Aplikasi pengolah data film

TUGAS UTS ALGORITMA DAN STRUKTUR DATA 2



Tugas ini untuk melengkapi nilai UTS pada Mata Kuliah ASDAT-2  
  
Nama : Hanif Ahmad Syauqi  
NIM : 10121161  
KELAS : IF-4 2021

**UNIVERSITAS KOMPUTER INDONESIA**

Daftar Isi

[A. Installasi 2](#_Toc105784693)

[B. Source Code 2](#_Toc105784694)

[1. Aplikasi 🡪 main.py 3](#_Toc105784695)

[2. Aplikasi 🡪 db 🡪 data.json 3](#_Toc105784696)

[3. Aplikasi 🡪 lib 🡪 doubleLinkedList.py 4](#_Toc105784697)

[C. Screen Capture 16](#_Toc105784698)

[1. Main Menu 16](#_Toc105784699)

[2. Tambah Data 17](#_Toc105784700)

[3. Edit Data 18](#_Toc105784701)

[4. Hapus Data 19](#_Toc105784702)

[5. Tampil Data 20](#_Toc105784703)

[6. Tentang Aplikasi dan Credits 21](#_Toc105784704)

[7. Keluar Aplikasi 21](#_Toc105784705)

# Installasi

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.

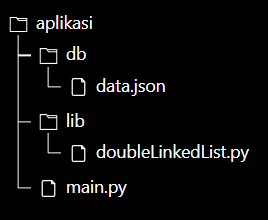
Untuk penginstallan silahkan unduh pada website python official <https://www.python.org/downloads/>

Lalu masukan perintah PIP berikut untuk menginstall module external

* pip install terminaltables colorama

# Source Code

Sebelum menulis source code saya kan memperlihatkan bagaimana struktur file dalam bentuk tree sebagai berikut



## Aplikasi 🡪 main.py

Ini adalah file utama untuk menjalankan aplikasi adna bisa menggunakan perintah berikut:

* Untuk windows
* py main.py
* Untuk Linux
* python3 main.py

Source code file ini adalah sebagai berikut :

*from* lib.doubleLinkedList *import* DoubleLinkedList

*if* \_\_name\_\_ == '\_\_main\_\_':

*try*:

        ls = DoubleLinkedList('./db/data.json')

        ls.menu()

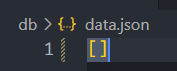
*except* KeyboardInterrupt:

        ls.keluar()

## Aplikasi 🡪 db 🡪 data.json

File ini bisa disebut juga dengan database yang berbentuk JSON, dimana ini akan menyimpan semua data yang telah dimanipulasi oleh LinkedList dan restore Kembali pada saat aplikasi di jalankan.

Cukup isikan file ini dengan array kosong saja seperti [], akan bertambah seiring waktu pada saat menambah data pada aplikasi.



## Aplikasi 🡪 lib 🡪 doubleLinkedList.py

Setiap aplikasi pasti mempunyai struktur algoritmanya tersendiri dimana hal tersebut mencaku semua fungsi-fungsi untuk jalannya aplikasi, file ini terdapat struktur dengan class doubleLinkedList dengan Node yang sebagian besar diisi dengan value dictionary, berikut ini adalah source code filenya :

*import* json, sys, os, re

*from* terminaltables *import* AsciiTable

*from* colorama *import* Fore, Style

class Node:

    def \_\_init\_\_(*self*, *info*):

*self*.info = *info*

*self*.prev = None

*self*.next = None

class DoubleLinkedList:

    def \_\_init\_\_(*self*, *path*):

*self*.awal = None

*self*.akhir = None

*self*.path = *path*

*self*.list = {}

*with* open(*self*.path) *as* json\_data:

            data = json.load(json\_data)

*for* obj *in* data:

*self*.addLast(obj)

*self*.list = data

    def clearConsole(*self*):

        os.system('cls' *if* os.name=='nt' *else* 'clear')

    def menu(*self*):

        def selectMenu():

            opsi = input(f'{Fore.YELLOW}\x1b[1m[ Pilih Salah Satu ]\x1b[0m{Fore.BLUE} >> {Style.RESET\_ALL}')

