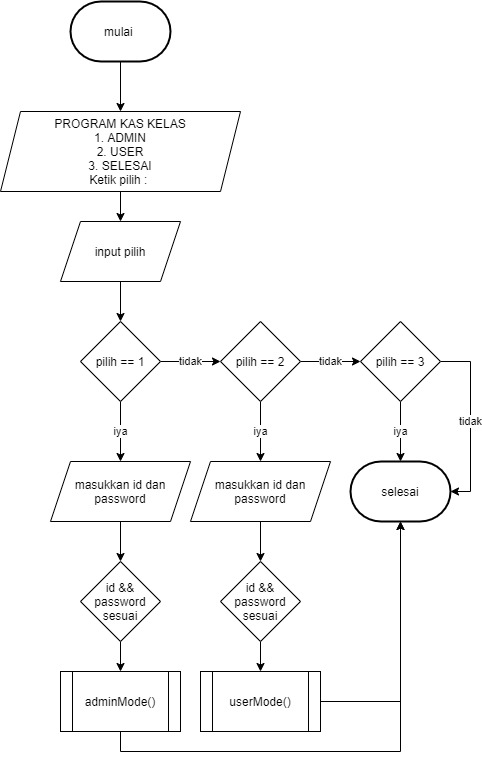
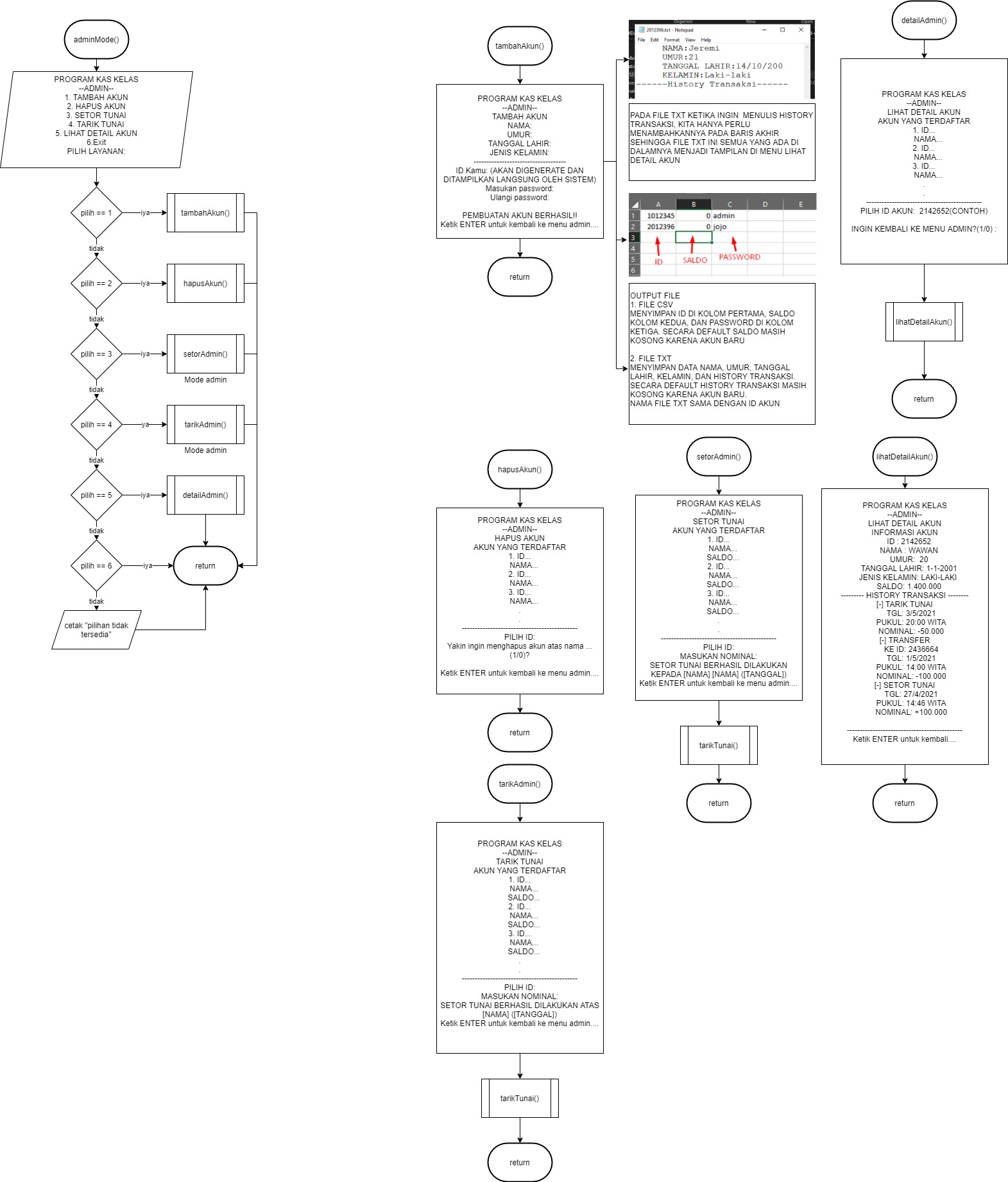
BAB 3

