

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



DATA CHIPSET APPLE BIONIC

Oleh:

Kelompok 4 B2 2024

Jovan Bagas Pangestu	2409106071
Muhammad Fathir Rahman	2409106075
Ezar Reinhart Banne	2409106086

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Alhamdulillah, puji dan syukur kami panjatkan ke hadirat Allah SWT Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan ini dengan baik. Laporan ini disusun sebagai bagian dari tugas untuk memahami dan mengimplementasikan konsep CRUD (Create, Read, Update, Delete) pada pemrograman, dengan topik yang berkaitan dengan pengelolaan data chipset Apple Bionic. Tujuan penyusunan laporan ini adalah untuk mendokumentasikan hasil kerja dan proses pembuatan program yang telah kami buat.

Program yang kami susun menggunakan struktur data dictionary untuk menyimpan data chipset Apple Bionic dari A11 hingga A16. Fitur utama dari program ini adalah kemampuan untuk menambah, melihat, mengubah, dan menghapus data chipset. Selain itu, program juga dilengkapi dengan sistem login yang membatasi percobaan login hingga tiga kali, sehingga lebih aman dan terkendali. Program ini ditujukan untuk memberikan pemahaman lebih dalam tentang pengelolaan data berbasis CRUD, sekaligus meningkatkan kemampuan logika pemrograman kami.

Selama proses penyusunan dan pembuatan program ini, kami menghadapi beberapa kendala, seperti kesulitan dalam menangani validasi input dan pengaturan logika sistem login. Namun, dengan bantuan dari berbagai pihak, kami berhasil menyelesaikan program dan laporan ini. Ucapan terima kasih yang sebesar-besarnya kami sampaikan kepada Bang Tito, selaku Asisten Laboratorium, yang telah memberikan bimbingan selama proses pengerjaan. Selain itu, kami juga mengucapkan terima kasih kepada seluruh anggota Kelompok 4, yang telah bekerja sama dengan baik dalam menyelesaikan tugas ini.

Kami berharap laporan ini dapat bermanfaat bagi pembaca dalam memahami konsep CRUD pada pemrograman. Segala kritik dan saran yang membangun sangat kami harapkan untuk menyempurnakan karya kami di masa yang akan datang.

Samarinda, 22 November 2024

Kelompok 4 B2 2024

TAKARIR

<i>Database</i>	Basis Data
<i>Algorithm</i>	Algoritma
<i>Management</i>	Mengelola
<i>Input</i>	Memasukkan
<i>Output</i>	Keluaran
<i>Login</i>	Masuk
<i>Admin</i>	Pengelola
<i>User</i>	Pengguna
<i>Data</i>	Data
<i>Create</i>	Membuat
<i>Read</i>	Membaca
<i>Update</i>	Memperbarui
<i>Delete</i>	Menghapus
<i>Statistik</i>	Statistik
<i>Core</i>	Inti
<i>Process Technology</i>	Teknologi Proses
<i>Loop</i>	Perulangan
<i>Exception Handling</i>	Penanganan Kesalahan
<i>Validation</i>	Validasi
<i>Recursive</i>	Rekursif
<i>Interface</i>	Antarmuka
<i>Backup</i>	Cadangan
<i>Security</i>	Keamanan
<i>Efficiency</i>	Efisiensi
<i>User Interface</i>	Antarmuka Pengguna
<i>Role-Based Access</i>	Akses Berdasarkan Peran

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR	iv
DAFTAR ISI	v
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional	1
1.3 Rumusan Masalah	2
1.4 Batasan Masalah	2
1.5 Tujuan	3
BAB II PERANCANGAN	4
2.1 Analisis Program	4
2.2 Flowchart	5
2.3 Konsep/Materi Praktikum yang dipakai	7
BAB III HASIL DAN PEMBAHASAN	11
3.1 Tampilan Program	11
1. Tampilan Menu	11
2. Menu register.	12
2. Menu pengelolaan chipset.	13
3.2 Source Code	14

BAB IV PENUTUP	21
4.1 Kesimpulan.....	21
4.2 Saran	21
DAFTAR PUSTAKA	23
LAMPIRAN.....	24

DAFTAR GAMBAR

Gambar 1 Flowchart.
Gambar 2 Tampilan Menu.
Gambar 3 Menu Register.....
Gambar 4 Menu Pengelolaan Chipset.....
Gambar 5 Kertas Konsul.....

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam konteks perkembangan teknologi, data chipset seperti tahun rilis, jumlah inti (cores), dan teknologi proses menjadi penting untuk dikelola secara sistematis. Data ini sering kali digunakan oleh pengembang, perusahaan, atau penggemar teknologi untuk perbandingan atau analisis.

