**APLIKASI HUBUNGAN *CELEBRITY***

**DENGAN *FOLLOWER* DAN SEBALIKNYA**

**LAPORAN TUGAS BESAR STRUKTUR DATA**

oleh:

Mar Ayu Fotina (NIM 1301174013)

Putri Apriyanti Windya (NIM 1301174169)

Kelas: IF-41-05



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

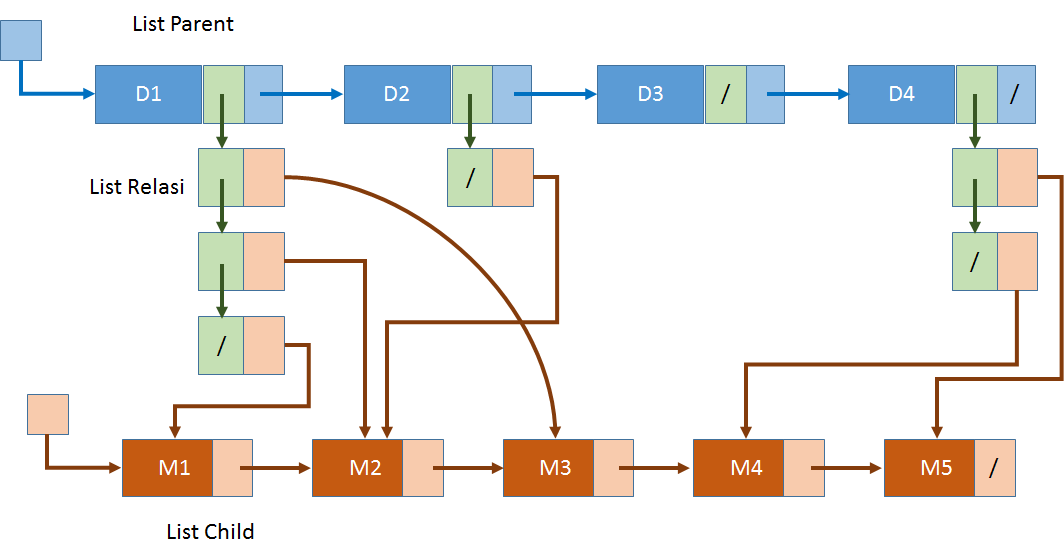
**BANDUNG**

**2018**

1. **Deskripsi Umum Aplikasi**

Aplikasi ini adalah aplikasi yang mengelola sebuah multi-linked list yang memodelkan *celebrity* dengan *follower* dan hubungan antara keduanya. Seorang *celebrity* bisa memiliki lebih dari satu *follower,* dan seorang *follower* bisa mengidolakan lebih dari satu *celebrity.*

Multi-linked list yang digunakan adalah relasi dari M-N. Dimana list relasi merupakan child list dari elemen parent.



1. **Struktur data dan ADT yang digunakan**
   1. **List parent**

* typedef string infotype\_celeb ;
* typedef struct elm\_celeb \*address\_celeb;
* struct elm\_celeb

infotype\_celeb info;

address\_celeb next;

address\_celeb prev;

List\_relasi followers;

* struct List\_celeb

address\_celeb first;

address\_celeb last;

* void createList\_celeb(List\_celeb &L);
* void insertFirst\_c(List\_celeb &L, address\_celeb P);
* void insertLast\_c(List\_celeb &L, address\_celeb P);
* void insertAfter\_c(address\_celeb Prec, address\_celeb P);
* void deleteFirst\_c(List\_celeb &L, address\_celeb &P);
* void deleteLast\_c(List\_celeb &L, address\_celeb &P);
* void deleteAfter\_c(List\_celeb &L,address\_celeb Prec, address\_celeb &P);
* address\_celeb alokasi\_c(string username);
* void dealokasi\_c(address\_celeb &P);
* address\_celeb findElm\_c (List\_celeb L, infotype\_celeb x);
* void printInfo\_c(List\_celeb L);
* void celebManyFolllower(List\_celeb L,List\_relasi R);
* void follower3celebrity(List\_followers F, List\_celeb C,List\_relasi LR);
* void sorting\_c(List\_celeb &L);
  1. **List relasi**
* typedef struct elm\_relasi \*address\_relasi;
* struct elm\_relasi

address\_relasi next;

address\_followers info;

