**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMOGRAMAN II**

**“TIPE DATA TERSTUKTUR”**

**(ARRAY)**



**Dosen : Rini Widyastuti S.Kom., M.Kom**

**Di Susun Oleh :**

**Nama : Asyrafil Huda**

**Npm : 2010013231008**

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER**

**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN**

**UNIVERSITAS BUNG HATTA**

**2021**

**KATA PENGANTAR**

Puji syukur kami panjatkan kehadirat Allah SWT,karena berkat rahmat dan karunia serta hidayah-Nya saya dapat menyelesaikan laporan praktikum mata kuliah Algoritma dan Pemogram II.

Selama pembuatan laporan ini, penulis tidak lepas dari banyak kesulitan dalam pembuatannya. Penulis mendapat banyak masukan danbimbingan dari berbagai pihak sehingga tidak lupa penulis mengucapkan terimakasih sebesar-besarnya kepada dosen pengampu, orang tua dan keluarga yang selalu memberi motivasi dan dukungan, serta teman-teman mahasiswa Program Studi Teknik Informatika dan Komputer yang juga mengikuti mata kuliah Praktikum Algoritma dan Pemrograman II ini.

Saya berharap laporan praktikum ini bisa memenuhi salah satu syarat mata kuliah  algoiritma pemrograman II. Semoga laporan ini dapat membantu mahasiswa pada umumnya khususnya bagi penulis.

Padang, Desember 2021

Penulis

**BAB I**

**TUJUAN DAN DASAR TEORI**

1. **Tujuan Praktikum**

Setelah melakukan pratikum, diharapkan praktikan dapat:

1. Mengetahui dan mampu menerapkan pemrograman arrray yang ada dalam algoritma. Salah satunya yang berjalan dalam pascal.
2. Mengetahui dan mampu menerapkan beberapa teknik pemrograman array yang digunakan untuk menyimpan.
3. Mengetahui teknik kerja yang berbeda dari beberapa struktur pemrograman array satu dimensi.
4. Mengetahui dan mampu menerapkan penggunaan array satu dimensi (larik).
5. Mengetahui manfaat penggunaan larik.
6. **Dasar Teori**

Larik adalah tipe terstruktur yang terdiri dari sejumlah elemen – elemen yang bertipe sama. Banyaknya elemen dalam suatu larik ditunjukkan oleh suatu indeks yang harus merupakan tipe data yang menyatakan keterurutan, misalnya integer atau karakter (karena ia menyatakan posisi data). Tiap – tiap elemen di larik dapat diakses langsung melalui indeksnya. Suatu larik memiliki jumlah elemen yang jumlahnya tetap, sehingga jumlah elemen larik tidak dapat diubah selama pelaksanaan program.

**DEKLARASI ARRAY**

Deklarasi perubah selalu diawali dengan kata VAR Dalam deklarasi perubah biasa, kita hanya perlu menentukan type data perubah tersebut misalnya: real, integer, boolean atau char, sedangkan pada deklarasi larik (array), kita perlu mendefinisikan type larik, juga harus menentukan banyaknya index yang akan mengisi larik tersebut.

Syntax :

VAR nmVariabel : ARRAY [index] OF typeData

Keterangan :

nmVariabel : Nama larik yang dideklarasikan

index : Batasan/Jumlah Index (cacah elemen)

typeData : Tipe data dari Larik

Larik dapat bertipe data sederhana byte, word, integer, real, boolean, char atau string dan tipe data skalar atau subrange. Index dari larik menunjukkan maksimum banyaknya elemen-elemen dari larik. Index larik ini dapat berupa tipe subrange atau skalar apapun (integer, byte, word, char ) kecuali tipe real.

Suatu larik tidak hanya dapat berupa suatu variabel yang dideklarasikan di bagian variabel, tetapi juga dapat berupa uatu konstanta yang dideklarasikan di bagian deklarasi konstanta.

**Contoh 1.**

Var

Nama : Array[1..10] of String[20] ;

I : Byte ;

Begin

Clrscr

{Bagian Input Data} For I := 1 To 10 Do Begin

Writeln(‘Inputkan Nama Ke ’, I , ‘:’) ; Readln( Nama[I] ) ;

End;

{Bagian Menampilkan Data yang telah diinputkan } For I := 1 To 10 Do

Writeln(‘Nama Ke ’, I , ‘:’, Nama[I] ) ; Readln;

End.

