

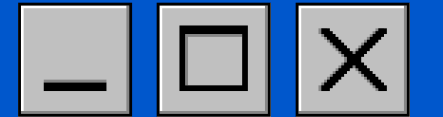
Tugas studi kasus

ORGANISASI DAN ARSITEKTUR KOMPUTER

2403010065_C - MUHAMMAD RAYA RAMADHAN

Start

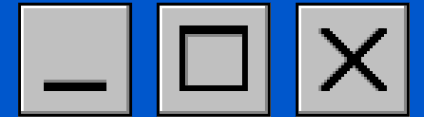




MODERNISASI INFRASTRUKTUR KOMPUTASI SOLUSI ARSITEKTUR UNTUK DATAGUARD SOLUTIONS

Analisis Efisiensi ISA, Optimasi
Pipelining, dan Hierarki Memori





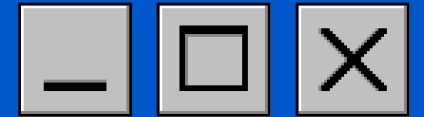
LATAR BELAKANG



DataGuard Solutions menghadapi high latency pada ribuan transaksi enkripsi/detik.

Beban kerja (workload) berat pada aritmatika repetitif dan akses memori yang tidak efisien.



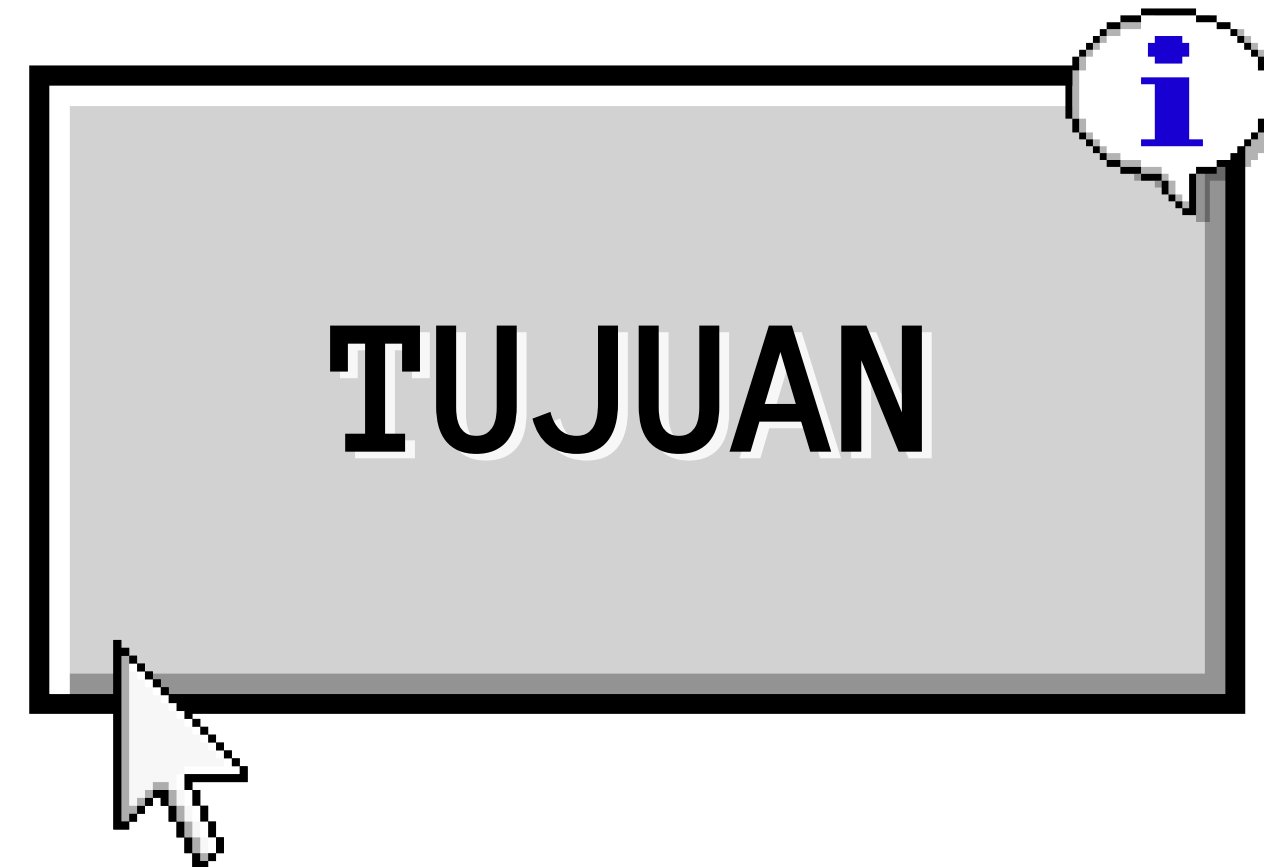


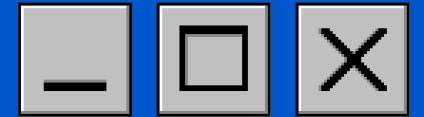
Home

Content

About

