

ALGORTIMA-2

PETUNJUK

PELAPORAN AKKHIR MANAGER (P A M)

Materi :

- 1. Problem/ Soal**
- 2. Ketentuan Tata Tulis**
- 3. Contoh Pelaporan**
- 4. Waktu & Cara Pengumpulan**

SOAL

Jenis Ujian : Take Home Test UAS
Matakuliah : Algoritma-2
Dosen : Sanyata Purwidayanta, MT

Waktu : Lihat Soal
Sifat : Open Resources
Catatan : Menyontek = Nilai Nol

Petunjuk :

- Gunakan akal pikiran Anda yang bersih, orisinal, dan percaya diri
- Prinsip : boleh eksplorasi melalui apa saja, tidak boleh ada jawaban yang sama / menyontek

A. PROBLEM

SOAL-1

Buatkan dokumen Algoritma dan Program Modular dalam bahasa Pascal untuk membuat aplikasi simulasi Array of Record untuk pengolahan data Nilai Indeks matakuliah Algoritma Pemrograman dengan 3 model input yaitu keyboard, generator random, dan file teks. Pemrosesan data yang diperlukan sebagai berikut :

1. Elemen adalah record <NIM, NAMA, KEHADIRAN, UTS, UAS, NA, INDEKS>
2. Jumlah data input kustomisasi, maksimal N = 15 agar bisa tampil di layar tanpa scroll. Hal ini membutuhkan Setting batasan indeks terlebih dulu
3. Data yang diinput hanya NIM, NAMA, jumlah KEHADIRAN, nilai UTS, nilai UAS dan bisa untuk simulasi data otomatis. (petunjuk : gunakan buah 3 Array/Tabel)
4. Jumlah kehadiran total yang seharusnya adalah 16 kali. Kehadiran < 50 % maka dianggap gagal dan diberi NA (Nilai Akhir) = 0. INDEKS nya = "Hadir -"
5. Rumus menghitung jika kehadiran $\geq 50\%$: $NA = 40\% \text{ UTS} + 60\% \text{ UAS}$,
INDEKS : 'A' : $100 \geq NA \geq 85$
'B' : $85 > NA \geq 70$
'C' : $70 > NA \geq 55$
'D' : $55 > NA \geq 40$
'E' : $40 > NA \geq 0$
6. Model interaksi menu terlihat seperti pada gambar di samping

| MENU IPO |
|-----------------------------------|
| [1] Input Data Keyboard Manual |
| [2] Input Data Generator Otomatis |
| [3] Input Data File Text Otomatis |
| ----- |
| [4] Output Laporan Data Manual |
| [5] Output Laporan Data Generator |
| [6] Output Laporan Data File Text |
| ----- |
| [0] Selesai |
| >>> Pilih nomor = ? |

7. Output Laporan Nilai

LAPORAN PENILAIAN MATAKULIAH ALGORITMA-2 TAHUN 2022

| No | NIM | NAMA LENGKAP | % Kehadiran | UTS | UAS | NA | Indeks |
|-----|-----|--------------|-------------|-----|-----|-----|--------|
| 1 | Xxx | xxx | 999 | 999 | 999 | 999 | xxx |
| 2 | Xxx | xxx | 999 | 999 | 999 | 999 | xxx |
| 3 | Xxx | xxx | 999 | 999 | 999 | 999 | xxx |
| ... | | | | | | | |

Rara-rata Nilai = Nilai Tertinggi = Nilai Terkecil = ...

SOAL-2

Buatkan dokumen Algoritma dan Program Modular dalam bahasa Pascal untuk membuat aplikasi simulasi Metode Internal Sorting dari suatu Tabel tertentu untuk mensimulasikan proses pengurutan data secara DESCENDING (urutan dari BESAR ke KECIL). Kriteria yang diperlukan sebagai berikut :

1. Elemen Tabel adalah record <NIM, NAMA, NILAI> yang datanya diinput dari file teks.
2. Jumlah data input kustomisasi, maksimal N = 15 agar bisa tampil di layar tanpa scroll. Hal ini membutuhkan Setting batasan indeks terlebih dulu saat Menyiapkan Tabel dari file teks.
3. Model interaksi menu terlihat seperti pada gambar di samping
4. Menu tetap ditulis lengkap, dan yang wajib kerjakan hanya 4 saja:
 - ✓ menu – [1]
 - ✓ menu – [3],[4],[5] pilih salah satu
 - ✓ menu – [6],[7] pilih salah satu
 - ✓ menu – [8],[9],[A],[B], [C],[D],E] pilih salah satu

MENU SIMULASI METODE INTERNAL SORTING

```
[1]. Menyiapan Tabel dari File Teks
---- << METODE COUNTING SORT >>
[2]. Counting Sort
---- << METODE INSERTION SORT >>
[3]. Natural Insertion-Sort
[4]. Binary Insertion-Sort
[5]. Shell Insertion-Sort
---- << METODE SELECTION SORT >>
[6]. Natural Selection-Sort
[7]. Heap Selection-Sort
---- << METODE EXCHANGE SORT >>
[8]. Natural Bubble Exchange-Sort
[9]. Optimal Bubble Exchange-Sort
[A]. Flag Bubble Exchange-Sort
---
[B]. Shaker Exchange-Sort
[C]. Quick Exchange-Sort
[D]. Radix Exchange-Sort
[E]. Merge Exchange-Sort
[0]--> Selesai
```

>>> Pilih nomor = ?

5. Output Simulasi bisa menyajikan Tabel Awal sebelum diurutkan di sebelah kiri dan Tabel Akhir setelah diurutkan dengan metode tertentu yang dipilih.

METODE : XXXXXXXX

Tabel AWAL

| No | NIM | NAMA | NA |
|-----|-----|------|-----|
| 1 | Xxx | xxx | 999 |
| 2 | Xxx | xxx | 999 |
| 3 | Xxx | xxx | 999 |
| ... | | | |

Tabel Akhir Setelah Nilai Diurutkan

| No | NIM | NAMA | NA |
|-----|-----|------|-----|
| 1 | Xxx | xxx | 999 |
| 2 | Xxx | xxx | 999 |
| 3 | Xxx | xxx | 999 |
| ... | | | |

6. Contoh program : Terlampir “InSort.Zip”

B. KETENTUAN PELAPORAN DAN PENGUMPULAN

- 1) Pelaporan jawaban setiap mahasiswa hanya berupa 1 file saja format **doc atau pdf**
- 2) Pelaporan dikumpulkan dengan jadwal tepat tanggal **7 Juni 2022 jam 01.00 - 23.59**
Wib via email "spurwidayanta@gmail.com" dengan contoh ketentuan :
 - nama dokumen = **Algo2_S1_UAS_NIM_NAMA.Docx**
 - nama Subject = **Algo2_S1_UAS_NIM_NAMA**
- 3) Laporan harus profesional, kreatif, rapi, jelas-padat dan elegan serta harus mengikuti "Petunjuk PAM" (Pelaporan Akhir Manager) dokumen ini.

