

LAPORAN RESMI FINAL PROJECT
MATA KULIAH PEMROGRAMAN LANJUT A
“ APLIKASI DIREKTORI (FOLDER) “



KELOMPOK 6 :

DIAN FITRIANA (18081010015)

DWI WAHYU EFFENDI (18081010017)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAWA TIMUR
2019

DEFINISI APLIKASI

Aplikasi Direktori (Folder) merupakan suatu aplikasi yang dibuat untuk memudahkan untuk pencarian dan pengurutan dari data-data yang sudah ada. Setiap item file dalam folder yang memiliki karakteristik srbagai berikut :

- ID File
- Nama File
- Jenis File (WORD, MUSIC, VIDEO, EXCEL, DLL)
- Ekstensi (EXE, DOCX, XLXS, DLL)
- Date Modified (Tanggal)
- Owner (Pemilik File)

Untuk operasinya yang ada pada aplikasi pelabuhan yakni :

- Tambah Data
- Ubah Data
- Pencarian Data Berdasarkan Kriteria Datanya (Nama File, ID File)
- Pengurutan Data Berdasarkan Kriteria Datanya (ID, Nama, Jenis Ekstensi, Owner)

METODE YANG DIGUNAKAN

1) Metode Sorting

Untuk metode sorting, kelompok kami menggunakan metode merge sort . Metode merge sort Merge sort merupakan algoritma pengurutan dalam ilmu komputer yang dirancang untuk memenuhi kebutuhan pengurutan atas suatu rangkaian data yang tidak memungkinkan untuk ditampung dalam memori komputer karena jumlahnya yang terlalu besar. Algoritma ini ditemukan oleh John von Neumann pada tahun 1945.

Algoritma pengurutan data merge sort dilakukan dengan menggunakan cara divide and conquer yaitu dengan memecah kemudian menyelesaikan setiap bagian kemudian menggabungkannya kembali. Pertama data dipecah menjadi 2 bagian dimana bagian pertama merupakan setengah (jika data genap) atau setengah minus satu (jika data ganjil) dari seluruh data, kemudian dilakukan pemecahan kembali untuk masing-masing blok sampai hanya terdiri dari satu data tiap blok.

Setelah itu digabungkan kembali dengan membandingkan pada blok yang sama apakah data pertama lebih besar daripada data ke-tengah+1, jika ya maka data ke-tengah+1 dipindah sebagai data pertama, kemudian data ke-pertama sampai ke-tengah digeser menjadi data ke-dua sampai ke-tengah+1, demikian seterusnya sampai menjadi satu blok utuh seperti awalnya. Sehingga metode merge sort merupakan metode yang membutuhkan fungsi rekursi untuk penyelesaiannya.

2) Metode Searching

Adalah algoritma pencarian yang mirip seperti binary search, karena sebelum pencarian dilakukan pengurutan terlebih dahulu. Keuntungan dari interpolation sort adalah, lebih cepat dalam pencarian. Kerugiannya adalah algoritma ini hanya bisa digunakan pada data yang terurut.

Konsep pencarian interpolation search adalah seperti pada pencarian dalam kamus. Jika kita ingin mencari kata dengan huruf depan S, maka kita tidak perlu mencarinya dari data awal namun dapat dicari langsung pada 2/3 atau 3/4 jumlah data.

interpolation sort, dirumuskan dengan :

$$\text{POSISI} = \text{keyword} - \text{data}[\text{low}] * (\text{high} - \text{low}) / (\text{data}[\text{high}] - \text{data}[\text{low}]) + \text{low}$$

- Jika $\text{data}[\text{posisi}] > \text{keyword}$, $\text{high} = \text{posisi} - 1$
- Jika $\text{data}[\text{posisi}] < \text{keyword}$, $\text{low} = \text{posisi} + 1$

HASIL FINAL PROJECT

Kode Sumber :

