

LAPORAN AKHIR PROJECT MATA KULIAH
ALGORITMA & PEMROGRAMAN I
KELAS C



TRAIN EXPRESS

Dosen Pengampu:

Bapak Tio Dharmawan, S.Kom, M.Kom

Disusun Oleh:

Farhaz Nurjananto (212410102005 / Teknologi Informasi)

Farlin Nurjananti (212410102041 / Teknologi Informasi)

FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER

2021

DAFTAR ISI

DAFTAR ISI	2
I. PENDAHULUAN	3
1.1. Latar Belakang	3
1.2. Tujuan	3
1.3. Batasan Masalah	4
II. RANCANGAN PROGRAM	5
2.1. Kebutuhan Sistem	5
2.2. Struktur Data	5
2.3. Rancangan Langkah Penggunaan Program	7
III. IMPLEMENTASI RANCANGAN	9
3.1. Fitur Login	9
3.2. Fitur Daftar	9
3.3. Fitur Mengganti Password Pada User dan Admin	10
3.4. Fitur Search pada User	10
3.5. Fitur Show Tiket pada User	12
3.6. Fitur Basket pada User	12
3.7. Fitur Delet pada User dan Admin	13
3.8. Fitur Menambah Jadwal Kereta pada Admin	14
3.9. Fitur Update Jadwal Kereta pada Admin	15
3.10. Fitur Reporting pada Admin	16
IV. HASIL DAN PEMBAHASAN	17
4.1. Fitur Login	17
4.2. Fitur Daftar	17
4.3. Fitur Menampilkan Jadwal Kereta pada Admin	18
4.4. Fitur Menambah Jadwal Kereta pada Admin	18
4.5. Fitur Update Jadwal Kereta Pada Admin	19
4.6. Fitur Menghapus Jadwal Kereta pada Admin	19
4.7. Fitur Keranjang dan Pembayaran Tiket Kereta pada User	20
4.8. Fitur Ganti Password pada Admin dan User	20
4.9. Fitur Cari tiket Kereta pada User	21
4.10. Fitur Pembatalan Tiket Kereta pada User	21
4.11. Fitur Keranjang dan Pembayaran Tiket Kereta pada User	21
LINK YOUTUBE	21

I. PENDAHULUAN

1.1.Latar Belakang

Teknologi informasi saat ini berkembang pesat seiring dengan perkembangan zaman. Kemajuan teknologi informasi ini sangat berdampak pada segala sektor, salah satunya sektor pariwisata. Sektor pariwisata merupakan sektor yang potensial untuk dikembangkan sebagai salah satu sumber pendapatan daerah. Dengan itu, artinya sektor pariwisata berpengaruh terhadap perekonomian di suatu daerah atau negara tujuan wisata. Sektor pariwisata ini banyak dikaitkan dengan perkembangan teknologi yang ada seperti pengembangan program yang digunakan untuk mempermudah wisatawan dalam berpergian. Dimana dulu ketika sebelum wisatawan hendak berpergian, mereka diharuskan untuk memesan serta membeli tiket transportasi ke tempat yang hendak dituju, kecuali jika wisatawan berpergian menggunakan kendaraan pribadi. Proses itu tentunya memakan banyak waktu, dimana wisatawan harus mengantri dengan ratusan bahkan ribuan orang. Salah satu contohnya yaitu dalam pemesanan tiket kereta api.

Dengan itu, penulis ingin mengembangkan suatu program yang dapat menangani permasalahan di atas dan program ini dinamakan “Train Express”. **Train Express merupakan** sebuah program pemesanan tiket kereta api bagi wisatawan. Dimana program ini mempermudah pengguna dalam memesan tiket kereta api. Program Train Express ini nantinya dapat meminimalisir interaksi sesama manusia di stasiun. Train Express ini memiliki dua level pengguna yaitu level admin dan level user. Train Ekspress ini diharapkan dapat mempermudah wisatawan dalam pemesanan tiket kereta api supaya lebih efektif dan efisien. **Kami memilih program pemesanan tiket kereta api dengan program yang bernama “Train Ekspress” karena** ingin memaksimalkan pendapatan ekonomi pada sektor pariwisata dan transportasi di kala pandemi sehingga akan membantu dalam mempermudah pihak kereta api serta wisatawan untuk tetap menjaga protokol kesehatan. Program Train Express ini juga dapat digunakan ketika pandemi sudah usai, kerna Train Express ini dapat mempermudah pemesanan tiket

1.2. Tujuan

Tujuan dari pembuatan program “Train Express” adalah:

1. Mengganti sistem manual atau tradisional ke sistem otomatis.
2. Memberi kemudahan pada pengguna.
3. Menciptakan sistem yang praktis dan efisien.
4. Menciptakan keamanan bagi pengguna.

Disini pengguna tidak perlu mengantri untuk membeli tiket karena pengguna hanya perlu memesan tiket melalui program ini dan program secara otomatis mencari tiket yang sesuai dengan pengguna pesan. Selain itu, melalui pemesanan secara otomatis ini, pengguna akan lebih aman karena tidak perlu datang ke stasiun untuk membeli tiket serta memberikan kepastian kepada pengguna.

1.3.Batasan Masalah

Batasan masalah dari pembuatan program “Train Express” adalah:

1. Sistem program dibuat menggunakan bahasa pemrograman *python* dan *csv* dan *json* sebagai *database*.
2. Sistem program dapat berjalan tanpa dalam jaringan internet.
3. Sistem program dapat mencari, mengupdate, melakukan pembatalan, menambah jadwal kereta api dan melakukan pembayaran tiket kepada pengguna.
4. Sistem program akan secara otomatis melakukan update terkait jadwal yang pengguna pesan dan batalkan.

II. RANCANGAN PROGRAM

2.1. Kebutuhan Sistem

Pada program “Train Express” ini terdapat dua level yang saling keterkaitan, yaitu level admin dan level pengguna (konsumen). Dimana dua level dapat menggunakan hak aksesnya masing-masing yang memiliki kebutuhan sendiri-sendiri, antara lain:

1. Level Admin

Admin dapat mengelola jadwal tiket yang meliputi penambahan, pembaruan jadwal, penghapusan jadwal, reporting serta menampilkan jadwal kereta api yang telah ada yang dapat membantu mempermudah memonitoring jadwal kereta api.

