LAPORAN TUGAS PEKAN 5
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
"BELAH KETUPAT"

Dosen Pengampu:

DR. Wahyudi, S.T., M.T.

Asisten Lab:

Muhammad Zaki Al Hafiz

Disusun Oleh:

Hamdi Sidqi Alifi

2511531009

Fakultas Teknologi Informasi

Departemen Informatika

Universitas Andalas

2025

KATA PENGANTAR

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi penugasan pasca-pratikum mata kuliah Algoritma Pemorograman pekan ke-5 yang dilaksanakan pada hari Rabu tanggal 29 Oktober 2025.

Pada kesempatan ini, Penulis ingin mengucapkan terima kasih kepada Bapak DR. Wahyudi. .S.T.M.T, dosen pengampu mata kuliah Algoritma Pemograman, yang telah memberikan tugas ini. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan tugas Algoritma Pemrograman ini.

Penulis sadar bahwa laporan ini masih memiliki beberapa kekurangan dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis harap adanya bentuk saran dan kritik yang membangun dari berbagai pihak.

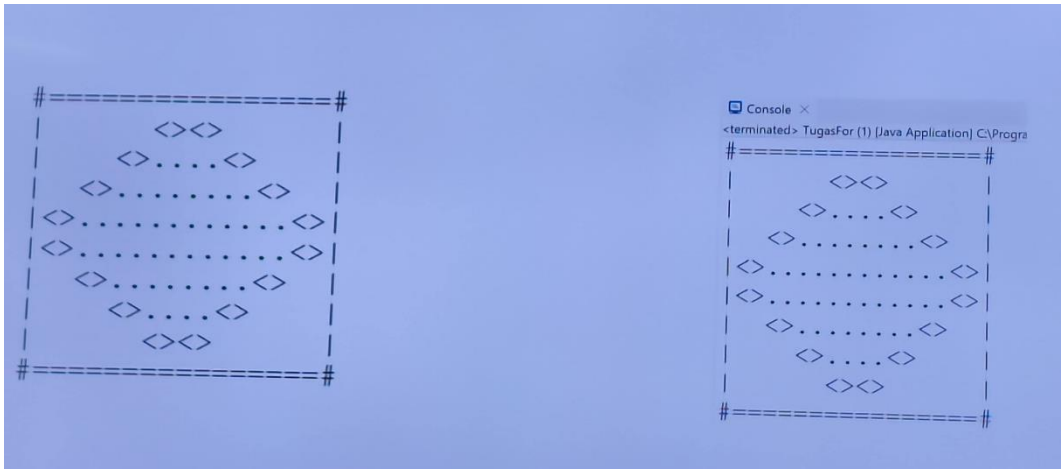
Akhir kata, penulis berharap laporan ini dapat bermanfaat bagi perkembangan dunia pendidikan.

DAFTAR ISI

BAB I
EKSPLORASI

1.1 Latar Belakang

Pada pekan ke-5 kita telah mempelajari tentang perulangan *for*. Untuk lebih memahami penggunaan fungsi *for-loop*, maka diberikan sebuah hasil output, yang mana mahasiswa Praktikum Algoritma dan Pemograman (Alpro) diharapkan dapat mereplikasi program belah ketupat yang telah dipaparkan pada monitor.



Gambar I.1 Hasil yang diinginkan

1.2 Tujuan

Mereplika program belah ketupat di dalam *Java* menggunakan fungsi *for-loop.*

1.3 Manfaat Pratikum

Memahami konsep perulangan for-loop dalam bahasa pemrograman Java secara lebih kompleks.