== Selamat Bekerja ==

KETENTUAN PENULISAN LAPORAN AKHIR

A. Konvensi Naskah

- Ukuran kertas : A4
- Huruf naskah : Times New Roman 12
- Spasi baris naskah : 1 (tunggal)
- Spasi Paragraf : 1,5
- Margin Kiri – Atas – Kanan – Bawah : 3 – 2 – 2 – 2 Cm
- Tabel diberi judul di atas dengan huruf : Arial 10
- Gambar harus diberi frame (kotak) diberi judul di bawah dengan huruf: Arial 10
- Penomoran halaman dokumen naskah dibagi 3 bagian
 - 1) Bagian Lembaran Awal
 - Cover : tanpa nomor halaman
(bebas berkreasi seindah dan se-elegant mungkin)
 - Filosofi Matakuliah : nomor format romawi kecil, i, ii, iii, dst (Maks 1 halaman)
 - Moto-Kesan-Saran : format romawi kecil, lanjutan filosofi (Maks 1 halaman)
 - Kata Pengantar : format romawi kecil, lanjutan M-K-S (Maks 1 halaman)
 - Daftar Isi : format romawi kecil, lanjutan kata pengantar
 - 2) Bagian Lembaran Isi Naskah
 - BAB I : nomor format arab 1, 2, 3, dst
 - BAB II : nomor format arab, lanjutan BAB I
 - BAB III : nomor format arab, lanjutan BAB II
 - BAB ... : dan seterusnya
 - 3) Bagian Lembaran Akhir
 - Daftar Pustaka : Nyambung nomor halaman Bagian Isi Naskah (BAB)
 - Curriculum Vitae : tanpa nomor halaman (bebas berkreasi, elegant)

B. Ketentuan Bagian Lembaran Isi Naskah

Catatan : BAB bisa bertambah menyesuaikan Produk yang dibuat

- BAB I : UNTUK KARYA PRODUK SOAL-1
- BAB II : UNTUK KARYA PRODUK SOAL-2
- BAB III : UNTUK KARYA TAMBAHAN MANDIRI LAINNYA
- Dst

MATAKULIAH ALGORITMA 2



LAPORAN AKHIR KINERJA MAHASISWA MANAGER STMIK DCI

SANYATA PURWIDAYANTA

11120007



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
STMIK DCI**

2022

FILOSOFI MATAKULIAH

Tulisan ini harus dari kata-kata sendiri dengan merenungi ide-ide yang terdapat dalam matakuliah ALGORITMA selama pembelajaran yang berguna untuk motivasi hidup diri sendiri.

TIDAK mengulang teori tertulis yang telah disampaikan.

TIDAK berisi teori yang dicopy- paste dari tulisan kuliah atau sumber lain.

TIDAK boleh lebih dari 1 halaman.

Satu contoh Filosofi ALGORITMA :

Pertama : ALGORITMA MODULAR mengajarkan untuk pribadi seorang manager informatika yang berkarakter mandiri, mampu bekerja dan berkomunikasi dalam team work, karena trampil membagi tugas dalam bentuk modul-modul dan bisa mengintegrasikan diri untuk tujuan bersama yaitu program besar..

Kedua : ...

Ketiga : ...

Dst....

(kata “pertama”, “kedua”, dst tidak harus ditulis, bisa diganti nomor 1,2,3, dst bergantung cara menyusun kalimat pengantar filosofinya)

MOTO – KESAN - SARAN

MOTO :

“ ”
.....

KESAN :

1.
.....
2.
.....
3. dst.

SARAN :

1.
.....
2.
.....
3. dst

KATA PENGANTAR

PEMBUKA : Kami senantiasa memanjatkan puji dan syukur ke hadirat Allah SWT yang telah mengajarkan kepada manusia dengan perantaraan pena. IlmuNya meliputi apa yang ada di bumi dan yang ada di langit dan yang ada diantaranya. Sholawat dan salam semoga terucurah kepada junjungan dan teladan alam Nabi Muhammad SAW. Kami berharap kelembutan dan keberkahan ilmu Allah SWT dilimpahkan kepada kami sehingga memberikan manfaat dan memberikan petunjuk menuju jalan kebenaran yang diridhoi. Dengan karuniaNya pula, kami mengembangkan model laporan atau jawaban kinerja manager agar dapat memberikan manfaat sesuai dengan harapan dan tujuan.

ISI : Sajian laporan dan jawaban merupakan resume eksekutif agar sistematis, sederhana, dan mudah dipahami dan dipelajari. Selanjutnya dapat dikembangkan ke arah yang lebih baik..... dst.

PENUTUP : Akhir kata, laporan ini dapat memberikan karakter yang baik Kami menyadari kinerja ini tentunya tidak sempurna dan ideal, banyak kekurangan sehingga memerlukan kritik dan penyempurnaan yang berkesinambungan. Untuk itu, kami sangat terbuka dengan segala saran dan kritik pada buku ini agar menjadi lebih baik di masa mendatang menyesuaikan zamannya. Harapan kami semoga karya ini dapat memberikan manfaat sesuai dengan maksud dan tujuannya.

