

**LAPORAN FINAL PROJECT**  
**PEMROGRAMAN LANJUT STUDI KASUS**  
**SISTEM INFORMASI PENDATAAN**  
**PEGAWAI PERUSAHAAN**

Mata Kuliah Pemrograman Lanjut

Dosen Pengampu :

Wahyu SJ Saputra, S.Kom, M.Kom



Disusun Oleh :

Mochammad Arya Salsabila (19081010001)

Kelas D

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UPN "VETERAN" JAWA TIMUR**  
**2020**

## PEMBAHASAN SOURCE CODE

### FILE HEADER

```
// Nama : Mochammad Arya Salsabila
// Npm  : 19081010001
// Kelas : Pemrograman Lanjut (D)
// Studi Kasus : Sistem Informasi Pendataan Pegawai Perusahaan

#include <iostream>
#include <fstream>

using namespace std;
```

Syntax `#include` digunakan untuk menambahkan atau menyisipkan isi file “Header” ke dalam kode program saat program dicompile. “*iostream*” adalah header standar yang digunakan untuk menampilkan input dan output melalui *std::cin* dan *std::cout*. Dengan menggunakan syntax *using namespace std* kita tidak perlu lagi menuliskan *std::cin* dan *std::cout* untuk input output, tetapi cukup dengan *cin* dan *cout*. “*fstream*” adalah header yang digunakan untuk melakukan operasi file pada bahasa C++.

### DEKLARASI STRUCT

```
// Deklarasi struct
typedef struct {
    char nip_pegawai[20];
    char nama_pegawai[30];
    char tempat_lahir[20];
    char tgl_lahir[30];
    char alamat[50];
    char jabatan[20];
    char status[20];
} data_pegawai;
```

Deklarasi tipe data *struct* dengan nama *data\_pegawai* yang mempunyai field *nip\_pegawai*, *nama\_pegawai*, *tempat\_lahir*, *tgl\_lahir*, *alamat*, *jabatan*, dan *status*.

### DEKLARASI PROTOTYPE PROCEDURE

```
// Prototype procedure input_data(), show_data(),
// simpan_ke_file(), baca_dari_file()
void input_data(data_pegawai &pegawai);
void show_data(data_pegawai pegawai);
```

```
void simpan_ke_file();
void baca_dari_file();
```

Mendeklarasikan prototype procedure (fungsi tanpa nilai balik) untuk masing-masing procedure *simpan\_ke\_file()*, *baca\_dari\_file()*, *input\_data()*, dan *show\_data()*. Dengan mendeklarasikan prototype procedure maka kita dapat meletakkan procedure yang dibuat di atas atau bisa juga di bawah fungsi *main()*. Namun jika tidak mendeklarasikan prototype procedure kita hanya bisa mendeklarasikan procedure diatas fungsi *main()*.

## DEKLARASI FUNGSI MAIN

```
// Fungsi main atau fungsi utama program
int main() {
    bool exit = false;

    while (!exit) {
        int choice;
        cout << "===== " << endl;
        cout << "| Sistem Informasi Pendataan Pegawai Perusahaan |" << endl;
        cout << "===== " << endl;
        cout << "| Menu : " << endl;
        cout << "| 1. Tambah Pegawai " << endl;
        cout << "| 2. Lihat Pegawai " << endl;
        cout << "| 3. Keluar " << endl;
        cout << "===== " << endl;
        cout << endl;

        cout << "Masukkan pilihan : ";
        cin >> choice;
        cout << endl;

        switch (choice) {
            case 1:
                simpan_ke_file();
                cout << "Data berhasil ditambahkan..." << endl << endl;
                system("pause");
                system("cls");
                break;
            case 2:
                baca_dari_file();
                cout << "Data berhasil dibaca..." << endl << endl;
                system("pause");
                system("cls");
                break;
            case 3:
                exit = true;
                cout << "Keluar dari program..." << endl << endl;
        }
    }
}
```

```

        system("pause");
        system("cls");
        break;
    default:
        cout << "Pilihan salah" << endl;
        system("pause");
        system("cls");
    }
}

return 0;
}

```

Ada variabel *exit* bertipe *boolean* yang bernilai *false*. Lalu ada deklarasi variabel *struct* dengan nama *pegawai*. Selanjutnya dilakukan looping dengan kondisi *!exit* (dengan kata lain jika variabel *exit* bernilai *false* maka looping akan dijalankan terus menerus karena negasi dari variabel *exit* yang bernilai *false* adalah *true*). Kemudian ada input pilihan menu yang disimpan pada variabel *choice* dan dilakukan pengecekan dengan *switch case*, jika pilihan :

- Angka 1 (Tambah Pegawai) maka akan memanggil procedure *simpan\_ke\_file()*.
- Angka 2 (Lihat Pegawai) maka akan memanggil procedure *baca\_dari\_file()*.
- Angka 3 (Keluar) maka nilai variabel *exit* akan di assign menjadi *true* sehingga *!exit* (negasi dari nilai variabel *exit*) bernilai *false* dan dengan begitu looping akan berhenti dan keluar dari program.
- Selain angka 1, 2, dan 3 maka akan menampilkan kata-kata “Pilihan salah”.

