

1. Buat program untuk menghitung **expexted value** dan **variance** dari soal berikut ini

Diketahui Random Variabel sebagai berikut

X	1	2	3	4	5
P(X)	0,16	0,22	0,28	0,20	0,14

Dengan output sebagai berikut

Expexted value

x	$P(x)$	$xP(x)$
1	0.16	$1(0.16) = 0.16$
2	0.22	$2(0.22) = 0.44$
3	0.28	$3(0.28) = 0.84$
4	0.20	$4(0.20) = 0.80$
5	0.14	$5(0.14) = 0.70$
$\Sigma P(x) = 1$		$\Sigma xP(x) = 2.94 \approx 2.9$

Expexted value $E(x) = 2,9$

variance

x	$P(x)$	$x - \mu$	$(x - \mu)^2$	$(x - \mu)^2 P(x)$
1	0.16	-1.94	3.7636	0.602176
2	0.22	-0.94	0.8836	0.194392
3	0.28	0.06	0.0036	0.001008
4	0.20	1.06	1.1236	0.224720
5	0.14	2.06	4.2436	0.594104
$\Sigma P(x) = 1$				$\Sigma (x - \mu)^2 P(x) = 1.6164$

$$\text{Variance} = \sigma = \sqrt{1,6164} = 1,3 \text{ *Salah}$$

$$\text{Variance} = \sigma = \sqrt{1,6164} = 1,27$$

Penyelesaian:

A. Code

```
Stochastic Version control
CalculatorRandomVariable.java
1 package number1;
2
3 import java.text.DecimalFormat;
4
5 public class CalculatorRandomVariable {
6     public static void main(String[] args) {
7         // 1. Data Input (berdasarkan soal)
8         double[] xValues = {1.0, 2.0, 3.0, 4.0, 5.0}; // Nilai-nilai X
9         double[] pValues = {0.16, 0.22, 0.28, 0.20, 0.14}; // Probabilitas P(X)
10
11         // Cek apakah jumlah probabilitas = 1 (opsional tapi bagus untuk validasi)
12         double sumP = 0;
13         for (double p : pValues) {
14             sumP += p;
15         }
16         // Toleransi kecil untuk perbandingan floating point
17         if (Math.abs(sumP - 1.0) > 1e-9) {
18             System.out.println("Peringatan: Jumlah probabilitas tidak sama dengan 1 (Jumlah = " + sumP + ")");
19             // Anda bisa memilih untuk menghentikan program di sini jika perlu
20             // return;
21         }
22
23         // Format Angka Desimal
24         DecimalFormat df2 = new DecimalFormat("0.00"); // 2 angka desimal
25         DecimalFormat df4 = new DecimalFormat("0.0000"); // 4 angka desimal
26         DecimalFormat df5 = new DecimalFormat("0.00000"); // 5 angka desimal
27
28         // 2. Menghitung Expected Value (E[X] atau  $\mu$ )
29         System.out.println("Menghitung Expected Value (Nilai Harapan):");
30         System.out.println("-----");
31         System.out.println("x | P(x) | x * P(x)");
32         System.out.println("----|-----|-----");
33
34         double expectedValue = 0.0;
35         for (int i = 0; i < xValues.length; i++) {
36             double x = xValues[i];
37             double p = pValues[i];
38             double xp = x * p;
39             expectedValue += xp;
40             System.out.println(
41                 String.format("%0f", x) + " | " +
42                 df2.format(p) + " | " +
43                 String.format("%.0f(%.2f)", x, p) + df2.format(xp)
44             );
45         }
46     }
47 }
```

```

46 System.out.println(" Σ [x * P(x)] = " + df4.format(expectedValue)); // Tampilkan dengan 4 desimal seperti di tabel
47 System.out.println("Expected Value E(X) = μ = " + df2.format(expectedValue)); // Hasil akhir biasanya 1 atau 2 desimal
48 System.out.println("\n-----\n");
49
50
51 // 3. Menghitung Variance (Var(X) atau σ²)
52 // Rumus: Var(X) = Σ [(x - μ)² * P(x)]
53 System.out.println("Menghitung Variance:");
54 System.out.println("-----");
55 System.out.println(" x | P(x) | x - μ | (x - μ)² | (x - μ)² * P(x)");
56 System.out.println("----|-----|-----|-----|-----");
57
58 double variance = 0.0;
59 double mu = expectedValue; // Gunakan nilai E(X) yang sudah dihitung
60
61 for (int i = 0; i < xValues.length; i++) {
62     double x = xValues[i];
63     double p = pValues[i];
64     double deviation = x - mu; // (x - μ)
65     double deviationSq = deviation * deviation; // (x - μ)²
66     double term = deviationSq * p; // (x - μ)² * P(x)
67     variance += term;
68
69     System.out.println(
70         String.format(" %.0f", x) + " | " +
71         df2.format(p) + " | " +
72         String.format("%.2f", deviation) + " | " + // % .2f memberi spasi untuk positif
73         df4.format(deviationSq) + " | " +
74         // Menampilkan perkalian seperti di tabel output soal
75         // String.format("%.4f * %.2f = ", deviationSq, p) +
76         df5.format(term) // Tampilkan hasil term dengan 5 desimal seperti di tabel
77     );
78 }
79 System.out.println("-----");
80 System.out.println(" Σ [(x - μ)² * P(x)] = " + df4.format(variance)); // Hasil penjumlahan variance
81 System.out.println("Variance Var(X) = σ² = " + df4.format(variance)); // Hasil akhir variance
82 System.out.println("-----");
83
84 // Opsional: Menghitung Standard Deviation (Simpangan Baku)
85 double standardDeviation = Math.sqrt(variance);
86 System.out.println("\nStandard Deviation σ = √" + df4.format(variance) + " = " + df2.format(standardDeviation));
87 }

