

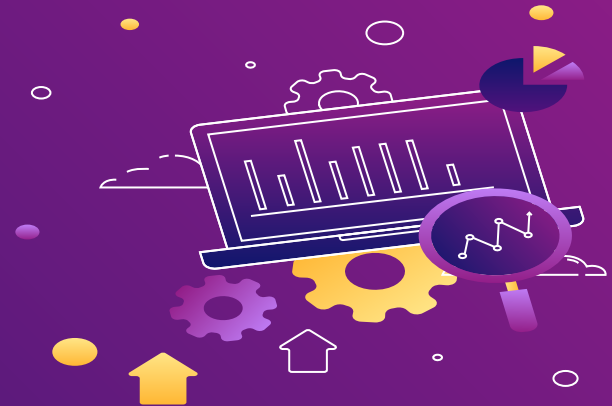


LINK IN PARK

Kelompok 8 (*Heatsink*)

Anggota Kelompok :

1. Bima Adinata Namara
(Animation + Display + Function)
2. Muhammad Brian Na'iman Hadi
(Login Menu + User Guide)
3. Ryan Daniel Asatama Sibagariang
(Payment Menu + Main Program)
4. Tio Larizky
(Main Menu + Selection Menu)



(2006574566)

(2006574534)

(2006574616)

(2006574433)

LATAR BELAKANG DAN TUJUAN



Pembangunan fasilitas umum membutuhkan lahan yang cukup luas bukan hanya untuk tempat fasilitas tersebut, namun juga untuk lahan parkir bagi para pengunjung yang mengunjungi tempat tersebut. Dengan keadaan tersebut, pengaturan lahan parkir sangat diperlukan terutama bagi manajemen fasilitas umum tersebut untuk mengetahui berapa kapasitas lahan parkir keseluruhan, berapa lahan parkir yang kosong, dan berapa biaya parkir kendaraan yang terparkir. Oleh karena itu, kami dari Kelompok Heatsink mencoba untuk menciptakan program yang akan memecahkan permasalahan tersebut dengan program kami yang bernama Link in Park.

Tujuan:

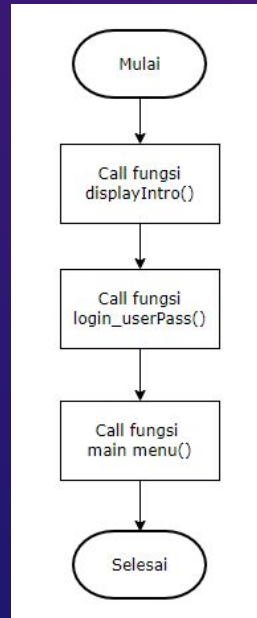
- Mempermudah operator untuk mengetahui berapa kapasitas keseluruhan lahan parkir.
- Mempermudah operator untuk mengetahui berapa sisa kapasitas lahan parkir yang tersedia bagi pengunjung.
- Mempermudah perhitungan dan proses pembayaran parkir bagi kendaraan yang akan keluar.





ALGORITMA

MAIN PROGRAM



FLOWCHART YANG DIMILIKI PROGRAM MEMILIKI DIMENSI YANG TIDAK DAPAT DIMUAT DALAM SLIDE. UNTUK MENGAKSES FLOWCHART DAPAT MELALUI LINK DRIVE ATAU DAPAT DISIMAK MELALUI VIDEO PENJELASAN DENGAN LINK YANG TERSEDIA DI BAWAH INI

LINK:

FLOWCHART : [HTTPS://BIT.LY/PROYEKAKHIR_GRP8](https://bit.ly/proyekakhir_grup8)
YOUTUBE : [HTTPS://BIT.LY/VIDEOPROYEKAKHIR_GRP8](https://bit.ly/vidioproyekakhir_grup8)





DISPLAY

```
void menuAnimation(int a, int b, int c, int d, int e, int f){
    Data j;

    printf("\t\t\t");
    for (j.loop = 0; j.loop < 70; j.loop++){
        printf("=");
        Sleep(a);
    }

    printf("\n");
    Sleep(b);
    printf("\t\t\t|\n");
    Sleep(c);
    printf("\t\t\t| - PROGRAM LINK IN PARK - |\n");
    Sleep(d);
    printf("\t\t\t|\n");
    Sleep(e);

    printf("\t\t\t");
    for (j.loop = 0; j.loop < 70; j.loop++){
        printf("=");
        Sleep(f);
    }

    printf("\n");
}
```





```
void login_userPass(){
    Login log;
    log.status = 0;

    while(log.status != 3){
        printf("Masukkan Username : ");
        gets(*log.username);
        printf("Masukkan Password : ");

        while(1){
            *log.pass = getch();

            if (*log.pass == 13){
                log.password[log.charPosition] = '\0';
                break;
            }

            else if (*log.pass == 8){
                if (log.charPosition > 0){
                    log.charPosition--;
                    log.password[log.charPosition] = '\0';
                    printf("\b \b");
                }
            }

            else if (*log.pass == 127){
                if (log.charPosition > 0){
                    for(log.status=0; log.status < log.charPosition; log.status++){
                        printf("\b \b");
                    }
                    log.charPosition--log.charPosition;
                    log.password[log.charPosition] = '\0';
                }
            }

            else if(*log.pass > 0 && *log.pass < 33){
                continue;
            }

            else {
                log.password[log.charPosition] = *log.pass;
                log.charPosition++;
                printf("*");
            }
        }
    }
}
```