Menentukan arsitektur terbaik (RISC vs CISC) dan strategi memori untuk menjamin kecepatan serta efisiensi energi.





[Home](#)

[Content](#)

[About](#)

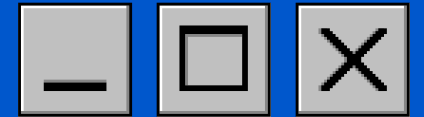


MENANGANI BEBAN KERJA ENKRIPSI

Analisis Komparatif

Karakteristik Enkripsi: Algoritma enkripsi (seperti AES) terdiri dari operasi logika/aritmatika sederhana yang berulang (repetitif) dalam jumlah masif.





[Home](#)

[Content](#)

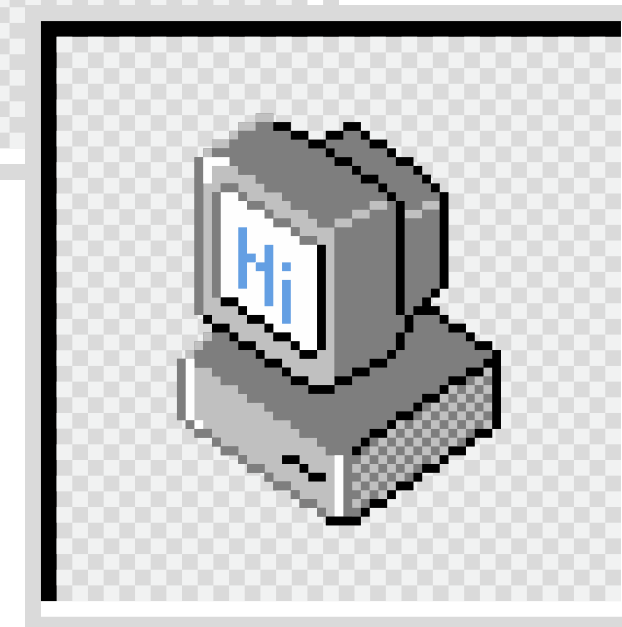
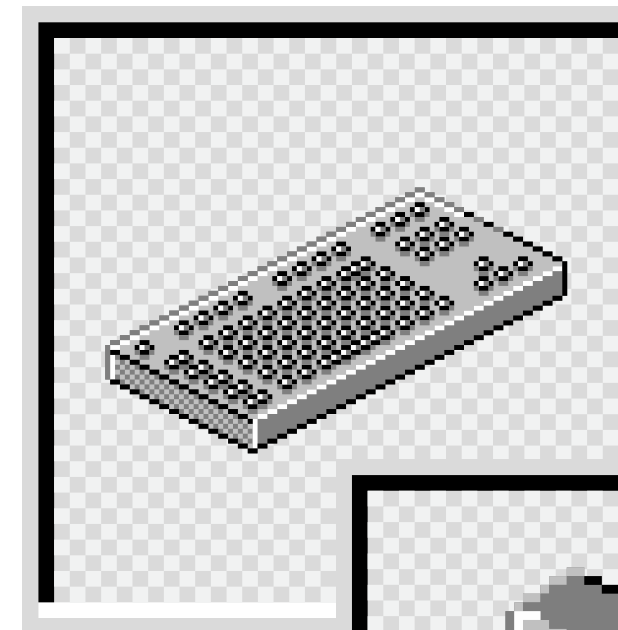
[About](#)

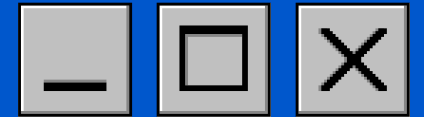


RISC (REDUCED INSTRUCTION SET COMPUTER)

Kelebihan: Menggunakan instruksi sederhana dengan panjang tetap (fixed-length). Sangat efisien untuk pipelining karena setiap tahap eksekusi terprediksi.