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>

void tambahdata();
void lihatdata();
void ubahdata();
void pencariandata();
void pengurutandata();

struct datemodified
{
    int tanggal;
    int bulan;
    int tahun;
    int cari;
    int urut;
}; typedef struct datemodified date;

struct jenis
{
    char jenisfile[50];
    char ekstensi[50];
}; typedef struct jenis jns;

struct file
{
```

```

        int idfile;
        char namafile[50];
        struct jenis jns;
        struct datemodified date;
        char owner[50];
    }; typedef struct file fl;
    fl folder[50];

    int i = 0, a = 0, b = 1, cariid, hasil;

    void tambahdata()
    {
        printf("\n\t=====");
        printf("\n\t ID FILE          : ");
        scanf("%d", &folder[i].idfile);
        printf("\t NAMA FILE          : ");
        scanf("%s", &folder[i].namafile);
        printf("\t JENIS FILE          : ");
        scanf("%s", &folder[i].jns.jenisfile);
        printf("\t EKSTENSI FILE          : ");
        scanf("%s", &folder[i].jns.ekstensi);
        printf("\t TANGGAL          : ");
        scanf("%d", &folder[i].date.tanggal);
        printf("\t BULAN          : ");
        scanf("%d", &folder[i].date.bulan);
        printf("\t TAHUN          : ");
        scanf("%d", &folder[i].date.tahun);
        printf("\t OWNER          : ");
        scanf("%s", &folder[i].owner);
        printf("\n\t=====");
        printf("\n\tTekan untuk kembali ke menu!");
        i++;
    }

```

```

        getch();
        system("cls");
    }

void lihatdata()
{
    if(i!=0)
    {
        while(a<i)
        {
            printf("\n\t=====");
            printf("\n\t      DATA FILE KE %d      =", b);
            printf("\n\t=====");
            printf("\n\t ID FILE          : %d", folder[a].idfile);
            printf("\n\t NAMA FILE          : %s", folder[a].namafile);
            printf("\n\t JENIS FILE         : %s", folder[a].jns.jenisfile);
            printf("\n\t EKSTENSI FILE      : %s", folder[a].jns.ekstensi);
            printf("\n\t TANGGAL           : %d", folder[a].date.tanggal);
            printf("\n\t BULAN             : %d", folder[a].date.bulan);
            printf("\n\t TAHUN             : %d", folder[a].date.tahun);
            printf("\n\t OWNER             : %s", folder[a].owner);
            printf("\n\t=====");

            a++;
            b++;
        }
        a=0;
        b=1;
        printf("\n\t TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!");
    }

    else if(i==0)
    {

```

```
printf("\n\tHARAP MENAMBAHKAN DATA TERLEBIH  
DAHULU!!!");
```

```
    }  
    getch();  
    system("cls");  
}
```

```
void mergeSort1(int low, int mid, int high)
```

```
{  
    int x = low;  
    int y = mid + 1;  
    int z = low;  
    struct file temp[high+1];  
  
    while(x <= mid && y <= high)  
    {  
        if (folder[x].idfile <= folder[y].idfile)  
        {  
            temp[z++] = folder[x++];  
        }  
        else if (folder[x].idfile >= folder[y].idfile)  
        {  
            temp[z++] = folder[y++];  
        }  
    }  
  
    while(x <= mid)  
    {  
        temp[z++] = folder[x++];  
    }  
  
    while(y <= high)
```

```

        {
            temp[z++] = folder[y++];
        }

    for(x = low; x <= high; x++)
    {
        folder[x] = temp[x];
    }
}

void mergeSort2(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;
    int z = low;
    struct file temp[high+1];

    while(x <= mid && y <= high)
    {
        if (strcmp(folder[x].namafile, folder[y].namafile) <= 0)
        {
            temp[z++] = folder[x++];
        }
        else if (strcmp(folder[x].namafile , folder[y].namafile) >= 0)
        {
            temp[z++] = folder[y++];
        }
    }

    while(x <= mid)
    {
        temp[z++] = folder[x++];
    }
}

```



```

    }

    while(y<=high)
    {
        temp[z++] = folder[y++];
    }

    for(x = low; x <= high; x++)
    {
        folder[x] = temp[x];
    }
}

void mergeSort3(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;
    int z = low;
    struct file temp[high+1];

    while(x <= mid && y <= high)
    {
        if (strcmp(folder[x].jns.jenisfile, folder[y].jns.jenisfile)<=0)
        {
            temp[z++] = folder[x++];
        }
        else if (strcmp(folder[x].jns.jenisfile, folder[y].jns.jenisfile)>=0)
        {
            temp[z++] = folder[y++];
        }
    }
}

```

```

while(x<=mid)
{
    temp[z++] = folder[x++];
}

while(y<=high)
{
    temp[z++] = folder[y++];
}

for(x = low; x <= high; x++)
{
    folder[x] = temp[x];
}
}

void mergeSort4(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;
    int z = low;
    struct file temp[high+1];

    while(x <= mid && y <= high)
    {
        if (strcmp(folder[x].jns.ekstensi, folder[y].jns.ekstensi) <= 0)
        {
            temp[z++] = folder[x++];
        }
        else if (strcmp(folder[x].jns.ekstensi, folder[y].jns.ekstensi) >= 0)
        {
            temp[z++] = folder[y++];
        }
    }
}

```

