

Strategi Praktis Mendalami

ALGORITMA & PEMROGRAMAN

Kata Pengantar

Kursus Algorithm and Programming mempelajari tentang definisi algoritma, dasar pemrograman bahasa C, cara membuat program menggunakan bahasa C dan fitur-fitur dalam bahasa C. Kursus ini akan membahas secara rinci materi dasar konsep pemrograman menggunakan bahasa C.

Setelah menempuh kursus ini, peserta diharapkan memiliki pengetahuan tentang algoritma, mengenal dasar bahasa C sehingga mampu membuat program menggunakan bahasa pemrograman C. Dengan adanya bekal tersebut, peserta dapat melanjutkan ke tingkat pemrograman yang lebih kompleks dan mempermudah mempelajari bahasa pemrograman lainnya.

Daftar Isi

Kata Pengantar.....	1
Daftar Isi.....	2
1. Pengenalan Bahasa C.....	3
2. Variabel dan Tipe Data.....	5
3. Sintaks Input Output.....	6
4. Operasi Aritmatika	9
5. Selection	10
6. Repetition.....	13
7. Array.....	16
8. Pointer.....	17
9. Function	18
10. Built-in Function.....	22
11. Recursion	23
12. File	24
13. Sorting	26

1. Pengenalan Bahasa C

Bahasa pemrograman C merupakan bahasa pemrograman komputer yang memiliki berbagai modul/ fungsi untuk menjalankan perintah-perintah. Modul-modul tersebut disimpan di Standard Library. Selain modul bawaan, pengguna juga dapat membuat modul atau fungsi sendiri.

Cara memulai project bahasa C menggunakan Visual Studio :

1. Buka program Microsoft Visual C++ Express.
2. Buka menu File → New → Project.
3. Pilih Empty Project.
4. Isi nama project → OK
5. Klik kanan di Source File → Add → New Item.
6. Pilih C++ File (.cpp).
7. Isi nama file.
8. Memulai coding

```
#include <stdio.h>
void main ()
{
}
```

Penjelasan coding:

- `#include` → cara menggunakan library fungsi agar dapat menggunakan fungsi atau modul yang sudah tersedia.
- `<stdio.h>` → standard library untuk input output dan lainnya.
- `void main ()` → fungsi utama yang akan dijalankan pertama kali ketika eksekusi.

Beberapa aturan dalam pemrograman bahasa C yaitu:

- Case sensitive : huruf kapital dan huruf kecil dianggap beda.
- Setiap statement diakhiri oleh tanda ;
- Nama variabel tidak boleh sama dan tidak boleh menggunakan kata kunci/ sintaks.
- Cara memberikan catatan (comment) adalah menggunakan tanda // untuk 1 baris, dan tanda /* */ untuk sekumpulan code.

2. Variabel dan Tipe Data

Variabel yang dideklarasikan memerlukan memori untuk menampung nilai yang disimpan. Program akan memesan tempat di alamat tertentu sesuai memori yang dibutuhkan oleh variabel. Memori yang dibutuhkan oleh variabel ditentukan oleh jenis tipe data dari variabel tersebut. Dengan demikian, setiap variabel mempunyai nama, tipe dan nilai serta menunjuk ke suatu tempat dari memori komputer.

Contoh:

```
int x;
```

Penjelasan:

Program akan memesan tempat di memori untuk menampung nilai yang mempunyai tipe data integer. Untuk mengakses lokasi memori tersebut, kita menggunakan nama variabel 'x'.

Dalam bahasa C terdapat beberapa tipe data. Masing-masing tipe data digunakan untuk menyimpan jenis data berbeda dan mengkonsumsi jumlah memori yang berbeda juga. Beberapa tipe data dalam bahasa C adalah sebagai berikut :

No.	Tipe Data	Fungsi
1.	int	Bilangan bulat
2.	float	Bilangan desimal
3.	char	Karakter
4.	string	Kumpulan karakter

Contoh :

```
int angka1 = 5;
float angka2 = 2.6;
char huruf = 'A';
char nama[20] = "Anto";
```

3. Sintaks Input Output

Sintaks input digunakan agar program dapat meminta input kepada user. Sedangkan sintaks output digunakan untuk mencetak hasil di program. Fungsi-fungsi input output telah tersedia di standard library, yaitu file header “stdio.h”.

Beberapa fungsi untuk input antara lain:

No.	Sintaks	Fungsi
1.	scanf	Untuk membaca input sesuai format.
2.	getchar	Untuk membaca input 1 karakter
3.	gets	Untuk membaca string.

Format dalam scanf sesuai tipe data yaitu :

No.	Tipe Data	Format
1.	int	%d
2.	float	%f
3.	char	%c
4.	string	%s

Contoh penggunaan sintaks input :

1. `int angka;`
`scanf ("%d", &angka);`
`fflush (stdin);`
2. `char huruf = getchar();`
`fflush (stdin);`
3. `char nama[20];`
`gets(nama);`
`fflush (stdin);`

Penjelasan :

1. Code tersebut membuat sebuah variabel dengan nama “angka” dan tipe data integer. Setelah itu program akan meminta inputan user dan ditampung ke dalam variabel “angka”. `fflush(stdin)` digunakan untuk mengosongkan buffer stdin (standard input), digunakan setelah `scanf`.
2. Code tersebut membuat meminta input 1 karakter dan ditampung ke dalam variabel huruf.
3. Code tersebut membuat sebuah variabel dengan nama “nama” dan tipe data kumpulan char. Setelah itu program akan meminta inputan user dan ditampung ke dalam variabel “nama”.

Beberapa fungsi untuk output antara lain:

No.	Sintaks	Fungsi
1.	<code>printf</code>	Untuk print sesuai dengan format
2.	<code>putchar</code>	Untuk print 1 karakter.
3.	<code>puts</code>	Untuk print string.

Contoh penggunaan sintaks output :

1. `int angka = 5;`
`printf("%d", angka);`
2. `char huruf = 'A';`
`putchar(huruf);`
3. `char nama[20] = "Anto";`
`puts(nama);`

Penjelasan :

1. Code tersebut membuat sebuah variabel dengan nama “angka”, tipe data integer dan berisi angka 5. Setelah itu program akan mencetak isi dari variabel “angka” yaitu 5.

2. Code tersebut membuat sebuah variabel dengan nama “huruf”, tipe data char dan berisi karakter A. Setelah itu program akan mencetak isi dari variabel “huruf” yaitu A.
3. Code tersebut membuat sebuah variabel dengan nama “nama”, tipe data kumpulan char dan berisi kata “Anto”. Setelah itu program akan mencetak isi dari variabel “nama” yaitu Anto.

