

Arsitektur Dan Organisasi Komputer

¹Veni Virda Hariati, ²Hendri Purnomo, ³Bayu Wibisana

Ilmu Komputer, Universitas Islam Al-Azhar

^{1,2,3}Alamat: jl.kampus unizar, Turida, barat, Kota Mataram, Nusa Tenggara Bar. 83233, Indonesia

venivirda23@gmail.com

Article Info

Article history:

Received ...

Revised ...

Accepted ...

ABSTRACT

Artikel ini membahas tentang konsep dasar arsitektur dan organisasi komputer, jenis-jenis arsitektur komputer yang umum digunakan, serta pentingnya pengetahuan tentang arsitektur dan organisasi komputer dalam pengembangan sistem komputer. Konsep dasar meliputi struktur fisik sistem komputer dan cara sumber daya sistem digunakan untuk menyelesaikan tugas tertentu. Jenis-jenis arsitektur komputer yang dibahas meliputi arsitektur Von Neumann, arsitektur Harvard, arsitektur RISC, dan arsitektur CISC. Arsitektur RISC dan CISC adalah dua pendekatan yang berbeda dalam merancang instruksi dan set instruksi dalam arsitektur komputer. Arsitektur RISC juga cenderung menggunakan register-register yang lebih banyak, sehingga mengurangi ketergantungan pada akses ke memori utama. Di sisi lain, arsitektur CISC menggunakan set instruksi yang lebih kompleks dan beragam. Instruksi-instruksi dalam arsitektur CISC dapat melakukan tugas yang lebih kompleks dalam satu instruksi, seperti operasi aritmatika yang kompleks atau akses langsung ke memori. Set instruksi CISC biasanya lebih besar dan beragam, termasuk instruksi-instruksi yang jarang digunakan. Meskipun instruksi-instruksi dalam arsitektur CISC dapat lebih kompleks, eksekusi instruksi biasanya membutuhkan lebih banyak siklus dan waktu. Kelebihan arsitektur RISC termasuk eksekusi instruksi yang lebih cepat dan efisien, konsistensi dalam ukuran dan format instruksi, serta ketergantungan yang lebih rendah pada memori utama. Arsitektur RISC juga cenderung lebih mudah diimplementasikan dalam perangkat keras dan memungkinkan pipelining yang efisien. Namun, kelemahan arsitektur RISC adalah jumlah instruksi yang terbatas, yang mungkin memerlukan lebih banyak instruksi untuk melakukan tugas yang kompleks. Sementara itu, kelebihan arsitektur CISC termasuk kemampuan untuk melakukan tugas yang kompleks dalam satu instruksi, dukungan untuk instruksi-instruksi yang jarang digunakan, dan ukuran kode yang lebih kecil. Arsitektur CISC juga dapat mengurangi jumlah instruksi yang diperlukan untuk melakukan tugas yang kompleks. Namun, kelemahan arsitektur CISC adalah kompleksitas eksekusi instruksi yang dapat mempengaruhi kinerja, serta ketergantungan yang lebih besar pada memori utama. Dalam praktiknya, kedua arsitektur ini memiliki aplikasi yang berbeda dalam sistem komputer modern. Arsitektur RISC cenderung digunakan dalam perangkat berperforma tinggi yang membutuhkan eksekusi instruksi yang cepat, seperti dalam superkomputer, server, atau mikrokontroler.

Keyword:

arsitektur komputer, RISC, CISC, instruksi.

I. PENDAHULUAN

Arsitektur dan organisasi komputer adalah dua konsep yang sangat penting dalam dunia computer dengan itu ada arsitektur RISC (Reduced Instruction Set Computer) dan CISC (Complex Instruction Set Computer). Masing-Masing arsitektur ini memiliki karakteristik yang unik dan mempengaruhi cara komputer menjalankan instruksi dan tugas tertentu dalam Arsitektur computer ini memiliki pendekatan yang berbeda dalam merancang instruksi dan set instruksi yang digunakan dalam komputer mencakup dalam artikel ini, peneliti akan membahas tentang arsitektur dan organisasi komputer, Dalam dunia komputer, arsitektur dan organisasi komputer adalah dua konsep yang sangat penting. Arsitektur komputer berhubungan dengan struktur fisik dari sistem komputer, seperti mikroprosesor, memori, dan perangkat keras lainnya.[1] Di sisi lain, organisasi komputer berkaitan dengan cara sumber daya sistem, seperti CPU, memori, dan perangkat input/output, diatur dan digunakan untuk menyelesaikan tugas tertentu. Dalam konteks arsitektur komputer, terdapat dua jenis arsitektur utama yang mempengaruhi cara komputer menjalankan instruksi dan tugas tertentu, yaitu RISC (Reduced Instruction Set Computer) dan CISC (Complex Instruction Set Computer). Setiap arsitektur memiliki karakteristik yang unik, termasuk konsep-konsep dasar, jenis-jenis arsitektur, serta pentingnya arsitektur dan organisasi komputer Arsitektur RISC didasarkan pada konsep penggunaan set instruksi yang lebih sederhana dan terbatas. Instruksi pada arsitektur RISC biasanya dieksekusi dalam satu siklus clock, sehingga mendorong kinerja yang lebih cepat. RISC juga cenderung memiliki jumlah register yang lebih banyak, menggunakan teknik register-windowing untuk mengoptimalkan penggunaan register. Contoh arsitektur [2] RISC populer adalah ARM dan MIPS. Di sisi lain, arsitektur CISC menggunakan set instruksi yang lebih kompleks dan bervariasi dalam ukuran. Instruksi pada arsitektur CISC mungkin membutuhkan beberapa siklus clock untuk dieksekusi. CISC memungkinkan instruksi yang kompleks dan lengkap dieksekusi dalam satu langkah, mengurangi jumlah instruksi yang diperlukan untuk mencapai suatu tugas. Contoh arsitektur CISC populer adalah x86 (Intel) dan AMD. computer dalam pengembangan sistem komputer. Selain RISC dan CISC, terdapat juga variasi arsitektur lainnya, seperti VLIW (Very Long Instruction Word) dan EPIC (Explicitly Parallel Instruction Computing). Arsitektur VLIW memanfaatkan instruksi-instruksi yang sangat panjang untuk mengizinkan paralelisme ekspisit dalam eksekusi instruksi. Sementara itu, arsitektur EPIC memanfaatkan informasi paralelisme yang dihasilkan oleh kompilator untuk meningkatkan kinerja eksekusi instruksi. Pentingnya pemahaman tentang arsitektur dan organisasi komputer dalam pengembangan sistem komputer tidak

dapat diabaikan. Pemahaman ini memungkinkan para insinyur dan pengembang untuk merancang sistem komputer yang efisien, dapat diandalkan, dan mampu menghadapi tuntutan aplikasi modern. Selain itu, pemahaman yang baik tentang arsitektur dan organisasi komputer juga membantu dalam pemecahan masalah, pemeliharaan, dan peningkatan kinerja sistem komputer.