**Contoh 2.**

Var

Nama : Array[1..30] of String[20] ; Gaji : Array[1..30] of Longint ; i,j : Byte ;

Begin

Clrscr ;

{Bagian Input Data}

Writeln(‘Inputkan Jumlah Data : ’) ; Readln( J ) ; For I := 1 To J Do

Begin

Writeln(‘Inputkan Nama Ke ’, I , ‘:’) ; Readln( Nama[I] ) ;

Writeln(‘Inputkan Gaji Ke ’, I , ‘:’) ; Readln( Gaji[I] ) ;

End;

Clrcsr ;

{Bagian Menampilkan Data yang telah diinputkan }

GotoXy(10, 4) ; Write(‘ ’) ; GotoXy(10, 4) ; Write(‘No. Nama Gaji’);

GotoXy(10, 6) ; Write(‘ ’) ; For I := 1 To J Do

Begin

GotoXy(10, 6+I) ; Writel( I ) ; GotoXy(14, 6+I) ; Write( Nama[I] ) ; GotoXy(34, 6+I) ; Write( Gaji[I] ) ;

End

GotoX(10, J+7) ; Writeln(‘...................................................................................................’);

Readln;

End.

**TIPE DATA RECORD (Rekaman)**

Seperti halnya array(larik), record juga punya elemen, di sini elemennya disebut "Field". Tiap elemen bisa punya type data yang berbeda. Banyaknya field dapat bervariasi atau bisa juga tetap.

Record yang punya field yang bervariasi disebut ‘Variant Record’. Deklarasi type data Record dapat ditetapkan di bagian deklarasi Var atau Type.

Syntax :

TYPE

nmPengenal = RECORD

Field1 : type1; Field2 : type2;

:

fieldn : typen;

END;

VAR

nmRec : nmPengenal ;

Keterangan :

nmPengenal : Nama Dari Record ;

Field1, Field2, Fieldn : Nama dari Variabel/Field yang terdapat dalam Record Type1, Type2, Typen : Type Dari tiap-tiap Field yang dideklarasikan

nmRec : Nama Dari Record yang di deklarasikan

Contoh :

TYPE

Mahasiswa = RECORD

NIM : String[8] ;

Nama : String[20] ;

Alamat : String[30] ;

Sex : Char ;

Umur : Byte ;

END;

VAR

RecMhs : Mahasiswa ;

**MEMBACA & MENULIS FIELD**

Dapat dilaksanakan pada seluruh record atau sebagian record.

Syntax :

nmRec.Field1 ;

Tanda titik harus ditulis

Notasi ini disebut Penandaan field (‘field designator’).

Berbeda dengan statement pemberian (assigment), maka untuk membaca dan menulis record selalu dilakukan menurut Field yang ada, tidak dilakukan secara keseluruhan record.

Contoh : Membaca Readln(RecMhs.NIM );

Readln(RecMhs.Nama ); Menulis Writeln(RecMhs.NIM );

Writeln(RecMhs.Nama );

**Contoh 3 :**

TYPE

Mahasiswa = RECORD

NIM : String[8] ; Nama : String[20] ; Umur : Byte ;

END;

Var

RecMhs : Mahasiswa ;

Begin

Clrscr ;

{Bagian Input Data}

Writeln(‘NIM Anda :’); readln( RecMhs.NIM ) ; Writeln(‘Nama :’); readln( RecMhs.Nama ) ; Writeln(‘Umur :’); readln( RecMhs.Umur ) ;

{Bagian Menampilkan Data yang telah diinputkan } Writeln(‘NIM Anda :’ , RecMhs.NIM ) ; Writeln(‘Nama :’ , RecMhs.Nama ) ; Writeln(‘Umur :’ , RecMhs.Umur ) ;

Readln;

End.

**STATEMENT WITH**

Digunakan untuk mengurangi penulisan yang berulang-ulang untuk nama recordnya. Maka dengan menggunakan statement WITH hanya cukup sekali saja menyebutkan nama recordnya.

**WITH** nmRecord **DO**

Syntax :

**Contoh 4 :**

TYPE

Mahasiswa = RECORD

NIM : String[8] ; Nama : String[20] ; Umur : Byte ;

END;

Var

RecMhs : Mahasiswa ;

Begin

Clrscr ;

With RecMhs Do Begin

{Bagian Input Data}

Writeln(‘NIM Anda :’); readln(NIM ) ; Writeln(‘Nama :’); readln(Nama ) ; Writeln(‘Umur :’); readln(Umur ) ;

{Bagian Menampilkan Data yang telah diinputkan }

Writeln(‘NIM Anda :’ , NIM ) ;

Writeln(‘Nama :’ , Nama ) ;

Writeln(‘Umur :’ , Umur ) ;

End;

