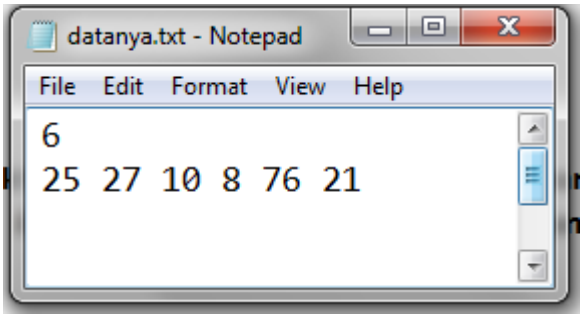
Materi Praktikum Algoritma dan Pemrograman II
Sorting Algorithm (Algoritma Pengurutan)



1. Bubble Sort

```
#include<stdio.h>
#include<conio.h>
void cetak(int A[],int n,int k,int j){
   int i=0;
   printf("Tahap %d - i=%d : ",k+1,j);
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   printf("\n");
}
void BubbleSort(int A[], int n){
   //isi algoritma ada pada poin a/b/c/d
}
main(){
    int i, k, n, A[50];
    FILE *Fin;
    Fin=fopen("datanya.txt","r");
    fscanf(Fin, "%d", &n);
    i=0;
    while(i<n){
      fscanf(Fin, "%d", &A[i]);
      i++;
   }
   BubbleSort(A,n);
   printf("Hasil akhir : ");
   i=0;
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   getch();
}
```

a. Menaik – Depan

```
int i,k,X;
for(k=0; k<n-1 ; k++){
      for  (i=n-1;  i>k;  i-- ){
           if (A[i] < A[i-1]){
              X = A[i];
              A[i] = A[i-1];
              A[i-1] = X;
           }
           cetak(A,n,k,i);
      }
}
```

b. Menaik – Belakang

```
int i,k,X;
for(k=0; k<n-1 ; k++){
      for  (i=0;  i<n-k-1;  i++ ){
           if (A[i] > A[i+1]){
              X = A[i];
              A[i] = A[i+1];
              A[i+1] = X;
           }
           cetak(A,n,k,i);
      }
}
```

c. Menurun – Depan

```
int i,k,X;
for(k=0; k<n-1 ; k++){
      for  (i=n-1;  i>k;  i-- ){
           if (A[i] > A[i-1]){
              X = A[i];
              A[i] = A[i-1];
              A[i-1] = X;
           }
           cetak(A,n,k,i);
      }
}
```

d. Menurun – Belakang

```
int i,k,X;
for(k=0; k<n-1 ; k++){
      for  (i=0;  i<n-k-1;  i++ ){
           if (A[i] < A[i+1]){
              X = A[i];
              A[i] = A[i+1];
              A[i+1] = X;
           }
           cetak(A,n,k,i);
      }
}
```

2. Selection Sort

```
#include<stdio.h>
#include<conio.h>
void cetak(int A[], int n,int j){
   int i=0;
   printf("Tahap %d : ",j+1);
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   printf("\n");
}
void selectionSort(int A[], int n){
//isi algoritma ada pada poin a/b/c/d
}
main(){
    int i, k, n, A[50];
    FILE *Fin;
    Fin=fopen("datanya.txt","r");
    fscanf(Fin, "%d", &n);
    i=0;
    while(i<n){
      fscanf(Fin, "%d", &A[i]);
      i++;
   }
   selectionSort(A,n);
   printf("Hasil akhir : ");
   i=0;
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   getch();
}
```

a. Menaik – Depan

```
   int i,j,x;
   for(i=0; i<n-1; i++){
      int imin = i;
      for(j=i+1; j<n; j++){
         if(A[j]<A[imin])
              imin=j;
       }
          x=A[imin];
          A[imin]=A[i];
          A[i]=x;
          cetak(A,n,i);

   }
```

b. Menaik – Belakang

```
int i,j,x;
for(i=n-1; i>0; i--){
   int imax = i;
   for(j=i-1; j>=0; j--){
      if(A[j]>A[imax])
           imax=j;
    }
       x=A[imax];
       A[imax]=A[i];
       A[i]=x;
       cetak(A,n,n-i);

}
```

c. Menurun – Depan

```
int i,j,x;
for(i=0; i<n-1; i++){
   int imax = i;
   for(j=i+1; j<n; j++){
      if(A[j]>A[imax])
           imax=j;
    }
       x=A[imax];
       A[imax]=A[i];
       A[i]=x;
       cetak(A,n,i);

}
```

d. Menurun – Belakang

```
int i,j,x;
for(i=n-1; i>0; i--){
   int imin = i;
   for(j=i-1; j>=0; j--){
      if(A[j]<A[imin])
           imin=j;
    }
       x=A[imin];
       A[imin]=A[i];
       A[i]=x;
       cetak(A,n,n-i);

}
```

