M. Bahrul Ulum, S.Kom, M.Kom



Modul Kuliah Struktur Data

POINTER

Program Studi Teknik Informatika

Fakultas Ilmu Komputer

Universitas Esa Unggul

2018

**POINTER**

**Pendahuluan**

**Setiap kali kita mendeklarasikan variabel, C/C++ akan menyediakan alokasi memori untuk masing-masing variabel. Ukuran memori yang dialokasikan berbeda-beda tergantung dari tipe data variabelnya. Misalnya untuk integer 2 bytes, untuk float 4 bytes, dst. Jadi,** setiap variabel yang kita deklarasikan mempunyai 2 atribut yaitu *address* dan *value*.

**Pengertian Pointer**

Pointer (variabel penunjuk) adalah suatu variabel yang berisi alamat memori dari suatu variabel lain. Pointer merupakan variabel level rendah yang dapat digunakan untuk menunjuk nilai integer, character, float, double, atau single, dan bahkan tipe-tipe data lain yang didukung oleh bahasa C.

Variabel biasa, sifatnya statis dan sudah pasti, sedangkan pada pointer sifatnya *dinamis* dan dapat lebih fleksibel. Variabel pointer yang tidak menunjuk pada nilai apapun berarti memiliki nilai NULL, dan disebut sebagai **dangling pointer** karena nilainya tidak diinisialisasi dan tidak dapat diprediksi.

1. **Operator Alamat / Dereference Operator(&)**

Setiap variabel yang dideklarasikan, disimpan dalam sebuah lokasi memori dan pengguna biasanya tidak mengetahui di alamat mana data tersebut disimpan. Dalam C++, untuk mengetahui alamat tempat penyimpanan data, dapat digunakan tanda ampersand (&) yang dapat diartikan “alamat”.

***Contoh :***

Bil1 = &Bil2;

dibaca: isi variabel bil1 sama dengan alamat bil2

1. **Operator Reference (\*)**

Penggunaan operator ini, berarti mengakses nilai sebuah alamat yang ditunjuk oleh variabel pointer.

***Contoh :***

Bil1=\*Bil2;

dibaca: bil1 sama dengan nilai yang ditunjuk oleh bil2.

**Deklarasi Pointer**

**Seperti halnya variabel yang lain, variabel pointer juga harus dideklarasikan terlebih dahulu sebelum digunakan.**

**Bentuk Umum : tipe\_data \*nama\_pointer;**

**Contoh : int \*nilai;**

**char \*huruf;**

**Pendeklarasian variabel pointer menggunakan tanda \* sebelum nama variabelnya, sedangkan untuk menampilkan nilai yang ditunjuk oleh suatu variabel pointer, juga digunakan operator \* (tanda asterisk). Jika diinginkan untuk menampilkan alamat tempat penyimpanan nilai yang ditunjuk oleh suatu variabel pointer, digunakan operator & (tanda ampersand).**

Pendeklarasian variabel biasa dan pointer:

//variabel biasa

int nilai1 = 4;

float nilai2 = 3.5;

char nama[10] = “anton”; //array of char (string)

//variabel pointer

int \*nilai\_p1; //dangling pointer

int \*nilai\_p2 = &nilai1; //menunjuk ke tipe data int

char \*nilai\_p3 = nama; //menunjuk ke tipe data array of char

char \*nilai\_p4 = &nama[0];

Ilustrasi Pointer:

4

nilai1

int

3.5

nilai2

float

“eka”

nama

char

nilai\_p1

nilai\_p2

nilai\_p3

ff4c

ff4a

ff47

4

3.5

eka

**Operasi pada Pointer**

1. **Operasi Penugasan**

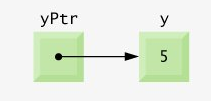
Suatu variable pointer seperti halnya variabel yang lain, juga bisa mengalami operasi penugasan. Nilai dari suatu variable pointer dapat disalin ke variable pointer yang lain Operator (&) dalam kaitannya dengan pointer adalah operator yang mengembalikan alamat memori dari operandnya. Contohnya :