2. Level User

User dapat mencari tiket kereta api yang ingin dipesan, menampilkan tiket kereta api yang telah dipesan, pembatalan tiket yang telah dipesan serta terdapat opsi bantuan sebagai panduan penggunaan program ini.

Kesimpulannya:

1. Program ini memiliki fitur login yang memisahkan level admin dengan level pengguna.
2. Program ini dapat menampilkan jadwal yang telah dipesan.
3. program ini memiliki pencarian tiket yang mudah.
4. Program ini memiliki fitur transaksi pembelian yang mudah.
5. Program ini terdapat fitur reporting yang dapat mempermudah pihak kereta api untuk memonitoring keuntungan serta sebaran tiket yang dibeli para pengguna kereta api.

2.2. Struktur Data

Struktur data dari program “Train Express” antara lain meliputi:

No	Fitur	Data	Tipe data
1.	Login	Username	String
		Password	String
2.	Daftar	Nama	String
		Username	String
		Password	String

		Re-Password	String
3.	Mengganti Password	Password Lama Password Baru Re-Password Baru	String String String
4.	Menambah jadwal pada level admin	Nama kereta Hari beroperasi Asal Tujuan Harga Berangkat pukul Tiba pukul	String String String String Integer String String
5.	Menghapus jadwal kereta api pada level admin	Nomor jadwal kereta yang dihapus	Integer
6.	Mengupdate jadwal pada level admin	Nomor kereta yang diupdate Kategori yang akan dihapus Nilai kategori yang akan diganti	Integer String String
7.	Menampilkan jadwal kereta api pada level admin	Menampilkan tiket kereta api	String
8.	Pencarian pada level user	Asal Tujuan Tanggal berangkat Bulan berangkat Tahun berangkat Pilih nomor kereta Jumlah penumpang Nama penumpang Total harga Pembayaran Kembalian	Integer Integer Integer Integer Integer Integer Integer String Integer Integer Integer

9.	Menampilkan tiket kereta api pada level user	Menampilkan tiket kereta api	String
10.	Menampilkan bantuan pada level user	Bantuan	String

2.3. Rancangan Langkah Penggunaan Program

1. Sebagai Admin

Pertama, menampilkan menu awal yang berisi 3 menu yaitu **1. Masuk, 2. Daftar, 3. keluar**. Admin dipersilahkan langsung melakukan login pada menu “Masuk” dengan pilih nomor 1. Kemudian masukkan username dan password, jika benar selanjutnya akan diarahkan pada tampilan awal program untuk admin yang berisi 6 menu. **1. Tampilkan jadwal kereta api, 2. Tambah jadwal kereta api, 3. Update jadwal kereta api, 4. Hapus jadwal kereta api, 5. Ganti password, 6. Reporting, 0. Keluar**. Jika ingin menampilkan jadwal kereta api pilih nomor 1, jika ingin menambah jadwal kererat api baru pilih no 2. Admin mengisikan nama kereta api, hari beroperasi, asal, tujuan, harga, berangkat pukul serta tiba pukul pada menu ini. Jika admin ingin mengupdate jadwal kereta api, pilih nomor 3. Admin mengisikan nomor kereta yang ingin di update, kategori yang ingin diganti dan menginputkan nilai baru dari kategori yang akan diganti. Jika admin ingin menghapus jadwal kereta api, maka pilih nomor 4. Admin mengisikan nomor jadwal kereta yang akan dihapus. Jika ingin mengganti password lama, maka pilih nomor 5. Admin mengisikan password lama dari admin, jika benar maka admin diarahkan untuk memasukkan password baru dan mengisikan Kembali password baru. Jika benar, admin akan mendapat konfirmasi “password berhasil dirubah”. Jika admin ingin melihat reporting catatan mengenai keuntungan total tiket yang terjual, sebaran tiket kereta api yang dibeli user dapat dilihat pada fitur nomor 6. Jika ingin keluar pilih nomor 0, maka program akan keluar dan kembali pada menu awal yaitu login dan daftar akun.

2. Sebagai user

Pertama, menampilkan menu awal yang berisi 3 menu yaitu **1. Masuk, 2. Daftar, 3. Keluar**. User dipersilahkan langsung melakukan login pada menu “Masuk” dengan pilih nomor 1. Kemudian masukkan username dan password jika benar selanjutnya akan diarahkan pada tampilan awal program untuk user yang berisi 6 menu. **1. Cari tiket kereta api, 2. Tampilkan kereta api anda, 3. Pembatalan tiket kereta api, 4. Ganti password, 5. Bantuan, 0. Keluar**. Jika user ingin mencari tiket kereta api, maka pilih nomor 1. User dipersilahkan untuk

mengisikan asal, tujuan, tanggal keberangkatan, bulan keberangkatan, dan tahun keberangkatan. Setelah mengisikan data dengan benar, maka program akan menampilkan jadwal kereta api yang sesuai dengan data yang telah diisikan oleh user. Setelah jadwal kereta yang diinginkan tampil, user diminta untuk memilih nomor tiket kereta api yang dipesan. Kemudian admin dipersilahkan mengisi jumlah penumpang yang akan membeli tiket. Setelah data benar, program akan menampilkan total harga yang dipesan dan user mengisikan total uang yang dibayar. Kemudian program akan menampilkan kembalian uang jika uang yang dibayarkan lebih. Jika user ingin melihat tiket kereta api yang dibeli, maka pilih 2. Program akan menampilkan tiket program yang dibeli oleh user. Jika user ingin membatalkan pemesanan tiket kereta api maka pilih nomor 3. User dipersilahkan untuk memilih nomor tiket yang ingin dibatalkan. Jika user ingin mengganti password lama, maka pilih nomor 4. User mengisikan password lama, jika benar maka user diarahkan untuk memasukkan password baru dan mengisikan kembali password baru. Jika benar, user akan mendapat konfirmasi “password berhasil dirubah”. Jika user ingin melihat panduan program, pilih nomor 5. Program akan menampilkan bantuan dari penggunaan program. Jika user ingin keluar pilih nomor 0, maka program akan keluar dan kembali pada menu awal yaitu login dan daftar akun.