* struct List\_relasi

address\_relasi first;

* void createList\_relation (List\_relasi &L);
* void insertFirst (List\_relasi &L, address\_relasi P);
* void insertLast (List\_relasi &L, address\_relasi P);
* void insertAfter (address\_relasi Prec, address\_relasi P);
* void deleteFirst (List\_relasi &L, address\_relasi &P);
* void deleteLast (List\_relasi &L, address\_relasi &P);
* void deleteAfter (address\_relasi Prec, address\_relasi &P);
* address\_relasi alokasi (address\_followers F);
* void dealokasi (address\_relasi &P);
* address\_relasi findElm (List\_relasi L, address\_followers F);
* void printInfo (List\_relasi L);
  1. **List child**
* typedef string infotype\_followers;
* typedef struct elm\_followers \*address\_followers;
* struct elm\_followers

infotype\_followers info;

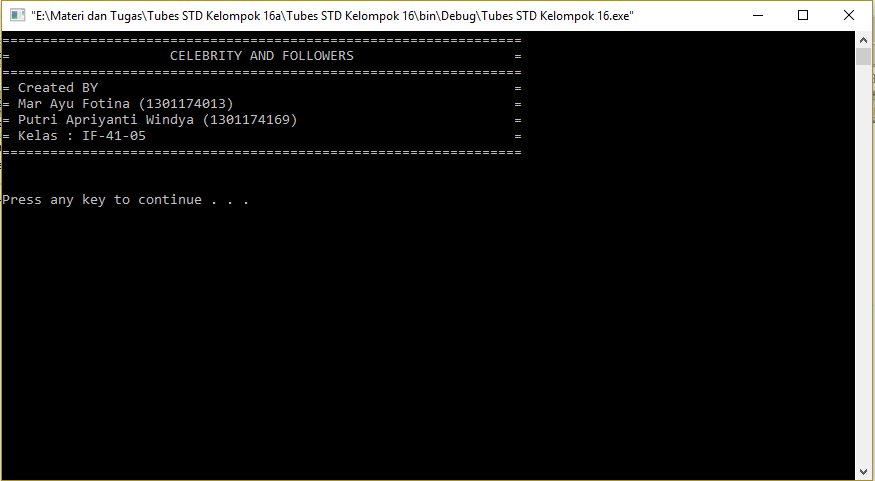
address\_followers next;

* struct List\_followers

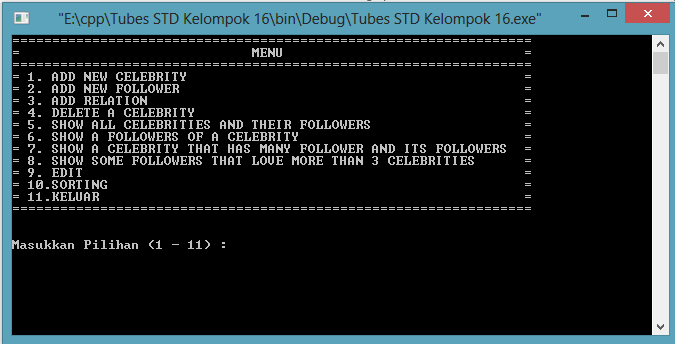
address\_followers first;

* void createList\_followers (List\_followers &L);
* address\_followers alokasi\_f (infotype\_followers x);
* void dealokasi\_f (address\_followers &P);
* void insertFirst\_f (List\_followers &L, address\_followers P);
* void insertLast\_f (List\_followers &L, address\_followers P);
* void insertAfter\_f (List\_followers &L,address\_followers Prec, address\_followers P);
* void deleteFirst\_f (List\_followers &L, address\_followers &P);
* void deleteLast\_f (List\_followers &L, address\_followers &P);
* void deleteAfter\_f (List\_followers &L,address\_followers Prec, address\_followers &P);
* address\_followers findElm\_f (List\_followers L, infotype\_followers x);
* void printInfo\_f(List\_followers L);
  1. **Kamus Global**
* void menu(int &pil);
* void edit(List\_celeb &C,List\_followers &F);
* void sorting(List\_celeb &C, List\_followers &F);