int y = 5; /\*deklarasi variabel y\*/

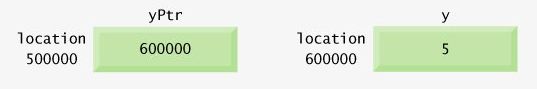
int \*yPtr; /\*deklarasi variabel pointer yPtr\*/

yPtr = &y; /\*mengisi variabel pointer yPtr dengan alamat dari variabel y\*/

Maka representasi dari operasi penugasan di atas adalah



Representasi yang lain, misalnya asumsikan bahwa variabel y berada pada alokasi memori 600000 dan variabel pointer yPtr berada pada alokasi memori 500000.

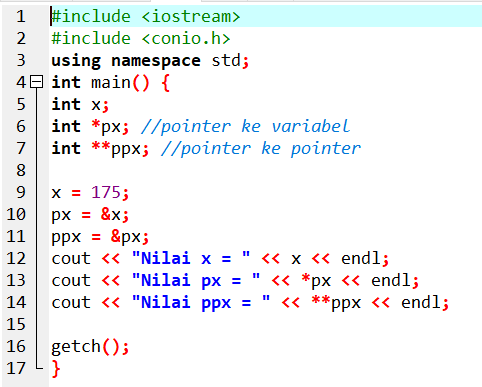


Contoh program (untuk dicoba)

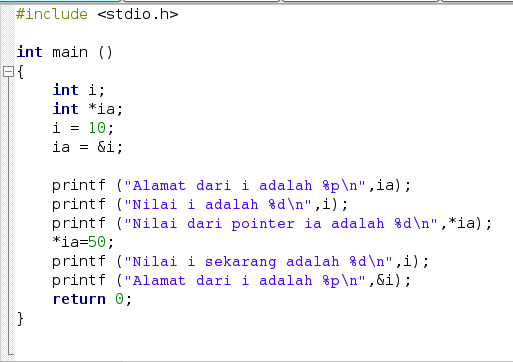
1. **Program 1**



1. **Program 2**



1. **Program 3**



Operasi pointer:

1. Operasi Pemberian nilai

Contoh 1:

#include <stdio.h>

#include <conio.h>

int main(){

float nilai,\*p1,\*p2;

nilai = 14.54;

printf("nilai = %2.2f, alamatnya %p\n",nilai,&nilai);

p1 = &nilai;

printf("nilai p1 = %2.2f, p1 menunjuk alamat %p\n",\*p1,p1);

//pada awalnya p2 masih dangling pointer

printf("mula-mula nilai p2 = %2.2f, p2 menunjuk alamat %p\n",\*p2,p2);

p2 = p1; //operasi pemberian nilai, berarti alamat x2 sama dengan x1

printf("sekarang nilai p2 = %2.2f, p2 menunjuk alamat %p\n",\*p2,p2);

getch();

}

Contoh 2:

#include <stdio.h>

#include <conio.h>

int main(){

int \*p,a=25,b;

//p masih dangling

printf("nilai a = %d di alamat a = %p,\n",a,&a);

printf("nilai p di alamat = %p\n",p);

p = &a;

printf("nilai p = %d di alamat %p\n",\*p,p);

//b diisi dengan nilai yang berasal dari nilai

//variabel a yang ditunjuk oleh pointer p

b = \*p;

printf("nilai b = %d di alamat %p\n",b,&b);

getch();

}

Contoh 3:

#include <conio.h>

#include <stdio.h>

int main(){

int a=25,b=12,t;

int \*p,\*q;

p = &a;

q = &b;

printf("nilai yang ditunjuk p = %d di alamat %p\n",\*p,p);

printf("nilai yang ditunjuk q = %d di alamat %p\n",\*q,q);

//Contoh kasus, penukaran nilai 2 variabel dengan pointer

t = \*p;

\*p = \*q;

\*q = t;

printf("nilai yang ditunjuk p sekarang = %d di alamat %p\n",\*p,p);

printf("nilai yang ditunjuk q sekarang = %d di alamat %p\n",\*q,q);

getch();