Mengelola data chipset secara manual (misalnya menggunakan spreadsheet atau pencatatan tertulis) memakan waktu dan rawan kesalahan. Oleh karena itu, diperlukan sistem terpusat yang dapat memfasilitasi pengelolaan data secara lebih efisien.

Dalam pengelolaan data, diperlukan pembagian peran seperti ADMIN (dengan hak akses penuh untuk menambah, mengubah, atau menghapus data) dan USER (hanya memiliki hak melihat data). Sistem berbasis peran ini meningkatkan keamanan dan keteraturan dalam pengelolaan data.

1.2 Kebutuhan Fungsional

Program ini memiliki sistem login untuk memastikan bahwa username dan password yang dimasukkan diverifikasi dengan benar. Pengguna diberikan hingga tiga kali kesempatan login, dan jika gagal, akses akan dihentikan secara otomatis. Program membedakan hak akses antara admin dan user, di mana admin memiliki kemampuan penuh untuk mengelola data, sedangkan user hanya dapat melihat data chipset.

Fungsi utama dari program adalah implementasi CRUD (Create, Read, Update, Delete) untuk data chipset Apple Bionic. Admin dapat menambahkan informasi baru seperti nama chipset, tahun rilis, jumlah core, dan teknologi proses. Selain itu, admin juga dapat memperbarui data yang ada atau menghapus data berdasarkan nama

chipset tertentu. Informasi chipset yang tersimpan dapat dilihat oleh semua pengguna, dengan data ditampilkan secara terstruktur menggunakan fungsi rekursif.

Selain fitur pengelolaan data, program menyediakan fungsi statistik untuk menghitung total jumlah chipset yang tersimpan. Ada pula opsi logout untuk kembali ke menu login atau keluar dari program sepenuhnya. Antarmuka berbasis teks dirancang sederhana untuk mempermudah interaksi pengguna, dilengkapi dengan validasi input guna meminimalkan kesalahan.

Melalui fitur-fitur ini, program menawarkan solusi yang aman, efisien, dan mudah digunakan untuk mengelola data chipset, sekaligus mendukung tujuan praktikum dengan implementasi konsep pemrograman dasar.

1.3 Rumusan Masalah

1. Bagaimana cara menggunakan struktur data dictionary dan fungsi rekursif untuk menyimpan dan menampilkan data chipset secara efisien dan terorganisir?
2. Bagaimana membatasi percobaan login hingga tiga kali untuk meningkatkan keamanan program?
3. Bagaimana memastikan hanya admin yang memiliki akses penuh untuk mengelola data, sementara selain admin hanya bisa melihat saja?

1.4 Batasan Masalah

1. Seri yang digunakan hanya apple bionic A11 hingga A16
2. Program hanya menyimpan data chipset selama program berjalan . Data akan hilang jika program ditutup.
3. Program hanya memberikan pesan sederhana jika terjadi kesalahan input atau login gagal. Tidak ada notifikasi lanjut untuk membantu pengguna.
4. Fungsi rekursif digunakan hanya menampilkan data satu per satu atau tidak kompleks.

5. Program ini dibuat sederhana dan tidak bisa kompleks seperti menampilkan grafik.

1.5 Tujuan

1. Dapat meningkatkan keamanan akses dengan melakukan pembatasan login sebanyak 3 kali.
2. Membuat data lebih terstruktur dan mudah digunakan.
3. Belajar mengelola system sederhana yang berbasis CRUD
4. Mampu menggunakan dictionary sebagai tempat untuk menyimpan data

BAB II

PERANCANGAN

2.1 Analisis Program

Program ini adalah implementasi sederhana dari sistem CRUD (Create, Read, Update, Delete) menggunakan bahasa Python untuk mengelola data chipset Apple Bionic. Berikut adalah alur programnya:

1. Sistem Login

- Program dimulai dengan meminta pengguna untuk login.
- Username dan password diverifikasi dengan data pada dictionary akun.
- Pengguna diberikan maksimal 3 kali percobaan untuk login. Jika salah tiga kali, program akan berhenti.
- Setelah login berhasil, pengguna diarahkan ke menu utama berdasarkan perannya (Admin atau User).