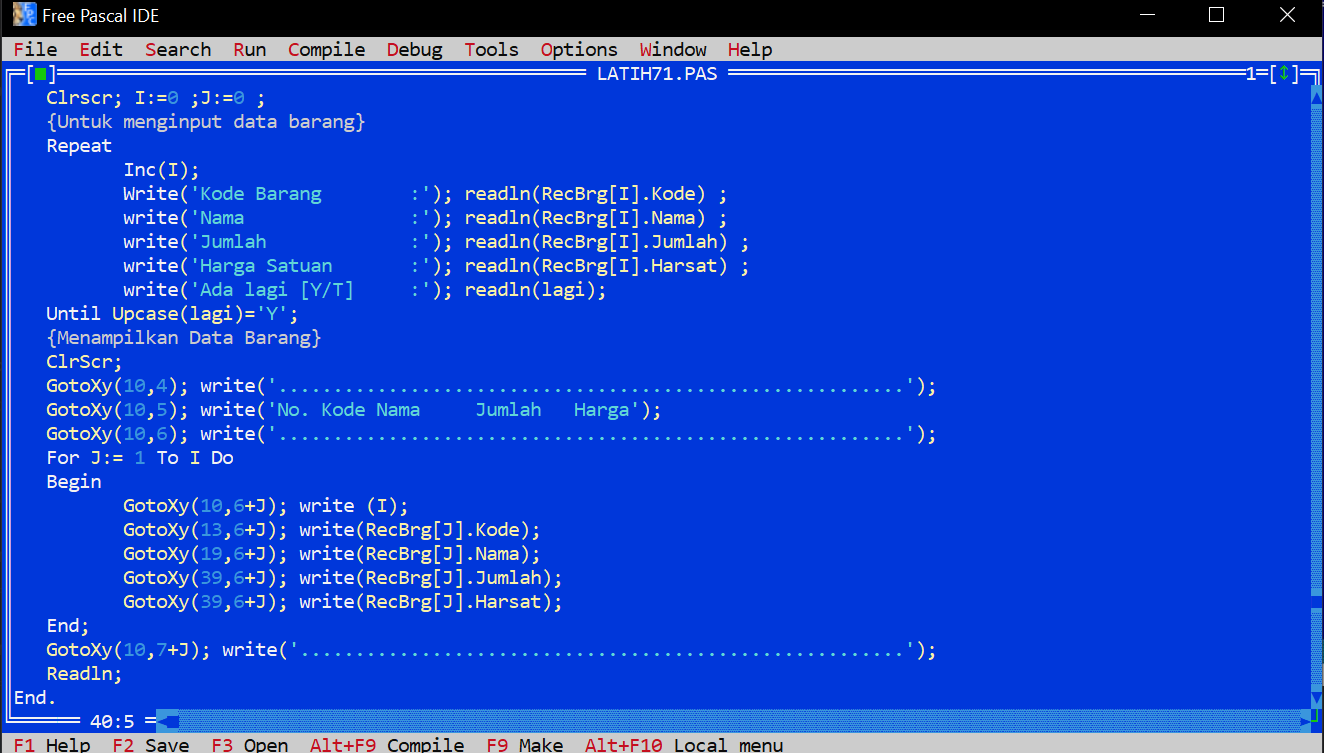
Readln;

End.