```

    }
}

while(x<=mid)
{
    temp[z++] = folder[x++];
}

while(y<=high)
{
    temp[z++] = folder[y++];
}

for(x = low; x <= high; x++)
{
    folder[x] = temp[x];
}
}

void mergeSort5(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;
    int z = low;
    struct file temp[high+1];

    while(x <= mid && y <= high)
    {
        if (strcmp(folder[x].owner, folder[y].owner) <= 0)
        {
            temp[z++] = folder[x++];
        }
    }
}

```

```

        else if (strcmp(folder[x].owner, folder[y].owner)>=0)
        {
            temp[z++] = folder[y++];
        }
    }

    while(x<=mid)
    {
        temp[z++] = folder[x++];
    }

    while(y<=high)
    {
        temp[z++] = folder[y++];
    }

    for(x = low; x <= high; x++)
    {
        folder[x] = temp[x];
    }
}

void mergeSort6(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;
    int z = low;
    struct file temp[high+1];

    while(x <= mid && y <= high)
    {
        if (folder[x].date.tanggal <= folder[y].date.tanggal)

```

```

    {
        temp[z++] = folder[x++];
    }
else if (folder[x].date.tanggal >= folder[y].date.tanggal)
{
    temp[z++] = folder[y++];
}
}

while(x<=mid)
{
    temp[z++] = folder[x++];
}

while(y<=high)
{
    temp[z++] = folder[y++];
}

for(x = low; x <= high; x++)
{
    folder[x] = temp[x];
}
}

void mergeSort7(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;
    int z = low;
    struct file temp[high+1];

```

```

        while(x <= mid && y <= high)
        {
            if (folder[x].date.bulan <= folder[y].date.bulan)
            {
                temp[z++] = folder[x++];
            }
            else if (folder[x].date.bulan >= folder[y].date.bulan)
            {
                temp[z++] = folder[y++];
            }
        }

        while(x<=mid)
        {
            temp[z++] = folder[x++];
        }

        while(y<=high)
        {
            temp[z++] = folder[y++];
        }

        for(x = low; x <= high; x++)
        {
            folder[x] = temp[x];
        }
    }

void mergeSort8(int low, int mid, int high)
{
    int x = low;
    int y = mid + 1;

```

```

int z = low;
struct file temp[high+1];

    while(x <= mid && y <= high)
    {
if (folder[x].date.tahun <= folder[y].date.tahun)
{
    temp[z++] = folder[x++];
}
else if (folder[x].date.tahun >= folder[y].date.tahun)
{
    temp[z++] = folder[y++];
}
}

while(x<=mid)
{
    temp[z++] = folder[x++];
}

while(y<=high)
{
    temp[z++] = folder[y++];
}

for(x = low; x <= high; x++)
{
    folder[x] = temp[x];
}
}

void partition(int low, int high, int cari)

```

```
{
    int mid;
    if(low < high)
    {
        if(cari==1)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort1(low, mid, high); //id file
        }
        else if(cari==2)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort2(low, mid, high); // nama file
        }
        else if(cari==3)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort3(low, mid, high); //jensi file
        }
        else if(cari==4)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort4(low, mid, high); //ekstensi file
        }
    }
}
```



```

        else if(cari==5)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort5(low, mid, high); //owner file
        }
        else if(cari==6)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort6(low, mid, high); //tanggal
        }
        else if(cari==7)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort7(low, mid, high); //bulan
        }
        else if(cari==8)
        {
            mid = (low + high) / 2;
            partition(low, mid, cari);
            partition(mid + 1, high, cari);
            mergeSort8(low, mid, high); //tahun
        }
    }
    else
    {
        return;
    }

```

```

    }
}

int intersearch1(int cari, int low, int high, int urut)
{
    int posisi;

    if(urut==1)
    {
        posisi = ((cari - folder[low].idfile) * ((high-low) + low))/ (folder[high].idfile -
folder[low].idfile);

        for(low = 0; low<=high ; low++)
        {
            if(folder[posisi].idfile == cari)
            {
                return posisi;
                break;
            }
            else if(folder[posisi].idfile < cari)
            {
                posisi = posisi + 1;
            }
        }
    }
    else if(urut==2)
    {
        posisi = ((cari - folder[low].date.tanggal) * ((high-low) + low))/
(folder[high].date.tanggal - folder[low].date.tanggal);

        for(low = 0; low<=high ; low++)
        {
            if(folder[posisi].date.tanggal == cari)

```

