

# **BUKU AJAR ARSITEKTUR KOMPUTER**

## **BAB 16**

### **ARSITEKTUR KOMPUTER MASA DEPAN: RISC-V, CHIPLET, DAN DOMAIN-SPECIFIC COMPUTING**

Buku Ajar Arsitektur Komputer 2026 | Berbasis RPS | 3 SKS

IDENTITAS BAB

| Komponen    | Keterangan                                                                                                                                |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Mata Kuliah | Arsitektur Komputer                                                                                                                       |
| Bab         | 16                                                                                                                                        |
| Judul       | Arsitektur Komputer Masa Depan: RISC-V, Chiplet, dan Domain-Specific Computing                                                            |
| Fokus       | Open ISA, workload-specific silicon, chiplet, advanced packaging, interconnect, memory, AI/HPC, keamanan, dan hardware–software co-design |
| Level       | Analisis, evaluasi trade-off, dan perancangan konseptual sistem masa depan                                                                |

CAPAIAN PEMBELAJARAN BAB

1. Menjelaskan perubahan arah desain arsitektur setelah perlambatan scaling tradisional.
2. Menjelaskan konsep domain-specific architecture dan workload-specific silicon.
3. Memahami prinsip RISC-V sebagai ISA terbuka, modular, dan extensible.
4. Menganalisis peran standard extension, custom extension, dan profile secara konseptual.
5. Menjelaskan konsep chiplet dan manfaat modularisasi silikon.
6. Membedakan monolithic SoC, 2.5D packaging, dan 3D integration.
7. Menganalisis kebutuhan die-to-die interconnect dan Network-on-Chip.
8. Menjelaskan hubungan AI/HPC dengan bandwidth memori dan interconnect.
9. Mengevaluasi keamanan, reliabilitas, dan software portability.
10. Merancang alternatif arsitektur masa depan untuk sebuah workload.

PETA KONSEP

Tantangan Scaling → DSA → Workload-Specific Silicon → Open ISA/RISC-V → Extensions & Profiles → Chiplet → Advanced Packaging → Die-to-Die Interconnect → NoC → High-Bandwidth Memory → AI/HPC → Co-Design → Security & Reliability → Future Architecture.

16.1 Pendahuluan

Bab ini merupakan sintesis lanjutan dari Bab 15. Jika Bab 15 membahas heterogeneous computing dan tren arsitektur modern, Bab 16 memusatkan perhatian pada bagaimana prosesor dan sistem masa depan dirancang secara lebih modular, terbuka, dan spesifik terhadap workload.

16.2 Tantangan Pasca-Scaling Tradisional

Peningkatan jumlah transistor dan performa tidak lagi dapat dipandang sebagai satu-satunya jalan peningkatan kemampuan komputasi. Konsumsi daya, panas, biaya desain, kompleksitas manufaktur, memory wall, dan komunikasi antarkomponen mendorong eksplorasi arsitektur yang lebih heterogen dan khusus.

## 16.3 Domain-Specific Architecture

Domain-Specific Architecture atau DSA menyesuaikan organisasi komputasi terhadap kelas workload tertentu. Keuntungannya dapat berupa efisiensi performa-per-watt, latensi, dan pemanfaatan area silikon. Kelemahannya adalah risiko berkurangnya fleksibilitas serta meningkatnya kebutuhan desain dan software khusus.

## 16.4 Workload-Specific Silicon

Workload-specific silicon membentuk kemampuan perangkat keras sesuai kebutuhan pekerjaan yang ditargetkan, bukan menyediakan semua kemungkinan fungsi dalam satu desain general-purpose. Konsep ini relevan ketika power, performance, dan area perlu dioptimalkan secara bersamaan.

## 16.5 RISC-V sebagai Open ISA

RISC-V adalah standar ISA terbuka berbasis prinsip RISC. Desainnya memisahkan base integer ISA yang kecil dari standard extensions sehingga implementasi dapat disesuaikan dengan kebutuhan. Keterbukaan ISA tidak berarti semua implementasi perangkat keras harus identik; implementasi dapat berbeda selama memenuhi spesifikasi yang relevan.

