0)

satu jalur mengandung link-link yang tidak gagal/tidak out of service. Tentukan probabilitas kota A dan B bisa berkomunikasi? BC tersambung / fidak gagal: 1 - (0.25) = 0,984

Ac tersambung/fidax  $9a941 = 1 - 0.25 \left(1 - \left(0.75(1 - \left(0.75(1 - \left(0.75^2\right)\right)\right) = 0.925 \right)$ 

1-0.25 : 0.75

AB tidak gagal = 1 - ((1-P(AB Dtas). (1-P(AB Bawah)) - (0.25)3 = 1 - (1- (PBC. PAC)) \* (1- PAB) - ( | - ( | - 0,9 | 1) , 0.25

Tabel f(x,y) Х 3 0.15 0.2 0.1 0.3 0.05 0.05 0.05 0.1

80.  $P(0 \le X \le 2, Y = 1) = ?$ 81. E(X) = ? 83. E(Y) = ?84. var(Y) = ?82. var(X) = ? 85. Covariance of X and Y = ?

86. Correlation coefficient of X and Y = ?import sympy as sp

# Tabel probabilitas gabungan f(x, y) f\_xy = sp.Matrix([ [a, b, c, d], # y = 0 # y = 1 [f, g, h, i] # Soal 80:  $P(0 \le X \le 2, Y = 1)$  $p_x_0_2_y_1 = f_xy[1, 0] + f_xy[1, 1] + f_xy[1, 2]$  $print("P(0 \le X \le 2, Y = 1) = ", p_x_0_2_y_1)$ # Distribusi marginal untuk X dan Y  $f_x = [f_{xy}[0, i] + f_{xy}[1, i] \text{ for } i \text{ in } range(4)]$  $f_y = [sum(f_xy.row(i)) for i in range(2)]$ # Soal 81: E(X)  $x_{values} = sp.Matrix([0, 1, 2, 3])$  $E_X = sum(x * f_x[i] for i, x in enumerate(x_values))$  $print("E(X) = ", E_X)$ # Soal 82: Var(X)  $var_X = sum((x - E_X)**2 * f_x[i] for i, x in enumerate(x_values))$ print("Var(X) =", var\_X) # Soal 83: E(Y) y\_values = sp.Matrix([0, 1]) E\_Y = sum(y \* f\_y[i] for i, y in enumerate(y\_values)) print("E(Y) =", E\_Y)

# Soal 84: Var(Y)  $var_Y = sum((y - E_Y)**2 * f_y[i] for i, y in enumerate(y_values))$ print("Var(Y) =", var\_Y) # Soal 85: Covariance(X, Y) # E(XY) calculation E\_XY = sum(x \* y \* f\_xy[j, i] for i, x in enumerate(x\_values) for j, y in enumerate(y\_values))  $cov_XY = E_XY - E_X * E_Y$ print("Covariance(X, Y) =", cov\_XY) # Soal 86: Correlation coefficient(X, Y) corr\_XY = cov\_XY / sp.sqrt(var\_X \* var\_Y) print("Correlation coefficient(X, Y) =", corr\_XY)

Sebuah universitas sedang memilih 1 dari 3 model komputer untuk digunakan di laboratorium penelitian: Model PQR, Model STU, dan Model VWX. • Komputer Model PQR memiliki usia pakai (waktu sampai komputer perlu diperbaiki atau diganti) yang terdistribusi normal dengan rata-rata 8500 jam dan standar deviasi 800 jam. Komputer Model STU memiliki usia pakai yang terdistribusi eksponensial dengan rata-rata 8500 jam.

 Komputer Model VWX memiliki usia pakai yang terdistribusi Weibull dengan rata-rata atau expected value 8500 jam dan parameter k=1.5k = 1.5k=1.5.

a. Probabilitas **komputer Model PQR** rusak antara 7000-8000 jam adalah... b. Expected value dari usia pakai **komputer Model PQR** adalah...

g. Apabila universitas memilih model komputer dengan pertimbangan: "dipilih model komputer dengan probabilitas rusak sebelum 8000 jam paling rendah," maka universitas akan memilih model...

import numpy as np import scipy.stats as stats from scipy.special import gamma # Mengimpor fungsi gamma dari scipy.special # Parameter mean\_abc = std\_dev\_abc = a. Probabilitas MRI brand ABC rusak antara 9000-10000 jam prob\_abc= stats.norm.cdf(8000, mean\_abc, std\_dev\_abc) - stats.norm.cdf(7000, mean\_abc, std\_dev\_abc) # b. Expected value ABC expected\_value\_abc = mean\_abc

# c. Probabilitas MRI brand DEF prob\_def\_9000\_10000 = (1 - np.exp(-7000 / mean\_def)) - (1 - np.exp(-80000 / mean\_def)) # d. Expected value dari umur MRI brand DEF expected\_value\_def = mean\_def

# Parameter untuk MRI brand XYZ (Weibull Distribution)  $k_xyz = 1.5$  $scale_{xyz} = (mean_{abc} / gamma(1 + 1/k_{xyz}))$  # Menggunakan gamma dari scipy.special # e. Probabilitas MRI brand XYZ rusak antara 9000-10000 jam (stats.weibull\_min.cdf(8000, k\_xyz, scale=scale\_xyz) prob\_xyz = stats.weibull\_min.cdf(7000, k\_xyz, scale=scale\_xyz)) # f. Expected value dari umur MRI brand XYZ expected\_value\_xyz = scale\_xyz \* gamma(1 + 1/k\_xyz) g. Memilih brand dengan probabilitas rusak sebelum 8000 jam paling rendah prob\_fail\_before\_8000\_abc = stats.norm.cdf(8000, mean\_abc, std\_dev\_abc)

# Menentukan brand dengan probabilitas rusak paling rendah min\_prob\_brand = np.argmin([prob\_fail\_before\_8000\_abc, prob\_fail\_before\_8000\_def, prob\_fail\_before\_8000\_xyz]) + 1 🖳 PDF (probability density function) dari random variabel A PDF (probability density function) dari random variabel B

prob\_fail\_before\_8000\_def = 1 - np.exp(-8000 / mean\_def)

prob\_fail\_before\_8000\_xyz = stats.weibull\_min.cdf(8000, k\_xyz, scale=scale\_xyz)

Expected value dari random variabel

Joint probability function dari random variabel a & bgh

Probabilitas PT. B memenangkan tender adalah

Dindependent  $f(a,b) = f(a)^*f(b)$