

```
#include <stdio.h>
#include <time.h>
#include <windows.h>
#define MAX 10

//int max;

int data[MAX];

void tukar(int *, int *);
void tulis();
void shellsort();
int sequential(int);
int binary(int);

main()
{
    int i;
    long s;
    char pil, cari, jwb, hsl;

    do
    {
        printf("Data sebelum diurutkan : \n");
        for(i=0;i<MAX;i++)
        {
            srand(time(&s)*(i+1));
```

```

        data[i]=rand()%100;
        printf("%d ",data[i]);
    }
    printf("\n\nPilihan : \n");
    printf("1. Sequential Searching \n");
    printf("2. Binary Searching\n");
    printf("\nMasukkan Pilihan anda : ");
    scanf("%d",&pil);
    printf("\nMasukkan bilangan yang dicari : ");
    scanf("%d",&cari);
    switch (pil){
    case 1:
        hsl=sequential(cari);
        if (hsl==0)printf("\nData tidak ditemukan!");
        else printf("\nData ditemukan pada index ke-
%d.",hsl);
        break;
    case 2:
        shellsort();
        hsl=binary(cari);
        if (hsl==0)printf("\nData tidak ditemukan!");
        else printf("\nData ditemukan pada index ke-
%d.",hsl);
        break;
    default:
        printf("Pilihan anda salah. Pilih 1 s/d 3!");
    }
}

```

```
    fflush(stdin);  
    printf("\nMau coba lagi?");  
    scanf("%c",&jwb);  
    }  
    while(jwb=='y');  
}
```

```
void tukar(int *a, int *b)  
{  
    int temp;  
    temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
void tulis()  
{  
    int i;  
    for(i=0;i<MAX;i++)  
        printf("%d ",data[i]);  
    printf("\n");  
}
```

```
void shellsort()  
{
```

```

int Jarak, i, j;
int Sudah;

Jarak = MAX;

while(Jarak > 1){
    Jarak = Jarak / 2;
    Sudah = 0;
    while(!Sudah){
        Sudah = 1;
        for(j=0; j<MAX-Jarak; j++){
            i = j + Jarak;
            if(data[j] > data[i]){
                tukar(&data[j], &data[i]);
                Sudah = 0;
            }
        }
    }
}

tuliskan();

}

int sequential(int cari)
{
    int hasil=0, i=0;

```

```

while ((hasil==0) && (i<MAX))
{
    if(cari==data[i])
    {
        hasil=1;
        return i+1;
    }
    else
        i++;
}
if(hasil==0)
    return hasil;
}

```

```

int binary(int cari)
{
    int L=0, R=MAX-1, M, hasil=0;
    int count=0;

    while((hasil==0) && (L<=R))
    {
        M=(L+R)/2;
        if(cari==data[M])
        {
            hasil=1;
            return M+1;
        }
    }
}

```

```
    }  
    else if (cari<data[M])  
    {  
        R=M-1;  
    }  
    else  
    {  
        L=M+1;  
    }  
}  
if(hasil==0)  
    return hasil;  
}
```