LOGIN MENU

```
log.password[log.charPosition]='\0';

if(strcmp(*log.username, "operator")==0 && strcmp(log.password, "operator")==0){
    log.status = 3;
    Sleep(5);
    printf("\n\n\n\n\t\t\t\t\tSAKSES DITERIMA\n\n");
    Sleep(500);
    printf("\t\t\t\t\tloading");

    for (log.anim = 0; log.anim < 5; log.anim++){
        Sleep(850);
        printf(".");
    }

    else{
        printf("\n--- Username dan Password Salah ---\n\n");
        log.status++;
        if(log.status == 3){
            system("cls");
            printf("\n\t\t\t\t\t#####");
            printf("\n\t\t\t\t\t#   Anda Memasukkan User dan Password Lebih dari Tiga Kali   #");
            printf("\n\t\t\t\t\t#   Akses ditolak   #");
            printf("\n\t\t\t\t\t#####\n");
            exit(0);
        }

        log.charPosition-=log.charPosition;
    }
    system("cls");
}
```





SELECTION MENU

```
void pilihan_menu(){
    printf("\n\nBerikut daftar menu yang tersedia: \n");
    printf("[1]  Kendaraan Motor Masuk\n");
    printf("[2]  Kendaraan Mobil Masuk\n");
    printf("[3]  Daftar Kendaraan dan Kapasitas\n");
    printf("[4]  Kendaraan Keluar\n");
    printf("[5]  Panduan Penggunaan Program\n");
    printf("[6]  Keluar dari Program\n");
    printf("Pilih Angka yang Ingin Anda Masukkan:\n");
}
```

```
void main_menu(){
    Data input;
    kendaraan motor;
    kendaraan mobil;

    char* input_karcis = calloc(4096,sizeof(char));

    int *pilih, pilihan;
    pilih = &pilihan;
    motor.temp = 0;
    mobil.temp = 0;
    motor.count = 0;
    mobil.count = 0;
    motor.loop = 0;
    mobil.loop = 0;

    pilihan_menu();

    do{
        printf(">> ");
        scanf("%d", pilih);

        if (*pilih < 1 || *pilih > 6){
            printf("\nMasukkan Angka yang Sesuai!\n");
        }

        switch (*pilih){
```




```
case 1:
    if(motor.count < 50 + motor.temp){
        motor.count++;
        printf("Plat Motor\t: ");
        scanf("%255[^\n]s", &motor.plat[motor.count]);
        fflush(stdin);

        if(validasiPlat(motor, "Motor", 50, &motor.temp, &motor.count)){
            strcpy(motor.duplicatePlat[motor.count], motor.plat[motor.count]);
            srand(time(NULL));
            snprintf(motor.karcis[motor.count], 20, "#karcis%d", rand() % 99999 + 10000);
            printf("Karcis Anda\t: %s", motor.karcis[motor.count]);
            printf("\n-----\n");
            printf("Total Motor\t: %d\n\n", motor.count - motor.temp);
        }
    }
    else{
        printf("Tempat Parkir Motor Penuh\n\n");
    }
    break;
```

```
case 2:
    if(mobil.count < 10 + mobil.temp){
        mobil.count++;
        printf("Plat Mobil\t: ");
        scanf("%255[^\n]s", &mobil.plat[mobil.count]);
        fflush(stdin);

        if(validasiPlat(mobil, "Mobil", 10, &mobil.temp, &mobil.count)){
            strcpy(mobil.duplicatePlat[mobil.count], mobil.plat[mobil.count]);
            srand(time(NULL));
            snprintf(mobil.karcis[mobil.count], 20, "#karcis%d", rand() % 99999 + 10000);
            printf("Karcis Anda\t: %s", mobil.karcis[mobil.count]);
            printf("\n-----\n");
            printf("Total Mobil\t: %d\n\n", mobil.count - mobil.temp);
            strcpy(mobil.duplicatePlat[mobil.count], mobil.plat[mobil.count]);
        }
    }
    else{
        printf("Tempat Parkir Mobil Penuh\n\n");
    }
    break;
```

```
case 3:
    if(motor.count == 0 && mobil.count == 0){
        system("cls");
        kendaraanKosong();
        system("pause");
    }

    else if (motor.count > 0 && mobil.count == 0) {
        system("cls");
        resultTable(motor, "Motor", 50+motor.temp);
        system("pause");
    }

    else if (mobil.count > 0 && motor.count == 0){
        system("cls");
        resultTable(mobil, "Mobil", 10+mobil.temp);
        system("pause");
    }

    else {
        system("cls");
        resultTable(motor, "Motor", 50+motor.temp);
        resultTable(mobil, "Mobil", 10+mobil.temp);
        system("pause");
    }

    system("cls");
    menuAnimation(0, 0, 0, 0, 0, 0);
    pilihan_menu();

    break;
```

```
case 4:
if(motor.count != motor.temp || mobil.count != mobil.temp){
    system("cls");
    printf("\n          ===== " );
