

Makalah Final Project

Pemrograman Dasar - 01 Kelompok 10

https://youtu.be/AfDwh7fh_Ls

Pembagian Tugas :

1. Cecilia Inez Reva // 2106636994
 1. Membuat function Opening
 2. Membuat function Kendaraan
 3. Membuat function Helpdesk
 4. Mencari bug pada program
 5. Membuat User Interface agar menarik
 6. Membuat rangkaian kasar Flowchart
 7. Mencari kekurangan Pseudocode
 8. mengupload ke youtube video presentasi
 9. Mengedit makalah dari list function hingga sourcecode
 10. Memfasilitasi Tempat bertemu virtual

2. Luthfi Misbachul Munir // 2106631961
 1. Membuat function Menu
 2. Membuat function InputData
 3. Membuat function TempatParkir
 4. Melakukan revisi serta finalisasi source code
 5. Memperbaiki bug program
 6. Menambahi serta merevisi Flowchart
 7. Membuat rangkaian kasar Pseudocode
 8. Menghias User Interface Program
 9. Mengedit video presentasi
 10. Mengedit makalah dari pembukaan hingga Flowchart

3. Raditya Ihsan Dhiaulhaq // 2106733912
 1. Membuat function Rand
 2. Membuat function ListHarga
 3. Membuat function Struk
 4. Membuat tempat parkir yang tersedia menggunakan function Rand
 5. Pengecekan keefisienan dan keefektifan program
 6. Mencari kekurangan flowchart
 7. Menambahi serta merevisi Pseudocode
 8. Menghias User Interface program
 9. Merecord video presentasi
 10. Mengedit makalah dari pseudocode hingga list variable

Tema Program

Tingkat mobilitas kendaraan kota besar di Indonesia semakin tinggi. Hal ini menciptakan masalah baru, yaitu ketersediaan lahan parkir. Solusi dari permasalahan ini, kami kelompok 10 menciptakan sebuah aplikasi smart parking system dengan nama "Honest Parking". Melalui aplikasi ini, pengendara dapat memesan tempat parkir secara daring dan memeriksa ketersediaan lahan parkir sebelum tiba di lokasi. Aplikasi Honest Parking sebagai solusi dari permasalahan lahan parkir penuh untuk membantu pengendara di lingkungan sekitar Universitas Indonesia.

Permasalahan tersebut juga sering dialami oleh masyarakat di kota-kota besar yang tidak jarang para pengguna kendaraan harus memarkirkan kendaraannya di pinggir jalan dan mengganggu arus lalu lintas pada jalan tersebut. Hal ini dapat menimbulkan masalah lain seperti macet atau bahkan dapat meningkatkan tingkat pencurian kendaraan di daerah tersebut.

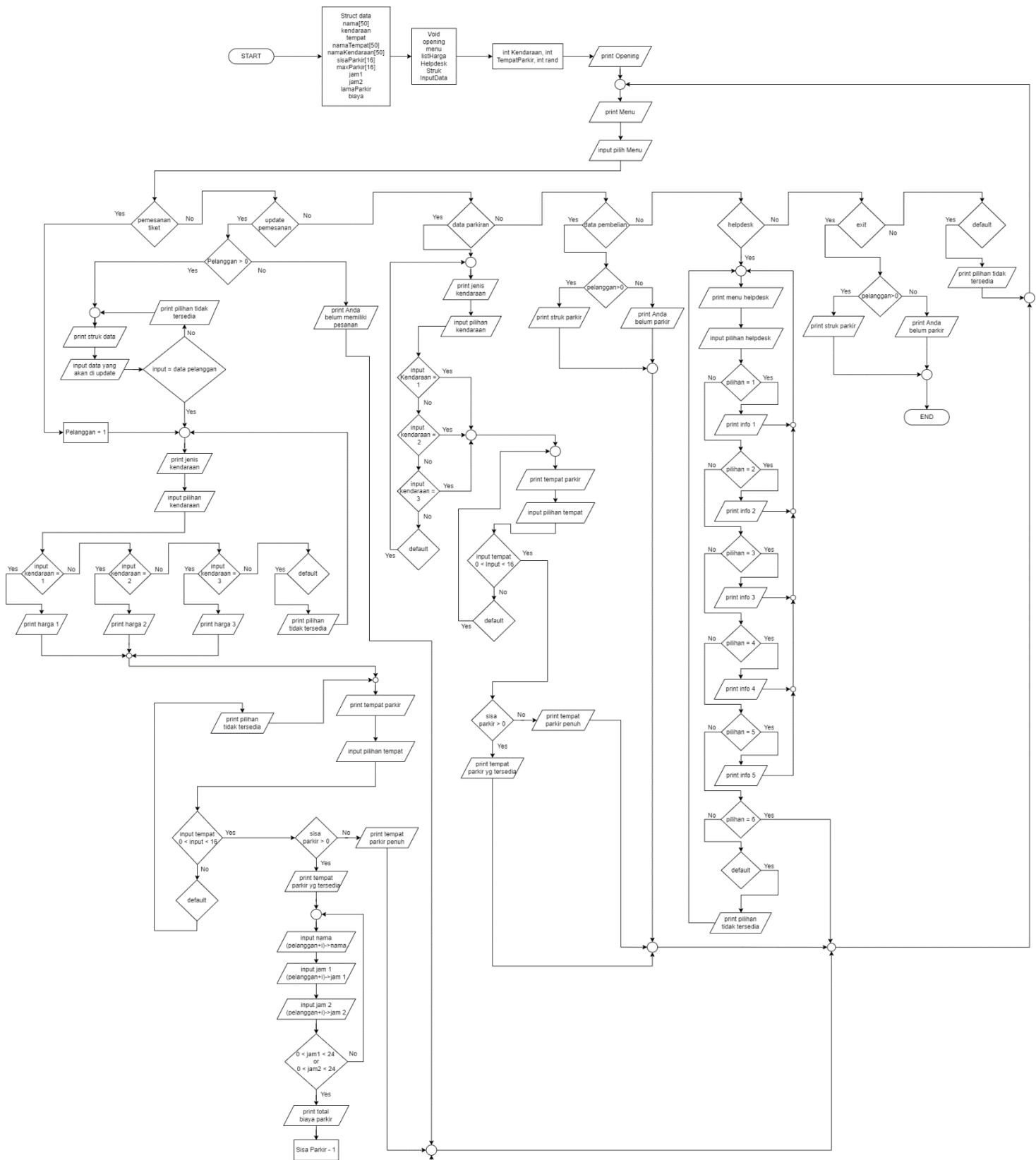
Solusi dari permasalahan ini, kami kelompok 10 menciptakan sebuah aplikasi smart parking system dengan nama "Honest Parking". Melalui aplikasi ini, pengendara dapat memesan tempat parkir secara daring dan memeriksa ketersediaan lahan parkir sebelum tiba di lokasi. Hal ini akan memudahkan para pengguna kendaraan yang akan mengunjungi area yang berada di lingkungan Universitas Indonesia, seperti Gedung rektorat, fakultas Teknik, fakultas kedokteran, masjid UI dan berbagai tempat yang tersedia lainnya. Pengguna kendaraan tidak perlu mencemaskan tempat parkir yang tersedia karena dapat di cek terlebih dahulu bahkan dapat dipesan terlebih dahulu sebelum pengguna datang ke tempat parkir tersebut.

