

# **LAPORAN PRAKTIKUM**

## **“Pertemuan ke-3: POST TEST – Array Dinamis”**

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 / Rabu 10.30 – 13.30 Lab. Jaringan

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

1. Ubahlah postes pertemuan 2 dengan array dinamis. Tambahkan untuk mengubah maksimal data yang bisa ditampung dalam array KTP itu. Gunakan data teman dari kelompok lain.

Karena KTP adalah privasi, maka tidak etis apabila saya meminta data yang sangat krusial ini, maka saya akan mengambil data KTP yang bersebaran di internet.

Berikut adalah source code baru dari pertemuan yang diubah dari array statis menjadi array dinamis dengan menggunakan konsep pointer.

```

1 #include <iostream>
2 #include <string>
3 #include <cstring>
4 #include <vector>
5 #include <ctime>
6
7 void errorInput() {
8     std::cout << "Input tidak valid" << std::endl;
9 }
10
11 struct Data {
12     std::string province;
13     std::string city;
14     std::string nik;
15     std::string name;
16     struct Born {
17         std::string place;
18         int date;
19         int month;
20         int year;
21     } born;
22     int gender;
23     std::string bloodtype;
24
25     struct Address {
26         std::string address;
27         int neighborhoodAssociation;
28         int communityAssociation;
29         std::string district;
30         std::string subdistrict;
31     } address;
32     std::string religion;
33     std::string license;
34     std::string job;
35     std::string citizenship;
36     std::string validuntil;
37
38     struct Created {
39         int date;
40         int month;
41         int year;
42     } created;
43 };
44
45 class KTP {
46 private:
47     Data *data = nullptr;
48     int count = 0;
49
50 public:
51     bool validateProvince(std::string province) {
52         return province.length() == 34;
53     }
54
55     bool validateCity(std::string city) {
56         return city.length() == 34;
57     }
58
59     bool validateNik(std::string nik) {
60         return nik.length() == 16;
61     }
62
63     bool validateName(std::string name) {
64         return name.length() == 50;
65     }
66
67     bool validateBornPlace(std::string bornPlace) {
68         return bornPlace.length() == 30;
69     }
70
71     bool validateDate(int date) {
72         return date >= 1 && date <= 31;
73     }
74
75     bool validateMonth(int month) {
76         return month >= 1 && month <= 12;
77     }
78
79     bool validateYear(int year) {
80         return year >= 1900 && year <= 2020;
81     }
82
83     bool validateGender(int gender) {
84         return gender == 0 || gender == 1;
85     }
86
87     bool validateBloodtype(std::string bloodtype) {
88         return bloodtype == "A" || bloodtype == "B" || bloodtype == "AB" || bloodtype == "O";
89     }
90
91     bool validateAddress(std::string address) {
92         return address.length() == 100;
93     }
94
95     bool validateNeighborhoodAssociation(int neighborhoodAssociation) {
96         return neighborhoodAssociation >= 0 && neighborhoodAssociation <= 999;
97     }
98
99     bool validateCommunityAssociation(int communityAssociation) {
100        return communityAssociation >= 0 && communityAssociation <= 999;
101    }
102
103    bool validateSubdistrict(std::string subdistrict) {
104        return subdistrict.length() == 20;
105    }
106
107    bool validateDistrict(std::string district) {
108        return district.length() == 20;
109    }
110
111    bool validateReligion(std::string religion) {
112        return religion == "Islam" || religion == "Kristen" || religion == "Katolik" || religion == "Hindu" || religion == "Buddha" || religion == "Tenghara";
113    }
114
115    bool validateLicense(std::string license) {
116        return license == "Kendri" || license == "Belum Kendri";
117    }
118
119    bool validateJob(std::string job) {
120        return job.length() == 30;
121    }
122
123
124

```

### Pertemuan ke-3: POST TEST – Array Dinamis

```

290 void output() {
291     std::cout << "===== " << std::endl;
292     std::cout << "          KARTU TANPA PEMUKUT << std::endl;
293     std::cout << "===== " << std::endl;
294     for (int i = 0; i < count; i++) {
295         int provinsiLength = data[i].province.length();
296         int cityLength = data[i].city.length();
297         int nameLength = provinsiLength + cityLength + provinsiLength + cityLength;
298         for (int j = 0; j < (nameLength - provinsiLength) / 2; j++) {
299             std::cout << " ";
300         }
301         std::cout << "          " << data[i].province << std::endl;
302         for (int j = 0; j < (nameLength - cityLength) / 2; j++) {
303             std::cout << " ";
304         }
305         std::cout << "          " << data[i].city << std::endl;
306         std::cout << "NIK          " << data[i].nik << std::endl;
307         std::cout << "Nama          " << data[i].name << std::endl;
308         std::cout << "tempatTgl lahir " << data[i].born_place << " " << data[i].born_date << " " << data[i].born_month << " " << data[i].born_year << std::endl;
309         std::cout << "Jenis Kelamin " << data[i].gender << " " << data[i].bloodType << " " << data[i].bloodType << std::endl;
310         std::cout << "Gol Darah " << data[i].bloodType << std::endl;
311         std::cout << "Alamat " << data[i].address << std::endl;
312         std::cout << "NoRum " << data[i].address_district << std::endl;
313         std::cout << "Agama " << data[i].religion << std::endl;
314         std::cout << "Pendidikan " << data[i].education << std::endl;
315         std::cout << "Status " << data[i].status << std::endl;
316         std::cout << "Kewarganegaraan " << data[i].citizenship << std::endl;
317         std::cout << "KTP " << data[i].KTP << std::endl;
318         std::cout << "Created date " << data[i].created_date << " " << data[i].created_month << " " << data[i].created_year << std::endl;
319         std::cout << "Created month " << data[i].created_month << " " << data[i].created_year << std::endl;
320         std::cout << "Created year " << data[i].created_year << " " << data[i].created_month << " " << data[i].created_year << std::endl;
321     }
322     std::cout << "===== " << std::endl;
323 }
324
325 void initialize() {
326     std::ifstream file;
327     file.open("data.txt");
328     std::string line;
329     count = 0;
330     while (std::getline(file, line)) {
331         if (line.find("NIK") != std::string::npos) {
332             count++;
333         }
334     }
335     std::cout << "Banyak KTP: " << count << std::endl;
336     file.close();
337     data = new Data[count];
338     int i = 0;
339     while (std::getline(file, line)) {
340         if (line.find("Provinsi") != std::string::npos) data[i].province = line.substr(line.find(" ") + 2);
341         else if (line.find("Kota/Kabupaten") != std::string::npos) data[i].city = line.substr(line.find(" ") + 2);
342         else if (line.find("NIK") != std::string::npos) {
343             data[i].nik = line.substr(line.find(" ") + 2);
344         }
345         else if (line.find("Nama") != std::string::npos) {
346             data[i].name = line.substr(line.find(" ") + 2);
347         }
348         else if (line.find("tempatTgl lahir") != std::string::npos) {
349             data[i].born_place = line.substr(line.find(" ") + 2, line.find(" ") - line.find(" ") - 2);
350             data[i].born_date = line.substr(line.find(" ") + 2, 2);
351             data[i].born_month = line.substr(line.find(" ") + 3, 2);
352             data[i].born_year = line.substr(line.find(" ") + 4, 4);
353         }
354         else if (line.find("Jenis Kelamin") != std::string::npos) {
355             data[i].gender = line.substr(line.find(" ") + 2) < "laki laki" ? 0 : 1;
356         }
357         else if (line.find("Gol Darah") != std::string::npos) data[i].bloodType = line.substr(line.find(" ") + 2);
358         else if (line.find("Alamat") != std::string::npos) data[i].address = line.substr(line.find(" ") + 2);
359         else if (line.find("NoRum") != std::string::npos) {
360             data[i].address_district = line.substr(line.find(" ") + 2, 4);
361             data[i].address_communityAssociation = line.substr(line.find(" ") + 5, 11);
362         }
363         else if (line.find("Agama") != std::string::npos) data[i].religion = line.substr(line.find(" ") + 2);
364         else if (line.find("Pendidikan") != std::string::npos) data[i].education = line.substr(line.find(" ") + 2);
365         else if (line.find("Status") != std::string::npos) data[i].status = line.substr(line.find(" ") + 2);
366         else if (line.find("Kewarganegaraan") != std::string::npos) data[i].citizenship = line.substr(line.find(" ") + 2);
367         else if (line.find("KTP") != std::string::npos) {
368             data[i].KTP = line.substr(line.find(" ") + 2, 2);
369             data[i].created_date = line.substr(line.find(" ") + 3, 2);
370             data[i].created_month = line.substr(line.find(" ") + 4, 2);
371             data[i].created_year = line.substr(line.find(" ") + 5, 4);
372         }
373         else if (line.find("=====") != std::string::npos) i++;
374     }
375     file.close();
376 }
377
378 int main() {
379     int ktp;
380     int choice;
381     while (true) {
382         ktp = initialize();
383         std::cout << "Menu: " << std::endl;
384         std::cout << "1. Input Data" << std::endl;
385         std::cout << "2. Tampilkan Data" << std::endl;
386         std::cout << "3. Keluar" << std::endl;
387         std::cout << "Pilihan: ";
388         std::cin >> choice;
389         system("cls");
390         std::cin.ignore();
391         switch (choice) {
392             case 1:
393                 ktp.input();
394                 std::cout << "Data berhasil ditinput" << std::endl;
395                 break;
396             case 2:
397                 ktp.output();
398                 break;
399             case 3:
400                 return EXIT_SUCCESS;
401             default:
402                 std::cout << "Pilihan tidak valid" << std::endl;
403                 break;
404         }
405         system("pause");
406         system("cls");
407     }
408 }