4. Operasi Aritmatika

Operasi aritmatika yang akan dibahas meliputi penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Simbol dalam operasi aritmatika meliputi:

No.	Simbol	Fungsi
1.	+	Penjumlahan
2.	-	Pengurangan
3.	*	Perkalian
4.	/	Pembagian
5.	%	Modulus (siswa pembagian)
6.	++	Increment (ditambah 1)
7.	--	Decrement (dikurang 1)

Contoh:

```
int angka1 = 9;
int angka2 = 2;
int hasil;
hasil = angka1 + angka2;    11
hasil = angka1 - angka2;    7
hasil = angka1 * angka2;    18
hasil = angka1 / angka2;    4
hasil = angka1 % angka2;    1
```

Pada increment dan decrement terdapat kondisi post dan pre. Untuk post, eksekusi increment/decrement dijalankan di akhir. Sedangkan pre akan menjalankan increment/decrement terlebih dahulu.

Contoh:

```
int angka = 5;                int angka=5;
printf("%d", angka++);        printf("%d", ++angka);
Hasil: 5                      Hasil: 6
```

5. Selection

Selection adalah konsep pemrograman untuk membuat struktur kendali pemilihan berdasarkan kondisi. Dalam proses seleksi, kondisi yang diuji dapat terdiri dari beberapa kondisi, sehingga membutuhkan operator logical. Logical operator yang sering digunakan yaitu && (AND) dan || (OR). Berikut tabel logical operator, yaitu :

Kondisi 1	Kondisi 2	AND	OR
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Dalam bahasa C terdapat 3 jenis selection yaitu :

1. IF ELSE

IF ELSE merupakan struktur kontrol yang menjalankan statement setelah memeriksa suatu kondisi. Jika kondisi benar, maka statement dijalankan. Jika tidak benar, maka statement lain dijalankan.

Struktur code:

```
if ( <<kondisi1>> )
{
    <<statement1>>;
}
else if ( <<kondisi2>> )
{
    <<statement2>>;
}
else
{
    <<statement3>>;
}
```

```
}
```

Penjelasan:

- Jika kondisi1 terpenuhi, maka statement1 jalan.
- Jika kondisi2 terpenuhi, maka statement2 jalan.
- Selain itu, statement3 jalan.

2. SWITCH CASE

SWITCH CASE merupakan struktur kontrol yang multiple-selection.

Struktur code:

```
switch ( <<variabel>> )
{
    case << nilai >> :
        << statement1 >>;
        break;
    case << nilai2 >> :
        << statement2 >>;
        break;
    default :
        << statement3 >>;
}
```

Penjelasan:

- Jika isi variabel sama dengan nilai1, maka statement1 jalan.
- Jika isi variabel sama dengan nilai2, maka statement2 jalan.
- Selain itu jalan statement3.

3. TERNARY

TERNARY merupakan selection yang sama seperti IF ELSE, akan tetapi memiliki code yang lebih singkat.

Struktur code:

```
<<variabel>> = ( <<kondisi>> ) ? <<statement1>> :  
<<statement2>>;
```

Penjelasan:

Jika kondisi terpenuhi, maka statement1 akan dijalankan dan ditampung ke variabel. Jika kondisi tidak terpenuhi, maka statement2 akan jalan dan ditampung ke variabel.

6. Repetition

Repetition merupakan konsep pemrograman untuk membuat struktur kendali perulangan jika memenuhi suatu kondisi. Perulangan dalam bahasa C dapat menggunakan 3 cara, yaitu :

1. FOR

Perulangan FOR digunakan jika jumlah perulangan sudah diketahui.

Struktur code :

```
for ( <<ekspresi1>>; <<ekspresi2>>; <<ekspresi3>> )
{
    <<statement>>;
}
```

Penjelasan:

1. Ekspresi1 dijalankan.
2. Ekspresi2 diperiksa apakah true atau false. Jika true maka statement dijalankan, jika false maka statement tidak dijalankan.
3. Setelah statement dijalankan, maka ekspresi3 dijalankan. Kemudian kembali ke poin nomor 2.

Contoh:

```
for (int i=0; i<5; i++)
{
    printf("%d\n", i);
}
```

Hasil:

```
0
1
2
3
4
```

2. WHILE

WHILE digunakan jika jumlah perulangan belum diketahui dan disesuaikan terhadap kondisi. Perulangan WHILE akan tidak dijalankan sama sekali jika kondisi tidak terpenuhi sejak awal.

Struktur code:

```
while ( <<kondisi>> )  
{  
    <<statement>>;  
}
```

Penjelasan:

1. Kondisi akan dicek apakah true atau false. Jika true maka statement dijalankan, jika false maka statement tidak dijalankan.
2. Setelah statement dijalankan, kembali ke poin 1.
3. Jika kondisi selalu bernilai true, maka akan terjadi infinite loop atau looping forever.

Contoh:

```
int angka = 1;  
while (angka<4)  
{  
    printf("%d\n", angka);  
    angka++;  
}
```

Hasil:

```
1  
2  
3
```

3. DO WHILE

DO WHILE digunakan jika jumlah perulangan belum diketahui dan disesuaikan terhadap kondisi. Perulangan DO WHILE akan menjalankan statement terlebih dahulu sebelum memeriksa kondisi.

Struktur code:

```
do
{
    <<statement>>;
}while ( <<kondisi>> );
```

Penjelasan:

1. Statement akan dijalankan terlebih dahulu.
2. Kemudian dilanjutkan cek kondisi.
3. Jika kondisi true, maka kembali ke poin 1. Jika false maka DO WHILE selesai.

Contoh:

```
int angka = 1;
do
{
    printf("%d\n", angka);
    angka++;
}while (angka<5);
```

Hasil:

```
1
2
3
4
```


7. Array

Array adalah sekumpulan data dengan tipe data dan nama variabel yang sama. Elemen dalam array diakses melalui indeks. Indeks dalam array dimulai dari indeks ke-0.

Array dapat dibuat dalam 1 dimensi dan lebih dari 1 dimensi. Contoh kasus penggunaan array adalah penyimpanan nilai 30 mahasiswa. Jika menggunakan array, kita dapat menampung nilai 30 mahasiswa tersebut dalam 1 variabel. Array juga diterapkan dalam pembentukan string yaitu menggunakan array of char.