Target pengguna dari aplikasi Honest Parking ini adalah para mahasiswa serta civitas Universitas Indonesia yang menggunakan kendaraan pribadi mereka. Kendaraan yang dapat digunakan pada aplikasi ini antara lain mobil, motor bahkan sepeda. Terdapat hal yang menarik dan berbeda dari aplikasi lainnya, di aplikasi ini pengguna aplikasi dapat melakukan pembayaran manual secara pribadi sesuai dengan input user. Dari sini diharapkan civitas Universitas Indonesia dapat meningkatkan nilai kejujuran pada lingkungan masyarakat, karena user sendiri lah yang menginput letak parkir, mulai jam parkir hingga selesai jam parkir. Sistem pembayarannya dilakukan secara online sehingga dapat meminimalisir terjadinya kontak langsung di kondisi pandemic ini. Dengan demikian, diharapkan pula civitas Universitas Indonesia dapat mengurangi rate kenaikan kasus Covid-19 dan siap untuk melakukan aktivitas offline di kampus.



Terdapat beberapa pilihan menu pada program seperti pemesanan tiket, mengupdate data pemesanan, melihat data sisa parkir, melihat data pemesanan hingga terdapat help desk untuk memudahkan user saat bingung dalam penggunaan aplikasi kami. Dengan dibuatnya aplikasi 'Honest Parking,' kami harap masalah parkir di UI dapat teratasi.

Flowchart :



Pseudocode :

Algoritma Honesty Parking

```
int pilih, i=0, j=0, loop
typedef struct {
    char nama[50]
    char namaTempat[50]
    char namaKendaraan[50]
    int kendaraan
    int tempat
    int sisaParkir[16]
    int maxParkir[16]
    int jam1
    int jam2
    int lamaParkir
    int biaya
} Data;

do {
    print ("Menu yang tersedia :
        1.  Pemesanan tiket
        2.  Update pemesanan
        3.  Data parkir
        4.  Data pembelian
        5.  Help desk
        6.  Exit app")
    get pilih
    switch (pilih)
        case 1 :
            call Function (Kendaraan,ListHarga,TempatParkir)
            if ((pelanggan)->sisaParkir[(pelanggan+i)->tempat]<1)
                break
            endif
            call Function (Inputdata)
            inc i
            break
        case 2 :
            do {
                call Function (Struk)
                if (j<0 || j>=i)
```

```

        print "Pilihan yang anda pilih tidak tersedia"
    endif
    else if (i<1)
        break
    } while (j<0 || j>=i)
    if (i<1)
        break
    call Function (Kendaraan,ListHarga,TempatParkir)
    if ((pelanggan)->sisaParkir[(pelanggan+j)->tempat])<1
        break
    call Function (InputData)
    break
case 3 :
    call Function (Kendaraan,TempatParkir)
    inc (pelanggan)->sisaParkir[(pelanggan+i)->tempat]
case 4 :
    call Function (Struk)
    break
case 5 :
    call Function (Helpdesk)
case 6 :
    exit
default :
    print (Pilihan anda tidak tersedia)
    break
} while (pilih != 6)