[3]Teori Organisasi Arsitektur Komputer dan Praktik Assembler adalah bidang yang mencakup pemahaman tentang struktur dan desain sistem komputer, termasuk komponen-komponen seperti CPU, memori, perangkat masukan/keluaran, dan sistem interkoneksi. Praktik Assembler mengacu pada pemrograman pada tingkat rendah menggunakan bahasa assembly yang merupakan representasi langsung dari instruksi-instruksi yang dipahami oleh komputer.

II. METODE

Metode yang digunakan dalam artikel ini adalah studi RISC, CISC yaitu dengan mengumpulkan dan menganalisis sumber-sumber yang relevan mengenai arsitektur dan organisasi komputer. Sumber-sumber yang digunakan meliputi buku-buku, jurnal ilmiah, artikel online, dan sumber-sumber terkait lainnya. Metode RISC (Reduced Instruction Set Computing): Metode ini melibatkan pendekatan desain prosesor yang menggunakan set instruksi yang relatif sederhana, dengan penekanan pada eksekusi instruksi yang cepat dan efisien. RISC menghindari instruksi yang kompleks dan memprioritaskan penggunaan register dalam eksekusi instruksi. Metode RISC biasanya mengimplementasikan teknik pipelining untuk meningkatkan kinerja eksekusi instruksi. Metode CISC Metode ini melibatkan pendekatan desain prosesor yang menggunakan set instruksi yang lebih kompleks dan beragam, dengan instruksi yang dapat melakukan operasi yang lebih kompleks dalam satu instruksi. Metode CISC mengutamakan tingkat abstraksi yang tinggi dan memberikan fitur-fitur kaya pada instruksi. Namun, eksekusi instruksi CISC mungkin memerlukan lebih banyak siklus clock dibandingkan dengan RISC.

A. Pengolahan Data:

Mengorganisasi dan menyusun informasi dari sumber-sumber yang telah dianalisis. Identifikasi pola, temuan, dan persamaan antara sumber-sumber yang berbeda.

B. Interpretasi:

Menganalisis dan menafsirkan temuan dari sumber-sumber yang telah diproses. Membuat sintesis yang jelas dan terstruktur dari informasi yang ditemukan.

C. Penulisan:

Menyusun artikel berdasarkan temuan dan analisis yang telah dilakukan. Memastikan penggunaan kutipan dan

referensi yang tepat untuk menghormati sumber-sumber yang digunakan.

D. Revitalisasi:

Melakukan peninjauan ulang dan penyempurnaan artikel untuk memastikan kejelasan, konsistensi, dan kesesuaian dengan tujuan penelitian.

E. Pengujian Kesahihan:

Melakukan pengujian kesahihan terhadap temuan dan analisis yang disajikan dalam artikel. Ini bisa melibatkan verifikasi ulang dengan menggunakan sumber-sumber lain atau mengajukan pertanyaan kritis untuk memastikan bahwa informasi yang disajikan benar dan akurat.

F. Kesimpulan:

Menyimpulkan temuan dan hasil dari analisis literatur yang dilakukan. Mengaitkan temuan tersebut dengan tujuan penelitian dan memberikan ringkasan yang jelas tentang kontribusi artikel terhadap pemahaman arsitektur dan organisasi komputer.

G. Referensi:

Mengacu pada semua sumber yang digunakan dalam artikel sesuai dengan aturan penulisan referensi yang berlaku. Memastikan keakuratan dan kelengkapan informasi mengenai sumber-sumber yang digunakan.

H. Pembaharuan:

Dalam kasus di mana artikel tersebut akan dipublikasikan atau digunakan sebagai referensi, periksalah kebijakan dan pedoman terkait dengan penulisan artikel. Lakukan revisi atau pembaharuan yang diperlukan untuk memenuhi persyaratan dan standar yang berlaku.

I. Evaluasi Kualitas Sumber:

Selama proses studi literatur, penting untuk melakukan evaluasi terhadap kualitas sumber yang digunakan. Pertimbangkan reputasi dan kredibilitas penulis, keakuratan informasi, metode penelitian yang digunakan, dan relevansi sumber dengan topik yang diteliti. Hal ini penting untuk memastikan bahwa informasi yang digunakan dalam artikel Anda bersumber dari sumber yang terpercaya dan berkualitas.

J. Sinergi dan Integrasi:

Selama analisis sumber, cari keterkaitan dan pola yang muncul antara berbagai sumber yang digunakan. Identifikasi persamaan, perbedaan, dan kontribusi unik dari masing-masing sumber. Integrasi informasi dari berbagai sumber akan membantu membangun pemahaman yang lebih lengkap dan menyeluruh tentang arsitektur dan organisasi komputer.

K. Analisis Kritis:

Selama proses studi literatur, penting untuk menerapkan analisis kritis terhadap informasi yang ditemukan. Evaluasilah kelemahan, kekuatan, atau bias potensial dalam sumber-sumber yang digunakan. Pemahaman yang mendalam tentang konteks dan implikasi dari setiap temuan akan memperkaya artikel Anda.

L. Perbandingan dan Kontras:

Selama analisis sumber, perhatikan perbedaan dan persamaan antara arsitektur RISC (Reduced Instruction Set Computer) dan CISC (Complex Instruction Set Computer). Teliti bagaimana kedua arsitektur ini mempengaruhi cara komputer menjalankan instruksi dan tugas tertentu. Jelaskan karakteristik unik dari masing-masing arsitektur dan dampaknya pada pengembangan sistem komputer.

M. Penggalan Temuan:

Identifikasi temuan atau kontribusi yang signifikan dari studi literatur yang dilakukan. Apakah ada tren, pola, atau wawasan baru yang muncul? Tinjau temuan Anda secara kritis dan jelaskan implikasinya dalam konteks arsitektur dan organisasi komputer.