BAB II
RANCANGAN

2.1 Analisis

Dari perolehan gambar I.1 dapat kita transkripsikan menjadi :

```
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

Berdasarkan gambar, terdapat 10 baris dan 18 kolom, dimana baris pertama dan terakhir terlihat seperti bingkai/batas, dan selebihnya memiliki tepi “|” saja. Di tengah tengah terdapat sebuah gambar yang membentuk belah ketupat. Belah ketupat itu terdiri dari 8 baris dan 4 baris pertama simetris dengan 4 baris terakhir.

BAB III
PROGRAM

3.1 Bahasa Natural

1) Inisialisasikan lebar = 16, baris = 4, h = 1, i, titik, selisih, kosong, p, d
2) Bandingkan h dengan nilai variabel lebar yang telah ditambah dengan dua, apakah h kecil sama dari nilai tersebut?
    a. Jika ya, maka tentukan apakah h bernilai 1 atau (lebar + 2)
        i. Jika ya, maka cetak "#", naikkan nilai h satu kali, ulangi nomor (2).
        ii. Jika tidak, maka cetak "=", naikkan nilai h satu kali, ulangi nomor (2).
    b. Jika tidak, lanjut nomor (3).
3) Cetak baris baru.
4) Atur nilai i menjadi nol.
5) Bandingkan i dengan baris, apakah i kecil dari nilai variabel baris?
    a. Jika ya :
        i. Atur nilai titik menjadi i kali 4
        ii. Atur nilai selisih menjadi 4 tambah titik
        iii. Atur nilai kosong menjadi (lebar – selisih) bagi dua
        iv. Cetak "|"
        v. Atur nilai p menjadi = 0
        vi. Bandingkan p dengan nilai variabel kosong, apakah p kecil dari nilai variabel kosong?
            - Jika ya, maka cetak " ", naikkan nilai p satu kali, ulangi langkah poin (vi)
            - Jika tidak, maka lanjut ke poin (vii)
        vii. Cetak "<>"
        viii. Atur nilai d menjadi = 0
        ix. Bandingkan d dengan nilai variabel titik, apakah d kecil dari nilai variabel titik?
            - Jika ya, maka cetak ".", naikkan nilai d satu kali, ulangi langkah poin (ix)

- Jika tidak, maka lanjut ke poin (x)

x. Cetak “<>”

xi. Atur nilai p menjadi = 0.

xii. Bandingkan p dengan nilai variabel kosong, apakah p kecil dari nilai variabel kosong?

- Jika ya, maka cetak “ “, naikkan nilai p satu kali, ulangi langkah poin (xii)
- Jika tidak, maka lanjut ke poin (xiii)

xiii. Cetak “|”, dan cetak baris baru

xiv. Naikkan nilai i satu kali. Ulangi langkah nomor (5).