DESAIN

Tampilan Awal

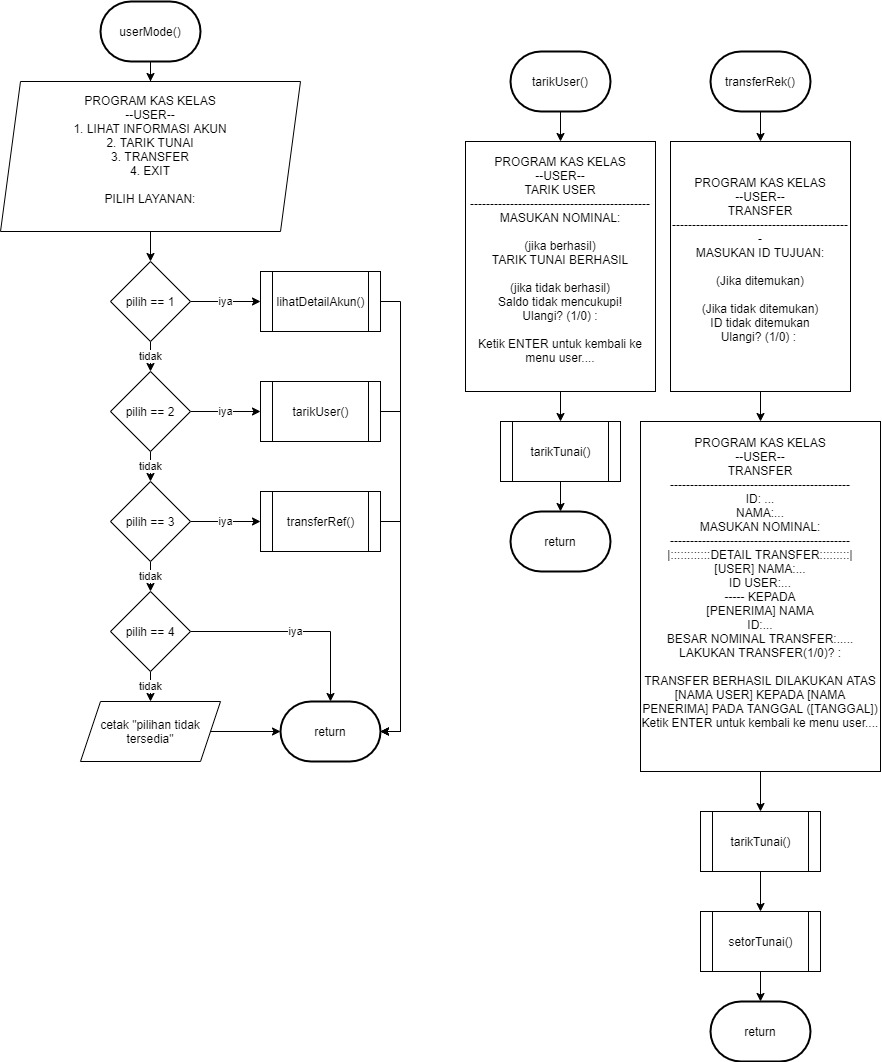
Flowchart di atas adalah desain bagaimana sebuah tampilan awal program, akan dibuat dua pilihan login, yaitu user dan admin.

