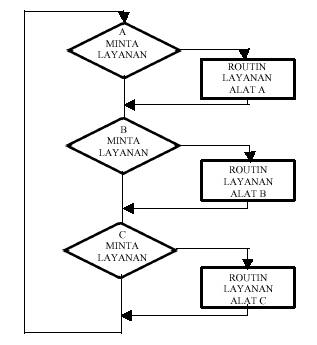
**METODE PENGENDALIAN INPUT OUTPUT ( I / O )**

1. **Metode Polling**

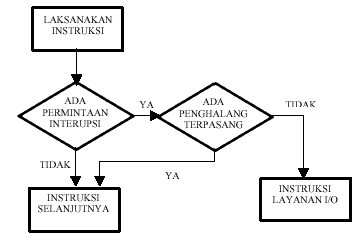
          Metode polling merupakan metode pengendalian I/O melalui program. Semua pengalihan data dari dan ke alat I/O diselengarakan oleh program. Prosesor mengirim dan meminta data sepenuhnya dibawah kendali program. Pengalihan data dapat dilaksanakan baik melalui mekanisme jabat tangan maupun tanpa jabat tangan. Dalam mekanisme jabat tangan isyarat diperiksa secara terus menerus. Program terus menerus berputar lewat sejumlah pengetesan untuk menentukan apakah masukan atau keluaran dapat diselenggarakan pelayanannya atau tidak. Bila ditemukan alat yang memerlukan pelayanan, rutin pelayanan diaktifkan dan pemilihan saluran diproses. Gambar 19. menunjukkan diagram alir pengendalian I/O dengan metode polling. Metode polling adalah metode pengendalian I/O yang paling sederhana dan paling umum digunakan. Metode ini tidak memerlukan perangkat keras khusus dan semua pengalihan I/O dikendalikan oleh program. Pengalihan semacam ini disebut pengalihan serempak dengan program.

[](http://2.bp.blogspot.com/-E1C1weg2EmM/UXY3NcnZqPI/AAAAAAAABPM/W12ApnKwVLo/s1600/gambar9.1.jpg)

 Gambar 1. Diagram Alir Pengendalian I/O Metode Polling

**b.** **Metode**

**Interupsi**  
  
          Pengendalian I/O dengan metode polling mempunyai dua kelemahan :  
- Pemborosan waktu prosesor karena status semua periferal diperiksa terus menerus secara berurutan.  
- Karena harus memeriksa status semua alat I/O maka waktu kerjanya menjadi lambat. Ini merupakan kelemahan dalam sistem waktu nyata (Real Time), dimana satu periferal mengharap layanan dalam satu waktu tertentu.  
          Kelemahan ini diatasi dengan menggunakan layanan waktu tak sinkron menggunakan interupsi. Tiap alat I/O atau pengendalinya dihubungkan ke sebuah saluran interupsi. Saluran interupsi menggerbangkan sebuah permintaan interupsi ke mikroprosesor. Bilamana sebuah alat I/O memerlukan layanan , alat akan membangkitkan pulsa interupsi atau status suatu tingkatan saluran untuk menarik perhatian mikroprosesor. Mikroprosesor akan memberikan layanan pada alat I/O jika ada interupsi dan jika tidak ada interupsi mikroprosesor melakukan instruksi selanjutnya. Logika pengendalian I/O dengan metode interupsi ditunjukkan pada diagram alir Gambar 2.

[](http://3.bp.blogspot.com/-bXkYpGd1x3Q/UXY4b0Xh4tI/AAAAAAAABPY/dpqqY2Zsqcw/s1600/gambar9.2.jpg)

Gambar 2. Diagram Alir Pengendalian I/O Metode Interupsi

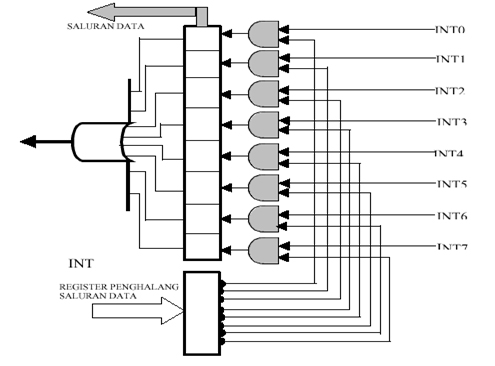
          Begitu permintaan interupsi diterima dan disetujui oleh mikroprosesor, alat I/O harus dilayani. Untuk melayani alat I/O, maka mikroprosesor melaksanakan suatu routin pelayanan khusus. Ada dua masalah yang muncul pada saat melakukan layanan interupsi :