}

Contoh 4:

#include <stdio.h>

#include <conio.h>

int main(){

int a,\*p;

p=&a;

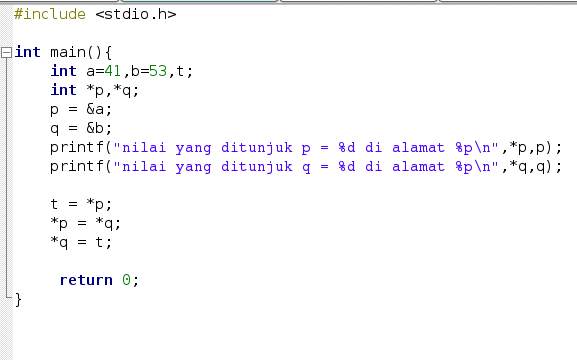
\*p=25;

printf("nilai a = %d",a);

}

Pertanyaan :

1. Perintah apa yang digunakan untuk menampilkan alamat dari variabel i, namun tanpa melalui pengaksesan variabel pointer ia?



Pertanyaan :

1. t = \*p;

\*p = \*q;

\*q = t;

Fungsi di atas ini (dengan pointer) untuk apakah itu?

Untuk membantu, coba tambahkan perintah di bawah ini pada program di atas:

printf("nilai yang ditunjuk p sekarang = %d di alamat %p\n",\*p,p);

printf("nilai yang ditunjuk q sekarang = %d di alamat %p\n",\*q,q);

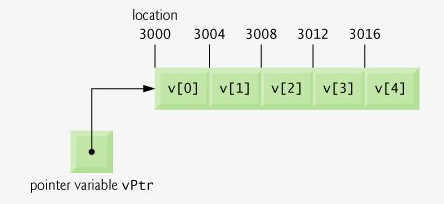
1. **Operasi Aritmatika**

Suatu variabel pointer hanya dapat dilakukan operasi aritmatika dengan nilai integer saja. Operasi yang biasa dilakukan adalah operasi penambahan dan pengurangan. Operasi penambahan dengan suatu nilai menunjukkan lokasi data berikutnya (index selanjutnya) dalam memori. Begitu juga operasi pengurangan.

Asumsikan kita telah mendeklarasikan sebuah array float v[5], dan anggap elemen pertamanya berada pada lokasi 3000 di memori. Lalu kita deklarasikan sebuah variabel pointer \*vPtr, ada dua cara untuk mengisi variabel pointer \*vPtr dengan alamat dari elemen pertama array v, yaitu :

float \*vPtr = v;

float \*vPtr = &v[0];

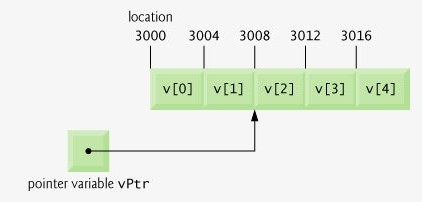


Jika kita melakukan operasi aritmatika seperti

vPtr += 2;

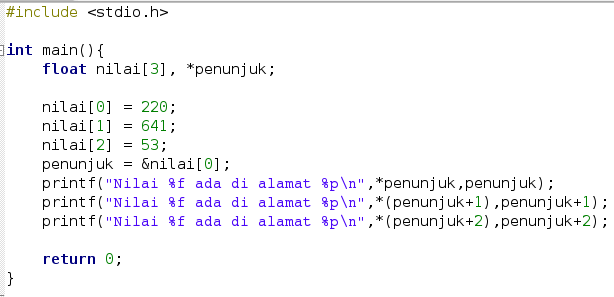
dalam perhitungan aritmatika biasa, maka vPtr yang semula berisi 3000 menjadi 3000 + 2 = 3002, namun ini tidak berlaku pada pointer. Operasi aritmatika pada pointer di atas artinya adalah vPtr yang semula menunjuk pada v[0] sekarang menunjuk ke v[2].

Karena vPtr menunjuk alamat memori dari variabel v yang bertipe float, maka vPtr sekarang berisi 3008 (3000 + 2 \* 4).