## 16.6 Modularitas: Base ISA, Extensions, dan Profiles

Base ISA menyediakan fondasi instruksi. Standard extensions menambah kemampuan yang didefinisikan secara bersama, sedangkan custom extensions dapat digunakan untuk kebutuhan khusus. Profiles mengelompokkan baseline fitur untuk membantu target perangkat lunak yang lebih seragam.

## 16.7 Standard dan Custom Extensions

Standard extensions memungkinkan ekosistem perangkat lunak menggunakan kemampuan yang terdokumentasi dan portable pada implementasi yang mendukungnya. Custom extensions dapat memberi optimasi khusus, tetapi menuntut perhatian terhadap toolchain dan kompatibilitas.

## 16.8 Open ISA dan Software Portability

ISA terbuka dapat mendorong eksperimen, pendidikan, dan pengembangan ekosistem. Namun portabilitas perangkat lunak tetap memerlukan baseline arsitektur, ABI, sistem operasi, compiler, library, dan target profile yang jelas.

## 16.9 Chiplet Architecture

Chiplet memecah sistem menjadi beberapa die fungsional yang dapat diintegrasikan dalam satu paket. Sebagai contoh konseptual, compute chiplet, I/O chiplet, accelerator chiplet, dan memory interface dapat dikembangkan sebagai komponen terpisah. Pendekatan ini mendukung modularitas dan heterogenitas.

## 16.10 Monolithic, 2.5D, dan 3D Integration

Monolithic SoC mengintegrasikan fungsi pada satu die. 2.5D dapat menghubungkan beberapa die melalui struktur packaging/interposer, sedangkan 3D integration menumpuk komponen secara vertikal. Setiap pendekatan memiliki trade-off pada bandwidth, densitas, termal, biaya, dan kompleksitas manufaktur.

## 16.11 Die-to-Die Interconnect

Pada sistem chiplet, komunikasi antardie harus mempertahankan bandwidth, latensi, reliabilitas, dan efisiensi energi. Interoperabilitas antarkomponen menjadi bagian penting dari keberhasilan ekosistem modular.

## 16.12 Network-on-Chip

Network-on-Chip atau NoC mengorganisasi komunikasi di dalam sistem multicore atau manycore. Router, topologi, routing, arbitration, dan congestion memengaruhi latency serta scalability. Konsep NoC dapat diperluas secara konseptual ketika sistem terdiri dari beberapa compute domain.

## 16.13 High-Bandwidth Memory dan Memory-Centric Design

AI dan HPC sering membutuhkan bandwidth data tinggi. Oleh karena itu, kedekatan memori dengan compute, hierarki cache, locality, memory controller, dan teknologi memori berbandwidth tinggi menjadi faktor desain. Arsitektur masa depan semakin mempertimbangkan biaya perpindahan data sebagai bagian utama dari optimasi.

## 16.14 Chiplet untuk AI dan HPC

AI dan HPC dapat menggunakan kombinasi compute chiplet, accelerator, memori berbandwidth tinggi, dan interconnect berkecepatan tinggi. Tantangan utamanya bukan hanya jumlah unit komputasi, tetapi bagaimana data, memori, sinkronisasi, dan komunikasi dapat mendukung utilisasi sistem secara efektif.

## 16.15 Hardware–Software Co-Design

Keunggulan arsitektur modern sering diperoleh dari perancangan bersama algoritma, compiler, runtime, format data, instruksi, memori, dan accelerator. Optimasi hanya pada perangkat keras tanpa dukungan software dapat menghasilkan utilisasi rendah; sebaliknya software dapat dibatasi oleh karakteristik fisik sistem.

## 16.16 Keamanan Arsitektur Masa Depan

Sistem heterogen dan modular memerlukan pertimbangan keamanan pada ISA, privilege, isolasi, memori, firmware, komunikasi, dan supply chain. Keamanan perlu dipandang sebagai bagian dari arsitektur sistem, bukan fitur tambahan setelah desain selesai.

