

INTEROPERABILITAS

TUGAS JSON DAN XML API



Dosen Pengampu :
Sepyan Purnama Kristanto, S.Kom., M.Kom.

Disusun Oleh :

Holut Yudawan 362358302014
A2 TRPL

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK
JURUSAN BISNIS DAN INFORMATIKA
POLITEKNIK NEGERI BANYUWANGI
TAHUN 2024

Pertama membuat endpoint get untuk api json

```
Command Prompt - node ser × + ▾
C:\Users\holut\tugas_API>npm init -y
Wrote to C:\Users\holut\tugas_API\package.json:

{
  "name": "tugas_api",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

C:\Users\holut\tugas_API>npm install express body-parser xml
added 66 packages, and audited 67 packages in 2s

13 packages are looking for funding
  run `npm fund` for details
```

➤ Membuat file server.js serta menambahkan kode untuk get json

```
const express = require("express");
const bodyParser = require("body-parser");
const app = express();
app.use(bodyParser.json());

//daftar item
let items = [
  { id: 1, name: "Buku Pada Zaman Dahulu", price: 100.0 },
  { id: 2, name: "Siksa Neraka", price: 45.0 },
  { id: 1, name: "Jembatan Ilmu", price: 100.0 },
];

//mengembalikan daftar item ke json
app.get("/api/json/items", (req, res) => {
  res.json(items);
});

const port = 3000;
app.listen(port, () => {
  console.log("Server running on port ${port}");
});
```

➤ Menambahkan endpoint get untuk xml

```
const express = require("express");
const bodyParser = require("body-parser");
const xml = require("xml");
const app = express();
app.use(bodyParser.json());

let items = [
  { id: 1, name: "Buku Pada Zaman Dahulu", price: 100.0 },
  { id: 2, name: "Siksa Neraka", price: 45.0 },
  { id: 1, name: "Jembatan Ilmu", price: 100.0 },
];

//mengembalikan daftar item ke json
app.get("/api/json/items", (req, res) => {
  res.json(items);
});

// GET - Mengembalikan daftar item dalam format XML
app.get("/api/xml/items", (req, res) => {
  const xmlItems = xml([
    {
      items: items.map((item) => ({
        item: [{ id: item.id }, { name: item.name }, { price: item.price }],
      })),
    },
  ]);
  res.set("Content-Type", "application/xml");
  res.send(xmlItems);
});

const port = 3000;
app.listen(port, () => {
  console.log("Server running on port ${port}");
});
```

➤ Kemudian menambahkan endpoint POST

```
const express = require("express");
const bodyParser = require("body-parser");
const xml = require("xml");
const app = express();
app.use(bodyParser.json());

let items = [
  { id: 1, name: "Buku Pada Zaman Dahulu", price: 100.0 },
  { id: 2, name: "Siksa Neraka", price: 45.0 },
  { id: 1, name: "Jembatan Ilmu", price: 100.0 },
```

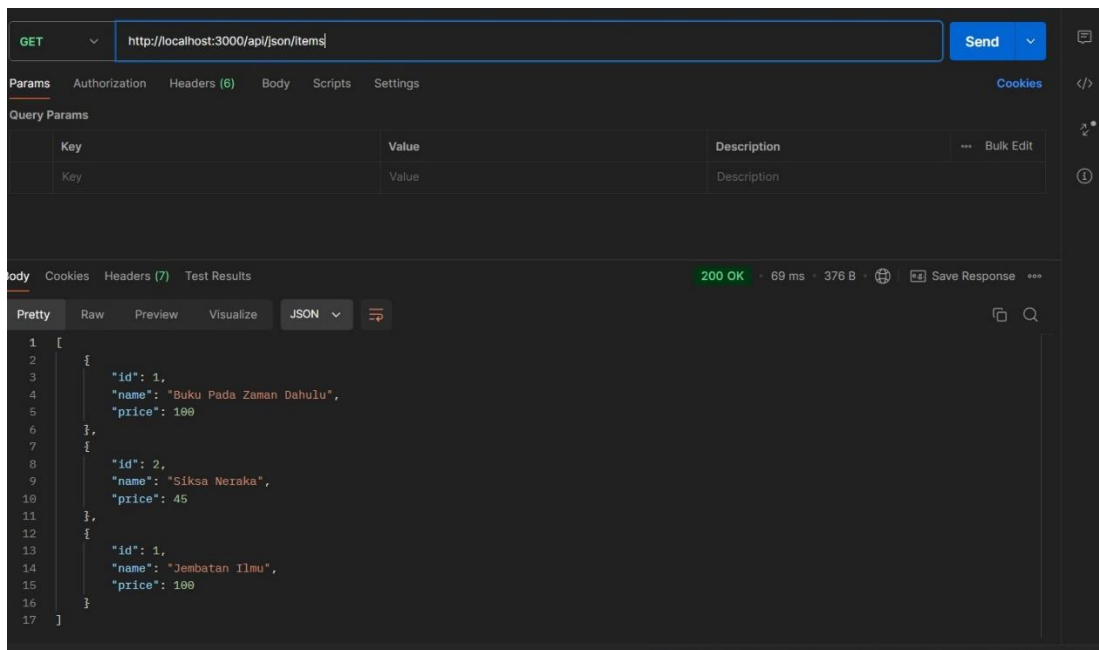
```
];
```

```
//mengembalikan daftar item ke json  
app.get("/api/json/items", (req, res) => {  
  res.json(items);  
});
```