III. IMPLEMENTASI RANCANGAN

3.1. Fitur Login

```
def masuk():
    while True:
        clear()
        header("MENU MASUK")
        print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
        print("="*140)

        global level
        global who
        global pw

        accounts = load_json()
        username = input("Username: ")
        if username.upper() == "B":
            strat_menu()
        else:
            password = input("Password: ")
            if password.upper() == "B":
                strat_menu()
            else:
                hashed_password = hashlib.md5(password.encode('utf-8')).hexdigest()
                for akun in accounts:
                    if username == akun['username'] and hashed_password == akun['password']:
                        level = akun["level"]
                        who = akun["username"]
                        pw = akun["password"]
                        main_menu()
                else:
                    input("\nUSERNAME ATAU PASSWORD SALAH")
```

3.2. Fitur Daftar

```
def daftar():
    while True:
        clear()
        header("MENU DAFTAR AKUN")
        print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
        print("="*140)

        new = {}

        accounts = load_json()
        reporting_data = load_csv(db_reporting)
        new['username'] = input("USERNAME: ")
        for account in accounts:
            if new["username"] in account["username"]:
                input("\nUSERNAME TELAH DIGUNAKAN")
                daftar()
        if new['username'].upper() == "B":
            strat_menu()
        else:
            new['password'] = input("PASSWORD: ")
            if new['password'].upper() == "B":
                strat_menu()
            else:
                password_confirm = input("RE-PASSWORD: ")
                if password_confirm == new["password"]:
                    new['password'] = hashlib.md5(new['password'].encode('utf-8')).hexdigest()
                    new['level'] = 'USER'
                    accounts.append(new)
                    with open(db_account, 'w') as output:
                        output.write(json.dumps(accounts, indent=2))

                    total_user = int(reporting_data[0]["total_user"]) + 1
                    reporting_data[0]["total_user"] = total_user

                    write_csv_reporting(reporting_data)
                    input("\nDAFTAR BERHASIL")
                    strat_menu()
                else:
                    input("\nPASSWORD TIDAK SAMA")
```

3.3. Fitur Mengganti Password pada User dan Admin

```
def change_pw():
    while True:
        clear()
        header("GANTI PASSWORD")
        print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
        print("="*140)

        accounts = load_json()

        old_pw = input("{:<30}: ".format("MASUKKAN PASSWORD LAMA"))
        hashed_password = hashlib.md5(old_pw.encode('utf-8')).hexdigest()
        if hashed_password == pw:
            new_pw1 = input("{:<30}: ".format("MASUKKAN PASSWORD BARU"))
            if new_pw1.upper() == "B":
                main_menu()
            else:
                new_pw2 = input("{:30}: ".format("MASUKKAN KEMBALI PASSWORD BARU"))
                if new_pw2.upper() == "B":
                    main_menu()
                elif new_pw2 == new_pw1:
                    for i in range(len(accounts)):
                        if accounts[i]['username'] == who:
                            accounts[i]['password'] = hashlib.md5(new_pw2.encode('utf-8')).hexdigest()

                            with open(db_account, "w") as output:
                                output.write(json.dumps(accounts, indent=2))

                            input("\nPASSWORD BERHASIL DIRUBAH")
                            main_menu()
                elif old_pw.upper() == "B":
                    main_menu()
                else:
                    input("\nPASSWORD LAMA SALAH")
```

3.4. Fitur Search pada User

```
def search():
    while True:
        clear()
        header("CARI TIKET KERETA")
        print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
        print("="*140)

        trains = load_csv(db_train)
        basket = load_csv(db_basket)

        asal = input("{:<25}: ".format("Asal"))
        if asal.upper() == "B":
            main_menu()
        else:
            tujuan = input("{:<25}: ".format("Tujuan"))
            if tujuan.upper() == "B":
                main_menu()
            else:
                tanggal = input("{:<25}: ".format("Tanggal Berangkat [1-31]"))
                if tanggal.upper() == "B":
                    main_menu()
                else:
                    bulan = input("{:<25}: ".format("Bulan Berangkat [1-12]"))
                    if bulan.upper() == "B":
                        main_menu()
                    else:
                        tahun = input("{:<25}: ".format("Tahun Berangkat"))
                        if tahun.upper() == "B":
                            main_menu()
                        else:
                            tanggal_berangkat = f"{tanggal} {bulan} {tahun}"

                            try:
                                now = datetime.datetime.now()
                                kalender = datetime.datetime(int(tahun), int(bulan), int(tanggal))
                                if kalender < now:
                                    input("\nTANGGAL/BULAN/TAHUN ANDA SALAH")
                                    search()
                            except:
```

```

input("\nTANGGAL/BULAN/TAHUN ANDA SALAH")
search()

hari = kalender.strftime("%A").upper()

konfirmasi = input("APA DATA SUDAH BENAR? [Y/N/B]: ")

if konfirmasi.upper() == "Y":
    rekomendasi_kereta = []
    not_rekomendasi_kereta = []
    for i in trains:
        if asal.upper() == i["asal"] and tujuan.upper() == i["tujuan"] and hari == i["hari"]:
            rekomendasi_kereta.append(i)
        else:
            not_rekomendasi_kereta.append(i)
    if rekomendasi_kereta != []:
        while True:
            show_train(rekomendasi_kereta)
            print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
            print("="*140)
            choice = input("{:<25}: ".format("Pilih Kereta Api [No]"))

            if choice.isnumeric():

                tmp_tiket = []
                choice = int(choice)

                if (choice-1) in range(len(rekomendasi_kereta)):
                    jumlah = input("{:<25}: ".format("Jumlah Penumpang"))
                    if jumlah.isnumeric():
                        jumlah = int(jumlah)
                        for i in range(jumlah):
                            nama = input("{:<25}: ".format("Nama Penumpang"))
                            tmp = { "pemesan": who,
                                    "penumpang": nama,
                                    "kereta": rekomendasi_kereta[choice-1]["kereta"],
                                    "asal": rekomendasi_kereta[choice-1]["asal"],
                                    "tujuan": rekomendasi_kereta[choice-1]["tujuan"],
                                    "berangkat": rekomendasi_kereta[choice-1]["berangkat"],
                                    "tiba": rekomendasi_kereta[choice-1]["tiba"],
                                    "harga": rekomendasi_kereta[choice-1]["harga"],
                                    "tanggal_berangkat": tanggal_berangkat,
                                    "hari": rekomendasi_kereta[choice-1]["hari"]
                                    }
                            tmp_tiket.append(tmp)

                    konfirmasi = input("APA DATA SUDAH BENAR? [Y/N/B]: ")

                    if konfirmasi.upper() == "Y":
                        sisa_tiket = int(rekomendasi_kereta[choice-1]["tiket"]) - jumlah
                        rekomendasi_kereta[choice-1]["tiket"] = sisa_tiket
                        update_db_train = rekomendasi_kereta + not_rekomendasi_kereta

                        for i in tmp_tiket:
                            basket.append(i)

                        write_csv_train(db_train, update_db_train)
                        write_csv_tiket(db_basket, basket)
                        input("\nTERIMA KASIH TELAH MEMESAN, LIHAT PEMBAYARAN ANDA PADA
                        MENU KERANJANG & BAYAR")
                        main_menu()
                    elif konfirmasi.upper() == "N":
                        input("\nKONFIRMASI DITERIMA")
                    elif konfirmasi.upper() == "B":
                        search()
                    else:
                        input("\nOPSI TIDAK ADA")
                elif jumlah.upper() == "B":
                    search()
                else:
                    input("\nANDA MEMASUKKAN SELAIN ANGKA")
            else:
                input("\nKERETA API TIDAK ADA")
        elif choice.upper() == "B":
            search()
        else:
            input("\nANDA MEMASUKKAN SELAIN ANGKA")
    else:
        input("\nTIDAK ADA JADWAL")
elif konfirmasi.upper() == "N":