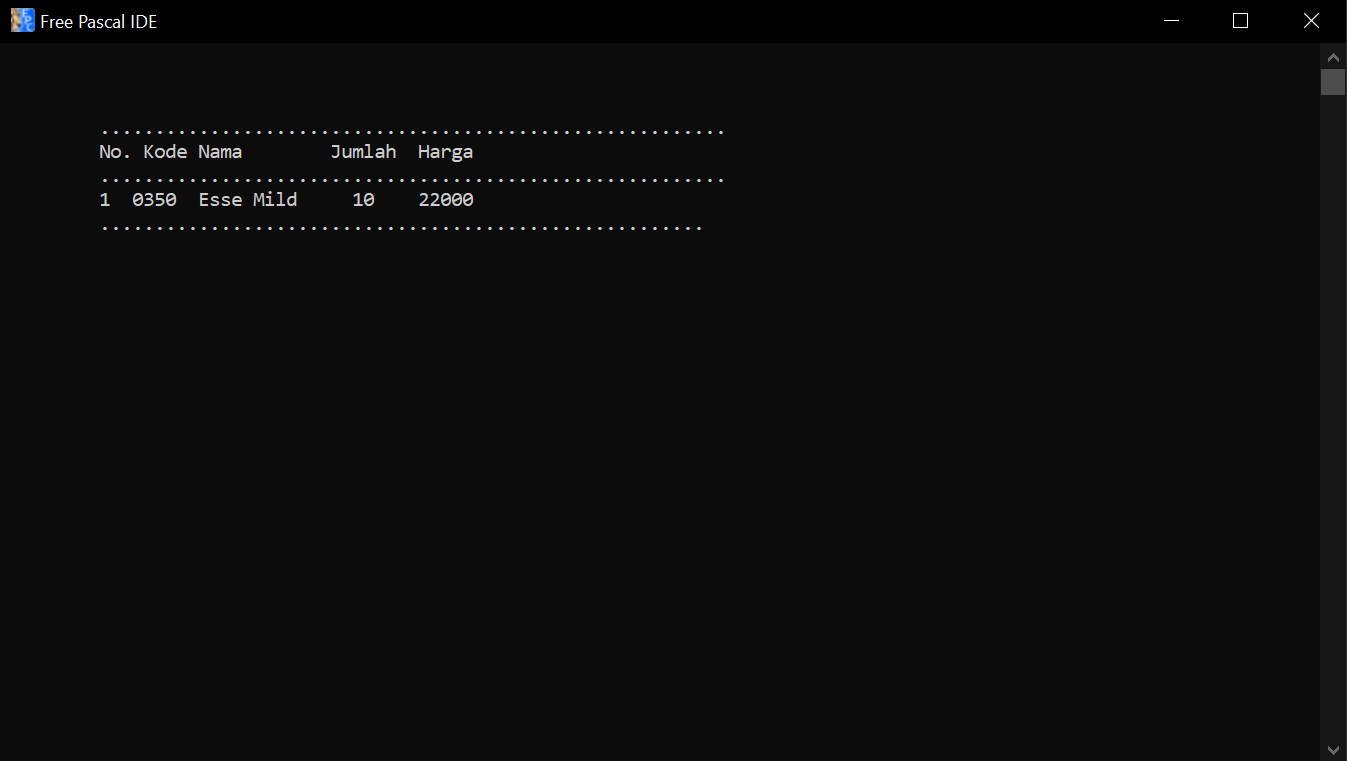
**BAB II**

**HASIL PENGAMATAN DAN ANALISA HASIL PENGAMATAN**

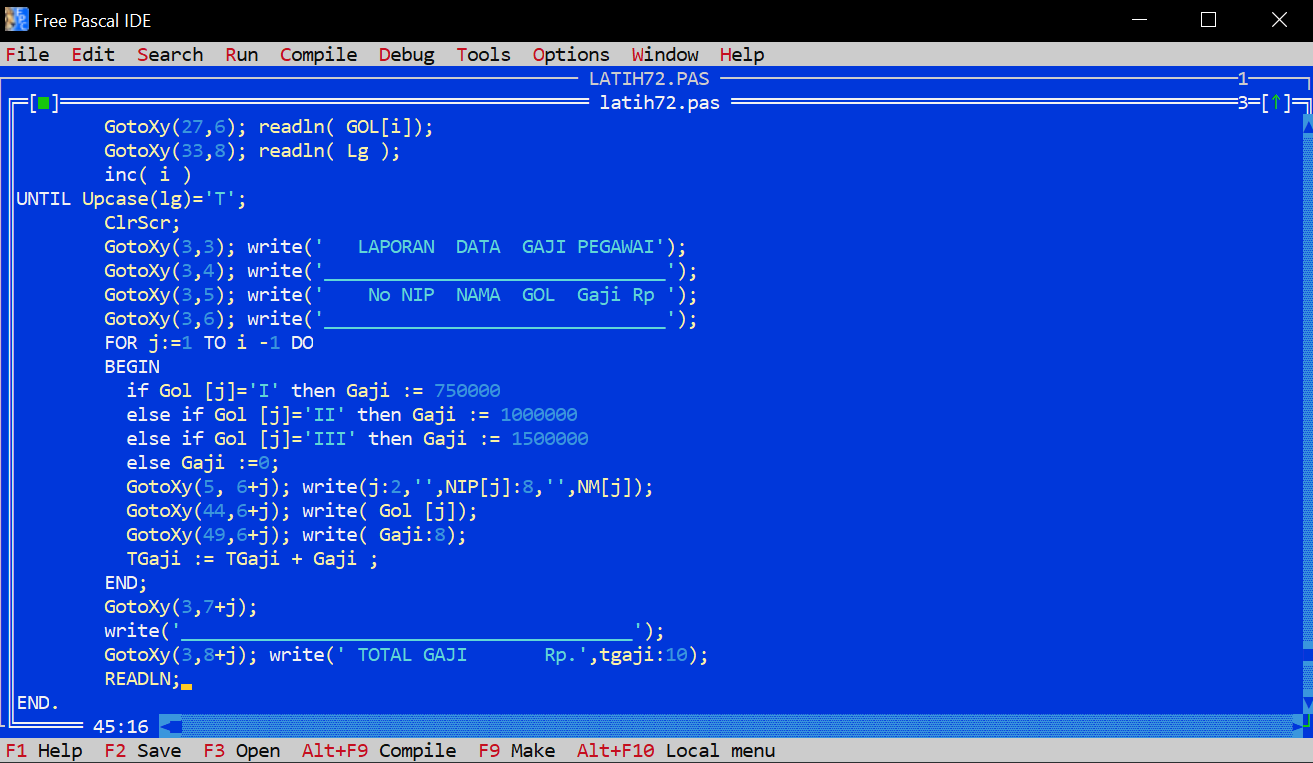
1. **Hasil Pengamatan**
2. Listing Program