Contoh program (untuk dicoba) Operasi Aritmatika

1. Program 1



1. Program 2

#include <stdio.h>

#include <conio.h>

int main(){

int a,b=10,\*p,\*q;

p=&a;

\*p=25;

printf("nilai a = %d\n",a);

printf("alamat p = %p\n",p);

q=&b;

printf("alamat q = %p\n",q);

printf("nilai a + b = %d\n",(\*p+\*q));

//posisi alamat p menjadi bergeser, nilai berubah

p=p+1;

printf("nilai p = %d, alamat = %p\n",\*p,&p);

q=q-1;

printf("nilai q = %d, alamat = %p\n",\*q,&q);

getch();

}

Pertanyaan

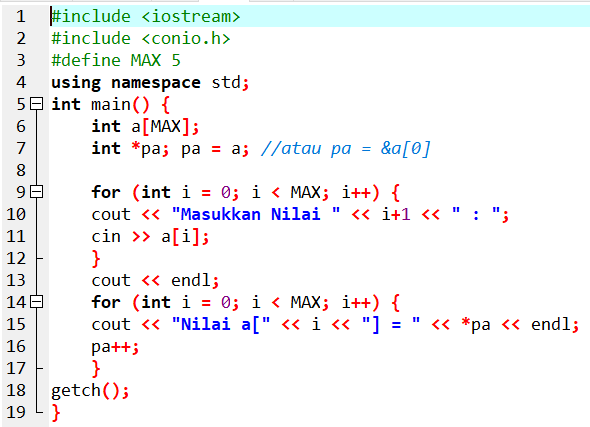
1. Berapa selisih nilai (menggunakan heksadesimal) antar penunjuk dengan penunjuk+1? Hal apa yangmempengaruhi besarnya selisih nilai tersebut? Untuk membantu, coba ganti tipe data variabel nilai dari *int* menjadi *float*.

**Pointer pada Array**

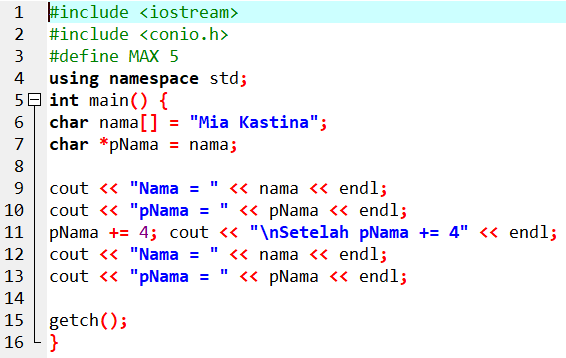
Pada array, pointer hanya perlu menunjuk pada alamat elemen pertama saja karena letak alamat array sudah berurutan pada memori.

Contoh program (untuk dicoba)

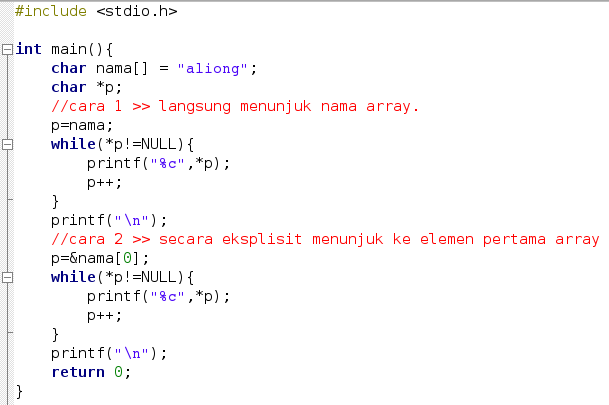
1. **Program 1**



1. **Program 2**



1. **Program 3**



Pertanyaan

1. Tambahkan potongan kode program pada program di atas untuk membalik suatu kata/kalimat dengan menggunakan pointer!

**Pointer pada Struct**

Pointer dapat bertipe **apa saja**, bahkan tipe buatan kita sendiri.