Admin Mode



Ini adalah flowchart untuk tampilan admin. Fitur-fitur yang dimiliki mode admin lebih banyak dibandingkan dengan mode User. Setiap pilihan user akan memanggil fungsinya masing-masing.

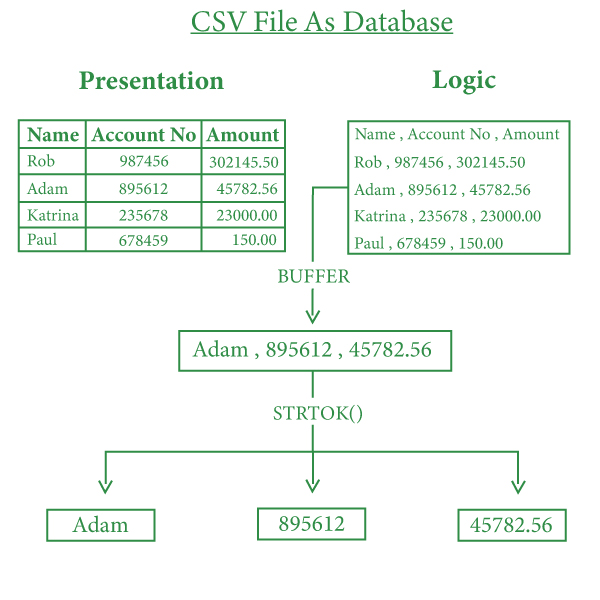
User Mode



Ini adalah flowchart untuk mode user. Mode user hanya memiliki 3 menu, yaitu lihat informasi akun, Tarik tunai, dan transfer ke sesame user.

METODE

Metode utama yang digunakan pada program ini adalah sebuah operasi file csv dan txt. File csv digunakan untuk menyimpan informasi akun seperti ID, saldo dan password. Untuk penerapan bagaimana operasi file pada sebuah file csv mungkin sedikir berbeda dengan file txt. Karena pada dasarnya file csv adalah file yang digunakan untuk menyimpan dan mengola data seperti file excel pada umunya.



File csv itu sendiri secara logika berbeda dengan file excel (.xls). file csv tidak bisa memuat formula dan bersifat text yang di mana tiap kolom dipisahkan dengan tanda koma, seperti nama csv itu sendiri adalah **C**omma **S**eparated **V**alues. Dalam pembacaannya kita hanya membacanya secara sederhana seperti biasa kita membaca file txt, hanya yang membedakan kita gunakan sebuah fungsi dari library string.h yaitu strtok untuk membagi tiap-tiap kolom. Dan ini dilakukan secara perulangan menggunakan while.

Ketika dalam pembacan filenya kita memasukannya dalam buffer. Buat buffer dari array karakter (dapat disebut sebagai string) yang mengambil semua data yang ada dalam file dan dengan menggunakan File Pointer dan fgets () data dapat diekstraksi. Gunakan dua variabel baris dan kolom yang akan mempertahankan identifikasi unik dari setiap entri.

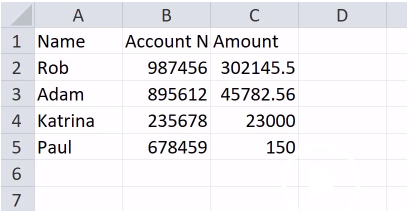
Karena string berisi koma ',' untuk memisahkan nilai, jadi idenya adalah menggunakan fungsi strtok () untuk memisahkan nilai. Fungsi ini membagi string menggunakan pembatas di sini kita menggunakan ','.

Cara Ekstrasi datanya

Ekstraksi Data berkaitan dengan Membuka file CSV yang ada dan mengekstrak serta mencetak seluruh data di konsol.