2. Menu Utama

Menu utama memberikan empat pilihan:

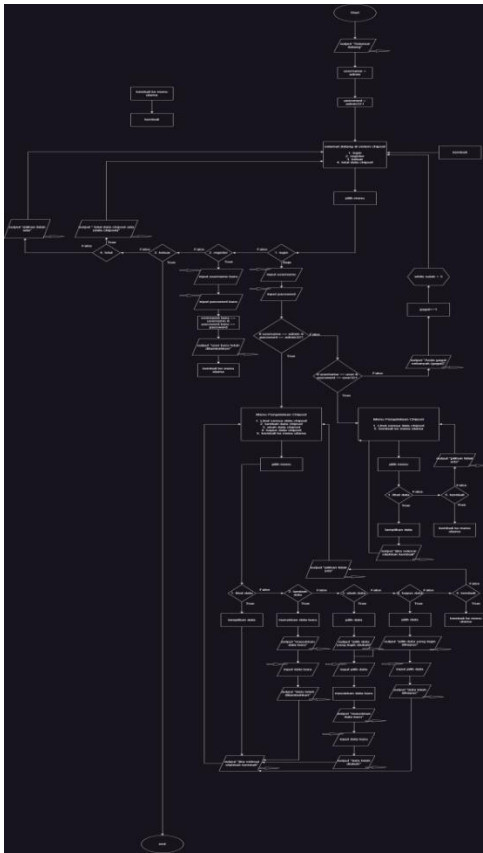
1. Lihat Data Chipset: Menampilkan seluruh data chipset yang disimpan dalam dictionary chipset_data secara rekursif.
2. Tambah Data Chipset: Admin dapat menambah data baru dengan memasukkan nama chipset, tahun rilis, jumlah core, dan proses fabrikasi.
3. Ubah Data Chipset: Admin dapat mengganti informasi chipset yang sudah ada.
4. Hapus Data Chipset: Admin dapat menghapus chipset dari data.

5. Logout: Keluar dari menu admin atau user dan kembali ke menu login.

3. Sistem Logout

Pengguna dapat keluar dari sistem kapan saja melalui opsi "Logout" atau dengan menutup program.

2.2 Flowchart



Gambar 1 Flowchart

Lampirkan flowchart dan jelaskan alurnya.

Pada flowchart ini kita dapat melihat alur yang bakalan ada di kode python nanti, alurnya kita mulai dengan start yang disambut dengan kata selamat datang,

dimenu utama kita diberi 4 pilihan yaitu, login, register, keluar dan total data chipset

1. login sendiri bisa kita lakukan dengan menggunakan akun admin atau akun user yang dibuat saat melakukan register
2. register sendiri adalah program untuk kita membuat akun yang statusnya sebagai user
3. keluar, sesuai namanya pilihan ini adalah pilihan untuk kita keluar dari program saat ini
4. total data chipset, kita bisa melihat total chipset yang ada di data saat ini

Jika kita selesai login maka disitu akan ada 5 pilihan yang merupakan program utama yang terdiri dari lihat data, tambah data, edit data, hapus data, dan kembali.

untuk keterangan kelimanya sebagai berikut

1. apabila kita memilih untuk melihat data maka kita akan dilihatkan data yang sudah ada, apabila data kosong maka akan bertuliskan data kosong
2. apabila kita memilih untuk menambahkan data, maka kita akan disuruh memasukkan data data yang dibutuhkan untuk menambahkan 1 data baru
3. edit sendiri bisa kita lakukan dengan memilih salah satu data yang sudah ada, kemudian kita disuruh untuk memperbarui atau mengubah apa yang ingin kita ganti
4. hapus data sendiri adalah kode untuk menghapus data chipset yang kita pilih
5. pilihan ini akan kembali ke menu utama yang terdapat opsi login, register, keluar dan lihat data

Ketika kita sudah selesai melakukan hal hal yang ingin kita tambah, lihat, ubah, atau hapus, pastinya kita akan kembali ke menu utama jika ingin berhenti dari program ini yang ada dipilihan keluar di menu utama.

Untuk 5 pilihan tadi hanya bisa dilakukan oleh admin. untuk user sendiri hanya bisa melihat data dan kembali.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

1. Fungsi Dasar:

Fungsi print

Fungsi ini digunakan untuk menampilkan teks dan informasi pada layar pengguna, seperti menu utama, hasil login, dan data chipset. Fungsi ini menjadi media komunikasi utama antara program dan pengguna.

Fungsi input

Digunakan untuk menerima input dari pengguna, seperti username, password, dan informasi baru terkait chipset. Fungsi ini memungkinkan interaksi pengguna dengan program secara dinamis.

2. Struktur Data

Dictionary

Program ini memanfaatkan dictionary sebagai struktur data utama untuk menyimpan data chipset. Nama chipset digunakan sebagai key, sedangkan detail

seperti tahun rilis, jumlah core, dan proses fabrikasi disimpan sebagai value dalam dictionary bersarang. Hal ini memudahkan pengelolaan data secara sistematis.

List

Dalam fungsi rekursif, list digunakan untuk menampung data sementara yang diekstrak dari dictionary untuk ditampilkan secara berurutan.

