KELOMPOK

DEKLARASI STRUCT

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <cstring>


struct item_project
{
        char id[14] = {};
        char nama[32] = {};
        char sku[4] = {};
        int harga;
        int diskon_nominal;
        int diskon_persen;
        int stok;
};

struct pegawai_project
{
        char id[4] = {};
        char nama[32] = {};
        int harga_nondiskon = 0; // sebelum diskon
        int diskon_total = 0;
        int harga_diskon = 0; // setelah diskon
        int kuantitas_terjual = 0;
};

struct log_project
{
        char time[20] = {};
        char id_kasir[4] = {};
        char id_konsumen[4] = {};
        char id_item[14] = {};
        int harga;
        int diskon;
        int kuantiti;
};
```

```cpp
int item_count = -1;
int pegawai_count = -1;
int log_count = -1;
std::string top_line;
std::string s_temp;

// input file item.txt
std::ifstream item_istream("./data/item.txt");

while (std::getline(item_istream, top_line)) item_count++;
item_project arr_item[item_count];

item_istream.clear();
item_istream.seekg(0);
std::getline(item_istream, top_line);
for (int i = 0; i < item_count; i++)
{
        std::getline(item_istream, s_temp, '\t');
        strcpy(arr_item[i].id, s_temp.c_str());
        std::getline(item_istream, s_temp, '\t');
        strcpy(arr_item[i].nama, s_temp.c_str());
        std::getline(item_istream, s_temp, '\t');
        strcpy(arr_item[i].sku, s_temp.c_str());
        std::getline(item_istream, s_temp, '\t');
        arr_item[i].harga = stoi(s_temp);
        std::getline(item_istream, s_temp, '\t');
        arr_item[i].diskon_nominal = stoi(s_temp);
        std::getline(item_istream, s_temp, '\t');
        arr_item[i].diskon_persen = stoi(s_temp);
        std::getline(item_istream, s_temp);
        arr_item[i].stok = stoi(s_temp);
}
item_istream.close();
```

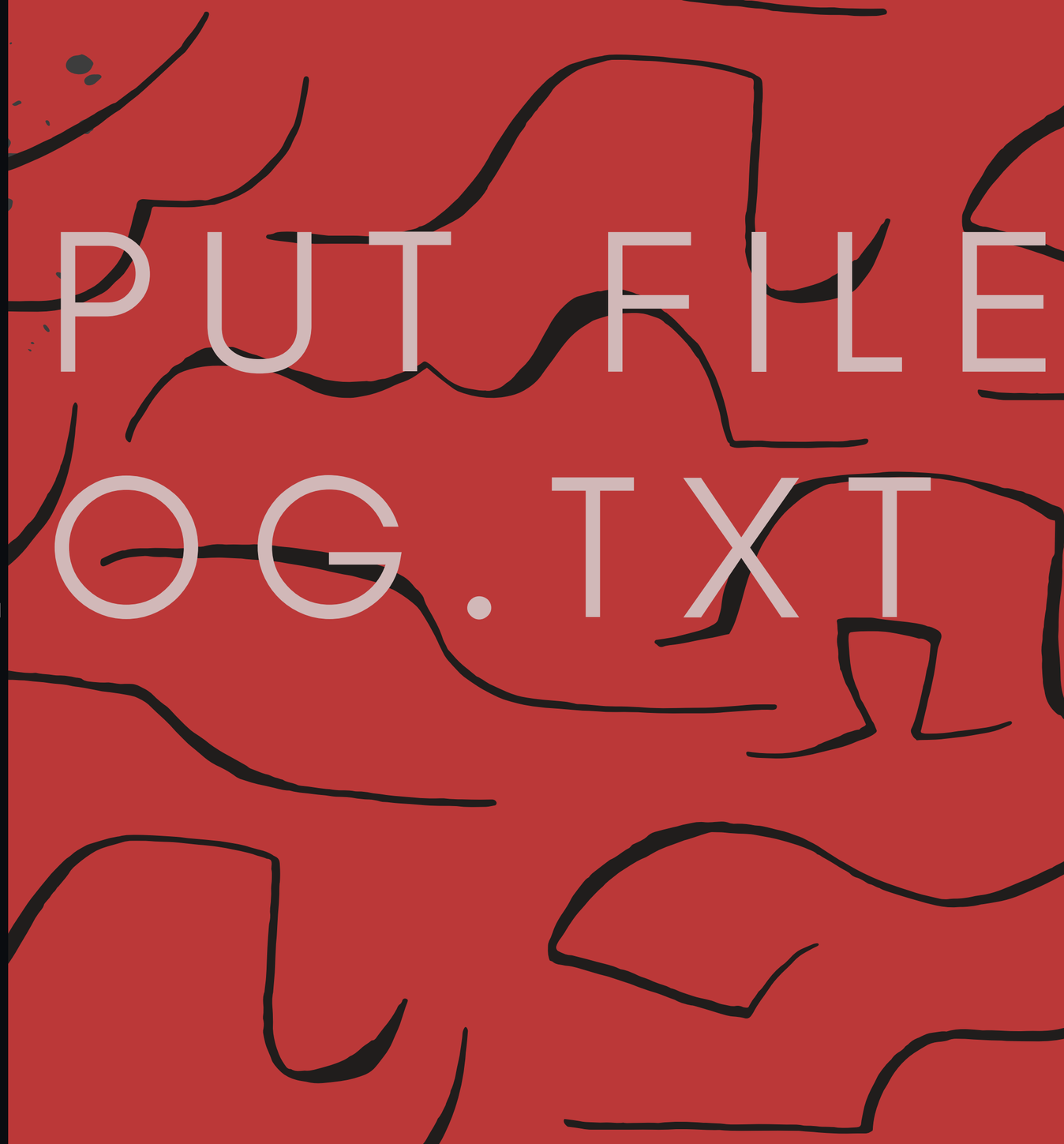
INPUT FILE ITEM.TXT

INPUT FILE

ITEM.TXT