*while* opsi == '1' *or* opsi == '2' *or* opsi == '3' *or* opsi == '4' *or* opsi == '5' *or* opsi == '6':

*if* opsi == '1':

*self*.tambahData()

*break*

*elif* opsi == '2':

*self*.editData()

*break*

*elif* opsi == '3':

*self*.hapusData()

*break*

*elif* opsi == '4':

*self*.tampilData()

*break*

*elif* opsi == '5':

*self*.credit()

*break*

*elif* opsi == '6':

*self*.keluar()

*else*:

                print(f'''\n{Fore.RED}Opsi dengan '{opsi}' tidak valid, pilih salah satu angka di menu!{Style.RESET\_ALL}\n''')

                selectMenu()

        data = *self*.list

        stringMenu = f'''{Fore.WHITE}

# ============================================== #

#    {Fore.RED}        [  Aplikasi Data FILM  ]  {Fore.WHITE}          #

# ============================================== #

{Fore.YELLOW}\x1b[3mInfo : Terdapat {len(data)} film di database.\x1b[0m

{Fore.WHITE}1) {Fore.GREEN}Tambah data FILM

{Fore.WHITE}2) {Fore.GREEN}Edit data FILM

{Fore.WHITE}3) {Fore.GREEN}Hapus data FILM

{Fore.WHITE}4) {Fore.GREEN}Tampilkan FILM yang tersedia

{Fore.LIGHTBLACK\_EX}--------------------------------

{Fore.WHITE}5) {Fore.CYAN}Tentang Aplikasi (Credit)

{Fore.WHITE}6) {Fore.RED}Keluar

'''

*self*.clearConsole()

        print(stringMenu)

        selectMenu()

    def write2json(*self*, *data*):

*with* open(*self*.path, 'w') *as* f:

            json.dump(*data*, f, *indent*=3)

    def tambahData(*self*):

        def confirmAdd():

            apakah = input(Fore.GREEN + '\nApakah anda ingin menambahkan data lagi [Y/T]? ' + Style.RESET\_ALL)

*if* apakah.upper() == 'N' *or* apakah.upper() == 'T':

*self*.menu()

*elif* apakah.upper() == 'Y':

*self*.tambahData()

*else*:

                print(Fore.RED + f"\ninput dengan kata kunci '{apakah}' tidak valid!")

                confirmAdd()

*self*.clearConsole()

        print(Fore.LIGHTWHITE\_EX + '------------ [ TAMBAH DATA ] ------------\n')

        judul           = input(Fore.MAGENTA + 'Masukan judul         : ' + Style.RESET\_ALL)

        genre           = input(Fore.MAGENTA + 'Masukan genre         : ' + Style.RESET\_ALL)

        sutradara       = input(Fore.MAGENTA + 'Masukan sutradara     : ' + Style.RESET\_ALL)

        tanggal\_rilis   = input(Fore.MAGENTA + 'Masukan tanggal rilis : ' + Style.RESET\_ALL)

        local\_dict = {

                'no': *self*.size() + 1,

                'judul': judul,

                'genre': genre,

                'sutradara': sutradara,

                'tanggal\_rilis': tanggal\_rilis

            }

*self*.addLast(local\_dict)

*self*.list.append(local\_dict)

*self*.write2json(*self*.list)

*self*.clearConsole()

        print(f'''{Fore.LIGHTBLUE\_EX}Penambahan Data Berhasil!\n''')

        confirmAdd()

    def urutNomorFilm(*self*):

        list = []

*for* i *in* range(len(*self*.list)):

            list.append({

                'no': i+1,

                'judul': *self*.list[i]['judul'],

                'genre': *self*.list[i]['genre'],

                'sutradara': *self*.list[i]['sutradara'],

                'tanggal\_rilis': *self*.list[i]['tanggal\_rilis']

            })

*self*.list = list

    def editData(*self*):

        searchKeyword = input(Fore.MAGENTA + '\n[ EDIT DATA ] -> Masukan nomor/judul film : ' + Style.RESET\_ALL)

        index = 0

        intSearchKeyword = None

*if* re.match('^[-+]?[0-9]+$', searchKeyword):

            intSearchKeyword = int(searchKeyword)

*for* i *in* range(len(*self*.list)):