Syntax *system(“pause”)* digunakan untuk mencetak kalimat “Press any key to continue . . .” dan menunggu user menginputkan key apa saja sebelum melanjutkan programnya. Syntax *system(“cls”)* digunakan untuk clear screen atau membersihkan layar pada program.

## DEFINISI PROCEDURE INPUT\_DATA

```

// Definisi procedure input_data(), show_data(),
// simpan_ke_file(), baca_dari_file()
void input_data(data_pegawai &pegawai) {
    cout << "Nip pegawai    : ";
    fflush(stdin); cin.get(pegawai.nip_pegawai, 20);
    cout << "Nama pegawai    : ";
    fflush(stdin); cin.get(pegawai.nama_pegawai, 30);
    cout << "Tempat lahir    : ";
    fflush(stdin); cin.get(pegawai.tempat_lahir, 20);
    cout << "Tanggal lahir    : ";
    fflush(stdin); cin.get(pegawai.tgl_lahir, 30);
    cout << "Alamat          : ";
    fflush(stdin); cin.get(pegawai.alamat, 50);
    cout << "Jabatan          : ";
    fflush(stdin); cin.get(pegawai.jabatan, 20);
}

```

```

    cout << "Status pegawai : ";
    fflush(stdin); cin.get(pegawai.status, 20);
    cout << endl;
}

```

Pertama ada definisi procedure *input\_data()*, procedure ini digunakan untuk input data pegawai yang diinputkan oleh user. Procedure *input\_data()* menerima parameter berupa *struct data\_pegawai* dengan nama variabel struct *&pegawai*. Tanda *&* pada nama variabel *struct* pegawai mempunyai arti bahwa parameter tersebut dilewatkan berdasarkan alamatnya (*pass by reference*) sehingga nilai di dalam procedure akan sama dengan nilai yang ada di luar procedure. Syntax *fflush(stdin)* digunakan untuk memeras *buffer* (*memori penyangga*) untuk menampung data string dari keyboard secara temporer untuk dimasukkan ke dalam variabel yang ditentukan. Untuk pengisian nilai *field struct* dari sebuah *struct* digunakan syntax *variabel\_struct.field\_struct*.

## DEFINISI PROCEDURE SHOW\_DATA

```

void show_data(data_pegawai pegawai) {
    cout << "Nip pegawai      : " << pegawai.nip_pegawai << endl;
    cout << "Nama pegawai       : " << pegawai.nama_pegawai << endl;
    cout << "Tempat lahir        : " << pegawai.tempat_lahir << endl;
    cout << "Tanggal lahir       : " << pegawai.tgl_lahir << endl;
    cout << "Alamat              : " << pegawai.alamat << endl;
    cout << "Jabatan             : " << pegawai.jabatan << endl;
    cout << "Status pegawai      : " << pegawai.status << endl << endl;
}

```

Yang kedua ada definisi procedure *show\_data()* yang menerima parameter berupa *struct data\_pegawai* dengan nama variabel struct *pegawai*. Procedure ini digunakan untuk menampilkan data pegawai. Untuk memanggil nilai *field struct* dari sebuah *struct* digunakan syntax *cout* lalu gunakan *variabel\_struct.field\_struct*.

## DEFINISI PROCEDURE SIMPAN\_KE\_FILE

```

void simpan_ke_file() {
    data_pegawai pegawai;

    // Untuk input data pegawai
    input_data(pegawai);

    ofstream file;
    file.open("data.txt", ios::binary | ios::app);

    // Menulis data pegawai ke file "data.txt"
    file.write((char*) &pegawai, sizeof(pegawai));
}

```

```
file.close();  
}
```

Yang ketiga ada definisi procedure *simpan\_ke\_file()*. Dalam procedure ini dideklarasikan *struct data\_pegawai* dengan nama variabel struct *pegawai*. Setelah itu dilakukan pemanggilan procedure *input\_data(pegawai)* untuk menginputkan data pegawai.

```
ofstream file;  
file.open("data.txt", ios::binary | ios::app);
```