    printf("\n          Jenis Kendaraan dan Tarif Parkir " );
    printf("\n          1. Motor : Rp. 2000 untuk 1 Jam Pertama, Rp. 1000/jam berikutnya. " );
    printf("\n          2. Mobil : Rp. 5000 untuk 1 Jam Pertama, Rp. 4000/jam berikutnya. " );
    printf("\n          0. Kembali ke Menu Awal " );
    printf("\n          ===== " );
    printf("\n          Masukkan Kategori Kendaraan (Berdasarkan Nomor Pilihan) \n\n");

    do{
        printf("Pilih : ");
        scanf("%d", &input.jenis);
        if(input.jenis<0 || input.jenis>2){
            printf("Masukkan Input yang Sesuai\n\n");
        }
    }while(input.jenis < 0 || input.jenis > 2);

    switch (input.jenis){

        case 0:
            system("cls");
            menuAnimation(0, 0, 0, 0, 0, 0);
            pilihan_menu();
            break;

        case 1:
            if(motor.count==motor.temp){
                kendaraanKosong();
                system("pause");
                system("cls");
                menuAnimation(0, 0, 0, 0, 0, 0);
                pilihan_menu();
            }
            else {
                input.sentinel = 0;
                printf("\nMasukkan Karcis Anda (misal : #karcis24122) : ");
```

```
do{
    input.sentinel++;
    scanf(" %255[^\n]s", input_karcis);
    for (input.loop = 1; input.loop <= 50 + motor.temp; input.loop++){
        if (strcmp(input_karcis, motor.karcis[input.loop])==0){
            printf ("Plat Motor\t\t\t\t\t : %s\n", motor.plat[input.loop]);
            printf ("Lama Parkir (per jam)?\t\t\t\t\t : ");
            scanf ("%f", &input.parkir);
            printf ("Total Tagihan\t\t\t\t\t : %d", mtr(input.parkir));
            printf ("\nTotal Pembayaran\t\t\t\t\t : ");
            scanf ("%d", &input.nominal);
            input.y = input.nominal - mtr(input.parkir);
            printf("\n");

            while (input.y < 0){
                printf ("Total Uang yang Dimasukkan\t\t\t\t\t : %d\n", input.nominal);
                printf ("\n----- Maaf, Uang Tidak Mencukupi ----- \n\n");
                printf ("Tambahkan Uang untuk Mencukupi Pembayaran : ");
                scanf ("%d", &input.x);
                printf ("\n");
                input.nominal += input.x;
                input.y = input.nominal - mtr(input.parkir);
            }

            printf ("\nKembalian\t\t\t\t\t\t\t : %d", input.y);
            printf("\n=====");
            printf ("\nCetak Struk? (Y/N) : ");

            do{
                scanf ("%s", &input.tanya);
                if (input.tanya=="Y" || input.tanya=="y"){
                    system("cls");
                    cetakStruk (motor, input, input.loop, "Motor", mtr(input.parkir));
                }
                else if(input.tanya == 'N' || input.tanya == 'n'){
                    system("cls");
                    skipStruk();
                }
                else {
                    printf("\nHuruf Tidak valid, Coba Lagi!");
                    printf ("\nCetak Struk? (Y/N)\n>> ");
                }
            }while(input.tanya != 'y' && input.tanya != 'Y' && input.tanya != 'n' && input.tanya != 'N');
```



```

        strncpy(motor.plat[input.loop], " (-)", 20);
        strncpy(motor.duplicatePlat[input.loop], "", 20);
        strncpy(motor.karcis[input.loop], " (-)", 20);
        motor.temp++;
        input.sentinel = -1;
        break;
    }
    else {
        continue;
    }
}

if(input.loop == 51 + motor.temp && input.sentinel != 5){
    printf("\nKarcis Salah, Silakan Input Ulang Karcis (misal: #karcis24122)\n>> ");
}

while(input.sentinel != 5 && input.sentinel != -1);

if(input.sentinel == 5){
    system("cls");
    printf("\n\t\t\t\t\t-----=\n");
    printf("\t\t\t\t\t Anda Salah Memasukkan Karcis Sebanyak Lima Kali |\n");
    printf("\t\t\t\t\t Silakan Input Ulang Melalui Main Menu Kembali |\n");
    printf("\t\t\t\t\t-----=\n");
    system("pause");
    system("cls");
    menuAnimation(0, 0, 0, 0, 0, 0);
    pilihan_menu();
}

else {
    system("pause");
    system("cls");
    menuAnimation(0, 0, 0, 0, 0, 0);
    pilihan_menu();
}
}

break;

case 2:

if(mobil.count==mobil.temp){
    kendaraanKosong();
    system("pause");
    system("cls");
    menuAnimation(0, 0, 0, 0, 0, 0);
    pilihan_menu();
}
}