O. Pembahasan Implikasi:

Dalam artikel Anda, diskusikan implikasi temuan yang dihasilkan dari studi literatur. Tinjau bagaimana arsitektur dan organisasi komputer memengaruhi pengembangan sistem komputer secara keseluruhan. Bahas keuntungan dan tantangan masing-masing arsitektur dan bagaimana hal itu dapat mempengaruhi desain, performa, dan efisiensi sistem.

P. Kesimpulan dan Rekomendasi:

Sintesis temuan utama dari artikel Anda dan sampaikan kesimpulan yang kuat berdasarkan studi literatur yang dilakukan. Berikan rekomendasi untuk pengembangan arsitektur dan organisasi komputer di masa depan atau area penelitian yang perlu ditgekspor lebih lanjut.

A. Arsitektur RISC

Metode arsitektur RISC (Reduced Instruction Set Computing) adalah sebuah pendekatan dalam arsitektur komputer yang memprioritaskan kecepatan eksekusi instruksi. Arsitektur RISC memperkecil jumlah instruksi yang diperlukan untuk menjalankan program, sehingga proses eksekusi instruksi menjadi lebih cepat.

B. Arsitektur CISC

Metode arsitektur CISC (Complex Instruction Set Computing) adalah sebuah pendekatan dalam arsitektur komputer yang menambahkan berbagai instruksi kompleks untuk menjalankan program secara efektif. Arsitektur CISC memperkenalkan banyak instruksi kompleks yang dapat melakukan banyak tugas dalam satu instruksi, sehingga program menjadi lebih singkat dan mudah dibaca.

C. Studi Literatur:

Artikel ini akan menggunakan pendekatan studi literatur yang melibatkan analisis terhadap literatur terkait, seperti jurnal ilmiah, buku teks, makalah konferensi, dan sumber-sumber akademik terpercaya lainnya. Melalui studi literatur yang komprehensif, kita dapat mengumpulkan informasi yang relevan tentang arsitektur RISC dan CISC, serta memperoleh pemahaman yang mendalam tentang perbedaan, kelebihan, dan kelemahan masing-masing pendekatan.

D Analisis Perbandingan:

Artikel ini akan melakukan analisis perbandingan secara komprehensif antara arsitektur RISC dan CISC berdasarkan karakteristik kunci yang mempengaruhi kinerja dan desain komputer. Beberapa faktor yang akan dianalisis meliputi kompleksitas instruksi, kecepatan eksekusi, penggunaan memori, dan fleksibilitas dalam menjalankan tugas-tugas komputasi yang kompleks. Analisis ini akan membantu pembaca memahami perbedaan dan implikasi praktis dari kedua pendekatan.

E Studi Kasus:

Artikel ini juga akan mencakup studi kasus yang mengilustrasikan penggunaan arsitektur RISC dan CISC dalam aplikasi nyata. Contoh-contoh ini dapat meliputi penggunaan arsitektur RISC pada perangkat mobile, dan penggunaan arsitektur CISC pada komputer desktop atau server. Dengan melibatkan studi kasus, kita dapat melihat bagaimana kelebihan dan kelemahan masing-masing arsitektur dapat mempengaruhi performa dan efisiensi sistem dalam situasi yang nyata.

F Analisis Kinerja:

Artikel ini akan melakukan analisis kinerja antara arsitektur RISC dan CISC menggunakan metode perbandingan yang valid dan terukur. Analisis ini dapat melibatkan pengujian dan pengukuran performa menggunakan perangkat lunak simulasi atau komputer fisik. Dengan analisis kinerja yang objektif, pembaca dapat mendapatkan pemahaman yang lebih baik tentang perbedaan performa antara kedua arsitektur.

G Tinjauan Terhadap Perkembangan Terkini:

Artikel ini juga akan menyajikan tinjauan terhadap perkembangan terkini dalam arsitektur komputer yang berkaitan dengan RISC dan CISC. Ini mencakup perkembangan seperti arsitektur hybrid, penggunaan teknik superskalar, VLIW, dan perkembangan terbaru lainnya yang berusaha menggabungkan keuntungan dari kedua arsitektur. Tinjauan ini memberikan wawasan tentang arah masa depan arsitektur komputer dan implikasinya terhadap

1. **Pipelining:** Metode ini digunakan dalam kedua arsitektur RISC dan CISC untuk meningkatkan kinerja eksekusi instruksi dengan membagi jalur eksekusi menjadi beberapa tahap. Setiap tahap memproses instruksi yang berbeda secara bersamaan, sehingga meningkatkan throughput (jumlah instruksi yang dapat dieksekusi dalam satu waktu).
2. **Superscalar Execution:** Metode ini digunakan dalam arsitektur CISC untuk menjalankan beberapa instruksi dalam satu siklus clock. Prosesor superscalar memiliki beberapa unit fungsional yang dapat mengeksekusi instruksi secara independen, seperti unit ALU (Arithmetic Logic Unit), unit pemrosesan floating-point, dan unit kontrol.

3. **Out-of-Order Execution:** Metode ini digunakan dalam arsitektur CISC untuk meningkatkan penggunaan sumber daya prosesor dengan mengeksekusi instruksi dalam urutan yang tidak tergantung pada ketergantungan data. Prosesor dengan out-of-order execution dapat menjalankan instruksi yang tidak memiliki ketergantungan data sebelum instruksi yang memiliki ketergantungan data selesai dieksekusi.

4. **Branch Prediction:** Metode ini digunakan dalam kedua arsitektur RISC dan CISC untuk mengurangi latensi yang disebabkan oleh instruksi percabangan (branch instructions). Prosesor menggunakan algoritma dan tabel prediksi untuk memprediksi apakah suatu branch akan diambil atau tidak diambil, sehingga memungkinkan eksekusi instruksi tetap berlanjut tanpa menunggu hasil pengambilan branch.

VLIW (Very Long Instruction Word): Metode ini digunakan dalam beberapa implementasi arsitektur RISC untuk meningkatkan tingkat paralelisme instruksi. Dalam VLIW, sebuah instruksi terdiri dari beberapa operasi/instruksi yang dieksekusi secara bersamaan dalam satu siklus clock. Compiler bertanggung jawab untuk mengatur instruksi-instruksi tersebut agar dapat dieksekusi secara paralel.

SIMD (Single Instruction, Multiple Data): Metode ini digunakan dalam kedua arsitektur RISC dan CISC untuk melakukan operasi yang sama pada multiple data secara paralel. Instruksi SIMD mengizinkan prosesor untuk melakukan operasi serupa pada set data yang lebih besar, seperti melakukan operasi aritmetika pada sekelompok bilangan dalam satu instruksi.