Pada syntax diatas dideklarasikan variabel bernama *file* yang bertipe *ofstream*. *Ofstream* adalah tipe data file pada bahasa C++ yang digunakan untuk menciptakan file dan menulis informasi ke dalam file. Dengan demikian jika file "*data.txt*" sebelumnya tidak ada pada direktori program tersebut, maka compiler akan membuat file "*data.txt*" pada direktori yang sama dengan direktori tempat program tersebut disimpan. File "*data.txt*" dibuka dalam mode *ios::binary* (membuka file dengan operasi baca atau tulis secara biner) dan *ios::app* (untuk menambahkan data pada suatu file. Data ditambahkan di akhir file, sehingga data sebelumnya pada file yang sama tidak hilang. Jika file yang dimaksud tidak ditemukan maka program akan membuat file baru).

```
// Menulis data pegawai ke file "data.txt"  
file.write((char*) &pegawai, sizeof(pegawai));
```

Pada syntax diatas data yang ada di variabel struct *pegawai* ditulis ke file "*data.txt*" dengan memanggil fungsi *write()*. *(\*char) &pegawai* adalah *pointer* yang digunakan untuk menunjuk ke awal struct pegawai dan *sizeof(pegawai)* digunakan untuk menghitung jumlah byte dari struct *pegawai* yang akan disalin atau ditulis ke dalam file "*data.txt*".

```
file.close();  
}
```

Setelah dibuka dan diproses, file harus ditutup kembali. Untuk menutup file pada bahasa C++ digunakan syntax seperti diatas.

## DEFINISI PROCEDURE BACA\_DARI\_FILE

```
void baca_dari_file() {  
    data_pegawai pegawai;  
  
    ifstream file;  
    file.open("data.txt", ios::binary);
```

```

    if (!file) {
        cout << "Belum ada data, silahkan tambahkan data terlebih dahulu...";
        cout << endl << endl;
    } else {
        // Membaca data pegawai dari file "data.txt"
        while (file.read((char *) &pegawai, sizeof(pegawai))) {
            // Untuk menampilkan data pegawai
            show_data(pegawai);
        }
    }

    file.close();
}

```

Yang keempat ada definisi procedure *baca\_dari\_file()*. Dalam procedure ini dideklarasikan *struct data\_pegawai* dengan nama variabel struct *pegawai*.

```

ifstream file;
file.open("data.txt", ios::binary);

```

Pada syntax diatas dideklarasikan variabel bernama *file* yang bertipe *ifstream*. *Ifstream* adalah tipe data file pada bahasa C++ yang digunakan untuk membaca informasi dari file. File “*data.txt*” dibuka dalam mode *ios::binary* (membuka file dengan operasi baca atau tulis secara biner).

```

if (!file) {
    cout << "Belum ada data, silahkan tambahkan data terlebih dahulu...";
    cout << endl << endl;
} else {
    // Membaca data pegawai dari file "data.txt"
    while (file.read((char *) &pegawai, sizeof(pegawai))) {
        // Untuk menampilkan data pegawai
        show_data(pegawai);
    }
}

```

Selanjutnya dilakukan pengecekan apakah file “*data.txt*” yang dibuka sudah ada atau belum dengan syntax *if(!file)*. Jika ternyata file “*data.txt*” tidak ada yang artinya belum ada data maka program akan menampilkan kata-kata “*Belum ada data, silahkan tambahkan data terlebih dahulu...*”. Namun jika file “*data.txt*” sudah ada, maka program akan menjalankan blok kode program yang ada di dalam *else*, yaitu :

```
// Membaca data pegawai dari file "data.txt"
while (file.read((char *) &pegawai, sizeof(pegawai))) {
    // Untuk menampilkan data pegawai
    show_data(pegawai);
}
```