```

```

        input("\nKONFIRMASI DITERIMA")
    elif konfirmasi.upper() == "B":
        main_menu()
    else:
        input("\nMENU TIDAK ADA")

```

3.5. Fitur Show Tiket pada User

```

def show_ticket():
    clear()
    user_ticket = []
    tickets = load_csv(db_ticket)
    for i in range(len(tickets)):
        if tickets[i]["pemesan"] == who:
            user_ticket.append(tickets[i])

    if isinstance(user_ticket, list):
        print("="*140)
        print("|{: ^138}|".format("TIKET KERETA API ANDA"))
        print("="*140)
        print("|{: ^3}|{: <25}|{: ^30}|{: ^20}|{: ^15}|{: ^15}|{: ^12}|{: ^11}|".format("NO", "KERETA", "PENUMPANG", "TANGGAL BERANGKAT", "ASAL", "TUJUAN", "BERANGKAT", "TIBA"))
        print("="*140)
        for idx, item in enumerate(user_ticket):
            print("|{: ^3}|{: <25}|{: ^30}|{: ^20}|{: ^15}|{: ^15}|{: ^12}|{: ^11}|".format((idx+1), item["kereta"], item["penumpang"], item["tanggal_berangkat"], item["asal"], item["tujuan"], item["berangkat"], item["tiba"]))
        print("="*140)
    else:
        print('TIPE DATA TIDAK SESUAI')

```

3.6. Fitur Basket pada User

```

def basket():
    while True:
        clear()
        header("KERANJANG & BAYAR")

        user_basket = []
        total_harga = 0
        baskets = load_csv(db_basket)
        tickets = load_csv(db_ticket)
        reporting_data = load_csv(db_reporting)
        reporting_data2 = load_csv(db_reporting2)

        for i in range(len(baskets)):
            if baskets[i]["pemesan"] == who:
                user_basket.append(baskets[i])
                total_harga += int(baskets[i]['harga'])

        print("|{: ^3}|{: <25}|{: ^30}|{: ^20}|{: ^15}|{: ^15}|{: ^12}|{: ^11}|".format("NO", "KERETA", "PENUMPANG", "TANGGAL BERANGKAT", "ASAL", "TUJUAN", "BERANGKAT", "TIBA"))
        print("="*140)
        for idx, item in enumerate(user_basket):
            print("|{: ^3}|{: <25}|{: ^30}|{: ^20}|{: ^15}|{: ^15}|{: ^12}|{: ^11}|".format((idx+1), item["kereta"], item["penumpang"], item["tanggal_berangkat"], item["asal"], item["tujuan"], item["berangkat"], item["tiba"]))
        print("="*140)

        if total_harga != 0:
            print("{: <15}: {}".format("TOTAL HARGA", total_harga))
            bayar = input("{: <15}: ".format("TOTAL HARGA"))
            if bayar.isnumeric():
                bayar = int(bayar)
                if bayar >= total_harga:
                    for i in user_basket:
                        tickets.append(i)
                        total_tiket_terjual = int(reporting_data[0]["total_tiket_terjual"]) + 1
                        reporting_data[0]["total_tiket_terjual"] = total_tiket_terjual

                    total_pendapatan = int(reporting_data[0]["total_pendapatan"]) + total_harga
                    reporting_data[0]["total_pendapatan"] = total_pendapatan

                    for i in range(len(reporting_data2)):
                        for idx in range(len(user_basket)):
                            if reporting_data2[i]["asal"] == user_basket[idx]["asal"] and reporting_data2[i]["tujuan"] == user_basket[idx]["tujuan"]:
                                pembeli = int(reporting_data2[i]["pembeli"]) + 1

```

```

pendapatan = int(reporting_data2[i]["pendapatan"]) + int(user_basket[idx]["harga"])
reporting_data2[i]["pembeli"] = pembeli
reporting_data2[i]["pendapatan"] = pendapatan
user_basket.clear()

write_csv_reporting2(reporting_data2)
write_csv_reporting(reporting_data)
write_csv_ticket(db_ticket, tickets)
write_csv_ticket(db_basket, user_basket)
kembalian = bayar - total_harga
print("{:<15}: {}".format("KEMBALI", kembalian))
input("\nTERIMA KASIH")
main_menu()
else:
    input("\nUANG ANDA TIDAK CUKUP")
elif bayar.upper() == "B":
    main_menu()
else:
    input("\nANDA MEMASUKKAN SELAIN ANGKA")
else:
    input("\nKERANJANG MASIH KOSONG")
    main_menu()

