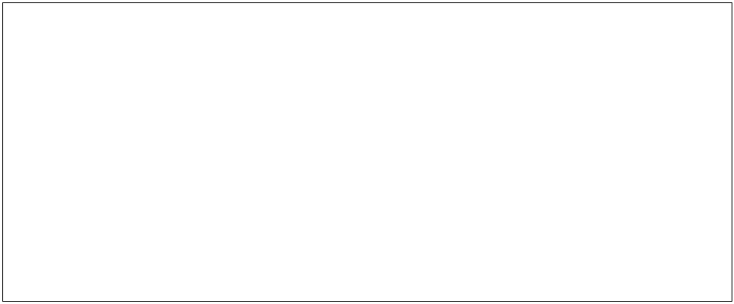
|  |  |
| --- | --- |
| **UNIVERSITAS IPWIJA**  PROGRAM STUDI INFORMATIKA (S1)  Semester Ganjil 2023/2024 | **UJIAN AKHIR SEMESTER (UAS)**  Mata Kuliah : Konsep Pemrogramman  Dosen : M.Maulana. Racman, S.Kom., M.Kom  Hari/Tanggal : 20 January 2024  Waktu : 09:00 s/d 11:00 wib |

**Pertauran Umum :**



1. Sebelum mengerjakan soal sebaiknya membaca do’a terlebih dahulu
2. Kerjakan soal dengan jujur, baik dan benar
3. Kerjakan soal dengan menggunakan Ms Word lalu di convert ke PDF
4. Untuk jawaban koding di zip diupload di : <https://1drv.ms/f/s!Ar443gAWbOTAgqVZk6lkNQRd2RCgFw?e=PVScz3>

Buat folder dengan nim\_nama\_uas

1. Pastikan koding yang diupload bisa dibuka dan di compail.
2. Untuk hasil dari koding di screenshoot masukan ke point 3 diatas.
3. Beri komentar yang sesuai dari setiap koding yang digunakan contoh :

***// menampilkan hasil inputan***

***for(byte b=0; b < 0 baris; b++)***

1. Mohon dikerja secara maksimal, jawaban tidak boleh sama dengan teman.
2. Tuliskan Nama, Nim, Prodi, Mata Kuliah. Semester, Tanggal pada kode program
3. Jika ada salah satu ketentuan tidak terpenuhi maka nilai dikurangkan atau tidak dinilai.
4. ***Permasalahan :***

Wanwans disuruh ibunya untuk kepasar membeli telur sebanyak 5 kilogram , gula merah 4 kilogram dan tepung terigu 10 kilogram untuk membuat kue, 1 kilogram telur adalah Rp. 28.000 /kg perkilogram, 1 kilogram gula merah Rp.12.000 /kg dan 1 kilogram tepung terigu adalah Rp.8000 /kg. Untuk pergi kepasar wanwans harus menaiki angkot pulang dan pergi dengan tarif Rp.5000 sekali naik angkot, berapakah sisa uamh wanwans jika membawa uang sebesar Rp.150.000

***Identifikasi masalah :***

* Input : berat telur, berat gula merah, berat tepung terigu, harga telur, harga gula, harga tepung, transportasi, uang wanwans.
* Output : Sisa Uang Wanwans

***Buatlah program dengan C# !***

1. ***Permasalahan :***

Kaisar adalah pedagang semangka yang menjual dagangannya yang setiap kilogramnya semangka dihargai dengan harga tertentu. Setiap pembeli membayar harga semangka yang dibeli sesuai dengan berat semangkat tersebut. Tentukan algoritma pedagang untuk menentukan harga yang harus dibayar pembeli. ***Identifikasi masalah :***

* Input : Harga /kg , berat pembeli
* Output : Harga yang dibayar pemebli.

***Buatlah program dengan C# !***

1. ***Permasalahan :***

Diketahui sebuah array memiliki sebuah data 10, 60, 20, 100, dan 40 **Identifikasi masalah :**

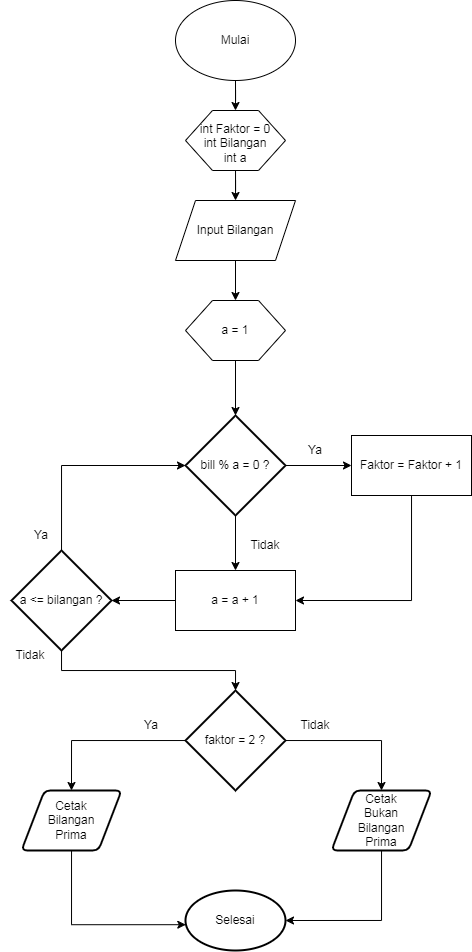
* Input : data array
* Output : nilai terbesar.

**Buatlah program dengan c# dan Flowchart !**

1. ***Permasalahan :***

Wanwans mempunyai kelereng sebanyak 20 lebih banyak dari kaisar, sedangka man’an mempunyai kelereng 3x jumlah kelerang wanwans dan kaisar. Sedangkan irwan memiliki 10 kelereng lebih sedikit dari jumlah kelereng wanwans, kaisar dan man’an. Berapakah jumlah kelereng kaisar, man’an dan irwan apabila jumlah kelereng wanwans diketahui ? ***Buatlah program dengan c# !***

1. Diberikan flowchart sebagai berikut, hasil yang tercetak pada bilangan tersebut adalah ?



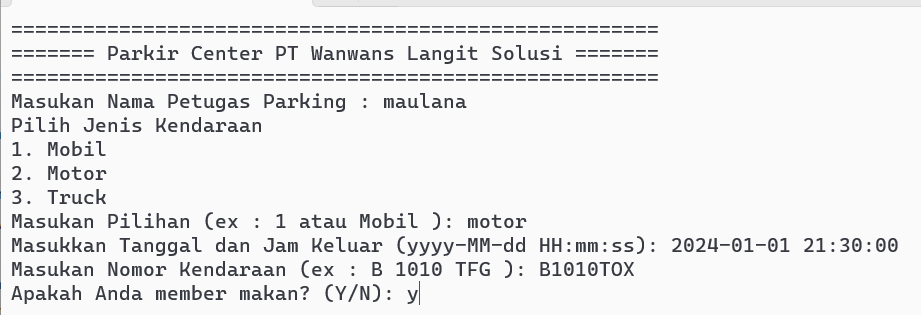
1. ***Permasalahaan***

Sebuah pengelola parkir pada PT Wanwans Langit Solusi memberlakukan ketentuan tarif sebagai beriku :

* Untuk tarif Motor pada 1 jam pertama dikenaka tarif parkir sebesar Rp.2000 /jam, sedangkan pada jam berikutnya dikenakan tarif parkir sebesar Rp.1000 /jam.
* Untuk tarif Mobil pada 1 jam pertama dikenaka tarif parkir sebesar Rp.5000 /jam, sedangkan pada jam berikutnya dikenakan tarif parkir sebesar Rp.2000 /jam.
* Untuk tarif Truck atau Bus pada 1 jam pertama dikenaka tarif parkir sebesar Rp.10000 /jam, sedangkan pada jam berikutnya dikenakan tarif parkir sebesar Rp.5000 /jam.
* Jika mamber makan diberikan diskon sebasar Rp 2000, jikan bukan mamber makan tidak diberikan diskon

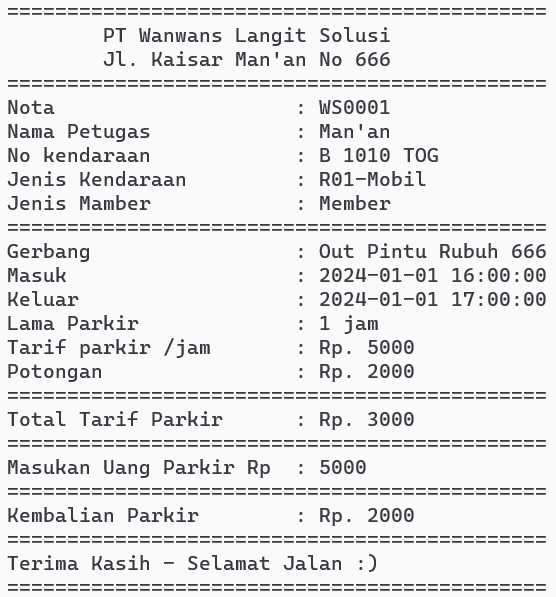
**Identifikasi masalah**

* **Input :**
* Input Nama Petugas Parkir
* Input Pilih Jenis Kendaraan (Motor,Mobil,Truck / Bus)
* Input Nomor Kendaraan
* Tanggal dan Waktu masuk parkir diperoleh saat melakukan inputan data parkir secara realtime
* Tanggal dan Jam ke (2024-01-01 16:39:00)
* Input Pilih Jenis member Inputan yang diharapkan :



* Output :
* Tampilkan Nomer Nota (Nomor nota dibuat secara automatis 10 digit gabungan huruf dan angka contoh : WS001)
* Tampilkan Nama Petugas Parkir
* Tampilkan Nomor Kendaraan dan jenis kendaraan
* Tampilkan Jenis Member
* Tampilkan Tanggal dan Jam Masuk
* Tampilkan Tanggal dan Jam keluar
* Tampilkan Lama Parkir dalam hitungan waktu (contoh : 1 jam 1 menit)
* Tampilkan Tarif Parkir Kendaraan
* Tampilkan Potongan Member
* Tampilkan Total Tarif Parkir
* Input Uang Bayar Parkir
* Tampilkan Kembalian Uang Bayar Parkir

Output Yang di Harapkan



* Diperbolehkan menggunakan OOP atau Tidak Menggunakan OOP

1. **Permasalahan :**

Buatlah sebuah program untuk menggambar pola segitiga dengan sebuah karakter. Karakter dapat berupa huruf atau karakter khusus (ex : \*,#,%,@,&,dll)

**Identifikasi Masalah :**

* Masukan terdiri dari dua baris, baris pertama untuk menginput bilangan bulat N dengan 1≤ N ≤ 100. Baris kedua masukan sebuah karakter yang diinginkan
* Keluaran program adalah huruf/krakter khusu yang menggambarkan pola segitiga sama kaki

1. ***Permasalahan :***

Buatlah program tentang lagu anak ayam sebagai berikut :

Tek Kotek Kotek Kotek, Anak Ayam turun berkotek

Anak ayam turunlah 5 mati satu tinggallah 4

Anak ayam turunlah 4 mati satu tinggallah 3

Anak ayam turunlah 3 mati satu tinggallah 1

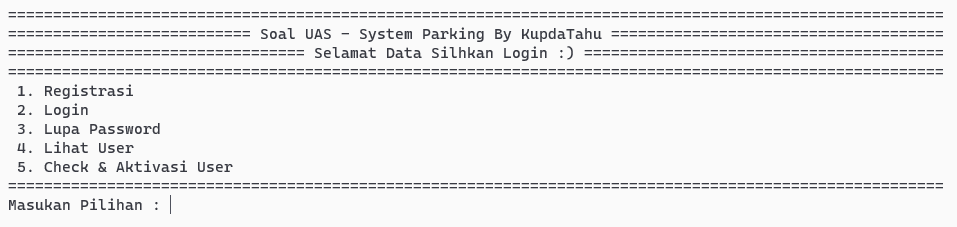
Anak ayam turunlah 2 mati satu tinggallah 1

Anak ayam turunlah 1 mati satu tinggallah induknya

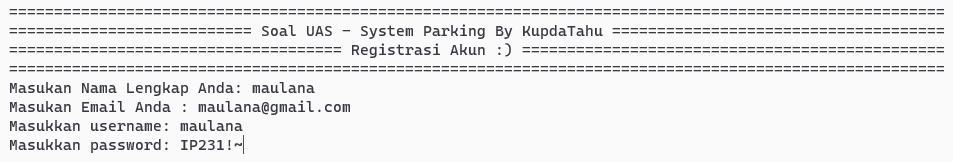
***Identifikasi masalah :***

* Masukan dari program adalah bilangan bulat N denganbatasan: 1≤ N ≤ 100.
* Keluaran program adalah berupa baris lirik lagu yangakan diuraikan sesuai dengan bilangan bulat yangdiinput

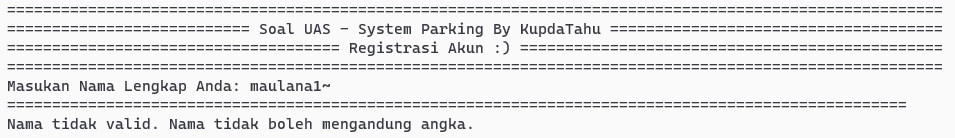
1. Buatlah sebuah program Login System Parking By KupadTahu



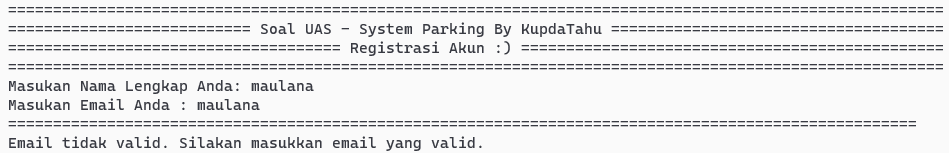
***Studi Kasus 1 Register :***



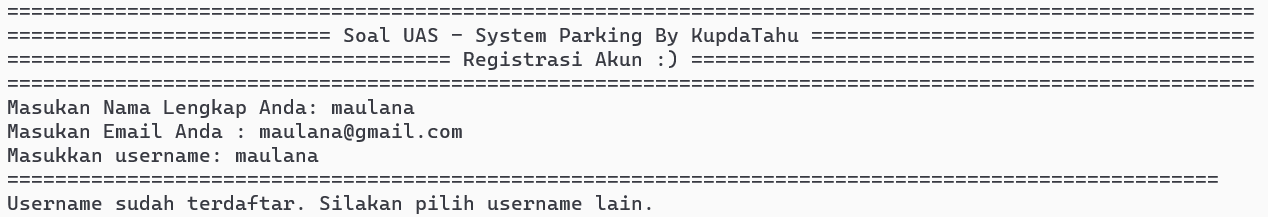
* Registrasi memiliki atribut : ID (dengan auto generate, dengan 5 karakter 2 huruf dan 3 angka secara acak), Nama Lengkap, Email, Username, Password di hidden dengan ***(\*)***, status : ***Aktif & Tidak Aktif***
* Buatlah inputan : Nama Lengkap, Email, username, password.
* Buatlah validasi :
* Nama tidak boleh mengandung spesial karakter dan angka, contoh berikut :



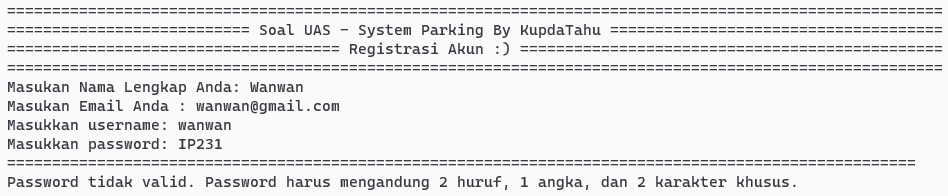
* Penulisan email harus benar, contoh sebagai berikut :



* Email tidak boleh sama dengan email user yang sudah terdaftar.
* Username tidak boleh sama dengan username yang sudah terdaftar. Contoh :



* Password harus mengandung 2 huruf, 1 angka, 2 Karakter Khusus. Contoh :



* Jika data tidak sesuai maka data tidak bisa disimpan.
* Tampilan Jika Berhasil Registrasi

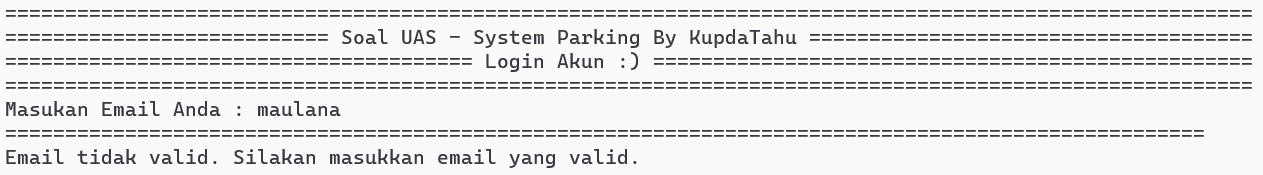


* Lakukan aktivasi, jika ya maka status user menjadi ***Aktif ,*** jika tidak maka status user ***Tidak Aktif***
* Tampilkan menu awal.