Contoh :

```
int nilai [30];  
nilai[0] = 90;  
nilai[1] = 100;  
int angka[2][3] = {{1,2,3}, {4,5,6}};  
angka [1][2]=100;  
char nama[50] = "Anto";
```

Penjelasan:

- Variabel nilai dapat tersedia 30 elemen. Namun hanya terisi elemen ke 0 dan 1 dengan nilai 90 dan 100.
- Variabel angka terdiri dari 2 dimensi, yaitu 2 baris dan 3 kolom. Baris pertama berisi angka 1, 2, 3. Sedangkan baris kedua berisi angka 4, 5, 6.
- Elemen indeks ke [1][2] yaitu angka 6 akan diganti dengan angka 100.
- Variabel nama bertipe data array of char (string) dan menampung kata "Anto".

8. Pointer

Ketika membuat variabel, maka variabel tersebut akan dialokasikan dalam memori komputer. Di dalam memori komputer, setiap variabel memiliki alamat.

Pointer adalah tipe data yang berisi alamat variabel lain. Pointer ditunjukkan dengan simbol *, sedangkan alamat variabel menggunakan simbol &.

Contoh:

Nama variabel	angka	ptr
Isi variabel	100	123
Alamat	123	789

Code:

```
int angka = 100;  
int *ptr;  
ptr = &angka;  
*ptr = 30;
```

Penjelasan:

- Variabel angka berisi nilai 100 dengan alamat memori 123.
- Variabel ptr berisi alamat variabel angka (123).
- Alamat dari variabel ptr adalah 789.
- Isi variabel angka dapat diubah dengan menggunakan *ptr.

9. Function

Function adalah modul-modul dari code. Function memiliki return type, parameter dan isi. Return type merupakan tipe data yang akan dikembalikan dari function tersebut. Parameter digunakan untuk passing data antar function.

Parameter terdiri dari 2 jenis yaitu parameter formal dan aktual. Parameter formal adalah parameter pada saat pembuatan function. Parameter aktual adalah parameter pada saat pemanggilan function.

Passing parameter dalam function terdiri dari 3 jenis, yaitu:

1. Passing by value.

Passing by value yaitu tipe passing parameter yang hanya mengirimkan nilai pada parameter aktual tanpa mengubah isi dari data yang terdapat pada parameter aktual.

Contoh:

```
void print (int a)
{
    a = 10;
    printf ("%d\n", a);
}

void main ()
{
    int a = 5;
    print(a);
    printf ("%d\n", a);
}
```

Hasil:

10

5

Penjelasan:

- Pada saat printf di dalam function print, hasil yang dicetak yaitu angka 10, karena a telah diganti menjadi angka 10 dalam function print.
- Pada saat printf di dalam function main, hasil yang dicetak yaitu angka 5. Hal ini dikarenakan passing by value tidak dapat mengubah nilai variabel melalui parameter aktual.

2. Passing by pointer.

Passing by pointer adalah mengirimkan alamat dari suatu variabel, parameter aktual menggunakan simbol "&" dan parameter formal menggunakan simbol pointer *. Passing by pointer dapat mengubah nilai dari variabel.

Contoh:

```
void print (int *a)
{
    *a = 10;
    printf ("%d\n", *a);
}

void main ()
{
    int a = 5;
    print(&a);
    printf ("%d\n", a);
}
```

Hasil:

10

10

Penjelasan:

Hasil cetak a di function main dan print sama-sama memberikan angka 10. Hal ini dikarenakan passing by pointer, sehingga penggantian nilai variabel a di function print juga akan mengganti nilai variabel a yang terdapat di dalam function main.

3. Passing by reference.

Passing by value adalah mengirimkan variabel melalui parameter aktual, alamat dari variabel akan diakses di parameter formal dengan penggunaan simbol "&".

Contoh:

```
void print (int &a)
{
    a = 10;
    printf ("%d\n", a);
}
```

```
void main ()
{
    int a = 5;
    print(a);
    printf ("%d\n", a);
}
```

Hasil:

10

10

Penjelasan:

Hasil cetak a di function main dan print sama-sama memberikan angka 10. Hal ini dikarenakan passing by reference, sehingga penggantian nilai variabel a di function print juga akan mengganti nilai variabel a yang terdapat di dalam function main.

10. Built-in Function

Built-in function adalah function yang telah disediakan oleh bahasa pemrograman. Dalam bahasa C, function dipanggil dengan melakukan include header file.

Beberapa contoh built-in function yang sering digunakan seperti:

Built-in function	Include	Fungsi
printf	stdio.h	Mencetak
scanf	stdio.h	Menerima input
strlen	string.h	Panjang string
strcpy	string.h	Copy string
strcmp	string.h	Compare string
strcat	string.h	Menambahkan string ke string lain
rand	stdlib.h	Random angka
pow	math.h	Perhitungan pangkat
isdigit	ctype.h	Memeriksa apakah angka
isalpha	ctype.h	Memeriksa apakah alfabet

11. Recursion

Recursion adalah function yang memanggil dirinya sendiri. Recursion dapat digunakan untuk menyelesaikan masalah seperti bilangan fibonacci dan faktorial.

Contoh:

```
int faktorial (int angka)
{
    if (angka==0 || angka==1)
    {
        return 1;
    }
    else
    {
        Return n*faktorial(n-1);
    }
}

void main()
{
    int angka = 5;
    printf("%d\n", faktorial(angka));
}
```

Penjelasan:

- Function faktorial merupakan function yang recursive.
- Angka akan dikurangi 1 terus menerus ketika pemanggilan ulang function faktorial.
- Jika angka mencapai 1 atau 0, maka function faktorial akan berakhir dan mengembalikan nilai ke function main.

12. File

Operasi file yang akan dibahas yaitu baca dan tulis file. File tersebut akan disimpan dalam bentuk teks. File tersebut akan dibaca dan ditulis melalui program C yang dikembangkan.

Function-function dalam operasi file yang sering digunakan yaitu :

Function	Kegunaan
fopen	Mengaktifkan / membuka file
fclose	Menutup file
fscanf	Membaca isi dari file
fprintf	Mencetak ke file
feof	Mendeteksi akhir dari file

Jenis operasi file :

Operasi	Kegunaan
r	read file.
w	write file, jika file ada isi maka akan ditimpa.
a	append file, jika file ada isi maka akan dilanjutkan di bawah.