b. Jika tidak, maka lanjut nomor (6).

6) Atur nilai i menjadi nilai variabel baris yang dikurangi satu.

7) Bandingkan i dengan baris, apakah i besar sama dari nol?

a. Jika ya :

i. Atur nilai titik menjadi i kali 4

ii. Atur nilai selisih menjadi 4 tambah titik

iii. Atur nilai kosong menjadi (lebar – selisih) bagi dua

iv. Cetak “|”

v. Atur nilai p menjadi = 0

vi. Bandingkan p dengan nilai variabel kosong, apakah p kecil dari nilai variabel kosong?

- Jika ya, maka cetak “ “, naikkan nilai p satu kali, ulangi langkah poin (vi)
- Jika tidak, maka lanjut ke poin (vii)

vii. Cetak “<>”

viii. Atur nilai d menjadi = 0

ix. Bandingkan d dengan nilai variabel titik, apakah d kecil dari nilai variabel titik?

- Jika ya, maka cetak “.“, naikkan nilai d satu kali, ulangi langkah poin (ix)
- Jika tidak, maka lanjut ke poin (x)

x. Cetak “<>”

xi. Atur nilai p menjadi = 0.

xii. Bandingkan p dengan nilai variabel kosong, apakah p kecil dari nilai variabel kosong?

- Jika ya, maka cetak " ", naikkan nilai p satu kali, ulangi langkah poin (xii)
- Jika tidak, maka lanjut ke poin (xiii)

xiii. Cetak "|", dan cetak baris baru

xiv. Naikkan nilai i satu kali. Ulangi langkah nomor (7).

b. Jika tidak, maka lanjut nomor (8).

8) Atur nilia h menjadi satu.

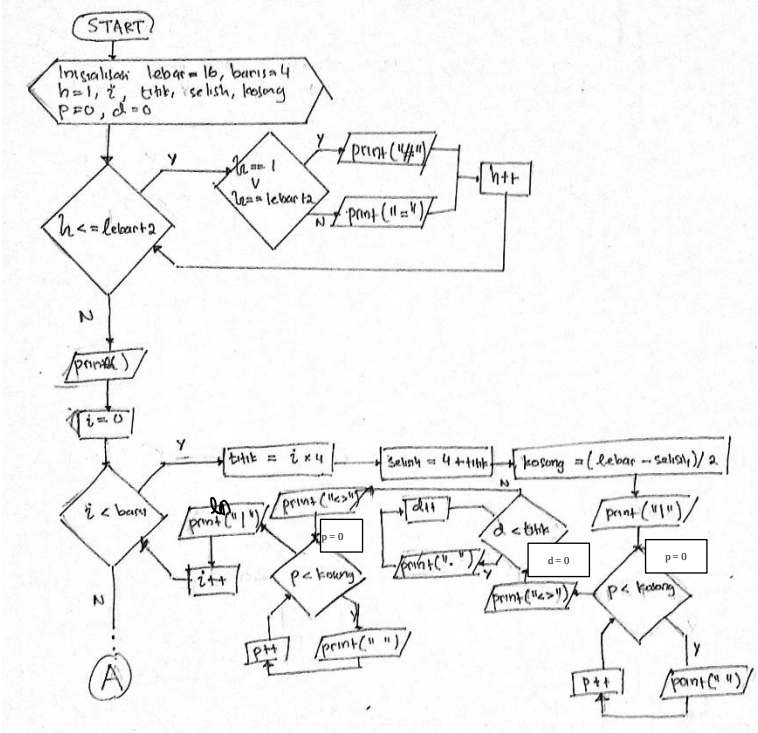
9) Bandingkan h dengan nilai variabel lebar yang telah ditambah dengan dua, apakah h kecil sama dari nilai tersebut?

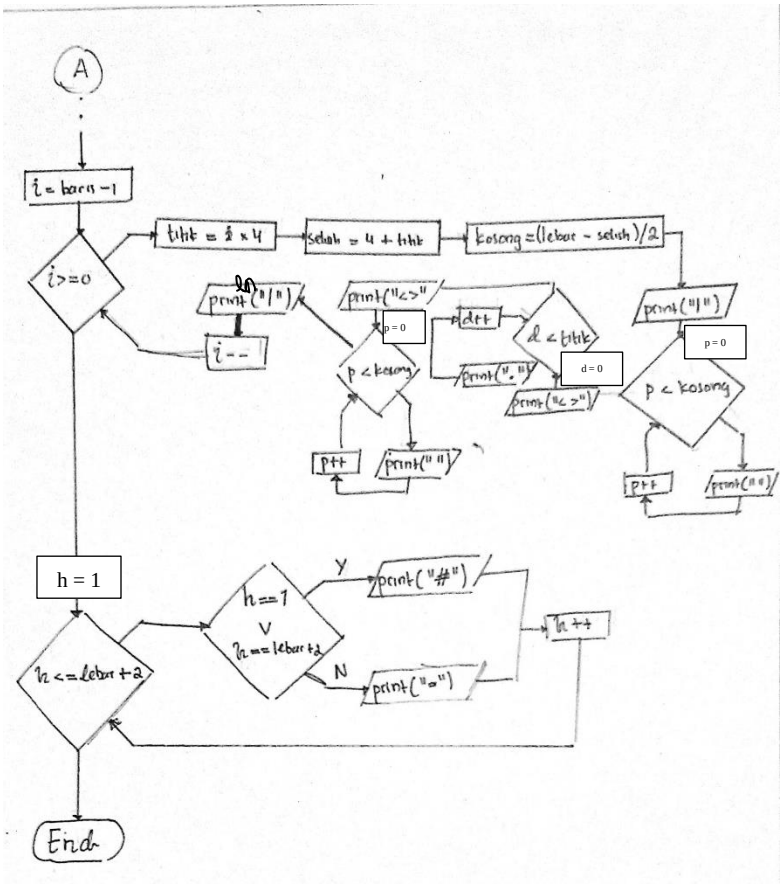
c. Jika ya, maka tentukan apakah h bernilai 1 atau (lebar + 2)

iii. Jika ya, maka cetak "#", naikkan nilai h satu kali, ulangi nomor (9).

iv. Jika tidak, maka cetak "=", naikkan nilai h satu kali, ulangi nomor (9).

3.2 Flowchart





3.3 Pseudo-code

Judul Program Belah Ketupat {menciptakan suatu struktur yang membentuk bangunan belah ketupat dengan simbol-simbol yang telah ditetapkan}
Deklarasi int baris = 4, lebar = baris * 4, h = 1, i, titik, selisih, kosong, p = 0, d = 0
Algoritma dan Logika 1. For (h = 1; h <= (lebar + 2); h++) 2. If (h == 1 || h == (lebar + 2) 3. Print(#) 4. Else 5. Print(=) 6. End for 7. Println 8. For (i = 0; I < baris; i++) 9. Titik = i * 4 10. Selisih = 4 + titik 11. Kosong = (lebar – selisih) / 2 12. Print(|)