```

## Program jalan (Screenshot input output)

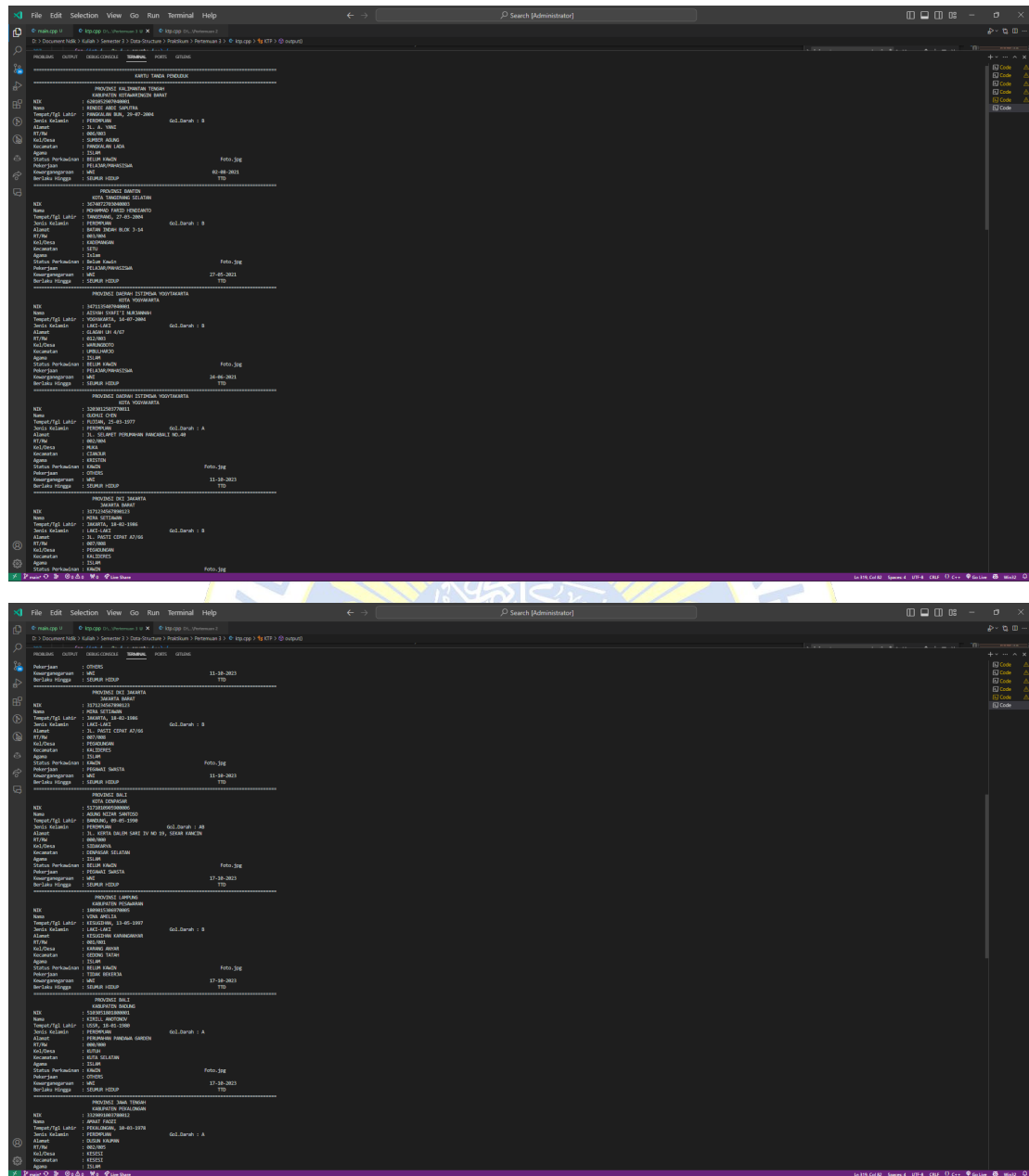
Program awal dijalankan

```

Banyak KTP: 11
Menu:
1. Input Data
2. Tampilkan Data
3. Keluar
Pilihan: 1