```

3.7. Fitur Delete pada User dan Admin

```

def delete():
    while True:
        if level == "USER":
            clear()
            load_file = []

            baskets = load_csv(db_basket)
            trains = load_csv(db_train)

            for i in range(len(baskets)):
                if baskets[i]["pemesan"] == who:
                    load_file.append(baskets[i])

            header("PEMBATALAN TIKET KERETA API")
            print("|{:<3}|{:<25}|{:<30}|{:<20}|{:<15}|{:<15}|{:<12}|{:<11}|".format("NO", "KERETA", "PENUMPANG", "TANGGAL
            BERANGKAT", "ASAL", "TUJUAN", "BERANGKAT", "TIBA"))
            print("="*140)
            for idx, item in enumerate(load_file):
                print("|{:<3}|{:<25}|{:<30}|{:<20}|{:<15}|{:<15}|{:<12}|{:<11}|".format((idx+1), item["kereta"],
                item["penumpang"], item["tanggal_berangkat"], item["asal"], item["tujuan"], item["berangkat"],
                item["tiba"]))
            print("="*140)

            print("|{:<138}|".format("B: BACK, UNTUK KEMBALI"))
            print("="*140)
            idx = (input('TIKET YANG AKAN DIBATALKAN [NO/B]: '))
        elif level == "ADMIN":
            load_file = load_csv(db_train)
            show_train(load_file)
            header("DELETE JADWAL")
            print("|{:<138}|".format("B: BACK, UNTUK KEMBALI"))
            print("="*140)
            idx = (input('JADWAL YANG AKAN DIHAPUS [NO/B]: '))

        if idx.isnumeric():
            idx = int(idx)
            if (idx - 1) in range(len(load_file)):
                choice = input("APAKAH ANDA YAKIN? [Y/N/B]: ")
                if choice.upper() == "Y":
                    if level == "USER":
                        baskets.remove(load_file[idx-1])

                    for i in range(len(trains)):
                        if trains[i]["kereta"] == load_file[idx-1]["kereta"] and trains[i]["asal"] == load_file[idx-
                        1]["asal"] and trains[i]["tujuan"] == load_file[idx-1]["tujuan"] and trains[i]["harga"] ==
                        load_file[idx-1]["harga"] and trains[i]["berangkat"] == load_file[idx-1]["berangkat"] and
                        trains[i]["tiba"] == load_file[idx-1]["tiba"] and trains[i]["hari"] == load_file[idx-
                        1]["hari"]:
                            sisa_tiket = int(trains[i]["tiket"]) + 1
                            trains[i]["tiket"] = sisa_tiket

                    load_file.pop(idx - 1)

```

```

        write_csv_train(db_train, trains)
        write_csv_ticket(db_basket, load_file)
    elif level == "ADMIN":
        load_file.pop(idx - 1)
        write_csv_train(db_train, load_file)

    input("\nTELAH DIHAPUS")
    elif choice.upper() == "N":
        input("\nKONFIRMASI DITERIMA")
    elif choice.upper() == "B":
        main_menu()
    else:
        input("\nMENU TIDAK ADA")
else:
    input("\nPILIHAN TIDAK ADA")
elif idx.upper() == "B":
    main_menu()
else:
    input("\nANDA MEMASUKKAN SELAIN ANGKA")

```

3.8. Fitur Menambah Jadwal Kerata pada Admin

```

def add_data():
    while True:
        clear()
        header("TAMBAH JADWAL KERETA API")
        print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
        print("="*140)

        trains = load_csv(db_train)
        data_reporting2 = load_csv(db_reporting2)
        tmp = {}
        rpt = {}

        tmp["kereta"] = input("{:<25}: ".format("NAMA KERETA")).upper()
        if tmp["kereta"].upper() == "B":
            main_menu()
        else:
            tmp["hari"] = input("{:<25}: ".format("HARI PEROPRASI [ENGLISH]")).upper()
            if tmp["hari"].upper() == "B":
                main_menu()
            elif tmp["hari"].upper() not in day:
                input("\nMASUKKAN HARI DENGAN BENAR")
            else:
                tmp["asal"] = input("{:<25}: ".format("ASAL")).upper()
                if tmp["asal"].upper() == "B":
                    main_menu()
                else:
                    tmp["tujuan"] = input("{:<25}: ".format("TUJUAN")).upper()
                    if tmp["tujuan"].upper() == "B":
                        main_menu()
                    else:
                        tmp["harga"] = input("{:<25}: ".format("HARGA"))
                        if tmp["harga"].isnumeric():
                            tmp["berangkat"] = input("{:<25}: ".format("BERANGKAT PUKUL [00:00]")).upper()
                            if tmp["berangkat"].upper() == "B":
                                main_menu()
                            else:
                                tmp["tiba"] = input("{:<25}: ".format("TIBA PUKUL [00:00]")).upper()
                                if tmp["tiba"].upper() == "B":
                                    main_menu()
                                else:
                                    tmp["tiket"] = input("{:<25}: ".format("JUMLAH TIKET")).upper()
                                    if tmp["tiket"].isnumeric():
                                        konfirmasi = input("APAKAH DATA SUDAH BENAR? [Y/N/B]: ")
                                        if konfirmasi.upper() == "Y":
                                            trains.append(tmp)

                                            rpt["asal"] = tmp["asal"]
                                            rpt["tujuan"] = tmp["tujuan"]
                                            rpt["pembeli"] = 0
                                            rpt["pendapatan"] = 0

                                            data_exist = []
                                            for i in range(len(data_reporting2)):
                                                if data_reporting2[i]["asal"] == rpt["asal"] and
                                                    data_reporting2[i]["tujuan"] == rpt["tujuan"]:
                                                    data_exist.append(data_reporting2[i])

```

```

        if data_exist == []:
            data_reporting2.append(rpt)
            write_csv_reporting2(data_reporting2)

        write_csv_train(db_train, trains)

        input("\nDATA BERHASIL DITAMBAHKAN")
        main_menu()
    elif konfirmasi.upper() == "N":
        input("\nKONFIRMASI DITERIMA")
    elif konfirmasi.upper() == "B":
        main_menu()
    else:
        input("\nOPSI TIDAK ADA")
    elif tmp["tiket"].upper() == "B":
        main_menu()
    else:
        input("\nMASUKKAN JUMLAH TIKET DENGAN BENAR")
elif tmp["harga"].upper() == "B":
    main_menu()
else:
    input("\nANDA MEMASUKKAN SELAIN ANGKA")