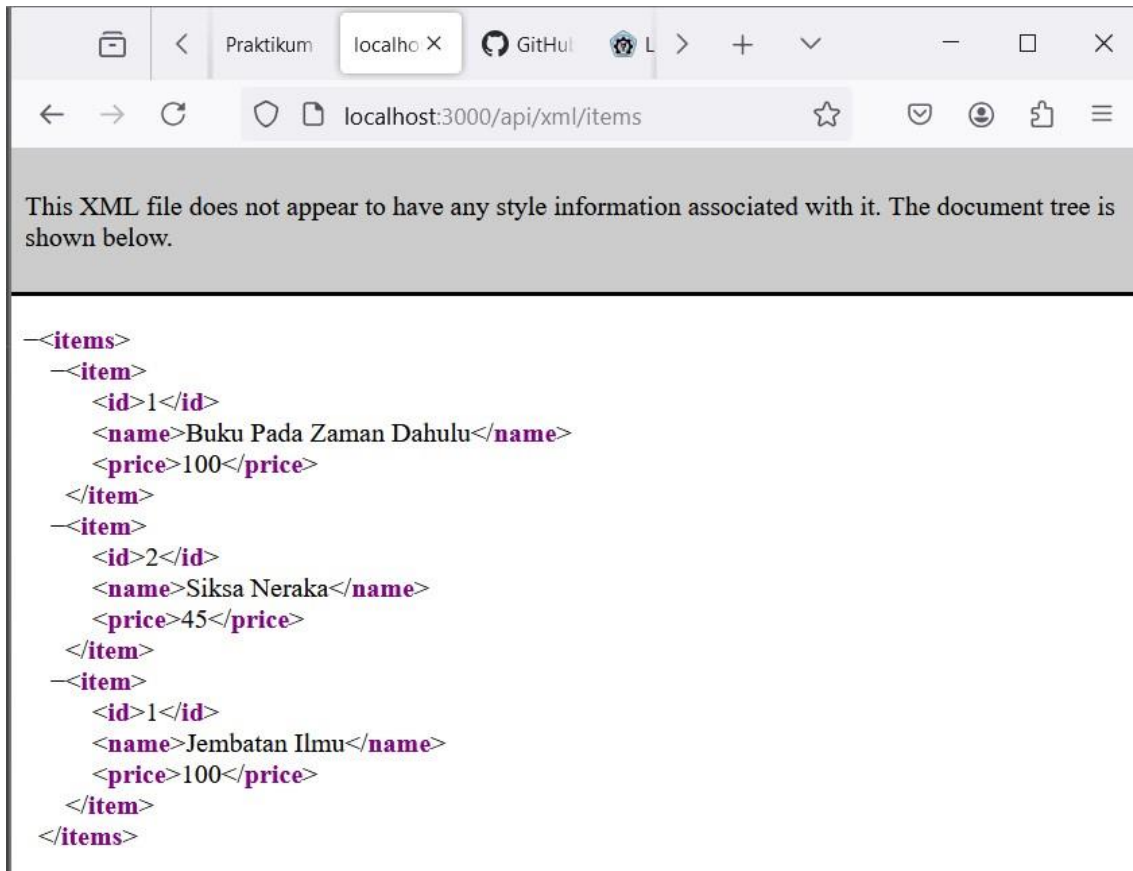
```
//menambahkan item baru ke json  
app.post("/api/json/items", (req, res) => {  
  const newItem = req.body;  
  items.push(newItem);  
  res.status(201).json(newItem);  
});
```

```
// GET - Mengembalikan daftar item dalam format XML  
app.get("/api/xml/items", (req, res) => {  
  const xmlItems = xml([  
    {  
      items: items.map((item) => ({  
        item: [{ id: item.id }, { name: item.name }, { price: item.price }],  
      })),  
    },  
  ]);  
  res.set("Content-Type", "application/xml");  
  res.send(xmlItems);  
});  
  
const port = 3000;  
app.listen(port, () => {  
  console.log("Server running on port ${port}");  
});
```