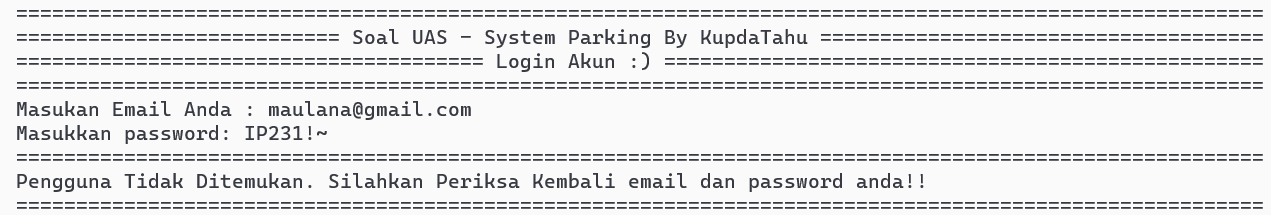
***Studi kasus 2 : Menu Login***

* Buatlah inputan : username dan password
* Buatlah validasi :
* Check format penulisan email

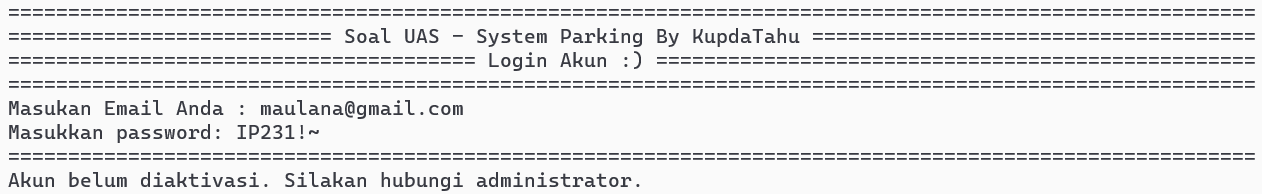
Output : ***” Email tidak valid. Silakan masukkan email yang valid.”***



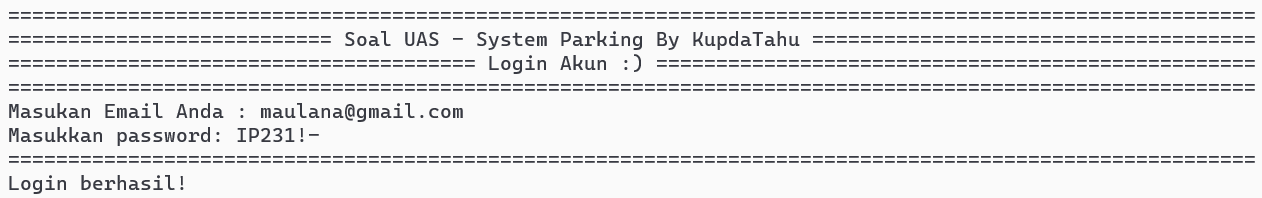
* Check apakah email sudah terdaftar jika blm terdaftar silahkan lakukan pendaftaran Output Notif : ***” Pengguna Tidak Ditemukan. Silahkan Periksa Kembali username / email dan password anda!!”***



* Check apakah status user ***Aktif*** atua ***Tidak Aktif***

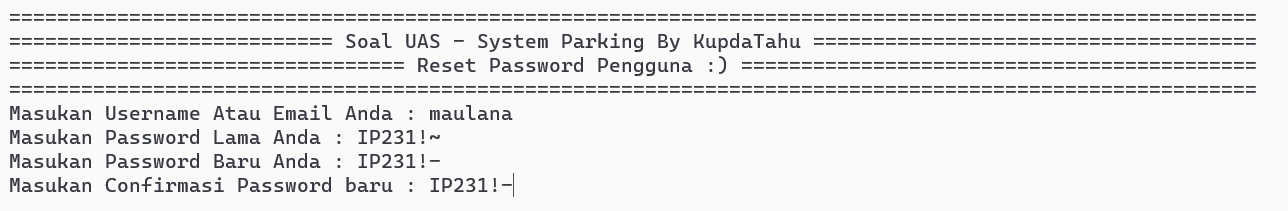


* Check username dan password sudah sesuai atau tidak sesuai, jika tidak sesuai makan tidak dapat login, salah password tidak boleh lebih dari 3 kali makan user di ***Tidak Aktifkan***
* Tampilan Login Jika berhasil Login

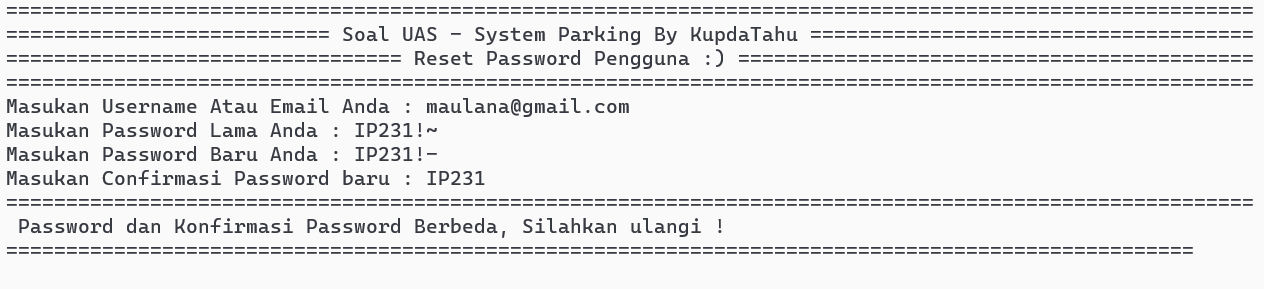


***Studi Kasus 3 : Menu Reset Password***

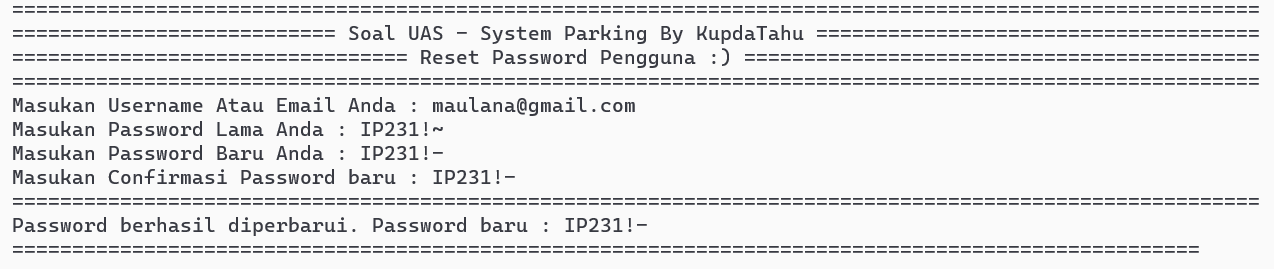
* Buatlah inputan :
* Masukan Username atau Email.
* Masukan Password Lama Anda
* Masukan Password Baru Anda
* Masukan Konfirmasi Password Baru Anda



* Buatlah validasi :
* Password harus mengandung 2 huruf, 1 angka, 2 Karakter Khusus.
* Konfirmasi Password haru sama dengan Password baru. Jika tidak maka error, tampilan sebagai berikut :

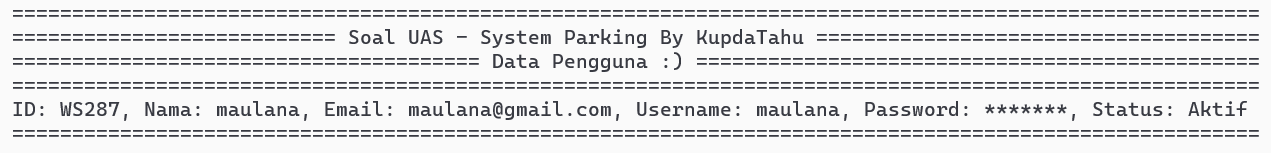


* Tampilan jika berhasil Reset Password



***Studi Kasus 4 : Tampilan User Pengguna***

* Buatlah program untuk menampilkan pengguna



***Studi Kasus 5 : Check & Aktivasi User***

* Buatlah Inputan :
  + Masukan Username atau Email Anda
* Buatlah validasi :
* Jika user pengguna sudah ***Aktif***, maka tampilkan program sebagai berikut :



* Jika user pengguna ***Tidak Aktif*** , maka tampilkan program sebagi berikut :

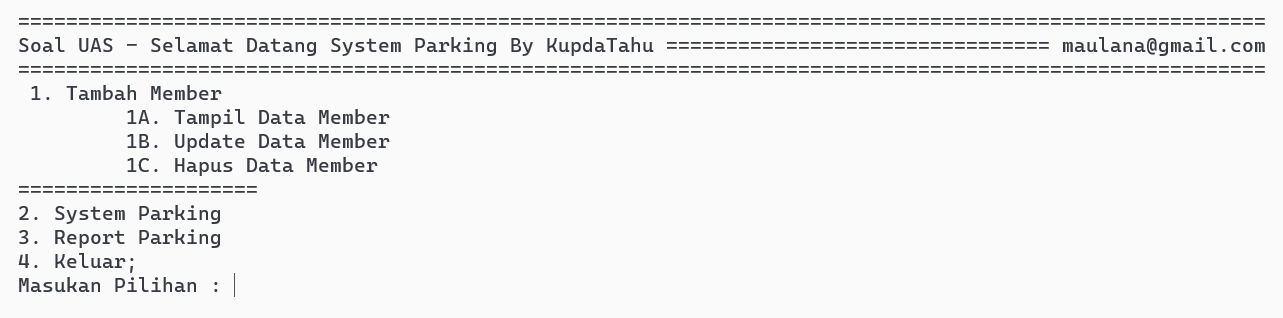


* Jika ya maka tampil sebagai berikut :



* Jika Tidak maka, kembali ke menu utama.

1. Buatlah System Parking by KupadTahu diatas menjadi lebih lengkap :
2. Ketika Login berhasil maka buat program sebagai berikut :



1. Tambah Member :

* Input data :
  + Id Member dengan auto generate Password dengan 5 karkter (3 huruf dan 2 angka)
  + Nama member
  + noHandphone
  + noKendaraan
  + Jenis Kendaraan
  + Periode Awal member
  + Periode Berakhir member
  + Status Member, secara default ***Tidak Aktif***
* Validasi Data :
* Nama Member tidak Boleh sama dengan nama member yang sudah terdaftar
* Nomer kendaraan tidak boleh sama dengan nomer kendaraan yang sudah terdaftar
* 1 member hanya boleh memeliki 2 jenis kendaraan yaitu mobil 1 dan motor 1, tidak diperbolehkan jenis kendaraan yang sama
* Periode awal member diperoleh dari tanggal dan waktu realtime
* Periode akhir member diperoleh dari tangaal dan waktu periode awal ditambah 1 hari

1. Tampil Member :

Tampilkan data mamber :

Id mamber, nama mamber, no handphone, no kendaraan, jenis kendaraan, periode awal, dan periode berakhir.

1. Update Member

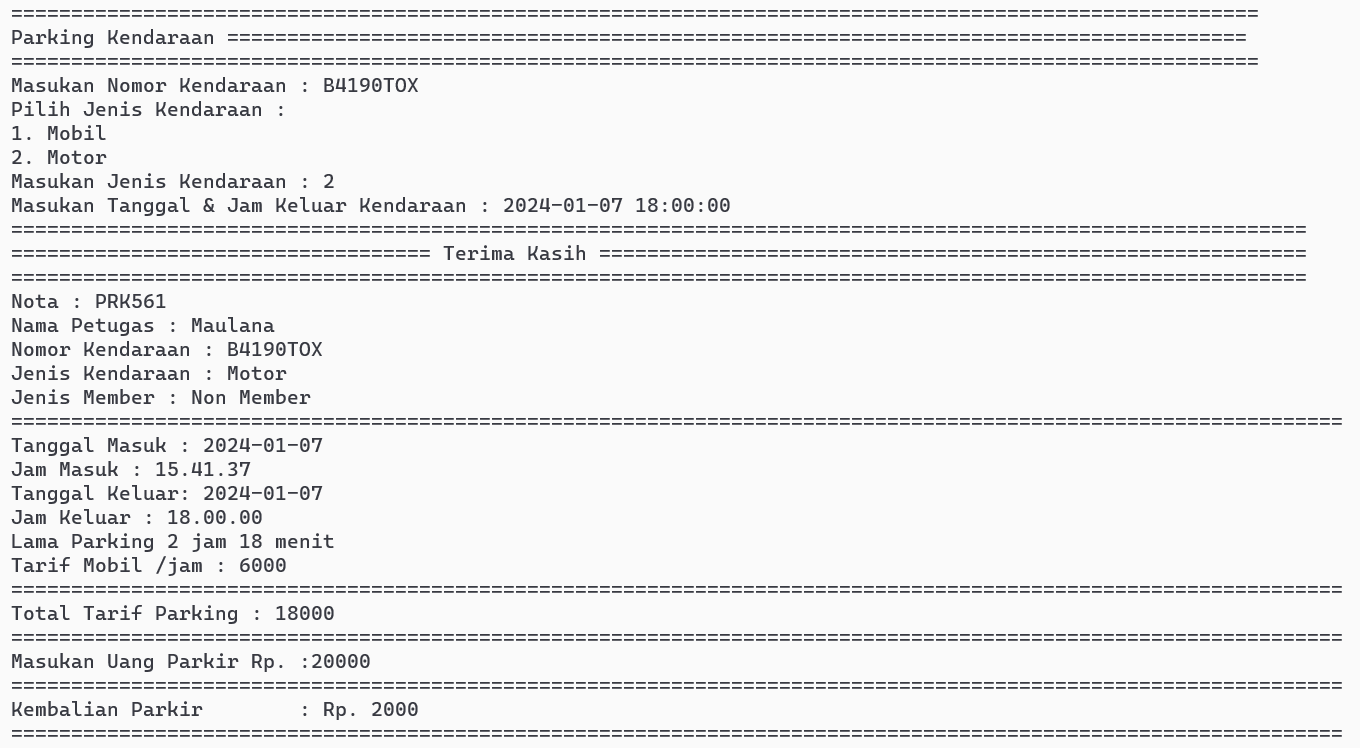
* Input data :
  + Masukan ID Mamber
  + Masukan Tanggal Periode Awal Member
* Update data :
* Tanggal Periode Awal Member sesuai dengan tanggal yang diinput dan Tanggal Periode Berakhir Member ditambah 1 hari dari tanggal awal member yang diupdate
* Status Member ***Aktif*** & ***Tidak Aktif***

1. Hapus Member

* Input Data :
* Masukan ID Mamber
* Maka Data member terhapus.