3. Insertion Sort

```
#include<stdio.h>
#include<conio.h>
void cetak(int A[], int n, int j){
   int i=0;
   printf("Tahap %d : ",j);
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   printf("\n");
}
void InsertionSort(int L[], int n){
//isi algoritma ada pada poin a/b/c/d
}
main(){
    int i, k, n, A[50];
    FILE *Fin;
    Fin=fopen("datanya.txt","r");
    fscanf(Fin, "%d", &n);
    i=0;
    while(i<n){
      fscanf(Fin, "%d", &A[i]);
      i++;
   }
   InsertionSort(A,n);
   printf("Hasil akhir : ");
   i=0;
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   getch();
}
```

a. Menaik – Depan

```
   int i,j,a,y;
   for (i=1;i<n;i++){
      y=L[i];
      j=i-1;
      a=0;
      while(j>=0 && !a){
         if(y<L[j]){
            L[j+1]=L[j];
            j--;
         }
         else
            a=1;
```

```
      }
      L[j+1]=y;
      cetak(L,n,i);
   }
```

b. Menaik – Belakang

```
   int i,j,a,y;
   for (i=n-2;i>=0;i--){
      y=L[i];
      j=i+1;
      a=0;
      while(j<n && !a){
         if(y>L[j]){
            L[j-1]=L[j];
            j++;
         }
         else
            a=1;
      }
      L[j-1]=y;
      cetak(L,n,n-i-1);
   }
```

c. Menurun – Depan

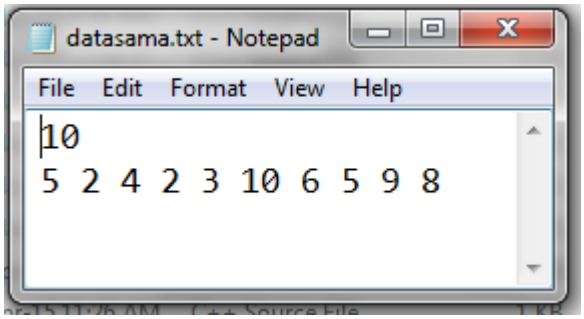
```
   int i,j,a,y;
   for (i=1;i<n;i++){
      y=L[i];
      j=i-1;
      a=0;
      while(j>=0 && !a){
         if(y>L[j]){
            L[j+1]=L[j];
            j--;
         }
         else
            a=1;
      }
      L[j+1]=y;
      cetak(L,n,i);
   }
```

d. Menurun – Belakang

```
   for (i=n-2;i>=0;i--){
      y=L[i];
      j=i+1;
      a=0;
      while(j<n && !a){
         if(y<L[j]){
            L[j-1]=L[j];
            j++;
         }
```

```
      else
         a=1;
   }
   L[j-1]=y;
   cetak(L,n,n-i-1);
}
```

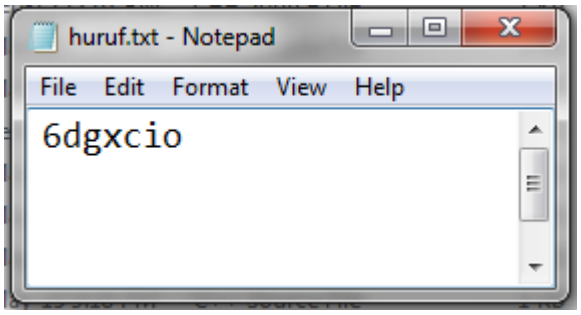
Contoh program sorting dengan beberapa data/elemen array yang sama nilainya:




```
#include<stdio.h>
#include<conio.h>
void cetak(int A[],int n,int k,int j){
   int i=0;
   printf("Tahap %d - i=%d : ",k+1,j);
   while(i<n){
       printf("%d ", A[i]);
       i++;
   }
   printf("\n");
}
void BubbleSort(int A[], int n){
int i,k,X;
for(k=0; k<n-1 ; k++){
      for  (i=0;  i<n-k-1;  i++ ){
           if (A[i] > A[i+1]){
              X = A[i];
              A[i] = A[i+1];
              A[i+1] = X;
           }
           cetak(A,n,k,i);
      }
}
}
main(){
    int i, k, n, A[50];
    FILE *Fin;
    Fin=fopen("datasama.txt","r");
    fscanf(Fin, "%d", &n);
    i=0;
    while(i<n){
      fscanf(Fin, "%d", &A[i]);
      i++;
   }
   BubbleSort(A,n);
   printf("Hasil akhir : ");
```

```
    i=0;
    while(i<n){
        printf("%d ", A[i]);
        i++;
    }
    getch();
}
```