Konteks DataGuard: Cocok untuk tugas enkripsi berulang karena compiler dapat mengoptimalkan loop instruksi sederhana dengan lebih baik dibanding instruksi kompleks CISC.





[Home](#)

[Content](#)

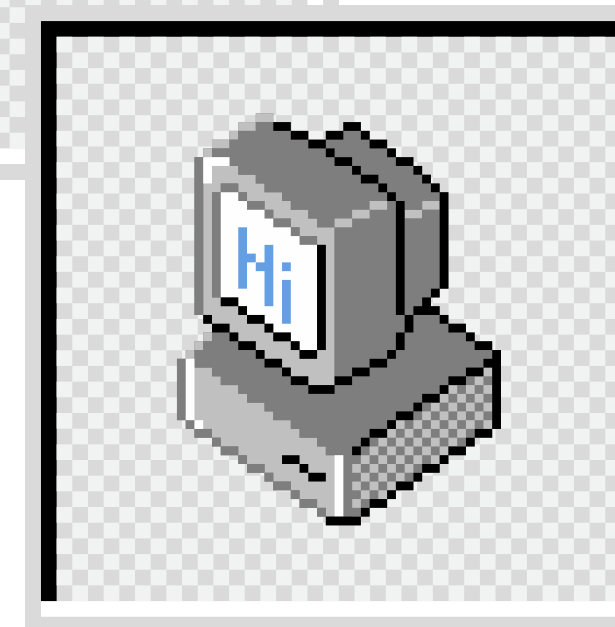
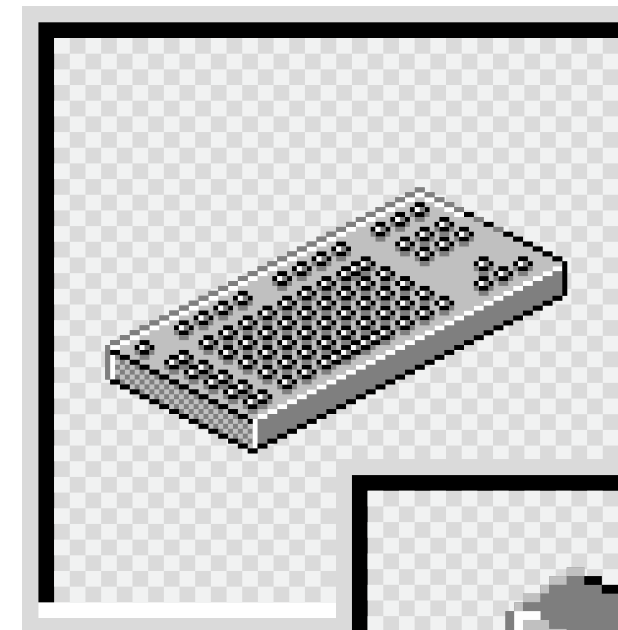
[About](#)

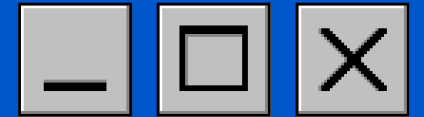


CISC (COMPLEX INSTRUCTION SET COMPUTER)

Kelebihan: Satu instruksi bisa melakukan banyak hal (load-compute-store). Kode lebih ringkas.

Kekurangan: Kompleksitas decoding instruksi memakan daya lebih besar dan menghasilkan panas berlebih, yang kurang ideal untuk server farm padat karya.





