

STUDI SISTEM INPUT/OUTPUT: PERANGKAT, INTERFACE, DAN OPTIMALISASI KINERJA KOMPUTER

Muhammad Rifki Hidayatulloh¹, Fierren Al-Hilal Saepul Bahri², Adila Muqtashida³,
Rio Gunawan⁴, Safira Azahra⁵

Universitas Islam Negeri Sultan Maulana Hasanuddin, Banten, Indonesia

Email¹²³⁴⁵: rifikiorangke2@gmail.com , fierrenalhilal0@gmail.com , dilamuqiew@gmail.com ,
061205rio@gmail.com , safiraazahra023@gmail.com

Informasi	Abstract
Volume : 2 Nomor : 6 Bulan : Juni Tahun : 2025 E-ISSN : 3062-9624	<i>The input/output (I/O) system is a crucial component in a computer architecture because it bridges the interaction between the external user/device and the processing unit. I/O devices consist of input devices (e.g. keyboards, sensors) to input data, and output devices (e.g. monitors, printers) to display processing results. The data from the input device is converted to a format that can be processed by the CPU, and then the results are passed to the output device via the I/O controller. This study discusses the definition and classification of I/O devices, types of I/O interfaces (Programmed I/O, Interrupt-Driven I/O, DMA), communication protocols, optimization strategies, and examples of implementations in embedded and cloud systems. An in-depth understanding of the relationship between input, processing, and output processes has proven to be key in improving the overall performance of a computer system.</i> Keywords : <i>input device, output device, I/O INTERFACE, DMA, performance optimization.</i> Abstrak <i>Sistem input/output (I/O) merupakan komponen krusial dalam arsitektur komputer karena menjembatani interaksi antara pengguna/perangkat luar dengan unit pemrosesan. Perangkat I/O terdiri atas perangkat input (misalnya keyboard, sensor) untuk memasukkan data, dan perangkat output (misalnya monitor, printer) untuk menampilkan hasil pemrosesan. Data dari perangkat input dikonversi ke format yang dapat diproses oleh CPU, lalu hasilnya diteruskan ke perangkat output melalui pengendali I/O. Studi ini membahas definisi dan klasifikasi perangkat I/O, tipe antarmuka I/O (Programmed I/O, Interrupt-Driven I/O, DMA), protokol komunikasi, strategi optimasi, serta contoh implementasi di sistem embedded dan cloud. Pemahaman mendalam mengenai hubungan antara proses input, pemrosesan, dan output terbukti menjadi kunci dalam meningkatkan kinerja sistem komputer secara keseluruhan.</i> Kata Kunci : <i>perangkat input, perangkat output, antarmuka I/O, DMA, optimasi kinerja.</i>

A. PENDAHULUAN

Komputer modern secara fundamental berfungsi menerima data masukan, memprosesnya, dan menghasilkan keluaran berupa informasi (teks, audio, citra, video). Perangkat I/O adalah perangkat keras yang memungkinkan proses tersebut berlangsung. Sebagai contoh, keyboard dan sensor mengirim data masukan ke CPU, sedangkan monitor dan printer menampilkan hasil proses. Menurut Kevin Dewa (2024), data dari perangkat input (misalnya keyboard atau sensor) diterima oleh bus data komputer, diproses oleh CPU, kemudian hasilnya disimpan atau dikirim ke perangkat output melalui pengendali I/O. Kecepatan dan efisiensi seluruh proses I/O sangat mempengaruhi kinerja sistem; peningkatan performa CPU saja tidak menjamin peningkatan kinerja I/O. Oleh karena itu, pengorganisasian subsistem I/O yang baik sangat penting untuk mencapai performa komputasi yang optimal. Dalam kajian ini, pertama-tama dipaparkan tinjauan pustaka mengenai definisi dan struktur I/O, dilanjutkan dengan metodologi, hasil analisis, studi kasus, dan kesimpulan.