```

```

else {
    input.sentinel = 0;
    printf("\nMasukkan Karcis Anda (misal: #karcis24122) : ");
    do{
        input.sentinel++;
        scanf("%255[^\n]s", input_karcis);
        for (input.loop = 1; input.loop <= 50 + motor.temp; input.loop++){

            if (strcmp(input_karcis, mobil.karcis[input.loop])==0){
                printf ("Plat Mobil \t\t\t\t\t : %s\n", mobil.plat[input.loop]);
                printf ("Lama Parkir (per jam)?\t\t\t\t\t : ");
                scanf ("%f", &input.parkir);
                printf ("Total Tagihan\t\t\t\t\t : %d", mbl(input.parkir));
                printf ("\nTotal Pembayaran\t\t\t\t\t : " );
                scanf ("%d", &input.nominal);
                input.y = input.nominal - mbl(input.parkir);
                printf("\n");

                while (input.y < 0){
                    printf ("Total Uang yang Dimasukkan\t\t\t : %d\n", input.nominal);
                    printf ("\n----- Maaaf, Uang Tidak Mencukupi ----- \n\n");
                    printf ("Tambahkan Uang untuk Mencukupi Pembayaran : " );
                    scanf ("%d", &input.x);
                    printf("\n");
                    input.nominal += input.x;
                    input.y = input.nominal - mbl(input.parkir);
                }

                printf ("\nKembalian\t\t\t\t\t : %d", input.y);
                printf ("\n-----");
                printf ("\nCetak Struk? (Y/N) : ");

                do{
                    scanf ("%s", &input.tanya);
                    if (input.tanya=='Y' || input.tanya=='y'){
                        system("cls");
                        cetakStruk (mobil, input, input.loop, "Mobil", mbl(input.parkir));
                    }
                    else if(input.tanya == 'N' || input.tanya == 'n'){
                        system("cls");
                        skipStruk();
                    }
                    else {
                        printf("\nHuruf Tidak valid, Coba Lagi!");
                        printf ("\nCetak Struk? (Y/N)\n>> ");
                    }
                }
            }
        }
    }
}

```



PLAT CHECKING FUNCTION

```
int validasiPlat(kendaraan motor_mobil, char *jenis, int kapasitas, int* temp, int* loop){
    Data input;
    for (input.loop = 1; input.loop <= kapasitas + *temp; input.loop++){
        if (strcmp(motor_mobil.plat[*loop], motor_mobil.duplicatePlat[input.loop])==0){
            printf("Plat %s Tidak Boleh Sama\n\n", jenis);
            *loop-=1;
            return 0;
            break;
        }
        else {
            continue;
        }
    }
    return 1;
}
```

PAYMENT FUNCTION

```
int mbl (int b){
    int bayar = 5000;
    int i;
    if (b == 1){
        bayar = 5000;
    }
    else{
        for (i = 1; i < b; i++){
            bayar += 4000;
        }
    }
    return bayar;
}
```

```
int mtr (int a){
    int bayar = 2000;
    int i;
    if (a == 1){
        bayar = 2000;
    }
    else{
        for (i = 1; i < a; i++){
            bayar += 1000;
        }
    }
    return bayar;
}
```

OUTPUT EXIT PROGRAM

```
int stringAnimation_exitProgram(){
    Data k;
    char* animationString = (char*) malloc (53 * sizeof(int));

    system ("cls");
    animationString = "\t\t\t\t\t Terimakasih Telah Menggunakan Program Ini ^_^\n";
    printf("\t\t\t\t\t");
    for (k.loop=0;k.loop<26;k.loop++){
        printf("+");
        usleep(1000);
    }
    printf("+\n");
    for (k.loop=0;k.loop<53;k.loop++){
        printf("%c", animationString[k.loop]);
        usleep(10000);
    }
    printf("\n");
    printf("\t\t\t\t\t");
    for (k.loop=0;k.loop<26;k.loop++){
        printf("-");
        usleep(1000);
    }
    printf("+\n\n");

    system ("pause");
    return 0;

    free(animationString);
}
```