```

3.9. Fitur Update Jadwal Kerata pada Admin

```

def update():
    while True:
        trains = load_csv(db_train)

        show_train(trains)
        header("UPDATE DATA KERETA")
        print("|{: ^138}|".format("B: BACK, UNTUK KEMBALI"))
        print("="*140)

        idx = input('{:<20}: '.format("NOMOR KERETA"))
        if idx.isnumeric():
            idx = int(idx)
            if (idx-1) in range(len(trains)):
                kategori = input('{:<20}: '.format("KATEGORI YANG AKAN DI GANTI")).lower()
                if kategori in trains[idx-1].keys():
                    if kategori == "harga":
                        data_baru = input('{:<20}: ".format(f"{kategori.upper()} BARU')).upper()
                        if data_baru.isnumeric():
                            trains[idx-1][kategori] = data_baru

                            write_csv_train(db_train, trains)

                            input("\nDATA BERHASIL DI UPDATE")
                    else:
                        input("\nANDA MEMASUKKAN SELAIN ANGKA")
                elif kategori == "tiket":
                    data_baru = input('{:<30}: ".format(f"JUMLAH {kategori.upper()} YANG AKAN DITAMBAHKAN')).upper()
                    if data_baru.isnumeric():
                        trains[idx-1][kategori] = int(trains[idx-1][kategori]) + int(data_baru)

                        write_csv_train(db_train, trains)

                        input("\nDATA BERHASIL DI UPDATE")
                    else:
                        input("\nANDA MEMASUKKAN SELAIN ANGKA")
                elif kategori == 'hari':
                    hari_baru = input('{:<20}: ".format(f"{kategori.upper()} BARU')).upper()
                    if hari_baru in day:
                        trains[idx-1][kategori] = hari_baru

                        write_csv_train(db_train, trains)

                        input("\nDATA BERHASIL DI UPDATE")
                    else:
                        input("\nMASUKKAN HARI DENGAN BENAR")
                elif kategori == 'berangkat' or kategori == 'tiba':
                    berangkat_baru = input('{:<20}: ".format("BERANGKAT BARU')).upper()
                    tiba_baru = input('{:<20}: ".format("TIBA BARU')).upper()
                    trains[idx-1]["berangkat"] = berangkat_baru
                    trains[idx-1]["tiba"] = tiba_baru

                    write_csv_train(db_train, trains)

```

```

        input("\nDATA BERHASIL DI UPDATE")
    else:
        trains[idx-1][kategori] = input("{:<20}: ".format(f"{kategori.upper()} BARU")).upper()

        write_csv_train(db_train, trains)

        input("\nDATA BERHASIL DI UPDATE")
    elif kategori == "b":
        main_menu()
    else:
        input("\nKATEGORI TIDAK ADA")
else:
    input("\nNOMOR KERETA API TIDAK ADA")
elif idx.upper() == "B":
    main_menu()
else:
    input("\nANDA MEMASUKKAN SELAIN ANGKA")

```

3.10. Fitur Reporting pada Admin

```

def reporting():
    clear()
    data_reporting = load_csv(db_reporting)
    data_reporting2 = load_csv(db_reporting2)
    if isinstance(data_reporting, list):
        print("="*83)
        print("|{: ^81}|".format("REPORTING DATA"))
        print("="*83)
        print("|{: ^3}|{:<22}|{: ^22}|{: ^15}|{: ^15}|".format("NO", "ASAL", "TUJUAN", "PEMBELI", "PENDAPATAN"))
        print("="*83)
        for idx, item in enumerate(data_reporting2):
            print("|{: ^3}|{:<22}|{: ^22}|{: ^15}|{: ^15}|".format((idx+1), item["asal"], item["tujuan"], item["pembeli"],
                item["pendapatan"]))
        print("="*83)
        print("| {:<40} | {: ^36} |".format("TOTAL PENGGUNA",data_reporting[0]["total_user"]))
        print("| {:<40} | {: ^36} |".format("TOTAL PENDAPATAN",data_reporting[0]["total_pendapatan"]))
        print("| {:<40} | {: ^36} |".format("TOTAL TIKET TERJUAL",data_reporting[0]["total_tiket_terjual"]))
        print("="*83)
        input("\nENTER UNTUK KEMBALI...")
        main_menu()
    else:
        print('TIPE DATA TIDAK SESUAI')

```


IV. HASIL DAN PEMBAHASAN

Setelah memilih menu, maka baik admin maupun user akan diarahkan pada fitur-fitur yang terdapat di dalam menu yaitu pada level admin akan tersedia fitur menampilkan jadwal kereta api, tambah jadwal kereta api, update jadwal kereta api, hapus jadwal kereta api, ganti password, reporting, dan keluar. Sedangkan pada level user akan tersedia fitur cari tiket kereta api, menampilkan kereta api yang telah dipesan, pembatalan tiket kereta api, ganti password, bantuan, dan keluar.

4.1. Fitur Login

Pada menu awal tadi terdapat tiga menu, yaitu 1. Masuk, 2. Daftar. 0.Keluar. Semua input dari menu tersebut menggunakan tipe data String. Disini menu awal dimana untuk menu 1 dapat login sebagai admin bagi petugas dan sebagai user bagi pengguna kereta api. Dan untuk menu 2, jika user belum memiliki akun, maka user dapat mendaftar akun terlebih dulu. dan jika ingin keluar dari program, maka pilih menu 3.

```
|
|                                     SELAMAT DATANG DI TRAIN EXPRESS                                     |
|-----|
|1 | MASUK
|2 | DAFTAR AKUN
|0 | KELUAR
|-----|
MENU:
```

Kemudian di bawah ini adalah tampilan output dari program “Train Express”, yaitu **menu masuk**.

```
|
|                                     MENU MASUK                                     |
|-----|
|                                     B: BACK, UNTUK KEMBALI                                     |
|-----|
Username: admin
Password: admin
```

Pada menu masuk ini, masukkan username “admin” jika sebagai petugas dan username “user” jika sebagai pengguna. Kemudian masukkan password sesuai dengan yang sudah dimiliki. Jika belum memiliki akun, maka dipersilahkan untuk kembali ke menu awal dan melakukan pendaftaran. Untuk pengisian username dan password menggunakan tipe data String.