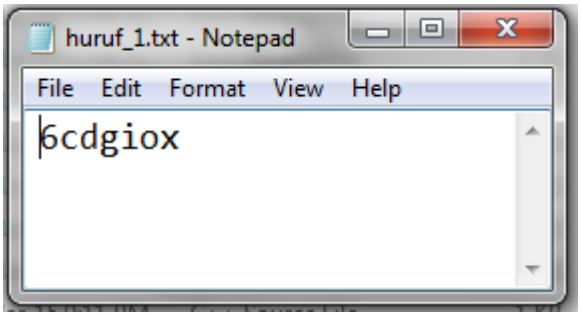
Contoh program sorting untuk data/elemen array berupa karakter:




```
#include<stdio.h>
#include<conio.h>
void cetak(int A[],int n,int k,int j){
    int i=0;
    printf("Tahap %d - i=%d : ",k+1,j);
    while(i<n){
        printf("%c ", A[i]);
        i++;
    }
    printf("\n");
}
void BubbleSort(int A[], int n){
int i,k,X;
for(k=0; k<n-1 ; k++){
      for  (i=0;  i<n-k-1;  i++ ){
           if (A[i] > A[i+1]){
              X = A[i];
              A[i] = A[i+1];
              A[i+1] = X;
           }
           cetak(A,n,k,i);
      }
}
}
main(){
    int i, k, n, A[50];
    FILE *Fin;
    Fin=fopen("huruf.txt","r");
    fscanf(Fin, "%d", &n);
    i=0;
    while(i<n){
```

```
      fscanf(Fin, "%c", &A[i]);
      i++;
   }
   BubbleSort(A,n);
   printf("Hasil akhir : ");
   i=0;
   while(i<n){
       printf("%c ", A[i]);
       i++;
   }
   getch();
}
```

Tambahan



Contoh program searching untuk data/elemen array berupa karakter dengan squential search :

```
#include<stdio.h>
#include<conio.h>
int Cari(char L[], int n, char x){
    int i=0;
    int a=0;
    while(i<n){
      if(L[i]==x){
        a=1;
        break;
      }
      i++;
   }
   if(a==1)
      return i;
   else
      return -1;
}
main(){
    int n,i=0, index;
    char x, L[50];
    FILE *Fin;
    Fin=fopen("huruf_1.txt", "r");
    fscanf(Fin, "%d", &n);
    while(i<n){
```

```
       fscanf(Fin, "%c", &L[i]);
       i++;
    }
    scanf("%c", &x);
    index = Cari(L,n,x);
    printf("%d", index);
    fclose(Fin);
    getch();
}
```

Contoh program searching untuk data/elemen array berupa karakter dengan binary search :

```
#include<stdio.h>
#include<conio.h>
int SS(int n, char L[],char x){
   int i,j,k;
   bool flag=false;
   i=0; j=n-1;
   while(!flag && i<=j){
      k=(i+j)/2;
      if(L[k]==x) flag=true;
      else if(L[k]<x) i=k+1;
      else j=k-1;
   }
   if(flag) return k;
   else return -1;
}
main(){
    char x, L[30];
    int n, i=0, indeks;
    scanf("%c", &x);
    FILE *Fin;
    Fin= fopen("huruf_1.txt", "r");
    fscanf(Fin, "%d", &n);
    while(i<n){
       fscanf(Fin, "%c", &L[i]);
       i++;
    }
    i=0;
    while(i<n){
       printf("%c ", L[i]);
       i++;
    }
    indeks = SS(n,L,x);
    printf("%d", indeks);
   getch();
}
```