Baca file :

```
FILE *read = fopen ("data.txt", "r");
char nama[50];
while (!feof(read))
{
    fscanf(read, "%[^\\n]\\n", nama);
    printf("%s\\n", nama);
}
```

Penjelasan:

- FILE akan melakukan open terhadap data.txt di folder project.
- Ketika tidak terdeteksi akhir file, maka akan dilakukan pembacaan file per baris dan mencetaknya.

Tulis file :

```
FILE *write = fopen ("data.txt", "w");  
fprintf(write, "Anto");  
fclose(write);
```

Penjelasan:

- FILE akan melakukan open terhadap data.txt di folder project.
- Program akan mencetak kata "Anto" ke dalam file tersebut.
- Karena jenis operasinya adalah w (write), maka data.txt akan ditimpa dengan kata "Anto".
- Jika menggunakan jenis operasi a (append), maka kata "Anto" akan ditambahkan di baris terakhir.

13. Sorting

Sorting adalah algoritma untuk menempatkan data sesuai urutan, baik ascending (kecil ke besar) maupun descending (besar ke kecil). Salah satu tujuan sorting yaitu mempercepat pencarian data.

Algoritma sorting terdiri dari beberapa jenis, yaitu :

- Buble sorting
- Selection sorting
- Insertion sorting
- Merge sorting
- Quick sorting

Contoh code (Bubble sort ascending) :

```
void main ()
{
    int angka[5] = {5,2,9,10,1};

    for(int i=0; i<5; i++)
    {
        for (int j=i; j<5; j++)
        {
            if (angka[i] > angka[j])
            {
                int temp = angka[i];
                angka[i] = angka[j];
                angka[j] = temp;
            }
        }
    }

    for (int i=0; i<5; i++)
    {
```

```
        printf ("%d\n", angka[i]);  
    }  
}
```

Penjelasan:

- Variabel angka terdiri dari 5 elemen integer yang tidak berurutan.
- Looping dimulai dari variabel i 0-4 untuk dijadikan elemen.
- Setiap looping i, variabel j juga dilooping sejumlah i sampai 4.
- Angka ke-i akan dibandingkan dengan angka ke-j.
- Jika angka ke-i lebih besar daripada angka ke-j, maka akan di-swap angka ke-i dan ke-j.
- Setelah looping i dan j selesai, maka data telah disorting secara ascending. Jika ingin menjadi descending, cukup mengganti simbol > yang terdapat di dalam kondisi IF menjadi <.
- Isi variabel angka dilooping untuk mencetak hasil dari indeks ke-0 hingga indeks ke-4.

Soal Latihan Pertemuan 1a

Buatlah sebuah program untuk input NIM, nama lengkap, umur, dan target IPK. Gunakan tipe data yang tepat untuk setiap variabel yang digunakan untuk menampung input, kemudian hasil input ditampilkan kembali.

```
Welcome to Course-Net
-----
Enter your NIM       : 2201464857
Enter your full name : Bob Smith
Enter your age       : 18
Enter your GPA target : 4

Below this, they are your data :

NIM       : 2201464857
Full Name : Bob Smith
Age       : 18
GPA Target : 4.00
```



Course-Net

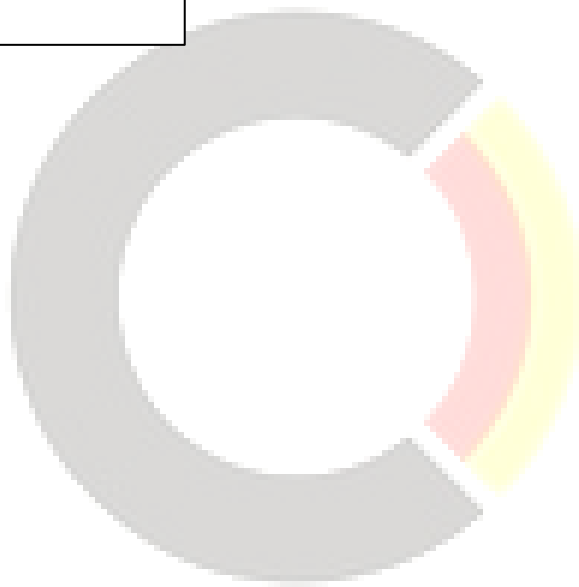
Empowering IT People

Soal Latihan Pertemuan 1b

Buatlah sebuah program untuk menginput kata. Setelah input, program akan menampilkan code dari aplikasi beserta inputan user.

```
Please Input : Course-Net

#include <stdio.h>
int main()
{
    char kata[] = "Course-Net";
    printf("%s\n", kata);
    getchar();
    return 0;
}
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 2a

Buatlah sebuah program untuk menghitung luas segitiga. Program akan meminta input alas dan tinggi. Setelah dinput, program akan memberikan output rangkuman soal dan jawaban luas segitiga.

```
Menentukan Luas Segitiga
-----
Input Alas      : 14
Input Tinggi    : 20

Diketahui :
Alas = 14
Tinggi = 20

Ditanya : Luas Segitiga?

Jawaban :
Luas Segitiga = Alas * Tinggi / 2
Luas Segitiga = 14 * 20 / 2
Luas Segitiga = 280 * 2
Luas Segitiga = 140.00
```

Soal Latihan Pertemuan 2b

Buatlah sebuah program untuk menghitung nilai akhir mata kuliah. Pada saat program dimulai, program akan menampilkan pembagian persentase nilai TM, UTS, dan UAS. Setelah itu user akan diminta untuk meng-input nilai TM, UTS, dan UAS. Setelah berhasil input, program akan menampilkan hasil kalkulasi nilai akhir.

```
Kalkulator Nilai Akhir
-----
Bobot :
Tugas Mandiri(TM)           : %20
Ujian Tengah Semester(UTS)   : %30
Ujian Akhir Semester(UAS)    : %50

Nilai Akhir = TM + UTS + UAS

Masukkan Nilai TM : 90
Masukkan Nilai UTS : 80
Masukkan Nilai UAS : 88

Nilai Akhir = 86.00
```


Soal Latihan Pertemuan 2c

Buatlah sebuah program untuk mengkonversi jam, menit, dan detik menjadi detik. Pada awal program akan meminta input jam, menit, dan detik. Setelah diinput, program akan kalkulasi dan menampilkan dalam satuan detik.

```
Time to Seconds Converter
-----
Input Hour(s)           : 2
Input Minutes(s)        : 42
Input Second(s)         : 21
Input Processed, Press Enter...