## 16.17 Reliabilitas dan Fault Tolerance

Semakin banyak komponen meningkatkan kebutuhan deteksi kesalahan, isolasi fault, redundancy, monitoring, dan recovery. Sistem masa depan perlu menyeimbangkan performa dengan kemampuan mempertahankan layanan ketika terjadi gangguan pada sebagian komponen.

### 16.18 Open Hardware Ecosystem

Ekosistem terbuka dapat mempercepat eksperimen dan kolaborasi, tetapi tetap membutuhkan spesifikasi, pengujian, verifikasi, toolchain, dokumentasi, dan tata kelola agar interoperabilitas tidak berubah menjadi fragmentasi.

### 16.19 Studi Kasus 1: RISC-V untuk Sistem Heterogen

Rancang sebuah sistem dengan core general-purpose, unit vector atau accelerator, controller I/O, dan modul keamanan. Tentukan mana yang menggunakan standard capability dan mana yang membutuhkan optimasi khusus. Evaluasi dampaknya terhadap portabilitas software.

### 16.20 Studi Kasus 2: Chiplet AI Accelerator

Sebuah organisasi membutuhkan sistem untuk inferensi dan pelatihan. Mahasiswa membandingkan desain monolithic dengan desain modular yang memisahkan compute, I/O, dan memory. Analisis mencakup bandwidth, latency, thermal, biaya, dan scalability.

### 16.21 Studi Kasus 3: Edge-to-Cloud Future Architecture

Sistem sensor menghasilkan data kontinu. Edge melakukan filtering dan inferensi awal, sedangkan pusat komputasi menangani pelatihan dan analitik. Mahasiswa merancang pembagian compute serta memilih interconnect dan memori berdasarkan kebutuhan latensi dan volume data.

### 16.22 Perbandingan Pendekatan Desain

| Pendekatan        | Kekuatan                                         | Trade-off                                         |
|-------------------|--------------------------------------------------|---------------------------------------------------|
| General-Purpose   | Fleksibel dan luas penggunaannya                 | Tidak selalu optimal untuk workload tertentu      |
| Domain-Specific   | Efisien pada domain target                       | Fleksibilitas lebih terbatas                      |
| Workload-Specific | Dioptimalkan untuk kebutuhan yang lebih spesifik | Risiko ketergantungan pada workload dan toolchain |
| Monolithic SoC    | Integrasi kuat                                   | Ukuran dan kompleksitas die dapat meningkat       |
| Chiplet           | Modular dan heterogen                            | Membutuhkan packaging dan interconnect kompleks   |

### 16.23 Aktivitas Pembelajaran

- Buat peta hubungan antara RISC-V, extension, profile, accelerator, dan software portability.
- Bandingkan tiga arsitektur untuk satu aplikasi: CPU general-purpose, GPU/accelerator, dan DSA.
- Rancang diagram chiplet konseptual untuk AI edge server.
- Analisis bottleneck komunikasi pada sistem manycore.
- Buat matriks monolithic versus chiplet berdasarkan biaya, bandwidth, thermal, dan scalability.
- Diskusikan bagaimana open ISA dapat meningkatkan inovasi sekaligus menciptakan risiko fragmentasi.

## 16.24 Rangkuman

- Arsitektur masa depan bergerak menuju heterogenitas, modularitas, dan optimasi berbasis workload.
- DSA dan workload-specific silicon berupaya meningkatkan efisiensi dengan mencocokkan perangkat keras dengan karakteristik pekerjaan.
- RISC-V menyediakan model ISA terbuka, modular, dan extensible.
- Extensions dan profiles membantu menyeimbangkan spesialisasi dengan portabilitas.
- Chiplet memungkinkan integrasi beberapa fungsi atau teknologi dalam satu sistem paket.
- Advanced packaging membuka peluang bandwidth dan densitas yang berbeda dari integrasi monolithic.
- Die-to-die interconnect dan NoC menjadi elemen penting bagi sistem modular dan scalable.
- Bandwidth memori dan data movement tetap menjadi batas utama pada AI dan HPC.
- Hardware–software co-design diperlukan untuk memanfaatkan kemampuan arsitektur secara efektif.
- Keamanan, reliabilitas, dan interoperabilitas harus dirancang sejak awal.