*if* intSearchKeyword == *self*.list[i]['no'] *or* searchKeyword.upper() *in* *self*.list[i]['judul'].upper():

                index = *self*.list[i]['no']

        def confirmUpdate():

            opsi = input(Fore.LIGHTBLUE\_EX + f'\nApakah anda ingin mengubah data ini [Y/T]? ' + Style.RESET\_ALL)

*if* opsi.upper() == 'N' *or* opsi.upper() == 'T':

*return* *self*.menu()

*elif* opsi.upper() == 'Y':

*pass*

*else*:

                print(Fore.RED + f"\ninput dengan kata kunci '{opsi}' tidak valid!")

                confirmUpdate()

        def confirmDataUpdate():

            opsi = input(Fore.LIGHTBLUE\_EX + f'\nTerapkan perubahan [Y/T]? ' + Style.RESET\_ALL)

*if* opsi.upper() == 'N' *or* opsi.upper() == 'T':

*self*.clearConsole()

                print(Fore.RED + f'Perubahan data dibatalkan!')

                input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

*elif* opsi.upper() == 'Y':

*pass*

*else*:

                print(Fore.RED + f"\ninput dengan kata kunci '{opsi}' tidak valid!")

                confirmDataUpdate()

        def successMessage():

*self*.urutNomorFilm()

*self*.write2json(*self*.list)

*self*.clearConsole()

            print(Fore.CYAN + f'Sukses edit data FILM dengan judul {*self*.list[index - 1]["judul"]}!')

            input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

*if* index != 0:

            table\_field = ['No','Judul', 'Genre', 'Sutradara', 'Tanggal Rilis']

            table\_data\_before = [

                table\_field

            ]

            table\_data\_before.append(

                [*self*.list[index - 1]['no'], *self*.list[index - 1]['judul'], *self*.list[index - 1]['genre'], *self*.list[index - 1]['sutradara'], *self*.list[index - 1]['tanggal\_rilis']]

            )

            table = AsciiTable(table\_data\_before, 'Data sebelum update')

            print(Fore.YELLOW + '\n' + table.table)

            confirmUpdate()

            print(Fore.WHITE + '\n------------ [ EDIT DATA ] ------------\n')

            judul           = input(Fore.MAGENTA + 'Masukan judul         : ' + Style.RESET\_ALL)

            genre           = input(Fore.MAGENTA + 'Masukan genre         : ' + Style.RESET\_ALL)

            sutradara       = input(Fore.MAGENTA + 'Masukan sutradara     : ' + Style.RESET\_ALL)

            tanggal\_rilis   = input(Fore.MAGENTA + 'Masukan tanggal rilis : ' + Style.RESET\_ALL)

            local\_dict = {

                    'no': index,

                    'judul': judul,

                    'genre': genre,

                    'sutradara': sutradara,

                    'tanggal\_rilis': tanggal\_rilis

                }

            table\_data\_after = [

                table\_field

            ]

            table\_data\_after.append(

                    [local\_dict['no'], local\_dict['judul'], local\_dict['genre'], local\_dict['sutradara'], local\_dict['tanggal\_rilis']]

            )

            table = AsciiTable(table\_data\_after, 'Data setelah update')

            print(Fore.YELLOW + '\n' + table.table)

            confirmDataUpdate()

*self*.list[index - 1] = local\_dict

            successMessage()

*else*:

*self*.clearConsole()

            print(Fore.RED + f"\nData dengan kata kunci '{searchKeyword}' tidak ditemukan!")

            input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

    def hapusData(*self*):

        searchKeyword = input(Fore.MAGENTA + '\n[ HAPUS DATA ] -> Masukan nomor/judul film : ' + Style.RESET\_ALL)

        intSearchKeyword = None

        index = 0

*if* re.match('^[-+]?[0-9]+$', searchKeyword):

            intSearchKeyword = int(searchKeyword)

*for* i *in* range(len(*self*.list)):

*if* intSearchKeyword == *self*.list[i]['no'] *or* searchKeyword.upper() *in* *self*.list[i]['judul'].upper():

                index = *self*.list[i]['no']

        temp\_list = *self*.list[len(*self*.list) - 1]

        def showTable(*data*):

            table\_data = [

                ['No','Judul', 'Genre', 'Sutradara', 'Tanggal Rilis']