1. System Parking

* Input Data :
  + Masukan Nomor Kendaraan
  + Pilih Jenis Kendaraan : Mobil / Motor
  + Masukan Tanggal & Jam Keluar Kendaraan
* Output :



* Nota dibuat secara auto generate 5 karakte (3 huruf 2 angka)
* Nama Petugas , diperoleh dari data user login
* Nomor kendaraan diperoleh dari data Nomor Kendaraan
* Jenis Kendaraan diperoleh dari data jenis kendaraan
* Jenis mamber :
  + Jika Nomor Kendaraan terdaftar member maka jenis member ***Mamber*** dan memperoleh penguraan biaya parkir sebesar 2000 untuk jenis kendaraan ***Motor*** Ataupun ***Mobil***
  + Jika Nomor Kendaraan tidak terdaftar member maka jenis member ***Non Mamber*** dan tidak memperoleh pengurangan biaya parkir
* Lama Parking dengan format /jam dan /menit seperti tampilan diatas
* Tarif Mobil : Rp. 6. 000 dan Tarif Motor : Rp. 3.000
* Total Tarif Parking : Jam keluar – jam masuk \* tarif mobil.
* Kembalian Parkir : Uang Bayar – total tarif parking

1. Report Parking :

* Tampilan data Parking :
* Nota
* Nama Petugas
* Nomor Kendaraan
* Jenis Kendaraan
* Tanggal Masuk
* Jam Masuk
* Tanggal Keluar
* Jam Keluar
* Jenis Member
* Total Tarif Mobil

Nama : Stevanus Andika Galih Setiawan

NIM : 202303110008

Prodi : Rekayasa Perangkat Lunak (RPL)

Mata Kuliah : Dasar Pemrograman

Semester : 1 (satu)

Tanggal : 20 Januari 2024

1. using System;

class HitungBelanjaan

{

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Harga 1 kilogram telur : Rp. 28.000 ");

Console.WriteLine("Harga 1 kilogram gula merah : Rp.12.000 ");

Console.WriteLine("Harga 1 kilogram tepung terigu : Rp. 8.000 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Input berat dan harga barang

Console.Write("Berat Telur (kg): ");

double beratTelur = Convert.ToDouble(Console.ReadLine());

Console.Write("Berat Gula Merah (kg): ");

double beratGulaMerah = Convert.ToDouble(Console.ReadLine());

Console.Write("Berat Tepung Terigu (kg): ");

double beratTepungTerigu = Convert.ToDouble(Console.ReadLine());

Console.Write("Harga Telur (/kg): ");

double hargaTelur = Convert.ToDouble(Console.ReadLine());

Console.Write("Harga Gula Merah (/kg): ");

double hargaGulaMerah = Convert.ToDouble(Console.ReadLine());

Console.Write("Harga Tepung Terigu (/kg): ");

double hargaTepungTerigu = Convert.ToDouble(Console.ReadLine());

// Input biaya transportasi

Console.Write("Biaya Transportasi (sekali naik angkot): ");

double biayaTransportasi = Convert.ToDouble(Console.ReadLine());

// Input uang yang dimiliki Wanwans

Console.Write("Uang Wanwans: ");

double uangWanwans = Convert.ToDouble(Console.ReadLine());

// Input uang ongkos pulang Wanwans

Console.Write("Uang Ongkos Pulang Wanwans: ");

double uangOngkosPulang = Convert.ToDouble(Console.ReadLine());

// Hitung total biaya belanja

double totalBiayaBelanja = (beratTelur \* hargaTelur) + (beratGulaMerah \* hargaGulaMerah) + (beratTepungTerigu \* hargaTepungTerigu);

// Hitung total biaya dengan tambahan biaya transportasi

double totalBiayaTotal = totalBiayaBelanja + biayaTransportasi;

// Hitung total biaya dengan tambahan uang ongkos pulang

totalBiayaTotal += uangOngkosPulang;

// Hitung sisa uang Wanwans

double sisaUang = uangWanwans - totalBiayaTotal;

// Output sisa uang Wanwans

Console.WriteLine($"Total Biaya Belanja: {totalBiayaBelanja}");

Console.WriteLine($"Total Biaya dengan Transportasi dan Ongkos Pulang: {totalBiayaTotal}");

Console.WriteLine($"Sisa Uang Wanwans: {sisaUang}");

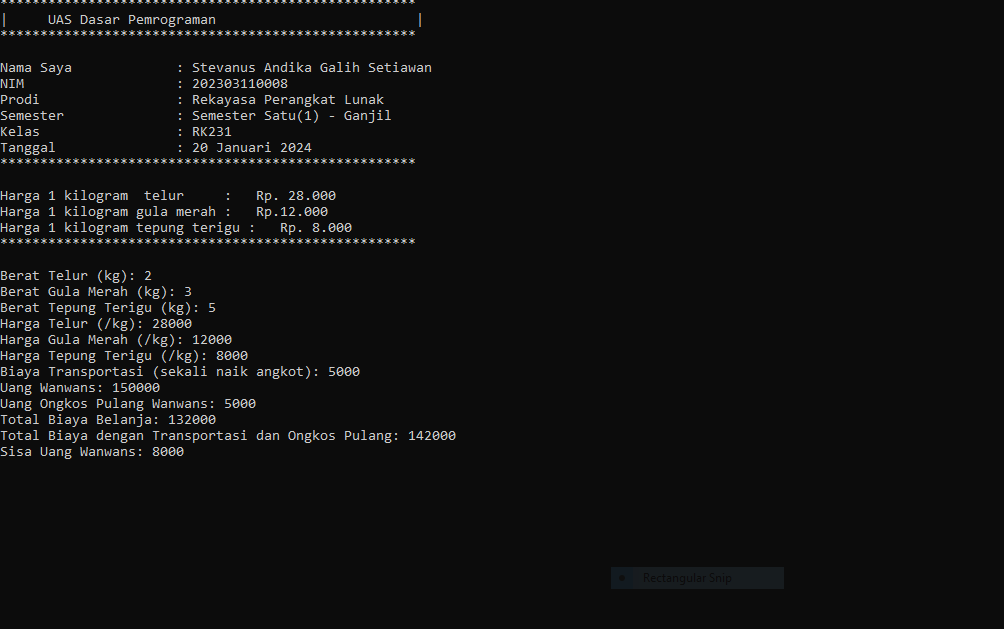
// Tunggu input sebelum menutup aplikasi

Console.ReadLine();

}

}

Output :



2. using System;

class BeratSemangkaDanHarga

{

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Input harga per kilogram semangka

Console.Write("Harga per Kilogram Semangka: ");

double hargaPerKilogram = Convert.ToDouble(Console.ReadLine());

// Input berat semangka yang dibeli oleh pembeli

Console.Write("Berat Semangka (kg): ");

double beratSemangka = Convert.ToDouble(Console.ReadLine());

// Hitung total harga yang harus dibayar oleh pembeli

double totalHarga = hargaPerKilogram \* beratSemangka;

// Output harga yang harus dibayar oleh pembeli

Console.WriteLine($"Harga yang harus dibayar: Rp. {totalHarga}");

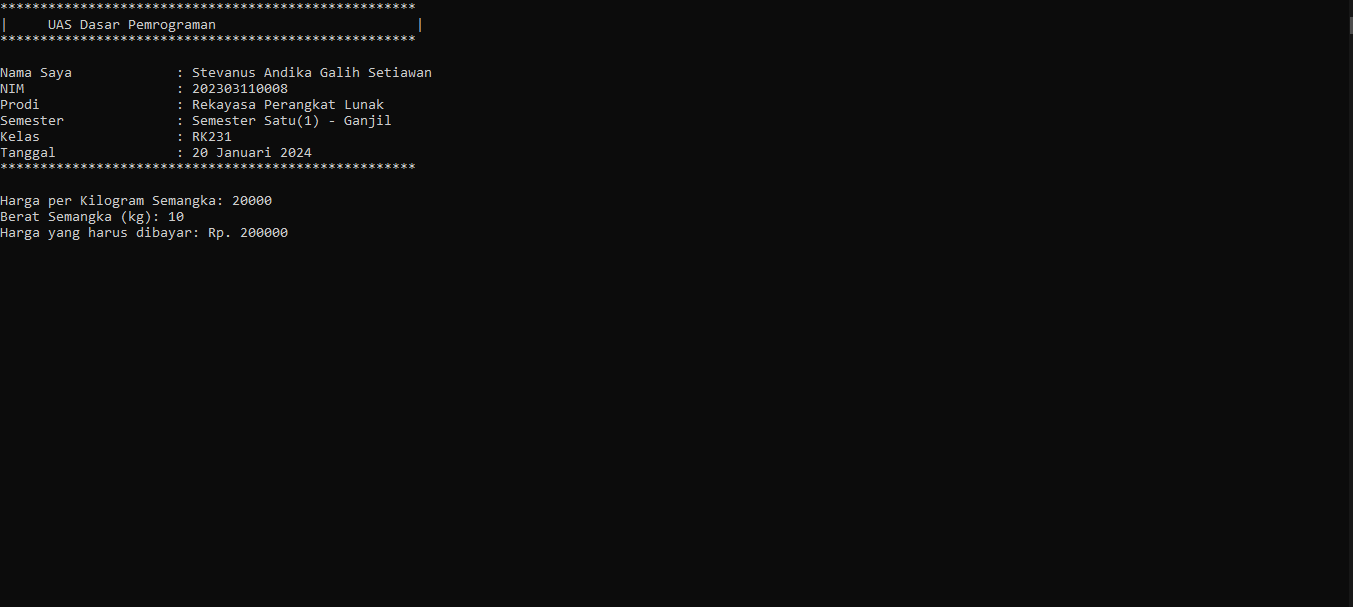
// Tunggu input sebelum menutup aplikasi

Console.ReadLine();

}

}

Output :



3. using System;

class ProgramUrutArray

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Array dengan data yang diberikan

int[] dataArray = { 10, 60, 20, 100, 40 };

// Panggil fungsi untuk menemukan nilai terbesar

int nilaiTerbesar = TemukanNilaiTerbesar(dataArray);

// Output nilai terbesar

Console.WriteLine($"Nilai Terbesar dalam Array: {nilaiTerbesar}");

// Tunggu input sebelum menutup aplikasi

Console.ReadLine();

}

static int TemukanNilaiTerbesar(int[] array)

{

// Inisialisasi nilai terbesar dengan nilai pertama dalam array

int nilaiTerbesar = array[0];

// Iterasi melalui array untuk mencari nilai terbesar

for (int i = 1; i < array.Length; i++)

{

if (array[i] > nilaiTerbesar)

{

nilaiTerbesar = array[i];

}

}

// Mengembalikan nilai terbesar

return nilaiTerbesar;

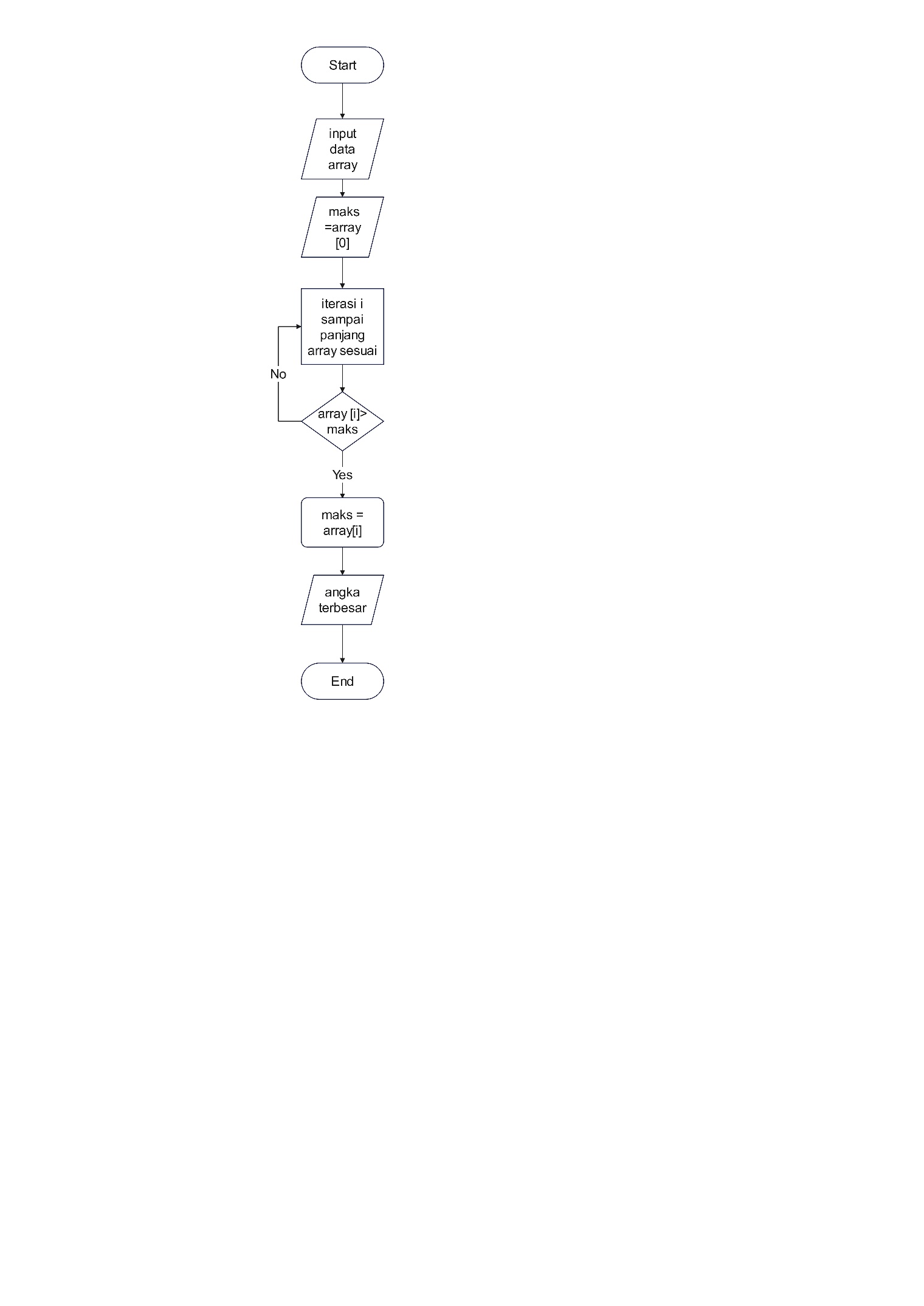
}

}

Output :



Flowchart :



4. using System;

class Program

{

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Jumlah kelereng Wanwans

int jumlahKelerengWanwans = 20;

// Menghitung jumlah kelereng Man'an

int jumlahKelerengManan = 3 \* jumlahKelerengWanwans;

// Menghitung jumlah kelereng Kaisar

int jumlahKelerengKaisar = jumlahKelerengWanwans;

// Menghitung jumlah kelereng Irwan

int jumlahKelerengIrwan = jumlahKelerengWanwans - 10;

// Menampilkan hasil

Console.WriteLine("Jumlah kelereng Kaisar: " + jumlahKelerengKaisar);

Console.WriteLine("Jumlah kelereng Man'an: " + jumlahKelerengManan);

Console.WriteLine("Jumlah kelereng Irwan: " + jumlahKelerengIrwan);

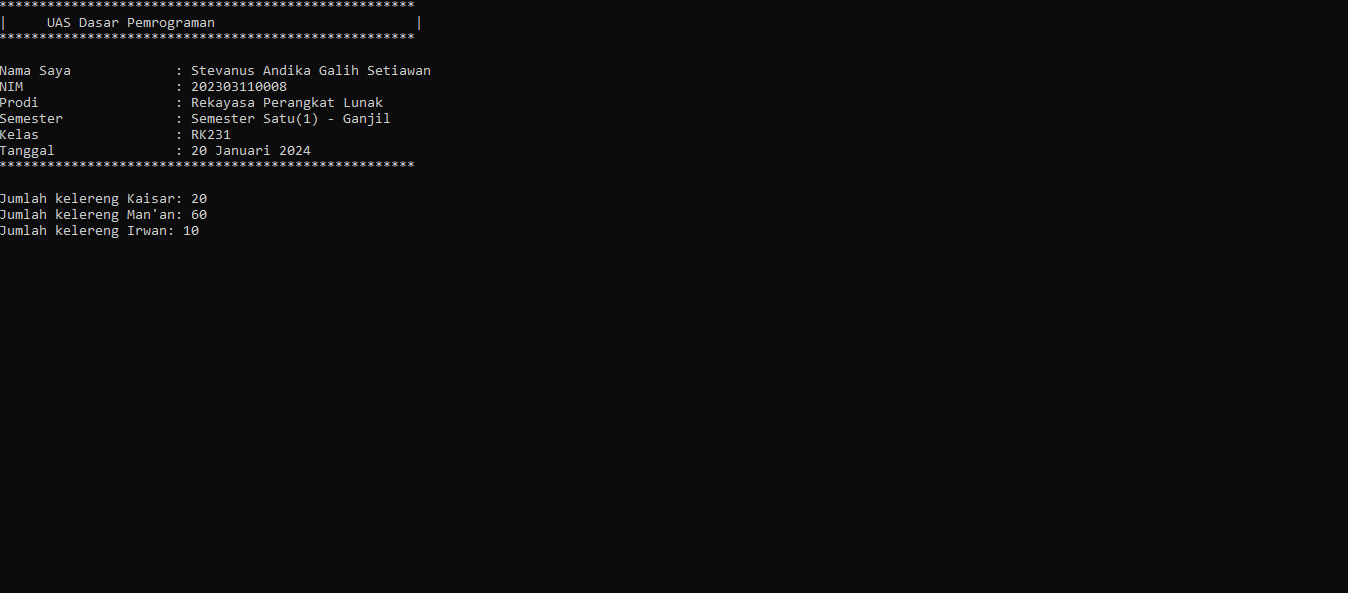
// Menunggu input pengguna sebelum menutup aplikasi

Console.ReadLine();

