

Arsitektur Harvard, Arsitektur von Neumann, ISA, dan Mikro Arsitektur

1 Arsitektur von Neumann

John von Neumann pertama kali menawarkan arsitektur von Neumann pada tahun 1945, yang menjadi dasar hampir semua komputer modern. Arsitektur ini terutama memanfaatkan memori yang sama untuk menyimpan data dan instruksi program. Metode ini memungkinkan komputer untuk menjalankan program yang lebih kompleks dan dapat disesuaikan.

1.1 Keuntungan dan Kelemahan

1. Kelebihan:

- **Kesederhanaan Desain:**

- Satu Memori untuk Data dan Instruksi: Menggunakan satu ruang memori untuk menyimpan data dan instruksi, menyederhanakan desain sistem memori.
- Desain yang Sederhana: Satu set jalur data dan alamat membuat desain perangkat keras menjadi lebih sederhana dan lebih murah.

- **Fleksibilitas:**

- Pemrograman yang Fleksibel: Instruksi dan data dapat disimpan dalam memori yang sama, memungkinkan program untuk diubah selama runtime.
- Kemudahan Pengembangan *Software*: Perubahan pada kode atau data tidak memerlukan perubahan pada perangkat keras.

- **Penggunaan Memori yang Efisien:**

- Penggunaan Memori yang lebih mudah: Memori dapat dialokasikan untuk instruksi atau data sesuai kebutuhan, tanpa harus memisahkan ruang secara fisik.

2. Kekurangan:

- **Bottleneck Memori (Von Neumann Bottleneck):**

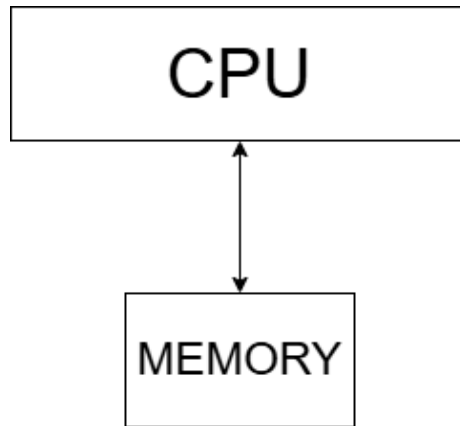


Figure 1: Van Neumann

- Kecepatan Transfer yang Terbatas: Satu jalur untuk mengakses memori instruksi dan data bisa menjadi hambatan ketika prosesor mencoba untuk melakukan banyak operasi secara bersamaan, memperlambat kinerja sistem.
- **Performa yang Kurang Optimal:**
 - Akses Memori yang Bersama: Proses pengambilan instruksi dan akses data harus dilakukan secara bergantian, yang dapat mengurangi kecepatan eksekusi jika kedua jenis operasi terjadi secara bersamaan.
- **Keamanan dan Stabilitas:**
 - Risiko Kesalahan Pengkodean: Karena data dan instruksi disimpan bersama, kesalahan dalam penanganan data atau instruksi dapat menyebabkan masalah seperti corrupt data atau instruksi.

2 Arsitektur Harvard

Arsitektur Harvard menggunakan dua ruang memori yang berbeda untuk menyimpan instruksi dan data, berbeda dengan arsitektur von Neumann. Ini memungkinkan pengambilan instruksi dan data secara bersamaan, yang dapat meningkatkan kinerja dan efisiensi sistem.

2.1 Kelebihan

1. Performa Lebih Baik:

- Akses Memori Paralel: Memiliki jalur terpisah untuk data dan instruksi memungkinkan akses simultan ke memori instruksi dan data,

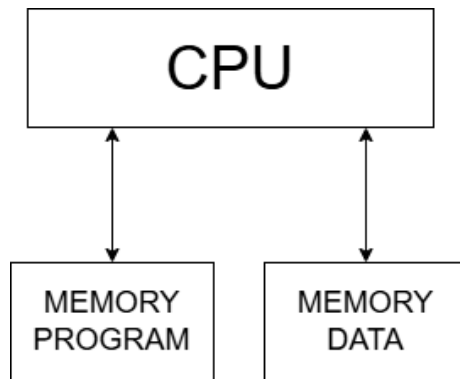


Figure 2: Harvard

meningkatkan kecepatan eksekusi dan mengurangi bottleneck memori.

- Eksekusi Instruksi yang Lebih Cepat: Dengan akses yang terpisah, *prosesor* dapat mengambil instruksi dan data secara bersamaan, mempercepat kinerja sistem.

2. Optimasi Memori:

- Pengaturan Memori yang Terpisah: Memori instruksi dan data yang terpisah memungkinkan optimasi yang lebih baik untuk masing-masing, seperti menggunakan cache yang berbeda untuk instruksi dan data.

3. Keamanan dan Stabilitas:

- Isolasi Data dan Instruksi: Dengan memisahkan data dan instruksi, risiko kesalahan pengkodean yang dapat menyebabkan data bisa rusak atau instruksi berkurang.

2.2 Kekurangan

1. Kompleksitas Desain:

- Desain yang Lebih sulit: memakai dua jalur memori terpisah dan pengaturan memori yang lebih kompleks, yang dapat meningkatkan biaya dan kompleksitas desain perangkat keras.
- Penggunaan Memori yang Tidak Fleksibel: Memori untuk instruksi dan data harus dialokasikan secara terpisah, yang bisa menyebabkan pemborosan jika salah satu jenis memori tidak digunakan sepenuhnya.