```

Void Kendaraan

```

do {
    print (Tempat parkir hanya tersedia untuk moda transportasi : 1.
        Mobil 2. Motor 3. Sepeda)
    print (Moda transportasi apa yang anda gunakan : )
    get (pelanggan+i)->kendaraan
    if (((pelanggan+i)->kendaraan)<1 || ((pelanggan+i)->kendaraan)>3)
        print (Input yang anda berikan tidak sesuai)
    endif
} while (((pelanggan+i)->kendaraan)<1 || ((pelanggan+i)->kendaraan)>3)

```

Void ListHarga

```

switch ((pelanggan+i)->kendaraan)

```

```

case 1 :
    print (List Biaya Parkir Mobil)
    break
case 2 :
    print (List Biaya Parkir Motor)
    break
case 3 :
    print (List Biaya Parkir Sepeda)
    break

```

Void TempatParkir

```

char nama_tempat[16][50] = { " ", "Gedung Rektorat", "Fakultas Teknik", "Pusat
                                Kegiatan Mahasiswa", "Stasiun UI", "Masjid UI" }

```

```

do {
    print ("Tempat parkir yang tersedia di UI :
        1. Gedung rektorat
        2. Fakultas teknik
        3. Pusat kegiatan mahasiswa
        4. Stasiun UI
        5. Masjid UI
        6. Fakultas Kesehatan Masyarakat
        7. Fakultas Hukum
        8. Fakultas Ilmu Administrasi
        9. Fakultas Ilmu Komputer
        10. Fakultas Ilmu Pengetahuan Budaya
        11. Fakultas Ekonomi dan Bisnis
        12. Fakultas Farmasi
        13. Fakultas Ilmu Keperawatan
        14. Fakultas Kedokteran
        15. Fakultas Ilmu Sosial dan Politik
    print (Tempat Parkir Tujuan anda)
    get (pelanggan+i)->tempat)
    if ((pelanggan+i)->tempat<1 || (pelanggan+i)->tempat>5)
        print(Input tidak sesuai) endif
    else
        decrement (pelanggan)->sisaParkir[(pelanggan+i)->tempat]
        (pelanggan+i)->namaTempat =
        nama_tempat[(pelanggan+i)->tempat]
    endif

```

```

} while ((pelanggan+i)->tempat<1 || (pelanggan+i)->tempat>5)
    print Tempat Parkir yang tersedia di "Tempat parkir yang dipilih"
    if ((pelanggan)->sisaParkir[(pelanggan+i)->tempat]<1)
        print (Telah Penuh Anda dapat mencari tempat parkir di tempat lain)
    else
        print(Tersisa sekian dari total sebanyak sekian)

```

Void InputData

```

int hargaAwal, harga
do {
    print (Nama Pemesan , Parkir dari Jam berapa sampai Jam berapa)
    get (pelanggan+i)->nama), (pelanggan+i)->jam1),
        (pelanggan+i)->jam2
    if ((pelanggan+i)->jam1 > 24 || (pelanggan+i)->jam1 < 1 ||
        (pelanggan+i)->jam2 > 24 || (pelanggan+i)->jam2 < 1 )
        print ("Harap masukan jam antara pukul 1 hingga 24")
} while ((pelanggan+i)->jam1 > 24 || (pelanggan+i)->jam1 < 1 ||
        (pelanggan+i)->jam2 > 24 || (pelanggan+i)->jam2 < 1 )

switch ((pelanggan+i)->kendaraan)
    case 1 :
        harga = 5000
        hargaAwal = 8000
        break
    case 2 :
        harga = 3000
        hargaAwal = 5000
        break
    case 3 :
        harga = 1000
        hargaAwal = 2000
        break
if ((pelanggan+i)->jam1 > (pelanggan+i)->jam2)
    (pelanggan+i)->lamaParkir = ((24 - (pelanggan+i)->jam1) +
    (pelanggan+i)->jam2)
    (pelanggan+i)->biaya = (((pelanggan+i)->lamaParkir - 1)* harga) +
    hargaAwal
endif
else if ((pelanggan+i)->jam1 < (pelanggan+i)->jam2 )
    (pelanggan+i)->lamaParkir = ((pelanggan+i)->jam2 -

```

```

        (pelanggan+i)->jam1)
        (pelanggan+i)->biaya = (((pelanggan+i)->lamaParkir - 1)* harga) +
        hargaAwal
    endif
else
    (pelanggan+i)->lamaParkir = 24
    (pelanggan+i)->biaya = (((pelanggan+i)->lamaParkir - 1)* harga) +
    hargaAwal
    endif
    print (Harga Total menjadi "(pelanggan+i)->biaya")

```

Void Struk

```

    int loop;
    for (loop=0; loop<i; loop++)
        print (Nama Pemesan, Tempat Parkir, Jam Masuk Parkir, Jam Keluar
        Parkir, Durasi ,Total Biaya Parkir)

```

Void Helpdesk

```

    int bantuan = 0
    get bantuan
    if bantuan == 1
        print ("Tiket Anda bisa dipesan pada bagian Menu dengan
        memasukkan pilihan 1. Selanjutnya, Anda dapat mengisi
        data pembelian tiket.")
    endif
    else if bantuan == 2
        print ("Anda dapat mengganti tiket pada bagian Menu dengan
        memasukkan pilihan 2. Selanjutnya, Anda dapat
        mengupdate/mengubah data tiket.")
    endif
    else if bantuan == 3
        print ("Masukkan pilihan 3 pada Menu untuk melihat data
        parkir yang tersedia.")
    endif
    else if bantuan == 4
        print ("Masukkan pilihan 4 pada Menu untuk melihat data
        pembelian tiket parkir yang sudah Anda pesan
        sebelumnya.")
    endif
    else if bantuan == 5