```

        {
            return posisi;
            break;
        }
        else if(folder[posisi].date.tanggal < cari)
        {
            posisi = posisi + 1;
        }
    }
}
else if(urut==3)
{
    posisi = ((cari - folder[low].date.bulan) * ((high-low) + low))/
(folder[high].date.bulan - folder[low].date.bulan);

    for(low = 0; low<=high ; low++)
    {
        if(folder[posisi].date.bulan == cari)
        {
            return posisi;
            break;
        }
        else if(folder[posisi].date.bulan < cari)
        {
            posisi = posisi + 1;
        }
    }
}
else if(urut==4)
{
    posisi = ((cari - folder[low].date.tahun) * ((high-low) + low))/
(folder[high].date.tahun - folder[low].date.tahun);

```

```

for(low = 0; low<=high ; low++)
{
    if(folder[posisi].date.tahun == cari)
    {
        return posisi;
        break;
    }
    else if(folder[posisi].date.tahun < cari)
    {
        posisi = posisi + 1;
    }
}

if (low > high)
{
    return -1;
}
}

void updatedata(int i)
{
    printf("\n\t=====");
    printf("\n\t ID FILE          : ");
    scanf("%d", &folder[i].idfile);
    printf("\t NAMA FILE              : ");
    scanf("%s", &folder[i].namafile);
    printf("\t JENIS FILE                : ");
    scanf("%s", &folder[i].jns.jenisfile);
    printf("\t EKSTENSI FILE              : ");
    scanf("%s", &folder[i].jns.ekstensi);
    printf("\t TANGGAL                    : ");

```

```

scanf("%d", &folder[i].date.tanggal);
printf("\t BULAN                : ");
scanf("%d", &folder[i].date.bulan);
printf("\tTAHUN                : ");
scanf("%d", &folder[i].date.tahun);
printf("\t OWNER                : ");
scanf("%s", &folder[i].owner);
printf("\t=====");
}

void ubahdata()
{
    int b = 0;
    if(i==0)
    {
        printf("DATA TIDAK DI TEMUKAN MASUKKAN DATA TERLEBIH
DAHULU!!!");
    }
    else
    {
        partition(0, i - 1, 1); //Memanggil fungsi partition
        printf("\n\t=====");
        printf("\n\tMASUKKAN ID FILE UNTUK MENGUBAH DATA : ");
        scanf("%d", &cariid);
        hasil = intersearch1(cariid, 0, i-1, 1);
        if(hasil==-1)
        {
            printf("\n\tDATA TIDAK DI TEMUKAN");
        }
        else
        {
            updatedata(hasil);

```

```

    }

    }
    getch();
    system("cls");
}

void pencariandata()
{
    int pilihan;
    printf("\n\t=====");
    printf("\n\t1. CARI BERDASARKAN ID FILE");
    printf("\n\t2. CARI BERDASARKAN TANGGAL");
    printf("\n\t3. CARI BERDASARKAN BULAN");
    printf("\n\t4. CARI BERDASARKAN TAHUN");
    printf("\n\tMASUKKAN PILIHAN URUT YANG DI INGINKAN : ");
    scanf("%d", &pilihan);

    if(pilihan==1)
    {
        printf("\n\t=====");
        printf("\n\tMASUKKAN ID FILE YANG INGIN DI CARI : ");
        scanf("%d", &cariid);
        partition(0, i-1, 1);
        hasil = intersearch1(cariid, 0, i-1, 1);

        if(hasil==-1)
        {
            printf("\n\tDATA TIDAK DI TEMUKAN");
        }
        else
        {
            printf("\n\t=====");

```