}

}

Ouput :



5.

using System;

class Program

{

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Inisialisasi variabel

int faktor = 0;

int bilangan;

int a;

// Input bilangan

Console.Write("Masukkan bilangan: ");

bilangan = int.Parse(Console.ReadLine());

// Inisialisasi variabel a

a = 1;

// Lakukan pengecekan

while (a <= bilangan)

{

faktor = bilangan % a == 0 ? faktor + 1 : faktor;

a++;

// Jika faktor > 2, cetak bukan bilangan prima dan keluar dari loop

if (faktor > 2)

{

Console.WriteLine("Bukan bilangan prima");

break;

}

}

// Jika faktor == 2, cetak bilangan prima

if (faktor == 2)

{

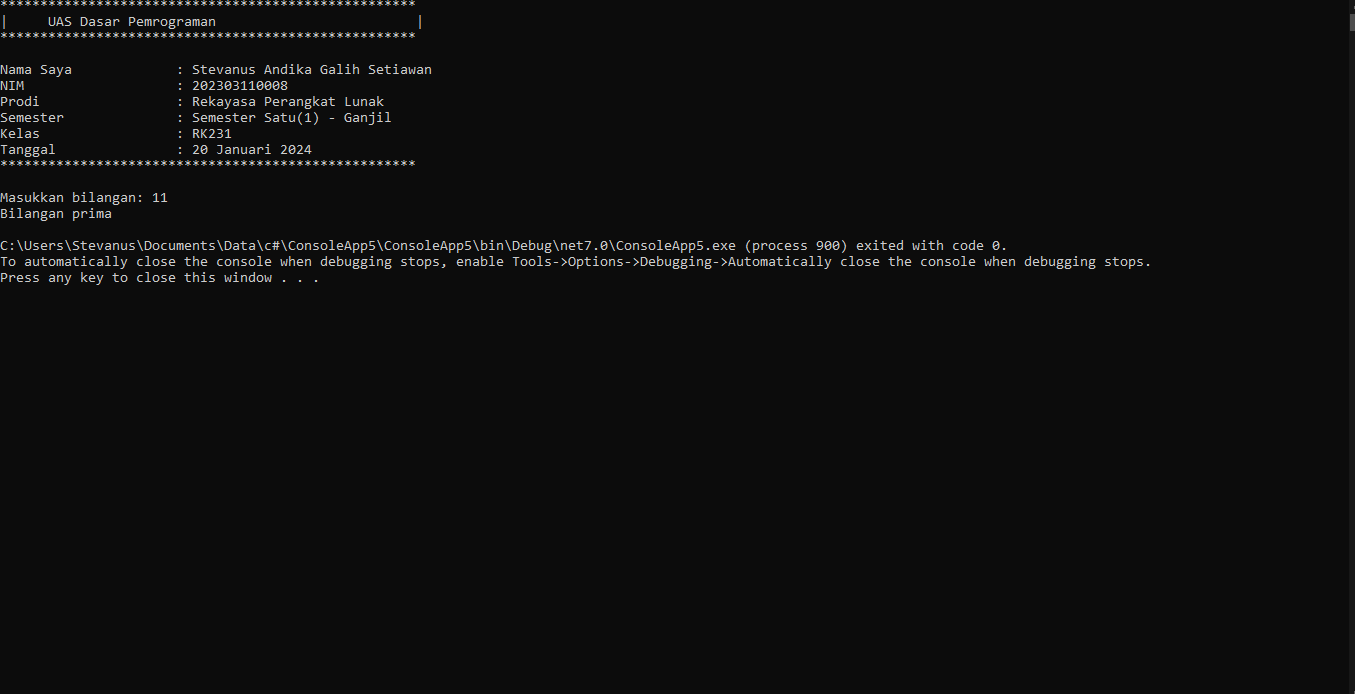
Console.WriteLine("Bilangan prima");

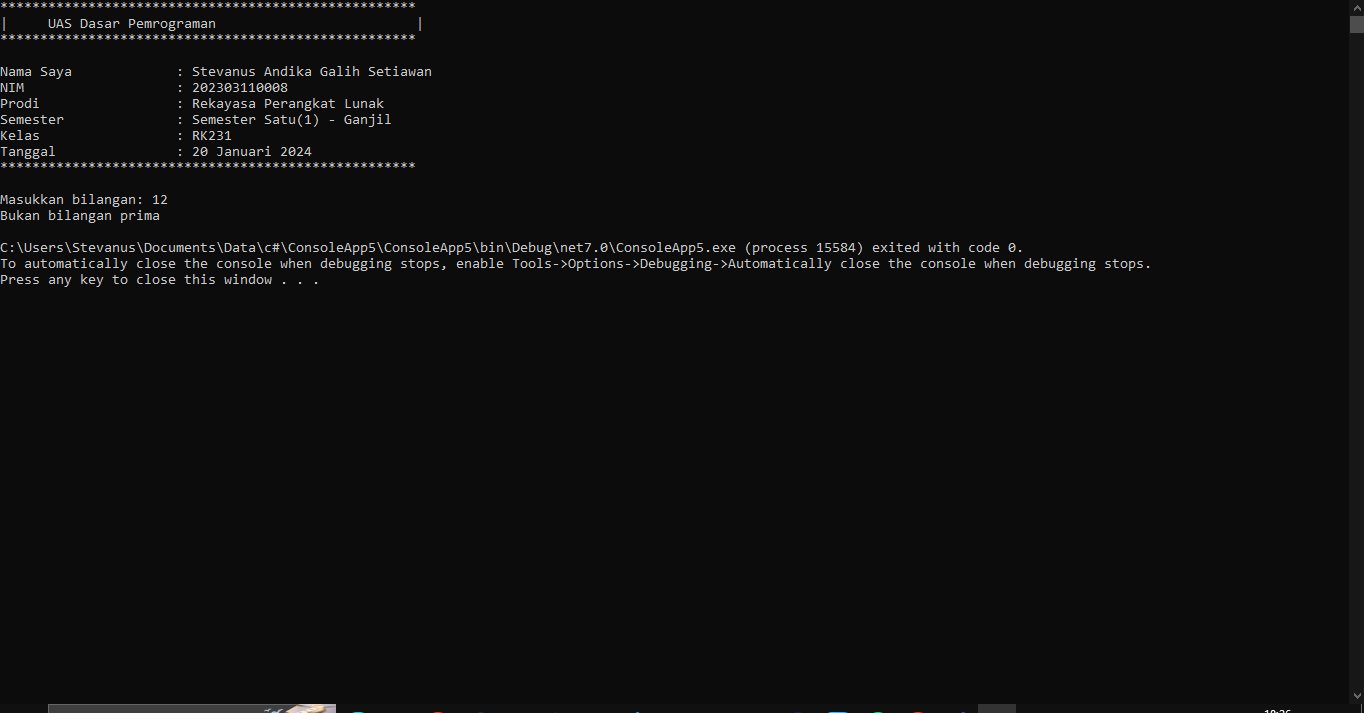
}

}

}

Output :





6. using System;

// Definisikan enum untuk jenis kendaraan

enum JenisKendaraan

{

Mobil = 1,

Motor,

Truck

}

class HitungBiayaParkir

{

// Properti

public string NamaPetugas { get; set; }

public string NomorKendaraan { get; set; }

public JenisKendaraan JenisKendaraan { get; set; }

public bool IsMember { get; set; }

public DateTime JamMasuk { get; set; }

public DateTime JamKeluar { get; set; }

// Konstruktor

public HitungBiayaParkir(string namaPetugas, string nomorKendaraan, JenisKendaraan jenisKendaraan, bool isMember)

{

NamaPetugas = namaPetugas;

NomorKendaraan = nomorKendaraan;

JenisKendaraan = jenisKendaraan;

IsMember = isMember;

JamMasuk = DateTime.Now;

}

// Metode untuk menghitung biaya parkir

public void HitungBiaya()

{

// Hitung lama parkir dalam hitungan waktu

TimeSpan lamaParkir = JamKeluar - JamMasuk;

// Tentukan tarif parkir sesuai jenis kendaraan dan member

int tarifJamPertama = 0;

int tarifJamBerikutnya = 0;

switch (JenisKendaraan)

{

case JenisKendaraan.Mobil:

tarifJamPertama = 5000;

tarifJamBerikutnya = 2000;

break;

case JenisKendaraan.Motor:

tarifJamPertama = 2000;

tarifJamBerikutnya = 1000;

break;

case JenisKendaraan.Truck:

tarifJamPertama = 10000;

tarifJamBerikutnya = 5000;

break;

}

// Hitung tarif parkir

int totalTarif = tarifJamPertama + (int)Math.Ceiling(lamaParkir.TotalHours - 1) \* tarifJamBerikutnya;

// Hitung potongan member jika ada

int potonganMember = IsMember ? 2000 : 0;

// Output informasi parkir

Console.WriteLine("===================================================");

Console.WriteLine(" PT Wanwans Langit Solusi");

Console.WriteLine("===================================================");

Console.WriteLine($"Nota : {GenerateNota()}");

Console.WriteLine($"Nama Petugas : {NamaPetugas}");

Console.WriteLine($"No.Kendaraan : {NomorKendaraan}");

Console.WriteLine($"Jenis Kendaraan : {JenisKendaraan}");

Console.WriteLine($"Gerbang : Out Pintu Rubuh 666");

Console.WriteLine($"Masuk : {JamMasuk.ToString("yyyy-MM-dd HH:mm:ss")}");

Console.WriteLine($"Keluar : {JamKeluar.ToString("yyyy-MM-dd HH:mm:ss")}");

Console.WriteLine($"Lama Parkir : {FormatLamaParkir(lamaParkir)}");

Console.WriteLine($"Tarif parkir/jam: {tarifJamPertama} (1 jam pertama) + {tarifJamBerikutnya} per jam berikutnya");

Console.WriteLine($"Potongan : {potonganMember}");

Console.WriteLine("===================================================");

Console.WriteLine($"Total tarif parkir : {totalTarif}");

Console.WriteLine("===================================================");

Console.Write("Masukkan uang parkir : RP");

int uangBayar = int.Parse(Console.ReadLine());

Console.WriteLine("===================================================");

Console.WriteLine($"Kembalian Parkir : {uangBayar - totalTarif}");

Console.WriteLine("===================================================");

Console.WriteLine("Terimakasih Selamat Jalan :}");

Console.WriteLine("===================================================");

}

// Metode untuk menghasilkan nomor nota secara otomatis

private string GenerateNota()

{

// Menggunakan kombinasi dari huruf WS dan angka acak

Random rand = new Random();

return $"WS{rand.Next(100, 999)}";

}

// Metode untuk memformat lama parkir dalam jam dan menit

private string FormatLamaParkir(TimeSpan lamaParkir)

{

return $"{(int)lamaParkir.TotalHours} jam {lamaParkir.Minutes} menit";

}

}

class Program

{

static void Main()

{

// Input data dari pengguna

Console.WriteLine("===================================================");

Console.WriteLine("=======Parking Center PT Wawans Langit Solusi=======");

Console.WriteLine("===================================================");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.Write("Masukkan Nama Petugas Parking : ");

string namaPetugas = Console.ReadLine();

Console.WriteLine("Pilih jenis kendaraan :");

Console.WriteLine("1. Mobil");

Console.WriteLine("2. Motor");

Console.WriteLine("3. Truck");

Console.Write("Masukkan pilihan (misalkan : 1 untuk mobil): ");

int pilihanJenisKendaraan = int.Parse(Console.ReadLine());

JenisKendaraan jenisKendaraan = (JenisKendaraan)pilihanJenisKendaraan;

Console.Write("Masukkan nomor kendaraan (ex: B 1010 TFG): ");

string nomorKendaraan = Console.ReadLine();

Console.WriteLine("Apakah anda member :");

Console.WriteLine("1. Ya");

Console.WriteLine("2. Tidak");

Console.Write("Masukkan Pilihan Anda (misalkan 1 atau Ya): ");

int pilihanMember = int.Parse(Console.ReadLine());

bool isMember = (pilihanMember == 1);

// Buat objek HitungBiayaParkir

HitungBiayaParkir hitungBiayaParkir = new HitungBiayaParkir(namaPetugas, nomorKendaraan, jenisKendaraan, isMember);

// Input jam keluar

Console.WriteLine("===================================================");

Console.WriteLine("Masukkan Tanggal dan Jam keluar (yyyy-MM-dd HH:mm:ss): ");

string inputJamKeluar = Console.ReadLine();

hitungBiayaParkir.JamKeluar = DateTime.ParseExact(inputJamKeluar, "yyyy-MM-dd HH:mm:ss", null);

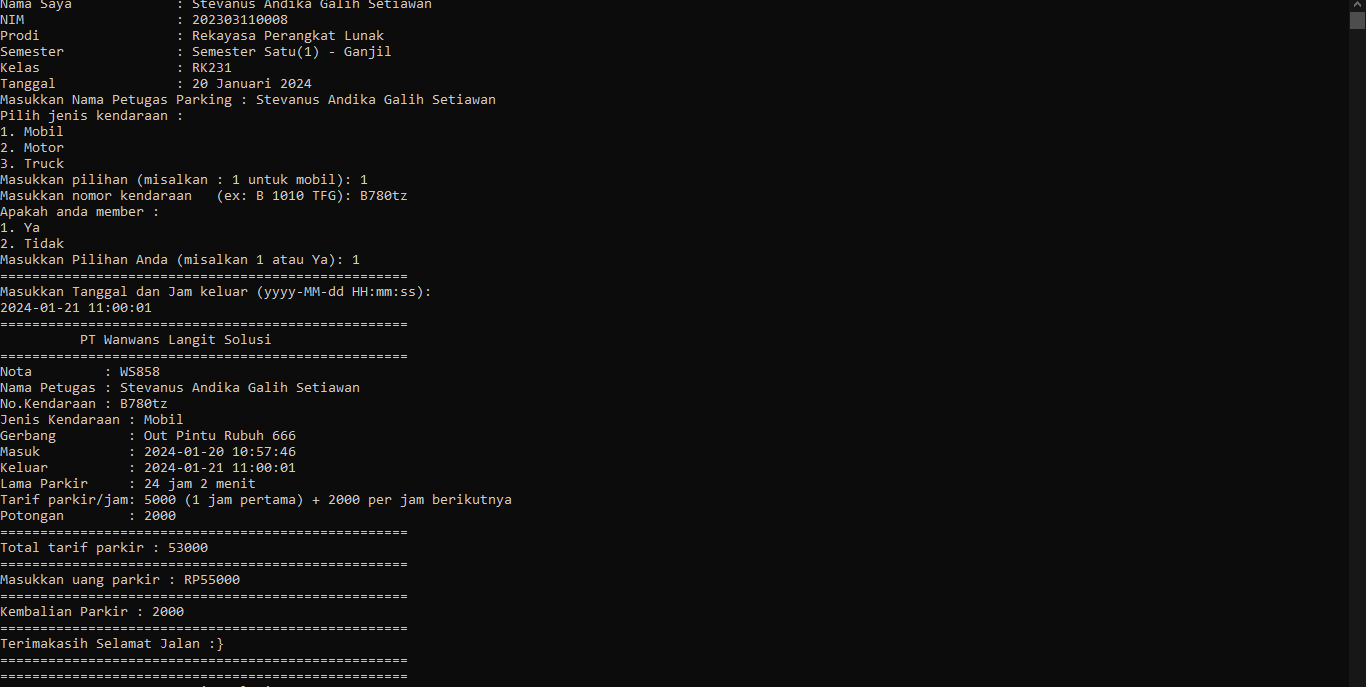
// Hitung biaya parkir dan tampilkan output