Memory Hierarchy: Metode ini berhubungan dengan organisasi komputer untuk meningkatkan kinerja akses ke memori. Arsitektur RISC dan CISC memanfaatkan hierarki memori yang terdiri dari cache level tinggi, cache level menengah, dan memori utama. Tujuan dari memori hierarki adalah untuk menyimpan data yang sering diakses secara lokal agar dapat diakses dengan cepat.

Branch Target Buffer (BTB): Metode ini digunakan untuk mempercepat pengambilan instruksi pada percabangan (branch). BTB menyimpan target dari percabangan sebelumnya yang dieksekusi, sehingga jika ada instruksi percabangan dengan alamat yang sama, prosesor dapat secara cepat menentukan target percabangan tersebut.

Register Renaming: Metode ini digunakan dalam beberapa implementasi arsitektur RISC dan CISC untuk mengatasi keterbatasan jumlah register dengan secara dinamis memetakan register sementara (temporary register) ke register fisik yang tersedia. Metode ini digunakan dalam beberapa implementasi

arsitektur RISC. Ketika terjadi instruksi percabangan, instruksi yang ada di slot delay (berikutnya setelah instruksi percabangan) dieksekusi terlebih dahulu sebelum hasil percabangan diketahui. Tujuannya adalah untuk mengisi waktu yang mungkin terbuang karena penundaan dalam pengambilan percabangan.

Microcoding: Metode ini digunakan dalam arsitektur CISC untuk mengimplementasikan instruksi kompleks dalam bentuk serangkaian instruksi yang lebih sederhana yang disebut mikroinstruksi. Mikroinstruksi ini kemudian dieksekusi oleh unit kontrol mikro dalam prosesor.

Hardware Acceleration: Metode ini melibatkan penggunaan perangkat keras khusus, seperti coprocessor atau unit pemrosesan grafis (GPU), untuk meningkatkan kinerja komputasi pada tugas-tugas tertentu.

Branch Target Prediction: Metode ini merupakan pengembangan lebih lanjut dari branch prediction. Algoritma dan mekanisme branch target prediction digunakan untuk memprediksi alamat target dari suatu percabangan. Ini membantu menghindari penundaan yang disebabkan oleh percabangan dan memungkinkan eksekusi instruksi yang lebih lancar.

Dynamic Instruction Scheduling: Metode ini digunakan dalam arsitektur RISC untuk mengoptimalkan urutan eksekusi instruksi secara dinamis. Prosesor dapat memindahkan instruksi yang tidak memiliki ketergantungan data ke depan untuk mengisi slot yang kosong di pipelining dan meningkatkan penggunaan unit eksekusi.

Superscalar Out-of-Order Execution: Metode ini merupakan kombinasi dari superscalar execution dan out-of-order execution. Prosesor superscalar out-of-order execution mampu menjalankan beberapa instruksi dalam satu siklus clock dan mengubah urutan eksekusi instruksi untuk memaksimalkan penggunaan sumber daya.

Vector Processing: Metode ini digunakan dalam arsitektur CISC dan RISC untuk melakukan operasi pada vektor atau larik data dalam satu instruksi. Prosesor yang mendukung vector processing memiliki unit khusus untuk menjalankan operasi vektor dengan cepat dan efisien.

[4] **Topologi Jaringan:** Metode yang digunakan untuk menentukan topologi atau struktur fisik jaringan komputer. Beberapa metode yang umum digunakan meliputi topologi bus, topologi star, topologi ring, topologi mesh, dan topologi tree. Pemilihan metode topologi harus disesuaikan dengan kebutuhan dan karakteristik jaringan pada Hotel Amaris Palembang.

Segmentasi Jaringan: Metode yang digunakan untuk membagi jaringan menjadi beberapa segmen logis atau subnet. Segmentasi jaringan dapat membantu meningkatkan performa dan keamanan jaringan

dengan membatasi lalu lintas jaringan dalam segmen-segmen yang terpisah.

Protokol Jaringan: Metode yang digunakan untuk menentukan protokol yang akan digunakan dalam jaringan komputer. Protokol jaringan mengatur komunikasi antar perangkat dan memastikan pengiriman data yang efisien dan andal. Contoh protokol yang umum digunakan adalah TCP/IP (Transmission Control Protocol/Internet Protocol) untuk komunikasi dalam jaringan internet.

Keamanan Jaringan: Metode yang digunakan untuk menjaga keamanan jaringan komputer. Ini melibatkan penerapan langkah-langkah keamanan seperti firewall, VPN (Virtual Private Network), enkripsi data, dan manajemen akses yang tepat. Metode keamanan yang dipilih harus sesuai dengan kebutuhan Hotel Amaris Palembang dan melindungi data dan sistem jaringan dari ancaman yang mungkin terjadi.

III. Hasil dan Pembahasan

Arsitektur dan Organisasi Komputer

Arsitektur dan organisasi komputer adalah dua konsep yang sangat penting dalam dunia komputer. Arsitektur komputer mencakup struktur fisik dari sistem komputer, sedangkan organisasi komputer mencakup cara di mana sumber daya sistem digunakan untuk menyelesaikan tugas tertentu. Dalam artikel ini, kita akan membahas tentang arsitektur dan organisasi komputer, termasuk konsep-konsep dasar, jenis-jenis arsitektur, serta pentingnya arsitektur dan organisasi komputer dalam pengembangan sistem komputer.

Konsep Dasar

Arsitektur komputer mencakup semua bagian fisik dari sistem komputer, termasuk perangkat keras, perangkat lunak, dan sistem operasi. Sementara itu, organisasi komputer mencakup cara di mana sumber daya sistem, seperti CPU, memori, dan sistem input/output, digunakan untuk menyelesaikan tugas-tugas tertentu.

Jenis-jenis Arsitektur

Terdapat beberapa jenis arsitektur komputer yang umum digunakan, antara lain:

Arsitektur Von Neumann: merupakan arsitektur komputer yang paling umum digunakan. Arsitektur ini memiliki satu bus utama yang menghubungkan CPU, memori, dan perangkat input/output.

Arsitektur Harvard: mirip dengan arsitektur Von Neumann, namun memiliki memori yang terpisah untuk instruksi dan data. **Arsitektur RISC (Reduced Instruction Set Computer):** merupakan arsitektur yang digunakan untuk mengoptimalkan kinerja sistem dengan membatasi jumlah instruksi yang digunakan. **Arsitektur CISC (Complex Instruction Set Computer):** merupakan arsitektur yang memungkinkan penggunaan instruksi yang lebih kompleks untuk meningkatkan kinerja sistem.