Bulan Tahun

Penyusun

Sanyata Purwidayanta

DAFTAR ISI

| | |
|---|------|
| SOAL | i |
| KETENTUAN PENULISAN LAPORAN AKHIR | iv |
| FILOSOFI MATAKULIAH | vi |
| MOTO – KESAN - SARAN..... | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI..... | ix |
| BAB 1 KARYA PRODUK PROGRAM “APLIKASI SIMULASI GENERATOR DATA ENTRI MODEL PROSES IPO – Input Proses Output” | |
| 1.1 PROBLEM | 1 |
| 1.2 SOLUSI ALGORITMA | 2 |
| 1.2.1. JUDUL/DEFINISI | 2 |
| 1.2.2. SKEMA KAMUS DATA | 2 |
| 1.2.3. SKEMA PROGRAM | 3 |
| 1.2.4. SKEMA MODUL | 3 |
| 1.2.5. SKEMA PROCEDURE/FUNCTION | 6 |
| 1.3 SOLUSI PROGRAM | 9 |
| 1.3.1. DESAIN ANTARMUKA USER | 9 |
| 1.3.2. SOURCE CODE : PROGRAM IPO | 9 |
| 1.3.3. SOURCE CODE : UNIT U_IPO2 | 11 |
| 1.3.4. SOURCE CODE : UNIT U_BASIC | 16 |
| 1.3.5. DUKUNGAN FILE DATA | 18 |
| 1.3.6. PETUNJUK SINGKAT INSTALASI..... | 19 |
| BAB 2 KARYA PRODUK PROGRAM “APLIKASI SIMULASI METODE INTERNAL SORTING” | |
| 2.1 PROBLEM | 20 |
| 2.2 SOLUSI ALGORITMA | 21 |
| 2.2.1. JUDUL/DEFINISI | 21 |
| 2.2.2. SKEMA KAMUS DATA | 21 |
| 2.2.3. SKEMA PROGRAM | 21 |
| 2.2.4. SKEMA MODUL | 21 |
| 2.2.5. SKEMA PROCEDURE/FUNCTION | 21 |
| 2.3 SOLUSI PROGRAM | 22 |
| 2.3.1. DESAIN ANTARMUKA USER | 22 |
| 2.3.2. SOURCE CODE : PROGRAM IPO | 22 |
| 2.3.3. SOURCE CODE : UNIT XXX | 22 |
| 2.3.4. SOURCE CODE : UNIT XXX | 22 |
| 2.3.5. DUKUNGAN FILE DATA | 23 |
| 2.3.6. PETUNJUK SINGKAT INSTALASI..... | 23 |
| DAFTAR PUSTAKA | 24 |

BAB 1

KARYA PRODUK PROGRAM

“APLIKASI SIMULASI GENERATOR DATA ENTRI MODEL PROSES IPO – Input Proses Output”

1.1 PROBLEM

Membuat dokumen Algoritma dan Program Modular dalam bahasa Pascal untuk membuat aplikasi simulasi Array of Record untuk pengolahan data Nilai Indeks matakuliah Algoritma Pemrograman dengan 3 model input yaitu keyboard, generator random, dan file teks.

Pemrosesan data yang diperlukan sebagai berikut :

1. Elemen adalah record <NIM, NAMA, KEHADIRAN, UTS, UAS, NA, INDEKS>
2. Jumlah data input kustomisasi, maksimal N = 15 agar bisa tampil di layar tanpa scroll. Hal ini membutuhkan Setting batasan indeks terlebih dulu
3. Data yang diinput hanya NIM, NAMA, jumlah KEHADIRAN, nilai UTS, nilai UAS dan bisa untuk simulasi data otomatis. (petunjuk : gunakan buah 3 Array/Tabel)
4. Jumlah kehadiran total yang seharusnya adalah 16 kali. Kehadiran < 50 % maka dianggap gagal dan diberi NA (Nilai Akhir) = 0. INDEKS nya = “Hadir -“
5. Rumus menghitung jika kehadiran $\geq 50\%$: $NA = 40\% \text{ UTS} + 60\% \text{ UAS}$,

INDEKS : ‘A’ : $100 \geq NA \geq 85$

‘B’ : $85 > NA \geq 70$

‘C’ : $70 > NA \geq 55$

‘D’ : $55 > NA \geq 40$

‘E’ : $40 > NA \geq 0$

6. Model interaksi menu dan kriteria modular

| MENU |
|---|
| [1] Input Data Keyboard Manual [2] Input Data Generator Otomatis [3] Input Data File Text Otomatis [4] Output Laporan Data Manual [5] Output Laporan Data Generator [6] Output Laporan Data File Text [0] Selesai |
| >>> Pilih nomor = ? |

7. Output Laporan Nilai

LAPORAN PENILAIAN MATAKULIAH ALGORITMA-2 TAHUN 2022

| No | NIM | NAMA LENGKAP | % Kehadiran | UTS | UAS | NA | Indeks |
|-----|-----|--------------|-------------|-----|-----|-----|--------|
| 1 | Xxx | xxx | 999 | 999 | 999 | 999 | xxx |
| 2 | Xxx | xxx | 999 | 999 | 999 | 999 | xxx |
| 3 | Xxx | xxx | 999 | 999 | 999 | 999 | xxx |
| ... | | | | | | | |

Rara-rata Nilai = Nilai Tertinggi = Nilai Terkecil = ...

1.2 SOLUSI ALGORITMA

1.2.1. JUDUL/DEFINISI

Program Simulasi_IPO_GeneratorDatadanFile diberi nama IPO2.exe perpanjangan dari Input-Proses-Output untuk matakuliah Algoritma-2. IPO2 merupakan skema problem program untuk mensimulasikan beberapa kasus pembelajaran Algoritma Modular yang meliputi :

- 1) Type Data : Array - Record - File
- 2) Kreativitas
 - a. Model Entri : Manual Keyboard - Generator Random - File Text
 - b. Model proses : Logika IPO (Input - Proses - Output)
Pemanfaatan Statistik dasar - Bilangan Random
 - c. Studi kasus : Pengelolaan Nilai Matakuliah

Skema IPO :

- ✓ **Input** : Record Nilai Matakuliah
<NIM, NAMA, KEHADIRAN, UTS, UAS, NA, INDEKS>
- ✓ **Proses** : Model input manual dari keyboard
Model input otomatis dari Generator
Model input otomatis dari File Teks
Proses Penilaian dari data setiap Model Input
Proses Penyajian laporan dari setiap Model Input
- ✓ **Output**: Pelaporan penilaian berbentuk tabular

1.2.2. SKEMA KAMUS DATA

KASUS :

- Array 1 Dimensi : TABEL OF ELEMEN RECORD
- Bilangan Random untuk Generator data
- File Teks untuk input file