1. Buka File CSV menggunakan File Pointer.
2. Ekstrak seluruh data file ke dalam array buffer karakter.
3. Sekarang inisialisasi variabel baris dan kolom dengan nilai 0.
4. Cetak data yang dipisahkan dengan koma dan tingkatkan variabel kolom.
5. Saat mencapai akhir entri baris, inisialisasi variabel kolom ke 0 dan tingkatkan variabel baris.
6. Ulangi langkah 4 dan 5, hingga penunjuk mencapai akhir file.
7. Tutup file.

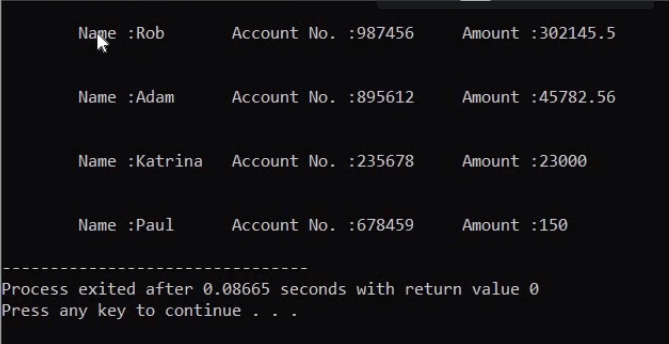
CONTOH



Berikut contoh implementasi dalam program

*// C program for the above approach*#include <conio.h>  
#include <stdio.h>  
#include <string.h>  
  
*// Driver Code  
int* main()  
{  
 *// Substitute the full file path  
 // for the string file\_path* FILE\* fp = fopen("file\_path", "r");  
  
 *if* (!fp)  
 printf("Can't open file\n");  
  
 *else* {  
 *// Here we have taken size of  
 // array 1024 you can modify it  
 char* buffer[1024];  
  
 *int* row = 0;  
 *int* column = 0;  
  
 *while* (fgets(buffer,  
 1024, fp)) {  
 column = 0;  
 row++;  
  
 *// To avoid printing of column  
 // names in file can be changed  
 // according to need  
 if* (row == 1)  
 *continue*;  
  
 *// Splitting the data  
 char*\* value = strtok(buffer, ", ");  
  
 *while* (value) {  
 *// Column 1  
 if* (column == 0) {  
 printf("Name :");  
 }  
  
 *// Column 2  
 if* (column == 1) {  
 printf("\tAccount No. :");  
 }  
  
 *// Column 3  
 if* (column == 2) {  
 printf("\tAmount :");  
 }  
  
 printf("%s", value);  
 value = strtok(**NULL**, ", ");  
 column++;  
 }  
  
 printf("\n");  
 }  
  
 *// Close the file* fclose(fp);  
 }  
 *return* 0;  
}

OUTPUT HASIL PROGRAMNYA



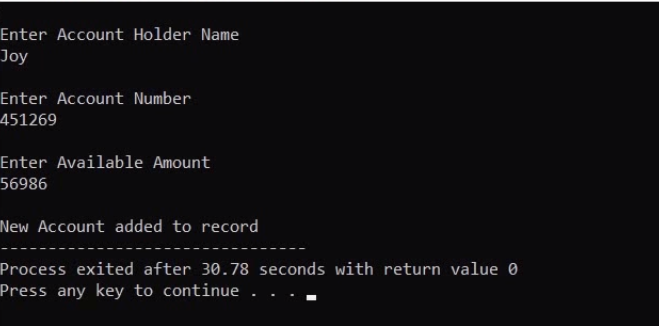
Untuk menambahkan datanya ke dalam file csv. Di sini kita gunakan fprintf(). Penambahan Data berkaitan dengan membuka file CSV yang ada, mengambil input pengguna untuk data yang akan ditambahkan ke file, dan kemudian menambahkan data ini ke file CSV

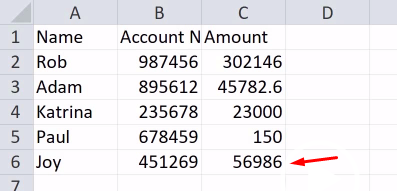
1. Buka File CSV menggunakan File Pointer dalam mode append yang akan menempatkan penunjuk ke bagian akhir file.
2. Ambil Masukan dari pengguna dalam variabel sementara.
3. Gunakan fprintf () dan pisahkan variabel sesuai dengan urutan dan komanya.
4. Tutup file.