```

        printf("\n\t=      DATA FILE KE %d      =", hasil+1);
        printf("\n\t=====");
        printf("\n\t ID FILE          : %d", folder[hasil].idfile);
        printf("\n\t NAMA FILE            : %s", folder[hasil].namafile);
        printf("\n\t JENIS FILE              : %s",
folder[hasil].jns.jenisfile);
        printf("\n\t EKSTENSI FILE          : %s",
folder[hasil].jns.ekstensi);
        printf("\n\t TANGGAL                : %d",
folder[hasil].date.tanggal);
        printf("\n\t BULAN                  : %d",
folder[hasil].date.bulan);
        printf("\n\t TAHUN                  : %d",
folder[hasil].date.tahun);
        printf("\n\t OWNER                  : %s", folder[hasil].owner);
        printf("\n\t=====");
    }
}
else if(pilihan==2)
{
    printf("\n\t=====");
    printf("\n\tMASUKKAN TANGGAL YANG INGIN DI CARI : ");
    scanf("%d", &cariid);
    partition(0, i-1, 6);
    hasil = intersearch1(cariid, 0, i-1, 2);

    if(hasil==-1)
    {
        printf("\n\tDATA TIDAK DI TEMUKAN");
    }
    else
    {
        printf("\n\t=====");
        printf("\n\t=      DATA FILE KE %d      =", hasil+1);

```

```

        printf("\n\t=====");
        printf("\n\t ID FILE          : %d", folder[hasil].idfile);
        printf("\n\t NAMA FILE            : %s", folder[hasil].namafile);
        printf("\n\t JENIS FILE              : %s",
folder[hasil].jns.jenisfile);
        printf("\n\t EKSTENSI FILE          : %s",
folder[hasil].jns.ekstensi);
        printf("\n\t TANGGAL                : %d",
folder[hasil].date.tanggal);
        printf("\n\t BULAN                  : %d",
folder[hasil].date.bulan);
        printf("\n\t TAHUN                  : %d",
folder[hasil].date.tahun);
        printf("\n\t OWNER                  : %s", folder[hasil].owner);
        printf("\n\t=====");
    }
}
else if(pilihan==3)
{
    printf("\n\t=====");
    printf("\n\tMASUKKAN BULAN YANG INGIN DI CARI : ");
    scanf("%d", &cariid);
    partition(0, i-1, 7);
    hasil = intersearch1(cariid, 0, i-1, 3);

    if(hasil==-1)
    {
        printf("\n\tDATA TIDAK DI TEMUKAN");
    }
    else
    {
        printf("\n\t=====");
        printf("\n\t      DATA FILE KE %d      =", hasil+1);

```



```

        printf("\n\t=====");
        printf("\n\t ID FILE          : %d", folder[hasil].idfile);
        printf("\n\t NAMA FILE            : %s", folder[hasil].namafile);
        printf("\n\t JENIS FILE              : %s",
folder[hasil].jns.jenisfile);
        printf("\n\t EKSTENSI FILE          : %s",
folder[hasil].jns.ekstensi);
        printf("\n\t TANGGAL                : %d",
folder[hasil].date.tanggal);
        printf("\n\t BULAN                  : %d",
folder[hasil].date.bulan);
        printf("\n\t TAHUN                  : %d",
folder[hasil].date.tahun);
        printf("\n\t OWNER                  : %s", folder[hasil].owner);
        printf("\n\t=====");
    }
}
else if(pilihan==4)
{
    printf("\n\t=====");
    printf("\n\tMASUKKAN TAHUN YANG INGIN DI CARI : ");
    scanf("%d", &cariid);
    partition(0, i-1, 8);
    hasil = intersearch1(cariid, 0, i-1, 4);

    if(hasil==-1)
    {
        printf("\n\tDATA TIDAK DI TEMUKAN");
    }
    else
    {
        printf("\n\t=====");
        printf("\n\t      DATA FILE KE %d      =", hasil+1);

```

```

                printf("\n\t=====");
                printf("\n\t ID FILE          : %d", folder[hasil].idfile);
                printf("\n\t NAMA FILE            : %s", folder[hasil].namafile);
                printf("\n\t JENIS FILE              : %s",
folder[hasil].jns.jenisfile);
                printf("\n\t EKSTENSI FILE          : %s",
folder[hasil].jns.ekstensi);
                printf("\n\t TANGGAL                : %d",
folder[hasil].date.tanggal);
                printf("\n\t BULAN                  : %d",
folder[hasil].date.bulan);
                printf("\n\t TAHUN                  : %d",
folder[hasil].date.tahun);
                printf("\n\t OWNER                  : %s", folder[hasil].owner);
                printf("\n\t=====");
            }
        }
    else
    {
        printf("\n\tDATA TIDAK DI TEMUKAN");
    }

    getch();
    system("cls");
}

void pengurutandata()
{
    int pilihan;
    printf("\n\t=====
=====");
    printf("\n\tURUTKAN DATA BERDASARKAN");
    printf("\n\t1. ID FILE");
    printf("\n\t2. NAMA FILE");

```