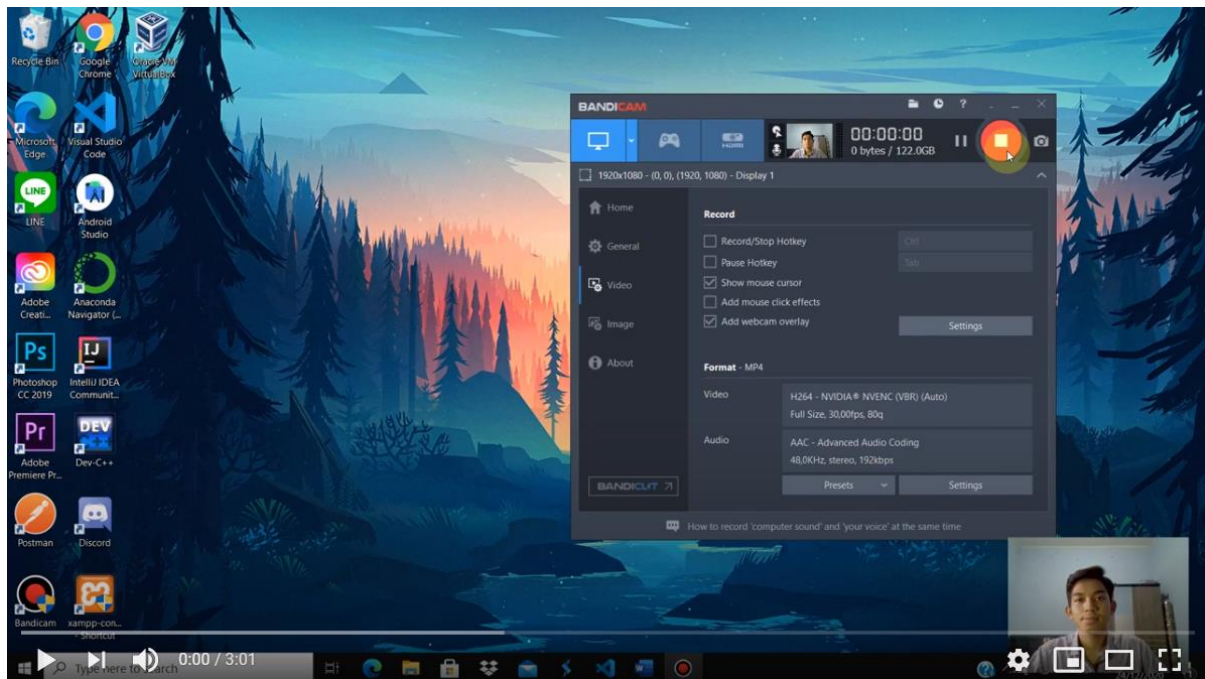
Pada syntax diatas dilakukan looping dengan kondisi data yang ada di variabel struct *pegawai* dibaca dari file “data.txt” dengan memanggil fungsi *read()*. (*\*char*) &*pegawai* adalah *pointer* yang digunakan untuk menunjuk ke awal struct pegawai dan *sizeof(pegawai)* digunakan untuk menghitung jumlah byte yang dibaca dari file “data.txt”. Selama looping terpenuhi akan dilakukan pemanggilan procedure *show\_data(pegawai)* untuk menampilkan data pegawai.

```
file.close();
}
```

Setelah dibuka dan diproses, file harus ditutup kembali. Untuk menutup file pada bahasa C++ digunakan syntax seperti diatas.