B. TINJAUAN PUSTAKA

Definisi dan klasifikasi perangkat I/O. Perangkat input adalah komponen yang memasukkan data ke sistem komputer (misalnya keyboard, mouse, scanner, mikrofon, sensor), sedangkan perangkat output menyajikan data dari komputer ke dunia luar (misalnya monitor, printer, aktuator). Dalam arsitektur sistem komputer, perangkat-perangkat ini diatur oleh pengendali (controller) khusus dan terhubung melalui bus I/O. Sebagai ilustrasi, mikrokontroler (MCU) adalah contoh sistem embedded yang mengintegrasikan unit I/O internal. Seperti dinyatakan oleh Saefullah (2012) dan diulas oleh Muzakky dkk. (2024), MCU adalah “gadget digital yang mempunyai input dan output selain mempunyai kendali terhadap suatu program”. Dengan demikian, mikrokontroler adalah perangkat elektronik berbasis chip kecil yang dilengkapi CPU, memori, dan modul I/O, yang memungkinkan pembacaan sensor dan pengaktifan aktuator dalam sistem tertanam. Tipe antarmuka I/O. Transfer data antara CPU dan perangkat I/O dapat dilakukan dengan beberapa metode utama. Programmed I/O (PIO) melibatkan prosesor secara langsung dalam setiap transfer data; prosesor mengirim perintah baca/tulis ke perangkat I/O dan aktif menunggu hingga transfer selesai. Kekurangannya adalah banyaknya waktu prosesor terbuang karena harus terus memeriksa status perangkat.

Interrupt-Driven I/O mengatasi pemborosan waktu dengan menggunakan interupsi; perangkat I/O memberitahu CPU saat siap mengirim/menerima data. Model ini lebih efisien daripada PIO dalam pemanfaatan CPU karena CPU dapat melakukan tugas lain saat menunggu interupsi, tetapi masih menyisakan overhead karena setiap transfer data masih memerlukan rutin interupsi oleh CPU. Direct Memory Access (DMA) adalah mekanisme di mana controller DMA melakukan transfer blok data langsung antara memori utama dan perangkat I/O tanpa campur tangan CPU setelah pengaturan awal. Penggunaan DMA sangat meningkatkan kinerja pada transfer data berukuran besar; controller DMA dapat mengirimkan blok data penuh secara otomatis setelah dikonfigurasi, tanpa memicu interupsi berulang. Misalnya, setelah pengaturan awal, DMA dapat menyalin seluruh blok memori ke tujuan dalam satu operasi penuhnya.

Dengan demikian, DMA mengurangi beban CPU dan meningkatkan throughput I/O, terutama dalam aplikasi transfer data masif. Protokol komunikasi dan manajemen data. Sistem I/O juga melibatkan standar komunikasi antarperangkat. Pada perangkat embedded, protokol serial seperti I²C (Inter-Integrated Circuit) dan SPI (Serial Peripheral Interface) lazim digunakan untuk komunikasi jarak pendek antar modul. I²C menggunakan dua saluran (SDA dan SCL) untuk komunikasi master-slave bertempo sedang, sedangkan SPI menggunakan empat saluran (SCLK, MOSI, MISO, SS) untuk transmisi data berkecepatan tinggi antar mikrokontroler dan periferal seperti sensor dan memori. Di sisi lain, pada komputer personal dan server, bus dan protokol antar muka seperti USB, PCIe, dan SATA mengatur aliran data. Selain itu, sistem operasi melakukan manajemen I/O melalui buffering dan queuing request.

Teknik seperti antrian penjadwalan (misalnya algoritme elevator/SCAN untuk disk), caching I/O, dan pengaturan urutan penulisan dapat meminimalkan latensi dan meningkatkan pemanfaatan bandwidth bus. Dalam konteks manajemen data, strategi transfer burst via DMA dan penggunaan buffer FIFO memungkinkan penulisan/blok data diatur secara efisien sehingga CPU tidak selalu harus berperan sebagai perantara. Strategi optimalisasi kinerja I/O. Berbagai pendekatan dapat diterapkan untuk mempercepat sistem I/O. Secara perangkat keras, penggunaan controller DMA dan bus dengan lebar lebih besar (misalnya upgrade SATA ke PCIe) memperbesar bandwidth transfer. Selain itu, penggandaan jalur (multichannel I/O) dan penggunaan RAID pada penyimpanan dapat meningkatkan

throughput. Secara perangkat lunak, teknik parallel I/O dan pemrosesan asinkron membantu memaksimalkan pemanfaatan I/O.

Sebagai contoh, kajian Hou dkk. (2017) menunjukkan bahwa pada sistem penyimpanan cloud, kombinasi paralelisasi transfer dan pembagian data ke dalam chunk yang optimal dapat secara signifikan mengoptimalkan bandwidth I/O. Dengan mengirimkan permintaan data dalam beberapa blok paralel sekaligus, beban I/O dapat didistribusikan dan waktu tunggu berkurang. Analisis mereka menemukan bahwa selain paralelisasi, faktor lokalisasi (lokasi geografis dan kemampuan jaringan klien) sangat mempengaruhi kinerja end-to-end. Temuan ini mempertegas pentingnya strategi sisi klien untuk mengatasi batasan latensi jaringan dan memori. Secara umum, peningkatan waktu CPU saja tidak cukup meningkatkan keseluruhan kinerja sistem; perlu adanya peningkatan pada subsistem I/O itu sendiri.