```
13.         For (p = 0; p < kosong; p++)
14.         Print( )
15.         End for
16.         Print(<>)
17.         For (d = 0; d < titik; d++)
18.         Print(.)
19.         End for
20.         Print(<>)
21.         For (p=0;p<kosong;p++)
22.         Print( )
23.         End for
24.         Print(|)
25.   For (i = baris - 1; I > 0; i--)
26.         Titik      = i * 4
27.         Selisih    = 4 + titik
28.         Kosong  = (lebar – selisih) / 2
29.         Print(|)
30.         For (p = 0; p < kosong; p++)
31.         Print( )
32.         End for
33.         Print(<>)
34.         For (d = 0; d < titik; d++)
35.         Print(.)
36.         End for
37.         Print(<>)
38.         For (p=0;p<kosong;p++)
39.         Print( )
40.         End for
41.         Print(|)
42.   For (h = 1; h <= (lebar + 2); h++)
43.         If (h == 1 || h == (lebar + 2)
44.                   Print(#)
45.         Else
46.                   Print(=)
47.   End for
```

3.4 Java language

Pembahasan *Java language* untuk program belah ketupat ini akan penulis bagi menjadi beberapa segmen

```java
package pekan5;

public class tugasBelahKetupat_2511531009 {

    public static void main(String[] args) {
        /* Dari contoh yang diberikan, terli
         * 16 dari 18 kolom tersebut menjadi
         * dan 4 baris atas dari 10 baris itu
         * Oleh karena itu...
         */
        int baris = 4;
        int lebar = baris * 4;

```

Pada line ke (11) kita menggunakan variabel baris dengan *value* 4, dan variabel lebar dengan *value* kelipatan 4 dari nilai baris. Baris disini bernilai 4 karena belah ketupat memiliki 8 baris dan setengahnya merupakan hasil *mirror*/pencerminan dari setengah lainnya. Lebar memiliki kelipatan 4 dari baris, disini penulis mengambil lebar hanya 16, hal ini akan kita bahas di bagian selanjutnya.