## LINK VIDEO DEMO PROGRAM

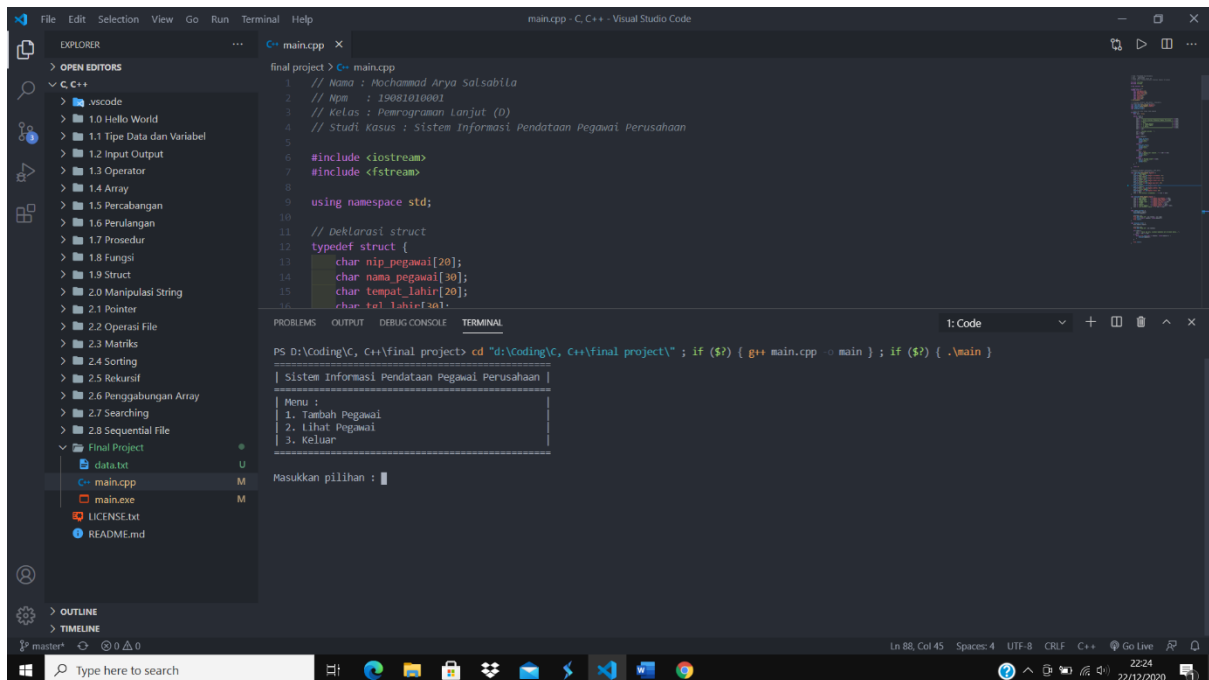
<https://youtu.be/mQwKyYcde4U>



Gambar 1 : Screenshot Video Demo Program



# OUTPUT PROGRAM



```
final project > C++ main.cpp
1 // Nama : Mochammad Arya Salsabila
2 // Npm : 19081010001
3 // Kelas : Pemrograman Lanjut (D)
4 // Studi Kasus : Sistem Informasi Pendataan Pegawai Perusahaan
5
6 #include <iostream>
7 #include <fstream>
8
9 using namespace std;
10
11 // Deklarasi struct
12 typedef struct {
13     char nip_pegawai[20];
14     char nama_pegawai[30];
15     char tempat_lahir[20];
16     char tel_lahir[30];
17 } Pegawai;
18
19 int main() {
20     int pilihan;
21     do {
22         system("cls");
23         cout << "=====
24         cout << "Sistem Informasi Pendataan Pegawai Perusahaan |
25         cout << "=====
26         cout << "Menu :
27         cout << "1. Tambah Pegawai
28         cout << "2. Lihat Pegawai
29         cout << "3. Keluar
30         cout << "=====
31         cout << "Masukkan pilihan : ";
32         int input;
33         while (input < 0 || input > 3) {
34             input = -1;
35             while (getchar() != '\n') continue;
36             cout << "Pilihan harus antara 1-3. Masukkan pilihan : ";
37             input = -1;
38         }
39         pilihan = input;
40         switch (pilihan) {
41             case 1:
42                 // Tambah Pegawai
43                 break;
44             case 2:
45                 // Lihat Pegawai
46                 break;
47             case 3:
48                 // Keluar
49                 break;
50             default:
51                 break;
52         }
53     } while (pilihan != 3);
54     return 0;
55 }
```

PS D:\Coding\C, C++\final project> cd "d:\Coding\C, C++\final project\" ; if (\$?) { g++ main.cpp -o main } ; if (\$?) { .\main }

| Sistem Informasi Pendataan Pegawai Perusahaan |

| Menu : |

| 1. Tambah Pegawai |

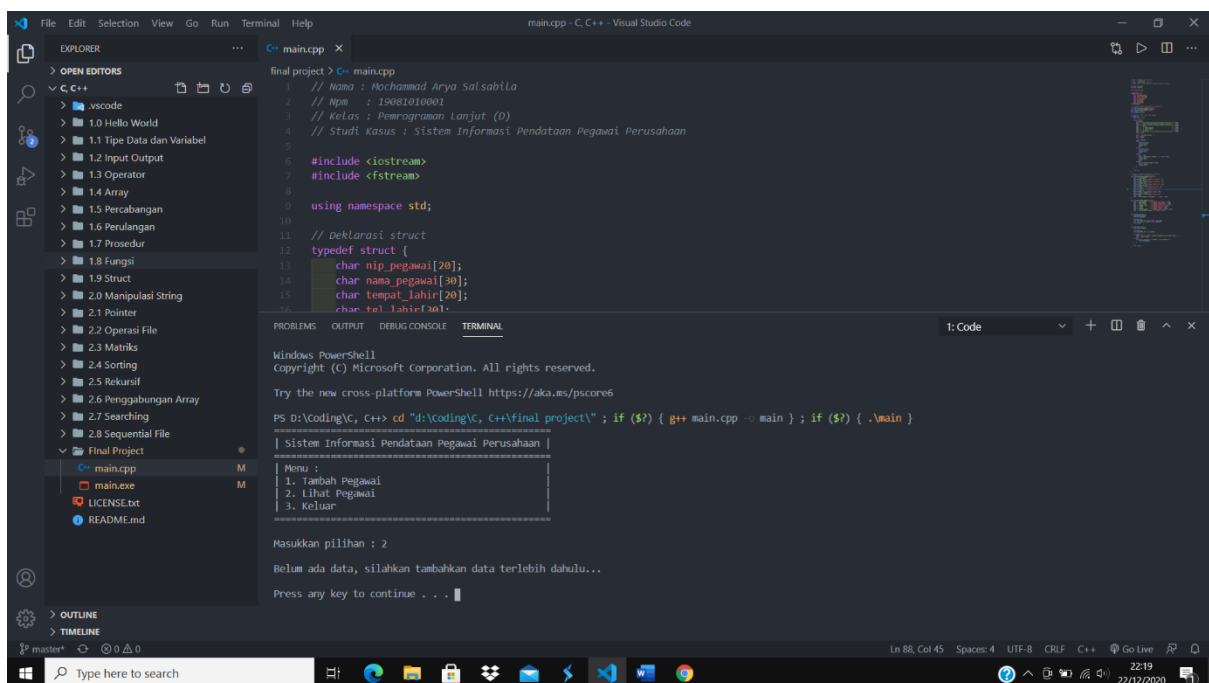
| 2. Lihat Pegawai |

| 3. Keluar |

|=====|

Masukkan pilihan :

