# LAPORAN

**Algoritma dan Struktur Data**

A logo with a yellow and blue design

Description automatically generated

**Nama : Arif Muhammad Ihsan Marbun**

**Kelas : 1 D4 Teknik Informatika A**

**NRP : 3124600001**

1. Listing Latihan
2. Menggunakan bubble and shell dengan outputan hasil dan cms
3. Program

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define n 10  int i, j, c, m, s, kondisi=0, did\_swap=1;  int awal[n] = {1, 2, 3, 4, 5, 6, 7, 8, 9 ,10};  int a[n];  void backup\_array ();  void cek\_menu (int, int);  void tampil ();  void insertion (int);  void selection (int);  void bubble (int);  void shell (int);  void tukar (int \*, int \*);  void reset\_perhitungan();  void display\_perhitungan();  int main(){  int pilih1, pilih2;  while (pilih1 != 5){  reset\_perhitungan();  backup\_array ();  puts ("MENU METODE SORTING");  puts ("1. Insertion Sort");  puts ("2. Selection Sort");  puts ("3. Bubble");  puts ("4. Shell");  puts ("5. Keluar");  printf ("Pilihan anda [1/2/3/4/5]: ");  scanf ("%d", &pilih1);  if (pilih1 != 5){  puts ("MENU METODE SORTING");  puts ("1. Ascending");  puts ("2. Descanding");  printf ("Pilihan anda [1/2]: ");  scanf ("%d", &pilih2);  printf ("Data sebelum diproses : ");  tampil ();  cek\_menu (pilih1, pilih2);  printf ("Data setelah diproses : ");  tampil ();  puts ("");  }  }  }  void backup\_array (){  for (i=0; i<n; i++){  a[i] = awal[i];  }  }  void cek\_menu (int menu1, int menu2){  if (menu1 == 1){  insertion(menu2);  display\_perhitungan();  }else if (menu1 == 2){  selection(menu2);  display\_perhitungan();  }else if (menu1 == 3){  bubble (menu2);  display\_perhitungan();  }else if (menu1 == 4){  shell (menu2);  display\_perhitungan();  }else  exit(0);  }  void insertion(int jenis) {  int key;  for (i = 1; i < n; i++) {  j = i - 1;  key = a[i];  m++; // Penyimpanan key  while (j >= 0) {  c++; // Perbandingan loop while  if (jenis == 1)  kondisi = key < a[j];  else  kondisi = key > a[j];  if (kondisi) {  a[j + 1] = a[j];  m++; // Pergeseran elemen  j--;  } else {  break;  }  }  a[j + 1] = key;  m++; // Penyisipan key  }  }  void selection (int jenis){  int maxmin;  for (i=0; i<n-1; i++){  maxmin = i;  j = i+1;  while (j < n){  if (jenis == 1)// maxmin = nilai minimum  kondisi = a[j] < a[maxmin];  else // maxmin = nilai maximum  kondisi = a[j] > a[maxmin];  if (kondisi){  maxmin = j;  }  j++;  c++;  }  tukar (&a[maxmin], &a[i]);  s++;  }  }  void bubble(int jenis) {  int batas = n - 1;  for (i = 0; i < n - 1; i++) {  did\_swap = 0;  for (j = 0; j < batas; j++) {  c++; // Perbandingan loop  if (jenis == 1)  kondisi = a[j] > a[j + 1];  else  kondisi = a[j] < a[j + 1];  if (kondisi) {  tukar(&a[j], &a[j + 1]);  did\_swap = 1;  s++; // Penukaran  }  }  batas--;  if (!did\_swap) break;  }  }  void shell(int jenis){  int jarak = n/2;  while (jarak >= 1){  did\_swap = 1;  while (did\_swap){  did\_swap = 0;  for (i=0; i<n-jarak; i++){  if (jenis == 1)  kondisi = a[i] > a[jarak+i];  else  kondisi = a[i] < a[jarak+i];  if (kondisi){  tukar(&a[i], &a[jarak+i]);  did\_swap = 1;  s++;  }  c++;  }  }  jarak /= 2;  }  }  void display\_perhitungan() {  printf(" Hasil Perhitungan \n");  printf("Perbandingan (C): %d\n", c);  printf("Pergeseran (M): %d\n", m);  printf("Penukaran (S): %d\n", s);  printf("Total Operasi: %d\n\n", c + m + s);  }  void reset\_perhitungan(){  c = 0;  m = 0;  s = 0;  }  void tukar (int \*x, int \*y){  int temp;  temp = \*x;  \*x = \*y;  \*y = temp;  m += 3;  }  void tampil (){  i= 0;  for (i=0; i<n; i++){  printf ("%d ", a[i]);  }  puts ("");  } |