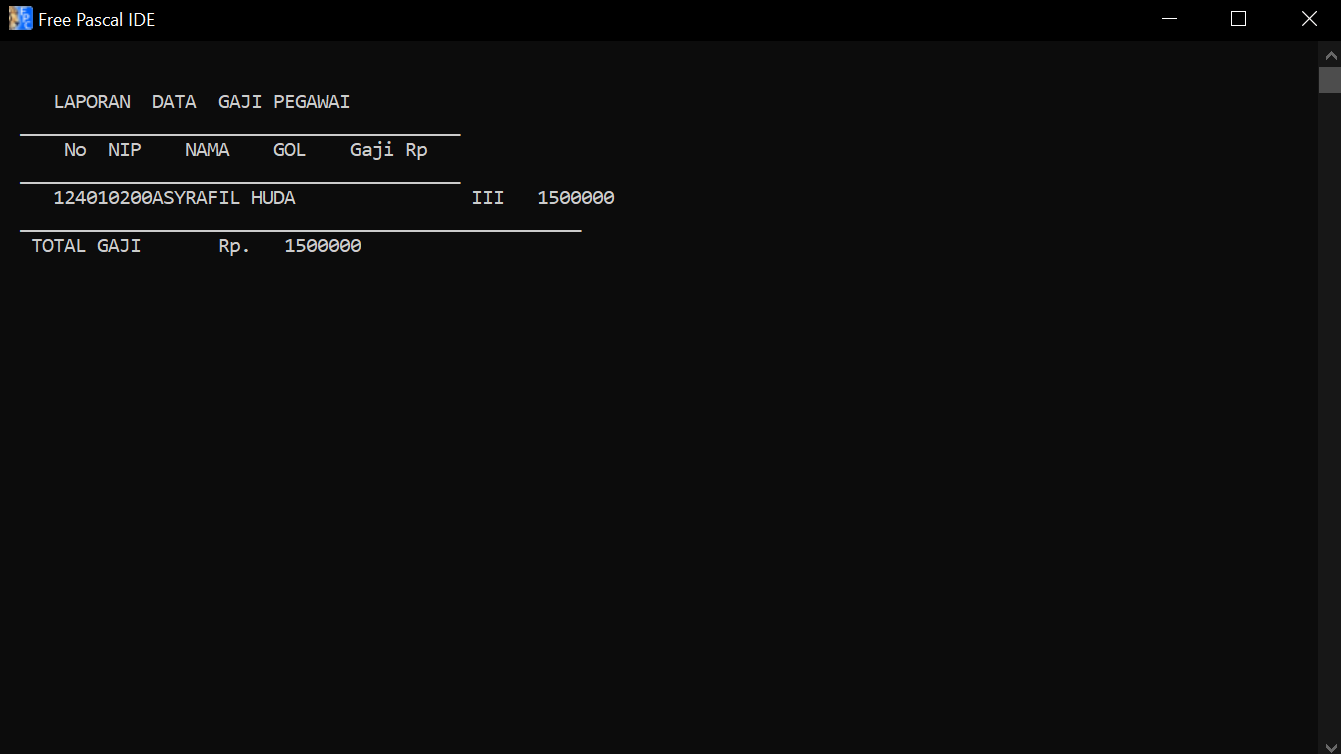
Hasil Eksekusi



1. Listing Program

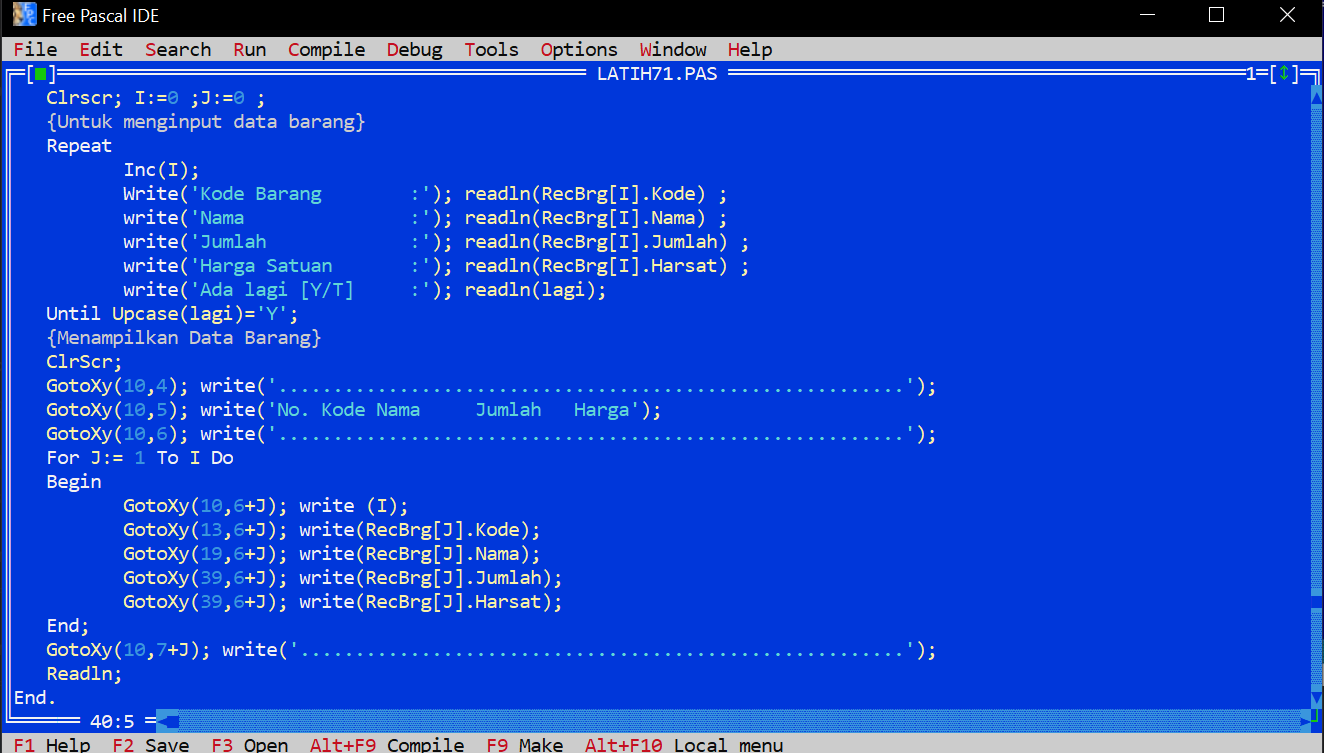