Contoh program (untuk dicoba)

Pointer pada Struct:

1. Program 1

#include <stdio.h>

#include <conio.h>

typedef struct{

int nim;

int umur;

float ipk;

} Mahasiswa;

Mahasiswa m;

Mahasiswa \*p = &m;

int main(){

//struct biasa

m.nim=123;

m.ipk=3.2;

m.umur=23;

printf("nim = %d\n",m.nim);

printf("ipk = %f\n",m.ipk);

printf("umur = %d\n",m.umur);

//struct pointer

p->ipk = 3.5;

p->nim = 321;

p->umur = 32;

printf("nim = %d\n",p->nim);

printf("ipk = %f\n",p->ipk);

printf("umur = %d\n",p->umur);

//mengacu pada variabel aslinya

printf("nim = %d\n",m.nim);

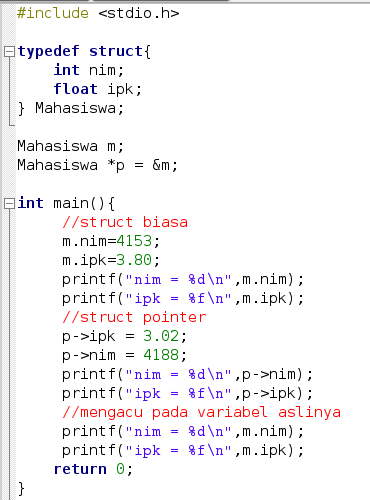
printf("ipk = %f\n",m.ipk);

printf("umur = %d\n",m.umur);

getch();

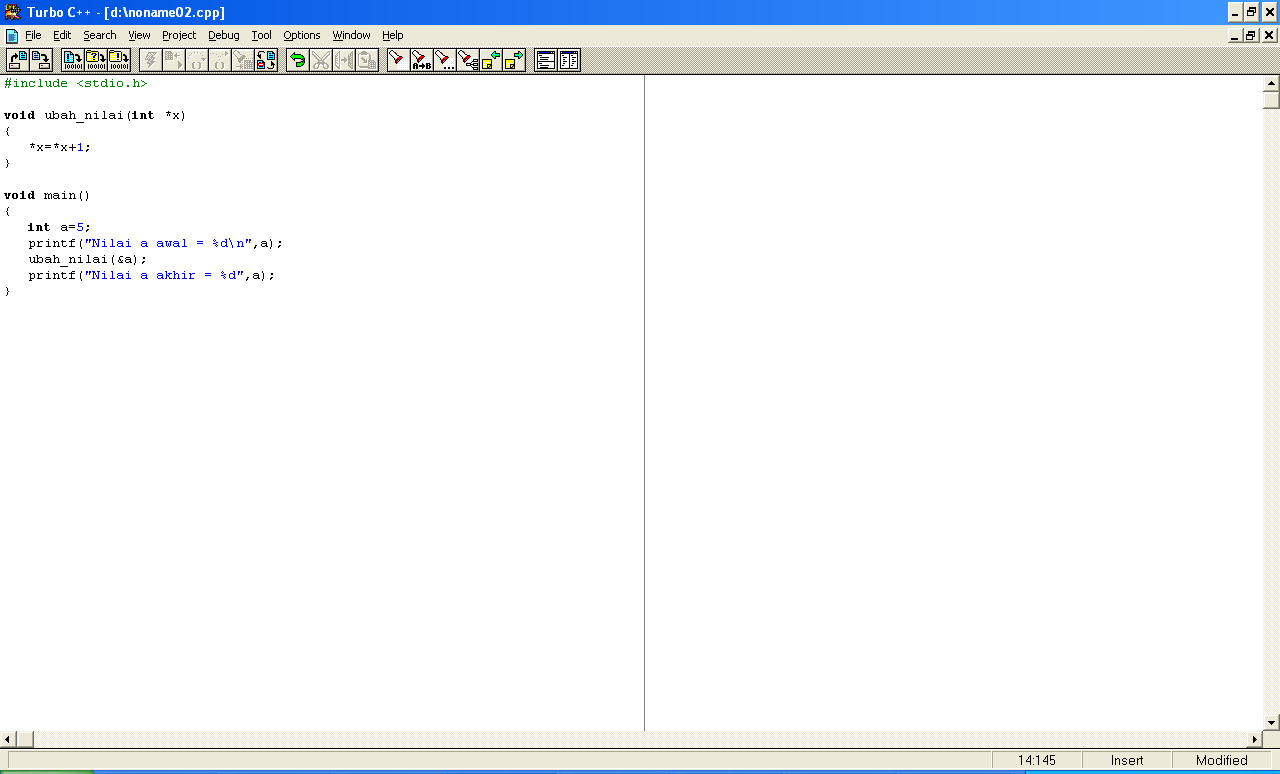
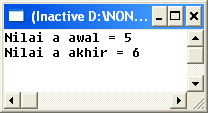
}