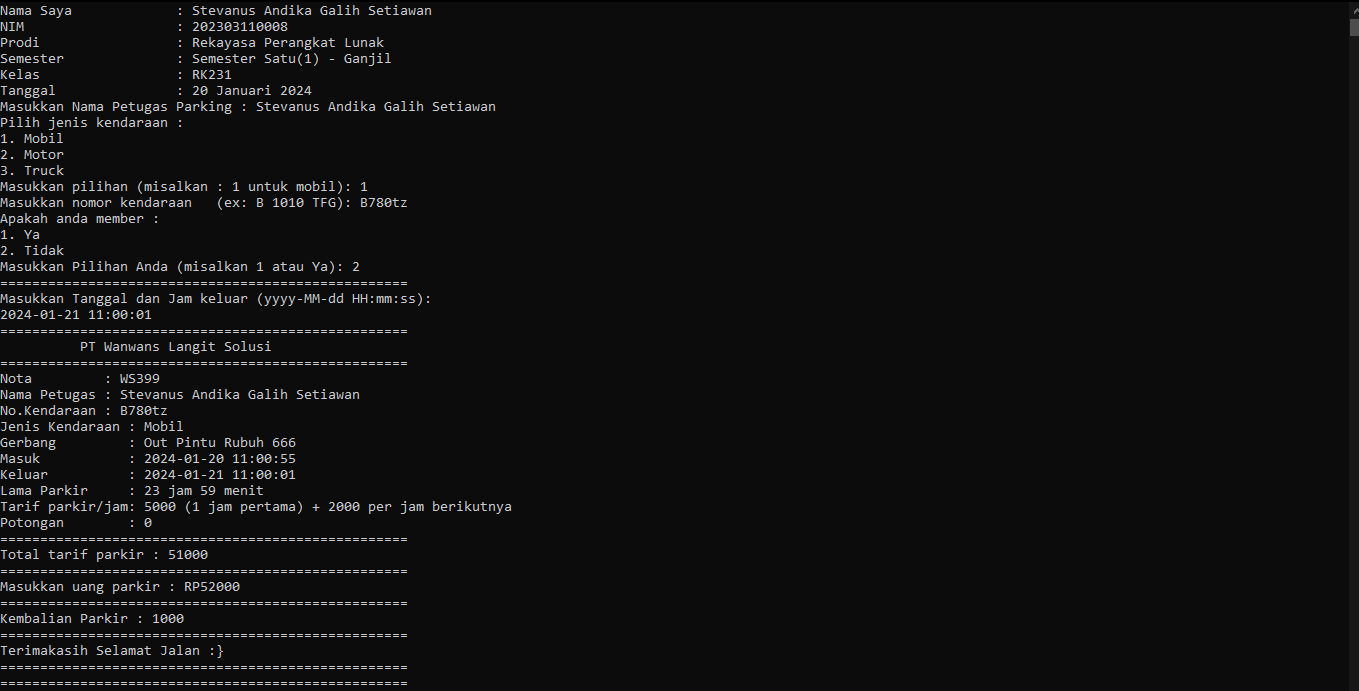
hitungBiayaParkir.HitungBiaya();

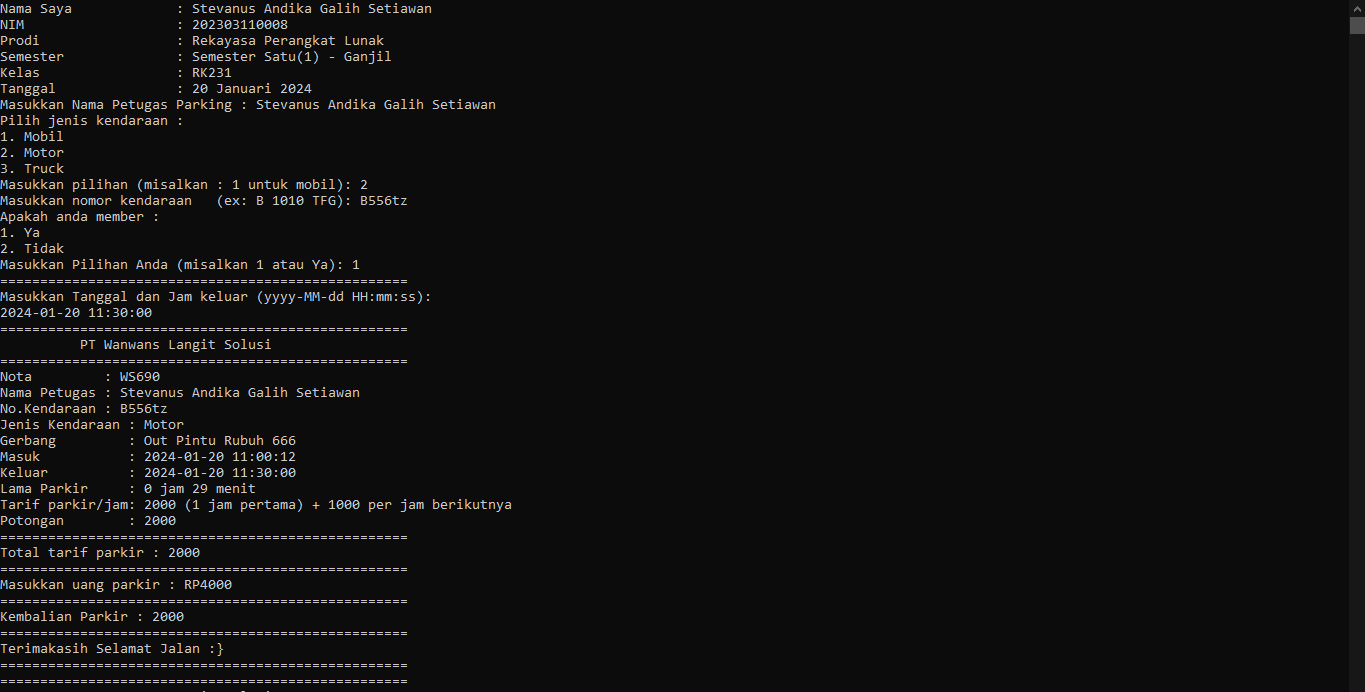
}

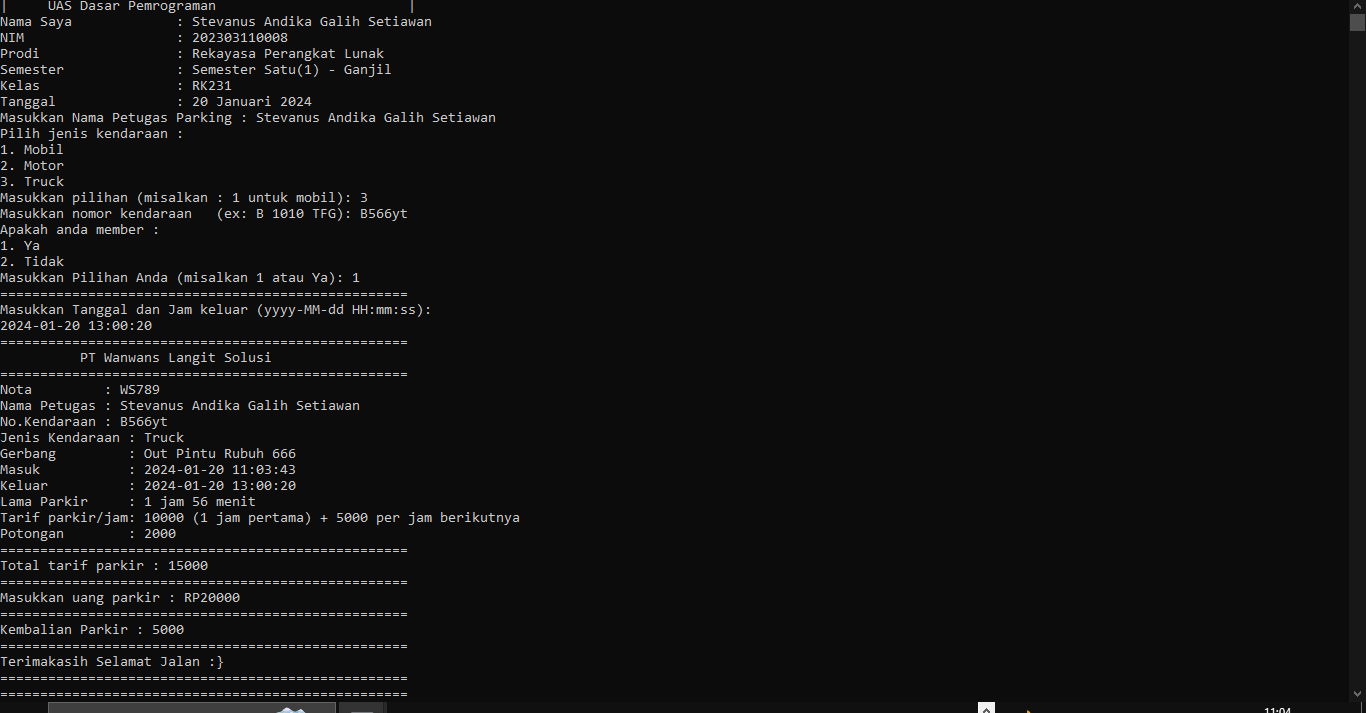
}

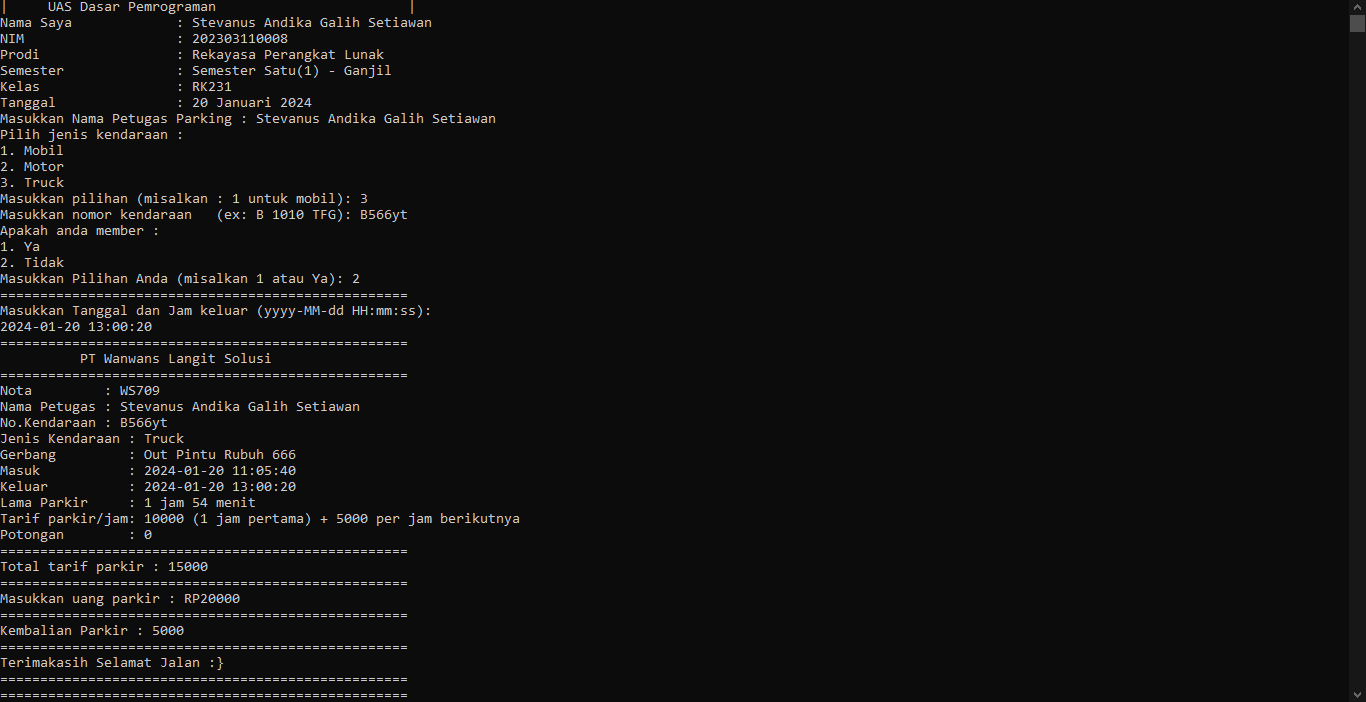
Output :











7. using System;

class loopingSimbol

{

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Input jumlah baris

Console.Write("Masukkan jumlah baris (1 ≤ N ≤ 100): ");

int n = int.Parse(Console.ReadLine());

// Input karakter yang diinginkan

Console.Write("Masukkan karakter yang diinginkan: ");

char karakter = char.Parse(Console.ReadLine());

// Membuat pola segitiga

for (int i = 1; i <= n; i++)

{

// Membuat spasi sebelum karakter

for (int j = 1; j <= n - i; j++)

{

Console.Write(" ");

}

// Menampilkan karakter sebanyak (2 \* i - 1)

for (int k = 1; k <= 2 \* i - 1; k++)

{

Console.Write(karakter);

}

// Pindah ke baris berikutnya

Console.WriteLine();

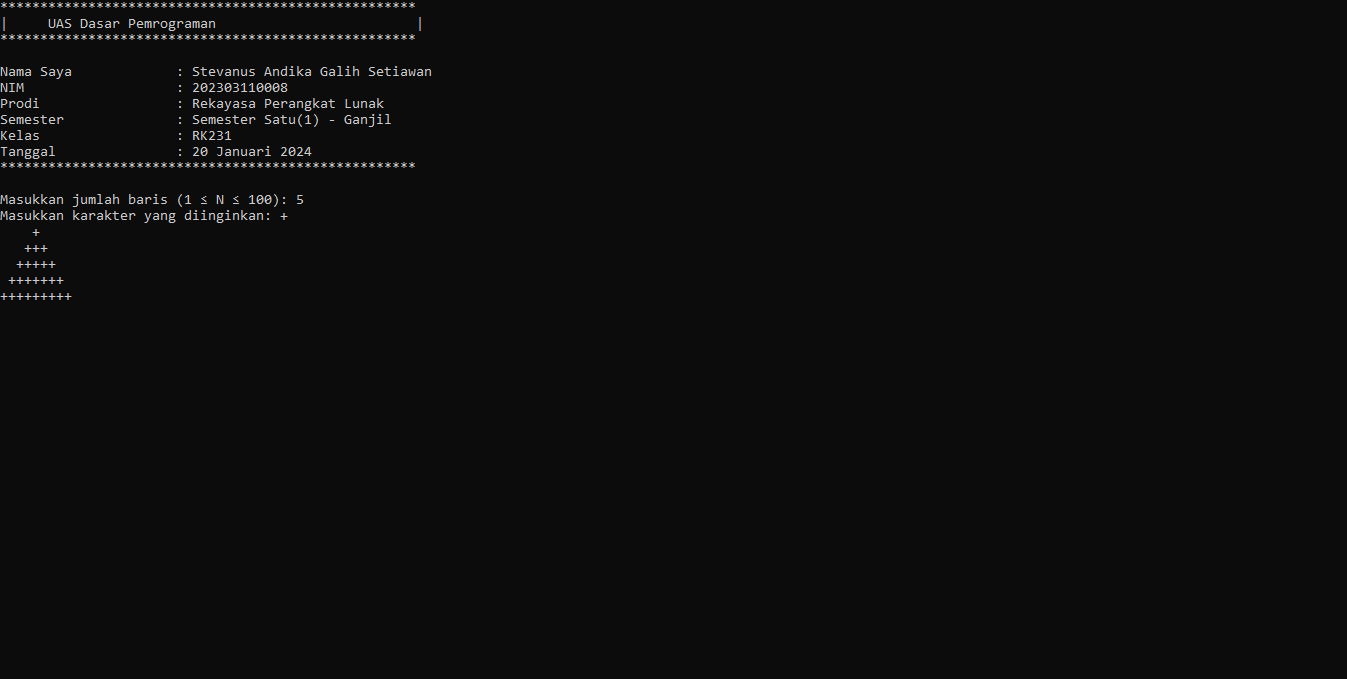
}

Console.ReadLine(); // Untuk menjaga agar konsol tidak langsung tertutup

}

}

Output :





8.using System;

class Program

{

static void Main()

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

// Input jumlah anak ayam

Console.Write("Masukkan jumlah anak ayam (1 ≤ N ≤ 100): ");

int n = int.Parse(Console.ReadLine());

// Memastikan bahwa jumlah anak ayam sesuai batasan

if (n < 1 || n > 100)

{

Console.WriteLine("Masukan tidak valid. Jumlah anak ayam harus antara 1 dan 100.");

return;

}

// Membuat lirik lagu

for (int i = n; i >= 1; i--)

{

Console.WriteLine($"Anak ayam turunlah {i} mati satu tinggallah {i - 1}");