```

printf("\n\t3. JENIS FILE");
printf("\n\t4. EKSISTENSI FILE");
printf("\n\t5. OWNER FILE");
printf("\n\t6. TANGGAL");
printf("\n\t7. BULAN");
printf("\n\t8. TAHUN");
printf("\n\t=====
====");
printf("\n\tMASUKKAN PILIHAN URUTAN YANG DI INGINKAN : ");
scanf("%d", &pilihan);
system("cls");

switch(pilihan)
{
    case 1 :
        partition(0, i-1, 1);
        lihatdata(); break;
    case 2 :
        partition(0, i-1, 2);
        lihatdata(); break;
    case 3 :
        partition(0, i-1, 3);
        lihatdata(); break;
    case 4 :
        partition(0, i-1, 4);
        lihatdata(); break;
    case 5 :
        partition(0, i-1, 5);
        lihatdata(); break;
    case 6 :
        partition(0, i-1, 6);
        lihatdata(); break;
}

```

```

        case 7 :
            partition(0, i-1, 7);
            lihatdata(); break;
        case 8 :
            partition(0, i-1, 8);
            lihatdata(); break;
        default :
            printf("\n\tMASUKKAN PILIHAN YANG BENAR (1-5)");
            getch();
            system("cls");
            break;
    }
}

main()
{
    int pilihan;

    printf("\n\t=====");
    printf("\n\t= FINAL PROJECT PEMROGRAMAN LANJUT A =");
    printf("\n\t=  APLIKASI DIREKTORI (FOLDER)  =");
    printf("\n\t=====");
    printf("\n\t= DIAN FITRIANA  [18081010015] =");
    printf("\n\t= DWI WAHYU EFFENDI [18081010017] =");
    printf("\n\t=====");
    printf("\n\tTEKAN TERSERAH UNTUK MEMULAI PROGRAM!!");
    getch();
    system("cls");
    menu:
    printf("\n\t=====");
    printf("\n\t= PILIH MENU OPSI DIREKTORI (FOLDER)  =");
    printf("\n\t=====");

```

```

printf("\n\t1. TAMBAH DATA\n");
printf("\n\t2. LIHAT DATA\n");
printf("\n\t3. UBAH DATA\n");
printf("\n\t4. PENCARIAN DATA\n");
printf("\n\t5. PENGURUTAN DATA\n");
printf("\n\t6. Keluar\n");
printf("\n\t=====");
printf("\n\tMASUKKAN PILIHAN OPSI : ");
scanf("%d", &pilihan);
system("cls");

switch(pilihan)
{
    case 1 :
        tambahdata();
        goto menu;
    case 2 :
        lihatdata();
        goto menu;
    case 3 :
        ubahdata();
        goto menu;
    case 4 :
        pencariandata();
        goto menu;
    case 5 :
        pengurutandata();
        goto menu;
    case 6 :
        printf("\n\t===== TERIMA KASIH TELAH MENGGUNAKAN
PROGRAM SEDERHANA INI =====");
        getch();

```

```
        return 0;
    default :
        printf("\n\tMASUKKAN PILIHAN YANG BENAR (1-6)");
        getch();
        system("cls");
        goto menu;
        break;
    }
}
```

HASIL EKSEKUSI KODE SUMBER :

1. Tampilan awal yang berisi penjelasan program dan nama pembuat.

```
=====
= FINAL PROJECT PEMROGRAMAN LANJUT A =
=   APLIKASI DIREKTORI (FOLDER)   =
=====
= DIAN FITRIANA   [18081010015] =
= DWI WAHYU EFFENDI [18081010017] =
=====
TEKAN TERSERAH UNTUK MEMULAI PROGRAM!!
```

2. Pilih opsi 1 untuk menambahkan data.

```
=====
=   PILIH MENU OPSI DIREKTORI (FOLDER)   =
=====
1. TAMBAH DATA
2. LIHAT DATA
3. UBAH DATA
4. PENCARIAN DATA
5. PENGURUTAN DATA
6. Keluar
=====
MASUKKAN PILIHAN OPSI : 1
```

3. Masukkan data yang di perlukan.

```
=====
ID FILE           : 123
NAMA FILE         : pmlan
JENIS FILE        : document
EKSTENSI FILE     : docx
TANGGAL           : 31
BULAN             : 05
TAHUN             : 2000
OWNER             : wahyu
=====
Tekan untuk kembali ke menu!
```

4. Pilih opsi 2 untuk melihat data yang telah di inputkan.