1. Program 2



**Pointer pada Fungsi**

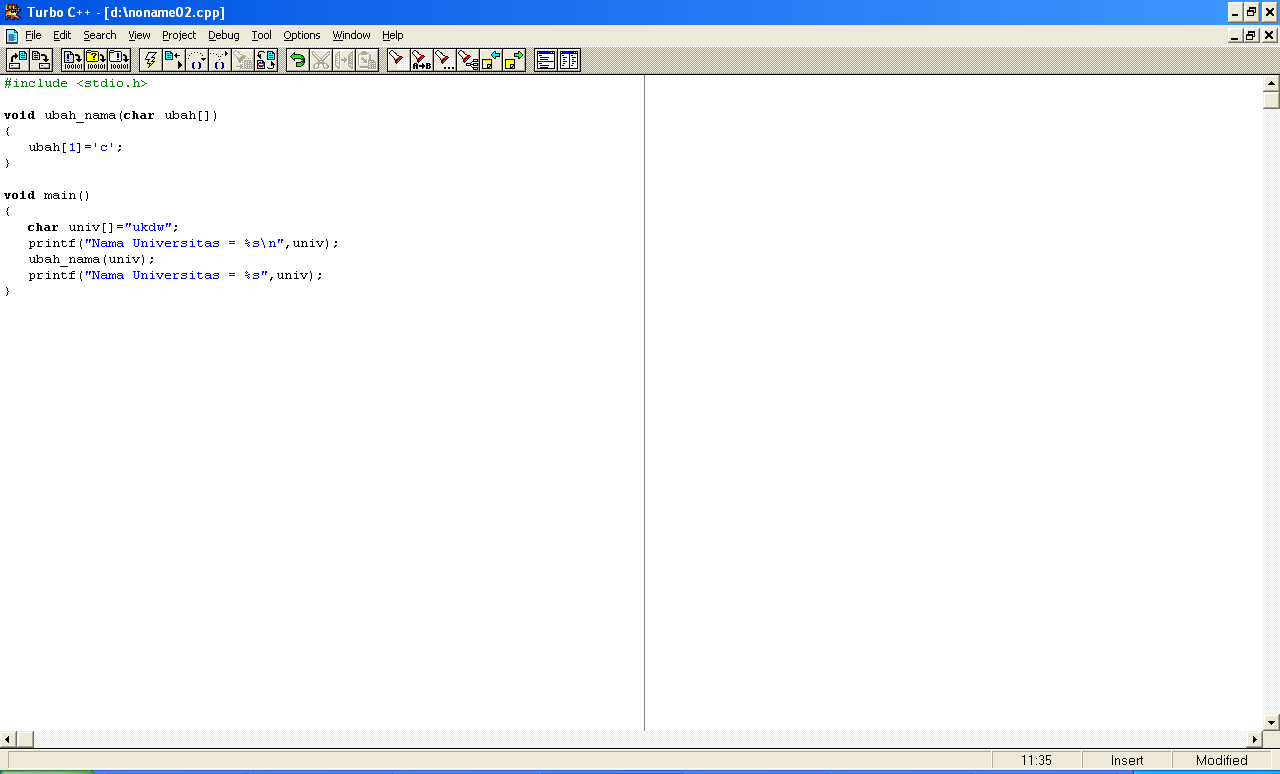
Pointer dapat juga digunakan sebagai parameter pada sebuah fungsi. Parameter ini akan bersifat by reference. Artinya yang dikirimkan sebagai parameter adalah alamat variabelnya, bukan nilainya. Jadi, perubahan nilai yang terjadi di dalam fungsi, akan mengubah juga nilai dari variabel yang alamatnya kita kirimkan. Agar lebih jelas penggunaannya, lihatlah contoh di bawah ini.