}

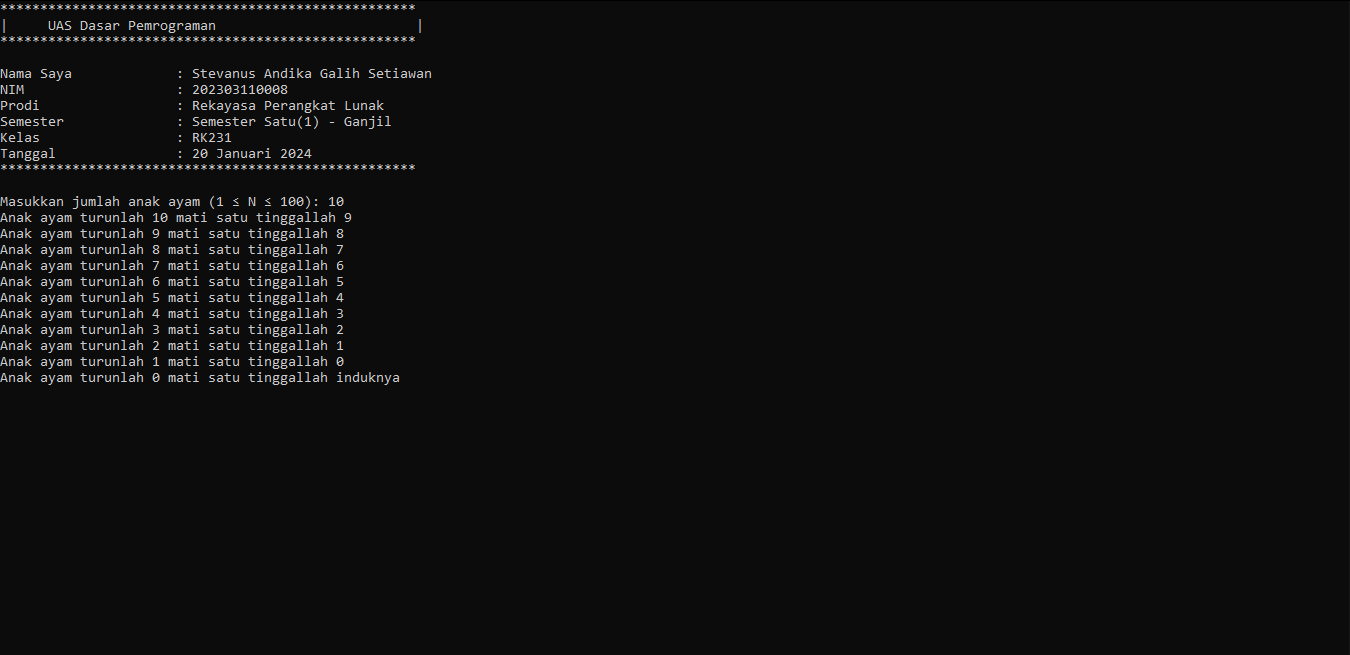
Console.WriteLine("Anak ayam turunlah 0 mati satu tinggallah induknya");

Console.ReadLine(); // Untuk menjaga agar konsol tidak langsung tertutup

}

}

Output :



9.

using System;

using System.Collections.Generic;

using System.Linq;

class User

{

//semua data yang diperlukan didefinisikan disini

//gunakan set dan get untuk mengatur data

public string ID { get; set; }

public string FullName { get; set; }

public string Email { get; set; }

public string Username { get; set; }

public string Password { get; set; }

public bool IsActive { get; set; }

}

class UserManager

// class usermanager untuk menampung data user

{

private List<User> users = new List<User>();

public void RegisterUser()

{

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("======================== Registrasi Akun :) ===========================");

Console.WriteLine("====================================================================================");

Console.Write("Masukkan Nama Lengkap Anda : ");

string fullName = Console.ReadLine();

// Validasi Nama

if (!IsValidName(fullName))

{

Console.WriteLine("Nama tidak valid, nama tidak mengandung angka.");

return;

}

Console.Write("Masukkan Email Anda : ");

string email = Console.ReadLine();

// Validasi Email

if (!IsValidEmail(email))

{

Console.WriteLine("Email tidak valid, silahkan masukkan email yang valid.");

return;

}

// Validasi Email Unik

if (IsEmailExists(email))

{

Console.WriteLine("Email sudah terdaftar, silahkan gunakan email lain.");

return;

}

Console.Write("Masukkan Username : ");

string username = Console.ReadLine();

// Validasi Username Unik

if (IsUsernameExists(username))

{

Console.WriteLine("Username sudah terdaftar, silahkan pilih username lain.");

return;

}

Console.Write("Masukkan Password : ");

string password = Console.ReadLine();

// Validasi Password

if (!IsValidPassword(password))

{

Console.WriteLine("Password tidak valid, harus mengandung 2 huruf, 1 angka, dan 2 karakter khusus.");

return;

}

// Generate ID secara otomatis

string generatedID = GenerateUserID();

User newUser = new User

{

ID = generatedID,

FullName = fullName,

Email = email,

Username = username,

Password = password,

IsActive = false // Set default status to inactive

};

users.Add(newUser);

Console.WriteLine("====================================================================================");

Console.WriteLine("Registrasi Berhasil !");

Console.WriteLine("====================================================================================");

Console.WriteLine("Silahkan Konfirmasi Akun Anda");

Console.WriteLine("1. Ya");

Console.WriteLine("2. Tidak");

Console.Write("Masukkan Pilihan : ");

string confirmationChoice = Console.ReadLine();

if (confirmationChoice == "1" || confirmationChoice.ToLower() == "ya")

{

ActivateUser(newUser);

}

}

public void Login()

{

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("======================== Login Akun :) ============================");

Console.WriteLine("====================================================================================");

Console.Write("Masukkan Email Anda : ");

string inputEmail = Console.ReadLine();

// Validasi Format Email

if (!IsValidEmail(inputEmail))

{

Console.WriteLine("Email tidak valid! Silahkan masukkan email yang valid.");

return;

}

User user = GetUserByEmail(inputEmail);

// Validasi User Ditemukan

if (user == null)

{

Console.WriteLine("Pengguna Tidak Ditemukan. Silahkan Periksa Kembali email dan password anda!!");

return;

}

// Validasi Akun Aktif

if (!user.IsActive)

{

Console.WriteLine("Akun belum diaktivasi, silahkan hubungi administrator.");

return;

}

Console.Write("Masukkan Password : ");

string inputPassword = Console.ReadLine();

// Validasi Password Sesuai

if (inputPassword != user.Password)

{

Console.WriteLine("Username atau Password salah. Silahkan coba lagi.");

return;

}

Console.WriteLine("====================================================================================");

Console.WriteLine("Login Berhasil !");

Console.WriteLine("====================================================================================");

}

public void ResetPassword()

{

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("=======================Reset Password :) ============================");

Console.WriteLine("====================================================================================");

Console.Write("Masukkan Username/Email Anda : ");

string inputIdentifier = Console.ReadLine();

// Validasi Format Email

if (!IsValidEmail(inputIdentifier))

{

Console.WriteLine("Email tidak valid! Silahkan masukkan email yang valid.");

return;

}

User user = GetUserByEmail(inputIdentifier);

// Validasi User Ditemukan

if (user == null)

{

Console.WriteLine("Pengguna Tidak Ditemukan. Silahkan coba lagi.");

return;

}

Console.Write("Masukkan Password Lama Anda : ");

string oldPassword = Console.ReadLine();

// Validasi Password Lama Sesuai

if (oldPassword != user.Password)

{

Console.WriteLine("Password lama tidak sesuai. Silahkan coba lagi.");

return;

}

Console.Write("Masukkan Password Baru Anda : ");

string newPassword = Console.ReadLine();

// Validasi Password Baru tampilkan notifikasi

if (!IsValidPassword(newPassword))

{

Console.WriteLine("Password tidak valid, harus mengandung 2 huruf, 1 angka, dan 2 karakter khusus.");

return;

}

Console.Write("Masukkan Konfirmasi Password Baru Anda : ");

string confirmPassword = Console.ReadLine();

// Validasi Konfirmasi Password,jika tidak sama tampilkan notifikasi

if (newPassword != confirmPassword)

{

Console.WriteLine("Password dan konfirmasi password tidak sama. Silahkan coba lagi.");

return;

}

// Update Password

user.Password = newPassword;

Console.WriteLine("====================================================================================");

Console.WriteLine("Password berhasil diperbarui, password baru anda: " + newPassword);

Console.WriteLine("====================================================================================");

}

public void DisplayUsers()

{

//tampilkan data menggunakan loop dari list User dengan nama lain users

//terdiri dari id,nama,email,username,password dan status

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("=======================Data Pengguna :) ============================");

Console.WriteLine("====================================================================================");

Console.WriteLine("ID\t\t\tNAMA\t\t\tEMAIL\t\t\tUSERNAME\t\t\tPASSWORD\t\t\tSTATUS");

foreach (User user in users)

{ //tampilkan data

Console.WriteLine($"{user.ID}\t\t\t{user.FullName}\t\t\t{user.Email}\t\t\t{user.Username}\t\t\t{user.Password}\t\t\t{(user.IsActive ? "Aktif" : "Tidak Aktif")}");

}

Console.WriteLine("====================================================================================");

}

public void CheckAndActivateUser()

{

//lakukan indetifikasi dan aktivasi user

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("=======================Check & Aktivasi User :) ============================");

Console.WriteLine("====================================================================================");

Console.Write("Masukkan Username atau Email Anda : ");

string inputIdentifier = Console.ReadLine();

// Validasi Format Email

if (!IsValidEmail(inputIdentifier))

{

Console.WriteLine("Email tidak valid! Silahkan masukkan email yang valid.");

return;

}

User user = GetUserByEmail(inputIdentifier);

// Validasi User Ditemukan

if (user == null)

{

Console.WriteLine("Pengguna Tidak Ditemukan. Silahkan coba lagi.");

return;

}

//tampilkan data dan notifikasi untuk aktivasi akun

Console.WriteLine("====================================================================================");

Console.WriteLine($"Informasi pengguna untuk email {user.Email}");

Console.WriteLine($"Nama: {user.FullName}");

Console.WriteLine($"Email: {user.Email}");

Console.WriteLine($"Status: {(user.IsActive ? "Aktif" : "Tidak Aktif")}");

Console.WriteLine("====================================================================================");

Console.WriteLine("Aktivasi Akun Anda??");

Console.WriteLine("1. Ya");

Console.WriteLine("2. Tidak");

Console.Write("Masukkan Pilihan : ");

string confirmationChoice = Console.ReadLine();

if (confirmationChoice == "1" || confirmationChoice.ToLower() == "ya")

{

ActivateUser(user);

}

}

private bool IsValidName(string name)

{

//jika nama valid

//kembalikan output dalam format character

return !name.Any(char.IsDigit);

}

private bool IsValidEmail(string email)

{

//jika email valid

//kembalikan output dalam format email

//jika tidak ada email yang dicari, maka error

try

{

var addr = new System.Net.Mail.MailAddress(email);

return addr.Address == email;

}

catch

{

return false;

}

}

private bool IsEmailExists(string email)

{

return users.Any(user => user.Email == email);

}

private bool IsUsernameExists(string username)

{

return users.Any(user => user.Username == username);

}

private bool IsValidPassword(string password)

{

//minimal 6 karakter, 2 huruf, 1 angka, dan 2 karakter khusus

//jika dapat return hasil

return password.Length >= 6 && password.Count(char.IsLetter) >= 2 &&

password.Count(char.IsDigit) >= 1 && password.Count(ch => !char.IsLetterOrDigit(ch)) >= 2;

}

private string GenerateUserID()

{

// generate user id baru

return Guid.NewGuid().ToString().Substring(0, 5).ToUpper();

}

//method untuk mencari user berdasarkan email dengan params email

private User GetUserByEmail(string email)

{

return users.FirstOrDefault(user => user.Email == email);

}

private void ActivateUser(User user)

{

user.IsActive = true;

Console.WriteLine("Selamat! Akun anda sudah aktif.");

}

}

class Program

{

static void Main()

//nametag informasi kelas

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("| UAS Dasar Pemrograman |");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("Nama Saya : Stevanus Andika Galih Setiawan");

Console.WriteLine("NIM : 202303110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Semester : Semester Satu(1) - Ganjil");

Console.WriteLine("Kelas : RK231");

Console.WriteLine("Tanggal : 20 Januari 2024 ");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

//jalankan method dan panggil constructor baru dengan keyword new

UserManager userManager = new UserManager();

//jalankan looping dalam kondisi true

while (true)

{

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("======================== Selamat Datang Silahkan Login :)===========================");

Console.WriteLine("====================================================================================");

Console.WriteLine("1. Registrasi");

Console.WriteLine("2. Login");

Console.WriteLine("3. Lupa Password");

Console.WriteLine("4. Liat user");

Console.WriteLine("5. Check & aktivasi User");

Console.WriteLine("===================================================================================");

Console.Write("Masukkan pilihan : ");

string choice = Console.ReadLine();//baca inputan user

//lakukan decision berdasarkan inputan user

switch (choice)

{

case "1":

userManager.RegisterUser();

break;

case "2":

userManager.Login();

break;

case "3":

userManager.ResetPassword();

break;

case "4":

userManager.DisplayUsers();

break;

case "5":

userManager.CheckAndActivateUser();

break;

default:

Console.WriteLine("Pilihan tidak valid. Silahkan coba lagi.");

break;

