

**EVALUASI MANDIRI OPTIMISASI
MATAKULIAH OPTIMISASI
MENGENALI OPTIMASASI LINEAR/NONLINEAR**



**Disusun oleh:
SHARA ALYA GIFANI MUHYISUNAH
G1D021038**

**Dosen Pengampu:
Ir. Novalio Daratha S.T., M.Sc., Ph.D.**

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2024**

1. Mengimpor Pustaka

`using JuMP`

`using Ipopt`

- JuMP adalah pustaka Julia untuk pemrograman matematis dan optimasi.
- Ipopt adalah solver untuk optimasi nonlinear, yang digunakan untuk menyelesaikan masalah optimasi yang dikendalikan oleh JuMP.

2. Membuat Model Optimasi

`model = Model(Ipopt.Optimizer)`

- Membuat model optimasi menggunakan solver Ipopt. Model ini akan digunakan untuk mengoptimalkan berbagai fungsi objektif.

3. Mendefinisikan Variabel Keputusan

`@variable(model, y1 >= 0)`

`@variable(model, y2 >= 0)`

- `@variable(model, y1 >= 0)` mendefinisikan variabel keputusan y_1 yang memiliki nilai non-negatif.
- `@variable(model, y2 >= 0)` mendefinisikan variabel keputusan y_2 yang juga non-negatif.

4. Mendefinisikan Kendala

`@constraint(model, 3*y1 + y2 >= 2)`

`@constraint(model, y1 + 4*y2 >= 3)`

- `@constraint(model, 3*y1 + y2 >= 2)` mendefinisikan kendala pertama, yaitu $3y_1 + y_2 \geq 2$.
- `@constraint(model, y1 + 4*y2 >= 3)` mendefinisikan kendala kedua, yaitu $y_1 + 4y_2 \geq 3$.

Kedua kendala ini harus dipenuhi oleh solusi optimal yang dicari.

5. Mendefinisikan Fungsi Objektif dan Menyelesaikan Masalah

Fungsi objektif yang berbeda digunakan dalam setiap iterasi, dan masalah dioptimalkan untuk setiap fungsi tersebut.

Fungsi Objektif (a): $y_1 + y_2$

`println("1(a) f0(y1, y2) = y1 + y2")`

`@objective(model, Min, (y1 + y2))`

`optimize!(model)`

```
println("Optimal Value for (a): ", objective_value(model))
println("Optimal Solution for (a): (" , value(y1), " , " , value(y2), ")")
```

- Fungsi objektif yang digunakan adalah $y1 + y2$, yang dimaksudkan untuk diminimalkan (Min).
- `optimize!(model)` menjalankan optimasi untuk menemukan solusi terbaik yang meminimalkan fungsi objektif.
- `objective_value(model)` memberikan nilai dari fungsi objektif setelah optimasi.
- `value(y1)` dan `value(y2)` memberikan nilai optimal untuk variabel $y1$ dan $y2$.

Fungsi Objektif (b): $y1 - y2$

```
println("1(b) f0(y1, y2) = y1 - y2")
@objective(model, Min, (y1 - y2))
optimize!(model)
println("Optimal Value for (b): ", objective_value(model))
println("Optimal Solution for (b): (" , value(y1), " , " , value(y2), ")")
```

- Fungsi objektif yang digunakan adalah $y1 - y2$, yang juga dimaksudkan untuk diminimalkan.
- Proses optimasi dilakukan dan solusi optimal dicetak.

Fungsi Objektif (c): $y1$

```
println("1(c) f0(y1, y2) = y1")
@objective(model, Min, (y1))
optimize!(model)
println("Optimal Value for (c): ", objective_value(model))
println("Optimal Solution for (c): (" , value(y1), " , " , value(y2), ")")
```

- Fungsi objektif yang digunakan adalah $y1$, yang dimaksudkan untuk diminimalkan.
- Proses optimasi dilakukan dan solusi optimal dicetak.

Fungsi Objektif (d): $\max(y1, y2)$

```
println("1(d) f0(y1, y2) = max{y1, y2}")
@objective(model, Min, y1)
optimize!(model)
println("Optimal Value for (d): ", objective_value(model))
println("Optimal Solution for (d): (" , value(y1), " , " , value(y2), ")")
```

- Fungsi objektif yang digunakan adalah $\max(y1, y2)$, yang merupakan fungsi nonlinear (karena melibatkan operasi maksimum).
- Pada bagian ini, hanya $y1$ yang dioptimasi karena fungsi $\max(y1, y2)$ akan mencari nilai terbesar antara $y1$ dan $y2$. Namun, karena solver Ipopt bekerja dengan baik pada fungsi kontinu, bagian ini mungkin tidak sepenuhnya optimal

untuk masalah max(), karena solver mungkin lebih cocok untuk fungsi yang lebih halus.

Fungsi Objektif (e): $(y1)^2 + 8*(y2)^2$

```
println("1(e) f0(y1, y2) = (y1)^2 + 8*(y2)^2")
```

```
@objective(model, Min, (y1)^2 + 8*(y2)^2)
```

```
optimize!(model)
```

```
println("Optimal Value for (e): ", objective_value(model))
```

```
println("Optimal Solution for (e): (", value(y1), ", ", value(y2), ")")
```

- Fungsi objektif yang digunakan adalah $y1^2 + 8*y2^2$, yang merupakan fungsi nonlinear (karena melibatkan kuadrat variabel).
- Proses optimasi dilakukan dan solusi optimal dicetak.

CODE PROGRAM

```
using JuMP
```

```
using Ipopt
```

```
# Membuat model dengan optimizer Ipopt
```

```
model = Model(Ipopt.Optimizer)
```

```
# Mendefinisikan variabel keputusan
```

```
@variable(model, y1 >= 0)
```

```
@variable(model, y2 >= 0)
```

```
# Mendefinisikan kendala
```

```
@constraint(model, 3*y1 + y2 >= 2)
```

```
@constraint(model, y1 + 4*y2 >= 3)
```

```
# Fungsi Objektif (a)
```

```
println("1(a) f0(y1, y2) = y1 + y2")
```

```
@objective(model, Min, (y1 + y2))
```

```
optimize!(model)
```

```
println("Optimal Value for (a): ", objective_value(model))
```

```
println("Optimal Solution for (a): (", value(y1), ", ", value(y2), ")")
```

```
# Fungsi Objektif (b)
```

```
println("1(b) f0(y1, y2) = y1 - y2")
```

```
@objective(model, Min, (y1 - y2))
optimize!(model)
println("Optimal Value for (b): ", objective_value(model))
println("Optimal Solution for (b): (" , value(y1), " , " , value(y2), ")")
```

```
# Fungsi Objektif (c)
println("l(c) f0(y1, y2) = y1")
@objective(model, Min, (y1))
optimize!(model)
println("Optimal Value for (c): ", objective_value(model))
println("Optimal Solution for (c): (" , value(y1), " , " , value(y2), ")")
```

```
# Fungsi Objektif (d)
println("l(d) f0(y1, y2) = max {y1, y2}")
@objective(model, Min, y1)
optimize!(model)
println("Optimal Value for (d): ", objective_value(model))
println("Optimal Solution for (d): (" , value(y1), " , " , value(y2), ")")
```

```
# Fungsi Objektif (e)
println("l(e) f0(y1, y2) = (y1)^2 + 8*(y2)^2")
@objective(model, Min, (y1)^2 + 8*(y2)^2)
optimize!(model)
println("Optimal Value for (e): ", objective_value(model))
println("Optimal Solution for (e): (" , value(y1), " , " , value(y2), ")")
```