```

### Melihat menu 2. Tampilkan Data



```

File Edit Selection View Go Run Terminal Help
D:\> Document Ndlk > Kuliah > Semester 3 > Data-Structure > Praktikum > Pertemuan 3 > ktp.cpp > KTP > output()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

Jenis Kelamin : LAKI-LAKI Gol.Darah : B
Alamat : KESUGIHAN KARANGANYAR
RT/Rw : 001/001
Kel/Desa : KARANG ANYAR
Kecamatan : GEDONG TATAH
Agama : ISLAM
Status Perkawinan : BELUM KAWIN Foto.jpg
Pekerjaan : TIDAK BEKERJA
Kewarganegaraan : WNI 17-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

=====
PROVINSI BALI
KABUPATEN BADUNG
NIK : 5103051801800001
Nama : KIRILL ANOTONOV
Tempat/Tgl Lahir : USSR, 18-01-1980
Jenis Kelamin : PEREMPUAN Gol.Darah : A
Alamat : PERUMAHAN PANDAWA GARDEN
RT/Rw : 000/000
Kel/Desa : KUTUH
Kecamatan : KUTA SELATAN
Agama : ISLAM
Status Perkawinan : KAWIN Foto.jpg
Pekerjaan : OTHERS
Kewarganegaraan : WNI 17-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

=====
PROVINSI JAWA TENGAH
KABUPATEN PEKALONGAN
NIK : 3329091003780012
Nama : AMAAT FAOZI
Tempat/Tgl Lahir : PEKALONGAN, 10-03-1978
Jenis Kelamin : PEREMPUAN Gol.Darah : A
Alamat : DUSUN KAUMAN
RT/Rw : 002/005
Kel/Desa : KESESI
Kecamatan : KESESI
Agama : ISLAM
Status Perkawinan : KAWIN Foto.jpg
Pekerjaan : Pedagang
Kewarganegaraan : WNI 17-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

=====
PROVINSI JAWA TENGAH
KABUPATEN KUDUS
NIK : 3319042612640001
Nama : BUDI SANTOSO
Tempat/Tgl Lahir : KUDUS, 29-12-1984
Jenis Kelamin : PEREMPUAN Gol.Darah : B
Alamat : LARIREJO
RT/Rw : 002/001
Kel/Desa : LARIREJO
Kecamatan : UNDAAN
Agama : ISLAM
Status Perkawinan : BELUM KAWIN Foto.jpg
Pekerjaan : PETANI/PEKEBUN
Kewarganegaraan : WNI 17-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

=====
PROVINSI JAWA TENGAH
KOTA SURAKARTA
NIK : 3372011210770001
Nama : SELAMAT IDUL ADHA
Tempat/Tgl Lahir : SURAKARTA, 12-10-1977
Jenis Kelamin : PEREMPUAN Gol.Darah : AB
Alamat : BARON CILIK
RT/Rw : 004/005
Kel/Desa : BUMI
Kecamatan : LAWEYAN
Agama : ISLAM
Status Perkawinan : KAWIN Foto.jpg
Pekerjaan : PERDAGANGAN
Kewarganegaraan : WNI 17-10-2023
Berlaku Hingga : SELUMUR HIDUP TTD

=====
Press any key to continue . . .
  
```

## Mencoba input data KTP dengan program array dinamis

```

D:\> Document Ndik > Kuliah > Semester 3 > Data-Structure > Praktikum > Pertemuan 3 > ktp.cpp > KTP > output()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

Provinsi: PROVINSI SULAWESI SELATAN
Kota: KABUPATEN SOPPENG
NIK: 7312042510720002
Nama: ABDURRAUF, S.Pd, M.Pd
Tempat Lahir: CELLENGENGE
Tanggal Lahir: 25
Bulan Lahir: 10
Tahun Lahir: 1972
Jenis kelamin (0=laki-laki, 1=perempuan): 0
Golongan Darah: 0
Alamat: JL. MERDEKA NO.43
RT: 1
RW: 4
Kel/Desa: BILA
Kecamatan: LALABATA
Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konghucu): Islam
Status Perkawinan (Kawin, Belum Kawin): KAWIN
Input tidak valid
Status Perkawinan (Kawin, Belum Kawin): PEGAWAI NEGERI SIPIL
Input tidak valid
Status Perkawinan (Kawin, Belum Kawin): Kawin
Pekerjaan: PEGAWAI NEGERI SIPIL
Data berhasil diinput
Press any key to continue . . .
  
```

## Melihat ktp lagi pada menu ke dua

```

=====
PROVINSI SULAWESI SELATAN
KABUPATEN SOPPENG
NIK      : 7312042510720002
Nama     : ABDURRAUF, S.Pd, M.Pd
Tempat/Tgl Lahir : CELLENGENGE, 25-10-1972
Jenis Kelamin : PEREMPUAN
Gol.Darah : 0
Alamat  : JL. MERDEKA NO.43
RT/RW   : 001/004
Kel/Desa : BILA
Kecamatan : LALABATA
Agama   : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : PEGAWAI NEGERI SIPIL
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
=====
Press any key to continue . . .
  
```

## Input data dari file

data.txt sebagai database, oleh karena itu pastinya sudah ada input data dari file

data.txt sebelum diinputkan ktp baru

```

data.txt
File Edit View
Provinsi      : PROVINSI BALI
Kota/Kabupaten : KABUPATEN BADUNG
NIK           : 5103051801800001
Nama          : KIRILL ANOTONOV
Tempat/Tgl Lahir : USSR, 18-01-1980
Jenis Kelamin  : LAKI-LAKI
Gol.Darah     : A
Alamat        : PERUMAHAN PANDAWA GARDEN
               RT/RW      : 000/000
               Kel/Desa   : KUTUH
               Kecamatan  : KUTA SELATAN
Agama         : ISLAM
Status Perkawinan : KAWIN
Pekerjaan     : OTHERS
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat        : 17-10-2023
=====
Provinsi      : PROVINSI JAWA TENGAH
Kota/Kabupaten : KABUPATEN PEKALONGAN
NIK           : 3329091003780012
Nama          : AMAAT FAOZI
Tempat/Tgl Lahir : PEKALONGAN, 10-03-1978
Jenis Kelamin  : LAKI-LAKI
Gol.Darah     : A
Alamat        : DUSUN KAUMAN
               RT/RW      : 002/005
               Kel/Desa   : KESESI
               Kecamatan  : KESESI
Agama         : ISLAM
Status Perkawinan : KAWIN
Pekerjaan     : Pedagang
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat        : 17-10-2023
=====
Provinsi      : PROVINSI JAWA TENGAH
Kota/Kabupaten : KABUPATEN KUDUS
NIK           : 3319042612640001
Nama          : BUDI SANTOSO
Tempat/Tgl Lahir : KUDUS, 29-12-1984
Jenis Kelamin  : LAKI-LAKI
Gol.Darah     : B
Alamat        : LARIREJO
               RT/RW      : 002/001
               Kel/Desa   : LARIREJO
               Kecamatan  : UNDAAN
Agama         : ISLAM
Status Perkawinan : BELUM KAWIN
Pekerjaan     : PETANI/PEKEBUN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat        : 17-10-2023
=====
Ln 146 Col 41