Gambar 2 : Menu Utama Program. Terdapat 3 Submenu Yaitu Tambah Pegawai, Lihat Pegawai, dan Keluar.



```
final project > C++ main.cpp
1 // Nama : Mochammad Arya Salsabila
2 // Npm : 19081010001
3 // Kelas : Pemrograman Lanjut (D)
4 // Studi Kasus : Sistem Informasi Pendataan Pegawai Perusahaan
5
6 #include <iostream>
7 #include <fstream>
8
9 using namespace std;
10
11 // Deklarasi struct
12 typedef struct {
13     char nip_pegawai[20];
14     char nama_pegawai[30];
15     char tempat_lahir[20];
16     char tel_lahir[30];
17 } Pegawai;
18
19 int main() {
20     int pilihan;
21     do {
22         system("cls");
23         cout << "=====
24         cout << "Sistem Informasi Pendataan Pegawai Perusahaan |
25         cout << "=====
26         cout << "Menu :
27         cout << "1. Tambah Pegawai
28         cout << "2. Lihat Pegawai
29         cout << "3. Keluar
30         cout << "=====
31         cout << "Masukkan pilihan : ";
32         int input;
33         while (input < 0 || input > 3) {
34             input = -1;
35             while (getchar() != '\n') continue;
36             cout << "Pilihan harus antara 1-3. Masukkan pilihan : ";
37             input = -1;
38         }
39         pilihan = input;
40         switch (pilihan) {
41             case 1:
42                 // Tambah Pegawai
43                 break;
44             case 2:
45                 // Lihat Pegawai
46                 break;
47             case 3:
48                 // Keluar
49                 break;
50             default:
51                 break;
52         }
53     } while (pilihan != 3);
54     return 0;
55 }
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/powershell

PS D:\Coding\C, C++> cd "d:\Coding\C, C++\final project\" ; if (\$?) { g++ main.cpp -o main } ; if (\$?) { .\main }