Pentingnya tentang Arsitektur dan Organisasi Komputer dalam Pengembangan Sistem Komputer, pengetahuan tentang arsitektur dan organisasi komputer sangat penting dalam pengembangan sistem komputer. Dengan memahami konsep, jenis, dan pentingnya arsitektur dan organisasi komputer, pengembang sistem komputer dapat merancang sistem yang dapat berfungsi dengan baik, meningkatkan performa sistem komputer, dan mengoptimalkan penggunaan sumber daya sistem. Sebagai contoh, pemahaman tentang arsitektur dan organisasi komputer dapat membantu pengembang sistem komputer untuk memilih komponen sistem yang tepat dan merancang sistem yang dapat berfungsi dengan baik.

Konsep dasar arsitektur dan organisasi komputer, yaitu struktur fisik dari sistem komputer dan cara sumber daya sistem digunakan untuk menyelesaikan tugas tertentu.

Jenis-jenis arsitektur komputer yang umum digunakan, seperti arsitektur Von Neumann, arsitektur Harvard, arsitektur RISC, dan arsitektur CISC.

Pentingnya pengetahuan tentang arsitektur dan organisasi komputer dalam pengembangan sistem komputer, seperti merancang sistem yang dapat berfungsi dengan baik, meningkatkan performa sistem komputer, dan mengoptimalkan penggunaan sumber daya sistem.

Dalam artikel yang membahas tentang arsitektur dan organisasi komputer, akan dibahas secara lebih mendetail tentang konsep-konsep tersebut, jenis-jenis arsitektur komputer, serta pentingnya pengetahuan tentang arsitektur dan organisasi komputer dalam pengembangan system.

[5]Arsitektur dan Organisasi Komputer RISC (Reduced Instruction Set Computing):

Metode pipelining pada arsitektur RISC membagi jalur eksekusi menjadi beberapa tahap, seperti fetch, decode, execute, dan write-back. Ini memungkinkan instruksi-instruksi berbeda dieksekusi secara bersamaan, meningkatkan throughput.

Register windowing pada beberapa implementasi arsitektur RISC, seperti SPARC, membagi kumpulan register menjadi beberapa "jendela" terkait dengan prosedur atau fungsi. Metode ini meningkatkan kinerja dengan mengurangi akses ke memori utama.

VLW (Very Long Instruction Word) pada arsitektur RISC memungkinkan beberapa operasi/instruksi dieksekusi secara bersamaan dalam satu siklus clock. Compiler bertanggung jawab untuk mengatur instruksi-instruksi tersebut agar dapat dieksekusi secara paralel.

[6]Arsitektur dan Organisasi Komputer CISC (Complex Instruction Set Computing):

Metode superscalar execution pada arsitektur CISC memungkinkan beberapa instruksi dieksekusi dalam satu siklus clock. Ini dilakukan dengan menggunakan beberapa unit fungsional yang dapat mengeksekusi instruksi secara independen.

Out-of-order execution pada arsitektur CISC memungkinkan instruksi dieksekusi dalam urutan yang

tidak tergantung pada ketergantungan data. Hal ini meningkatkan penggunaan sumber daya prosesor dengan menjalankan instruksi yang tidak memiliki ketergantungan data sebelum instruksi yang memiliki ketergantungan data selesai dieksekusi.

[7]Pembahasan:

RISC (Reduced Instruction Set Computing) adalah pendekatan arsitektur komputer yang berfokus pada penggunaan set instruksi yang sederhana, tetapi lebih sedikit, dengan penekanan pada eksekusi instruksi yang cepat. RISC cenderung memiliki pipelining yang baik, set instruksi yang seragam, dan penekanan pada akses ke register.

CISC (Complex Instruction Set Computing) adalah pendekatan arsitektur komputer yang menggunakan set instruksi yang lebih kompleks dan beragam. CISC cenderung memiliki instruksi yang lebih rumit, tetapi lebih sedikit instruksi yang diperlukan untuk mencapai tugas tertentu.

Metode yang digunakan dalam arsitektur RISC dan CISC bertujuan untuk meningkatkan kinerja, efisiensi, dan fleksibilitas eksekusi instruksi. Pipelining, superscalar execution, out-of-order execution, dan register windowing adalah beberapa metode yang digunakan dalam keduanya untuk mencapai tujuan ini.

Metode RISC (Reduced Instruction Set Computing):

Hasil: Metode RISC bertujuan untuk memberikan eksekusi instruksi yang cepat dan efisien dengan menggunakan set instruksi yang relatif sederhana. Prosesor RISC memiliki sedikit instruksi yang dieksekusi dalam satu siklus clock.

Pembahasan: RISC mengutamakan efisiensi dan kemampuan untuk melakukan pipelining, yaitu menjalankan beberapa tahap eksekusi instruksi secara bersamaan. Pendekatan ini memungkinkan eksekusi instruksi yang lebih cepat dan meningkatkan kinerja prosesor.

Metode CISC (Complex Instruction Set Computing):

Hasil: Metode CISC memiliki set instruksi yang lebih kompleks dan beragam, dengan instruksi yang dapat melakukan operasi yang lebih kompleks dalam satu instruksi. Prosesor CISC dapat menjalankan instruksi yang lebih kompleks dengan satu instruksi saja.

Pembahasan: CISC mengutamakan tingkat abstraksi yang tinggi, sehingga memungkinkan programmer untuk menulis kode yang lebih singkat dan lebih mudah dibaca. Metode ini biasanya digunakan dalam lingkungan **pemrograman tingkat tinggi**.

Metode Pipelining:

Hasil: Metode pipelining memungkinkan instruksi-instruksi dieksekusi secara paralel dalam beberapa tahap, sehingga meningkatkan throughput dan kinerja prosesor.

Pembahasan: Dalam pipelining, setiap tahap melibatkan eksekusi dari instruksi yang berbeda secara bersamaan. Namun, adanya ketergantungan antara instruksi dapat menyebabkan bottleneck dan mengurangi keuntungan

pipelining. Selain itu, kemungkinan branch misprediction juga dapat mengganggu kinerja pipelining.

Metode SIMD (Single Instruction, Multiple Data):

Hasil: Metode SIMD memungkinkan eksekusi operasi yang sama pada beberapa data secara paralel dalam satu instruksi.