[Home](#)

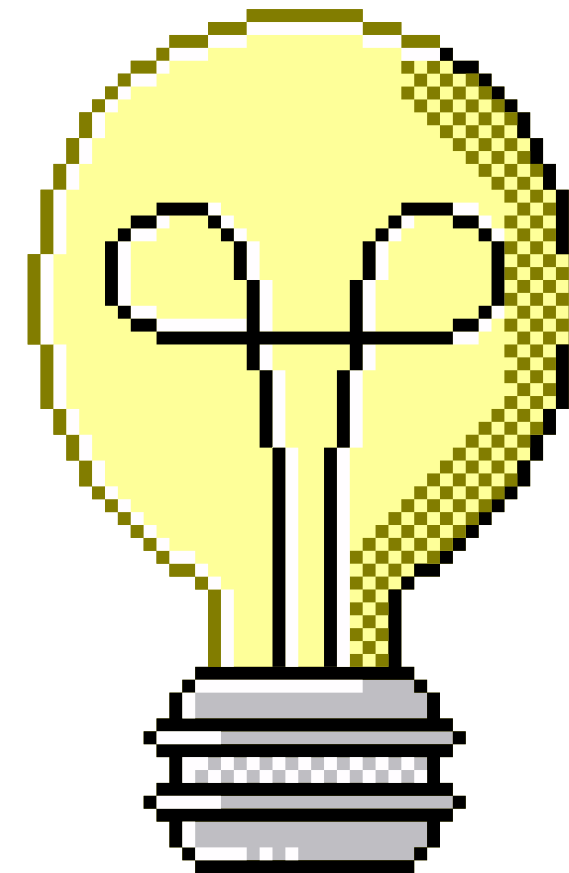
[Content](#)

[About](#)



KESIMPULAN

RISC lebih unggul untuk throughput data enkripsi yang tinggi dan stabil.






Home

Content

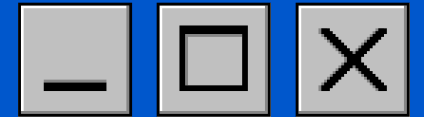
About


$$T = N \times CPI \times \frac{1}{f}$$

Rumus Waktu Eksekusi (CPU Time)

N = Jumlah Instruksi, CPI = Cycles Per Instruction, f = Frekuensi Clock





[Home](#)

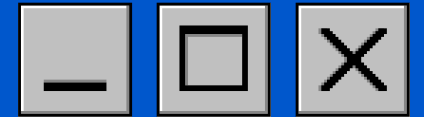
[Content](#)

[About](#)

Perbandingan Skenario DataGuard

1. CISC (x86): N Kecil (instruksi padat), namun CPI Tinggi (bisa >4 siklus/instruksi). Konsumsi daya tinggi karena kompleksitas transistor.
2. RISC (ARM): N Besar (perlu banyak instruksi sederhana), namun CPI Sangat Rendah (mendekati 1).
 - Analisis Energi: Riset terbaru (2024-2025) menunjukkan arsitektur RISC modern (seperti ARM Neoverse) menawarkan efisiensi performance-per-watt yang jauh lebih baik untuk beban kerja cloud/server dibanding x86 tradisional.





[Home](#)

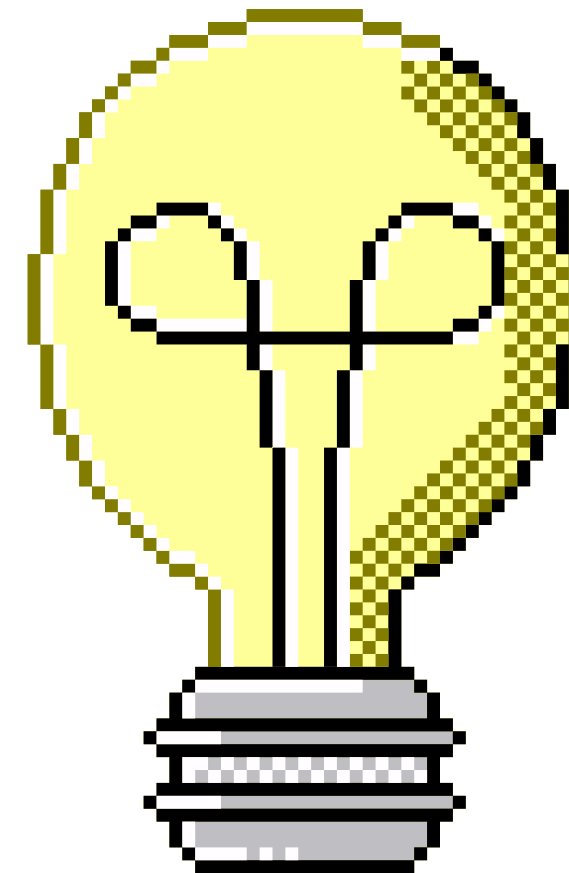
[Content](#)