3. Fungsi Rekursif:

Fungsi menampilkan rekursif

Fungsi ini digunakan untuk menampilkan data chipset satu per satu. Fungsi ini memanfaatkan konsep rekursi, di mana fungsi memanggil dirinya sendiri hingga seluruh data dalam list chipset ditampilkan. Rekursi mempermudah pengolahan iterasi secara elegan dalam kondisi tertentu.

4. Percabangan

If-Else

Percabangan ini digunakan di berbagai bagian program, seperti:

- Memeriksa kecocokan username dan password saat login.
- Mengarahkan pengguna ke menu admin jika login berhasil.
- Memastikan validasi input saat menambah atau mengubah data chipset.

5. Perulangan

While Loop

Perulangan ini digunakan untuk:

- Membatasi percobaan login hingga maksimal 3 kali.
- Mengulangi tampilan menu utama hingga pengguna memilih untuk keluar.

- Memberikan kontrol dinamis kepada pengguna dalam pengelolaan data chipset.

6. Error Handling

Validasi Input

Program secara eksplisit memeriksa input dari pengguna, seperti memastikan username dan password sesuai dengan data yang ada. Selain itu, validasi dilakukan untuk menghindari penambahan data yang duplikat, seperti menambah chipset yang sudah ada.

7. Variabel Global dan Lokal:

Variabel Global

- akun: Menyimpan informasi username, password, dan peran pengguna (admin atau user).
- chipset_data: Menyimpan seluruh data chipset Apple Bionic.

Variabel Lokal

Variabel lokal digunakan di dalam fungsi atau prosedur, seperti:

- username dan password: Untuk menyimpan input pengguna.
- nama_chipset, tahun_rilis, cores, dan proses: Untuk menyimpan data baru atau perubahan terkait chipset.
- pilihanadmin: Untuk menangkap pilihan menu dari admin.

8. Prosedur dan Fungsi:

Fungsi proses_login

Fungsi ini memeriksa kecocokan antara username dan password yang dimasukkan pengguna dengan data di akun. Fungsi ini juga bertanggung jawab mengarahkan pengguna ke menu admin jika login berhasil.

Prosedur manage_chipset_data

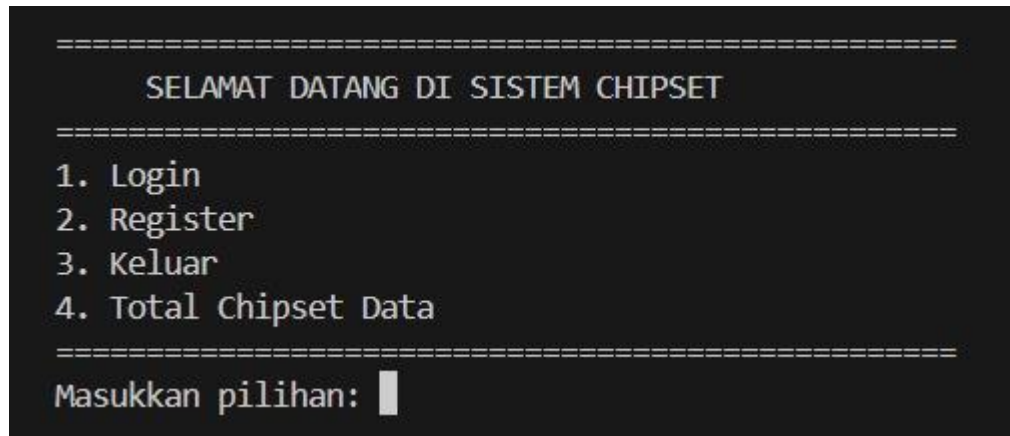
Prosedur ini memberikan kontrol kepada admin untuk mengelola data chipset, seperti menambah, melihat, mengubah, dan menghapus data.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu

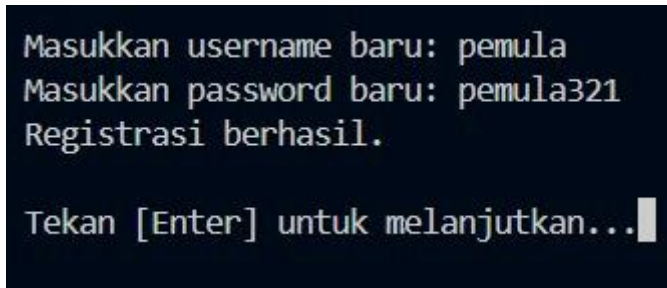


Gambar 2 Tampilan Menu

Ini adalah tampilan awal program kami, berisikan pesan selamat datang. Kemudian menampilkan beberapa pilihan yang akan diakses oleh pengguna.