Mengirimkan alamat memory variabel a

Di sini dapat kita lihat bahwa pendeklarasian fungsi by reference, akan mendeklarasikan pointer sebagai parameternya. Kemudian, yang menjadi parameter aktualnya adalah alamat memory.

Selain itu, perlu kita ketahui bahwa pengiriman parameter yang berupa **array** pada sebuah fungsi adalah pengiriman by reference, sebab yang dikirim adalah alamat memory indeks pertama dari variabel array tersebut. Mari kita lihat contoh di bawah ini.



**Pengembangan:**

Buatlah sebuah program untuk mengecek apakah suatu kata palindrom atau bukan, tanpa memperhatikan spasi dan huruf besar/kecilnya. Program dibuat dengan menggunakan template struct sebagai berikut:

typedef struct{

char elemen[30];

int jml\_kata;

} Kata;

Kata kata;

Kata \*p\_kata=&kata;

Lanjutkanlah program berikut agar hasilnya sesuai dengan soal di atas:

#include <stdio.h>

#include <conio.h>

typedef struct{

char elemen[30];

int jml\_kata;

} Kata;

Kata kata;

Kata \*p\_kata=&kata;

int main(){

char kalimat[30];

p\_kata->jml\_kata=0;

char \*p = p\_kata->elemen;

printf("Masukkan kata : ");gets(kalimat);

fflush(stdin);

printf("Kalimat : %s\n",kalimat);

for(int i=0;i<kalimat[i];i++){

\*p=kalimat[i];

p\_kata->jml\_kata++;

p++;

}

p=p\_kata->elemen;

//tampilkan kembali kalimat tersebut

for(int i=0;i<=p\_kata->jml\_kata;i++){

printf("%c ",\*p);

p++;

}

//kembangkan….

getch();

}

Soal:

Buatlah program data KTP, dengan menggunakan pointer pada struct KTP sebagai berikut:

typedef struct

{

int tgl;

int bln;

int th;

}Tanggal;

typedef struct

{

char noID[5];

char nama[30];

char jenis\_kelamin; //’L’ atau ‘P’

Tanggal t;

}KTP;

typedef struct

{

KTP ktp;

int jml;

}Data\_KTP;

Data\_KTP data\_ktp;

Data\_KTP \*p;

Buatlah fungsi untuk:

1. Menambah data
2. Mencari data berdasarkan tahun kelahiran tertentu
3. Menampilkan data berdasarkan L dan P
4. Mengedit data

Semua pengaksesan dilakukan dengann menggunakan pointer.

Latihan

Kalian tentu telah mengenal fitur T9 yang ada di handphone. Misalnya saat kita mengaktifkan T9 input yang bahasa inggris, lalu kita ingin mengetik school, maka kita cukup menekan 724665, maka otomatis akan keluar kata “school”.

Nah, sekarang kita coba menerapkan T9 didalam program latihan kita dengan menggunakan struct dan pointer tentunya.

Spesifikasi Input : Program menampilkan tulisan “Masukkan sebuah kalimat : “ kemudian program menunggu user memasukkan sebuah kalimat (tidak lebih dari 100 karakter).

Spesifikasi output : Program akan mengubah kalimat yang dimasukkan user menjadi rangkaian angka dengan menggunakan tabel T9.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | # |
| Data | 0 | Default 1 | A B C 2 | D E F 3 | G H I 4 | J K L 5 | M N O 6 | P Q R S 7 | T U V 8 | W X Y Z 9 | Spasi |

Program ini tidak case sensitif, artinya APA dengan apa atau Apa adalah sama.