```



```
        print ("Masukkan pilihan 6 untuk exit dari Menu.")
    endif
else
    print ("Pilihan tidak tersedia. Harap masukkan kembali.")
endif
```

List Variabel/Array/Struct/Pointer

- Struct bernama **Data**, terdiri dari variabel :
 - **nama** : menyimpan nama user
 - **Array namaTempat** : mengubah input int user menjadi char
 - **Array namaKendaraan** : mengubah input int user menjadi char
 - **kendaraan** : menentukan biaya parkir user sesuai kendaraan yang digunakan
 - **tempat** : menentukan tempat parkir yang diinginkan user
 - **Array sisaParkir** : menentukan sisa parkir yang tersedia di tempat yang telah ditentukan
 - **Array maxParkir** : menentukan angka maksimal parkir yang terdapat di tempat yang telah ditentukan
 - **jam1** : memberikan data kapan user memulai parkir
 - **jam2** : memberikan data kapan user mengakhiri parkir
 - **lamaParkir** : menghitung durasi parkir dengan mengurangi variabel jam2 dengan jam 1
 - **biaya** : menghitung biaya parkir user
- Variabel dalam program :
 - **loop** : me-looping program
 - **i** : memberikan data orang ke-i
 - **j** : memilih data ke-j yang ingin di update oleh user
 - **pilih** : memilih submenu program
 - **hargaAwal** : biaya parkir 1 jam pertama sesuai kendaraan user
 - **harga** : biaya parkir setelah 1 jam berikutnya
 - **bantuan** : input user memilih pilihan dalam Helpdesk
- Pointer :
 - **pelanggan** : menyimpan seluruh data pelanggan terhadap struct
- Array:
 - **nama_tempat** : menyimpan data nama tempat parkir yang tersedia

List Fungsi dan Kegunaannya

1. **Opening ()** : menampilkan data anggota kelompok 10
2. **Menu ()** : menampilkan isi dari semua program Honest Parking serta function untuk memilih menu
3. **Kendaraan ()** : memilih kendaraan yang digunakan
4. **ListHarga ()** : menampilkan harga tiket parkir sesuai kendaraan
5. **TempatParkir ()** : menampilkan tempat parkir di Universitas Indonesia dan memeriksa ketersediaan lahan parkir di tempat tersebut

6. **InputData ()** : menerima data dari user seperti nama dan waktu parkir, melakukan perhitungan dan menampilkan informasi biaya parkir sesuai kendaraan dan lama durasi parkir
7. **Helpdesk ()** : membantu user dengan menampilkan informasi dalam mengoperasikan aplikasi
8. **Struk ()** : memberikan struk pembayaran berisi nama, tempat parkir, jam masuk parkir, jam keluar parkir, lama durasi parkir dan total biaya parkir.
9. **Rand ()** : untuk menggunakan deklarasi angka random yang diterapkan programmer untuk variabel sisaParkir dan maxParkir