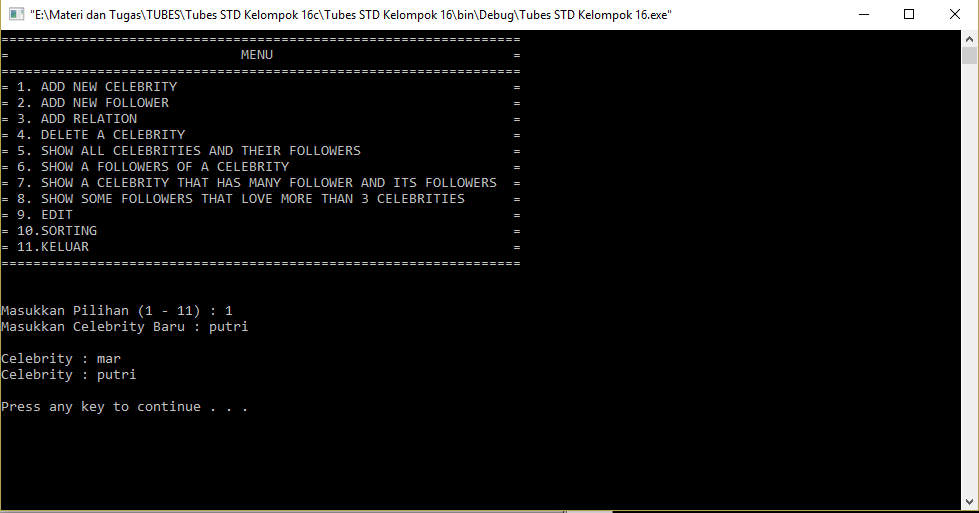
1. **Fungsional Aplikasi**
   1. Menambahkan *celebrity* baru
   2. Menambahkan *follower* baru
   3. Menambahkan hubungan antara *follower* dan seorang *celebrity*
   4. Menghapus *celebrity* tertentu
   5. Menampilkan semua *celebrity* dengan semua *follower-*nya
   6. Menampilkan *follower* yang dimiliki oleh *celebrity* tertentu
   7. Menampilkan *celebrity* yang memiliki *follower* paling banyak dan semua *follower*-nya
   8. Menampilkan *follower* yang mengidolakan lebih dari tiga *celebrity*
   9. Menampilkan menu edit yang dapat mengubah nama *celebrity* maupun *follower* yang diinginkan
   10. Mengurutkan data berdasarkan nama *celebrity* dan *follower* sesuai dengan urutan huruf alphabet
2. **Tampilan Aplikasi** 
   1. Halaman pembuka