            ]

            table\_data.append(

                [*data*['no'], *data*['judul'], *data*['genre'], *data*['sutradara'], *data*['tanggal\_rilis']]

            )

            table = AsciiTable(table\_data)

            print(Fore.YELLOW + '\n' + table.table)

        def confirmDelete():

            opsi = input(Fore.LIGHTBLUE\_EX + f'\nApakah anda ingin menghapus data ini [Y/T]? ' + Style.RESET\_ALL)

*if* opsi.upper() == 'N' *or* opsi.upper() == 'T':

*return* *self*.menu()

*elif* opsi.upper() == 'Y':

*pass*

*else*:

                print(Fore.RED + f"\ninput dengan kata kunci '{opsi}' tidak valid!")

                confirmDelete()

        def successMessage():

*self*.urutNomorFilm()

*self*.write2json(*self*.list)

*self*.clearConsole()

            print(Fore.CYAN + f'Sukses hapus data FILM dengan judul {temp\_list["judul"]}!')

            input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

*if* index != 0:

*if* index == 1:

                showTable(*self*.list[index - 1])

                confirmDelete()

*self*.removeFirst()

*self*.list.pop()

                successMessage()

*elif* index == *self*.size():

                showTable(*self*.list[index - 1])

                confirmDelete()

*self*.removeLast()

*self*.list.pop()

                successMessage()

*else*:

                showTable(*self*.list[index - 1])

                confirmDelete()

                removedNode = *self*.get(index)

*if* removedNode *is* *not* None:

                    removedNode.prev.next = removedNode.next

                    removedNode.next.prev = removedNode.prev

*del* removedNode

*self*.list.pop(index - 1)

                    successMessage()

*else*:

*self*.clearConsole()

                    print(Fore.RED + f"Hapus data gagal. Nomor {index} tidak ditemukan!")

                    input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

*else*:

*self*.clearConsole()

            print(Fore.RED + f"\nData dengan kata kunci '{searchKeyword}' tidak ditemukan!")

            input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

    def dictPage(*self*, *page*, *list*):

        result = []

        dictSeparate = [*list*[i:i+10] *for* i *in* range(0, len(*list*), 10)]

        listLength = len(dictSeparate)

        index = *page* - 1

*if* index < listLength *and* index >= 0:

            result = dictSeparate[index]

*elif* index < 0:

            result = []

*return* result

    def tampilData(*self*, *page*=1):

        table\_data = [

            ['No','Judul', 'Genre', 'Sutradara', 'Tanggal Rilis']

        ]

        data = *self*.dictPage(*page*, *self*.list)

*for* i *in* range(len(data)):

            table\_data.append(

                [data[i]['no'], data[i]['judul'], data[i]['genre'], data[i]['sutradara'], data[i]['tanggal\_rilis']]

            )

*self*.clearConsole()

        allPage = len([*self*.list[i:i+10] *for* i *in* range(0, len(*self*.list), 10)])

        judul = f"[ List Film - Page {*page*}/{allPage} ]"

        table = AsciiTable(table\_data, judul)

        print(Fore.YELLOW + '\n' + table.table)

        pageid = input(f'''\n\n{Fore.BLUE}[ TAMPIL DATA ] Masukan No Page / Enter untuk kembali : {Style.RESET\_ALL}''')

*if* re.match('^[-+]?[0-9]+$', pageid):

*self*.tampilData(int(pageid))

*pass*

*else*:

*self*.menu()

    def credit(*self*):

*self*.clearConsole()

        print(f'''{Fore.RED}

# ============================================== #

#            {Fore.LIGHTWHITE\_EX}\033[1m[  Tentang Aplikasi  ]\x1b[0m{Fore.RED}              #

#      {Fore.LIGHTWHITE\_EX}\x1b[3mLoving yourself will save your soul\x1b[0m{Fore.RED}       #

# ============================================== #

#                                                #

#    {Fore.CYAN}Dibuat Oleh Hanif Ahmad Syauqi - 1012161{Fore.RED}    #

{Fore.WHITE}#          {Fore.GREEN}Teknik Informatika 4 / 2021 {Fore.WHITE}          #