```
=====
=   PILIH MENU OPSI DIREKTORI (FOLDER)   =
=====
1. TAMBAH DATA
2. LIHAT DATA
3. UBAH DATA
4. PENCARIAN DATA
5. PENGURUTAN DATA
6. Keluar
=====
MASUKKAN PILIHAN OPSI : 2
```


5. Tampilan semua data yang telah di inputkan.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 123
NAMA FILE         : pmlan
JENIS FILE        : document
EKSTENSI FILE     : docx
TANGGAL           : 12
BULAN             : 12
TAHUN             : 2019
OWNER            : wahyu
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

6. Pilih opsi 3 untuk mengubah data.

```
=====
=   PILIH MENU OPSI DIREKTORI (FOLDER)   =
=====
1. TAMBAH DATA
2. LIHAT DATA
3. UBAH DATA
4. Pencarian DATA
5. PENGURUTAN DATA
6. Keluar
=====
MASUKKAN PILIHAN OPSI : 3
```

7. Masukkan id file yang ingin di ubah, dan masukkan data baru.

```
=====
MASUKKAN ID FILE UNTUK MENGUBAH DATA : 123

=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER             : dedy
=====
```

8. Pilih opsi 4 untuk pencarian data.

```
=====
=   PILIH MENU OPSI DIREKTORI (FOLDER)   =
=====
1. TAMBAH DATA
2. LIHAT DATA
3. UBAH DATA
4. Pencarian DATA
5. PENGURUTAN DATA
6. Keluar
=====
MASUKKAN PILIHAN OPSI : 4
```

9. Pilih 1 untuk mencari data berdasarkan id file.

```
=====
1. CARI BERDASARKAN ID FILE
2. CARI BERDASARKAN TANGGAL
3. CARI BERDASARKAN BULAN
4. CARI BERDASARKAN TAHUN
MASUKKAN PILIHAN URUT YANG DI INGINKAN : 1
```

10. Masukkan id file yang ingin di cari, dan tampilan jika data di temukan.

```
=====
1. CARI BERDASARKAN ID FILE
2. CARI BERDASARKAN TANGGAL
3. CARI BERDASARKAN BULAN
4. CARI BERDASARKAN TAHUN
MASUKKAN PILIHAN URUT YANG DI INGINKAN : 1

=====
MASUKKAN ID FILE YANG INGIN DI CARI : 234

=====
=          DATA FILE KE 1          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER             : dian
=====
```

11. Masukkan id file yang ingin di cari, dan jika data tidak di temukan.

```
=====
1. CARI BERDASARKAN ID FILE
2. CARI BERDASARKAN TANGGAL
3. CARI BERDASARKAN BULAN
4. CARI BERDASARKAN TAHUN
MASUKKAN PILIHAN URUT YANG DI INGINKAN : 1

=====
MASUKKAN ID FILE YANG INGIN DI CARI : 213

DATA TIDAK DI TEMUKAN
```

12. Pilih 2 untuk mencari data berdasarkan tanggal.

```
=====
1. CARI BERDASARKAN ID FILE
2. CARI BERDASARKAN TANGGAL
3. CARI BERDASARKAN BULAN
4. CARI BERDASARKAN TAHUN
MASUKKAN PILIHAN URUT YANG DI INGINKAN : 2

=====
MASUKKAN TANGGAL YANG INGIN DI CARI : 13

=====
=          DATA FILE KE 1          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
```

13. Pilih 3 untuk mencari data berdasarkan bulan.

```
=====
1. CARI BERDASARKAN ID FILE
2. CARI BERDASARKAN TANGGAL
3. CARI BERDASARKAN BULAN
4. CARI BERDASARKAN TAHUN
MASUKKAN PILIHAN URUT YANG DI INGINKAN : 3

=====
MASUKKAN BULAN YANG INGIN DI CARI : 7

=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
```

14. Pilih 4 untuk mencari data berdasarkan tahun.

```
=====
1. CARI BERDASARKAN ID FILE
2. CARI BERDASARKAN TANGGAL
3. CARI BERDASARKAN BULAN
4. CARI BERDASARKAN TAHUN
MASUKKAN PILIHAN URUT YANG DI INGINKAN : 4

=====
MASUKKAN TAHUN YANG INGIN DI CARI : 2001

=====
=          DATA FILE KE 1          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
```

15. Pilih opsi 5 untuk mengurutan data.