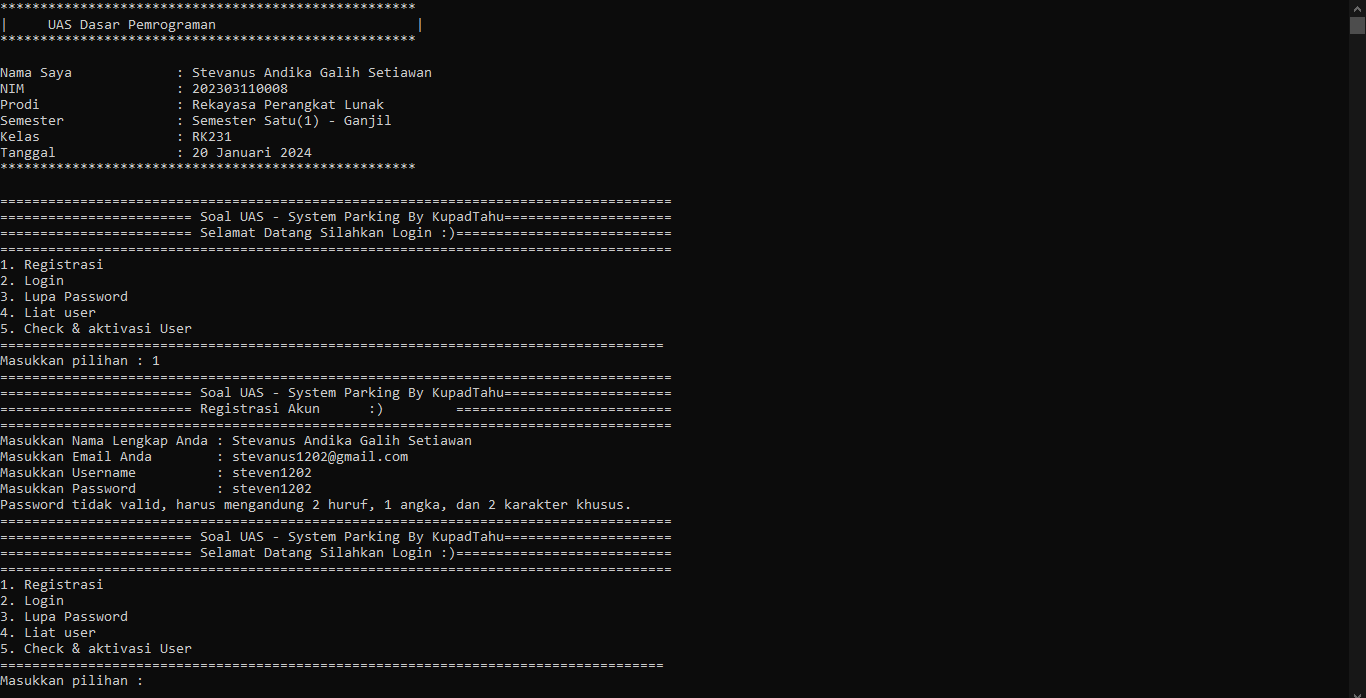
}

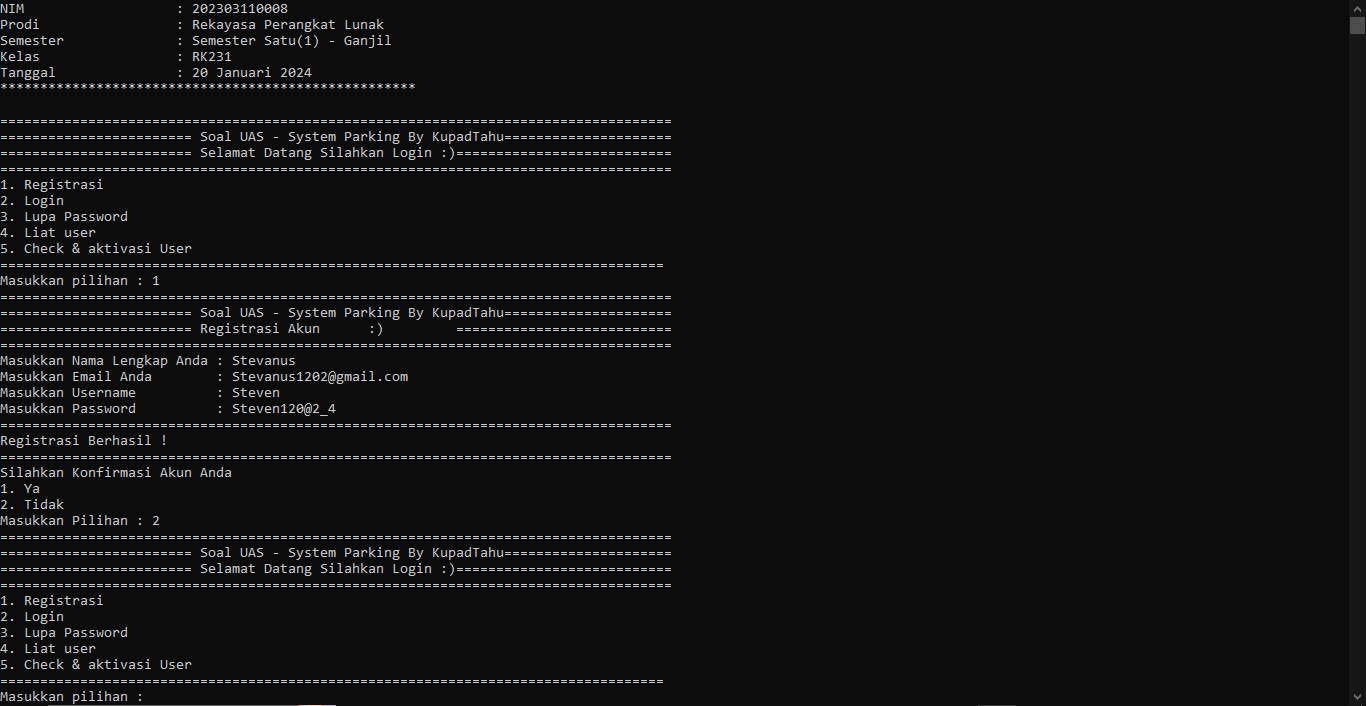
}

}

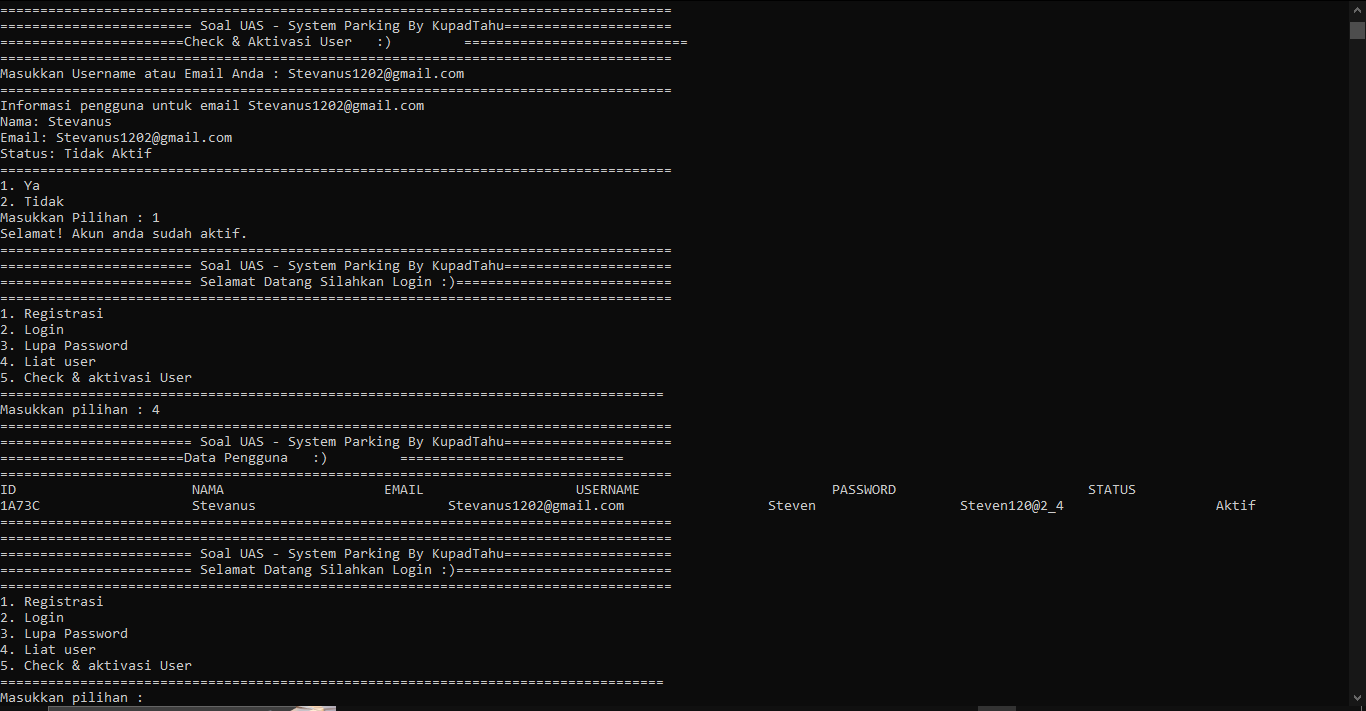
}

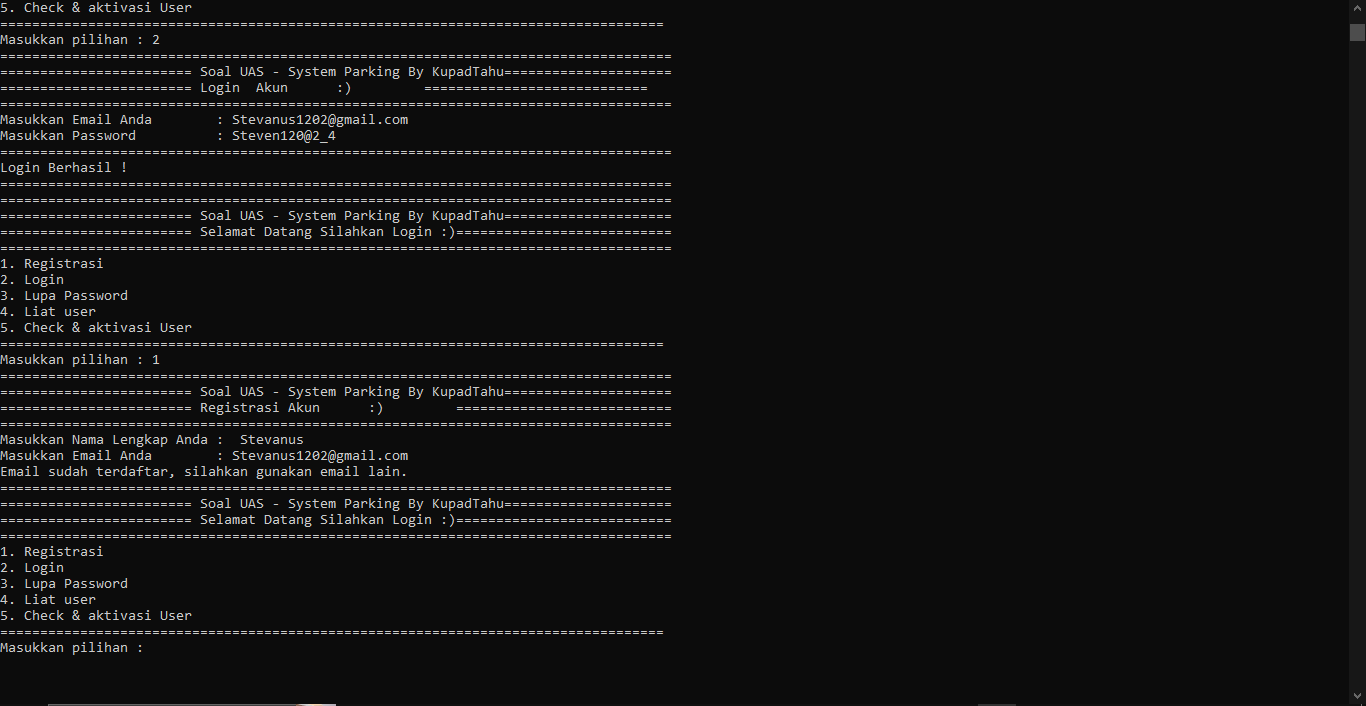
Output :