****

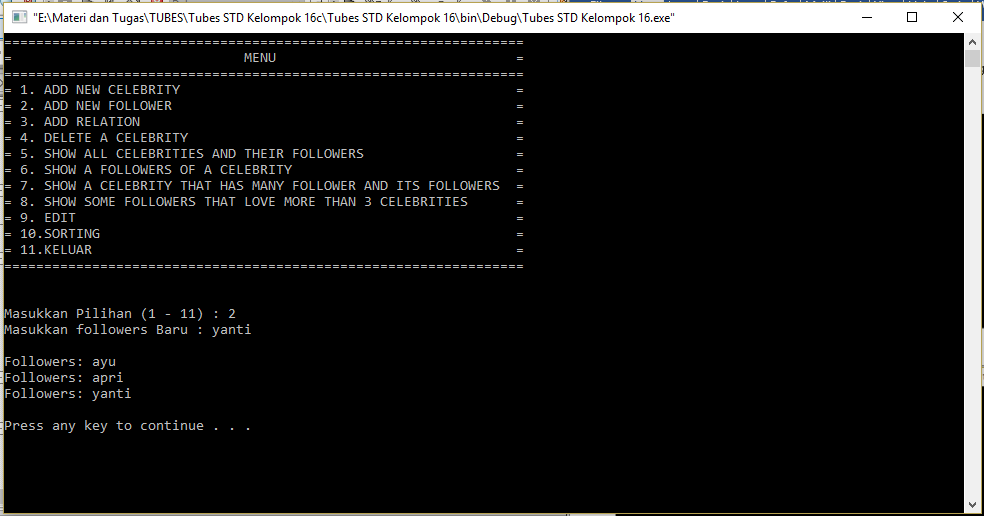
* 1. Menu Utama



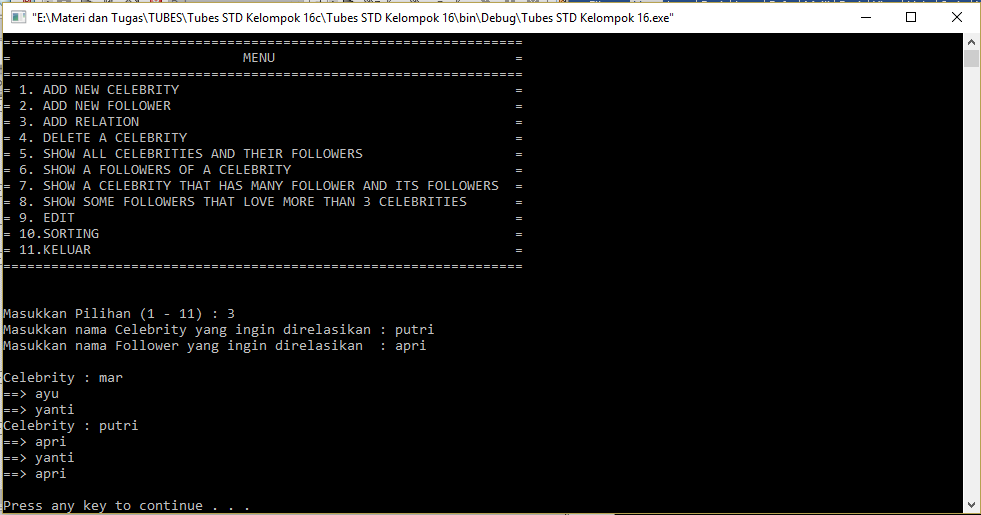
* 1. 1) Menu menambahkan *celebrity* baru



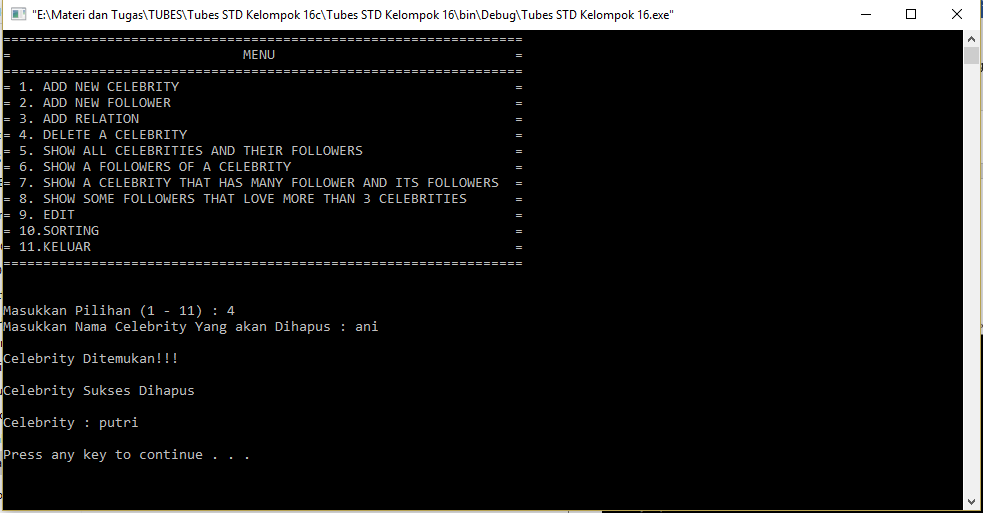
* 1. 2) Menu menambahkan *follower* baru