#                                                #

#        {Fore.LIGHTMAGENTA\_EX}github.com/MRHRTZ/Crud-Data-Film{Fore.WHITE}        #

#              {Fore.YELLOW}wa.me/6285559038021{Fore.WHITE}               #

#                                                #

# ============================================== # {Style.RESET\_ALL}''')

        input(Fore.GREEN + '\n\n[ Kembali ]')

*self*.menu()

    def keluar(*self*):

        print(Fore.YELLOW + '\n\nBye 🤞\n' + Style.RESET\_ALL)

        sys.exit(0)

    def isEmpty(*self*):

*return* *self*.awal *is* None

    def size(*self*):

*if* *self*.isEmpty():

            banyakNode = 0

*else*:

            bantu = *self*.awal

            banyakNode = 1

*while* bantu.next *is* *not* None:

                banyakNode = banyakNode + 1

                bantu = bantu.next

*return* banyakNode

    def getFirst(*self*):

*return* *self*.awal

    def getLast(*self*):

*return* *self*.akhir

    def get(*self*, *index*):

*if* *self*.isEmpty() *or* *index*<1 *or* *index*>*self*.size():

*return* None

*else*:

            bantu = *self*.awal

            posisi = 1

*while* posisi<*index*:

                bantu = bantu.next

                posisi = posisi + 1

*return* bantu

    def addLast(*self*, *info*):

        newNode = Node(*info*)

*if* *self*.isEmpty():

*self*.awal = newNode

*self*.akhir = newNode

*else*:

*self*.akhir.next = newNode

            newNode.prev = *self*.akhir

*self*.akhir = newNode

    def removeFirst(*self*):

*if* *self*.awal == *self*.akhir:

            removedNode = *self*.awal

*self*.awal = None

*self*.akhir = None

*del* removedNode

*else*:

            removedNode = *self*.awal

*self*.awal = *self*.awal.next

*self*.awal.prev = None

*del* removedNode

    def removeLast(*self*):

*if* *self*.awal == *self*.akhir:

            removedNode = *self*.awal

*self*.awal = None

*self*.akhir = None

*del* removedNode

*else*:

            removedNode = *self*.akhir

*self*.akhir = *self*.akhir.prev

*self*.akhir.next = None

*del* removedNode

    def remove(*self*, *index*):

*if* *index*==1:

*self*.removeFirst()

*elif* *index*==*self*.size():

*self*.removeLast()

*else*:

            removedNode = *self*.get(*index*)

*if* removedNode *is* *not* None:

                removedNode.prev.next = removedNode.next

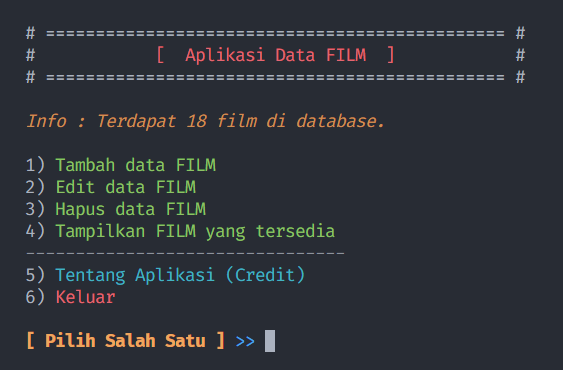
                removedNode.next.prev = removedNode.prev

*del* removedNode

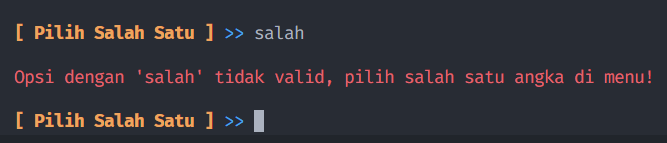
# Screen Capture

Berikut ini adalah screen capture dari aplikasi pengolah data FILM.