2 Hour(s) 42 Minute(s) 21 Second(s) converted to 9741 second(s)
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 2d

Buatlah sebuah program untuk mengkonversi detik menjadi jam, menit, dan detik. Pada awal program akan meminta input detik. Setelah diinput, program akan kalkulasi dan menampilkan dalam satuan jam, menit, dan detik.

```
Seconds to Time Converter
-----
Input Second(s) : 7329
Input Processed, Press Enter...
7329 second(s) = 2 Hour(s) 2 Minute(s) 9 Second(s)
```



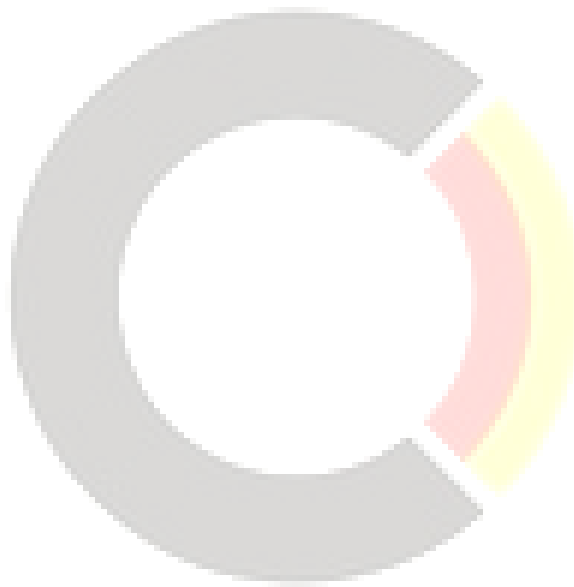
Course-Net
Empowering IT People

Soal Latihan Pertemuan 3a

Buatlah sebuah program untuk menentukan bilangan ganjil atau genap. Saat program dimulai, program akan meminta input sebuah bilangan dan program akan memberikan hasil bilangan tersebut bilangan genap atau ganjil.

```
Odd or Even Number
-----
Input A Number : 64
64 is Even Number
```

```
Odd or Even Number
-----
Input A Number : 11
11 is Odd Number
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 3b

Buatlah sebuah program untuk menentukan grade nilai huruf dari input nilai angka. Pembagian grade nilai sebagai berikut:

- Grade A : 90 – 100
- Grade A- : 85 – 89
- Grade B+ : 80 – 84
- Grade B : 75 – 79
- Grade B- : 70 – 74
- Grade C : 65 – 69
- Grade D : 50 – 64
- Grade E : 0 – 49

Score to Grade

Input Score : <input type="text" value="88"/>
Your Grade : A-



Course-Net
Empowering IT People

Soal Latihan Pertemuan 3c

Buatlah sebuah program untuk menentukan jenis huruf, apakah vokal atau konsonan ataupun inputan bukan huruf.

```
Consonant or Vocal Alphabet
-----
Input an Alphabet : 4
4 is not an Alphabet
```

```
Consonant or Vocal Alphabet
-----
Input an Alphabet : B
B is Consonant Alphabet
```

```
Consonant or Vocal Alphabet
-----
Input an Alphabet : e
e is Vocal Alphabet
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 4a

Buatlah sebuah program untuk menerima input kata, kemudian menampilkan kembali dengan ketentuan:

- Huruf di urutan ganjil ditampilkan dalam huruf kapital.
- Huruf di urutan genap ditampilkan dalam huruf kecil.

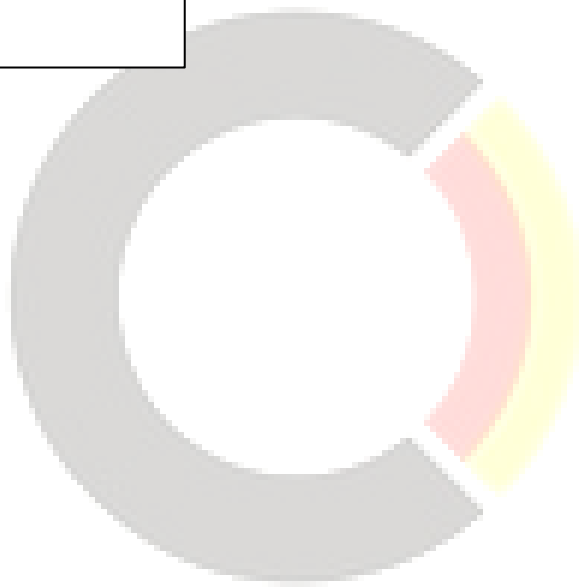
```
Odd Upper and Even Lower
-----
Input Words : coursenet indonesia
Your words = CoUrSeNeT InDoNeSiA
```



Soal Latihan Pertemuan 4b

Buatlah sebuah program untuk menerima input sebuah karakter dan angka. Program akan mencetak segitiga siku-siku kiri dengan menggunakan karakter yang diinput setinggi angka yang diinput.

```
Bottom Left Elbow Triangle
-----
Input a Character and Height : C 7
C
CC
CCC
CCCC
CCCCC
CCCCCC
CCCCCCC
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 4c

Buatlah sebuah program untuk validasi inputan angka dengan kisaran antara 1 – 100. Jika user meng-input angka di luar kisaran tersebut ataupun bukan angka, program akan meminta kembali inputan hingga benar. Setelah input benar, program akan mencetak kembali angka yang di-input user dengan benar.

```
1 - 100 Number
-----
Input a Number[1..100]: 103
Input a Number[1..100]: Course
Input a Number[1..100]: 86
Your Number : 86
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 4d

Buatlah sebuah program untuk menghitung jumlah vokal dan konsonan dari kata yang diinput user. Pada awal program, user akan diminta untuk meng-input kata. Setelah diinput, program akan menampilkan jumlah huruf vokal maupun konsonan dari kata yang diinput.

```
Consonant or Vocal Total
-----
Input Word(s) : CourseNet Indonesia
Total :
Consonant = 9
Vocal = 9
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 5a

Buatlah sebuah program untuk menjumlahkan angka-angka. Pada saat program dijalankan, program akan meminta input sebuah angka (misalkan angka x). Program akan meminta input angka sebanyak x kali. Setelah user menginput semua angka, program akan menjumlahkan angka-angka urutan ganjil dan genap kemudian akan ditampilkan.