```



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

---

Provinsi : PROVINSI DAERAH ISTIMEWA YOGYAKARTA

Kota/Kabupaten : KOTA YOGYAKARTA

NIK : 3471135407040001

Nama : AISYAH SYAF'I NURJANNAH

Tempat/Tgl Lahir : YOGYAKARTA, 14-07-2004

Jenis Kelamin : PEREMPUAN

Gol.Darah : B

Alamat : GLAGAH UH 4/67

RT/RW : 012/003

Kel/Desa : WARUNGBOTO

Kecamatan : UMBULHARJO

Agama : ISLAM

Status Perkawinan : BELUM KAWIN

Pekerjaan : PELAJAR/MAHASISWA

Kewarganegaraan : WNI

Berlaku Hingga : SEUMUR HIDUP

Dibuat : 24-06-2021

---

Provinsi : PROVINSI DAERAH ISTIMEWA YOGYAKARTA  
Kota/Kabupaten : KOTA YOGYAKARTA  
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/Kabupaten : 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

---

Provinsi : PROVINSI BALI

Kota/Kabupaten : KOTA DENPASAR

NIK : 5171010905900006

Nama : AGUNG NIZAR SANTOSO

Tempat/Tgl Lahir : BANDUNG, 09-05-1990

Jenis Kelamin : LAKI-LAKI

Gol.Darah : AB

Alamat : JL. KERTA DALEM SARI IV NO 19, SEKAR KANCIN

RT/RW : 000/000

Kel/Desa : SIDA KARYA

Kecamatan : DENPASAR SELATAN

Agama : ISLAM

Status Perkawinan : BELUM KAWIN

Pekerjaan : PEGAWAI SWASTA

Kewarganegaraan : WNI

Berlaku Hingga : SEUMUR HIDUP

Dibuat : 17-10-2023

---

Provinsi : PROVINSI LAMPUNG  
Kota/Kabupaten : KABUPATEN PESAWARAN  
NIK : 1809015306970005  
Nama : VINA AMELIA  
Tempat/Tgl Lahir : KESUGIHAN, 13-05-1997  
Jenis Kelamin : PEREMPUAN  
Gol.Darah : B  
Alamat : KESUGIHAN KARANGANYAR  
RT/RW : 001/001  
Kel/Desa : KARANG ANYAR  
Kecamatan : GEDONG TATAH  
Agama : ISLAM  
Status Perkawinan : BELUM KAWIN  
Pekerjaan : TIDAK BEKERJA  
Kewarganegaraan : WNI  
Berlaku Hingga : SEUMUR HIDUP  
Dibuat : 17-10-2023

---

Provinsi : PROVINSI BALI  
Kota/Kabupaten : KABUPATEN BADUNG<sub>60</sub>  
NIK : 5103051801800001  
Nama : KIRILL ANOTONOV  
Tempat/Tgl Lahir : USSR, 18-01-1980  
Jenis Kelamin : LAKI-LAKI  
Gol.Darah : A  
Alamat : PERUMAHAN PANDAWA GARDEN  
RT/RW : 000/000

Kel/Desa : KUTUH

Kecamatan : KUTA SELATAN

Agama : ISLAM

Status Perkawinan : KAWIN

Pekerjaan : OTHERS

Kewarganegaraan : WNI

Berlaku Hingga : SEUMUR HIDUP

Dibuat : 17-10-2023

---

Provinsi : PROVINSI JAWA TENGAH

Kota/Kabupaten : KABUPATEN PEKALONGAN

NIK : 3329091003780012

Nama : AMAAT FAOZI

Tempat/Tgl Lahir : PEKALONGAN, 10-03-1978

Jenis Kelamin : LAKI-LAKI

Gol.Darah : A

Alamat : DUSUN KAUMAN

RT/RW : 002/005

Kel/Desa : KESESI

Kecamatan : KESESI

Agama : ISLAM

Status Perkawinan : KAWIN

Pekerjaan : Pedagang

Kewarganegaraan : WNI

Berlaku Hingga : SEUMUR HIDUP

Dibuat : 17-10-2023

---



Provinsi : PROVINSI JAWA TENGAH

Kota/Kabupaten : KABUPATEN KUDUS

NIK : 3319042612640001

Nama : BUDI SANTOSO

Tempat/Tgl Lahir : KUDUS, 29-12-1984

Jenis Kelamin : LAKI-LAKI

Gol.Darah : B

Alamat : LARIREJO

RT/RW : 002/001

Kel/Desa : LARIREJO

Kecamatan : UNDAAN

Agama : ISLAM

Status Perkawinan : BELUM KAWIN

Pekerjaan : PETANI/PEKEBUN

Kewarganegaraan : WNI

Berlaku Hingga : SEUMUR HIDUP

Dibuat : 17-10-2023

---

Setelah mengisi 1 KTP lagi, mengupdate di data.txt

```

data.txt
File Edit View
Provinsi : PROVINSI JAWA TENGAH
Kota/Kabupaten : KABUPATEN KUDUS
NIK : 3319042612640001
Nama : BUDI SANTOSO
Tempat/Tgl Lahir : KUDUS, 29-12-1984
Jenis Kelamin : LAKI-LAKI
Gol.Darah : B
Alamat : LARIREJO
RT/RW : 002/001
Kel/Desa : LARIREJO
Kecamatan : UNDAAN
Agama : ISLAM
Status Perkawinan : BELUM KAWIN
Pekerjaan : PETANI/PEKEBUN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat : 17-10-2023
=====
Provinsi : PROVINSI JAWA TENGAH
Kota/Kabupaten : KOTA SURAKARTA
NIK : 3372011210770001
Nama : SELAMAT IDUL ADHA
Tempat/Tgl Lahir : SURAKARTA, 12-10-1977
Jenis Kelamin : LAKI-LAKI
Gol.Darah : AB
Alamat : BARON CILIK
RT/RW : 004/005
Kel/Desa : BUMI
Kecamatan : LAWEYAN
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : PERDAGANGAN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat : 17-10-2023
=====
Provinsi : PROVINSI SULAWESI SELATAN
Kota/Kabupaten : KABUPATEN SOPPENG
NIK : 7312042510720002
Nama : ABDURRAUF, S.Pd, M.Pd
Tempat/Tgl Lahir : CELLENGENGE, 25-10-1972
Jenis Kelamin : LAKI-LAKI
Gol.Darah : O
Alamat : JL. MERDEKA NO.43
RT/RW : 001/004
Kel/Desa : BILA
Kecamatan : LALABATA
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : PEGAWAI NEGERI SIPIL
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP
Dibuat : 17-10-2023
=====
Ln 8, Col 32

```



## Struct

Karena kodingan melanjutkan pertemuan ke-2, pada sebelumnya sudah menggunakan struct yaitu pada inisialisasi data KTP.