4.2. Fitur Daftar

Selanjutnya ini adalah tampilan output dari **fitur daftar**, dimana fitur ini hanya dikhususkan untuk para user saja. Seperti yang dibahas sebelumnya, jika user belum memiliki akun maka dipersilahkan untuk mendaftar akun terlebih dulu. Fitur daftar akun ini berisi data username, password, dan re-password, dimana semuanya menggunakan tipe data String. Jika user telah mengisi datanya maka akan muncul “Daftar Berhasil” yang artinya user sudah bisa masuk ke program ini.

```
|
|                                     MENU DAFTAR AKUN                                     |
|-----|
|                                     B: BACK, UNTUK KEMBALI                                     |
|-----|
USERNAME: farlin
PASSWORD: nurjananti
RE-PASSWORD: nurjananti
DAFTAR BERHASIL
```

4.3. Fitur Menampilkan Jadwal Kereta Api Pada Admin

Berikut tampilan menu “Masuk” jika sebagai admin. Pada menu ini admin dapat menampilkan jadwal kereta api, menambah, meng-update, menghapus jadwal kereta api, mengganti password, reporting dan keluar.

```
|
=====
|                                     PROGRAM PEMESANAN TIKET KERETA API                                     |
=====
| 1 | TAMPILKAN JADWAL KERETA API |
| 2 | TAMBAH JADWAL KERETA API  |
| 3 | UPDATE JADWAL KERETA API  |
| 4 | HAPUS JADWAL KERETA API   |
| 5 | GANTI PASSWORD            |
| 6 | REPORTING                 |
| 0 | KELUAR                     |
=====
Menu: |
```

Kemudian di bawah ini adalah tampilan output dari program “Train Express”, yaitu **fitur menampilkan jadwal kereta api pada admin**.

```
|
=====
|                                     TIKET KERETA API                                     |
=====
| NO | KERETA | HARI | ASAL | TUJUAN | HARGA | BERANGKAT | TIBA | TIKET |
=====
| 1 | PROBOWANGI | MONDAY | PROBOLINGGO | JEMBER | 27000 | 08:00 | 10:00 | 30 |
| 2 | PROBOWANGI | TUESDAY | JEMBER | PROBOLINGGO | 27000 | 16:00 | 18:00 | 30 |
| 3 | PROBOWANGI | WEDNESDAY | JEMBER | PROBOLINGGO | 27000 | 08:00 | 10:00 | 30 |
=====
ENTER UNTUK KEMBALI KE MENU UTAMA.... |
```

Disini admin dapat melihat daftar tiker kereta api. Dimana terdapat nama kereta, hari kereta beroperasi, asal serta tujuan kereta, harga tiket, berangkat serta tiba kereta dan jumlah tiket yang tersedia.

4.4. Fitur Menambah Jadwal Kereta Api Pada Admin

```
|
=====
|                                     TAMBAH JADWAL KERETA API                                     |
=====
|                                     B: BACK, UNTUK KEMBALI                                     |
=====
NAMA KERETA      : probowangi
HARI PEROPRASI [ENGLISH] : monday
ASAL             : probolinggo
TUJUAN           : jember
HARGA            : 27000
BERANGKAT PUKUL [00:00] : 08:00
TIBA PUKUL [00:00]    : 10:00
JUMLAH TIKET     : 30
APAKAH DATA SUDAH BENAR? [Y/N/B]: y
DATA BERHASIL DITAMBAHKAN |
```

Pada fitur tambah jadwwal ini, admin dapat menambah jadwal kereta yang akan ditambahkan meliputi nama kereta, hari kereta beroperasi, asal serta tujuan kereta, harga tiket, berangkat serta tiba kereta dan jumlah tiket. Dimana tipe data yang dimasukkan yaitu nama kereta, hari kereta beroperasi, asal, tujuan kereta, berangkat, tiba kereta dan jumlah tiket berupa String, sedangkan untuk harga tiket yaitu tipe data Integer. Data yang dimasukkan sudah benar maka masukkan “Y”, jika salah masukkan “N” dan untuk Kembali masukkan “B”. Jika data sudah ditambahkan maka program akan menampilkan “Data Berhasil Ditambahkan”.

4.5. Fitur Update Jadwal Kereta Api pada Admin

TIKET KERETA API								
NO	KERETA	HARI	ASAL	TUJUAN	HARGA	BERANGKAT	TIBA	TIKET
1	PROBOWANGI	MONDAY	PROBOLINGGO	JEMBER	27000	08:00	10:00	30
2	PROBOWANGI	TUESDAY	JEMBER	PROBOLINGGO	27000	16:00	18:00	30
3	PROBOWANGI	WEDNESDAY	JEMBER	PROBOLINGGO	27000	08:00	10:00	30
UPDATE DATA KERETA								
B: BACK, UNTUK KEMBALI								
NOMOR KERETA : 2 KATEGORI YANG AKAN DI GANTI: berangkat BERANGKAT BARU : 17:00 TIBA BARU : 19:00 DATA BERHASIL DI UPDATE								

Berikut tampilan update jadwal kereta api pada admin. Admin dapat memilih nomor kereta dan kategori yang akan diganti. Dimana jika admin akan mengganti keberangkatan / tiba kereta, maka program akan otomatis menampilkan “berangkat baru” dan “tiba baru” dari kereta. Jika admin akan mengganti asal / tujuan kereta, maka program akan otomatis menampilkan “asal” dan “tujuan” kereta. Tipe data yang dimasukkan yaitu nama kereta, hari kereta beroperasi, asal, tujuan kereta, berangkat, tiba kereta dan jumlah tiket berupa String, sedangkan untuk harga tiket yaitu tipe data Integer. Jika data admin sudah sesuai yang akan diganti, maka program akan menampilkan “Data Berhasil Di Update”.