TYPE

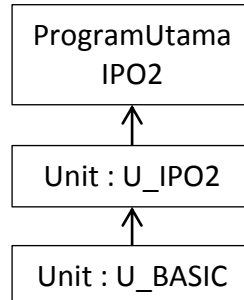
```
ElemenNilai = Record {Elemen Tabel}
                Nim    : String[8];
                Nama    : String[30];
                Khdr    : Integer;    // Kehadiran
                UTS     : Integer;
                UAS     : Integer;
                NA      : Real;        // Nilai Akhir
                Idx     : String[10]; // Indeks Nilai
            End;
```

```
TabelNilai = Array[1..NMax] of ElemenNilai;
File_Text  = TEXT;
```

1.2.3. SKEMA PROGRAM

KASUS :

- Skema Menu
- Skema Modul linear



PROGRAM IPO2

Link modul : U_IPO2

Procedure **SettingMaxIndeks**(IdxAktual)

Procedure **INISIALISASI**

Procedure **SELESAI**

BEGIN {-Algoritma Utama-}

Call **SettingMaxIndeks**(IdxAktual)

Call **INISIALISASI**

Repeat

Call **MenuIPO**(Pilihan)

Call **ProsesIPO**(Pilihan, IdxAktual)

Until (Pilihan = '0');

Call **SELESAI**

END.

UNIT IPO2

Link modul : U_BASIC

--- Rencana Procedure dan Function utama :

Procedure **MenuIPO**

Procedure **ProsesIPO**(Pilihan, IdxAktual)

Procedure **Input_KEYBOARD**

Procedure **Input_GENERATOR**

Procedure **OpenFileText**

Procedure **Input_FILE**

Procedure **Proses_NILAI** (dari input Keyboard/Generator/File)

Procedure **Proses_STATISTIK** (dari input Keyboard/Generator/File)

Procedure **Pelaporan** (dari input Keyboard/Generator/File)

--- Untuk generator Random

Function **GetNIM**

Function **GetNAMA**

Function **GetNILAI** (UTS, UAS, dan Kehadiran)

1.2.4. SKEMA MODUL

PROGRAM UTAMA: { ini dibuat detil untuk contoh }

```
Procedure SettingMaxIndeks(var N : Integer);
{I.S : N integer sembarang, maksimum N = 15 }
{F.S : N terdefinisi 0 < N <= 15 }
Begin
    Input(KB,'Jumlah Indeks Array ? ',N);
End;

Procedure INISIALISASI;
{I.S : - }
{F.S : untuk insisiasi proses sebelum pengulangan}
Begin
    Pilihan ← '0';
End;

Procedure SELESAI;
{I.S : - }
{F.S : Hanya sekedar pesan selesai }
Begin
    Output(MNT,'Selesai ! <Enter> to continue...');
    // Tahan sampai tekan enter
End;
```

UNIT IPO2 :

```
Procedure MenuIPO(in/out Pil : Char);
{I.S: Pil sembarang karakter }
{F.S: Pil terdefinisi '0' s/d '6' }

Procedure ProsesIPO(in Pil: Char; in Idx :Integer);
{I.S: Pil aktual tertentu, Idx aktual tertentu }
{F.S: Pil tetap, Idx Tetap }

Procedure ViewLayarEntry(in TC,TB : Integer);
{I.S: TC warna textcolor, TB warna TextBackround }
{F.S: Layar etry tersajikan sesuai warna, di koordinat fixed }

Procedure Input_KEYBOARD(in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx sembarang, indek N terdefinisi }
{F.S: Tx terisi N record data input manual }

Procedure Input_GENERATOR(in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx sembarang, indek N terdefinisi }
{F.S: Tx terisi N record data hasil Generator }

Procedure OpenFileText(in/out F : TEXT; in NamaFile: String);
{I.S: F fileteks, "Namafilename.ekstensi" maks 8.3 karakter }
{F.S: File F aktif }

Procedure Input_FILE (in/out F: FileText;
```

```

        in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx sembarang, indek N terdefinisi }
{F.S: Tx terisi N record data hasil dari input file }

Procedure Proses_NILAI(in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx Tabel terdefinisi Inputnya }
{F.S: Field Tx.NA & Tx.Hhr terkomputasi }

Procedure Proses_STATISTIK(in Tx: TabelNilai; in gN: Integer;
        in/out RR, NMax, NMin : Real);
{I.S: Tx tertefinisi, gN banyak indeks data }
{F.S: RR= TX rata-rata, NMax = NA terbesar, Nmin = NA terkecil }

Procedure Pelaporan(in Tx: TabelNilai; in/out gN:Integer;
        in Sx: String);
{I.S: Tx terdefinisi, indek N terdefinisi, Sx terdefinisi judul }
{F.S: Tersajikan Tx terisi N record data ke layar }

Function GetNim → String;
{IN : - }
{R.V: Nim random 5 digit }

Function GetNama → String;
{IN : - }
{R.V: Nama random 11 dgit }

Function GetNilai(N:Word) → Word; // khusus untuk random pascal
{IN : - }
{R.V: Nnilai random 11 dgit }

```

1.2.5. SKEMA PROCEDURE/FUNCTION

Catatan :

- Realisasi Algoritma Sengaja Tidak Disajikan untuk bahan belajar dan eksplorasi.
- Beberapa disajikan untuk contoh kajian studi