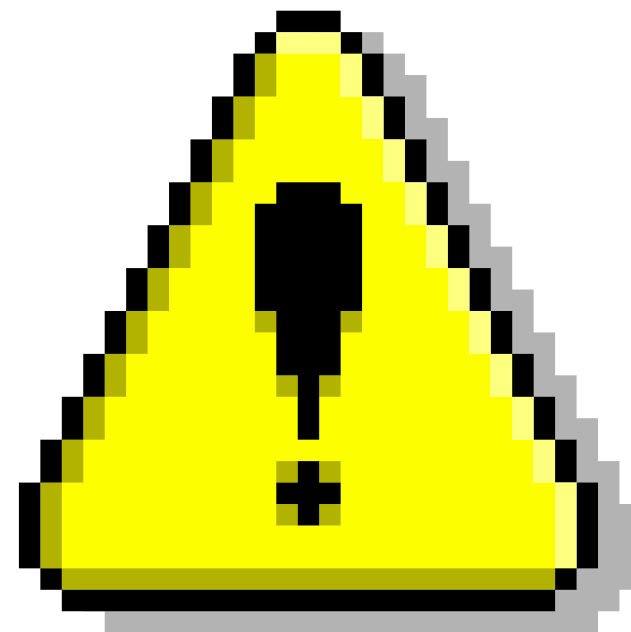
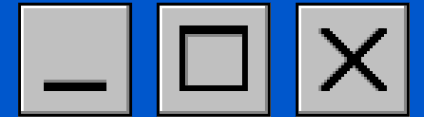
[About](#)



KESIMPULAN

Pemengangnya yaitu RISC. Efisiensi daya berarti biaya operasional (OpEx) lebih rendah dan manajemen panas yang lebih mudah bagi DataGuard.

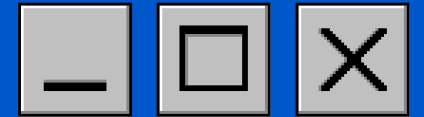




MENGAMANKAN ALIRAN DATA IDENTIFIKASI HAZARD

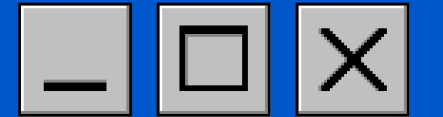
Pipelining memungkinkan eksekusi instruksi secara paralel (seperti ban berjalan). Namun, pada transaksi keuangan, hazard (hambatan) berikut sering terjadi:





1. Data Hazard (RAW - Read After Write):
 - Kasus: Instruksi enkripsi tahap 2 membutuhkan hasil dari tahap 1 yang belum selesai ditulis ke register.
 - Dampak: Stall (penundaan) pipa eksekusi.
2. Structural Hazard: Modul enkripsi dan modul I/O berebut akses ke memori data (cache) secara bersamaan.
3. Control Hazard (Branching): Sistem memprediksi transaksi "Valid" (lanjut proses), ternyata hasilnya "Fraud" (batal). Pipa harus dikosongkan (flush), membuang siklus CPU berharga saat memvalidasi ribuan transaksi/detik.





[Home](#)

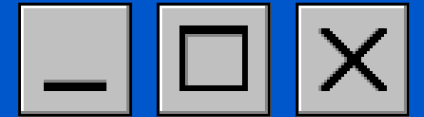
[Content](#)

[About](#)

Bagaimana Arsitek
Sistem mengatasi
latency?

**SOLUSI PARALELISME:
MEMBUKA SUMBATAN
KINERJA**





[Home](#)

[Content](#)

[About](#)

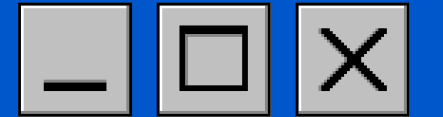
1. Teknik Superscalar (Jalan Tol Banyak Jalur):

- Konsep: CPU memiliki banyak unit eksekusi (ALU). Memungkinkan pemrosesan beberapa instruksi enkripsi secara paralel dalam satu siklus clock.
- Manfaat: Meningkatkan IPC (Instructions Per Cycle) secara drastis.

2. Multithreading (Seni Menunggu):

- Konsep: Saat satu thread transaksi macet menunggu data dari memori (Memory Stall), CPU langsung beralih mengerjakan thread nasabah lain.
- Relevansi: Sangat krusial untuk mengatasi "akses memori tidak efisien" yang dialami DataGuard. Teknik Simultaneous Multithreading (SMT) terbukti menggandakan throughput pada beban kerja server.