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

## Class

Menggunakan class KTP dan inialisasikan dari KTP menjadi ktp.

```

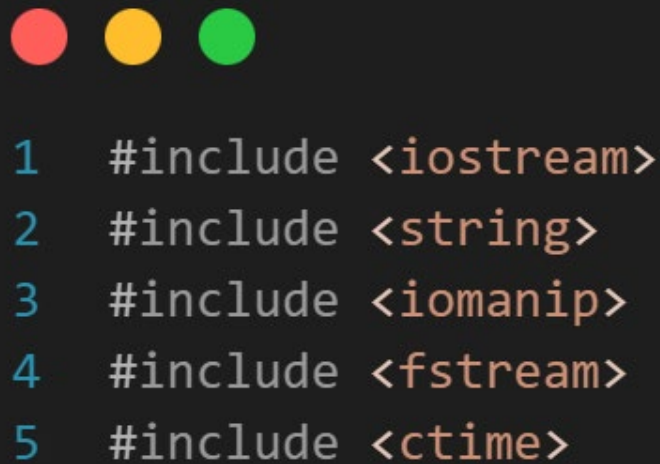
1 class KTP {
2 private:
3     data *data = nullptr;
4     int count = 0;
5 public:
6     bool validateProvince(string province) {
7         return province.length() <= 50;
8     }
9     bool validateCity(string city) {
10        return city.length() <= 50;
11    }
12    bool validateDistrict(string dist) {
13        return dist.length() <= 20;
14    }
15    bool validateSuburb(string sub) {
16        return sub.length() <= 50;
17    }
18    bool validateHouse(string name) {
19        return name.length() <= 50;
20    }
21    bool validateRoadName(string roadName) {
22        return roadName.length() <= 20;
23    }
24    bool validateDate(int date) {
25        return date >= 1 && date <= 31;
26    }
27    bool validateMonth(int month) {
28        return month >= 1 && month <= 12;
29    }
30    bool validateYear(int year) {
31        return year >= 1000 && year <= 2020;
32    }
33    bool validateGender(int gender) {
34        return gender == 0 || gender == 1;
35    }
36    bool validateBloodType(string bloodType) {
37        return bloodType == "A" || bloodType == "B" || bloodType == "AB" || bloodType == "O";
38    }
39    bool validateReferral(string ref) {
40        return ref.length() <= 100;
41    }
42    bool validateNeighborhoodAssociation(string neighborhoodAssociation) {
43        return neighborhoodAssociation >= 0 && neighborhoodAssociation <= 999;
44    }
45    bool validateCommunityAssociation(string communityAssociation) {
46        return communityAssociation >= 0 && communityAssociation <= 999;
47    }
48    bool validateDistrict(string district) {
49        return district.length() <= 20;
50    }
51    bool validateReligion(string religion) {
52        return religion == "Islam" || religion == "Kristen" || religion == "Katolik" || religion == "Hindu" || religion == "Buddha" || religion == "Sikhisme";
53    }
54    bool validateReferral(string ref) {
55        return ref == "Klinik" || ref == "Rumah Sakit";
56    }
57    bool validateReferral(string ref) {
58        return ref.length() <= 50;
59    }
60 }
61
62 void Input() {
63     data *newData = new data(count + 1);
64     do {
65         cout << "Masukkan ";
66         int i = 0;
67         while (i < count) {
68             cout << "Provinsi: ";
69             newData[i].province = newData[i].province;
70             if (!validateProvince(newData[i].province)) {
71                 errorInput();
72             }
73             i++;
74         }
75         while (i < count) {
76             cout << "Kota: ";
77             newData[i].city = newData[i].city;
78             if (!validateCity(newData[i].city)) {
79                 errorInput();
80             }
81             i++;
82         }
83         while (i < count) {
84             cout << "Kabupaten: ";
85             newData[i].district = newData[i].district;
86             if (!validateDistrict(newData[i].district)) {
87                 errorInput();
88             }
89             i++;
90         }
91         while (i < count) {
92             cout << "Kecamatan: ";
93             newData[i].suburb = newData[i].suburb;
94             if (!validateSuburb(newData[i].suburb)) {
95                 errorInput();
96             }
97             i++;
98         }
99         while (i < count) {
100            cout << "Nama Jalan: ";
101            newData[i].roadName = newData[i].roadName;
102            if (!validateRoadName(newData[i].roadName)) {
103                errorInput();
104            }
105            i++;
106        }
107        while (i < count) {
108            cout << "Tanggal Lahir: ";
109            newData[i].date = newData[i].date;
110            if (!validateDate(newData[i].date)) {
111                errorInput();
112            }
113            i++;
114        }
115        while (i < count) {
116            cout << "Bulan Lahir: ";
117            newData[i].month = newData[i].month;
118            if (!validateMonth(newData[i].month)) {
119                errorInput();
120            }
121            i++;
122        }
123        while (i < count) {
124            cout << "Tahun Lahir: ";
125            newData[i].year = newData[i].year;
126            if (!validateYear(newData[i].year)) {
127                errorInput();
128            }
129            i++;
130        }
131        while (i < count) {
132            cout << "Jenis Kelamin: (0=Jaki, 1=Perempuan) ";
133            newData[i].gender = newData[i].gender;
134            if (!validateGender(newData[i].gender)) {
135                errorInput();
136            }
137            i++;
138        }
139        while (i < count) {
140            cout << "Golongan Darah: ";
141            newData[i].bloodType = newData[i].bloodType;
142            if (!validateBloodType(newData[i].bloodType)) {
143                errorInput();
144            }
145            i++;
146        }
147        while (i < count) {
148            cout << "Alamat: ";
149            newData[i].address = newData[i].address;
150            if (!validateAddress(newData[i].address)) {
151                errorInput();
152            }
153            i++;
154        }
155        while (i < count) {
156            cout << "RT: ";
157            newData[i].addressNeighborhoodAssociation = newData[i].addressNeighborhoodAssociation;
158            if (!validateNeighborhoodAssociation(newData[i].addressNeighborhoodAssociation)) {
159                errorInput();
160            }
161            i++;
162        }
163        while (i < count) {
164            cout << "RW: ";
165            newData[i].addressCommunityAssociation = newData[i].addressCommunityAssociation;
166            if (!validateCommunityAssociation(newData[i].addressCommunityAssociation)) {
167                errorInput();
168            }
169            i++;
170        }
171        while (i < count) {
172            cout << "Kecamatan: ";
173            newData[i].district = newData[i].district;
174            if (!validateDistrict(newData[i].district)) {
175                errorInput();
176            }
177            i++;
178        }
179        while (i < count) {
180            cout << "Kota: ";
181            newData[i].city = newData[i].city;
182            if (!validateCity(newData[i].city)) {
183                errorInput();
184            }
185            i++;
186        }
187        while (i < count) {
188            cout << "Provinsi: ";
189            newData[i].province = newData[i].province;
190            if (!validateProvince(newData[i].province)) {
191                errorInput();
192            }
193            i++;
194        }
195    } while (i < count);
196 }