VAR

- **Tabel1, Tabel2, Tabel3 : TabelNilai;**
- **FT : File_Text : TEXT;**

```
-----

Procedure OpenFileText(in/out F : TEXT; in NamaFile: String);
{I.S: F fileteks, "Namafilename.ekstensi" maks 8.3 karakter }
{F.S: File F aktif }
Begin
    {====proses File }
    Assign(F,NamaFile);
    {$I-}                                { non-aktifkan Compiler Pascal }
    Reset(F);                            { mengambil variabel file IOResult }
    {$I+}                                { aktifkan Compiler Pascal }
    If (IOResult <> 0) Then                { file belum siap, file tidak ada }
        Rewrite(F);                      { buat file baru karena file tidak ada }
                                         { IOResult=0 berarti file sudah siap }
End;
```



```
Procedure ProsesIPO(in Pil: Char; in Idx :Integer);
{I.S: Pil aktual tertentu, Idx aktual tertentu }
{F.S: Pil tetap, Idx Tetap }
Begin
    Depend Of Pil
        '1' : Input_KEYBOARD(Tabel1,Idx);
        '2' : Input_GENERATOR(Tabel2,Idx);
        '3' : OpenFileText(FT,'FileData.txt');
              Input_FILE(FT,Tabel3,Idx);
        '4' : Proses_Nilai(Tabel1,Idx);
              Pelaporan(Tabel1,Idx,'KEYBOARD MANUAL');
        '5' : Proses_Nilai(Tabel2,Idx);
              Pelaporan(Tabel2,Idx,'GENERATOR OTOMATIS');
        '6' : Proses_Nilai(Tabel3,Idx);
              Pelaporan(Tabel3,Idx,'FILE OTOMATIS');
    EndDepend
end;
```



```

Procedure Input_KEYBOARD(in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx sembarang, indeks N terdefinisi      }
{F.S: Tx terisi N record data input manual    }
Var i : integer;
Begin
  i traversal 1..gN
    Output(MNT,'INPUT DATA ke - ', i);
    ViewLayarEntry(White,Brown);
    Input(KB, (Tx[i].Nim,Tx[i].Nama,Tx[i].Khdr,Tx[i].UTS,Tx[i].UAS));
  Endi
End;

Procedure Input_GENERATOR(in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx sembarang, indeks N terdefinisi      }
{F.S: Tx terisi N record data hasil Generator }
Var i : integer;
Begin
  i traversal 1..gN
    Output(MNT,'INPUT DATA ke - ', i);
    ViewLayarEntry(White,Brown);
    Tx[i].Nim ← GetNim
    Tx[i].Nama ← GetNama
    Tx[i].Khdr ← GetNilai(16)
    Tx[i].UTS ← GetNilai(100)
    Tx[i].UAS ← GetNilai(100)
  Endi
End;

Procedure Input_FILE (in/out F: FileText;
                     in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx sembarang, indeks N terdefinisi      }
{F.S: Tx terisi N record data hasil dari input file }
Var i : integer;
Begin
  i traversal 1..gN
    Output(MNT,'INPUT DATA ke - ', i);
    ViewLayarEntry(White,Brown);
    Input(F, (Tx[i].Nim,Tx[i].Nama,Tx[i].Khdr,Tx[i].UTS,Tx[i].UAS));
  Endi
End;

```

```

Procedure Proses_NILAI(in/out Tx: TabelNilai; in/out gN:Integer);
{I.S: Tx Tabel terdefinisi Inputnya          }
{F.S: Field Tx.NA & Tx.Hhr terkomputasi      }
Var i : integer;
Begin
  i traversal 1..gN
    If (Tx[i].Khdr/16 >= 0.50) Then begin
      Tx[i].NA := 0.60 * Tx[i].UAS + 0.40 * Tx[i].UTS;
      If (Tx[i].NA >= 85) Then
        Tx[i].IDX := 'A'
      else If (Tx[i].NA >= 70) then
        Tx[i].IDX := 'B'
      else If (Tx[i].NA >= 55) then
        Tx[i].IDX := 'C'
      else If (Tx[i].NA >= 40) then
        Tx[i].IDX := 'D'
      else
        Tx[i].IDX := 'E';
    end else begin
      Tx[i].NA := 0;
      Tx[i].IDX := 'Hadir -';
    end;

  Endi
End;

```

```

Procedure Pelaporan(in Tx: TabelNilai; in/out gN:Integer;
                   in Sx: String);
{I.S: Tx terdefinisi, indek N terdefinisi, Sx terdefinisi judul }
{F.S: Tersajikan Tx terisi N record data ke layar                }

Var NArr, NAmax, NAmin : Real;
  i: Integer; ch: Char;
Begin
  Proses_STATISTIK(Tx,gN, NArr, NAmax, NAmin);
  // tampilkan laporan tabular
  Ouput(MNT,' Rata-rata= ',NArr,' NA Terbesar= ',NAmax,
        ' NA Terkecil= ',NAmin)
End;

```

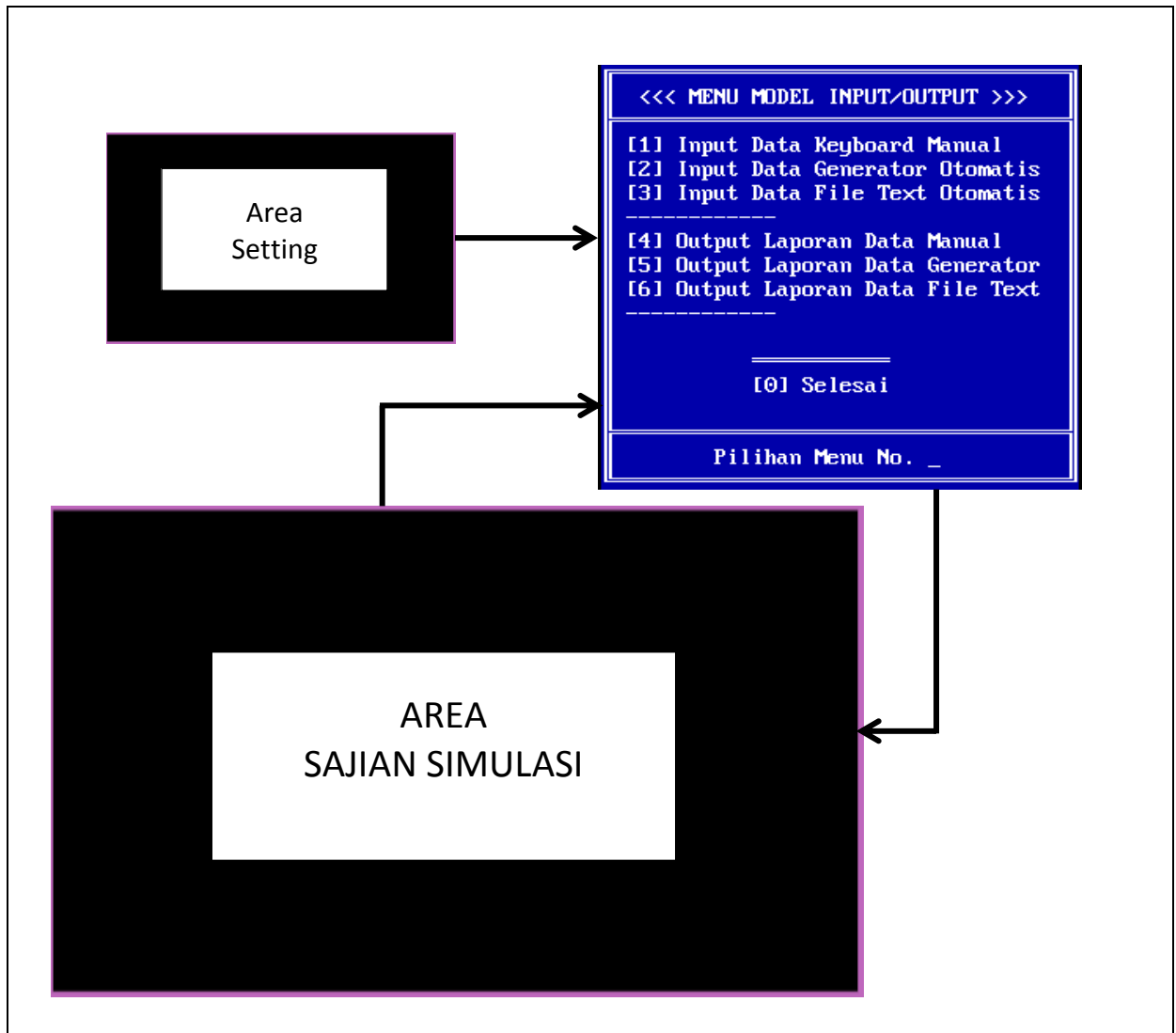
```

Function GetNilai(N:Word) → Word;
{IN : -                      }
{R.V: Nnilai random 11 dgit }
Var N1 : Word;
Begin
  N1 := Random(N); {manipulasi untuk UTS, UAS, kehadiran}
  → (N1)
End;