```
Odd and Even Total
-----
Enter how many numbers : 7
Input num #1 : 75
Input num #2 : 4
Input num #3 : 28
Input num #4 : 17
Input num #5 : 42
Input num #6 : 88
Input num #7 : 10

Odd Numbers Total = 155
Even Numbers Total = 109
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 5b

Buatlah sebuah program untuk menjumlahkan matrix 3x3. Pada awal program, program akan meminta user untuk input angka-angka untuk matrix A dan B. Setelah itu program akan melakukan kalkulasi dan menampilkan hasil dari penjumlahan matrix A dan B.

```
3x3 Matrix Addition
-----

Matrix A
Input row 1 column 1 : 2
Input row 1 column 2 : 1
Input row 1 column 3 : 3
Input row 2 column 1 : 5
Input row 2 column 2 : 2
Input row 2 column 3 : 7
Input row 3 column 1 : 6
Input row 3 column 2 : 8
Input row 3 column 3 : 3

Matrix B
Input row 1 column 1 : 1
Input row 1 column 2 : 3
Input row 1 column 3 : 5
Input row 2 column 1 : 3
Input row 2 column 2 : 3
Input row 2 column 3 : 7
Input row 3 column 1 : 2
Input row 3 column 2 : 8
Input row 3 column 3 : 3

Matrix A
 2  1  3
 5  2  7
 6  8  3

Matrix B
 1  3  5
 3  3  7
 2  8  3

Matrix [A+B] =   3  4  8
                  8  5 14
                  8 16  6
```

Soal Latihan Pertemuan 7a

Buatlah sebuah program swap angka. User akan diminta untuk menginput angka X dan Y. Setelah diinput, program akan melakukan swap pada kedua variabel tersebut dan menampilkannya kembali.

```
Swap Number
-----
Enter X : 78
Enter Y : 56

Before Swap
X : 78
Y : 56

Swap Process...

After Swap
X : 56
Y : 78
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 7b

Buatlah sebuah aplikasi kasir sederhana. Aplikasi tersebut akan menampilkan list barang yang dijual dan menu untuk jual, beli, dan exit aplikasi.

Pada awal, program akan menampilkan list barang yang dijual beserta menu untuk dipilih user. Menu hanya bisa dipilih antara angka 1 – 3.

No.	Name	Quantity	Price,-
1.	Pulpen	56	Rp. 15000,-
2.	Mouse	34	Rp. 90000,-
3.	Keyboard	32	Rp. 100000,-

Total Income : Rp. 0

Course-Net Mart

1. Add Item
2. Sell Item
3. Exit

Enter Your Choice[1..3] :

Jika belum ada barang, maka aplikasi akan menampilkan no data.

No Data

Total Income : Rp. 0

Course-Net Mart

1. Add Item
2. Sell Item
3. Exit

Enter Your Choice[1..3] :

Ketika user meng-input menu 1, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Name : 5 – 15 karakter.
- Quantity : 1 – 100
- Price : 10.000 – 100.000

Jika input user salah, maka aplikasi akan meminta input kembali.

Enter Your Choice[1..3] : **1**

Input Item's name [5..15] : **Flashdisk**

Input Item's quantity [1..100] : **54**

Input Item's price [10000..100000] : **99**

Input Item's price [10000..100000] : **99000**

Flashdisk(s) have been succesfully added

Ketika user meng-input menu 2, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Item yang ingin dijual : input antara 1 sampai jumlah jenis barang.
- Quantity : input antara 1 sampai jumlah quantity barang.

Setelah diinput maka aplikasi akan menghitung jumlah nominal yang harus dibayar user.

```
Enter Your Choice[1..3] : 2
Which item do you want to sell[1..4] : 3
Input Quantity[1..32] : 20

20 Keyboard(s) have been sold successfully for Rp. 2000000,-
```

Jika user memilih menu 2 dan belum ada list barang, maka aplikasi akan menampilkan “Nothing to Update”.

```
No Data

Total Income : Rp. 0

Course-Net Mart
-----
1. Add Item
2. Sell Item
3. Exit

Enter Your Choice[1..3] : 2
Nothing to Update
```



Course-Net

Empowering IT People

Soal Latihan Pertemuan 8a

Buatlah sebuah program untuk validasi input user agar hanya berisi huruf dan spasi. Jika input berisi karakter lain, maka program akan meminta user untuk input kembali.

```
Alphabets and Space Validation
-----
Input Words : 1nd0n3Si4
Input Words : Indonesia
Input Success
```



Course-Net
Empowering IT People

Soal Latihan Pertemuan 8b

Buatlah sebuah program untuk memeriksa kata yang diinput user merupakan palindrome atau tidak. Palindrome adalah kata yang pembacaan dari depan ataupun belakang akan menghasilkan kata yang sama.

```
Palindrome Checker
-----
Input Words[4..20] : kodok
kodok is palindrome
```

```
Palindrome Checker
-----
Input Words[4..20] : coursenet
coursenet is not palindrome
```



Soal Latihan Pertemuan 9a

Bilangan fibonacci adalah barisan bilangan didefinisikan secara rekursif dengan dimulai angka 0 dan 1, bilangan berikutnya diperoleh dari menjumlahkan kedua bilangan yang berurutan sebelumnya. Buatlah sebuah program untuk custom fibonacci. Custom yang dilakukan yaitu barisan bilangan akan dimulai dengan angka 0 dan n, n adalah bilangan yang diinput oleh user.

Setelah itu, user akan diminta untuk menginput jumlah angka fibonacci yang akan ditampilkan. Program akan menampilkan barisan sesuai jumlah bilangan yang diisi dan user dapat meng-input kembali untuk index fibonacci yang ingin ditampilkan.