Berikut contoh implementasi dalam program

*// C program for the above approach*#include <conio.h>  
#include <stdio.h>  
#include <string.h>  
  
*// Driver Code  
int* main()  
{  
 *// Substitute the file\_path string  
 // with full path of CSV file* FILE\* fp = fopen("file\_path", "a+");  
  
 *char* name[50];  
 *int* accountno, amount;  
  
 *if* (!fp) {  
 *// Error in file opening* printf("Can't open file\n");  
 *return* 0;  
 }  
  
 *// Asking user input for the  
 // new record to be added* printf("\nEnter Account Holder Name\n");  
 scanf("%s", &name);  
 printf("\nEnter Account Number\n");  
 scanf("%d", &accountno);  
 printf("\nEnter Available Amount\n");  
 scanf("%d", &amount);  
  
 *// Saving data in file* fprintf(fp, "%s, %d, %d\n", name,  
 accountno, amount);  
  
 printf("\nNew Account added to record");  
  
 fclose(fp);  
 *return* 0;  
}

OUTPUT PROGRAM





Operasi File Text

Operasi file text digunakan untuk menyimpan detail dari sebuah akun meliputi nama, umur, tanggal lahir dan jenis kelamin. Selain itu file text juga digunakan sebagai penyimpan riwayat transaksi yang dilakukan. Tidak ada metode khusus yang digunakan dalam operasi file text ini, data hanya ditambahkan baris per baris begitu pula untuk sistem pembacaan file.

Pembacaa file text

Pembacaan file dilakukan dengan menggunakan fungsi fgets() karena file tidak akan dibaca secara terformat melainkan hanya sebagai string. Langkah-langkah pembacaan sebuah file text yaitu:

1. Buka File txt menggunakan File Pointer dalam mode read karena file text dibuka hanya untuk tujuan pembacaan.
2. Jika file tidak ada maka program tidak akan dilanjutkan.
3. Deklarasikan sebuah array sebagai penampung karakter yang akan dibaca.
4. Fungsi fgets() digunakan untuk membaca setiap karakter yang ada pada file kemudian disimpan pada array penampung.
5. Karakter yang telah tersimpan dalam array selanjutnya dicetak menggunakan fungsi printf().
6. Selama fungsi fgets() tidak mengembalikan nilai NULL, ulangi langkah 3 dan 4.
7. Jika fgets() mengembalikan nilai NULL, menandakan bahwa penunjuk telah mencapai akhir dari file.
8. Tutup file.

Implementasi pembacaan file text dalam program untuk menampilkan detail dari sebuah akun:

|  |  |
| --- | --- |
| Kode program | Output program |
|  |  |

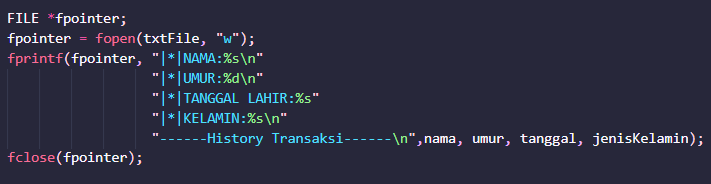
Penulisan file text

Isi file text ditulis dengan menggunakan fungsi fprintf(). Identitas dan transaksi yang dilakukan akan ditulis kedalam file text tersebut. Langkah-langkah menulis file text yaitu:

1. Buka File txt menggunakan File Pointer dalam mode append karena akan dilakukan penulisan pada akhir file.
2. Tulis data ke dalam file menggunakan fungsi fprintf.
3. Setelah penulisan selesai, tutup kembali file.

Implementasi penulisan file text dalam program :

Kode program :



Output pada file :