Pembahasan: SIMD cocok digunakan dalam aplikasi yang membutuhkan pemrosesan data dalam skala besar, seperti pemrosesan gambar, video, dan pemrosesan paralel lainnya. Metode ini memberikan kinerja yang tinggi dengan memanfaatkan keparalelan eksekusi instruksi.

Metode Multithreading:

Hasil: Metode multithreading memungkinkan prosesor menjalankan beberapa thread secara bersamaan, meningkatkan kinerja dan responsifitas sistem.

Pembahasan: Dalam multithreading, setiap thread memiliki urutan eksekusi yang berbeda dan dapat memanfaatkan idle time prosesor. Metode ini cocok digunakan dalam aplikasi yang membutuhkan tingkat paralelisme tinggi dan responsifitas yang cepat.

Metode Caching:

Hasil: Metode caching bertujuan untuk mengurangi waktu akses ke memori utama dengan menggunakan lapisan penyimpanan yang lebih cepat dan dekat dengan prosesor.

[8]Arsitektur RISC cenderung lebih cocok untuk aplikasi yang membutuhkan eksekusi instruksi yang cepat dan paralel, seperti pemrosesan data dalam skala besar, pemrosesan grafis, dan pemrosesan media. Dengan set instruksi yang lebih sederhana dan fokus pada penggunaan register, RISC dapat mencapai kinerja yang tinggi dengan memanfaatkan teknik seperti pipelining dan VLIW.

Di sisi lain, arsitektur CISC cenderung lebih cocok untuk aplikasi yang membutuhkan instruksi yang kompleks dan memiliki tingkat abstraksi yang tinggi, seperti pemrograman tingkat tinggi dan pemrosesan data yang membutuhkan instruksi kaya fitur. Dengan instruksi yang lebih kompleks, CISC dapat mengurangi jumlah instruksi yang diperlukan untuk menyelesaikan tugas tertentu. Keduanya memiliki kelebihan dan kekurangan masing-masing. RISC cenderung memiliki latensi instruksi yang lebih rendah, konsumsi daya yang lebih rendah, dan lebih mudah diimplementasikan secara fisik. Sementara itu, CISC dapat mengurangi kebutuhan akan instruksi yang kompleks dan mengurangi penggunaan memori. Dalam perkembangan terkini, batasan antara RISC dan CISC telah menjadi lebih kabur dengan munculnya arsitektur hybrid atau arsitektur yang memadukan fitur-fitur dari keduanya. Beberapa prosesor modern menggabungkan keuntungan pipelining dan paralelisme instruksi RISC dengan instruksi yang lebih kompleks dan instruksi vektor dari CISC. Penting untuk dicatat bahwa pilihan arsitektur (RISC atau CISC) dapat dipengaruhi oleh berbagai faktor, termasuk kebutuhan aplikasi, performa yang diinginkan, kompleksitas pengembangan perangkat lunak, dan ketersediaan sumber daya. Selain itu, perkembangan

teknologi terus mengubah dan memperluas fitur-fitur yang tersedia dalam arsitektur dan organisasi komputer.

[9]Meskipun RISC dan CISC adalah pendekatan yang berbeda dalam merancang arsitektur dan organisasi komputer, keduanya memiliki tujuan yang sama, yaitu meningkatkan kinerja, efisiensi, dan fleksibilitas eksekusi instruksi. Dalam beberapa kasus, perbedaan antara RISC dan CISC menjadi kurang signifikan seiring dengan perkembangan teknologi. Beberapa arsitektur modern, seperti x86-64, yang dikenal sebagai CISC, telah mengadopsi beberapa fitur RISC dalam implementasinya, seperti pipelining dan out-of-order execution. Salah satu tren terbaru dalam arsitektur komputer adalah meningkatnya fokus pada energi yang efisien dan komputasi yang cepat dengan memanfaatkan fitur-fitur dari kedua pendekatan. Perkembangan seperti instruksi vektor, teknologi multithreading, dan desain prosesor dengan konsumsi daya yang rendah telah membawa perpaduan antara RISC dan CISC yang lebih maju.

1. Analisis kebutuhan: Melakukan analisis kebutuhan jaringan komputer yang spesifik untuk Hotel Amaris Palembang. Ini termasuk memahami jumlah pengguna, jenis aplikasi yang akan digunakan, kecepatan dan kapasitas yang diperlukan, dan persyaratan keamanan.

2. Konfigurasi jaringan: Mengkonfigurasi perangkat jaringan dengan menggunakan metode rekayasa sistem jaringan komputer yang tepat. Ini meliputi pengaturan IP address, routing, VLAN, pengamanan jaringan, dan pengaturan layanan jaringan seperti DHCP (Dynamic Host Configuration Protocol) dan DNS (Domain Name System).

3. Pengujian dan pemeliharaan: Melakukan pengujian untuk memastikan kinerja dan keamanan jaringan yang baik. Selain itu, melakukan pemeliharaan rutin seperti pemantauan jaringan, pembaruan perangkat lunak, dan penanganan masalah yang muncul.

Analisis Perbedaan RISC dan CISC:

Mengidentifikasi perbedaan utama dalam desain dan karakteristik arsitektur RISC dan CISC.

Menganalisis konsep dasar dan prinsip desain yang digunakan dalam kedua arsitektur.

Menyoroti perbedaan dalam kompleksitas instruksi, ukuran kode, dan kecepatan eksekusi.

[10]Kelebihan dan Kelemahan Arsitektur RISC:

Menjelaskan keuntungan arsitektur RISC dalam hal kecepatan eksekusi instruksi sederhana.

Menggambarkan penggunaan sumber daya perangkat keras yang efisien.

Menganalisis kelemahan seperti kebutuhan akan kode yang lebih banyak dan keterbatasan dalam menjalankan instruksi kompleks.

Kelebihan dan Kelemahan Arsitektur CISC:

Menjelaskan keuntungan arsitektur CISC dalam hal kemampuan menjalankan instruksi kompleks dalam satu siklus.

Menganalisis penggunaan memori yang lebih efisien dengan instruksi yang lebih kompleks.

Menggambarkan kelemahan seperti kompleksitas perangkat keras dan potensi keterbatasan dalam kecepatan eksekusi.

Perbandingan Kinerja RISC dan CISC:

Melakukan analisis kinerja komparatif menggunakan pengujian dan pengukuran.

Membahas contoh kasus penggunaan RISC dan CISC dalam aplikasi tertentu.