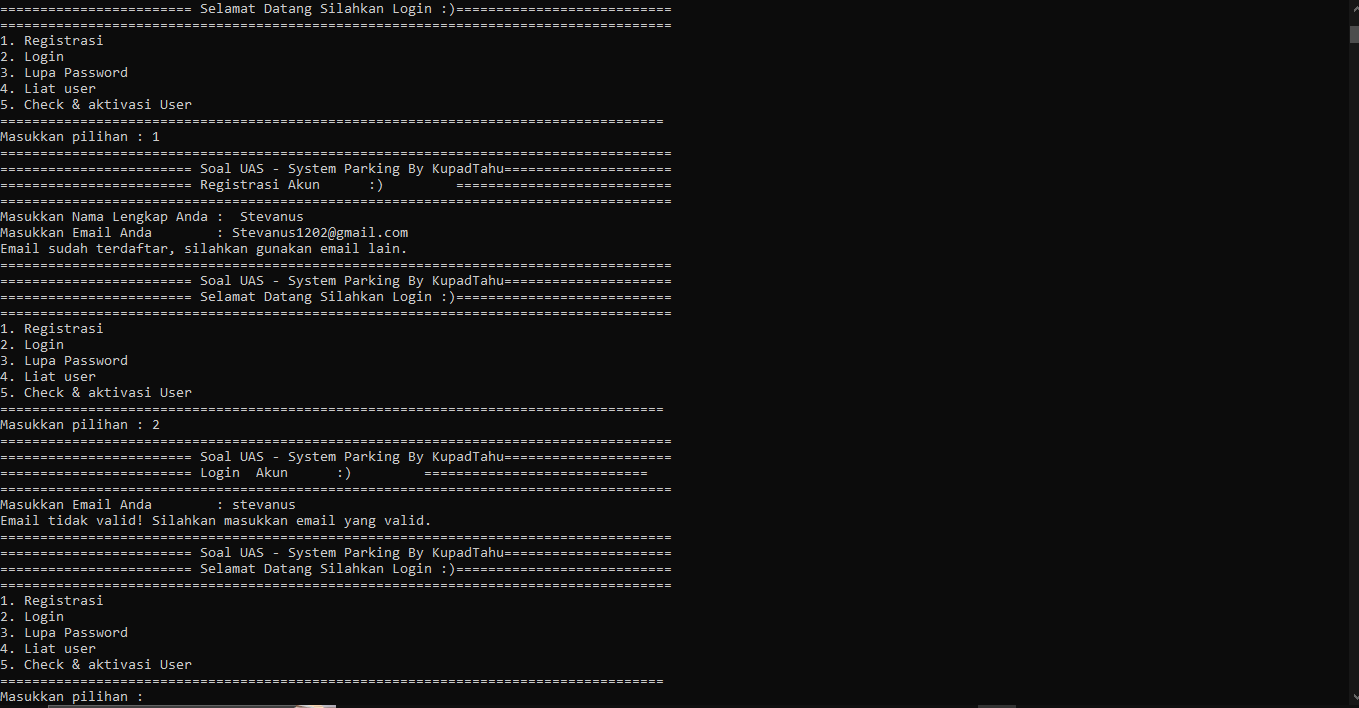


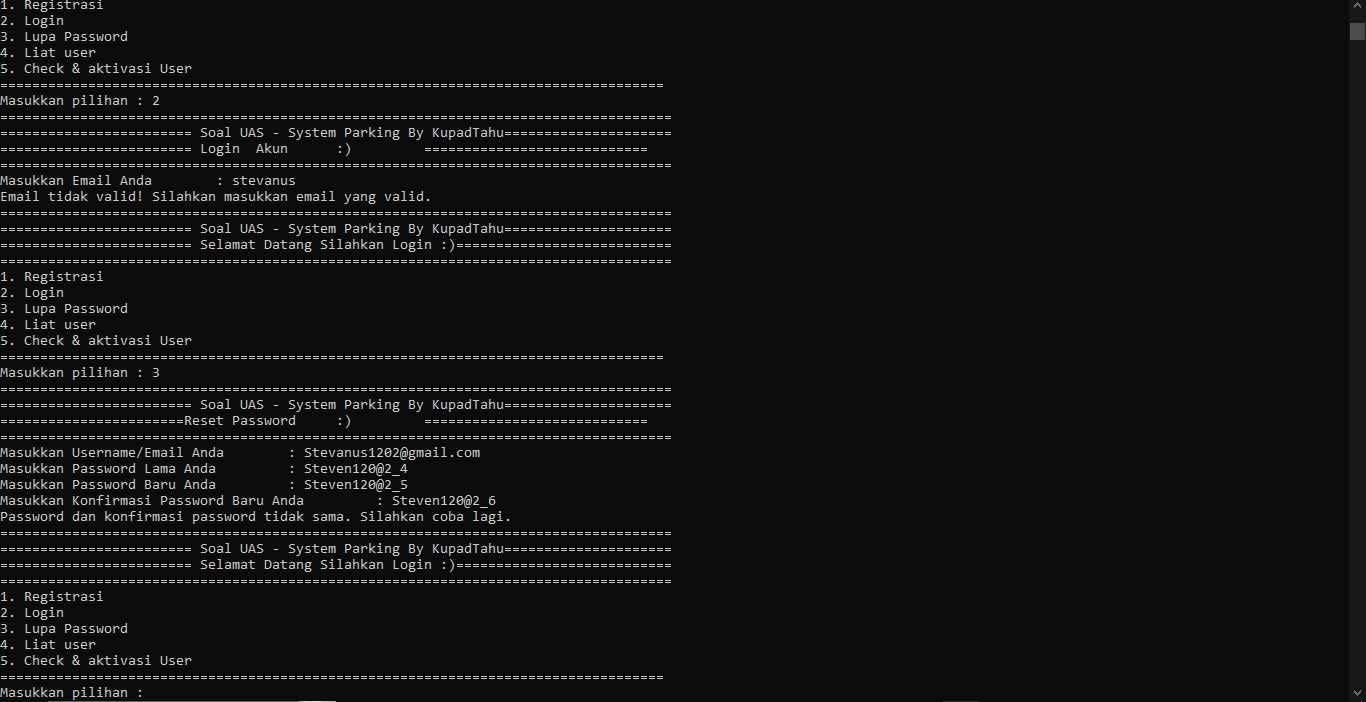


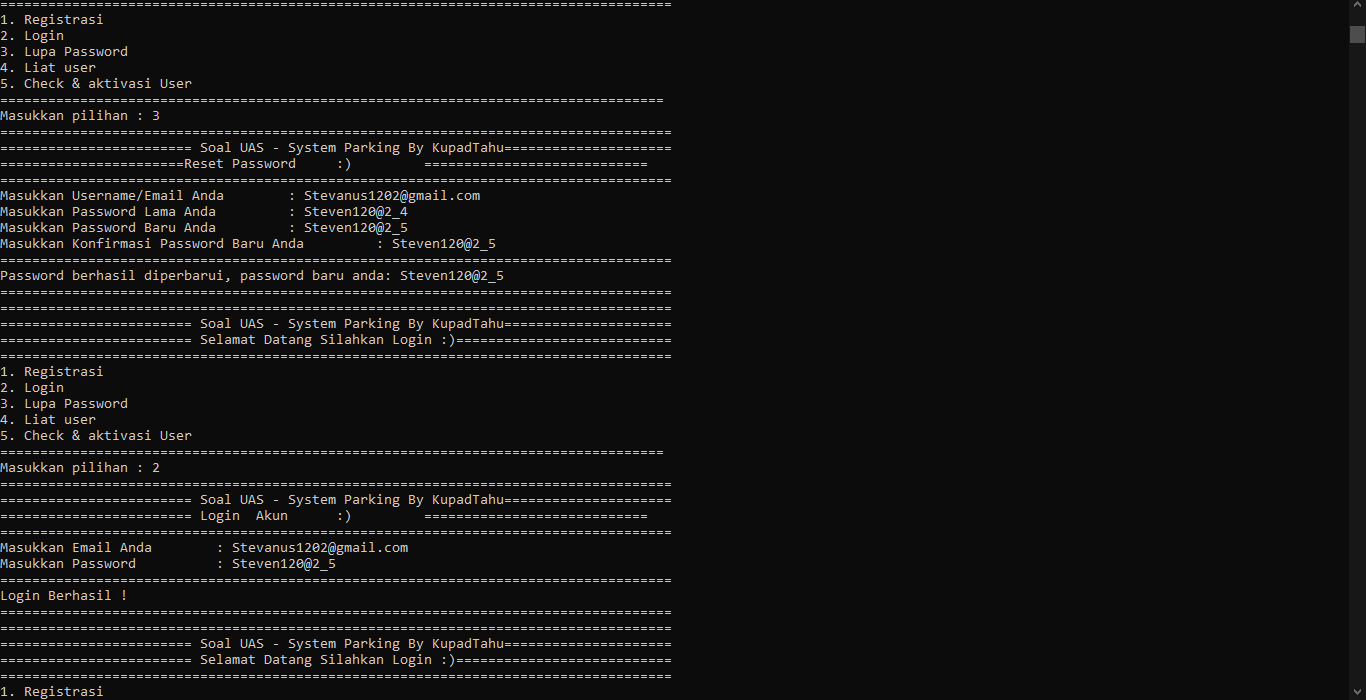


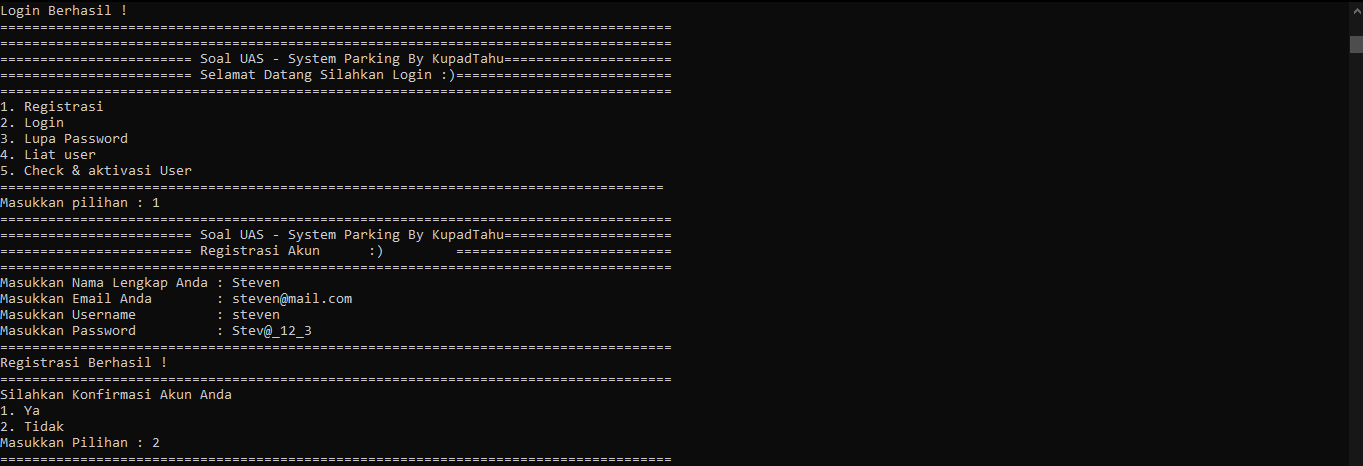














10. using System;

using System.Collections.Generic;

using System.ComponentModel;

class Program

{

static List<Member> members = new List<Member>();

static Dictionary<string, string> memberNameToId = new Dictionary<string, string>();

static List<ParkingData> parkingDataList = new List<ParkingData>();

static string loggedInUser;

public static string? Password\_user { get; private set; }

static void Main()

{

Console.WriteLine("===============================================================================");

Console.WriteLine("Nama : Stevanus Andika Galih Setiawan");

Console.WriteLine("Nim : 202301110008");

Console.WriteLine("Prodi : Rekayasa Perangkat Lunak");

Console.WriteLine("Mata Kuliah : Dasar Pemrograman");

Console.WriteLine("Semester : 1");

Console.WriteLine("Tanggal : 20 Januari 2024");

Console.WriteLine("====================================================================================");

Console.WriteLine("======================== Soal UAS - System Parking By KupadTahu=====================");

Console.WriteLine("======================== Login Akun :) ============================");

Console.WriteLine("====================================================================================");

Login();

int menu;

do

{

Console.WriteLine("====Menu==========");

Console.WriteLine("\n1. Tambah Member");

Console.WriteLine("2. Tampil Member");

Console.WriteLine("3. Update Member");

Console.WriteLine("4. Hapus Member");

Console.WriteLine("5. System Parking");

Console.WriteLine("6. Report Parking");

Console.WriteLine("7. Keluar");

Console.Write("Masukkan Pilihan: ");

if (!int.TryParse(Console.ReadLine(), out menu))

{

Console.WriteLine("Invalid input. Please enter a valid menu choice.");

continue;

}

switch (menu)

{

case 1:

TambahMember();

break;

case 2:

TampilMember();

break;

case 3:

UpdateMember();

break;

case 4:

HapusMember();

break;

case 5:

SystemParking();

break;

case 6:

ReportParking();

break;

case 7:

Console.WriteLine("Terima kasih telah menggunakan layanan System Parking By KupadTahui. Jaga selalu kesehatan!");

break;

default:

Console.WriteLine("Pilihan tidak valid. Silakan pilih kembali.");

break;

}

} while (menu != 7);

}

static void Login()

{

Console.Write("Masukkan Nama Email Anda: ");

loggedInUser = Console.ReadLine();

Console.WriteLine("Masukkan Password Anda:");

Password\_user = Console.ReadLine();

Console.WriteLine("===============================================================================");

Console.WriteLine($"Soal UAS - Selamat Datang System Parking By KupadTahu==={loggedInUser}!");

Console.WriteLine("===============================================================================\n");

}

static void TambahMember()

{

Console.WriteLine("====================================================================================");

Console.WriteLine("\n1. Tambah Member :");

Console.Write("Id Member (otomatis): ");

Console.WriteLine("\n");

string idMember = GenerateIdMember();

Console.Write("Nama Member: ");

string namaMember = Console.ReadLine();

Console.WriteLine("====================================================================================");

if (IsNamaMemberExist(namaMember))

{

Console.WriteLine("Nama Member sudah terdaftar. Tambah Member Gagal.");

return;

}

Console.Write("No Handphone: ");

string noHandphone = Console.ReadLine();

Console.Write("No Kendaraan: ");

string noKendaraan = Console.ReadLine();

Console.Write("Jenis Kendaraan (Mobil/Motor): ");

string jenisKendaraan = Console.ReadLine();

if (IsNoKendaraanExist(noKendaraan))

{

Console.WriteLine("Nomor Kendaraan sudah terdaftar. Tambah Member Gagal.");

return;

}

if (!IsJenisKendaraanValid(idMember, jenisKendaraan))

{

Console.WriteLine("Jenis Kendaraan tidak valid. Tambah Member Gagal.");

return;

}

DateTime periodeAwal = DateTime.Now;

DateTime periodeBerakhir = periodeAwal.AddDays(1);

bool statusMember = false; // Default Tidak Aktif

Member newMember = new Member

{

Id = idMember,

Nama = namaMember,

NoHandphone = noHandphone,

NoKendaraan = noKendaraan,

JenisKendaraan = jenisKendaraan,

PeriodeAwal = periodeAwal,

PeriodeBerakhir = periodeBerakhir,

Status = statusMember

};

members.Add(newMember);

memberNameToId[namaMember] = idMember;

Console.WriteLine("Tambah Member Berhasil.");

}

private static string GenerateIdMember()

{

Random rand = new Random();

return $"{(char)rand.Next('A', 'Z' + 1)}{(char)rand.Next('A', 'Z' + 1)}{(char)rand.Next('A', 'Z' + 1)}" +

$"{rand.Next(10, 99)}";

}

static bool IsNamaMemberExist(string namaMember)

{

return members.Exists(member => member.Nama == namaMember);

}

static bool IsNoKendaraanExist(string noKendaraan)

{

return members.Exists(member => member.NoKendaraan == noKendaraan);

}

static bool IsJenisKendaraanValid(string idMember, string jenisKendaraan)

{

if (jenisKendaraan.Equals("mobil", StringComparison.OrdinalIgnoreCase))

{

return members.Count(member => member.Id == idMember && member.JenisKendaraan.Equals("mobil", StringComparison.OrdinalIgnoreCase)) < 1;

}

else if (jenisKendaraan.Equals("motor", StringComparison.OrdinalIgnoreCase))

{

return members.Count(member => member.Id == idMember && member.JenisKendaraan.Equals("motor", StringComparison.OrdinalIgnoreCase)) < 1;

}

return false;

}

static void TampilMember()

{

Console.WriteLine("\n2. Tampil Member :");

Console.WriteLine("====================================");

foreach (var member in members)

{

Console.WriteLine("====================================");

Console.WriteLine($"Id Member: {member.Id}");

Console.WriteLine($"Nama Member: {member.Nama}");

Console.WriteLine($"No Handphone: {member.NoHandphone}");

Console.WriteLine($"No Kendaraan: {member.NoKendaraan}");

Console.WriteLine($"Jenis Kendaraan: {member.JenisKendaraan}");

Console.WriteLine($"Periode Awal Member: {member.PeriodeAwal.ToString("yyyy-MM-dd HH:mm:ss")}");

Console.WriteLine($"Periode Berakhir Member: {member.PeriodeBerakhir.ToString("yyyy-MM-dd HH:mm:ss")}");

Console.WriteLine($"Status Member: {(member.Status ? "Aktif" : "Tidak Aktif")}");

Console.WriteLine("====================================");

}

}

static void UpdateMember()

{

Console.WriteLine("\n3. Update Member :");

Console.WriteLine("====================================");

Console.Write("Masukkan Nama Member: ");

string namaMember = Console.ReadLine();

// Use the dictionary to get the corresponding member ID

if (memberNameToId.TryGetValue(namaMember, out string idMember))

{

Member selectedMember = members.Find(member => member.Id == idMember);

if (selectedMember == null)

{

Console.WriteLine("ID Member tidak ditemukan.");

return;

}

Console.Write("Masukkan Tanggal Periode Awal Member: ");

DateTime newPeriodeAwal;

Console.WriteLine("====================================");

if (!DateTime.TryParse(Console.ReadLine(), out newPeriodeAwal))

{

Console.WriteLine("Invalid input for date. Update Member Gagal.");

return;

}

selectedMember.PeriodeAwal = newPeriodeAwal;

selectedMember.PeriodeBerakhir = newPeriodeAwal.AddDays(1);

Console.Write("Status Member (Aktif/Tidak Aktif): ");

string statusInput = Console.ReadLine();

selectedMember.Status = (statusInput.Equals("aktif", StringComparison.OrdinalIgnoreCase));

Console.WriteLine("Update Member Berhasil.");

}

else

{

Console.WriteLine("Nama Member tidak ditemukan.");

}

}

static void HapusMember()

{

Console.WriteLine("\n5. Hapus Member :");

Console.Write("Masukkan Nama Member: ");

string namaMember = Console.ReadLine();

// Use the dictionary to get the corresponding member ID

if (memberNameToId.TryGetValue(namaMember, out string idMember))

{

Member memberToRemove = members.Find(member => member.Id == idMember);

if (memberToRemove == null)

{

Console.WriteLine("ID Member tidak ditemukan.");

return;

}

members.Remove(memberToRemove);

// Remove the member from the dictionary

memberNameToId.Remove(namaMember);

Console.WriteLine("Hapus Member Berhasil.");

}

else

{

Console.WriteLine("Nama Member tidak ditemukan.");

}

}

static void SystemParking()

{

Console.WriteLine("\n5. System Parking :");

Console.Write("Masukkan Nomor Kendaraan: ");

string nomorKendaraan = Console.ReadLine();

Console.Write("Pilih Jenis Kendaraan (Mobil/Motor): ");

string jenisKendaraan = Console.ReadLine();

Console.Write("Masukkan Tanggal & Jam Keluar Kendaraan: ");

DateTime tanggalJamKeluar;

if (!DateTime.TryParse(Console.ReadLine(), out tanggalJamKeluar))

{

Console.WriteLine("Invalid input for date. System Parking Gagal.");

return;

}

string nota = GenerateNota();

string namaPetugas = loggedInUser;

string jenisMember = (IsMember(nomorKendaraan) ? "Member" : "Non Member");

TimeSpan lamaParkir = tanggalJamKeluar - DateTime.UtcNow;

double tarifMobil = 6000;

double tarifMotor = 3000;

double totalTarifParking = (jenisKendaraan.Equals("mobil", StringComparison.OrdinalIgnoreCase)) ? lamaParkir.TotalHours \* tarifMobil : lamaParkir.TotalHours \* tarifMotor;

double uangBayar;

Console.Write("Input Uang Bayar Parkir: ");

if (!double.TryParse(Console.ReadLine(), out uangBayar))

{

Console.WriteLine("Invalid input for amount. System Parking Gagal.");

return;

}

double kembalian = uangBayar - totalTarifParking;

Console.WriteLine("====================================");

Console.WriteLine($"Nota: {nota}");

Console.WriteLine($"Nama Petugas: {namaPetugas}");

Console.WriteLine($"Nomor Kendaraan: {nomorKendaraan}");

Console.WriteLine($"Jenis Kendaraan: {jenisKendaraan}");

Console.WriteLine($"Jenis Member: {jenisMember}");

Console.WriteLine($"Lama Parking: {lamaParkir.Hours} jam {lamaParkir.Minutes} menit");

Console.WriteLine($"Tarif Mobil: Rp. {tarifMobil}");

Console.WriteLine($"Tarif Motor: Rp. {tarifMotor}");

Console.WriteLine($"Total Tarif Parking: Rp. {totalTarifParking}");

Console.WriteLine($"Kembalian Parkir: Rp. {kembalian}");

Console.WriteLine("====================================");

// Tambahkan data parking ke list

ParkingData parkingData = new ParkingData

{

Nota = nota,

NamaPetugas = namaPetugas,

NomorKendaraan = nomorKendaraan,

JenisKendaraan = jenisKendaraan,

TanggalMasuk = DateTime.UtcNow,

TanggalKeluar = tanggalJamKeluar,

JenisMember = jenisMember,

TotalTarifMobil = (jenisKendaraan.Equals("mobil", StringComparison.OrdinalIgnoreCase)) ? totalTarifParking : 0

};

parkingDataList.Add(parkingData);

}

static void ReportParking()

{

Console.WriteLine("\n6. Report Parking :");

foreach (var parkingData in parkingDataList)

{

Console.WriteLine("====================================");

Console.WriteLine($"Nota: {parkingData.Nota}");

Console.WriteLine($"Nama Petugas: {parkingData.NamaPetugas}");

Console.WriteLine($"Nomor Kendaraan: {parkingData.NomorKendaraan}");

Console.WriteLine($"Jenis Kendaraan: {parkingData.JenisKendaraan}");

Console.WriteLine($"Tanggal Masuk: {parkingData.TanggalMasuk.ToString("yyyy-MM-dd HH:mm:ss")}");

Console.WriteLine($"Tanggal Keluar: {parkingData.TanggalKeluar.ToString("yyyy-MM-dd HH:mm:ss")}");

Console.WriteLine($"Jenis Member: {parkingData.JenisMember}");

Console.WriteLine($"Total Tarif Mobil: Rp. {parkingData.TotalTarifMobil}");

Console.WriteLine("====================================");

}

}

static string GenerateNota()

{

Random rand = new Random();

return $"{(char)rand.Next('A', 'Z' + 1)}{(char)rand.Next('A', 'Z' + 1)}{(char)rand.Next('A', 'Z' + 1)}" +

$"{rand.Next(10, 99)}";

}

static bool IsMember(string nomorKendaraan)

{

return members.Exists(member => member.NoKendaraan == nomorKendaraan);

}

}

class Member

{

public string Id { get; set; }

public string Nama { get; set; }

public string NoHandphone { get; set; }

public string NoKendaraan { get; set; }

public string JenisKendaraan { get; set; }

public DateTime PeriodeAwal { get; set; }

public DateTime PeriodeBerakhir { get; set; }

public bool Status { get; set; }

}

class ParkingData

{

public string Nota { get; set; }

public string NamaPetugas { get; set; }

public string NomorKendaraan { get; set; }

public string JenisKendaraan { get; set; }

public DateTime TanggalMasuk { get; set; }

public DateTime TanggalKeluar { get; set; }

public string JenisMember { get; set; }

public double TotalTarifMobil { get; set; }

}

Output :

