

# Express.js

## Request Validation

# Huh?

Proses memastikan user input (masukan pengguna) memenuhi kriteria tertentu, sebelum diproses lebih lanjut. Hal ini dilakukan untuk memastikan keakuratan dan kebersihan data yang masuk ke dalam sistem.

## **Validasi Dilakukan Melalui**

- Checking (Pemeriksaan)
- Cleaning (Pembersihan)
- Transformation (Transformasi)

# Why?

- Menjaga Kualitas Data (Data Integrity)
- Security (Keamanan)
- User Experience (Pengalaman Pengguna)

# Data Integrity

Integritas data adalah kualitas data yang menunjukkan bahwa data tersebut akurat dan dapat diandalkan.

## Mengapa validasi penting?

- Keandalan (reliability) dan kepercayaan pada sistem
- Pengambilan keputusan berbasis data
- Mematuhi regulasi dan standar

# Security

## Mengapa Validasi Penting untuk Keamanan

- Mencegah SQL Injection
- Menghindari Cross-Site Scripting (XSS)
- Melindungi dari serangan lain berbasis input tidak valid

# User Experience

## Mengapa Validasi Penting untuk Pengalaman Pengguna

- Mencegah kesalahan input
- Memberikan feedback yang jelas dan informatif kepada pengguna jika terjadi kesalahan
- Memastikan data yang dimasukkan sesuai dengan
  - format yang diharapkan
  - aturan bisnis
  - aturan keamanan

# Tipe-Tipe Data Validation

## **Format Validation**

Memastikan format data sesuai (mis. email, tanggal)

## **Data Type Validation**

Menentukan tipe data (mis. integer, string)

## **Range Validation**

Memeriksa data berada dalam range tertentu (mis. usia > 18)

## **Custom Validation**

Aturan validasi khusus (mis. username tidak boleh mengandung karakter tertentu)

# Validator di Express.js

Tidak ada built-in validators di Express.js



# Third-Party Validator

- express-validator
- joi

# Menggunakan express-validator di Express.js

- Middleware-based
- Customizable
- Supports async validation

# express-validator

```
npm install express-validator
```

# express-validator example

```
import express from 'express'
import { query, validationResult } from 'express-validator'

const app = express()

app.get('/', query('person').notEmpty(), (req, res) => {
  res.json({ message: `Hello ${req.query.person}` })
})
```

# express-validator example

```
import express from 'express'
import { query, validationResult } from 'express-validator'

const app = express()

app.get('/', query('person').notEmpty(), (req, res) => {
  const result = validationResult(req)
  if (!result.isEmpty()) {
    return res.status(400).json({ errors: result.array() })
  }

  res.json({ message: `Hello ${req.query.person}` })
})
```

# express-validator example

```
import express from 'express'
import { query, validationResult } from 'express-validator'

const app = express()

app.post('/login', [body('email').isEmail(), body('password').isLength({ min: 5 })], (req, res) => {
  const result = validationResult(req)
  if (!result.isEmpty()) {
    return res.status(400).json({ errors: result.array() })
  }
  res.json({ message: 'Login successful!' })
})
```