Contoh:

**Latihan 1**

Input:

Masukkan sebuah kalimat : Aku ingin pulang

Output:

258#46446#785264

**Latihan 2**

Input:

Masukkan sebuah kalimat : Struktur Data 85 dapat A

Output:

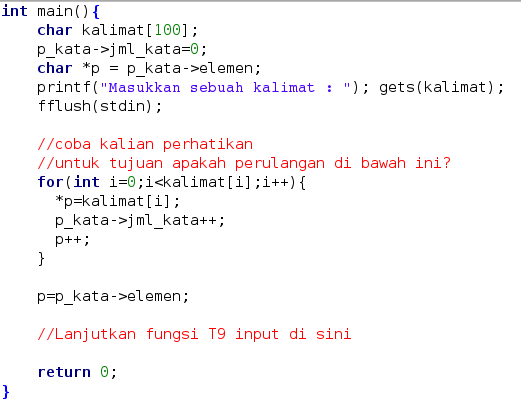
78785887#3282#85#32728#2

Langkah-langkahnya:

1. Include kan file stdio.h dan ctype.h (ctype nantinya digunakan untuk memanggil fungsi **toupper**(char c)).
2. Buat struct Kata berisi 2 elemen yaitu **elemen** bertipe array of character (elemennya 100) dan **jml\_kata** bertipe integer.
3. Buat sebuah variabel **kata** bertipe Kata dan sebuah variabel pointer **p\_kata** bertipe **Kata** dan langsung mengacu pada alamat dari variabel **kata**.



1. Dalam fungsi **main()** buat variabel **kalimat** bertipe array of character dengan 100 elemen.
2. Selanjutnya ikuti code di bawah ini.



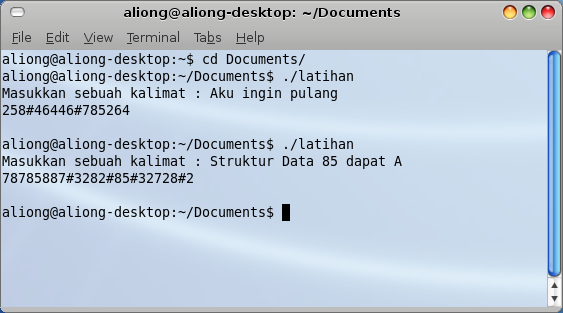
Lanjutkan program di atas agar sesuai dengan spesifikasi input dan output!

Hint :





Contoh Output Program :

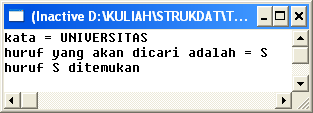


Latihan

1. Deskripsikan program pelaksanaan praktikum tersebut !
2. Buatlah sebuah program yang menampilkan kalimat terbalik dari suatu kalimat yang telah diinputkan maupun data yang telah ada dalam program. Menggunakan pointer dan beberapa fungsi.
3. Buatlah program sequential searching dengan menggunakan variabel pointer!
4. Buatlah program sequential searching dengan menggunakan variabel pointer!

**Take Home assignment**

1. Buatlah sebuah program yang dapat mencari sebuah huruf dalam sebuah kata. Kata dapat diinisialisasi terlebih dahulu. Gunakan pointer untuk menjadi counter.



1. Bila soal no.1 sudah terselesaikan, cobalah untuk membuat program yang sama tetapi dengan input kata yang dinamis.

## **DAFTAR PUSTAKA**

Deitel, Harvey M dan Paul J Deitel. 2005. *C++ How to Program, Fifth Edition.* New Jersey: Prentice Hall.

Deshpande, P. S dan O. G. Kakde. 2003*. C & Data Structures*. Massachusetts : Dreamtech Press.

Solichin, Achmad. 2003. *Pemograman Bahasa C dengan Turbo C*. IlmuKomputer.Com.

Materi Struktur Data TI UKDW oleh Antonius Rachmat C. S.Kom, M.Sc.

Materi Praktikum Struktur Data 2007 oleh TIM Asisten Praktikum Struktur Data 2007.