Source Code

```
#include <stdio.h> //lib untuk input output
#include <stdlib.h> //lib untuk menyimpan data di pointer
#include <string.h> //lib untuk menyimpan string with space dan penggunaan
strcpy

//deklarasi struct dengan nama berupa Data
typedef struct{
    char nama[50];
    char namaTempat[50];
    char namaKendaraan[50];
    int kendaraan;
    int tempat;
    int sisaParkir[16];
    int maxParkir[16];
    int jam1;
    int jam2;
    int lamaParkir;
    int biaya;
}Data;

//prototype func untuk menampilkan opening program
void Opening ();

//prototype func untuk memilih menu
void Menu();

//prototype func untuk memilih kendaraan yang digunakan
int Kendaraan(Data *pelanggan, int i);

//prototype func untuk menampilkan harga parkir sesuai kendaraan yang
digunakan
void ListHarga(Data *pelanggan, int i);

//prototype func untuk memilih tempat parkir dan mengecek ketersediaan tempat
int TempatParkir(Data *pelanggan, int i);

//prototype func untuk memberikan bantuan kepada user jika mengalami kendala
void Helpdesk();

//prototype func untuk menginput data user seperti nama hingga jam parkir tsb
void InputData(Data *pelanggan, int i);

//prototype func untuk memberikan struk pembayaran serta data user yang
diinput
void Struk(Data *pelanggan, int i);
```

```
int main(Data *pelanggan, int i) {
    opening(); //memanggil function opening
    Menu(); //memanggil funtion menu
    if (i>0){
        system("pause"); system("CLS");
    }
    printf("Terima kasih telah menggunakan Application Honesty\n\n");
    return 0;
}

//func untuk menampilkan opening program
void opening () {
    int loop;

    printf ("\t\t\t\t\t=====\\n");
    sleep(500);
    printf ("\t\t\t\t\t||\t Proyek Akhir Pemrograman Dasar ||\n");
    sleep(500);
    printf ("\t\t\t\t\t|_|\\n");
    sleep(500);
    printf ("\t\t\t\t\tThis app made by : |\\n");
    sleep(500);
    printf ("\t\t\t\t\t1. Cecilia Inez Reva // 2106636994 ||\\n");
    sleep(500);
    printf ("\t\t\t\t\t2. Luthfi Misbachul Munir // 2106631961||\\n");
    sleep(500);
    printf ("\t\t\t\t\t3. Raditya Ihsan Dhiaulhaq // 2106733912||\\n");
    sleep(500);
    printf ("\t\t\t\t\t|_|\\n");
    sleep(500);
    printf ("\t\t\t\t\t=====\\n\\n");
    sleep(500);
    printf("\tPlease wait"); sleep(1000); //Sleep digunakan untuk memberikan delay
    for(loop=0; loop<3; loop++){
        printf("\a."); system("color 74"); sleep(500);
        system("color 07"); sleep(500); //sleep(500) berarti terdapat delay selama 500ms atau 0.5s
    }
    printf("\\n\\n"); system("pause"); system("CLS"); //digunakan untuk membersihkan tampilan saat melanjutkan program
}

//func untuk memilih menu
void Menu(){
    int pilih, i=0, j=0, loop;
    Data *pelanggan;

    //untuk mengalokasikan ruang sesuai dengan jumlah pemesanan
    pelanggan = (Data*) calloc (15, sizeof(Data));
    //untuk memberikan informasi jika memory untuk pointer tidak tercukupi
    if(pelanggan == NULL){
        printf("Error! Memory Not Allocated");
        exit(0);
    }

    //Untuk deklarasi sisa parkir dan max parkir pada tempat yang tersedia
    for(loop=1; loop<=15; loop++){
        (pelanggan)->sisaParkir[loop] = rand() % 25;
        //menggunakan func random
        (pelanggan)->maxParkir[loop] = rand() % 100 + 25; //supaya nilai max parkir > sisa parkir
    }
}
```

```

do {
    printf("\t\t|=====
=====||\n");
    printf("\t\t|
Welcome to Honesty Parking
||\n");
    printf("\t\t|
    printf("\t\t|_____Menu
Transaksi_____||\n"); sleep(500);
    printf("\t\t|
    ||\n");

    for(loop=0; loop<6; loop++){
        printf("\t\t|");
        for(j=0; j<28; j++){
            printf(" ");
        }
        switch(loop){
            case 0: printf("1. Pemesanan tiket ");
            break;
            case 1: printf("2. Update pemesanan "); break;
            case 2: printf("3. Data parkir "); break;
            case 3: printf("4. Data pembelian "); break;
            case 4: printf("5. Help desk "); break;
            case 5: printf("6. Exit app "); break;
            default: printf("ERROR"); break;
        }
        for(j=0; j<28; j++){
            printf(" ");
        } printf("||\n");
    }

    printf("\t\t|=====
=====||\n");
    printf("\t\t|    silahkan pilih menu yang ingin anda gunakan : ");
    scanf("%d", &pilih);
    sleep(500); system("CLS");

    switch(pilih) {
        case 1 : //case saat user memilih pemesanan tiket
            kendaraan(pelanggan, i);
            ListHarga(pelanggan, i);
            TempatParkir(pelanggan, i);
            //jika sisa parkir 0 maka program langsung di break
            sebelum menginput data pelanggan
            if ((pelanggan)->sisaParkir[(pelanggan+i)-
>tempat]<1){
                break;
            }
            InputData(pelanggan, i);
            i++; //untuk deklarasi user selanjutnya
            break;

        case 2 : //case saat user memilih update data
            do{
                Struk(pelanggan, i);
                if(i>0){

```

```

                                printf("Silahkan pilih pesanan yang ingin
anda update : ");
                                scanf("%d", &j);
                                j -= 1;
                                //menyesuaikan dengan variabel orang pemesan atau i
                                if(j<0 || j>=i){ //user hanya bisa
memilih update data sesuai nomer pesanan yang tersedia
                                system("color 74"); //mengganti
warna background dan font di terminal
                                printf("\aPilihan yang anda pilih
tidak tersedia!\n");
                                sleep(2000); system("CLS");
                                system("color 07");
                                }
                                } else if(i<1){ //ketika user belum memesan
tiket sama sekali maka tidak dapat mengupdate data
                                break;
                                }
                                }while(j<0 || j>=i); //user hanya bisa memilih
update data sesuai nomer pesanan yang tersedia
                                //jika user belum memesan parkir sama sekali, program
akan di break
                                if(i<1){
                                    break;
                                }
                                system("CLS");
                                Kendaraan(pelanggan, j);
                                ListHarga(pelanggan, j);
                                TempatParkir(pelanggan, j);
                                //jika sisa parkir di suatu tempat habis maka program
akan di break
                                if((pelanggan)->sisaParkir[(pelanggan+j)->tempat]<1){
                                    break;
                                }
                                InputData(pelanggan, j);
                                break;

                                case 3 : //case jika user ingin melihat sisa parkir yang
tersedia
                                Kendaraan(pelanggan, i);
                                system("pause"); system("CLS");
                                TempatParkir(pelanggan, i);
                                //menambahkan sisa parkir karena di func
TempatParkir, sisa parkir berkurang 1
                                (pelanggan)->sisaParkir[(pelanggan+i)->tempat]++;
                                system("pause"); system("CLS");
                                break;

                                case 4 : //case jika user ingin melihat data pesanan yang
telah dipesan
                                Struk(pelanggan, i);
                                if(i>0){
                                    system("pause"); system("CLS");
                                }
                                break;

                                case 5 : //case jika user membutuhkan bantuan untuk
penggunaan program ini
                                Helpdesk();
                                break;

                                case 6 : //case jika user ingin exit dari program
                                break;