```

### Pertemuan ke-3: POST TEST – Array Dinamis

### Penjelasan lanjut

Program di atas menggunakan array dinamis karena pada fungsi input(), setiap kali data baru dimasukkan, array Data \*newData akan dialokasikan dengan ukuran count + 1. Kemudian, data lama pada array data akan dihapus dengan perintah delete[] data dan digantikan dengan data baru pada array newData. Hal ini dilakukan agar ukuran array data dapat berubah-ubah sesuai dengan jumlah data yang dimasukkan oleh pengguna.



```
1  #include <iostream>
2  #include <string>
3  #include <iomanip>
4  #include <fstream>
5  #include <ctime>
```

Kode tersebut merupakan header file yang digunakan untuk menyertakan library yang dibutuhkan dalam program C++.

- `iostream` digunakan untuk input dan output stream.
- `string` digunakan untuk manipulasi string.
- `iomanip` digunakan untuk manipulasi input dan output stream.
- `fstream` digunakan untuk manipulasi file stream.
- `ctime` digunakan untuk manipulasi waktu dan tanggal.



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

Kode tersebut merupakan sebuah fungsi bernama `errorInput()` yang bertujuan untuk menampilkan pesan "Input tidak valid" pada console output. Fungsi ini kemungkinan dipanggil ketika terjadi kesalahan input pada program.



```

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

```

Kode tersebut merupakan definisi dari sebuah struct bernama Data yang memiliki beberapa atribut seperti province, city, nik, name, born, gender, bloodType, address, religion, isMaried, job, citizenship, validUntil, dan created.

Struct Data memiliki beberapa nested struct seperti Born dan Address yang masing-masing memiliki atribut sendiri.

Struct Born memiliki atribut place, date, month, dan year yang merepresentasikan tempat, tanggal, bulan, dan tahun lahir seseorang.

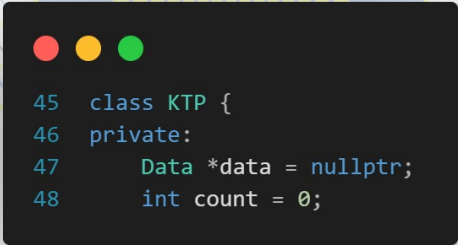
Struct Address memiliki atribut address, neighborhoodAssociation, communityAssociation, district, dan subDistrict yang merepresentasikan alamat, RT, RW, kecamatan, dan kelurahan seseorang.

Atribut gender memiliki tipe data boolean yang merepresentasikan jenis kelamin seseorang.

Atribut bloodType memiliki tipe data string yang merepresentasikan golongan darah seseorang.

Atribut religion, isMaried, job, citizenship, dan validUntil memiliki tipe data string yang merepresentasikan agama, status pernikahan, pekerjaan, kewarganegaraan, dan masa berlaku KTP seseorang.

Struct Created memiliki atribut date, month, dan year yang merepresentasikan tanggal, bulan, dan tahun pembuatan KTP.



```
45 class KTP {
46 private:
47     Data *data = nullptr;
48     int count = 0;
```

Kode tersebut merupakan definisi dari sebuah class bernama KTP yang memiliki dua atribut private yaitu data dan count.

Atribut data memiliki tipe pointer ke struct Data dan diinisialisasi dengan nilai null. Atribut ini kemungkinan digunakan untuk menyimpan data-data KTP yang telah diinputkan.

Atribut count memiliki tipe data integer dan diinisialisasi dengan nilai 0. Atribut ini kemungkinan digunakan untuk menghitung jumlah data KTP yang telah diinputkan.

```

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

```

Kode tersebut merupakan definisi dari beberapa fungsi validasi yang digunakan untuk memeriksa apakah data yang diinputkan sesuai dengan format yang diinginkan.

Fungsi-fungsi validasi tersebut antara lain:

- validateProvince untuk memeriksa apakah nama provinsi yang diinputkan memiliki panjang maksimal 50 karakter.