```

B. Output

- Expected Value

```

Run CalculatorRandomVariable x
/usr/lib/jvm/jdk-23.0.2-oracle-x64/bin/java -javaagent:/snap/intellij-idea-community/588/lib/idea
Menghitung Expected Value (Nilai Harapan):
-----
 x | P(x) | x * P(x)
----|-----|-----
 1 | 0.16 | 1(0.16) = 0.16
 2 | 0.22 | 2(0.22) = 0.44
 3 | 0.28 | 3(0.28) = 0.84
 4 | 0.20 | 4(0.20) = 0.80
 5 | 0.14 | 5(0.14) = 0.70
-----
 Σ [x * P(x)] = 2.9400
Expected Value E(X) = μ = 2.94
-----

```

- Variance

```
Menghitung Variance:
-----
x | P(x) | x - μ | (x - μ)2 | (x - μ)2 * P(x)
-----|-----|-----|-----
1 | 0.16 | -1.94 | 3.7636 | 0.60218
2 | 0.22 | -0.94 | 0.8836 | 0.19439
3 | 0.28 | 0.06 | 0.0036 | 0.00101
4 | 0.20 | 1.06 | 1.1236 | 0.22472
5 | 0.14 | 2.06 | 4.2436 | 0.59410
-----
Σ [(x - μ)2 * P(x)] = 1.6164
Variance Var(X) = σ2 = 1.6164
-----

Standard Deviation σ = √1.6164 = 1.27

Process finished with exit code 0
```

***Koreksi: Hasil Standar Deviasi pada soal salah.**

- Output keseluruhan dalam code, untuk **expexted value** dan **variance**

```
/usr/lib/jvm/jdk-23.0.2-oracle-x64/bin/java -javaagent:/snap/intellij-idea-community/588/lib/i
Menghitung Expected Value (Nilai Harapan):
-----
x | P(x) | x * P(x)
-----|-----|-----
1 | 0.16 | 1(0.16) = 0.16
2 | 0.22 | 2(0.22) = 0.44
3 | 0.28 | 3(0.28) = 0.84
4 | 0.20 | 4(0.20) = 0.80
5 | 0.14 | 5(0.14) = 0.70
-----
Σ [x * P(x)] = 2.9400
Expected Value E(X) = μ = 2.94
-----

Menghitung Variance:
-----
x | P(x) | x - μ | (x - μ)2 | (x - μ)2 * P(x)
-----|-----|-----|-----|-----
1 | 0.16 | -1.94 | 3.7636 | 0.60218
2 | 0.22 | -0.94 | 0.8836 | 0.19439
3 | 0.28 | 0.06 | 0.0036 | 0.00101
4 | 0.20 | 1.06 | 1.1236 | 0.22472
5 | 0.14 | 2.06 | 4.2436 | 0.59410
-----
Σ [(x - μ)2 * P(x)] = 1.6164
Variance Var(X) = σ2 = 1.6164
-----

Standard Deviation σ = √1.6164 = 1.27

Process finished with exit code 0
```

2. Buat Coding untuk problem berikut

Bila terjadi pelemparan mata uang sebanyak K kali dengan random variable adalah X= banyaknya muncul uang muka (head pada mata uang). Buat program untuk menampilkan random variable X beserta distribusinya.

Penyelesaian:

A. Code

```
Stochastic Version control
DistribusiBinomialKoin.java x

1 package number2;
2
3 import java.text.DecimalFormat;
4 import java.util.Scanner;
5
6 public class DistribusiBinomialKoin {
7     // Fungsi untuk menghitung faktorial (menggunakan long untuk jangkauan lebih besar)
8     public static long faktorial(int n) { no usages
9         if (n < 0) {
10             // Faktorial tidak terdefinisi untuk negatif, tapi dalam konteks binomial tidak akan terjadi
11             return -1; // Atau throw exception
12         }
13         if (n == 0) {
14             return 1;
15         }
16         long hasil = 1;
17         for (int i = 2; i <= n; i++) {
18             hasil *= i;
19         }
20         return hasil;
21     }
22     // Fungsi untuk menghitung koefisien binomial C(n, k) = n! / (k! * (n-k)!)
23     // Menggunakan double untuk hasil akhir karena pembagian bisa menghasilkan desimal
24     // dan untuk menghindari overflow pada perkalian faktorial besar sebelum dibagi
25     public static double koefisienBinomial(int n, int k) { 1usage
26         if (k < 0 || k > n) {
27             return 0; // Kombinasi tidak valid
28         }
29         if (k == 0 || k == n) {
30             return 1; // C(n, 0) = 1, C(n, n) = 1
31         }
32         // Optimasi: C(n, k) = C(n, n-k), hitung dengan k yang lebih kecil
33         if (k > n / 2) {
34             k = n - k;
35         }
36
37         // Perhitungan yang lebih stabil untuk menghindari overflow besar
38         // C(n, k) = (n * (n-1) * ... * (n-k+1)) / k!
39         double hasil = 1.0;
40         for (int i = 1; i <= k; i++) {
41             // Lakukan perkalian dan pembagian secara bergantian
42             hasil = hasil * (n - i + 1) / i;
43         }
44         return hasil;
45     }
46 }
```

```

46
47 // Fungsi untuk menghitung probabilitas binomial  $P(X=x)$ 
48 // n: jumlah percobaan (K lemparan)
49 // k: jumlah sukses (jumlah head, x)
50 // p: probabilitas sukses dalam satu percobaan (probabilitas head, 0.5)
51 public static double probabilitasBinomial(int n, int k, double p) { // usage
52     double kombinasi = koefisienBinomial(n, k);
53     if (kombinasi < 0) return -1; // Propagasi error jika ada
54
55     double probSukses = Math.pow(p, k); //  $p^k$ 
56     double probGagal = Math.pow(1.0 - p, n - k); //  $(1-p)^{(n-k)}$ 
57
58     return kombinasi * probSukses * probGagal;
59 }
60
61 public static void main(String[] args) {
62     Scanner scanner = new Scanner(System.in);
63
64     System.out.print("Masukkan jumlah pelemparan mata uang (K): ");
65     int k = -1; // Inisialisasi dengan nilai tidak valid
66
67     // Validasi input K harus non-negatif
68     while (k < 0) {
69         try {
70             k = scanner.nextInt();
71             if (k < 0) {
72                 System.out.print("Jumlah pelemparan tidak boleh negatif. Masukkan lagi: ");
73             }
74         } catch (java.util.InputMismatchException e) {
75             System.out.print("Input tidak valid. Masukkan angka bulat non-negatif: ");
76             scanner.next(); // Buang input yang salah
77         }
78     }
79
80     // Probabilitas muncul muka (head) pada satu lemparan koin (diasumsikan adil)
81     double pHead = 0.5;
82
83     System.out.println("\nDistribusi Probabilitas Variabel Acak X (Jumlah Muka/Head)");
84     System.out.println("Jumlah Pelemparan (K) = " + k);
85     System.out.println("Probabilitas Head (p) = " + pHead);
86     System.out.println("-----");
87     System.out.println("X (Jumlah Head) | P(X = x) "); // Header tabel
88     System.out.println("-----|-----");
89
90     DecimalFormat df = new DecimalFormat("0.000000"); // Format untuk probabilitas
91     double totalProbabilitas = 0.0;

```

```

92
93     // Loop untuk setiap kemungkinan nilai X (dari 0 sampai K)
94     for (int x = 0; x <= k; x++) {
95         double probabilitasX = probabilitasBinomial(k, x, pHead);
96         if (probabilitasX >= 0) { // Cek jika tidak ada error perhitungan
97             System.out.printf(" %-15d | %s\n", x, df.format(probabilitasX));
98             totalProbabilitas += probabilitasX;
99         } else {
100             System.out.printf(" %-15d | Error Hitung\n", x);
101         }
102     }
103
104     System.out.println("-----");
105     // Verifikasi: Jumlah semua probabilitas harus mendekati 1
106     System.out.println("Total Probabilitas = " + df.format(totalProbabilitas));
107     System.out.println("-----");
108
109
110     scanner.close(); // Tutup scanner setelah selesai digunakan
111 }
112 }
113

```

B. Output

Pada persoalan ini kita akan melempar mata uang (K) = 10.

```

Run DistribusiBinomialKoin x
/usr/lib/jvm/jdk-23.0.2-oracle-x64/bin/java -javaagent:/snap/intellij-idea-community/588/lib
Masukkan jumlah pelemparan mata uang (K): 10

Distribusi Probabilitas Variabel Acak X (Jumlah Muka/Head)
Jumlah Pelemparan (K) = 10
Probabilitas Head (p) = 0.5

-----
X (Jumlah Head) | P(X = x)
-----|-----
0 | 0.000977
1 | 0.009766
2 | 0.043945
3 | 0.117188
4 | 0.205078
5 | 0.246094
6 | 0.205078
7 | 0.117188
8 | 0.043945
9 | 0.009766
10 | 0.000977
-----

Total Probabilitas = 1.000000
-----

Process finished with exit code 0

```