Hasil Eksekusi

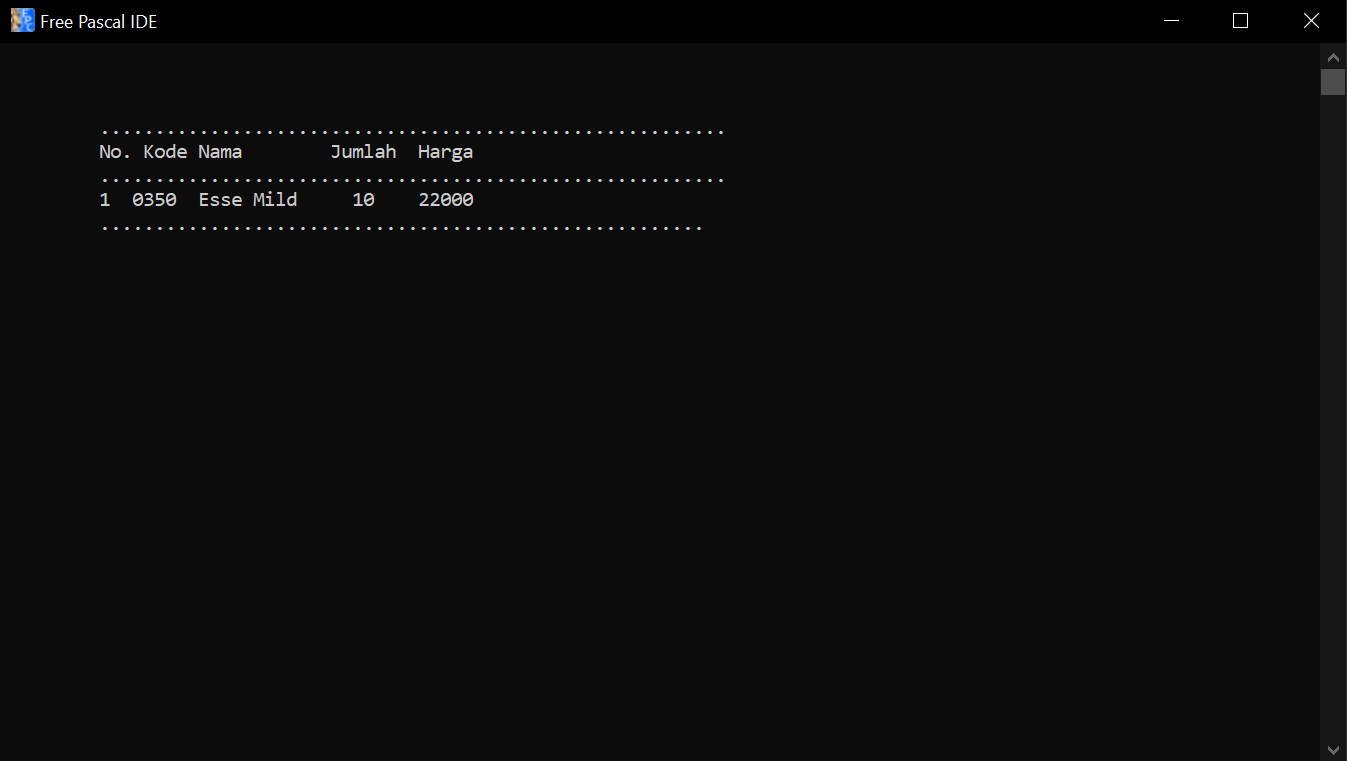


1. **Analisa Hasil Pengamatan**
2. Analisa hasil pengamatan