[Home](#)

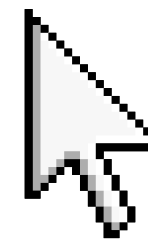
[Content](#)

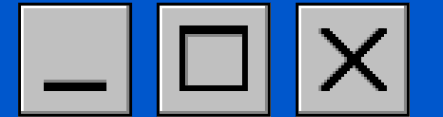
[About](#)



MENURUNKAN EFFECTIVE ACCESS TIME (EAT)

Akses ke RAM (Main Memory) lambat.
Implementasi Hierarki Cache (L1, L2, L3).





[Home](#)

[Content](#)

[About](#)

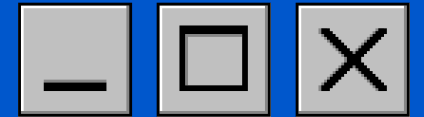


$$EAT = H \times T_{cache} + (1 - H) \times T_{main}$$

Konsep EAT:

H = Hit Rate, T = Waktu Akses





Home

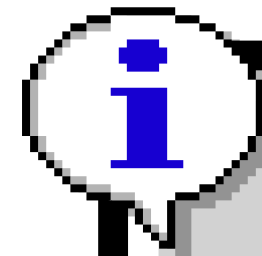
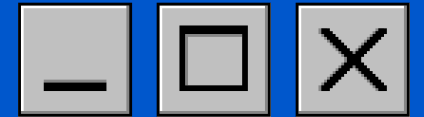
Content

About

Strategi Konfigurasi:

- L1 Cache (Split Instruction/Data): Khusus menyimpan instruksi enkripsi yang sering dipakai (High Hit Rate).
- L3 Cache (Shared): Ukuran besar (misal 32MB+) untuk menampung data transaksi sementara agar tidak perlu sering mengakses RAM utama yang lambat.
- Dampak: Dengan menjaga data "panas" (sering diakses) di Cache, EAT turun drastis, mengurangi latency total sistem.



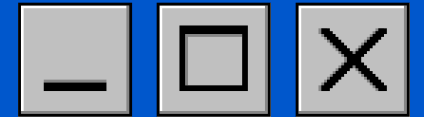


KEPUTUSAN ARSITEKTURAL

Sebagai Sistem Arsitek, rekomendasi untuk DataGuard Solutions adalah:

1. Arsitektur: Migrasi ke RISC (Server-Grade ARM).
 - Alasan: Unggul dalam efisiensi daya dan pemrosesan instruksi repetitif (enkripsi).
2. Optimasi: Gunakan prosesor dengan dukungan Superscalar & SMT (Simultaneous Multithreading).
 - Alasan: Menutupi latency memori dan memaksimalkan throughput transaksi.
3. Memori: Fokus pada Cache L2/L3 yang besar untuk meningkatkan Hit Rate dan menurunkan EAT.





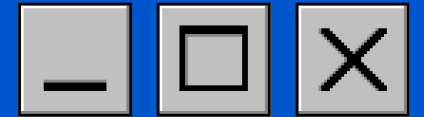
[Home](#)

[Content](#)

[About](#)

1. Maulana, A., et al. (2025). "RISC vs CISC: Studi Kinerja dan Efisiensi dalam Organisasi Arsitektur Komputer." *Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro Dan Informatika*, 3(4), 157-171.
2. Komunitas Dosen Indonesia. (2025). "Comparative Analysis of RISC and CISC Architectures in Modern Embedded System Development." *Jurnal KDI*.
3. Padoin, E., et al. (2024). "Towards green data centers: A comparison of x86 and ARM architectures power efficiency." *ResearchGate*.
4. Tullsen, D. M., et al. (2022). "Simultaneous Multithreading: A Platform for Next-generation Processors." *University of Washington Computer Science Strategy Reports*.
5. StudySmarter. (2023). "Pipeline Hazards: Causes, Examples & Impact Analysis in High Performance Computing."





TERIMA KASIH!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus efficitur finibus fermentum. Quisque luctus elit nulla, at auctor risus sodales ultrices. Ut imperdiet, magna eget eleifend eleifend, est.

rayardevskuy.com

@rayyrm_

2403010065