- validateCity untuk memeriksa apakah nama kota yang diinputkan memiliki panjang maksimal 50 karakter.
- validateNik untuk memeriksa apakah NIK yang diinputkan memiliki panjang tepat 16 karakter.
- validateName untuk memeriksa apakah nama yang diinputkan memiliki panjang maksimal 50 karakter.
- validateBornPlace untuk memeriksa apakah tempat lahir yang diinputkan memiliki panjang maksimal 20 karakter.
- validateDate untuk memeriksa apakah tanggal yang diinputkan berada dalam rentang 1-31.
- validateMonth untuk memeriksa apakah bulan yang diinputkan berada dalam rentang 1-12.
- validateYear untuk memeriksa apakah tahun yang diinputkan berada dalam rentang 1900-2020.
- validateGender untuk memeriksa apakah jenis kelamin yang diinputkan bernilai 0 atau 1.
- validateBloodType untuk memeriksa apakah golongan darah yang diinputkan bernilai "A", "B", "AB", atau "O".
- validateAddress untuk memeriksa apakah alamat yang diinputkan memiliki panjang maksimal 100 karakter.
- validateNeighborhoodAssociation untuk memeriksa apakah RT yang diinputkan berada dalam rentang 0-999.
- validateCommunityAssociation untuk memeriksa apakah RW yang diinputkan berada dalam rentang 0-999.
- validateSubDistrict untuk memeriksa apakah kelurahan yang diinputkan memiliki panjang maksimal 20 karakter.
- validateDistrict untuk memeriksa apakah kecamatan yang diinputkan memiliki panjang maksimal 20 karakter.
- validateReligion untuk memeriksa apakah agama yang diinputkan bernilai "Islam", "Kristen", "Katolik", "Hindu", "Buddha", atau "Konghucu".
- validateIsMarried untuk memeriksa apakah status pernikahan yang diinputkan bernilai "Kawin" atau "Belum Kawin".
- validateJob untuk memeriksa apakah pekerjaan yang diinputkan memiliki panjang maksimal 30 karakter.

```

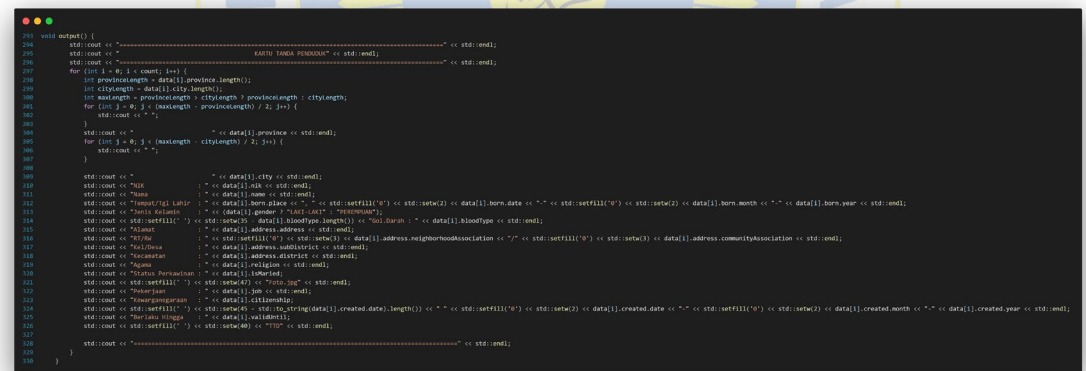
121 void Input() {
122     Data *newData = new Data[count + 1];
123     do {
124         std::cout << "Provinsi: ";
125         std::getline(std::cin, newData[count].province);
126         if (!validateProvince(newData[count].province)) {
127             errorInput();
128         }
129     } while (!validateProvince(newData[count].province));
130     do {
131         std::cout << "Kota: ";
132         std::getline(std::cin, newData[count].city);
133         if (!validateCity(newData[count].city)) {
134             errorInput();
135         }
136     } while (!validateCity(newData[count].city));
137     do {
138         std::cout << "Kec: ";
139         std::getline(std::cin, newData[count].village);
140         if (!validateVillage(newData[count].village)) {
141             errorInput();
142         }
143     } while (!validateVillage(newData[count].village));
144     do {
145         std::cout << "Nama: ";
146         std::getline(std::cin, newData[count].name);
147         if (!validateName(newData[count].name)) {
148             errorInput();
149         }
150     } while (!validateName(newData[count].name));
151     do {
152         std::cout << "Tempat Lahir: ";
153         std::getline(std::cin, newData[count].born_place);
154         if (!validateBornPlace(newData[count].born_place)) {
155             errorInput();
156         }
157     } while (!validateBornPlace(newData[count].born_place));
158     do {
159         std::cout << "Tanggal Lahir: ";
160         std::cin >> newData[count].born_date;
161         if (!validateDate(newData[count].born_date)) {
162             errorInput();
163         }
164     } while (!validateDate(newData[count].born_date));
165     do {
166         std::cout << "Bulan Lahir: ";
167         std::cin >> newData[count].born_month;
168         if (!validateMonth(newData[count].born_month)) {
169             errorInput();
170         }
171     } while (!validateMonth(newData[count].born_month));
172     do {
173         std::cout << "Tahun Lahir: ";
174         std::cin >> newData[count].born_year;
175         if (!validateYear(newData[count].born_year)) {
176             errorInput();
177         }
178     } while (!validateYear(newData[count].born_year));
179     do {
180         std::cout << "Jenis Kelamin (M=Male, F=Female): ";
181         std::cin >> newData[count].gender;
182         if (!validateGender(newData[count].gender)) {
183             errorInput();
184         }
185     } while (!validateGender(newData[count].gender));
186     std::cin.ignore();
187     do {
188         std::cout << "Golongan Darah: ";
189         std::getline(std::cin, newData[count].bloodtype);
190         if (!validateBloodType(newData[count].bloodtype)) {
191             errorInput();
192         }
193     } while (!validateBloodType(newData[count].bloodtype));
194     do {
195         std::cout << "Alamat: ";
196         std::getline(std::cin, newData[count].address);
197         if (!validateAddress(newData[count].address)) {
198             errorInput();
199         }
200     } while (!validateAddress(newData[count].address));
201     do {
202         std::cout << "RT/RW: ";
203         std::cin >> newData[count].address_neighborhoodAssociation;
204         if (!validateNeighborhoodAssociation(newData[count].address_neighborhoodAssociation)) {
205             errorInput();
206         }
207     } while (!validateNeighborhoodAssociation(newData[count].address_neighborhoodAssociation));
208     do {
209         std::cout << "Desa: ";
210         std::cin >> newData[count].address_communityAssociation;
211         if (!validateCommunityAssociation(newData[count].address_communityAssociation)) {
212             errorInput();
213         }
214     } while (!validateCommunityAssociation(newData[count].address_communityAssociation));
215     std::cin.ignore();
216     do {
217         std::cout << "Kecamatan: ";
218         std::getline(std::cin, newData[count].address_subdistrict);
219         if (!validateSubdistrict(newData[count].address_subdistrict)) {
220             errorInput();
221         }
222     } while (!validateSubdistrict(newData[count].address_subdistrict));
223     do {
224         std::cout << "Kabupaten: ";
225         std::getline(std::cin, newData[count].address_district);
226         if (!validateDistrict(newData[count].address_district)) {
227             errorInput();
228         }
229     } while (!validateDistrict(newData[count].address_district));
230     do {
231         std::cout << "Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konghucu): ";
232         std::getline(std::cin, newData[count].religion);
233         if (!validateReligion(newData[count].religion)) {
234             errorInput();
235         }
236     } while (!validateReligion(newData[count].religion));
237     do {
238         std::cout << "Status Perkawinan (Kawin, Belum Kawin): ";
239         std::getline(std::cin, newData[count].isMarried);
240         if (!validateIsMarried(newData[count].isMarried)) {
241             errorInput();
242         }
243     } while (!validateIsMarried(newData[count].isMarried));
244     do {
245         std::cout << "Penerapan: ";
246         std::cin >> newData[count].job;
247         if (!validateJob(newData[count].job)) {
248             errorInput();
249         }
250     } while (!validateJob(newData[count].job));
251     newData[count].citizenship = "Indo";
252     newData[count].validUntil = "31/12/2024";
253     time_t now = time(0);
254     tm *ltm = localtime(now);
255     newData[count].created_date = ltm->tm_mday;
256     newData[count].created_month = ltm->tm_mon + 1;
257     newData[count].created_year = ltm->tm_year + 1900;
258     for (int i = 0; i < newData[count].religion.length(); i++) {
259         newData[count].religion[i] = toUpper(newData[count].religion[i]);
260     }
261     for (int i = 0; i < newData[count].isMarried.length(); i++) {
262         newData[count].isMarried[i] = toUpper(newData[count].isMarried[i]);
263     }
264     std::ofstream file;
265     file.open("data.txt", std::ios::app);
266     file << "Provinsi: " << newData[count].province << std::endl;
267     file << "Kota: " << newData[count].city << std::endl;
268     file << "Kec: " << newData[count].village << std::endl;
269     file << "Nama: " << newData[count].name << std::endl;
270     file << "Tempat Lahir: " << newData[count].born_place << ", " << std::setw(10) << std::setw(2) << newData[count].born_date << ", " << std::setw(10) << std::setw(2) << newData[count].born_month << ", " << std::setw(10) << std::setw(2) << newData[count].born_year << std::endl;
271     file << "Jenis Kelamin: " << newData[count].gender << ", " << std::setw(10) << std::setw(2) << "Golongan Darah: " << newData[count].bloodtype << std::endl;
272     file << "Golongan Darah: " << newData[count].bloodtype << std::endl;
273     file << "Alamat: " << newData[count].address << std::endl;
274     file << "RT/RW: " << std::setw(10) << std::setw(2) << newData[count].address_neighborhoodAssociation << ", " << std::setw(10) << std::setw(2) << newData[count].address_communityAssociation << std::endl;
275     file << "Desa: " << newData[count].address_subdistrict << std::endl;
276     file << "Kabupaten: " << newData[count].address_district << std::endl;
277     file << "Agama: " << newData[count].religion << std::endl;
278     file << "Status Perkawinan: " << newData[count].isMarried << std::endl;
279     file << "Penerapan: " << newData[count].job << std::endl;
280     file << "Penerapan: " << newData[count].citizenship << std::endl;
281     file << "Validasi: " << newData[count].validUntil << std::endl;
282     file << "Masa Berlaku: " << std::setw(10) << std::setw(2) << newData[count].created_date << ", " << std::setw(10) << std::setw(2) << newData[count].created_month << ", " << std::setw(10) << std::setw(2) << newData[count].created_year << std::endl;
283     file << "Masa Berlaku: " << std::setw(10) << std::setw(2) << "Masa Berlaku: " << std::endl;
284     file.close();
285     cout << "Data berhasil disimpan!\n";
286     delete[] newData;
287 }