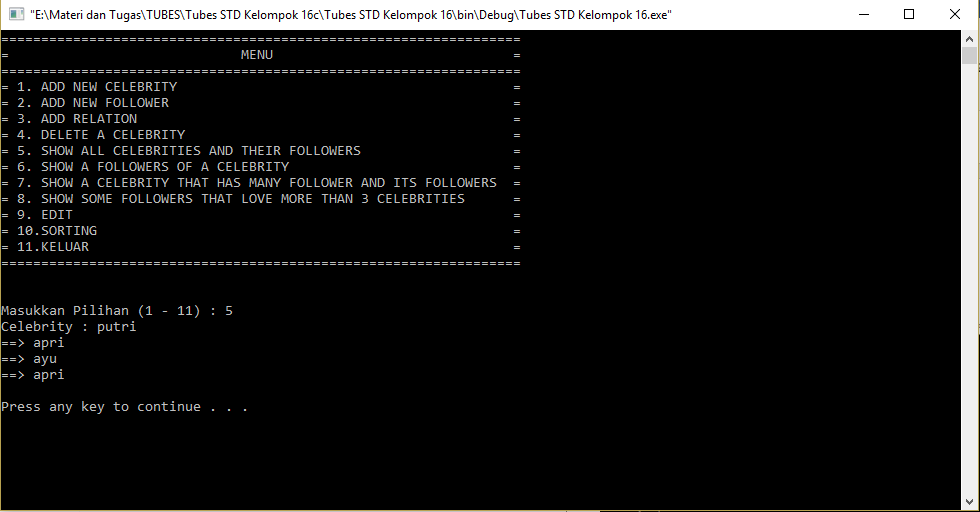
* 1. 3) Menu menambahkan hubungan antara *follower* dan seorang *celebrity*



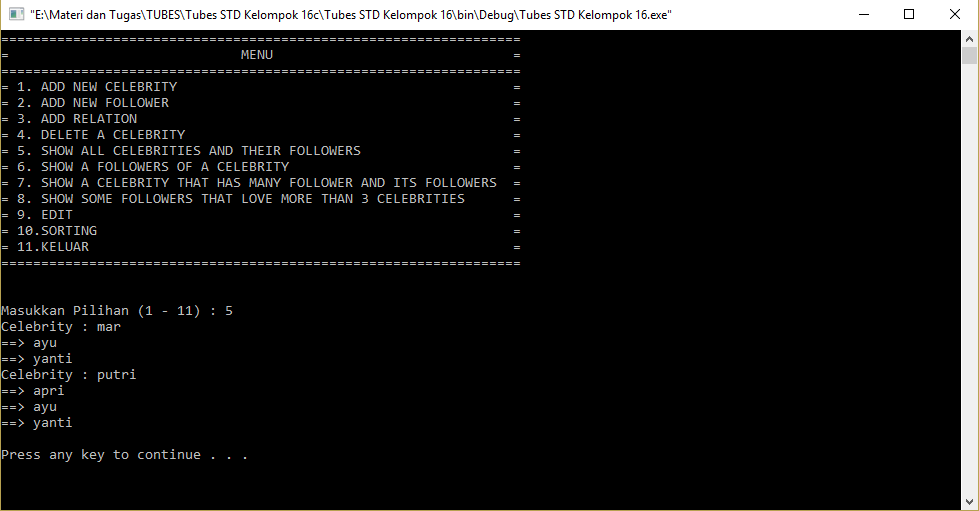
* 1. 4) Menu menghapus *celebrity* tertentu



