship << std::endl;
280         std::cout << "Merkala Higgs: ";
281         file << data[i].validuntil << std::endl;
282         std::cout << "Tgl Buat: ";
283         file << std::setfill('0') << std::setw(2) << data[i].created_date << " " << std::setfill('0') << std::setw(2) << data[i].created_month << " " << data[i].created_year << std::endl;
284         std::cout << "-----" << std::endl;
285     }
286 }

```

Kode tersebut merupakan sebuah fungsi bernama output yang tidak memiliki parameter dan tidak mengembalikan nilai apapun (void). Fungsi ini bertujuan untuk menampilkan data KTP yang telah disimpan dalam array data ke dalam output stream menggunakan std::cout. Setiap data KTP akan ditampilkan dalam format yang sudah ditentukan, yaitu sesuai dengan format KTP yang sebenarnya. Fungsi ini menggunakan beberapa manipulator seperti std::setfill, std::setw, dan std::to_string untuk memformat output yang dihasilkan. Setelah semua data KTP ditampilkan, fungsi ini akan menampilkan garis pemisah menggunakan std::cout.

```

331 void initialize(){
332     std::ifstream file;
333     file.open("data.txt");
334
335     std::string line;
336     count = 0;
337     while (std::getline(file, line)) {
338         if (line.find("NIK") != std::string::npos) {
339             count++;
340         }
341     }
342
343     std::cout << "Banyak KTP: " << count << std::endl;
344     file.clear();file.seekg(0, std::ios::beg);
345
346     int i = 0;
347     while(std::getline(file,line)){
348         if (line.find("Provinsi") != std::string::npos) data[i].province = line.substr(line.find(":") + 2);
349         else if (line.find("Kota/Kabupaten") != std::string::npos) data[i].city = line.substr(line.find(":") + 2);
350         else if (line.find("NIK") != std::string::npos){
351             data[i].nik = line.substr(line.find(":") + 2);
352         }
353         else if (line.find("Nama") != std::string::npos){
354             data[i].nik = line.substr(line.find(":") + 2);
355         }
356         else if (line.find("Tempat/Tgl Lahir") != std::string::npos) {
357             data[i].born.place = line.substr(line.find(":") + 2, line.find(",") - line.find(":") - 2);
358             data[i].born.date = std::stoi(line.substr(line.find(",") + 2, 2));
359             data[i].born.month = std::stoi(line.substr(line.find("-") + 1, 2));
360             data[i].born.year = std::stoi(line.substr(line.find_last_of("-") + 1));
361         }
362         else if (line.find("Jenis Kelamin") != std::string::npos){
363             data[i].gender = line.substr(line.find(":") + 2) == "LAKI-LAKI" ? 0 : 1;
364         }
365         else if (line.find("Gol.Darah") != std::string::npos) data[i].bloodType = line.substr(line.find(":") + 2);
366         else if (line.find("Alamat") != std::string::npos) data[i].address.address= line.substr(line.find(":") + 2);
367         else if (line.find("RT/RW") != std::string::npos) {
368             data[i].address.neighborhoodAssociation = std::stoi(line.substr(line.find(":") + 2, 3));
369             data[i].address.communityAssociation = std::stoi(line.substr(line.find_last_of("/") + 1));
370         }
371         else if (line.find("Kel/Desa") != std::string::npos) data[i].address.subDistrict = line.substr(line.find(":") + 2);
372         else if (line.find("Kecamatan") != std::string::npos) data[i].address.district = line.substr(line.find(":") + 2);
373         else if (line.find("Agama") != std::string::npos) data[i].religion = line.substr(line.find(":") + 2);
374         else if (line.find("Status Perkawinan") != std::string::npos) data[i].isMarried = line.substr(line.find(":") + 2);
375         else if (line.find("Pekerjaan") != std::string::npos) data[i].job = line.substr(line.find(":") + 2);
376         else if (line.find("Kewarganegaraan") != std::string::npos) data[i].citizenship = line.substr(line.find(":") + 2);
377         else if (line.find("Berlaku Hingga") != std::string::npos) data[i].validUntil = line.substr(line.find(":") + 2);
378         else if (line.find("Dibuat") != std::string::npos) {
379             data[i].created.date = std::stoi(line.substr(line.find(":") + 2, 2));
380             data[i].created.month = std::stoi(line.substr(line.find("-") + 1, 2));
381             data[i].created.year = std::stoi(line.substr(line.find_last_of("-") + 1));
382         }
383     }
384     else if (line.find("=====") != std::string::npos) i++;
385 }
386 file.close();
387 }

```

Kode tersebut merupakan sebuah fungsi bernama initialize yang tidak memiliki parameter dan tidak mengembalikan nilai apapun (void). Fungsi ini bertujuan untuk membaca data KTP yang telah disimpan dalam file "data.txt" dan menyimpannya dalam array data. Fungsi ini menggunakan std::ifstream untuk membuka file "data.txt" dan membaca setiap baris dalam file tersebut menggunakan std::getline. Setiap baris yang mengandung informasi KTP akan diproses dan disimpan dalam array data. Setelah semua data KTP dibaca, fungsi ini akan menutup file "data.txt" menggunakan std::ifstream::close.


```
393 int main() {
394     KTP ktp;
395     ktp.initialize();
396     int choice;
397     while(true){
398         std::cout << "Menu: " << std::endl;
399         std::cout << "1. Input Data" << std::endl;
400         std::cout << "2. Tampilkan Data" << std::endl;
401         std::cout << "3. Keluar" << std::endl;
402         std::cout << "Pilihan: ";
403         std::cin >> choice;
404         system("cls");
405         std::cin.ignore();
406         switch (choice) {
407             case 1:
408                 ktp.input();
409                 std::cout << "Data berhasil diinput" << std::endl;
410                 break;
411             case 2:
412                 ktp.output();
413                 break;
414             case 3:
415                 return 0;
416             default:
417                 std::cout << "Pilihan tidak valid" << std::endl;
418                 break;
419         }
420         system("pause");
421         system("cls");
422     }
423 }
```

Kode tersebut merupakan sebuah fungsi main yang merupakan program utama dari aplikasi KTP. Fungsi ini pertama-tama membuat sebuah objek dari class KTP dan memanggil fungsi initialize untuk membaca data KTP yang telah disimpan dalam file "data.txt". Selanjutnya, fungsi ini akan menampilkan menu pilihan yang terdiri dari 3 opsi: input data, tampilkan data, dan keluar. Pengguna dapat memilih opsi yang diinginkan dengan memasukkan angka pilihan melalui std::cin. Setelah pengguna memilih opsi, fungsi ini akan memanggil fungsi yang sesuai dengan pilihan pengguna, yaitu input atau output. Jika pengguna memilih opsi keluar, maka program akan berakhir dengan mengembalikan nilai 0. Fungsi ini juga menggunakan beberapa fungsi dari library cstdlib seperti system("cls") dan

system("pause") untuk membersihkan layar dan menunggu pengguna menekan tombol sebelum melanjutkan program.

Untuk mengakses kodingan dapat melihat link berikut:

[IRedDragonICY/Data-Structure \(github.com\)](https://github.com/IRedDragonICY/Data-Structure)