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

1. Menggunakan bubble and shell tanpa outputan hasil dan cms

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <math.h>  #include <time.h>  int i, j, c, m, s, kondisi=0, did\_swap=1;  int n;  void generate(int []);  void backup\_array (int [], int []);  void cek\_menu (int, int, int []);  void tampil (int []);  void insertion (int, int []);  void selection (int, int []);  void bubble (int, int []);  void shell (int, int []);  void tukar (int \*, int \*);  void c\_m\_s ();  int main(){  int jml, pilih1, pilih2;  printf ("Berapa jumlah data (maks 100000)? ");  scanf ("%d", &n);  int x[n];  int a[n];  while (pilih1 != 5){  c = 0, m = 0, s = 0;  srand(time(NULL));  //puts ("a");  generate(x);  backup\_array (a, x);  puts ("MENU METODE SORTING");  puts ("1. Insertion Sort");  puts ("2. Selection Sort");  puts ("3. Bubble");  puts ("4. Shell");  puts ("5. Keluar");  printf ("Pilihan anda [1/2/3/4/5]: ");  scanf ("%d", &pilih1);  if (pilih1 != 5){  puts ("MENU METODE SORTING");  puts ("1. Ascending");  puts ("2. Descanding");  printf ("Pilihan anda [1/2]: ");  scanf ("%d", &pilih2);  //printf ("Data sebelum diproses : ");  //tampil (a);  cek\_menu (pilih1, pilih2, a);  //printf ("Data setelah diproses : ");  //tampil (a);  //c\_m\_s ();  puts ("");  }  }  }  void backup\_array (int a[], int x[]){  for (i=0; i<n; i++){  a[i] = x[i];  }  }  void cek\_menu (int menu1, int menu2, int a[]){  clock\_t start, end;  double waktu;  if (menu1 == 1)  insertion(menu2,a);  else if (menu1 == 2)  selection(menu2, a);  else if (menu1 == 3)  bubble (menu2, a);  else if (menu1 == 4)  shell (menu2, a);  else  exit(0);  end = clock() - start;  waktu = ((double)end) / CLOCKS\_PER\_SEC;  printf ("Waktu komputasi : %g \n", waktu);  }  void insertion (int jenis, int a[]){  int key;  for (i=1; i<n; i++){  j = i-1;  key = a[i];  m++;  while (j>=0){  if(jenis == 1)  kondisi = key < a[j];  else  kondisi = key > a[j];  if (kondisi){  a[j+1] = a[j];  m++;  c++;  j--;  }  else  break;  }  a[j+1] = key;  m++;  }  }  void selection (int jenis, int a[]){  int maxmin;  for (i=0; i<n-1; i++){  maxmin = i;  j = i+1;  while (j < n){  if (jenis == 1)// maxmin = nilai minimum  kondisi = a[j] < a[maxmin];  else // maxmin = nilai maximum  kondisi = a[j] > a[maxmin];  if (kondisi){  maxmin = j;  }  j++;  c++;  }  tukar (&a[maxmin], &a[i]);  s++;  }  }  void bubble (int jenis, int a[]){  int batas = n-1;  for (i=0; i<n-1; i++){  did\_swap = 1;  if (did\_swap){  did\_swap = 0;  for (j=0; j<batas; j++){  if (jenis == 1)  kondisi = a[j] > a[j+1];  else  kondisi = a[j] < a[j+1];  if (kondisi){  tukar(&a[j], &a[j+1]);  did\_swap = 1;  s++;  }  c++;  }  batas--;  }  }  }  void shell(int jenis, int a[]){  int jarak = n/2;  while (jarak >= 1){  did\_swap = 1;  while (did\_swap){  did\_swap = 0;  for (i=0; i<n-jarak; i++){  if (jenis == 1)  kondisi = a[i] > a[jarak+i];  else  kondisi = a[i] < a[jarak+i];  if (kondisi){  tukar(&a[i], &a[jarak+i]);  did\_swap = 1;  s++;  }  c++;  }  }  jarak /= 2;  }  }  void tukar (int \*x, int \*y){  int temp;  temp = \*x;  \*x = \*y;  \*y = temp;  m += 3;  }  void tampil (int a[]){  i= 0;  for (i=0; i<n; i++){  printf ("%d ", a[i]);  }  puts ("");  }  void generate(int x[]){  for(i=0; i<n; i++){  x[i] = rand()/1000;  }  }  void c\_m\_s (){  printf ("Hasil compare : %d\n", c);  printf ("Hasil movement : %d\n", m);  printf ("Hasil swap : %d\n", s);  } |

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.