1. **Login**
Memungkinkan pengguna yang telah terdaftar untuk masuk ke sistem. Pengguna harus memasukkan username dan password yang sesuai. Jika berhasil, mereka dapat mengakses menu tambahan berdasarkan peran (ADMIN atau USER).
2. **Register**
Memberikan opsi kepada pengguna baru untuk mendaftar ke sistem. Pengguna dapat membuat akun dengan username dan password. Akun baru secara default memiliki peran **USER**.
3. **Keluar**
Menutup program. Setelah memilih opsi ini, program akan menampilkan pesan perpisahan dan selesai dijalankan.
4. **Total Chipset Data**
Menampilkan jumlah data chipset yang tersedia dalam sistem. Data chipset meliputi informasi seperti nama, tahun rilis, jumlah core, dan teknologi proses.

2. Menu register



```
Masukkan username baru: pemula
Masukkan password baru: pemula321
Registrasi berhasil.

Tekan [Enter] untuk melanjutkan...
```

Gambar 3 Menu Register

1. Masukkan Username Baru

Program meminta pengguna untuk memasukkan nama pengguna (username) yang ingin didaftarkan. Username ini harus unik, dan program akan memeriksa apakah username tersebut sudah terdaftar dengan memanfaatkan fungsi `cek_username()`. Jika username sudah terdaftar, program akan menampilkan pesan: Username sudah terdaftar. dan tidak melanjutkan proses pendaftaran.

2. Masukkan Password Baru

Jika username belum terdaftar, program meminta pengguna untuk membuat password baru. Password ini akan disimpan bersamaan dengan username dalam struktur data akun.

3. Menambahkan ke

Setelah username dan password diterima, program akan menyimpan data pengguna baru dengan peran default **USER**.

4. Kembali ke Menu Utama

Setelah proses selesai, pengguna diminta untuk menekan [Enter] agar kembali ke menu utama.

3. Menu pengelolaan chipset



Gambar 4 Menu Pengelolaan Chipset

1. Lihat Semua Data Chipset

- Deskripsi: Menampilkan seluruh data chipset yang tersimpan dalam sistem.
- Akses: Tersedia untuk semua pengguna, termasuk ADMIN dan USER.
- Fungsi terkait: tampilkan_chipset.

2. Tambah Data Chipset (khusus ADMIN)

- Deskripsi: Menambahkan data chipset baru, termasuk nama, tahun rilis, jumlah core, dan teknologi proses.
- Akses: Hanya dapat diakses oleh pengguna dengan peran ADMIN.
- Proses:
 1. Pengguna diminta untuk memasukkan nama chipset, tahun rilis, jumlah core, dan teknologi proses.
 2. Program memeriksa apakah nama chipset sudah ada.
 3. Jika valid, data chipset baru akan ditambahkan.
- Fungsi terkait: Ditangani langsung dalam fungsi manage_chipset_data.

3. Ubah Data Chipset (khusus ADMIN)

- Deskripsi: Mengubah data chipset yang sudah ada.

- Akses: Hanya dapat diakses oleh pengguna dengan peran ADMIN.
 - Proses:
 1. Pengguna diminta untuk memilih chipset yang ingin diubah.
 2. Program menampilkan data lama chipset tersebut.
 3. Pengguna memasukkan data baru (tahun rilis, jumlah core, dan teknologi proses).
 4. Jika valid, data chipset diperbarui.
 - Fungsi terkait: Ditangani langsung dalam fungsi `manage_chipset_data`.
4. Hapus Data Chipset (khusus ADMIN)
- Deskripsi: Menghapus data chipset berdasarkan nama.
 - Akses: Hanya dapat diakses oleh pengguna dengan peran ADMIN.
 - Proses:
 1. Pengguna memilih chipset yang ingin dihapus.
 2. Program memverifikasi keberadaan chipset tersebut.
 3. Jika ditemukan, data chipset dihapus.
 - Fungsi terkait: Ditangani langsung dalam fungsi `manage_chipset_data()`.
5. Kembali ke Menu Utama
- Deskripsi: Mengakhiri menu pengelolaan chipset dan kembali ke menu utama.
 - Akses: Tersedia untuk semua pengguna.