```

```

        default : //error handling ketika user memberikan input
yang tidak tersedia
        system("color 74");
        printf("\aPilihan yang anda pilih tidak tersedia!\n");
        Sleep(2000); system("CLS");
        system("color 07");
        break;
    }
} while(pilih != 6); //program akan selalu ter-loop kecuali user
ingin mengakhiri program dengan menginput pilihan 6

    printf("Berikut data pesanan yang anda pesan");
    for(loop=0; loop<5; loop++){
        printf("."); Sleep(1000);
    } printf("\n\n");
    //memberikan struk pesanan
    Struk(pelanggan, i);
    //membebaskan memory pada pointer pelanggan
    free(pelanggan);
}

//func untuk memilih kendaraan yang digunakan
int kendaraan(Data *pelanggan, int i){
    char nama_kendaraan[4][50] = {"", "Mobil", "Motor", "Sepeda"};
    do{
        printf("Tempat parkir hanya tersedia untuk moda transportasi
:\n1. Mobil\n2. Motor\n3. Sepeda\n\n");
        printf("Moda transportasi apa yang anda gunakan : ");
        scanf("%d", &(pelanggan+i)->kendaraan);

        //error handling
        if (((pelanggan+i)->kendaraan)<1 || ((pelanggan+i)->kendaraan)>3)
        {
            system("color 74");
            printf("\n\nInput yang anda berikan tidak sesuai\nHarap
masukan angka yang sesuai\n");
            Sleep(2000); system("CLS"); system("color 07");
        };
    } while(((pelanggan+i)->kendaraan)<1 || ((pelanggan+i)->kendaraan)>3);
    strcpy((pelanggan+i)->namaKendaraan, nama_kendaraan[(pelanggan+i)-
>kendaraan]);
    //me-return input user mengenai kendaraan
    return (pelanggan+i)->kendaraan;
}

//func untuk menampilkan harga parkir sesuai kendaraan yang digunakan
void ListHarga(Data *pelanggan, int i){
    switch ((pelanggan+i)->kendaraan) {
        case 1 : //case saat user menggunakan mobil
            printf("\n=====\\n");
            printf("|| List biaya parkir kendaraan mobil : ||\\n");
            printf("|| 1 jam pertama    = Rp 8.000      ||\\n");
            printf("|| jam selanjutnya = Rp 5.000      ||\\n");
            printf("=====\\n");
            system("pause"); system("CLS"); break;

        case 2 : //case saat user menggunakan motor
            printf("\n=====\\n");
            printf("|| List biaya parkir kendaraan motor : ||\\n");
            printf("|| 1 jam pertama    = Rp 5.000      ||\\n");
            printf("|| jam selanjutnya = Rp 3.000      ||\\n");
            printf("=====\\n");
            system("pause"); system("CLS"); break;
    }
}

```

```

        case 3 : //case saat user menggunakan sepeda
            printf("\n=====\\n");
            printf("|| List biaya parkir kendaraan sepeda : ||\\n");
            printf("|| 1 jam pertama      = Rp 2.000      ||\\n");
            printf("|| jam selanjutnya = Rp 1.000      ||\\n");
            printf("=====\\n");
            system("pause"); system("CLS"); break;
        default :
            break;
    }
}

//func untuk memilih tempat parkir dan mengecek ketersediaan tempat
int TempatParkir(Data *pelanggan, int i){
    //deklarasi nama tempat yang tersedia
    char nama_tempat[16][50] = {" ", "Gedung Rektorat", "Fakultas
Teknik", "Pusat Kegiatan Mahasiswa", "Stasiun UI", "Masjid UI", "Fakultas
Kesehatan Masyarakat ", "Fakultas Hukum", "Fakultas Ilmu
Administrasi", "Fakultas Ilmu Komputer", "Fakultas Ilmu Pengetahuan
Budaya", "Fakultas Ekonomi dan Bisnis", "Fakultas Farmasi", "Fakultas Ilmu
Keperawatan", "Fakultas Kedokteran", "Fakultas Ilmu Sosial dan Politik"};

    do{

        printf("
=====\\n");
        printf(" ||                                     || HONESTY
PARKING                                     || \\n");
        printf("
=====|| \\n");
        printf(" ||                                     || Tempat parkir yang
tersedia di UI :                                     || \\n");
        printf("
=====|| \\n");
        printf(" || 1. Gedung Rektorat || 6.Fakultas Kesehatan
Masyarakat || 11.Fakultas Ekonomi dan Bisnis || \\n");
        printf(" || 2. Fakultas Teknik || 7.Fakultas Hukum
|| 12.Fakultas Farmasi || \\n");
        printf(" || 3. Pusat Kegiatan Mahasiswa || 8.Fakultas Ilmu
Administrasi || 13.Fakultas Ilmu Keperawatan || \\n");
        printf(" || 4. Stasiun UI || 9.Fakultas Ilmu
Komputer || 14.Fakultas Kedokteran || \\n");
        printf(" || 5. Masjid UI || 10.Fakultas Ilmu
Pengetahuan Budaya || 15.Fakultas Ilmu Sosial dan Politik || \\n");
        printf("
=====\\n");
        printf("\\n Tempat parkir tujuan anda : ");
        scanf("%d", &(pelanggan+i)->tempat);

        //error handling
        if ((pelanggan+i)->tempat<1 || (pelanggan+i)->tempat>15){
            system("color 74");
            printf("\\n\\aInput yang anda berikan tidak sesuai dengan
pilihan yang tersedia\\n");
            sleep(2000); system("CLS"); system("color 07");
        } else //mengurangkan kuota tempat parkir yang tersedia
            (pelanggan)->sisaParkir[(pelanggan+i)->tempat]--;
    } while (1);
}