C. METODE PENELITIAN

penelitian ini bersifat telaah pustaka dan analitis. Data diperoleh dari studi literatur atas publikasi ilmiah, buku teks arsitektur komputer, serta sumber terbuka yang kredibel. Alur kajian dimulai dengan identifikasi definisi dan konsep dasar perangkat I/O, dilanjutkan dengan klasifikasi metode transfer data (Programmed I/O, Interrupt, DMA) dan protokol interkoneksi. Selanjutnya, teknik optimasi kinerja I/O dianalisis dengan mengacu pada hasil eksperimen dan simulasi terdahulu. Pendekatan kualitatif digunakan untuk menelaah peran masing-masing komponen dalam sistem I/O dan mengevaluasi efektivitasnya berdasarkan ukuran kinerja seperti waktu respon dan throughput. Studi kasus dipilih untuk mengilustrasikan implementasi nyata konsep-konsep tersebut.

D. HASIL DAN PEMBAHASAN

Klasifikasi perangkat I/O. Beragam perangkat I/O memiliki karakteristik dan fungsi spesifik. Perangkat input meliputi alat-alat seperti keyboard, mouse, joystick, kamera, sensor suhu, serta modul komunikasi (misalnya modem) yang mengirimkan data ke CPU. Perangkat output mencakup monitor, proyektor, printer, aktuator (motor, lampu), dan speaker. Setiap perangkat I/O memerlukan interface yang sesuai; misalnya keyboard tradisional menggunakan protokol USB atau PS/2, sedangkan kamera mungkin terhubung melalui MIPI CSI atau USB. Fungsi utama perangkat I/O adalah mendukung input → proses → output, yang secara sistemik memastikan komputer dapat menerima perintah dan data eksternal, lalu menampilkan hasilnya. Keberagaman perangkat ini mengharuskan perancangan antarmuka

yang fleksibel dan protokol yang kompatibel sehingga berbagai I/O device dapat bekerja pada satu sistem.

Antarmuka I/O dan transfer data. Tabel berikut merangkum karakteristik tiap antarmuka I/O: Programmed I/O:

1. Prosesor menjalankan instruksi khusus untuk membaca/menulis ke register I/O perangkat. Sederhana tapi sangat membebani CPU, sehingga tidak efisien untuk transfer data besar.
2. Interrupt-Driven I/O: Perangkat I/O men-trigger interupsi saat siap transfer data. CPU bisa melakukan tugas lain sambil menunggu. Lebih efisien dibanding PIO, namun setiap byte/word masih melalui CPU.
3. Direct Memory Access (DMA): Controller DMA mentransfer blok data langsung antara memori dan perangkat I/O setelah dikonfigurasi. CPU tidak terlibat selama transfer, sehingga meningkatkan throughput. DMA ideal untuk transfer data berukuran besar atau terus-menerus (misalnya streaming audio/video).

Penerapan DMA memungkinkan mekanisme burst transfer: setelah setup awal, controller DMA mentransfer seluruh blok memori secara langsung ke tujuan tanpa intervensi CPU. Hal ini mengoptimalkan penggunaan jalur data dan memori, mengurangi latency dan jitter dibanding interupsi berulang. Dalam praktek, banyak sistem modern menggabungkan interrupt-driven dan DMA; CPU memerintahkan transfer DMA, lalu tugas I/O diselesaikan sepenuhnya oleh hardware.

Protokol dan pengelolaan data. Untuk komunikasi antar modul, beberapa protokol umum digunakan. Misalnya, pada sistem embedded sering terdapat protokol I²C dan SPI. I²C mendukung komunikasi multiple-master/multiple-slave dengan dua saluran, cocok untuk jarak pendek di papan sirkuit. SPI menggunakan empat saluran untuk koneksi berkecepatan tinggi dengan satu master dan beberapa slave. Pengaturan protokol ini harus hati-hati memilih frekuensi dan urutan bit agar latensi minimal. Di 12 3 3 • 6 • • 7 8 8 9 10 3 sistem dengan banyak data, teknik buffering (penyimpanan sementara) dan pengatur antrian (queue) I/O sangat penting. Misalnya, driver perangkat dapat memasukkan permintaan I/O ke antrian sehingga CPU tidak perlu menghalangi eksekusi program saat menunggu operasi I/O selesai. Teknik read-ahead (membaca data sebelum diminta) atau write-behind (menunda penulisan) juga membantu meningkatkan efisiensi penyimpanan.