2. Kesulitan dalam Pengembangan:

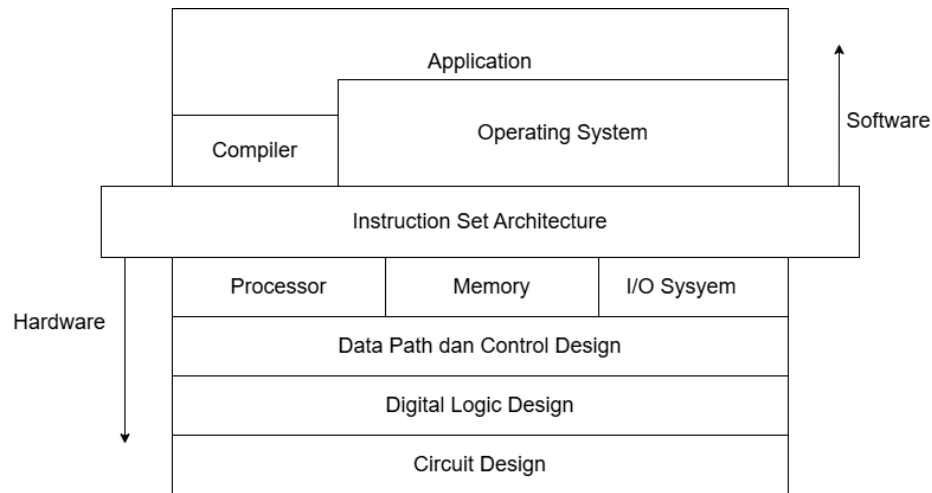


Figure 3: Arsitektur Set Instruksi

- Pengembangan Perangkat Lunak yang Lebih Kompleks: Memerlukan pemisahan antara data dan instruksi dalam pemrograman, yang dapat menambah kompleksitas pengembangan perangkat lunak.

3. Keterbatasan dalam Adaptasi:

- Kurang Fleksibel: Penambahan atau perubahan pada memori instruksi atau data mungkin memerlukan perubahan perangkat keras, mengurangi fleksibilitas dalam pembaruan atau peningkatan sistem.

3 Instruction Set Architecture (ISA)

(ISA) Instruction Set Architecture atau Arsitektur Set Instruksi adalah bagian dari arsitektur komputer yang menentukan cara kerja perangkat keras dan perangkat lunak dalam menjalankan instruksi program. ISA berperan sebagai jembatan antara perangkat lunak (software) dan perangkat keras (hardware) dalam sistem komputer.

3.1 Komponen Utama ISA

- Set Instruksi: Kumpulan perintah yang dapat dijalankan oleh *prosesor*.
- Mode Pengalamatan: Cara dalam menentukan lokasi data yang diperlukan oleh instruksi.
- Format Instruksi: Tata letak atau struktur instruksi di dalam memori.

3.2 Jenis-Jenis ISA

1. **CISC (*Complex Instruction Set Computing* atau *Komputasi Set Instruksi Kompleks*):** Set instruksi yang lebih kompleks dengan tujuan untuk mengurangi jumlah instruksi yang perlu dieksekusi oleh *prosesor*.
2. **RISC (*Reduced Instruction Set Computing* atau *Pengurangan Komputasi Set Instruksi*):** Set instruksi yang lebih sederhana, tetapi dapat dieksekusi dengan cepat oleh *prosesor*.

ISA itu penting karena menentukan performa *prosesor* serta kemampuannya dalam menjalankan berbagai aplikasi perangkat lunak.

4 Mikro Arsitektur

Mikro arsitektur adalah desain internal *prosesor* yang mencakup semua bagian, seperti unit kontrol, unit aritmatika dan logika (ALU), dan cache memori. Ini melibatkan implementasi fisik dari ISA. Mikro arsitektur bertujuan untuk membuat *prosesor* menangani dan menjalankan instruksi seefektif mungkin.

4.1 Komponen Utama Mikro Arsitektur

- ***Pipelining*:** Teknik untuk meningkatkan efisiensi *prosesor* dengan mengeksekusi beberapa instruksi sekaligus pada berbagai tahap pemrosesan.
- ***Branch Prediction*:** Teknologi yang memungkinkan *prosesor* untuk memprediksi cabang dari instruksi kondisional sehingga dapat mengurangi penundaan dalam eksekusi.

Mikro arsitektur berfungsi sebagai penghubung antara ISA dan desain perangkat keras, yang berarti bahwa implementasi fisik dari sebuah *prosesor* dapat bervariasi meskipun memiliki ISA yang sama.

5 Kesimpulan

Arsitektur sebuah sistem komputer sangat penting untuk kinerja, efisiensi, dan fleksibilitasnya. Setiap metode arsitektur von Neumann dan Harvard untuk menangani data dan instruksi memiliki kelebihan dan kekurangan. Sementara Arsitektur Set Instruksi berfungsi sebagai jembatan antara perangkat lunak dan perangkat keras, mikro arsitektur berkonsentrasi pada desain fisik *prosesor* untuk melaksanakan ISA. Memahami perbedaan dan manfaat masing-masing arsitektur ini sangat penting untuk mengembangkan dan mengoptimalkan sistem komputer.

6 Referensi

1. Hennessy, J. L., & Patterson, D. A. (2011). *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann.
2. Stallings, W. (2015). *Computer Organization and Architecture: Designing for Performance*. Pearson.
3. UNIKOM.M Zusane Oematan (2010). *Ringkasan Set Instruksi Dan Mode pengalamatan (Addressing Mode)*.