

# Modul: Program Daftar Nilai Siswa dengan JSON

## A. Tujuan Pembelajaran

Setelah mempelajari modul ini, peserta didik mampu: 1. Menjelaskan apa itu JSON dan fungsinya dalam pemrograman.

2. Menggunakan list dan dictionary di Python untuk menyimpan data siswa.

3. Menyimpan dan membaca data dari file JSON.

4. Membuat program daftar nilai siswa yang dapat menambah data secara berulang dan tersimpan permanen.

## B. Konsep Dasar

### 1. Apa itu JSON?

- **JSON** (JavaScript Object Notation) adalah format pertukaran data yang sederhana dan mudah dibaca.
- JSON sering digunakan untuk menyimpan data agar bisa digunakan kembali.
- Contoh struktur JSON:

```
{
  "siswa": [
    {"nama": "Andi", "nilai": 85},
    {"nama": "Budi", "nilai": 90}
  ]
}
```

### 2. Mengapa JSON digunakan?

- Data tersimpan di file, sehingga **tidak hilang** meskipun program ditutup.
- Bisa diakses dan dibaca ulang kapan saja.
- Formatnya sederhana dan bisa digunakan di berbagai bahasa pemrograman.

## C. Program Daftar Nilai Siswa

### 1. Tanpa JSON (Data Hilang Setelah Program Ditutup)

```
siswa = [
    {"nama": "Andi", "nilai": 85},
    {"nama": "Budi", "nilai": 90},
    {"nama": "Citra", "nilai": 78}
]
```

```

nama_baru = input("Masukkan nama siswa: ")
nilai_baru = int(input("Masukkan nilai: "))
siswa.append({"nama": nama_baru, "nilai": nilai_baru})

print(siswa)

```

👉 Kekurangan: Saat program ditutup, data tambahan **hilang**.

## 2. Dengan JSON (Data Tersimpan di File)

Program berikut akan membaca data dari file `siswa.json` jika ada, lalu menambahkan data baru, kemudian menyimpan kembali.

```

import json
import os

# Cek apakah file sudah ada
if os.path.exists("siswa.json"):
    with open("siswa.json", "r") as f:
        data = json.load(f)
        siswa = data["siswa"]
else:
    siswa = [
        {"nama": "Andi", "nilai": 85},
        {"nama": "Budi", "nilai": 90},
        {"nama": "Citra", "nilai": 78}
    ]

# Menampilkan daftar siswa
print("Daftar Nilai Siswa:")
print("-----")
for data in siswa:
    print(f>Nama: {data['nama']}, Nilai: {data['nilai']}")

# Input berulang
while True:
    print("\nTambah Data Siswa")
    nama_baru = input("Masukkan nama siswa (atau ketik 'selesai' untuk berhenti): ")
    if nama_baru.lower() == "selesai":
        break
    nilai_baru = int(input("Masukkan nilai: "))

    siswa.append({"nama": nama_baru, "nilai": nilai_baru})

# Simpan ke file JSON
with open("siswa.json", "w") as f:
    json.dump({"siswa": siswa}, f, indent=4)

```

```
# Menampilkan hasil akhir
print("\nDaftar Nilai Siswa Terbaru:")
print("-----")
for data in siswa:
    print(f>Nama: {data['nama']}, Nilai: {data['nilai']}")
```

## D. Alur Program

1. **Cek file JSON** → jika ada, data lama dimuat.
2. **Tampilkan daftar nilai siswa.**
3. **Input data baru** secara berulang sampai pengguna mengetik `selesai`.
4. **Simpan kembali ke file JSON** agar data tidak hilang.
5. **Tampilkan daftar terbaru.**

## E. Hasil Program

Misalnya pengguna menambahkan:

```
Nama: Dedi, Nilai: 88
Nama: Evi, Nilai: 92
```

Maka isi file `siswa.json` akan menjadi:

```
{
  "siswa": [
    {"nama": "Andi", "nilai": 85},
    {"nama": "Budi", "nilai": 90},
    {"nama": "Citra", "nilai": 78},
    {"nama": "Dedi", "nilai": 88},
    {"nama": "Evi", "nilai": 92}
  ]
}
```

## F. Latihan Peserta Didik

1. Jalankan program dan tambahkan minimal 5 data siswa baru.
2. Tampilkan isi file `siswa.json` menggunakan text editor, lalu bandingkan dengan output di terminal.
3. Modifikasi program agar:
4. Hanya menerima **nilai 0–100**.
5. Menolak input kosong pada nama siswa.

6. Berhenti otomatis jika jumlah siswa sudah mencapai **50 orang**.

---

## G. Kesimpulan

- JSON digunakan agar data **tetap tersimpan** meskipun program selesai.
- Python menyediakan modul `json` untuk **membaca** dan **menulis** data JSON.
- Dengan JSON, program daftar nilai siswa bisa menyimpan data permanen dan terus ditambahkan sesuai kebutuhan.