```



```

//menggunakan strcpy untuk memberikan bentruk string terhadap
pilihan tempat user
strcpy((pelanggan+i)->namaTempat,nama_tempat[(pelanggan+i)-
>tempat]);
} while ((pelanggan+i)->tempat<1 || (pelanggan+i)->tempat>15);

printf("\n\n Tempat Parkir yang tersedia di %s",(pelanggan+i)-
>namaTempat);

//kondisi ketika tempat parkir sudah penuh
if ((pelanggan)->sisaParkir[(pelanggan+i)->tempat]<1){
    system("color 74");
    printf("\a\aTelah penuh\nAnda dapat mencari tempat parkir lain
disekitar UI!\n");
    system("pause"); system("CLS"); system("color 07");
} else
    printf("\n tersisa %d dari total sebanyak %d\n\n\n", (pelanggan)-
>sisaParkir[(pelanggan+i)->tempat], (pelanggan)->maxParkir[(pelanggan+i)-
>tempat]);
    system("pause"); system("CLS");
    //mereturn tempat parkir yang dipilih user ke-i
    return (pelanggan+i)->tempat;
}

//func untuk menginput data user seperti nama hingga jam parkir tsb
void InputData(Data *pelanggan, int i){
    int hargaAwal, harga;

    do{
        printf("
        printf("
        OF INDONESIA\n");
        printf("
        =====\n
        =====\n\n");
        //meminta data nama pemesan
        printf("        Nama        : ");
        scanf(" %49[^\n]s", &(pelanggan+i)->nama);
        //mengeprint kendaraan
        printf("        Kendaraan yang digunakan : ");
        switch ((pelanggan+i)->kendaraan) {
            case 1 : //case saat user menggunakan mobil
                printf("Mobil\n");
                break;

            case 2 : //case saat user menggunakan motor
                printf("Motor \n");
                break;

            case 3 : //case saat user menggunakan sepeda
                printf("Sepeda\n");
                break;

            default :
                printf("ERROR");
                break;
        }

        //meminta jam awal parkir
        printf("        Jam Masuk Parkir (1-24) : ");
        scanf("%d",&(pelanggan+i)->jam1);
        //meminta jam selesai parkir
        printf("        Jam Keluar Parkir (1-24) : ");
        scanf("%d",&(pelanggan+i)->jam2);
        //error handling
    }
}

```

```

        if((pelanggan+i)->jam1 > 24 || (pelanggan+i)->jam1 < 1 ||
        (pelanggan+i)->jam2 > 24 || (pelanggan+i)->jam2 < 1 ){
            system("color 74");
            printf("\aHarap masukan jam antara pukul 1 hingga 24\n\n");
            sleep(2000); system("color 07");
        }
    }while((pelanggan+i)->jam1 > 24 || (pelanggan+i)->jam1 < 1 ||
    (pelanggan+i)->jam2 > 24 || (pelanggan+i)->jam2 < 1 );

    //penyesuaian biaya parkir sesuai dengan kendaraan yang digunakan user
    switch((pelanggan+i)->kendaraan){
        case 1 : //ketika user menggunakan mobil
            harga = 5000;
            hargaAwal = 8000;
            break;

        case 2: //ketika user menggunakan motor
            harga = 3000;
            hargaAwal = 5000;
            break;

        case 3 : //ketika user menggunakan sepeda
            harga = 1000;
            hargaAwal = 2000;
            break;
    }

    //rumus perhitungan serta error handling
    if((pelanggan+i)->jam1 > (pelanggan+i)->jam2){
        (pelanggan+i)->lamaParkir = ((24 - (pelanggan+i)->jam1)+
        (pelanggan+i)->jam2);
        (pelanggan+i)->biaya = (((pelanggan+i)->lamaParkir - 1)*
        harga)+hargaAwal;
    }
    else if((pelanggan+i)->jam1 < (pelanggan+i)->jam2 ){
        (pelanggan+i)->lamaParkir = ((pelanggan+i)->jam2 - (pelanggan+i)-
        >jam1);
        (pelanggan+i)->biaya = (((pelanggan+i)->lamaParkir - 1)*
        harga)+hargaAwal;
    }
    else{
        (pelanggan+i)->lamaParkir = 24;
        (pelanggan+i)->biaya = (((pelanggan+i)->lamaParkir - 1)*
        harga)+hargaAwal;
    }

    //mengeprint lama parkir
    printf("        Durasi parkir                : %d jam\n", (pelanggan+i)-
    >lamaParkir);
    printf("        Harga Total menjadi            : Rp %d \n", (pelanggan+i)-
    >biaya);
    printf("        =====\n
    =====\n\n");
    system("pause"); system("CLS");
}

//func untuk menginput data user seperti nama hingga jam parkir tsb
void Helpdesk(){
    int bantuan;

    do {
        printf("==== Ada Masalah dan Memerlukan Bantuan? ====");
        printf("\n1. Bagaimana cara memesan tiket\n2. Bagaimana saya bisa
        mengganti pemesanan"