dari Program luas persegi panjang

Program



Output

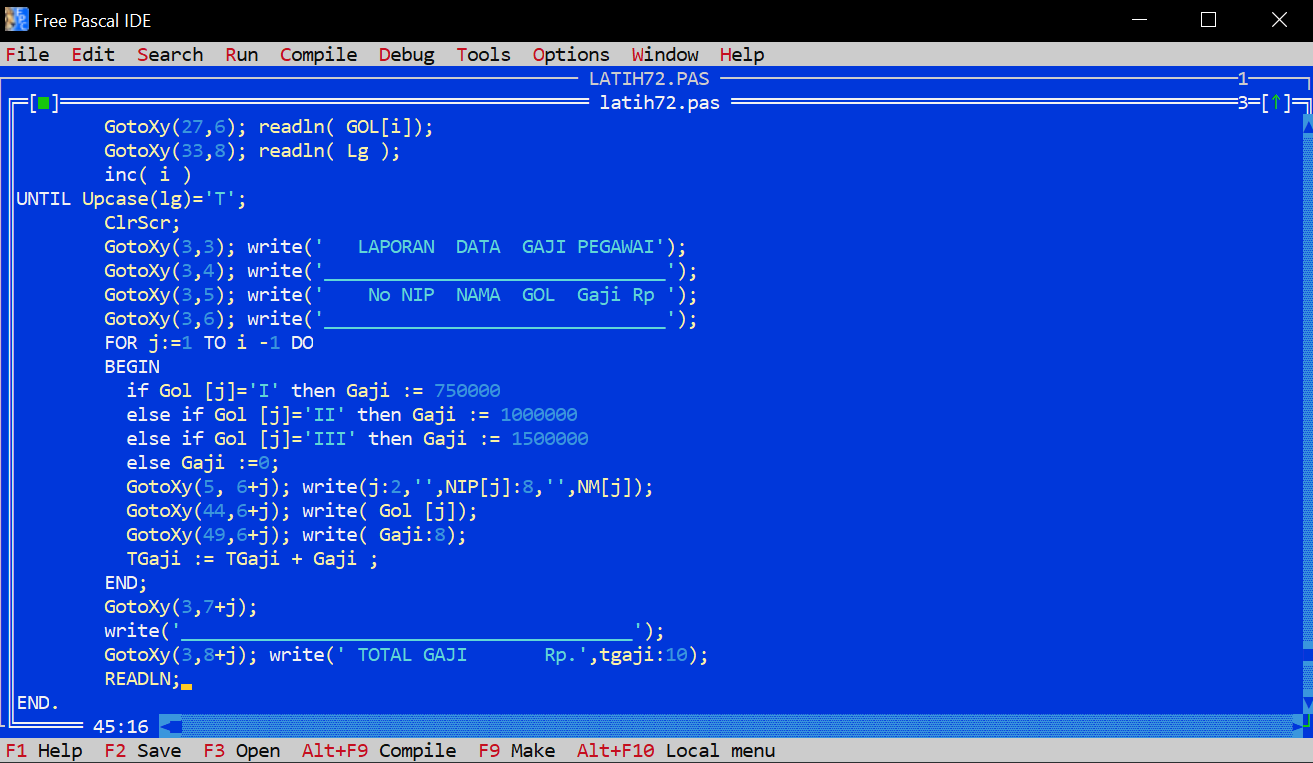


Penjelasan Program :