Optimalisasi kinerja sistem I/O. Analisis menunjukkan bahwa peningkatan kinerja sistem komputer menuntut perbaikan subsistem I/O, bukan sekadar peningkatan frekuensi

CPU . Strategi optimasi meliputi peningkatan jalur akses (misalnya bus paralel), penerapan DMA, dan peningkatan algoritma scheduling. Studi Hou et al. (2017) menyoroti pentingnya pemecahan data dan paralelisasi pada klien cloud untuk memaksimalkan bandwidth penyimpanan . Dengan memecah permintaan baca/tulis menjadi banyak permintaan lebih kecil yang dieksekusi bersamaan, penyimpanan awan dapat melayani lebih banyak I/O dalam waktu singkat, mengurangi waktu tunggu kumulatif. Pendekatan semacam ini menunjukkan bahwa pemrosesan I/O secara paralel (baik di sisi perangkat maupun software) secara signifikan meningkatkan throughput. Selain itu, penggunaan storage SSD dan RAID juga populer untuk meningkatkan bandwidth I/O disk dibanding HDD konvensional. Secara keseluruhan, kombinasi teknik perangkat keras (DMA, bus cepat, penyimpanan paralel) dan perangkat lunak (buffering adaptif, penjadwalan dinamis) terbukti penting dalam optimalisasi kinerja I/O .

E. KESIMPULAN

Sistem I/O adalah komponen kunci yang mempengaruhi kinerja keseluruhan komputer. Perangkat input dan output beragam jenisnya (dari keyboard hingga sensor dan aktuator) memungkinkan komputer berinteraksi dengan dunia luar. Setiap jenis perangkat memerlukan antarmuka I/O yang tepat; metode transfer data seperti Programmed I/O, Interrupt-Driven I/O, dan terutama DMA secara signifikan mempengaruhi efisiensi transfer. Pemilihan protokol komunikasi (misalnya I²C/SPI pada embedded) serta pengelolaan data (buffering, scheduling) turut berperan penting. Analisis literatur menunjukkan bahwa optimasi kinerja I/O dilakukan dengan kombinasi peningkatan hardware (bus lebar, controller DMA, RAID/ SSD) dan software (paralelisasi, pengaturan antrian). Studi kasus memperlihatkan bahwa dalam sistem embedded, penggunaan DMA pada MCU mempercepat penanganan sensor tanpa membebani CPU, sementara pada arsitektur cloud, pemecahan tugas I/O dan pemrosesan paralel meningkatkan throughput secara signifikan . Oleh karena itu, untuk meningkatkan kinerja sistem komputer, diperlukan pemahaman menyeluruh akan mekanisme I/O dan penerapan strategi optimalisasi yang tepat.

F. DAFTAR PUSTAKA

- Dewa, O. D. K. F. (2024). Tech Function of Input and Output Technology in Computer Hardware and Software Development. *Journal Basic Science and Technology*, 13(2), 81–85 .

- Fadilah, S. A., Rizky, M., Sukira, S., & Aribowo, D. (2023). Mengevaluasi Efisiensi Pengontrol Input-Output dalam Arsitektur Komputer Modern. *Jurnal Teknik Mesin, Industri, Elektro dan Informatika*, 2(4), 96– 113 .
- Goswami, D. (2014). Onchip Memory Data Transfer Methods: An Overview. *IJSRD – International Journal for Scientific Research & Development*, 2(10), 498–502 .
- Hou, B., Chen, F., Ou, Z., Wang, R., & Mesnier, M. (2017). Understanding I/O Performance Behaviors of Cloud Storage from a Client’s Perspective. *ACM Transactions on Storage*, 13(2), Article 16 .
- Muzakkty, F., Muttaqin, M. R., Mubbin, A. F., Kartika, A., & Azizah, Z. (2024). Sistem berbasis RISC & CISC pada organisasi dan arsitektur komputer. *Kohesi: Jurnal Multidisiplin Saintek*, 3(7), 100–111 .
- Pradhana, C., & Sulaiman, M. (2024). Komunikasi Arduino I2C, SPI dan UART. *SWITCH: Jurnal Sains dan Teknologi*, 2(4), 80–83 .