3.2 Source Code

```
import os

# Data akun dan chipset
akun = {"admin": {"password": "admin321", "role": "ADMIN"}}
chipsetdata = {
    "A11 Bionic": {"Release Year": 2017, "Cores": 6, "Process":
"10nm"},
    "A12 Bionic": {"Release Year": 2018, "Cores": 6, "Process":
"7nm"},
    "A13 Bionic": {"Release Year": 2019, "Cores": 6, "Process":
```

```

"7nm+"},
    "A14 Bionic": {"Release Year": 2020, "Cores": 6, "Process":
"5nm"},
    "A15 Bionic": {"Release Year": 2021, "Cores": 6, "Process":
"5nm"},
    "A16 Bionic": {"Release Year": 2022, "Cores": 6, "Process":
"4nm"},
}

# Fungsi utilitas
def clearscreen():
    """Membersihkan layar untuk tampilan lebih rapi."""
    os.system("cls" if os.name == "nt" else "clear")

def garispemisah():
    """Mencetak garis pemisah untuk memperindah tampilan."""
    print("=" * 50)

def tekanlanjut():
    """Memberikan waktu jeda agar user bisa membaca sebelum
melanjutkan."""
    input("\nTekan [Enter] untuk melanjutkan...")

# Fungsi utama
def welcomeMessage():
    """Menampilkan menu utama."""
    clearscreen()
    garispemisah()
    print("        SELAMAT DATANG DI SISTEM CHIPSET")
    garispemisah()
    print("1. Login")
    print("2. Register")
    print("3. Keluar")
    print("4. Total Chipset Data")
    garispemisah()

def cekusername(username):
    """Memeriksa apakah username sudah terdaftar."""
    return username in akun

def tampilkanchipset():

```

```

    """Menampilkan data chipset secara iteratif."""
    if not chipsetdata:
        print("\nData chipset masih kosong.")
    else:
        garispemisah()
        print("                DATA CHIPSET")
        garispemisah()
        for nama_chipset, spesifikasi in chipsetdata.items():
            print(f"\nChipset: {nama_chipset}")
            for key, value in spesifikasi.items():
                print(f"    {key}: {value}")
            garispemisah()

def totalchipsetdata():
    """Menghitung jumlah chipset yang tersedia."""
    return len(chipsetdata)

def managechipsetdata(role):
    """Mengelola data chipset berdasarkan peran pengguna."""
    while True:
        clearscreen()
        garispemisah()
        print("                MENU PENGELOLAAN CHIPSET")
        garispemisah()
        print("1. Lihat semua data chipset")
        if role == "ADMIN":
            print("2. Tambah data chipset")
            print("3. Ubah data chipset")
            print("4. Hapus data chipset")
        print("5. Kembali ke menu utama")
        garispemisah()

        pilihan = input("Masukkan pilihan: ").strip()

        if pilihan == "1": # Lihat semua data chipset
            clearscreen()
            if not chipsetdata:
                print("\nData chipset masih kosong.")
            else:
                tampilkanchipset()
            tekanlanjut()

```



```

        elif pilihan == "2" and role == "ADMIN": # Tambah data
chipset
            clearscreen()
            print("          TAMBAH DATA CHIPSET")
            garispemisah()
            nama_chipset = input("Masukkan nama chipset baru:
").strip()
            if nama_chipset in chipsetdata:
                print("Chipset sudah terdaftar. Tidak dapat
menambahkan duplikat.")
            else:
                try:
                    tahun_rilis = int(input("Masukkan tahun rilis
(contoh: 2023): ").strip())
                    cores = int(input("Masukkan jumlah core (contoh:
6): ").strip())
                    proses = input("Masukkan teknologi proses
(contoh: 5nm): ").strip())
                    chipsetdata[nama_chipset] = {
                        "Release Year": tahun_rilis,
                        "Cores": cores,
                        "Process": proses,
                    }
                    print(f>Data chipset '{nama_chipset}' berhasil
ditambahkan.")
                except ValueError:
                    print("Input tidak valid. Pastikan tahun dan core
berupa angka.")
                tekanlanjut()

            elif pilihan == "3" and role == "ADMIN": # Ubah data chipset
                clearscreen()
                print("          UBAH DATA CHIPSET")
                garispemisah()
                nama_chipset = input("Masukkan nama chipset yang ingin
diubah: ").strip()
                if nama_chipset in chipsetdata:
                    try:
                        print(f>Data lama chipset '{nama_chipset}':
{chipsetdata[nama_chipset]}")

```

```

        tahun_rilis = int(input("Masukkan tahun rilis
baru: ").strip())
        cores = int(input("Masukkan jumlah core baru:
").strip())
        proses = input("Masukkan teknologi proses baru:
").strip()

        chipsetdata[nama_chipset] = {
            "Release Year": tahun_rilis,
            "Cores": cores,
            "Process": proses,
        }
        print(f>Data chipset '{nama_chipset}' berhasil
diperbarui.")
    except ValueError:
        print("Input tidak valid. Pastikan tahun dan core
berupa angka.")
    else:
        print("Chipset tidak ditemukan.")
        tekanlanjut()

elif pilihan == "4" and role == "ADMIN": # Hapus data
chipset
    clearscreen()
    print("          HAPUS DATA CHIPSET")
    garispemisah()
    if not chipsetdata:
        print("Tidak ada data chipset yang dapat dihapus.")
    else:
        nama_chipset = input("Masukkan nama chipset yang
ingin dihapus: ").strip()
        if nama_chipset in chipsetdata:
            del chipsetdata[nama_chipset]
            print(f>Data chipset '{nama_chipset}' berhasil
dihapus.")
        else:
            print("Chipset tidak ditemukan.")
            tekanlanjut()

elif pilihan == "5": # Kembali ke menu utama
    print("Kembali ke menu utama.")
    tekanlanjut()