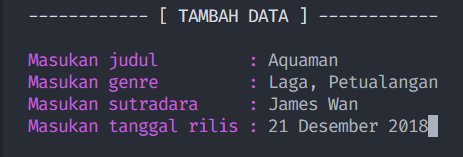
## Main Menu



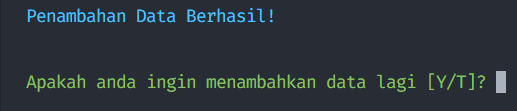
Tulisan yang muncul ketika opsi pilihan salah



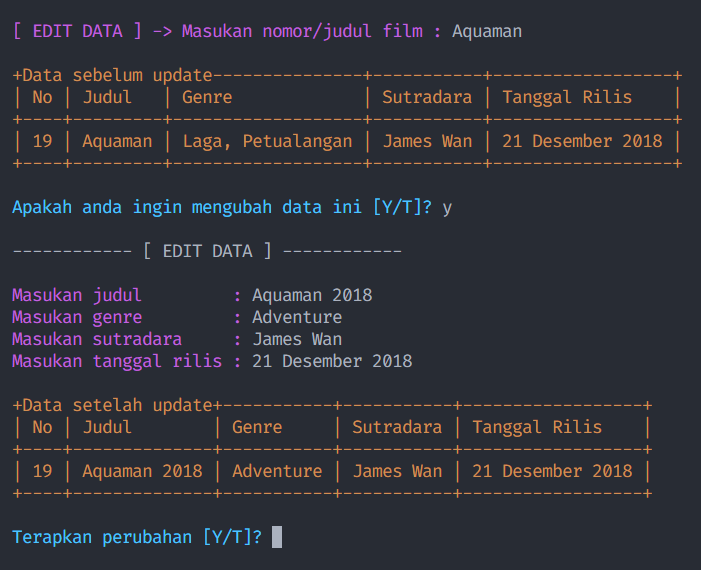
## Tambah Data



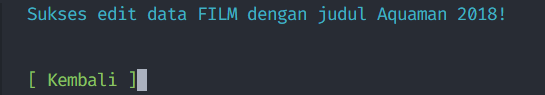
Screen capture penambahan data berhasil



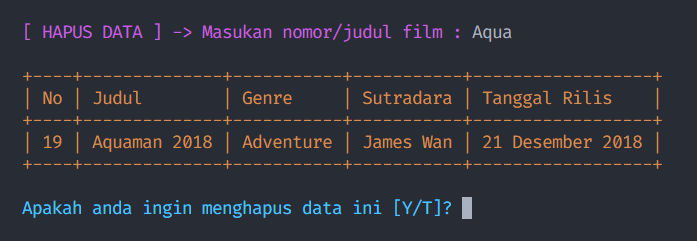
## Edit Data



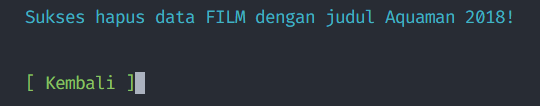
Screen capture setelah edit data berhasil



## Hapus Data



Screen capture setelah berhasil penghapusan data

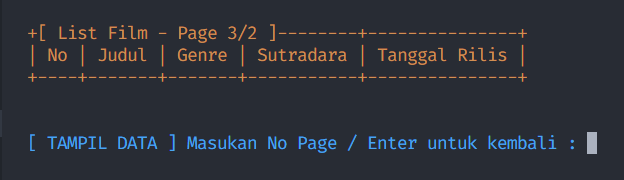


## Tampil Data

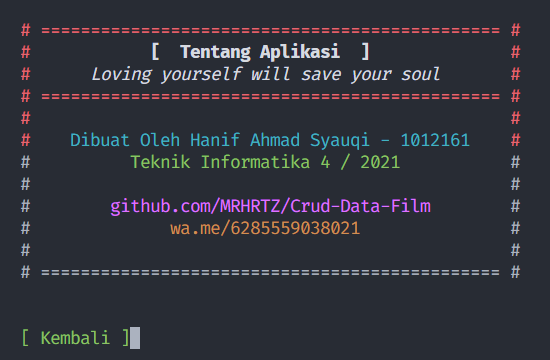
Pada menu tampil data, tabel akan dibatasi 10 baris per halaman, anda bisa memilih halaman/page dengan ketik angka pada console dan bisa Kembali ke menu hanya ketik enter, apabila halaman tidak valid maka tabel akan menampilkan data kosong.







## Tentang Aplikasi dan Credits



## Keluar Aplikasi