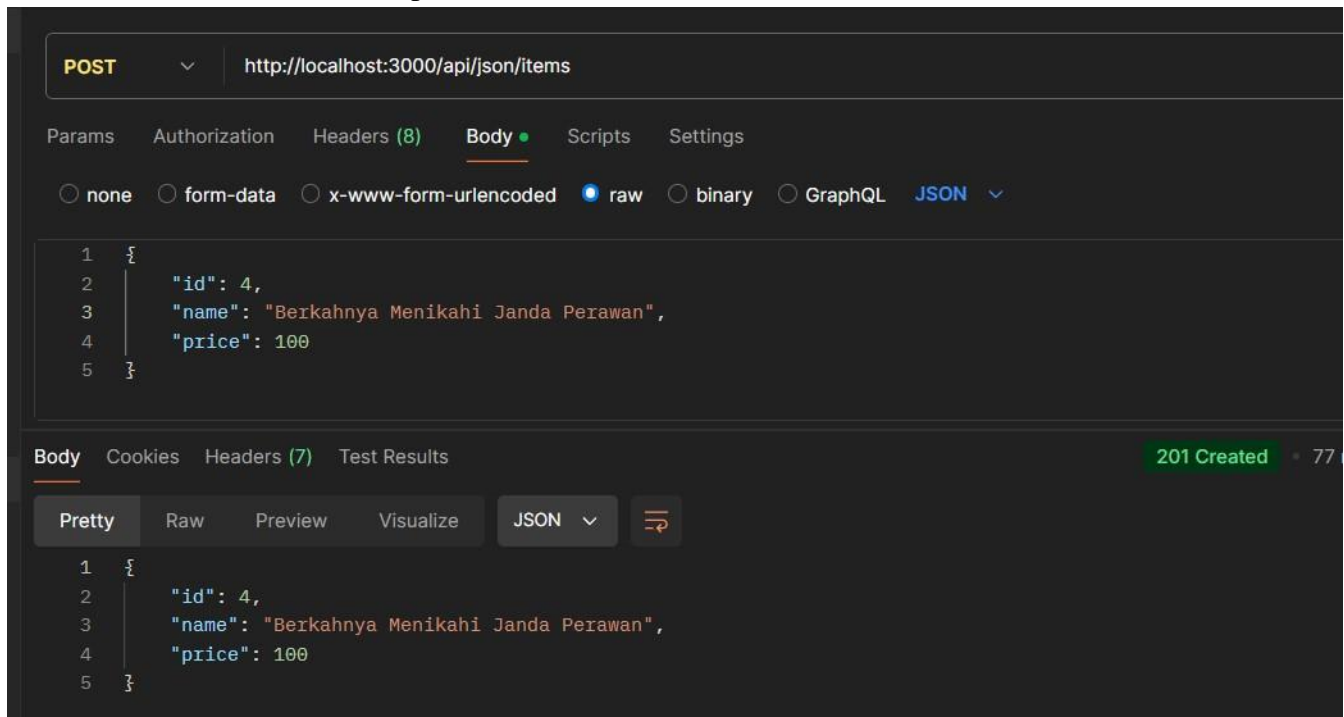
➤ **Buat endpoint GET di API JSON dan XML yang mengembalikan daftar item**



Berikut ketika xml yang mengembalikan item



Test menambahkan item ke post



➤ Membuat script untuk mengambil data dari API JSON

```
const axios = require("axios");
const xml = require("xml");

async function convert() {
  try {
    const response = await axios.get("http://localhost:3000/api/json/items");
    const items = response.data;

    const xmlItems = xml([
      {
        items: items.map((item) => ({
          item: [{ id: item.id }, { name: item.name }, { price: item.price }],
        })),
      },
    ]);

    console.log("Data dalam format XML:\n", xmlItems);
  } catch (error) {
    console.error("Error fetching data:", error);
  }
}

convert();
```

➤ Output yang di hasilkan dari mengubah menjadi xml

