

## LAMPIRAN

### Lampiran.1 Hasil Kuisisioner Pengujian UAT

#### • Hasil Kuisisioner Pengujian UAT

No.	Pertanyaan	Respon User		
		User 1	User 2	User 3
1.	Nama Lengkap	Fajar Adriyanto, M.B.A., M.Si	Ir. M. Zuhdizul, MT.	Bayu Anugerah R.P., S.H.
2.	Pangkat	Marsekal Pertama	Kolonel Teknik	Kapten Pnb
3.	Jabatan	Kapuspotdirga	Dosen Utama	P.s. Komandan Flight C Skadik 101
4.	Instansi	Mabes TNI-AU	Seskoau	Skadik 101 Wingdikterbang
5.	Apakah Sistem dapat dioperasikan dengan baik pada perangkat?	80 %	-	80%
6.	Apakah tampilan Sistem user friendly atau mudah digunakan?	70 %	-	90 %
7.	Apakah fitur-fitur Sistem ini mudah dipahami?	70%	-	70 %
8.	Apakah hasil (output) yang dari Sistem sesuai dengan kebutuhan?	80 %	-	40 %
9.	Apakah Sistem ini telah menyediakan fitur-fitur yang dibutuhkan?	70 %	-	50 %
10.	Apakah informasi yang disediakan oleh Sistem mudah dimengerti?	75 %	-	40 %
11.	Apakah Sistem menunjukkan kinerja fungsi sesuai yang dengan kebutuhan?	75 %	-	50 %
12.	Apakah hasil prediksi dari Sistem dapat membantu perencanaan pemeliharaan dan pengoperasi Alutsista di masa mendatang?	70 %	-	50 %

## Lampiran. 2 Source Code Perhitungan Metode

- **Source Code Regresi Linier**

```
# Import Library
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Read Data
data = pd.read_csv("data2019.csv", delimiter=",")
print(data.shape)
data

# Set x to our input Periode
# Set y to our output Kesiapan, this is the value we trying to predict
X = data['sirkulasi'].values
Y = data['kesiapan'].values

# Plot Kesiapan against Sirkulasi
plt.scatter(X, Y, label='Data')

plt.xlabel('Sirkulasi')
plt.ylabel('Rata-rata Kesiapan')

plt.legend()
plt.show()

mean_x = np.mean(X.copy())
mean_y = np.mean(Y.copy())

n = len(X)

num = 0
den = 0
for i in range(n):
    num += (X[i] - mean_x) * (Y[i] - mean_y)
    den += (X[i] - mean_x) **2

b1 = float(num) / float(den)
b0 = mean_y - (b1 * mean_x)

print('Mean Sirkulasi = '+ str(mean_x))
print('Mean Kesiapan = '+ str(mean_y))
print(b0, b1)
```

```
max_x = np.max(X) + 10
min_x = np.min(X) - 10

x = np.linspace(min_x, max_x, 100)
y = b0 + b1 * x

plt.plot(x,y, label='Linear Regression')
plt.scatter(X, Y, label='Scatter Plot')

plt.xlabel('Sirkulasi')
plt.ylabel('Rata-rata Kesiapan')

plt.legend()
plt.show()

ss_t = 0
ss_r = 0

for i in range(n):
    y_pred = b0 + b1 * X[i]
    ss_t += (Y[i] - mean_y) ** 2
    ss_r += (Y[i] - y_pred) ** 2

r2 = 1 - (ss_r/ss_t)
print(r2)
```