## 16.25 Glosarium

| Istilah                      | Pengertian                                                                           |
|------------------------------|--------------------------------------------------------------------------------------|
| ASIC                         | Integrated circuit yang dirancang untuk fungsi atau aplikasi khusus.                 |
| Chiplet                      | Die modular yang menjadi bagian dari sistem paket yang lebih besar.                  |
| Custom Extension             | Ekstensi ISA yang dirancang untuk kebutuhan khusus implementasi.                     |
| Die-to-Die Interconnect      | Mekanisme komunikasi antara die dalam sistem paket.                                  |
| Domain-Specific Architecture | Arsitektur yang dioptimalkan untuk domain atau kelas workload tertentu.              |
| NoC                          | Infrastruktur komunikasi internal untuk banyak komponen komputasi.                   |
| Open ISA                     | Spesifikasi instruction set yang tersedia sebagai standar terbuka.                   |
| Profile                      | Baseline fitur arsitektur yang dikelompokkan dan didefinisikan untuk target bersama. |
| RISC-V                       | ISA terbuka dan modular berdasarkan prinsip RISC.                                    |
| Workload-Specific Silicon    | Silikon yang disesuaikan dengan kebutuhan workload target.                           |

## 16.26 Soal Latihan

### A. Pemahaman

11. Jelaskan mengapa desain general-purpose tidak selalu optimal.
12. Apa yang dimaksud DSA?
13. Jelaskan workload-specific silicon.
14. Apa karakteristik utama RISC-V sebagai ISA?

15. Bedakan base ISA, extension, dan profile.
16. Apa keuntungan dan tantangan chiplet?
17. Bandingkan 2.5D dan 3D integration secara konseptual.
18. Mengapa interconnect penting pada AI/HPC?

## B. Analisis

19. Analisis kapan custom extension dapat meningkatkan performa tetapi mengurangi portability.
20. Bandingkan desain monolithic dan chiplet untuk sistem AI.
21. Identifikasi bottleneck NoC pada sistem manycore.
22. Rancang pembagian compute-memory-interconnect untuk workload analitik besar.
23. Evaluasi risiko keamanan pada sistem modular dan terbuka.

## C. HOTS

24. Rancang prosesor masa depan untuk sistem edge AI dengan batas daya ketat.
25. Buat tiga alternatif desain chiplet dan pilih satu menggunakan matriks trade-off.
26. Usulkan strategi agar sistem RISC-V dapat menggabungkan standardization dan inovasi custom tanpa fragmentasi berlebihan.
27. Rancang arsitektur AI/HPC yang mempertimbangkan compute, memory, interconnect, energy, thermal, dan reliability.

## 16.27 Proyek Mini: Future Computer Architecture Designer

Mahasiswa merancang sebuah arsitektur komputer masa depan untuk studi kasus yang dipilih, misalnya edge AI, data center, robotika, kendaraan cerdas, atau scientific computing.

- Deskripsikan workload dan kebutuhan sistem.
- Pilih model ISA dan alasan pemilihannya.
- Tentukan komponen general-purpose dan domain-specific.
- Rancang organisasi chiplet atau monolithic.
- Tentukan kebutuhan memori dan interconnect.
- Analisis performa, energi, thermal, keamanan, reliabilitas, dan biaya.
- Buat minimal tiga alternatif desain dan matriks keputusan.
- Sajikan diagram arsitektur dan argumentasi desain.

## 16.28 Daftar Pustaka dan Sumber Pengayaan

- Patterson, D. A., & Hennessy, J. L. Computer Organization and Design: The Hardware/Software Interface.
- Hennessy, J. L., & Patterson, D. A. Computer Architecture: A Quantitative Approach.
- Stallings, W. Computer Organization and Architecture: Designing for Performance.
- RISC-V International. RISC-V Ratified Specifications Library, termasuk Unprivileged dan Privileged ISA.
- RISC-V International. Workload-Specific Silicon dan Open ISA resources.
- Dokumen RPS Mata Kuliah Arsitektur Komputer sebagai acuan organisasi pembelajaran buku.