```
Custom Fibonacci
-----
Enter start[>= 1] : 1

How many Fibonacci you want to show[2..10] : 10

Custom Fibonacci starts by 1 :
0 1 1 2 3 5 8 13 21 34

Check fibonacci index[1..10] : 8
Fibonacci index-8 : 13
```

```
Custom Fibonacci
-----
Enter start[>= 1] : 7

How many Fibonacci you want to show[2..10] : 8

Custom Fibonacci starts by 7 :
0 7 7 14 21 35 56 91

Check fibonacci index[1..8] : 7
Fibonacci index-7 : 56
```

Soal Latihan Pertemuan 9b

Buatlah sebuah aplikasi untuk toko Course-Net Burger. Aplikasi tersebut akan menampilkan list burger yang dijual, total pemasukan dan menu untuk jual, beli, update, dan exit aplikasi. List burger disimpan dalam struct.

Menu Home

Pada awal, program akan menampilkan list barang yang dijual, total pemasukan beserta menu untuk dipilih user. Menu hanya bisa dipilih antara angka 1 – 4. Total income diperoleh dari akumulasi penjualan.

No.	Name	Quantity	Price
1.	Cheese Burger	15	\$ 4.30
2.	Chicken Burger	20	\$ 4.70
3.	Beef Burger	23	\$ 5.40
4.	Salad Burger	13	\$ 4.00

Total Income : \$ 0.00

Course-Net Burger

1. Add Burger
2. Sell Burger
3. Update Data Burger
4. Exit

Enter Your Choice[1..4] :

Jika belum ada barang, maka aplikasi akan menampilkan no data.

No Data

Total Income : \$ 0.00

Course-Net Burger

1. Add Burger
2. Sell Burger
3. Update Data Burger
4. Exit

Enter Your Choice[1..4] :

Menu 1

Ketika user meng-input menu 1, maka aplikasi akan menampilkan list burger dan meminta input dan validasi sebagai berikut:

- Name : 5 – 15 karakter.
- Quantity : 1 – 100
- Price : 4 – 9

Jika input user salah, maka aplikasi akan meminta input kembali.

No.	Name	Quantity	Price
1.	Chicken Burger	25	\$ 7.50

Add Burger

Input Burger's name[5..15] : Beef Burger

Input Burger's quantity[1..100] : 51

Input Burger's price \$[4..9] : 8.9

Beef Burger(s) have been succesfully added

Menu 2

Ketika user meng-input menu 2, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Burger yang ingin dijual: input antara 1 sampai jumlah jenis burger.
- Quantity : input antara 1 sampai jumlah quantity burger.
- Add extra : Y atau N
- Jika Y maka akan ada pilihan :
 - Bacon Gravy : \$2.3
 - Chicken Strips : \$2.5
 - Liquid Cheese : \$1.7
 - Mozza Sticks : \$2.0
 - Onion Rings : 1.9

Setelah diinput maka aplikasi akan menghitung jumlah nominal yang harus dibayar user.

Jika burger habis terjual (quantity 0), maka burger tersebut akan dihapus dari list burger.

No.	Name	Quantity	Price
1.	Cheese Burger	15	\$ 4.30
2.	Chicken Burger	20	\$ 4.70
3.	Beef Burger	23	\$ 5.40
4.	Salad Burger	13	\$ 4.00

Sell Burger

Which burger do you want to sell[1..4] : 7

Which burger do you want to sell[1..4] : 2

Input Quantity[1..20] : 3

Add extra(s)[Y | N] : Y

Choose extra(s)[Bacon Gravy | Chicken Strips | Liquid Cheese | Mozza Sticks | Onion Rings] : chicken

Choose extra(s)[Bacon Gravy | Chicken Strips | Liquid Cheese | Mozza Sticks | Onion Rings] : chicken strips

Choose extra(s)[Bacon Gravy | Chicken Strips | Liquid Cheese | Mozza Sticks | Onion Rings] : Chicken Strips

You have been charges \$2.5 each burger for extra Chicken Strips

3 Chicken Burger(s) with extra Chicken Strips have been sold successfully for \$ 21.60

Jika user memilih menu 2 dan belum ada list burger, maka aplikasi akan menampilkan "Nothing to sell".

Menu 3

Ketika user meng-input menu 3, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Burger yang ingin di-update : input antara 1 sampai jumlah jenis burger.
- Nama burger : 5 – 15 karakter.
- Quantity : 1 – 100.
- Price : 4 – 9.

Setelah diinput maka aplikasi akan update list burger.

No.	Name	Quantity	Price
1.	Cheese Burger	15	\$ 4.30
2.	Chicken Burger	20	\$ 4.70
3.	Beef Burger	23	\$ 5.40
4.	Salad Burger	13	\$ 4.00

Update Burger

Which burger do you want to update[1..4] : 3

Input Burger's name[5..15] : Thai Beef Burger

Input Burger's name[5..15] : Ori Beef Burger

Input Burger's quantity[1..100] : 14

Input Burger's price \$[4..9] : 5.8

Beef Burger(s) have been edited

No.	Name	Quantity	Price
1.	Cheese Burger	15	\$ 4.30
2.	Chicken Burger	20	\$ 4.70
3.	Ori Beef Burger	14	\$ 5.80
4.	Salad Burger	13	\$ 4.00

Jika user memilih menu 3 dan belum ada list burger, maka aplikasi akan menampilkan “Nothing to update”.



Soal Latihan Pertemuan 10

Buatlah sebuah aplikasi untuk Course-Net Book Store. Aplikasi tersebut akan digunakan untuk mengelola buku yang terdapat di Course-Net Book Store seperti untuk menambah, update, ataupun hapus buku. Data-data buku tersebut akan diambil dan disimpan ke dalam file .txt ketika ada perubahan data.

Berikut list buku yang telah tersedia saat ini dan dapat disimpan ke dalam file .txt.

```
Algorithm and Programming#Clifford Stein#8#500000
Calculus#Morris Kline#13#200000
Linear Algebra#Steven Roman#4#300000
```

Menu Home

Pada awal, program akan membaca file .txt, menampilkan list buku, dan menu untuk dipilih user. Menu hanya bisa dipilih antara angka 1 – 4.

```
Course-Net Book Store
No  Title                                     Author                Quantity  Price
1   Algorithm and Programming               Clifford Stein         8         500000
2   Calculus                               Morris Kline          13        200000
3   Linear Algebra                         Steven Roman          4         300000

Menu
1. Add Book
2. Update Book
3. Delete Book
4. Save and Exit
Choice :
```

Menu 1

Ketika user meng-input menu 1, maka aplikasi akan menampilkan list buku dan meminta input dan validasi sebagai berikut:

- Title : 5 – 30 karakter.
- Author : 5 – 30 karakter.
- Quantity : 1 – 20.
- Price : 100.000 – 800.000.

Jika input user salah, maka aplikasi akan meminta input kembali. Setelah berhasil input maka buku tersebut akan masuk ke list buku.