```

1.3 SOLUSI PROGRAM

1.3.1. DESAIN ANTARMUKA USER



1.3.2. SOURCE CODE : PROGRAM IPO

A). PROGRAM UTAMA

Program Simulasi_IPO_GeneratorDatadanFile;

```
{=====
Input   : Record Nilai Matakuliah
          <NIM, NAMA, KEHADIRAN, UTS, UAS, NA, INDEKS>
Proses  : Model input manual dari keyboard
          Model input otomatis dari Generator
          Model input otomatis dari File Teks
          Proses Penilaian dari data setiap Model Input
          Proses Penyajian laporan dari setiap Model Input
Output  : Pelaporan penilaian berbentuk tabular
=====}
```

```

{----- Kamus Global -----}

Uses Crt, U_IPO2;

Type FileText  = TEXT;
    Index      = Integer;
    TipeChar   = Char;

Var  FT          : FileText;
     IdxAktual   : Index;
     Pilihan     : TipeChar;

Procedure SettingMaxIndeks(var N : Integer);
{I.S : N integer sembarang, maksimum N = 15 }
{F.S : N terdefinisi 0 < N <= 15           }
Begin
    TextBackground(Black); TextColor(White); ClrScr;
    GotoXY(20,10); Write('Setting dulu...');
    GotoXY(20,11); Write('-----');
    GotoXY(20,13); Write('Jumlah Indeks array ? _');
    GotoXY(20,14); Write('(Maksimum = 15)');
    GotoXY(42,13); Readln(N);
End;

Procedure INISIALISASI;
Begin
    Pilihan := '0';
End;

Procedure SELESAI;
Begin
    TextBackGround(Black); ClrScr;
    GotoXY(30,12); Write('Selesai ! <Enter> to continue...');
    readln;
End;

{-Program Utama-}
Begin
    SettingMaxIndeks(IdxAktual);
    INISIALISASI;
    Repeat
        MenuIPO(Pilihan);
        ProsesIPO(Pilihan, IdxAktual);
    Until (Pilihan = '0');
    SELESAI;
End.

```

1.3.3. SOURCE CODE : UNIT U_IPO2

```
UNIT U_IPO2;
INTERFACE
{----- Kamus Global -----}
Uses Crt, U_Basic;

Const NMax = 15;
Type ElemenNilai = Record {struktur data record nilai matakuliah}
    Nim      : String[8];
    Nama     : String[30];
    Khdr     : Integer;
    UTS      : Integer;
    UAS      : Integer;
    NA       : Real;
    Idx      : String[10];
End;
TabelNilai  = Array[1..NMax] of ElemenNilai;
FileText    = TEXT;

Var  Tabel1, Tabel2, Tabel3 : TabelNilai;
     FT      : FileText;

{-----Interface Procedure/Function -----}
Procedure MenuIPO(var Pil : Char);
{I.S: Pil sembarang karakter      }
{F.S: Pil terdefinisi '0' s/d '6' }
Procedure OpenFileText(var F : TEXT; NamaFile: String);
{I.S: F fileteks, "Namafile.ekstensi" maks 8.3 karakter }
{F.S: File F aktif                  }
Procedure ProsesIPO(Pil: Char; Idx :Integer);
{I.S: Pil aktual tertentu, Idx aktual tertentu }
{F.S: Pil tetap, Idx Tetap                  }
Procedure ViewLayarEntry(TC,TB : Integer);
{I.S: TC warna textcolor, TB warna TextBackround      }
{F.S: Layar etry tersajikan sesuai warna, di koordinat fixed }
Procedure Input_KEYBOARD(var Tx: TabelNilai; var gN:Integer);
{I.S: Tx sembarang, indeks N terdefinisi      }
{F.S: Tx terisi N record data input manual    }
Procedure Input_GENERATOR(var Tx: TabelNilai; var gN:Integer);
{I.S: Tx sembarang, indeks N terdefinisi      }
{F.S: Tx terisi N record data hasil Generator  }
Procedure Input_FILE (var F: FileText; var Tx: TabelNilai; var gN:Integer);
{I.S: Tx sembarang, indeks N terdefinisi      }
{F.S: Tx terisi N record data hasil dari input file }
Procedure Proses_NILAI(var Tx: TabelNilai; var gN:Integer);
{I.S: Tx Tabel terdefinisi Inputnya          }
{F.S: Field Tx.NA & Tx.Hhr terkomputasi      }
Procedure Proses_STATISTIK(Tx: TabelNilai; gN: Integer; var RR, NMax, NMin : Real);
{I.S: Tx terdefinisi, gN banyak indeks data      }
{F.S: RR= TX rata-rata, NMax = NA terbesar, Nmin = NA terkecil }
Procedure Pelaporan(Tx: TabelNilai; var gN:Integer; Sx: String);
{I.S: Tx terdefinisi, indeks N terdefinisi, Sx terdefinisi judul }
{F.S: Tersajikan Tx terisi N record data ke layar      }

Function GetNim : String;
{IN : -          }
{R.V: Nim random 5 digit }
Function GetNama : String;
{IN : -          }
{R.V: Nama random 11 dgit }
Function GetNilai(N:Word): Word;
{IN : -          }
{R.V: Nnilai random 11 dgit }
```