```

```

melihat data pembelian" "\n3. Bagaimana melihat data parkir\n4. Bagaimana
Menu\n\nSilahkan pilih menu : ");
scanf("%d", &bantuan);

switch(bantuan){
case 1 :
printf("\n===== Panduan Pemesanan Tiket
=====\\n");
printf("\n1. Pada halaman Menu, masukkan pilihan 1
untuk Pemesanan Tiket."
"\n2. Anda akan diarahkan ke halaman
selanjutnya. Anda dapat mengisi data pembelian tiket"
"\n seperti moda transportasi, lokasi
tempat parkir, nama pengguna, dan jam parkir."
"\n3. Di akhir pemesanan, tampilan harga
tiket parkir akan diperlihatkan."
"\n\\nNB : Struk parkir dapat dilihat pada
halaman Menu pilihan 4"
"\n Pilih 6 untuk kembali ke
Menu.\\n\\n\\n");
system("pause"); system("CLS");
break;

case 2 :
printf("\n===== Panduan Update Pemesanan
=====\\n");
printf("\n1. Pada halaman Menu, masukkan pilihan 2
untuk Update Pemesanan."
"\n2. Masukkan pilihan tiket yang Anda
ingin ubah."
"\n3. Anda dapat mengisi data pembelian
tiket kembali."
"\n4. Data pemesanan Anda sudah otomatis
diubah."
"\n\\nNB : Jika Anda belum memesan tiket,
data pembelian tidak akan terlihat."
"\n Pesan tiket terlebih dahulu pada
halaman Menu pilihan 1."
"\n Pilih 6 untuk kembali ke
Menu.\\n\\n\\n");
system("pause"); system("CLS");
break;

case 3 :
printf("\n===== Panduan Ketersediaan Tempat Parkir
=====\\n");
printf("\n1. Pada halaman Menu, masukkan pilihan 3
untuk melihat Data Tempat Parkir."
"\n2. Anda dapat mengisi pilihan
informasi tempat parkir yang diminta."
"\n3. Akhir pemesanan, tampilan
ketersediaan tempat parkir akan diperlihatkan."
"\n\\nNB : Tampilan akan menunjukkan
informasi mengenai ketersediaan tempat
parkir pada setiap lokasi tempat
masing-masing."
"\n Pilih 6 untuk kembali ke
Menu.\\n\\n\\n");
system("pause"); system("CLS");
break;

case 4 :

```

```

        printf("\n===== Panduan Melihat Struk Parkir
=====\\n");
        printf("\\n1. Pada halaman Menu, masukkan pilihan 4
untuk melihat Struk Parkir."
        "\\n2. Tampilan ketersediaan tempat parkir
akan diperlihatkan."
        "\\n\\nNB : Struk parkir akan ditampilkan
pada pilihan ini."
        "\\n      Pesan tiket terlebih dahulu pada
halaman Menu pilihan 1."
        "\\n      Pilih 6 untuk kembali ke
Menu.\\n\\n\\n");
        system("pause"); system("CLS");
        break;

    case 5 :
        printf("\\n===== Panduan Exit dari App
=====\\n");
        printf("\\n1. Pada halaman Menu, masukkan pilihan 6
untuk Exit dari App."
        "\\n2. Tampilan struk parkir akan
diperlihatkan."
        "\\n\\nNB : Pilih 6 untuk kembali ke
Menu.\\n\\n\\n");
        system("pause"); system("CLS");
        break;

    case 6 :
        break;

    default :
        system("color 74");
        printf("\\n\\n\\nPilihan menu tidak tersedia.\\n\\n\\n");
        Sleep(2000); system("CLS");
        system("color 07");
        break;
    }
} while(bantuan!=6);

system("pause"); system("CLS");
}

//func untuk memberikan struk pembayaran serta data user yang diinput
void Struk(Data *pelanggan, int i){
    int loop;

    //mengeprint struk sesuai pesanan yang tersedia
    for (loop=0; loop<i; loop++){
        printf("_____(%d)\\n",
loop+1);
        printf("
                                HONESTY PARKING\\n
                                UNIVERSITY
OF INDONESIA\\n");
        printf("=====\\n
=====\\n\\n");
        printf("      Nama : %s \\n", (pelanggan+loop)-
>nama);
        printf("      Kendaraan yang Digunakan : %s \\n", (pelanggan+loop)-
>namaKendaraan);
        printf("      Tempat Parkir : %s \\n", (pelanggan+loop)-
>namaTempat);
        printf("      Jam Masuk Parkir : %d.00 \\n",
(pelanggan+loop)->jam1);
        printf("      Jam Keluar Parkir : %d.00 \\n",
(pelanggan+loop)->jam2);

```

```

        printf("        Lama Durasi Parkir          : %d jam\n",
(pelanggan+loop)->lamaParkir);
        printf("        Total Biaya Parkir           : Rp %d \n\n",
(pelanggan+loop)->biaya);
        printf("        =====\n
===== \n\n");
    }

    //kondisi ketika user belum memesan tempat parkir sama sekali
    if (i<1){
        printf("Anda belum parkir ditempat mana pun\nSilahkan pilih
tempat parkir dahulu untuk mendapat data parkir\n\n");
        system("pause"); system("CLS");
    }
}

```