Menyajikan data dan grafik untuk membandingkan kinerja dalam berbagai situasi.

Perkembangan Terbaru dalam Arsitektur Komputer:

Menggambarkan perkembangan terkini dalam arsitektur komputer yang berhubungan dengan RISC dan CISC.

Menyajikan informasi tentang arsitektur hybrid dan teknik lain yang menggabungkan elemen-elemen RISC dan CISC. Membahas implikasi perkembangan tersebut dalam perbandingan RISC dan CISC.

Arsitektur RISC cenderung lebih cocok untuk aplikasi yang membutuhkan eksekusi instruksi yang cepat dan paralel, seperti pemrosesan data dalam skala besar, pemrosesan grafis, dan pemrosesan media. Dengan set instruksi yang lebih sederhana dan fokus pada penggunaan register, RISC dapat mencapai kinerja yang tinggi dengan memanfaatkan teknik seperti pipelining dan VLIW.

Keduanya memiliki kelebihan dan kekurangan masing-masing. RISC cenderung memiliki latensi instruksi yang lebih rendah, konsumsi daya yang lebih rendah, dan lebih mudah diimplementasikan secara fisik. Sementara itu, CISC dapat mengurangi kebutuhan akan instruksi yang kompleks dan mengurangi penggunaan memori.

Dalam perkembangan terkini, batasan antara RISC dan CISC telah menjadi lebih kabur dengan munculnya arsitektur hybrid atau arsitektur yang memadukan fitur-fitur dari keduanya. Beberapa prosesor modern menggabungkan keuntungan pipelining dan paralelisme instruksi RISC dengan instruksi yang lebih kompleks dan instruksi vektor dari CISC.

Penting untuk dicatat bahwa pilihan arsitektur (RISC atau CISC) dapat dipengaruhi oleh berbagai faktor, termasuk kebutuhan aplikasi, performa yang diinginkan, kompleksitas pengembangan perangkat lunak, dan ketersediaan sumber daya. Selain itu, perkembangan teknologi terus mengubah dan memperluas fitur-fitur yang tersedia dalam arsitektur dan organisasi komputer.

V. KESIMPULAN

Dari pembahasan di atas, dapat disimpulkan bahwa arsitektur dan organisasi komputer sangat penting dalam pengembangan sistem komputer. Konsep, jenis, dan pentingnya arsitektur dan organisasi komputer dapat membantu pengembang sistem komputer untuk merancang sistem yang dapat berfungsi dengan baik, meningkatkan performa sistem komputer, dan mengoptimalkan

penggunaan sumber daya sistem. Arsitektur komputer mencakup rancangan instruksi dan set instruksi yang digunakan dalam komputer. Konsep dasar arsitektur komputer meliputi ukuran dan format instruksi, jumlah register yang tersedia, mode addressing, dan mekanisme interaksi antara instruksi dan perangkat keras.

Arsitektur komputer dan organisasi komputer melibatkan desain dan pengorganisasian komponen-komponen dalam sistem komputer untuk mencapai kinerja, efisiensi, dan responsifitas yang diinginkan.

Metode RISC (Reduced Instruction Set Computing) menggunakan set instruksi yang sederhana dan fokus pada eksekusi instruksi yang cepat dan efisien. Metode ini dapat meningkatkan kinerja prosesor dengan pendekatan pipelining.

Metode CISC (Complex Instruction Set Computing) menggunakan set instruksi yang kompleks dan beragam, memungkinkan instruksi yang lebih kompleks dieksekusi dalam satu instruksi. Metode ini menawarkan tingkat abstraksi tinggi, membuat pemrograman lebih mudah, dan biasanya digunakan dalam lingkungan pemrograman tingkat tinggi.

Pipelining memungkinkan instruksi-instruksi dieksekusi secara paralel dalam beberapa tahap, meningkatkan throughput dan kinerja prosesor.

SIMD (Single Instruction, Multiple Data) memungkinkan operasi yang sama pada beberapa data dieksekusi secara paralel dalam satu instruksi, cocok digunakan dalam pemrosesan data dalam skala besar.

Kelebihan Arsitektur RISC:

RISC memiliki instruksi yang sederhana, sehingga dapat dieksekusi dengan cepat.

RISC memungkinkan penggunaan sumber daya perangkat keras yang lebih efisien.

RISC cocok untuk aplikasi dengan instruksi yang relatif sederhana, seperti perangkat mobile.

Kelebihan Arsitektur CISC:

CISC dapat menjalankan instruksi kompleks dalam satu siklus, sehingga lebih efisien dalam beberapa kasus penggunaan.

CISC memungkinkan penggunaan memori yang lebih efisien dengan instruksi yang lebih kompleks.

CISC lebih cocok untuk aplikasi dengan instruksi yang kompleks, seperti komputer desktop atau server.

1. Arsitektur dan organisasi komputer RISC (Reduced Instruction Set Computing) dan CISC (Complex Instruction Set Computing) adalah dua pendekatan yang berbeda dalam merancang prosesor dan set instruksi.

2. RISC lebih fokus pada penggunaan set instruksi yang sederhana, eksekusi instruksi yang cepat, pipelining yang efisien, dan penekanan pada akses ke register. RISC cocok untuk aplikasi yang membutuhkan eksekusi instruksi yang cepat dan paralel, seperti pemrosesan data dalam skala besar, pemrosesan grafis, dan pemrosesan media.