```java

        // Bingkai atas
        for (int h = 1; h <= (lebar + 2); h++) {
            if (h == 1 || h == (lebar + 2)) {
                System.out.print("#");
            }
            else {
                System.out.print("=");
            }
        }
        System.out.println();

```

Ini merupakan segmen bingkai atas. Sesuai namanya, kodingan ini bertujuan untuk menciptakan "`#================#`". Alasan penulis memisahkan bingkai atas dan bingkai bawah dengan fungsi perulangan *for* yang berbeda adalah dikarenakan demi menghemat line, jika bingkai atas dan bawah berada dalam satu fungsi *loop* maka akan menjadi lebih rumit. Kemudian, batas kondisional `h <= (lebar + 2)` bertujuan agar bentuk belah ketupat dapat disesuaikan nantinya, yang akan penulis bahas nanti.

Perhatikan pada *line* (15), disini penulis meng*import* fungsi *for* dan menginisialisasikan variabel h dengan nilai 1 untuk membentuk barisan bingkai. Di dalam fungsi *for-loop* terdapat fungsi *if-else*, yang mana disyaratkan dengan kondisi nilai h harus yang pertama atau yang terakhir. Jika ya, maka mesin akan memprogram "#" dan jika tidak maka mesin akan memprogram "=". Setelah itu program akan mengiterasikan nilai h dengan naik 1 nilai secara positif, hal ini akan terus berulag hingga nilai h mencapai kolom terakir (di dalam konteks ini kolom terakhir bernilai 18).

Kemudian pada line (23) penulis menyatakan fungsi cetak baris tanpa memasukkan value apapun, hal in bertujuan agar program nantinya membuat baris baru demi melanjutkan pembuatan belah ketupat tersebut.

```
24
25          // Setengah ketupat atas
26          for (int i = 0; i < baris; i++) {
27              int titik = i * 4; // 0, 4, 8, 12
28              int selisih = 4 + titik; // "<>" + titik +
29              int kosong = (lebar - selisih) / 2;
30
31              System.out.print("|");
32              for (int p = 0; p < kosong; p++) {
33                  System.out.print(" ");
34              }
35              System.out.print("<>");
36              for (int d = 0; d < titik; d++) {
37                  System.out.print(".");
38              }
39              System.out.print("<>");
40              for (int p = 0; p < kosong; p++) {
41                  System.out.print(" ");
42              }
43              System.out.println("|");
44          }
45
```

Pada segmen setengah belah ketupat atas ini, karena fungsi *for* dengan variabel h tadi sudah penulis tutup, maka kita harus membuat fungsi *for* dengan inisialisasi yang baru, kali ini penulis memberikan variabel i, sebagai penanda baris.

Perhatikan pada *line* (26), penulis menginisialisasikan variabel i dengan bilangan 0, mengapa demikian? Karena pada *line* (27), penulis mendeklarasikan variabel titik dengan value i dikali dengan 4, variable ini

menunjukkan berapa jumlah titik yang harus di *loop* oleh fungsi *for* nantinya, mengapa harus 4? Karena **"<>"** (mari kita sebut saja *diamond*) mengenakan 2 ruang sehingga 2 *diamond* akan memakan 4 ruang, dan karena posisi yang sebelumnya ditempati oleh 2 *diamond* pada baris pertama (dihitung setelah bingkai) akan digantikan menjadi titik pada baris kedua. Kemudian begitu seterusnya hingga baris keempat. Baris kelima hingga kedelapan memiliki sistem yang sama, hanya saja prosesnya dibuat berlawanan (*invers*) demi membentuk sebuah belah ketupat.

Kemudian untuk saat ini, mari kita ambil nilai titik pada i = 3.

$$i \rightarrow 3 \Rightarrow titik_{atas} = (3) * 4$$

$$titik_{atas} = 12 \ldots (pers.\ 1)$$