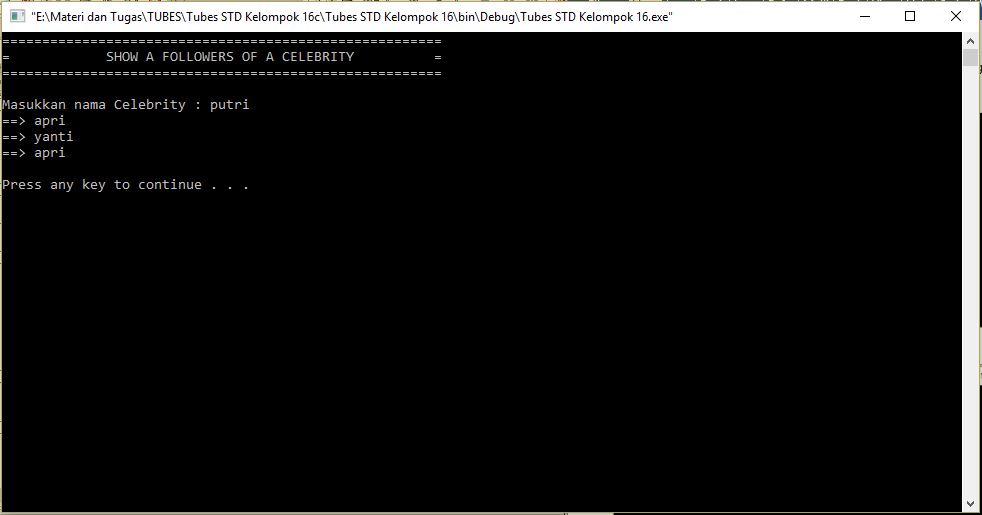
Setelah dihapus



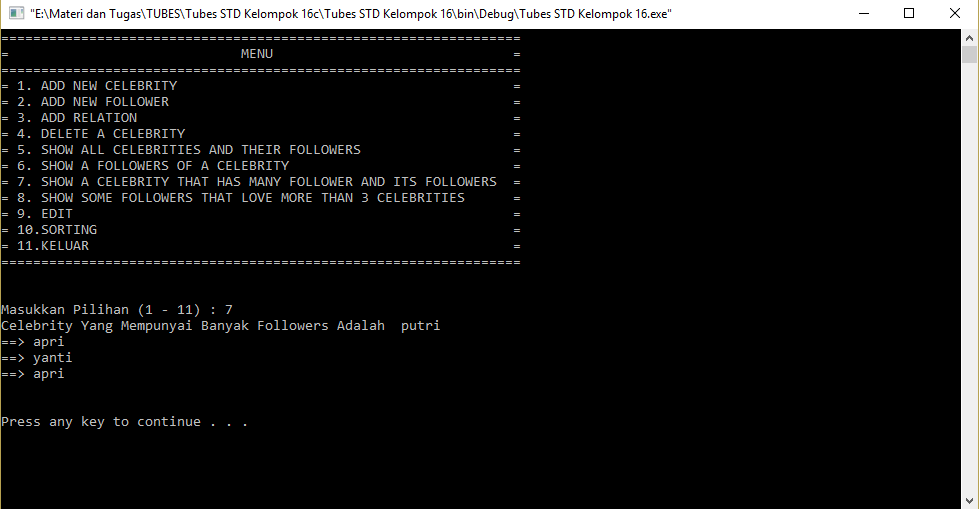
* 1. 5) Menu menampilkan semua *celebrity* dengan semua *follower-*nya

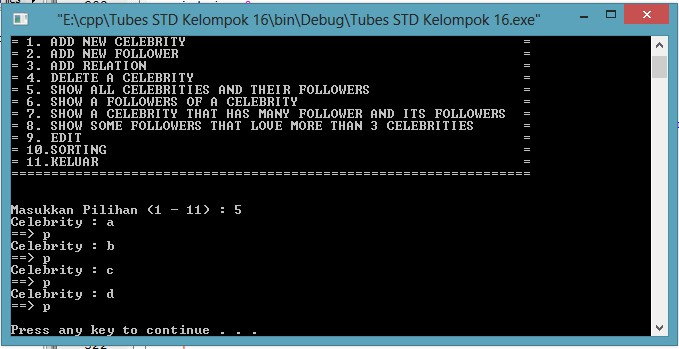


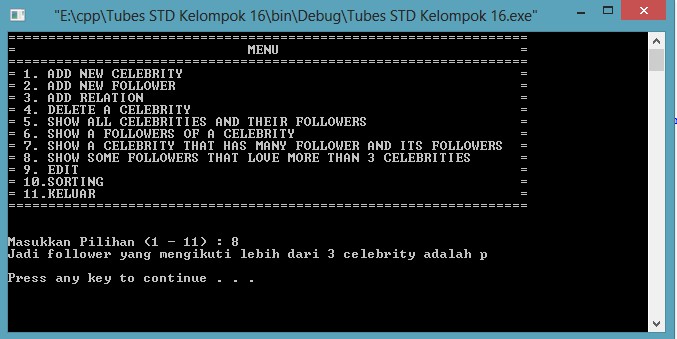
* 1. 6) Menu menampilkan *follower* yang dimiliki oleh *celebrity* tertentu