```

```

        main()
        break

    else: # Penanganan input salah
        print("Pilihan tidak valid. Silakan coba lagi.")
        tekanlanjut()

def proses_login():
    """Proses login pengguna."""
    username = input("Masukkan username: ").strip()
    password = input("Masukkan password: ").strip()
    if cekusername(username) and akun[username]["password"] == password:
        print("Login berhasil!")
        tekanlanjut()
        managechipsetdata(akun[username]["role"])
        return True
    else:
        print("Username atau password salah.")
        tekanlanjut()
        return False

def prosesregister():
    """Proses registrasi pengguna baru."""
    username_baru = input("Masukkan username baru: ").strip()
    if cekusername(username_baru):
        print("Username sudah terdaftar.")
    else:
        password_baru = input("Masukkan password baru: ").strip()
        akun[username_baru] = {"password": password_baru, "role": "USER"}
        print("Registrasi berhasil.")
        tekanlanjut()

# Program utama
def main():
    salah = 0
    while salah < 3:
        welcomeMessage()
        pilihan = input("Masukkan pilihan: ").strip()

```

```

    if pilihan == "1":
        clearscreen()
        if proses_login():
            break
        else:
            salah += 1
    elif pilihan == "2":
        clearscreen()
        prosesregister()
    elif pilihan == "3":
        clearscreen()
        garispemisah()
        print("Terima kasih telah menggunakan program ini!")
        garispemisah()
        break
    elif pilihan == "4":
        clearscreen()
        garispemisah()
        print(f"Total Chipset Data: {totalchipsetdata()}")
        garispemisah()
        tekanlanjut()
    else:
        print("Pilihan tidak valid.")
        tekanlanjut()

    if salah >= 3:
        print("Anda telah gagal login sebanyak 3 kali. Program selesai.")

if __name__ == "__main__":
    main()

```

BAB IV

PENUTUP

4.1 Kesimpulan

Program ini berhasil mengimplementasikan sistem pengelolaan data yang efisien dan aman menggunakan konsep dasar pemrograman Python. Dengan memanfaatkan fitur CRUD (Create, Read, Update, Delete), program memungkinkan pengguna untuk menambah, melihat, memperbarui, dan menghapus data chipset Apple Bionic secara terstruktur.

Sistem login yang diterapkan membedakan hak akses antara admin dan user, di mana admin memiliki kontrol penuh atas data, sedangkan user hanya dapat melihat informasi yang tersedia. Keamanan ditingkatkan melalui pembatasan percobaan login hingga tiga kali, mencegah akses yang ilegal pada program. Fitur fungsi rekursif digunakan untuk menampilkan data chipset secara berurutan, menambah nilai pembelajaran dalam pengembangan program.

Program ini berbasis teks yang sederhana namun fungsional, sehingga mudah digunakan oleh pengguna. Meski memiliki keterbatasan, seperti tidak adanya penyimpanan permanen, program ini sudah memenuhi tujuan utama sebagai alat pembelajaran untuk memahami konsep CRUD, sistem login, validasi input, dan pengelolaan data dalam pemrograman Python.

Secara keseluruhan, program ini dirancang untuk memberikan solusi yang aman, efisien, dan mudah digunakan, sekaligus mendukung pembelajaran algoritma dasar pemrograman dengan aplikasi yang nyata.