```
Add Book
No  Title                                     Author                Quantity  Price
1   Algorithm and Programming               Clifford Stein         8         500000
2   Calculus                               Morris Kline          13        200000
3   Linear Algebra                         Steven Roman          4         300000

Insert book title[5..30] : Object Oriented Programming
Insert book author[5..30] : Carroll
Insert book quantity[1..20] : 5
Insert book price [100000..800000] : 280000

Book Added
```

Menu 2

Ketika user meng-input menu 2, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Buku yang ingin di-update : input antara 1 sampai jumlah jenis buku.
- Title : 5 – 30 karakter.
- Author : 5 – 30 karakter.
- Quantity : 1 – 20.

- Price : 100.000 – 800.000.

Setelah diinput maka aplikasi akan mengubah data buku yang di list.

Update Book				
No	Title	Author	Quantity	Price
1	Algorithm and Programming	Clifford Stein	8	500000
2	Calculus	Morris Kline	13	200000
3	Linear Algebra	Steven Roman	4	300000
4	Object Oriented Programming	Carroll	5	280000

Update book no[1..4] : 3
 Insert book title[5..30] : Mathematic Discrete
 Insert book author[5..30] : Steven Roman
 Insert book quantity[1..20] : 10
 Insert book price[100000..800000] : 150000

Book Updated

Jika user memilih menu 2 dan belum ada list buku, maka aplikasi akan menampilkan “Nothing to update”.

Menu 3

Ketika user meng-input menu 3, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Buku yang ingin dihapus : input antara 1 sampai jumlah jenis buku.

Setelah diinput maka buku akan dihapus dari list.

Delete Book				
No	Title	Author	Quantity	Price
1	Algorithm and Programming	Clifford Stein	8	500000
2	Calculus	Morris Kline	13	200000
3	Mathematic Discrete	Steven Roman	10	150000
4	Object Oriented Programming	Carroll	5	280000

Delete book no[1..4] : 2

Book Deleted

Jika user memilih menu 3 dan belum ada list buku, maka aplikasi akan menampilkan “Nothing to delete”.

Menu 4

Jika user memilih menu 4, maka aplikasi akan menyimpan list buku ke dalam file .txt.

Course-Net
Empowering IT People

Soal Latihan Pertemuan 11

Buatlah sebuah aplikasi untuk Course-Net Training Center. Aplikasi tersebut akan digunakan untuk mengelola data peserta yang terdapat di Course-Net Training seperti untuk menambah, update, hapus, dan sorting data peserta. Data-data peserta tersebut akan diambil dan disimpan ke dalam file .txt ketika ada perubahan data.

Berikut list peserta yang telah tersedia saat ini dan dapat disimpan ke dalam file .txt.

```
2201453637#Budie#3.80#DKV
2207758329#Andie#3.78#IT
2205280592#Kayleen#3.50#Accountant
2201984757#Barry#3.47#MTI
```

Menu Home

Pada awal, program akan membaca file .txt, menampilkan list peserta, dan menu untuk dipilih user. Menu hanya bisa dipilih antara angka 1 – 6.

```
Course-Net Training Center
No  NIM          Name          GPA    Major
1   2201453637   Budie         3.80   DKV
2   2207758329   Andie         3.78   IT
3   2205280592   Kayleen       3.50   Accountant
4   2201984757   Barry         3.47   MTI

Menu
1. Add Student
2. Update Student
3. Delete Student
4. Sort Student NIM descending
5. Sort Student Name ascending
6. Save and Exit

Choice :
```

Course-Net
Empowering IT People

Menu 1

Ketika user meng-input menu 1, maka aplikasi akan menampilkan list peserta dan meminta input dan validasi sebagai berikut:

- NIM : angka dan berjumlah 10 digit.
- Name : 5 – 30 karakter.
- GPA : 1 – 4.
- Major : 2 – 15 karakter.

Jika input user salah, maka aplikasi akan meminta input kembali. Setelah berhasil input maka peserta tersebut akan masuk ke list peserta.

```
Add Student
No  NIM                Name                GPA  Major
1   2201453637         Budie                3.80 DKV
2   2207758329         Andie                3.78 IT
3   2205280592         Kayleen              3.50 Accountant
4   2201984757         Barry                3.47 MTI

Insert NIM[10 Digits] : 1601263505
Insert student name[5..30] : Bryan
Insert Bryan's GPA[1..4] : 4
Insert Bryan's Major[2..15 Length] : IT

Student Added
```

Menu 2

Ketika user meng-input menu 2, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Peserta yang ingin di-update : input antara 1 sampai jumlah peserta.
- NIM : angka dan berjumlah 10 digit.
- Name : 5 – 30 karakter.
- GPA : 1 – 4.
- Major : 2 – 15 karakter.

Setelah diinput maka aplikasi akan mengubah data peserta yang di list.

```
Update Student
No  NIM                Name                GPA  Major
1   2201453637         Budie                3.80 DKV
2   2207758329         Andie                3.78 IT
3   2205280592         Kayleen              3.50 Accountant
4   2201984757         Barry                3.47 MTI
5   1601263505         Bryan                4.00 IT

Update student no[1..5] : 5
Insert NIM[10 digits] : 2218819382
Insert student name[5..30] : Michael
Insert Michael's GPA[1..4] : 3.8
Insert Michael's Major[2..15 Length] : DKV

Michael Updated
```

Jika user memilih menu 2 dan belum ada list peserta, maka aplikasi akan menampilkan “Nothing to update”.

Menu 3

Ketika user meng-input menu 3, maka aplikasi akan meminta input dan validasi sebagai berikut:

- Peserta yang ingin dihapus : input antara 1 sampai jumlah peserta.

Setelah diinput maka peserta akan dihapus dari list.

Delete Student				
No	NIM	Name	GPA	Major
1	2201453637	Budie	3.80	DKV
2	2207758329	Andie	3.78	IT
3	2205280592	Kayleen	3.50	Accountant
4	2201984757	Barry	3.47	MTI
5	2218819382	Michael	3.80	DKV

Delete student no[1..5] : 5

Michael Deleted

Jika user memilih menu 3 dan belum ada list peserta, maka aplikasi akan menampilkan “Nothing to delete”.

Menu 4

Jika user memilih menu 4, maka list peserta akan di-sorting berdasarkan NIM secara descending.

Course-Net Training Center				
No	NIM	Name	GPA	Major
1	2207758329	Andie	3.78	IT
2	2205280592	Kayleen	3.50	Accountant
3	2201984757	Barry	3.47	MTI
4	2201453637	Budie	3.80	DKV

Menu 5

Jika user memilih menu 5, maka list peserta akan di-sorting berdasarkan nama secara ascending.

Course-Net Training Center				
No	NIM	Name	GPA	Major
1	2207758329	Andie	3.78	IT
2	2201984757	Barry	3.47	MTI
3	2201453637	Budie	3.80	DKV
4	2205280592	Kayleen	3.50	Accountant

Menu 6

Jika user memilih menu 4, maka aplikasi akan menyimpan list peserta ke dalam file .txt.

Course-Net
Empowering IT People