```
=====
=   PILIH MENU OPSI DIREKTORI (FOLDER)   =
=====
1. TAMBAH DATA
2. LIHAT DATA
3. UBAH DATA
4. PENCARIAN DATA
5. PENGURUTAN DATA
6. Keluar
=====
MASUKKAN PILIHAN OPSI : 5
```

16. Masukkan pilihan urutan yang di inginkan.

```
=====
URUTKAN DATA BERDASARKAN
1. ID FILE
2. NAMA FILE
3. JENIS FILE
4. EKSISTENSI FILE
5. OWNER FILE
6. TANGGAL
7. BULAN
8. TAHUN
=====
MASUKKAN PILIHAN URUTAN YANG DI INGINKAN :
```

17. Pilih 1 mengurutkan data berdasarkan id file.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

18. Pilih 2 mengurutkan data berdasarkan nama file.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```


19. Pilih 3 mengurutkan data berdasarkan jenis file.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

20. Pilih 3 mengurutkan data berdasarkan ekstensi file.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

21. Pilih 5 mengurutkan data berdasarkan owner file.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

22. Pilih 6 mengurutkan data berdasarkan tanggal file.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

23. Pilih 7 mengurutkan data berdasarkan bulan.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

24. Pilih 8 mengurutkan data berdasarkan tahun.

```
=====
=          DATA FILE KE 1          =
=====
ID FILE           : 345
NAMA FILE         : statkom
JENIS FILE        : video
EKSTENSI FILE     : mkv
TANGGAL           : 15
BULAN             : 7
TAHUN             : 2001
OWNER            : rehan
=====
=====
=          DATA FILE KE 2          =
=====
ID FILE           : 234
NAMA FILE         : sisop
JENIS FILE        : musik
EKSTENSI FILE     : mp3
TANGGAL           : 13
BULAN             : 5
TAHUN             : 2007
OWNER            : dian
=====
=====
=          DATA FILE KE 3          =
=====
ID FILE           : 789
NAMA FILE         : metnum
JENIS FILE        : powerpoint
EKSTENSI FILE     : ppt
TANGGAL           : 14
BULAN             : 1
TAHUN             : 2010
OWNER            : dedy
=====
TEKAN TERSERAH UNTUK KEMBALI KE MENU!!!
```

25. Pilih opsi untuk keluar dari program.

```
=====
=   PILIH MENU OPSI DIREKTORI (FOLDER)   =
=====
1. TAMBAH DATA
2. LIHAT DATA
3. UBAH DATA
4. PENCARIAN DATA
5. PENGURUTAN DATA
6. Keluar
=====
MASUKKAN PILIHAN OPSI : 6
```

26. Tampilan keluar dari program.

```
===== TERIMAKASIH TELAH MENGGUNAKAN PROGRAM SEDERHANA INI =====
```

PENUTUP DAN KESIMPULAN

Dari final project kali ini kami membuat program dengan menggunakan bahasa C. Program yang kami buat yakni program tentang aplikasi Direktori (Folder). Di dalam program tersebut kami menggunakan beberapa fungsi yakni fungsi tambah data, fungsi ubah data, fungsi sorting data, fungsi searching data, fungsi lihat data serta fungsi utama. Program yang kami buat ini bersifat dinamis, jadi user dapat menginputkan data terserah sesuai dengan keinginannya.

Program aplikasi pelabuhan ini merupakan program yang mengimplementasikan metode *CRU (Creat, Read, Update)*. Metode ini digunakan untuk menambah, melihat, serta mengupdate data. Disamping itu, dalam aplikasi Direktori (Folder) kali ini juga terdapat metode sorting yakni metode *Merge Sort*. Dengan adanya metode tersebut dapat mempermudah pengguna program dalam mengakses berbagai fitur yang telah kami sediakan. Sekian laporan yang dapat kami jabarkan terkait final project mata kuliah pemrograman lanjut dengan judul “ Aplikasi Direktori (Folder)”. Dengan minimnya pengetahuan yang kami miliki dalam pembuatan laporan ini, sehingga kami mohon kritik dan saran yang membangun agar laporan ini dapat lebih bermanfaat. Terima kasih.

DAFTAR PUSTAKA

<https://arfianhidayat.com/algorithm-merge-sort>

<http://haajarrifat.blogspot.com/2016/04/pencarian-interpolasi-interpolation.html>

<https://kazwini13.wordpress.com/2010/12/22/searching-interpolation-search/>