$$\{ \text{-----} \}$$

12

[illegible]


```
GotoXY(52,i+4);Writeln(Tx[i].UAS);
GotoXY(62,i+4);Writeln(Tx[i].NA:3:0);
GotoXY(72,i+4);Writeln(Tx[i].Idx);
End;

GotoXY(01,gN+4+1);Writeln('iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii');
GotoXY(01,gN+4+2);Writeln('*** NA Rata-rata = ',NArr:3:0,'   NA Terbesar =
',NAmay:3:0,'   NA Terkecil = ',NAMin:3:0);
GotoXY(01,gN+4+3);Writeln(' ');
GotoXY(01,gN+4+4);Writeln('- Kehadiran Minimal 50% dari 16 pertemuan,
jika kurang NA=0 & HADIR-/kurang ');
GotoXY(01,gN+4+5);Writeln('- Indeks A: NA>=85; B: NA>=70; C: NA>=55; D:
NA>=40; E: NA <40 ');
Ch:=Readkey;
End;

Function GetNim : String;
{IN : - }
{R.V: Nim random 5 digit }
Var Nim: Word; S: String; L: Integer;
Begin
Randomize;
Nim := Random (10000);
Str(Nim,S); S := '0000'+S;
L := length(S);
GetNim := Copy(S,L-4,L);
end;

Function GetNama : String;
{IN : - }
{R.V: Nama random 11 dgit }
Var Nama: Word; S: String; j: Integer;
Begin
{- Sengaja tidak disajikan untuk belajar dan eksplorasi
- LIHAT ALGORITMA }
end;

Function GetNilai(N:Word): Word;
{IN : - }
{R.V: Nnilai random 11 dgit }
Var Nl : Word;
Begin
Nl := Random(N); {manipulasi untuk UTS, UAS, kehadiran}
GetNilai := Nl;
End;

BEGIN
{---- tanpa inisialisasi ---}
END.
```

1.3.4. SOURCE CODE : UNIT U_BASIC

```
UNIT U_Basic;

INTERFACE
  USES Crt;
  Const
    TLeft    = 'É'; { chr(201) }   TRight   = '»'; { chr(187) }
    BLeft    = 'È'; { chr(200) }   BRight   = '¼'; { chr(188) }
    BMedium  = 'Ê'; { chr(202) }   TMedium  = 'Ë'; { chr(203) }
    RMedium  = '¹'; { chr(185) }   LMedium  = '¨'; { chr(204) }
    Cross    = 'Î'; { chr(206) }   HLine    = 'Í'; { chr(205) }
    VLine    = 'º'; { chr(186) }

    TlLeft   = chr(218);  TlRight  = chr(191);
    BlLeft   = chr(192);  BlRight  = chr(217);
    TlMedium = chr(194);  BlMedium = chr(193);
    LlMedium = chr(195);  RlMedium = chr(180);
    Cross1   = chr(197);  HlLine   = chr(196);
    VlLine   = chr(179);

  Procedure CreateBox(X1,Y1,X2,Y2 :Integer; var CF, CB:Integer);
  {IS: (X1,Y1)-(X2,Y2) koordinat sembarang, Cl warna sembarang}
  {FS: terbentuk kotak koordinat (X1,Y1)-(X2,Y2) dgn warna CF
    dan Background CB, dan menormalkan kembali putih-hitam}

  Procedure VReplicate(CC: Char; N,X,Y: Integer);
  { I.S.: Karakter CC dan banyaknya pengulangan N, minimal N=1 }
  { F.S.: Tercetak string CC sebanyak N kali Vertikal di kolom X}

  Function Replicate(CC: Char; N: Integer): String;
  { IN :Karakter CC dan banyaknya pengulangan N, minimal N=1}
  { R.V:Sebuah string = konkatenasi dari CC sebanyak N kali }

IMPLEMENTATION
{=====}
  Procedure CreateBox(X1,Y1,X2,Y2 :Integer; var CF, CB:Integer);
  {IS: (X1,Y1)-(X2,Y2) koordinat sembarang, Cl warna sembarang}
  {FS: terbentuk kotak koordinat (X1,Y1)-(X2,Y2) dgn warna CF
    dan Background CB, dan menormalkan kembali putih-hitam}

  var i : integer; { i adalah identifikasi kolom dan baris}
  begin
    {menentukan warna pilihan}
    textcolor(CF);
    textbackground(CB);
    {membuat kotak}
    GotoXY(X1,Y1); write(TLeft,Replicate(HLine,X2-X1-1),TRight);
    for i := Y1+1 to Y2-1 do begin
      gotoxy(X1,i); writeln(VLine);
      gotoxy(X2,i); writeln(VLine);
    end;
```

```

    GotoXY(X1,Y2); write(BLeft,Replicate(HLine,X2-X1-1),BRight);
    {mengembalikan teks ke warna asal putih}
    textcolor(White);
    textbackground(Black);
end;

Procedure VReplicate(CC: Char; N,X,Y: Integer);
{I.S: Karakter CC dan banyaknya pengulangan N, minimal N=1 }
{F.S: Tercetak string CC sebanyak N kali Vertikal di kolom X}
Var i: integer;
Begin {Algoritma}
    for i:=1 to n do begin { i traversal 1..n}
        gotoXY(X,Y+i-1); Write(CC);
    end;
End;

Function Replicate(CC: Char; N: Integer): String;
{IN :Karakter CC dan banyaknya pengulangan N, minimal N=1}
{R.V:Sebuah string = konkatenasi dari CC sebanyak N kali }
Var i: integer; s : String;
Begin {Algoritma}
    s := ''; {inisialisasi}
    for i:=1 to n do { i traversal 1..n}
        s := s + CC;
    Replicate := s; {pengembalian fungsi --> s}
End;

BEGIN
    {---- tanpa inisialisasi ---}
END.