```

Kode tersebut merupakan definisi dari sebuah fungsi bernama `input()` yang digunakan untuk mengambil input dari pengguna dan menyimpannya ke dalam array `Data`.

Fungsi `input()` melakukan beberapa hal seperti:

- Membuat array baru dengan ukuran `count + 1` menggunakan operator `new`.
- Meminta pengguna untuk memasukkan data seperti provinsi, kota, NIK, nama, tempat lahir, tanggal lahir, jenis kelamin, golongan darah, alamat, RT, RW, kelurahan, kecamatan, agama, status pernikahan, dan pekerjaan.
- Memeriksa apakah data yang dimasukkan sesuai dengan format yang diinginkan menggunakan fungsi-fungsi validasi yang telah didefinisikan sebelumnya.
- Jika data yang dimasukkan tidak valid, maka fungsi `errorInput()` akan dipanggil untuk menampilkan pesan error pada console output.
- Jika data yang dimasukkan valid, maka data tersebut akan disimpan ke dalam array `newData`.
- Data yang telah dimasukkan akan ditulis ke dalam file `data.txt` menggunakan objek `ofstream`.
- Array data akan dihapus menggunakan operator `delete[]`.
- Array `newData` akan disimpan ke dalam array data.
- Jumlah data `count` akan ditambah 1.



Kode tersebut merupakan definisi dari sebuah fungsi bernama `output()` yang digunakan untuk menampilkan data-data KTP yang telah diinputkan ke dalam console output.

Fungsi `output()` melakukan beberapa hal seperti:

- Menampilkan header KTP pada console output.
- Melakukan perulangan untuk setiap data KTP yang telah diinputkan.
- Menghitung panjang maksimal antara nama provinsi dan nama kota, kemudian menampilkan nama provinsi dan nama kota dengan rata tengah.

- Menampilkan data-data KTP seperti NIK, nama, tempat dan tanggal lahir, jenis kelamin, golongan darah, alamat, RT, RW, kelurahan, kecamatan, agama, status pernikahan, pekerjaan, kewarganegaraan, masa berlaku KTP, dan tanda tangan pada console output

```

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     data = new Data[count];
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         else if (line.find("=====") != std::string::npos) i++;
384     }
385     file.close();
386 }
387 }
388
389 };

```

Kode tersebut merupakan definisi dari sebuah fungsi bernama initialize() yang digunakan untuk menginisialisasi data-data KTP yang telah disimpan dalam file data.txt ke dalam array Data.

Fungsi initialize() melakukan beberapa hal seperti:

- Membuka file data.txt menggunakan objek ifstream.
- Menghitung jumlah data KTP yang telah disimpan dalam file data.txt dengan mencari jumlah baris yang mengandung kata "NIK".
- Mengalokasikan memori untuk array data dengan ukuran sesuai dengan jumlah data KTP yang telah dihitung sebelumnya.
- Membaca setiap baris dalam file data.txt dan menyimpan data-data KTP ke dalam array data.
- Mengubah tipe data beberapa atribut seperti tanggal lahir, jenis kelamin, RT, RW, dan tanggal pembuatan KTP dari string ke tipe data yang sesuai.
- Menutup file data.txt.



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

Kode tersebut merupakan definisi dari fungsi main() yang merupakan program utama dari aplikasi KTP.

Fungsi main() melakukan beberapa hal seperti:

- Membuat objek KTP dengan nama ktp.
- Menampilkan menu pilihan untuk input data, tampilkan data, atau keluar dari program.
- Meminta pengguna untuk memilih salah satu pilihan.
- Jika pengguna memilih pilihan 1, maka fungsi input() dari objek ktp akan dipanggil untuk mengambil input data dari pengguna.
- Jika pengguna memilih pilihan 2, maka fungsi output() dari objek ktp akan dipanggil untuk menampilkan data-data KTP yang telah diinputkan.
- Jika pengguna memilih pilihan 3, maka program akan keluar.
- Jika pengguna memilih pilihan selain 1, 2, atau 3, maka program akan menampilkan pesan "Pilihan tidak valid".
- Setelah pengguna memilih salah satu pilihan, program akan menunggu pengguna menekan tombol untuk melanjutkan dan membersihkan layar console

Pada kode tersebut, terdapat array dinamis yang dideklarasikan sebagai pointer ke struct Data. Array dinamis ini digunakan untuk menyimpan data KTP yang diinputkan oleh pengguna. Array dinamis ini dideklarasikan sebagai berikut:

```
Data *data = nullptr;
```

Pada saat program dijalankan, array dinamis ini diinisialisasi dengan alokasi memori sebanyak jumlah data yang tersimpan pada file "data.txt". Hal ini dilakukan pada fungsi "initialize()" dengan menggunakan operator "new":

```
data = new Data[count];
```

Ketika pengguna memasukkan data baru, array dinamis ini akan diperbesar sebanyak satu elemen dengan menggunakan operator "new" dan data lama akan disalin ke array baru. Hal ini dilakukan pada fungsi "input()":



```

Data *newData = new Data[count + 1];
// ...
for (int i = 0; i < count; i++) {
    newData[i] = data[i];
}
delete[] data;
data = newData;
count++;
    
```