```
45
46          // Setengah ketupat bawah (kebalikan dari se
47          for (int i = baris - 1; i >= 0; i--) {
48              int titik = i * 4;
49              int selisih = 4 + titik;
50              int kosong = (lebar - selisih) / 2;
51
52              System.out.print("|");
53              for (int p = 0; p < kosong; p++) {
54                  System.out.print(" ");
55              }
56              System.out.print("<>");
57              for (int d = 0; d < titik; d++) {
58                  System.out.print(".");
59              }
60              System.out.print("<>");
61              for (int p = 0; p < kosong; p++) {
62                  System.out.print(" ");
63              }
64              System.out.println("|");
65          }
```

Ini adalah segmen lanjutan untuk menyempurnakan bentuk belah ketupat yang kita inginkan, dan algoritmanya sama seperti setengah sebelumnya. Namun, perhatikan inisialisasi pada *line* (47), disini penulis menyatakan i = baris – 1, berdasarkan contoh yang kita miliki, yaitu nilai variabel baris adalah 4 sehingga untuk nilai i pertama didapatkan :

$$i = baris - 1$$

$$= 4 - 1$$

$$= 3$$

Sehingga nilai titik adalah :

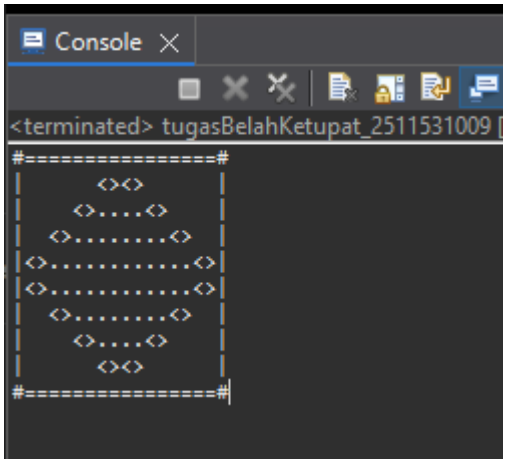
$$titik_{bawah} = i * 4$$
$$titik_{bawah} = (3) * 4$$
$$titik_{bawah} = 12 \ldots (pers.2)$$

Dapat dilihat bahwa $(pers.1) = (pers.2)$ sehingga dapat dipastikan bahwa baris ke-4 memiliki hasil yang sama dengan baris ke-5. Hal ini menandakan bahwa kesimetrisan belah ketupat tersebut dapat membantu kita menghemat *line* kodingan kita.