3. CISC menggunakan set instruksi yang lebih kompleks dan beragam, dengan instruksi yang lebih rumit tetapi lebih sedikit instruksi yang diperlukan untuk tugas tertentu. CISC cocok untuk aplikasi yang membutuhkan instruksi yang kompleks dan tingkat abstraksi yang tinggi, seperti pemrograman tingkat tinggi dan pemrosesan data yang membutuhkan instruksi kaya fitur. Arsitektur komputer dan organisasi komputer melibatkan desain dan pengorganisasian komponen-komponen dalam sistem komputer untuk mencapai kinerja, efisiensi, dan responsifitas yang diinginkan. Metode RISC (Reduced Instruction Set Computing) menggunakan set instruksi yang sederhana dan fokus pada eksekusi instruksi yang cepat dan efisien. Metode ini dapat meningkatkan kinerja prosesor dengan pendekatan pipelining. Metode CISC (Complex Instruction Set Computing) menggunakan set instruksi yang kompleks dan beragam, memungkinkan instruksi yang lebih kompleks dieksekusi dalam satu instruksi. Metode ini menawarkan tingkat abstraksi tinggi, membuat pemrograman lebih mudah, dan biasanya digunakan dalam lingkungan pemrograman tingkat tinggi. Pipelining memungkinkan instruksi-instruksi dieksekusi secara paralel dalam beberapa tahap, meningkatkan throughput dan kinerja prosesor. Namun, ketergantungan antar instruksi dan branch misprediction dapat mempengaruhi kinerja pipelining. SIMD (Single Instruction, Multiple Data) memungkinkan operasi yang sama pada beberapa data dieksekusi secara paralel dalam satu instruksi, cocok digunakan dalam pemrosesan data dalam skala besar. Multithreading memungkinkan prosesor menjalankan beberapa thread secara bersamaan, meningkatkan kinerja dan responsifitas sistem, terutama dalam aplikasi yang membutuhkan paralelisme tinggi. Caching digunakan untuk mengurangi waktu akses ke memori utama dengan menggunakan lapisan penyimpanan yang lebih cepat dan dekat dengan prosesor. Rancangan blueprint jaringan komputer perlu mempertimbangkan analisis kebutuhan yang spesifik untuk Hotel Amaris Palembang. Dalam hal ini, analisis kebutuhan meliputi jumlah pengguna, jenis aplikasi yang akan digunakan, kecepatan dan kapasitas yang diperlukan, serta persyaratan keamanan. Topologi jaringan yang tepat harus direncanakan untuk memenuhi kebutuhan hotel. Pemilihan

topologi jaringan seperti bus, star, atau mesh harus disesuaikan dengan kebutuhan hotel dalam hal konektivitas, skalabilitas, dan kehandalan.

Infrastruktur jaringan, termasuk perangkat keras jaringan seperti server, switch, router, dan access point, harus dipilih dan diatur dengan hati-hati untuk mendukung kebutuhan hotel. Ini melibatkan pemilihan perangkat dengan spesifikasi yang sesuai, memastikan ketersediaan dan keandalan yang memadai, serta menempatkan perangkat dengan benar dalam lingkungan hotel.

Konfigurasi jaringan yang tepat harus dilakukan menggunakan metode rekayasa sistem jaringan komputer. Pengaturan IP address, routing, VLAN, keamanan jaringan, dan layanan jaringan lainnya harus diterapkan dengan cermat untuk menjaga kestabilan dan keamanan jaringan.

Pengujian dan pemeliharaan rutin menjadi bagian penting dari rancangan blueprint jaringan komputer. Pengujian harus dilakukan untuk memastikan kinerja yang baik dan mengidentifikasi potensi masalah. Pemeliharaan rutin termasuk pemantauan jaringan, pembaruan perangkat lunak, dan penanganan masalah yang muncul untuk menjaga kinerja jaringan yang optimal.

UCAPAN TERIMA KASIH

Peneliti ini ingin mengucapkan terima kasih yang sebesar-besarnya kepada peneliti semua yang telah membaca artikel. Peneliti sangat berterima kasih atas waktu dan perhatian yang peneliti berikan. Menulis artikel memang merupakan hal yang menyenangkan bagi peneliti, tetapi tanpa pembaca seperti Peneliti artikel. Setiap kali peneliti menerima umpan balik dari pembaca, peneliti merasa sangat terhormat dan senang. Umpan balik peneliti membantu peneliti untuk terus meningkatkan kualitas tulisan peneliti dan memberikan inspirasi untuk topik-topik yang akan datang. Sekali lagi, terima kasih banyak atas dukungan dan perhatian peneliti. Peneliti berharap artikel-artikel peneliti terus memberikan manfaat dan inspirasi bagi peneliti. Jangan ragu untuk memberikan umpan balik atau saran untuk topik yang ingin peneliti baca di masa depan, Salam.

Daftar Pustaka

- [1] W. Saputra and B. E. Purnama, "Pengembangan Multimedia Pembelajaran Interaktif Untuk Mata Kuliah Organisasi Komputer," *Speed - Sentra Penelitian Engineering dan Edukasi*, vol. 4, no. 2, 2015.
- [2] S. Adrian and R. Cahyana, "Perancangan Arsitektur Jaringan Kampus Menggunakan Metode Network Development Life Cycle," *Jurnal Algoritma*, vol. 18, no. 2, 2022, doi: 10.33364/algoritma/v.18-2.845.
- [3] Ibnu Surya, A. Urip Ari Wibowo, J. Nurma Sari, M. Rois Adin Saf, and M. Ihsan Zul, "Sistem Bilangan Organisasi Dan Arsitektur Komputer Pada Siswa Man 2 Pekanbaru," *J-PEMAS - Jurnal Pengabdian Masyarakat*, vol. 3, no. 2, 2022, doi: 10.33372/j-pemas.v3i2.848.
- [4] H. Harjono, "Pelatihan Pembangunan Jaringan Komputer Dalam Perencanaan Arsitektur Jaringan Komputer Kabupaten Banyumas," *Jurnal Pengabdian Teknik dan Sains (JPTS)*, vol. 1, no. 1, 2021, doi: 10.30595/v1i1.9373.
- [5] A. Prasetya Nanda, N. Aminudin, Zulkifli, and M. Islamahdi, "PERANCANGAN ARSITEKTUR JARINGAN LOCAL AREA NETWORK PADA SMP

MUHAMMADIYAH 01 PRINGSEWU,” *Aisyah Journal Of Informatics and Electrical Engineering (A.J.I.E.E)*, vol. 2, no. 2, 2020, doi: 10.30604/jti.v2i2.36.

- [6] Sriani, *Diktat: Arsitektur Dan Organisasi Komputer*. 2020.
- [7] W. Mastiara and A. Afriyudi, “Rancangan Blueprint Jaringan Komputer Pada Hotel Amaris Palembang Menggunakan Metode Rekayasa Sistem Jaringan Komputer ...,” *Bina Darma Conference on ...*, 2020.
- [8] A. D. Sutiasih and R. P. Saputri, “Pengembangan mobile learning berbasis android sebagai media pembelajaran organisasi arsitektur komputer,” *Jurnal Inovasi Teknologi Pendidikan*, vol. 6, no. 2, 2019, doi: 10.21831/jitp.v6i2.27772.
- [9] E. Sinduningrum, *Teori Organisasi Arsitektur Komputer dan Praktik Assembleer untuk Pemula*, vol. 13, no. 1. 2020.
- [10] M. Azis, *Arsitektur Komputer dan Organisasi Komputer*. 2020.