4.6. Fitur Menghapus Jadwal Kereta Api pada Admin

TIKET KERETA API								
NO	KERETA	HARI	ASAL	TUJUAN	HARGA	BERANGKAT	TIBA	TIKET
1	PROBOWANGI	MONDAY	PROBOLINGGO	JEMBER	27000	08:00	10:00	30
2	PROBOWANGI	TUESDAY	JEMBER	PROBOLINGGO	27000	17:00	19:00	30
3	PROBOWANGI	WEDNESDAY	JEMBER	PROBOLINGGO	27000	08:00	10:00	30
DELETE JADWAL								
B: BACK, UNTUK KEMBALI								
JADWAL YANG AKAN DIHAPUS [NO/B]: 3 APAKAH ANDA YAKIN? [Y/N/B]: y TELAH DIHAPUS								

Pada fitur hapus jadwal ini, admin hanya memasukkan nomor yang akan dihapus. Kemudian jika data sudah benar maka masukkan “Y”, jika salah masukkan “N” dan untuk Kembali masukkan “B”. Jika data sudah ditambahkan maka program akan menampilkan “Telah Dihapus”. Tipe data yang dimasukkan yaitu Integer.

4.7. Fitur Reporting pada Admin

REPORTING DATA				
NO	ASAL	TUJUAN	PEMBELI	PENDAPATAN
1	PROBOLINGGO	JEMBER	3	81000
TOTAL PENGGUNA			1	
TOTAL PENDAPATAN			81000	
TOTAL TIKET TERJUAL			3	
ENTER UNTUK KEMBALI....				

Disini admin dapat mengetahui jumlah pengguna, total pendapatan, dan total tiket terjual serta sebaran tiket kereta api yang banyak dibeli pengguna.

4.8. Fitur Ganti Password Pada Admin Dan User

```
=====
|                                     GANTI PASSWORD                                     |
|=====|
|                                     B: BACK, UNTUK KEMBALI                             |
|=====|
MASUKKAN PASSWORD LAMA      : nurjananti
MASUKKAN PASSWORD BARU     : password1
MASUKKAN KEMBALI PASSWORD BARU: password1
PASSWORD BERHASIL DIRUBAH
```

Disini admin memasukkan password lama, password baru dan re-password baru. Dimana tipe datanya yaitu String. Jika data selesai dimasukkan maka program akan menampilkan “Password Berhasil Dirubah”.

4.9. Fitur Cari Tiket Kereta Api Pada User

Disini akan menampilkan data yaitu asal, tujuan, tanggal berangkat, bulan berangkat, dan tahun berangkat kereta api. User memasukkan data sesuai dengan keberangkatan user. Kemudian jika data sudah benar maka masukkan “Y”, jika salah masukkan “N” dan untuk Kembali masukkan “B”.

```
=====
|                                     CARI TIKET KERETA                                     |
|=====|
|                                     B: BACK, UNTUK KEMBALI                             |
|=====|
Asal      : probolinggo
Tujuan    : jember
Tanggal Berangkat [1-31] : 13
Bulan Berangkat [1-12]  : 12
Tahun Berangkat      : 2021
APA DATA SUDAH BENAR? [Y/N/B]:
```

Setelah itu, user akan diarahkan untuk memasukkan data yaitu pilih nomor kereta api, jumlah penumpang, dan nama penumpang sesuai jumlah penumpang yang dimasukkan. Jika data sudah benar, maka program akan menampilkan “Terimakasih telah Memesan, Lihat Pembayaran Anda Pada Menu Keranjang & Bayar”.

TIKET KERETA API								
NO	KERETA	HARI	ASAL	TUJUAN	HARGA	BERANGKAT	TIBA	TIKET
1	PROBOWANGI	MONDAY	PROBOLINGGO	JEMBER	27000	08:00	10:00	30
B: BACK, UNTUK KEMBALI								
Pilih Kereta Api [No] : 1								
Jumlah Penumpang : 3								
Nama Penumpang : penumpang1								
Nama Penumpang : penumpang2								
Nama Penumpang : penumpang3								
APA DATA SUDAH BENAR? [Y/N/B]: y								
TERIMA KASIH TELAH MEMESAN, LIHAT PEMBAYARAN ANDA PADA MENU KERANJANG & BAYAR								

4.10. Fitur Pembatalan Tiket Kereta Api Pada User

PEMBATALAN TIKET KERETA API								
NO	KERETA	PENUMPANG	TANGGAL BERANGKAT	ASAL	TUJUAN	BERANGKAT	TIBA	
1	PROBOWANGI	penumpang1	13 12 2021	PROBOLINGGO	JEMBER	08:00	10:00	
2	PROBOWANGI	penumpang2	13 12 2021	PROBOLINGGO	JEMBER	08:00	10:00	
3	PROBOWANGI	penumpang3	13 12 2021	PROBOLINGGO	JEMBER	08:00	10:00	
B: BACK, UNTUK KEMBALI								
TIKET YANG AKAN DIBATALKAN [NO/B]: 1								
APAKAH ANDA YAKIN? [Y/N/B]: y								
TELAH DIHAPUS								

Disini user dapat melakukan pembatalan dari tiket yang dipesan sebelumnya dengan memasukkan tiket yang akan dibatalkan dalam bentuk nomor. Kemudian jika data sudah benar maka masukkan “Y”, jika salah masukkan “N” dan untuk Kembali masukkan “B”. Jika data sudah dibatalkan maka program akan menampilkan “Telah Dihapus”.

4.11. Fitur Keranjang Dan Pembayaran Tiket Kereta Api Pada User

KERANJANG & BAYAR								
NO	KERETA	PENUMPANG	TANGGAL BERANGKAT	ASAL	TUJUAN	BERANGKAT	TIBA	
1	PROBOWANGI	penumpang2	13 12 2021	PROBOLINGGO	JEMBER	08:00	10:00	
2	PROBOWANGI	penumpang3	13 12 2021	PROBOLINGGO	JEMBER	08:00	10:00	
TOTAL HARGA : 54000								
TOTAL HARGA : 60000								
KEMBALI : 6000								
TERIMA KASIH								

Pada fitur keranjang dan pembayaran tiket kereta api, user memasukkan nominal uang untuk pembayaran tiket. Jika nominal uang yang dimasukkan kurang dari total harga tiket maka program akan menampilkan “Uang Tidak Cukup” dan jika nominal uang yang dimasukkan lebih dari total harga maka program akan menampilkan kembali dari uang yang user bayar Dan program akan secara otomatis menampilkan “Terima Kasih”.

LINK VIDEO YOUTUBE: https://youtu.be/LKbIXid_4pA