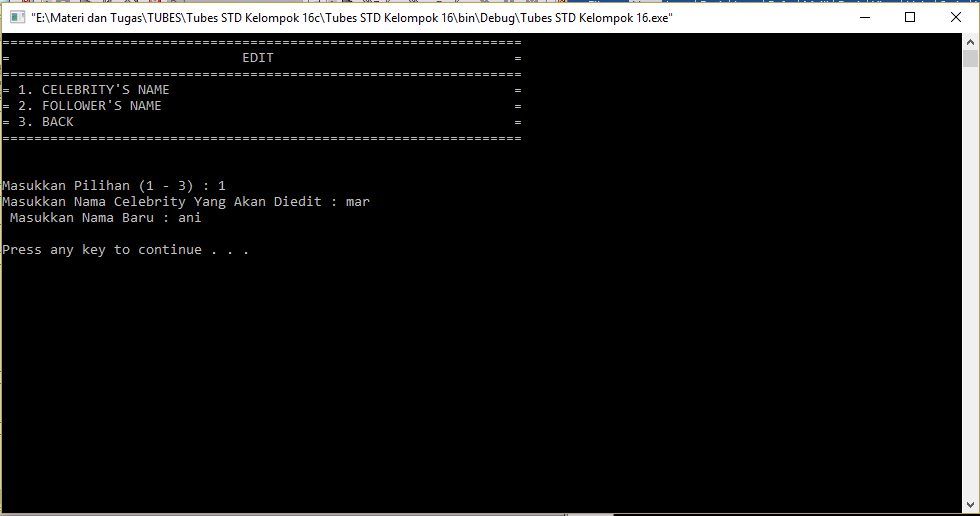
* 1. 7) Menu menampilkan *celebrity* yang memiliki *follower* paling banyak dan semua *follower*-nya



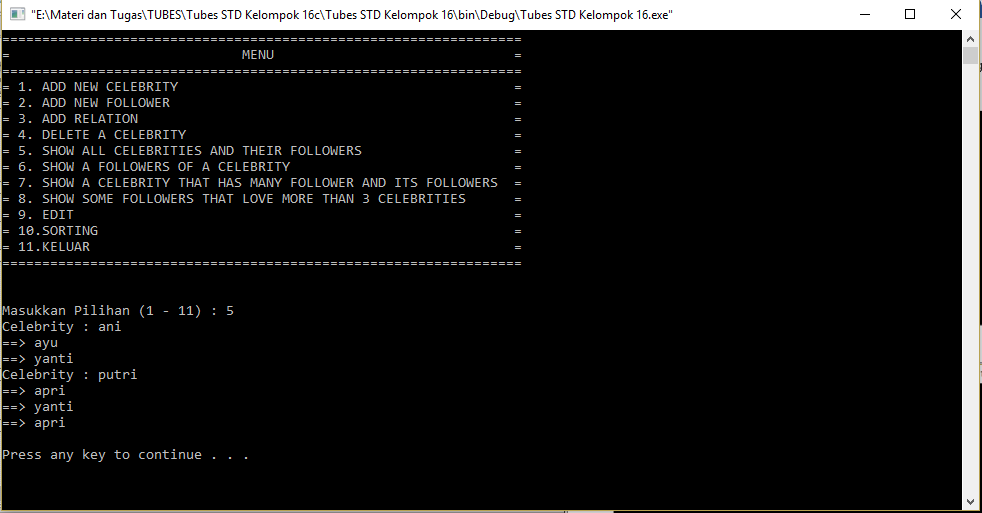
* 1. 8) Menu menampilkan *follower* yang mengidolakan lebih dari tiga *celebrity*



* 1. 9) Menu edit



Setelah di edit



* 1. Menu sorting