```
// input file pegawai.txt
std::ifstream pegawai_istream("./data/pegawai.txt");

while (std::getline(pegawai_istream, top_line)) pegawai_count++;
pegawai_project arr_pegawai[pegawai_count];

pegawai_istream.clear();
pegawai_istream.seekg(0);
std::getline(pegawai_istream, top_line);
for (int i = 0; i < pegawai_count; i++)
{
        std::getline(pegawai_istream, s_temp, '\t');
        strcpy(arr_pegawai[i].id, s_temp.c_str());
        std::getline(pegawai_istream, s_temp);
        strcpy(arr_pegawai[i].nama, s_temp.c_str());
}
pegawai_istream.close();
```

```
// input file log.txt
std::ifstream log_istream("./data/log.txt");

while (std::getline(log_istream, top_line)) log_count++;
log_project arr_log[log_count];

log_istream.clear();
log_istream.seekg(0);
std::getline(log_istream, top_line);
for (int i = 0; i < log_count; i++)
{
        std::getline(log_istream, s_temp, '\t');
        strcpy(arr_log[i].time, s_temp.c_str());
        std::getline(log_istream, s_temp, '\t');
        strcpy(arr_log[i].id_kasir, s_temp.c_str());
        std::getline(log_istream, s_temp, '\t');
        strcpy(arr_log[i].id_konsumen, s_temp.c_str());
        std::getline(log_istream, s_temp, '\t');
        strcpy(arr_log[i].id_item, s_temp.c_str());
        std::getline(log_istream, s_temp, '\t');
        arr_log[i].harga = stoi(s_temp);
        std::getline(log_istream, s_temp, '\t');
        arr_log[i].diskon = stoi(s_temp);
        std::getline(log_istream, s_temp);
        arr_log[i].kuantiti = stoi(s_temp);
}
log_istream.close();
```



HITUNG PENJUALAN

```
// hitung total penjualan setiap pegawai
for (int i = 0; i < log_count; i++)
{
        for (int j = 0; j < pegawai_count; j++)
        {
                if (strcmp(arr_log[i].id_kasir, arr_pegawai[j].id) == 0)
                {
                        arr_pegawai[j].harga_nondiskon += arr_log[i].harga * arr_log[i].kuantiti;
                        arr_pegawai[j].diskon_total += arr_log[i].diskon * arr_log[i].kuantiti;
                        arr_pegawai[j].harga_diskon += (arr_log[i].harga - arr_log[i].diskon) * arr_log[i].kuantiti;
                        arr_pegawai[j].kuantitas_terjual += arr_log[i].kuantiti;
                        break;
                }
        }
}
```

OUTPUT PENJUALAN

```
// mengoutput file binary item.bin
std::ofstream bin_item_ostream("./data/item.bin", std::ios::out | std::ios::binary);
bin_item_ostream.write((char*)arr_item, sizeof(item_project) * item_count);
bin_item_ostream.close();

// mengoutput file binary pegawai.bin
std::ofstream bin_pegawai_ostream("./data/pegawai.bin", std::ios::out | std::ios::binary);
bin_pegawai_ostream.write((char*)arr_pegawai, sizeof(pegawai_project) * pegawai_count);
bin_pegawai_ostream.close();

// mengoutput file binary log.bin
std::ofstream bin_log_ostream("./data/log.bin", std::ios::out | std::ios::binary);
bin_log_ostream.write((char*)arr_log, sizeof(log_project) * log_count);
bin_log_ostream.close();

// mengoutput file total_penjualan.txt
std::ofstream txt_tp_ostream("./data/total_penjualan.txt");
txt_tp_ostream << "Nama" << '\t' << "Harga Nondiskon" << '\t' << "Diskon Total" << '\t' << "Harga Diskon" << '\t' << "Kuantitas" << '\n';
for (int i = 0; i < pegawai_count; i++)
{
        txt_tp_ostream << arr_pegawai[i].nama << '\t' << arr_pegawai[i].harga_nondiskon << '\t' << arr_pegawai[i].diskon_total << '\t' << arr_pegawai[i].harga_diskon
                                << '\t' << arr_pegawai[i].kuantitas_terjual << '\n';
}
txt_tp_ostream.close();

return 0;
```

TERIMAKASIH