| Sistem Informasi Pendataan Pegawai Perusahaan |

| Menu : |

| 1. Tambah Pegawai |

| 2. Lihat Pegawai |

| 3. Keluar |

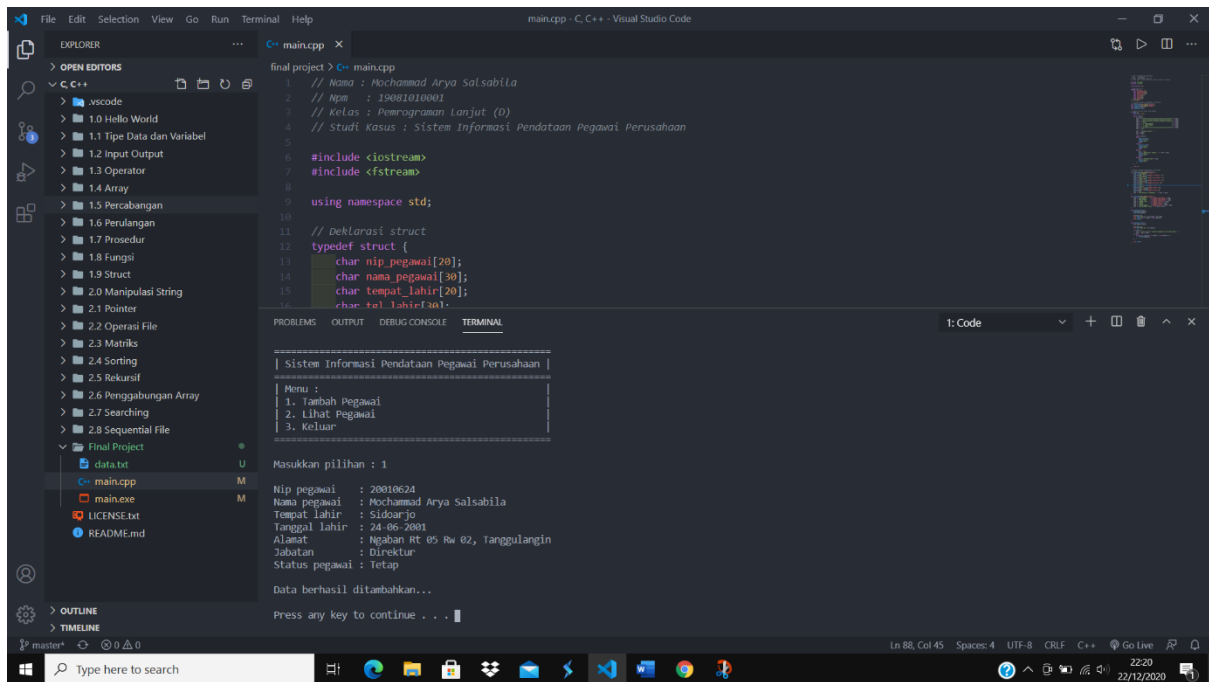
|=====|

Masukkan pilihan : 2

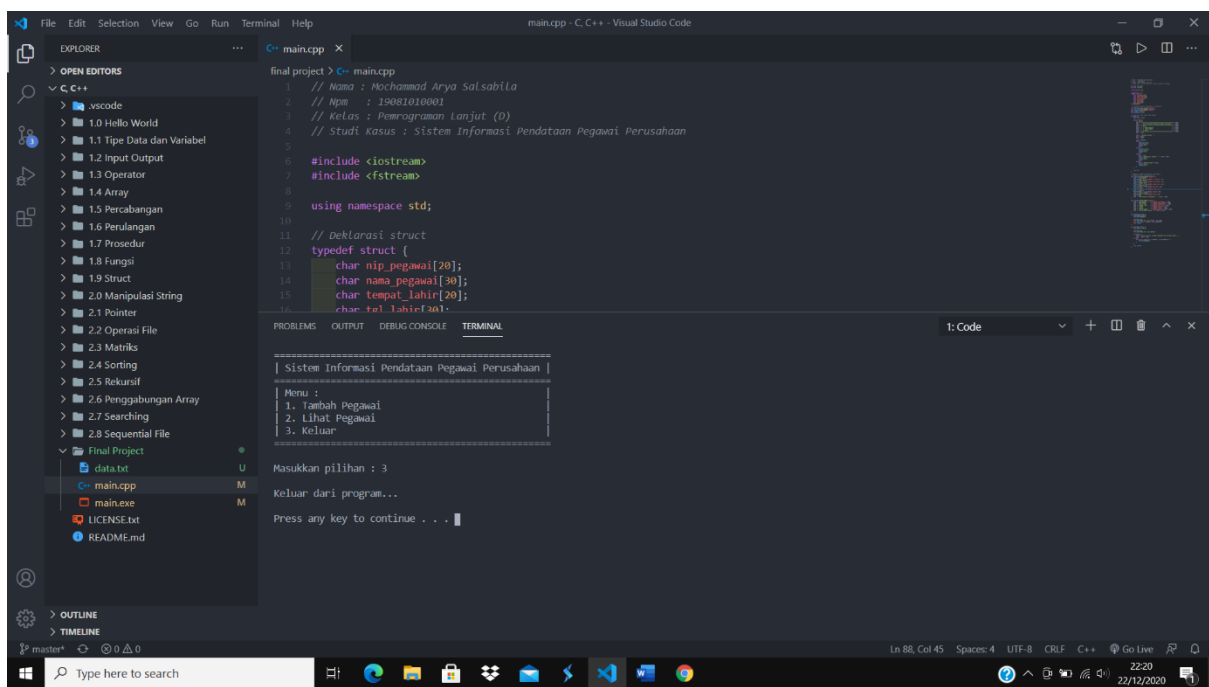
Belum ada data, silahkan tambahkan data terlebih dahulu...

Press any key to continue . . .

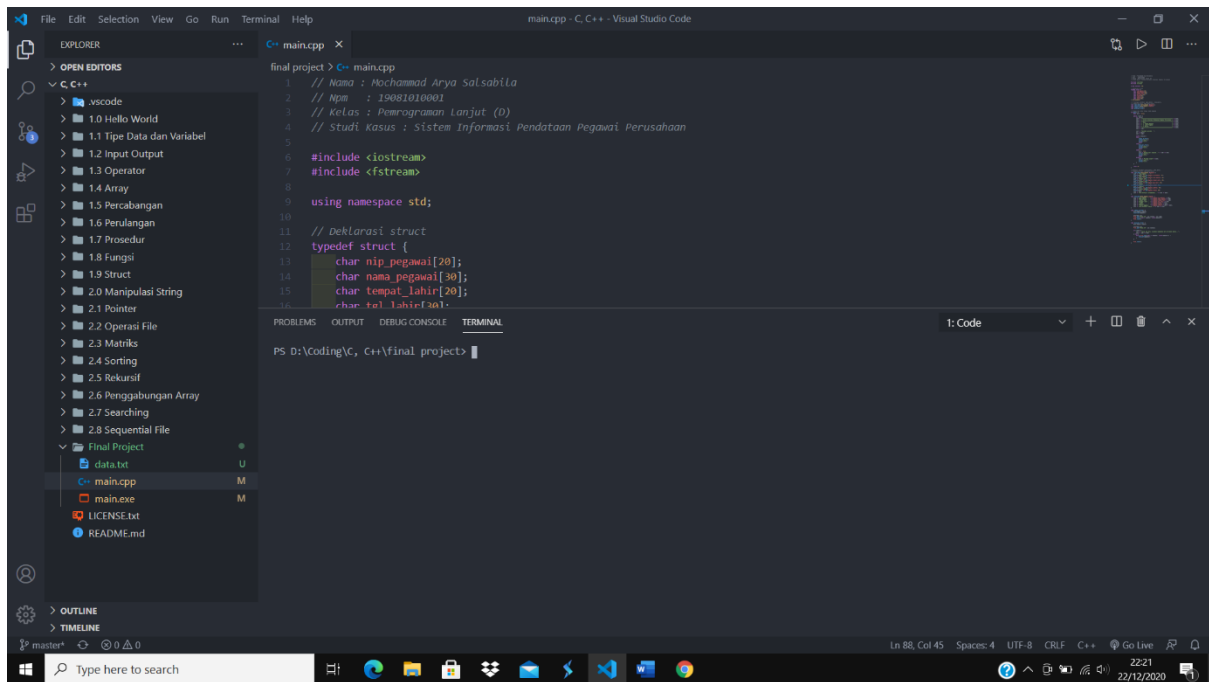
Gambar 3 : Menu Lihat Pegawai Ketika Belum Menambahkan Data Pegawai Sama Sekali.



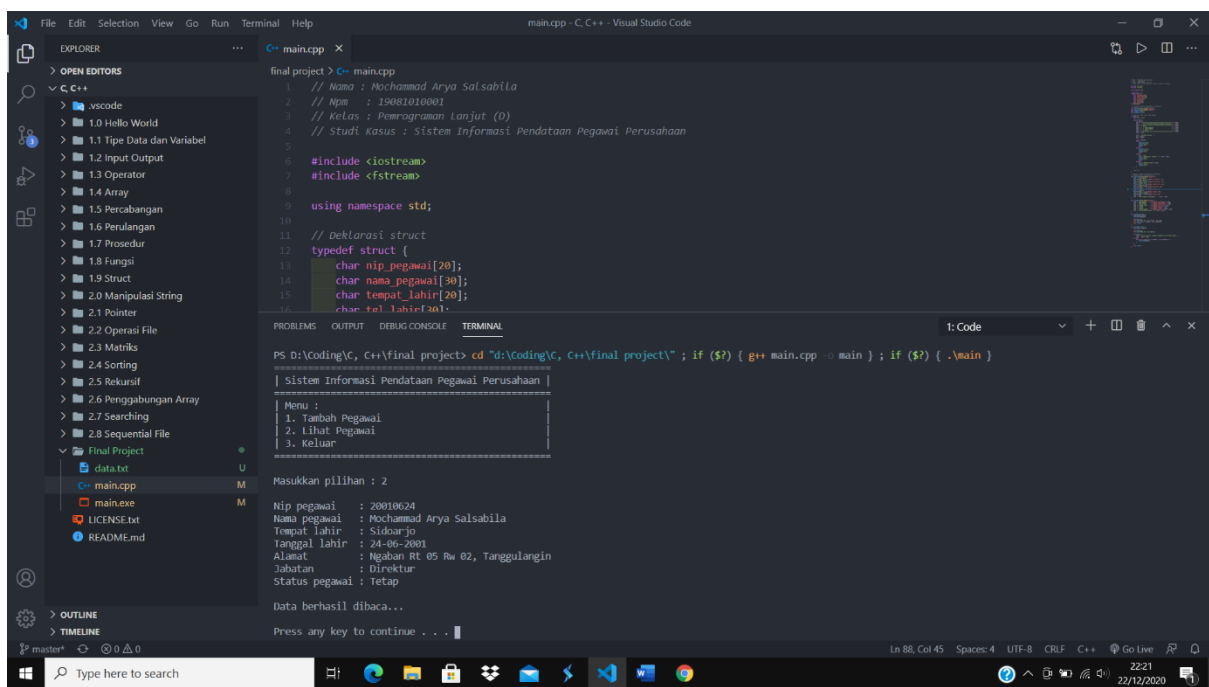
Gambar 4 : Menu Tambah Pegawai Lalu Input Data Pegawai Dan Terbentuk File “data.txt” Pada Direktori “Final Project”.



Gambar 5 : Menu Keluar Untuk Keluar Dari Program.



Gambar 6 : Tampilan Saat Telah Keluar Dari Program.



Gambar 7 : Mencoba Run Program Kembali Lalu Memilih Menu Lihat Pegawai Dan Ternyata Data Pegawai Masih Ada.