```

1.3.5. DUKUNGAN FILE DATA

FILEDATA.TXT

11122001

Ali Nordin

16

90

100

11122002

Budiman Irawan

15

80

90

11122003

Cantika Arum

14

75

85

11122004

Diana Rahmani

13

100

100

11122005

Emilia Larasati

15

100

80

11122006

Ferriansyah

12

40

60

11122007

Gun Gun Gunawan

10

50

50

... dst disiapkan sampai 15 data

1.3.6. PETUNJUK SINGKAT INSTALASI

README.TXT

**** IDE :

Program KREASI Simulasi Sederhana untuk mempelajari kasus Input-Proses-Output dengan kreasi :

- 1) Input dalam 3 model : Input Keyboard Manual, Input Generate Otomatis , Input Otomatis dari Fileteks
- 2) Proses menggunakan Type data : ARRAY & RECORD & FILE TEKS
- 3) Output dari 3 model input

DOKUMENTASI :

Program Simulasi_IPO_GeneratorDatadanFile merupakan program Kreasi integrasi untuk mensimulasikan beberapa kasus pembelajaran Algoritma modular yang meliputi :

- 1) Type Data : Array - Record - File
- 2) Kreativitas :
 - a. Model Entri : Manual Keyboard - Generator Random - File Text
 - b. Model proses : Logika IPO (Input - Proses - Output)
Pemanfaatan Statistik dasar - Bilangan Random
 - c. Studi kasus : Pengelolaan Nilai Matakuliah

***** Platform :

Program diimplementasikan dengan platform :

- Bahasa Pemrograman Primitif : Pascal
- Tools : Dosbox
- Sistem Operasi : Windows based

***** INSTALASI dan RUNNING PROGRAM:

untuk dapat menjalankan program ini diperlukan langkah-langkah instalasi:

1. Instalasi terlebih Dosbox di komputer Anda
2. Copy program dalam satu folder tertentu
 - misal buat foilder D:\DEMOS
 - Copy Program ke foilder D:\DEMOS berisi
 - IPO2.EXE
 - U_IPO2.TPU
 - U_BASIC.TPU
 - FILEDATA.TXT
3. Jalankan Program
 - Jalankan Dosbox
 - lakukan mounting direktori, misal
MOUNT X D:\DEMOS
 - Pindah ke drive X --> path>X :
 - Jalankan X> IPO2
4. Selesai.

BAB 2
KARYA PRODUK PROGRAM
“APLIKASI SIMULASI METODE INTERNAL SORTING”

2.1 PROBLEM

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-
SEBELUMNYA

2.2 SOLUSI ALGORITMA

2.2.1. JUDUL/DEFINISI

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-
SEBELUMNYA

2.2.2. SKEMA KAMUS DATA

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-
SEBELUMNYA

2.2.3. SKEMA PROGRAM

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-
SEBELUMNYA

2.2.4. SKEMA MODUL

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-
SEBELUMNYA

2.2.5. SKEMA PROCEDURE/FUNCTION

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-
SEBELUMNYA

2.3 SOLUSI PROGRAM

2.3.1. DESAIN ANTARMUKA USER

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-SEBELUMNYA

2.3.2. SOURCE CODE : PROGRAM IPO

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-SEBELUMNYA

2.3.3. SOURCE CODE : UNIT XXX

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-SEBELUMNYA

2.3.4. SOURCE CODE : UNIT XXX

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-SEBELUMNYA

2.3.5. DUKUNGAN FILE DATA

FILEDATA.TXT

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-SEBELUMNYA

2.3.6. PETUNJUK SINGKAT INSTALASI

README.TXT

Catatan :

- Sengaja dokumen ini tidak disajikan untuk Belajar Mandiri & Eksplorasi
- Cara menyajikan Dokumen dapat dilihat seperti contoh BAB-SEBELUMNYA

DAFTAR PUSTAKA

CATATAN : Tuliskan Daftar Pusaka di Sini

BIODATA

NAMA : SANYATA PURWIDAYANTA, S.T., M.T.
TEMPAT LAHIR : SRAGEN, Jawa Tengah
TANGGAL LAHIR : 10 OKTOBER 1967
JENIS KELAMIN : LAKI-LAKI
AGAMA : ISLAM
GOLONGAN DARAH : O
STATUS : MENIKAH
KEWARGANEGARAAN : INDONESIA
ALAMAT TERAKHIR : KOMPLEKS PERMATA BIRU,
Blok-K No. 107 D
DESA CINUNUK, KEC. CILEUNYI,
KAB. BANDUNG



RIWAYAT PENDIDIKAN

| No | Tahun Lulus | Nomor Ijasah /nomor SK /NIM | Keterangan |
|----|-------------|--|--|
| 1. | 2000 | <ul style="list-style-type: none">235023/K01/PP.3.6.2/I/02/2001SK: 016/SK/K.01/PP.3.5.3/2001NIM : 23597033 | STRATA-2 (S2) PROGRAM STUDI TEKNIK INFORMATIKA ITB, Gelar M.T. |
| 2. | 1996 | <ul style="list-style-type: none">135020/PT07.H14/16.03/04/1996SK: 155/SK/PT07/.H/Q/1996NIM: 12587305 | STRATA-1 (S1) PROGRAM STUDI TEKNIK INFORMATIKA ITB, Gelar S.T. |
| 3. | 1987 | 03 OC oh 0083463 SK : 019/I03/H.87 | SMA NEGERI I SRAGEN |
| 4. | 1984 | 03 OB ob 0994583 - SK : 034/I03/H.84 | SMP SAVERIUS I SRAGEN |
| 5. | 1981 | 03 OA oa No 180678 - SK : 002/I 03/H.81 | SD NEGERI 5 SRAGEN |
| 6. | 1975 | - | TK. TRISULA SRAGEN |

TIGA PENGALAMAN TERAKHIR

| No | Tahun | Keterangan |
|----|-------|---|
| 1. | 2022 | Tenaga Ahli, Team Leader Pelaksana Pekerjaan ““Pengukuran Indeks Ketahanan Nasional Dan Sistem Integrasi Data”, Lemhanas RI, Jakarta. Referensi : PT. Dharma Kreasi Nusantara |
| 2. | 2021 | Tenaga Ahli, Analis & Penguji Sistem Pelaksana Pekerjaan “Implementasi TB Information System (SITB)”, Kementerian Kesehatan Republik Indonesia, Jakarta. Referensi : PT. Trimitra Sistem Solusindo (Trimitrasis) |
| 3. | 2020 | Tenaga Ahli, Analis Sistem Pelaksana Pekerjaan ”Kajian Analisis Pengelolaan Pendidikan Berbasis Teknologi Informasi di Kabupaten Bekasi”, Badan Penelitian dan Pengembangan Daerah, Pemda Bekasi. Referensi : PT. Trineka Sarana Mandiri |