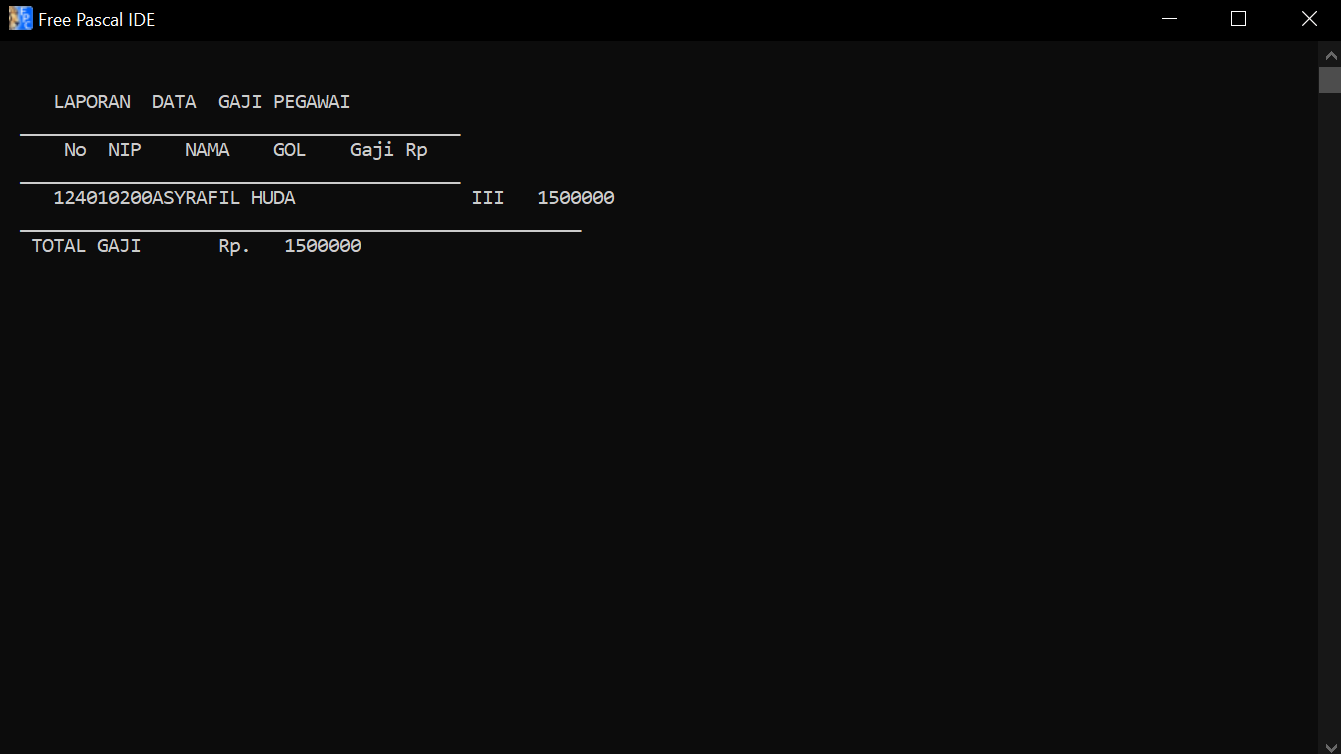
Program di atas adalah untuk menginput data dan menampilkannya. Program ini merupakan program penjualan yg berfungsi menginput data berupa no, kode, nama,jumlah dan harga barang, dan selanjutnya akan ditampilkan seperti pada gambar output di atas

1. Analisa hasil pengamatan Program ulang 1

Program



Output



Penjelasan Program :

Program di atas adalah membuat laporan data gaji pegawai menurut golongannya. Pada program ini menginputkan data berupa nip, nama, gol.

**BAB III**

**KESIMPULAN**

1. Array adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Nilai-nilai data di suatu array disebut dengan elemen-elemen array. Letak urutan dari elemen-elemen array ditunjukkan oleh suatu subscript atau indeks.
2. Untuk array satu dimensi:

a. Setiap elemen array dapat diakses melalui indeks

b. Indeks array secara default dimulai dari 0.

c. Deklarasi array dalam bentuk umum: Tipe\_array nama\_array[ukuran]; Contoh : int Nilai [4].

1. Untuk mengisi dan menampilkan isi elemen array ada dua cara yaitu:

a. Row Major Order (secara baris per baris)

b. Column Major Order (secara kolom per kolom)

1. Bentuknya dapat berupa matriks atau tabel. Bentuk Umum: Tipenama\_array[baris] [kolom];

**BAB IV**

**DAFTAR PUSTAKA**

1. <https://www.termasmedia.com/pemrograman/java/123-pengertian-dan-deklarasi-larik-array-di-java.html>
2. <https://ekanursafitriblog.wordpress.com/2018/04/08/array-larik-beserta-contoh-program/>
3. <https://www.ilmuit.id/2020/03/algoritma-2-pembahasan-tentang-larik.html>