```
66
67              // Bingkai bawah
68              for (int h = 1; h <= (lebar + 2); h++) {
69                  if (h == 1 || h == (lebar + 2)) {
70                      System.out.print("#");
71                  }
72                  else {
73                      System.out.print("=");
74                  }
75              }
76      }
77
78  }
79
```

Segmen bingkai bawah ini memiliki kodingan yang sama saja dengan segmen bingkai atas hanya saja setelah fungsi *for* berakhir, penulis tidak menambahkan *fungsi* `System.out.println();` karena penulis tidak ada gunanya.

Apabila kodingan tersebut telah ditulis dengan benar, maka hasil yang akan keluar adalah :



Dapat dilihat bahwa hasil yang keluar sesuai dengan yang diharapkan.

BAB IV
EVALUASI

4.1 ADJUSTABLE SIZE WITH FIXED RATIO (UKURAN YANG DAPAT DI SESUAIKAN DENGAN RASIO TETAP)

Tadi penulis ada membahas tentang keunikan dari kodingan ini, seperti yang telah kita ketahui bahwa ukuran panjang dan lebar dari belah ketupat tersebut adalah 10×18, dan disini penulis memprogram dengan lebar 16 dan baris 4 (karena simetris atas bawah). Jadi, kita sebenarnya bisa menyesuaikan bentuk belah ketupatnya dengan apapun yang kita inginkan, dengan rasio :

$$baris : lebar$$

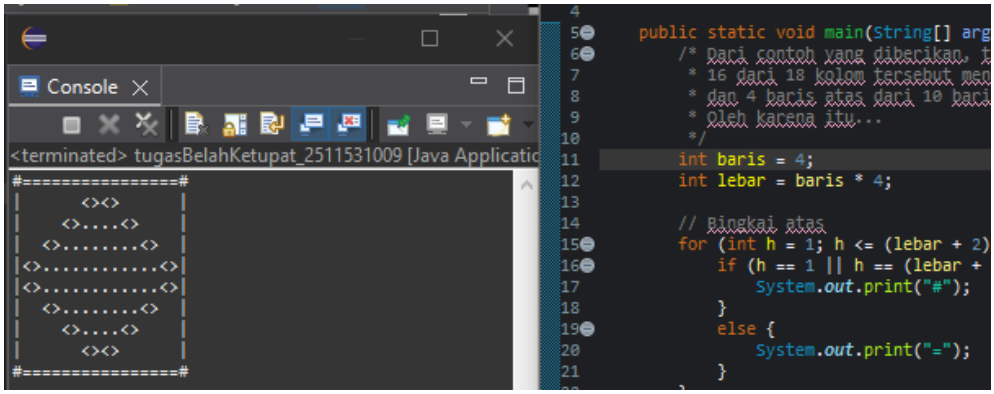
$$4 : 16$$

$$4 : 4 * 4$$

$$1 : 4$$

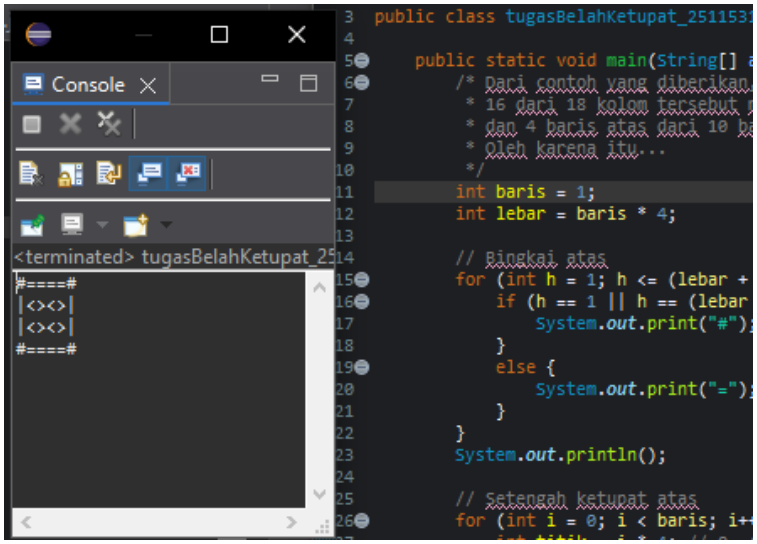
$$\therefore baris : 4 * baris$$

Contohnya saja, jika yang dijadikan sebagai tugas adalah sebagai berikut :

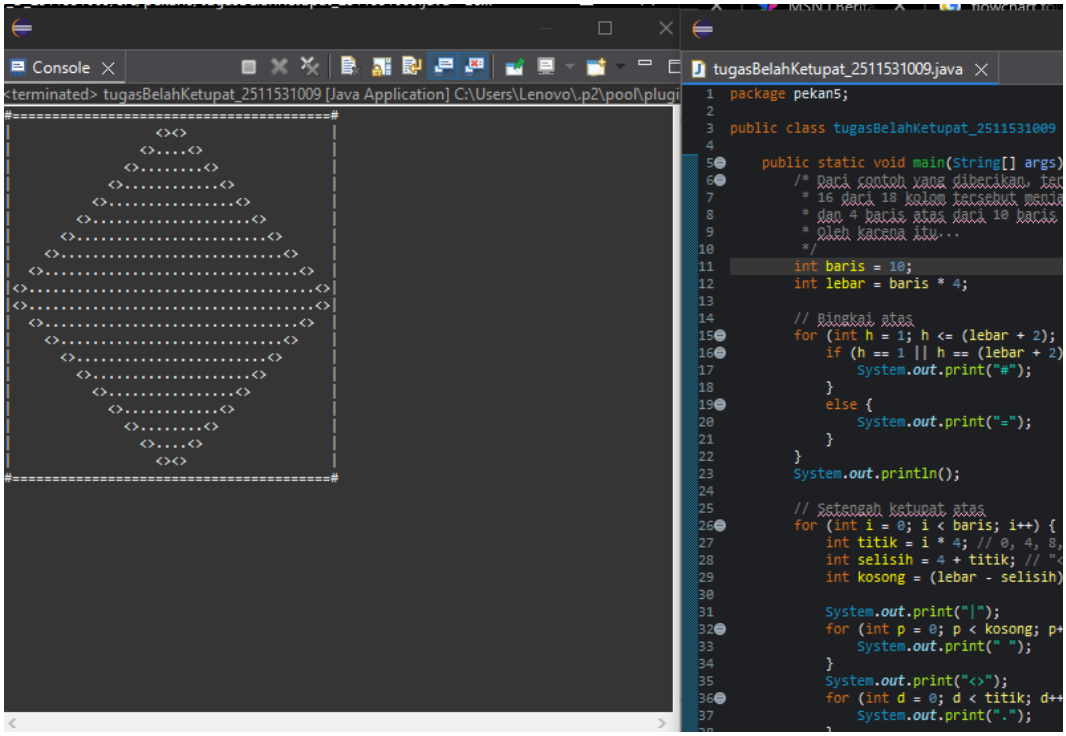


Dapat dilihat bahwa pada *line* (11), penulis menaruh angka 4 untuk nilai variabel baris, dan nilai baris * 4 untuk variabel bebas. Sehingga kita bisa meng*custom* bentuk sesuai keinginan kita, seperti sebagai berikut

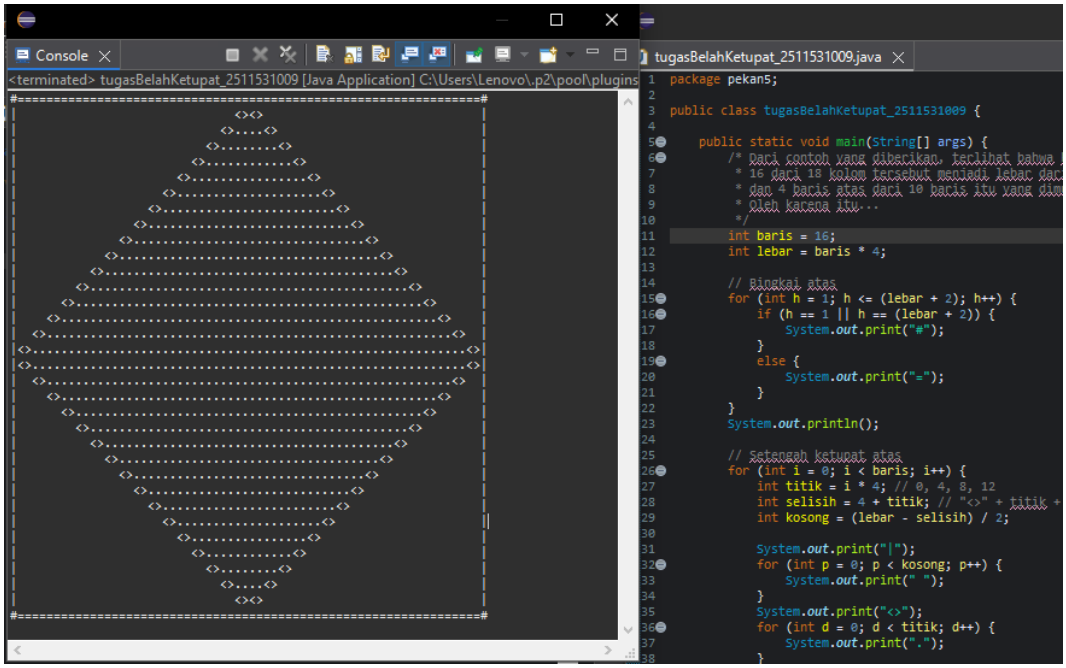
Rasio 1 : 4



Rasio 10 : 40



Rasio 16 : 64



Hal ini bisa kita kembangkan lagi dengan meng*import* salah satu utilitas java yaitu Scanner, dan membuatnya menjadi suatu aplikasi.