4.2 Saran

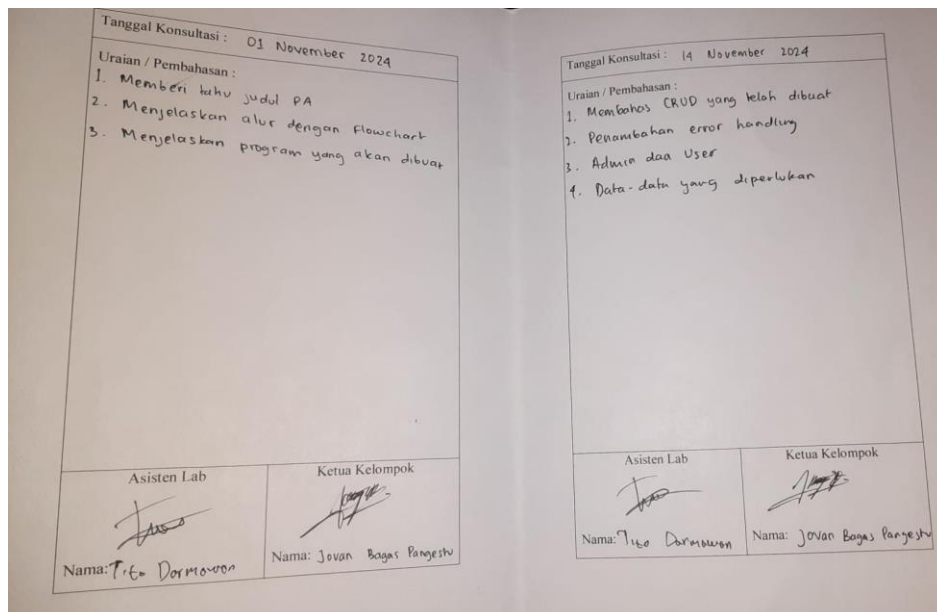
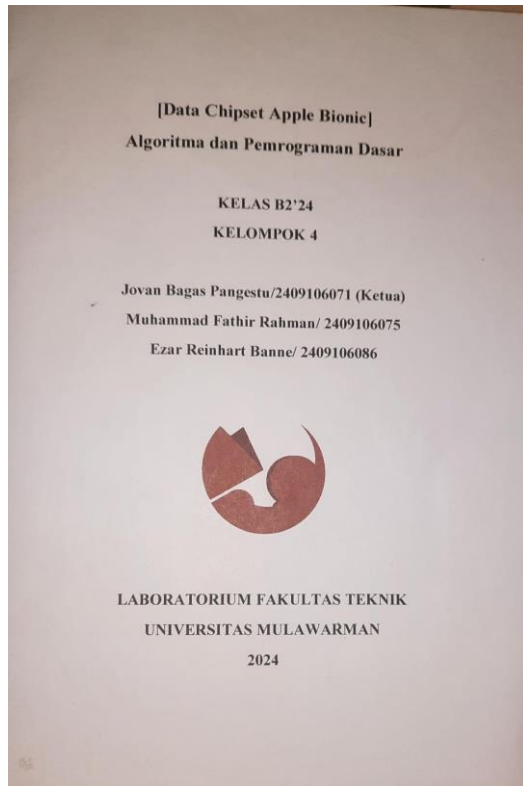
1. Diperlukan validasi input yang lebih ketat agar pengguna tidak bisa memasukkan data yang salah, seperti angka yang tidak sesuai atau karakter yang tidak diinginkan.

2. Program bisa menggunakan sistem penyimpanan data yang lebih efisien, seperti menyimpan data di file eksternal atau database, agar data tetap aman dan mudah diakses setelah program ditutup.
3. Kinerja Program perlu dioptimalkan agar berjalan lebih cepat dan efisien, misalnya dengan mengurangi kode yang berulang atau memperbaiki proses perhitungan.
4. Menambah Fitur backup dan restore data bisa ditambahkan untuk mencegah kehilangan data. Dengan cara ini, admin bisa menyimpan cadangan data yang ada.
5. Dokumentasi program perlu dibuat lebih jelas agar memudahkan pengguna dan pengembang lainnya memahami cara kerja program ini. Dokumentasi yang baik juga akan membantu dalam pengembangan program di masa depan..

DAFTAR PUSTAKA

- Djunaedi, H. Pemrograman Dasar dengan Python. Jakarta: PT Gramedia Pustaka Utama, 2020.
- Kurniawan, R. Belajar Python untuk Pemula: Panduan Lengkap dan Praktis. Bandung: Informatika, 2018.
- Tanenbaum, A. S. Struktur Data dan Algoritma dalam Pemrograman. Yogyakarta: Andi Offset, 2003.
- Apple Inc. Chipset Apple Bionic: Teknologi dan Spesifikasi. Diperoleh dari <https://www.apple.com>.
- Riyanto, A. Teknik Pengelolaan Basis Data dengan CRUD. Surabaya: Penerbit Informatika, 2022.
- Subakti, Y. A. Penerapan Fungsi dan Validasi pada Python. Jurnal Teknologi Informasi, 12(2), 45-58, 2019.
- Whittaker, J. A. Principles of Software Engineering and Design. Boston: Pearson Education, 2016.
- Tim Asisten Laboratorium Informatika Universitas Mulawarman. Modul Praktikum Pemrograman Dasar. Samarinda: Universitas Mulawarman, 2024.

LAMPIRAN



Gambar 5 Kartu Konsul