- Bagaimana status program yang dilaksanakan pada mikroprosesor pada saat interupsi harus diperilahara dalam *stack.*

*-*Bagaimana mikroprosesor dapat mengenali secara tepat alat I/O mana yang membangkitkan interupsi. Identifikasi ini dapat dilakukan dengan perangkat keras, perangkat lunak, atau kombinasi perangkat keras dan perangkat lunak. Pencabangan ke alamat alat I/O disebut **Pemvektoran Interupsi**. Secara rutin perangkat lunak menetapkan identitas alat yang meminta layanan interupsi. Rutin identifikasi interupsi akan memilih saluran setiap alat yang dihubungkan dengan sistem. Setelah dikenal alat mana yang mencetuskan interupsi maka ia kemudian

bercabang ke alamat rutin penanganan interupsi yang sesuai. Metode ke dua yang digerakkan oleh perangkat lunak, tetapi dengan pertolongan beberapa perangkat keras tambahan. Metode ini menggunakan rantai beranting (daisy chain) untuk mengenal alat yang mencetuskan interupsi. Metode tercepat adalah interupsi yang divektorkan. Adalah menjadi tanggung jawab pengendali alat I/O untuk memberikan baik interupsi maupun pengenal alat yang menyebabkan interupsi atau lebih baik lagi alamat pencabangan bagi rutin penanganan interupsi. Bila pengendali hanya memberikan pengenal alat, adalah tugas perangkat lunak mencari tabel alamat pencabangan bagi tiap alat. Ini sederhana bagi perangkat keras tapi tak mencapai performansi tertinggi.

- Prioritas, beberapa interupsi dapat dibangkitkan serentak. Mikroprosesor diberi tugas untuk memutuskan bagaimana urutan pelayanannya. Setiap alat diberikan suatu prioritas. Mikroprosesor melayani setiap alat sesuai prioritasnya. Dalam dunia komputer prioritas 0, menurut konvensi memiliki prioritas. tertinggi, prioritas 1 yang kedua demikian seterusnya. Prioritas dapat diset baik pada perangkat keras maupun perangkat lunak. Pengaturan prioritas dengan perangkat keras dikerjakan oleh suatu piranti yang disebut Programmable Interrupt Controller (PIC). Struktur dasar logika PIC dapat digambarkan seperti Gambar 3.

[](http://2.bp.blogspot.com/-D9BWAYM8-Hw/UXY5jS6SOqI/AAAAAAAABPo/RUmfs1lLtuk/s1600/gambar9.3.jpg)

Gambar 3. Struktur Dasar Logika PIC

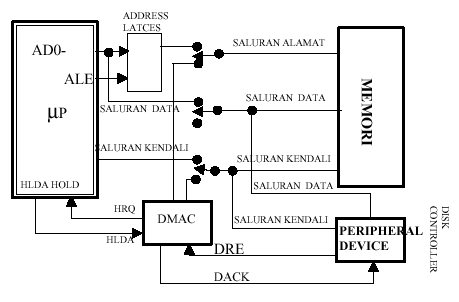
**c.**  **Akses Memori Langsung**

          Interupsi menjamin tanggapan yang paling cepat dari proses pengendalian data pada I/O. Akan tetapi pelayanan pada alat masih diselenggarakan oleh perangkat lunak. Kecepatan transfer paralel sebuah mikroprosesor dibatasi oleh overhead perangkat lunak yang terlibat dalam pengiriman kata-kata berurutan. Ini mungkin masih tidak cukup cepat bagi pengolahan yang melibatkan alih memori cepat. Kembali disini menggantikan perangkat lunak dengan perangkat keras. Rutin perangkat lunak yang menyelenggarakan alih data antara memori dengan alat I/O digantikan oleh prosesor perangkat keras khusus yang disebut dengan Direct Memory Access Controller (DMAC). Sebuah DMAC adalah prosesor khusus yang dirancang untuk menyelenggarakan alih data berkecepatan tinggi antara memori dengan alat luar. Dalam akses memori langsung digunakan dua teknik untuk berhubungan dengan memori :

- Prosesor dihentikan atau ditangguhkan oleh DMAC. DMAC memegang pengendalian bus dan membiarkan alat I/O berhubungan langsung dengan memori.

- DMAC mencuri satu siklus memori dari mikroprosesor, memberinya kepada pengiriman data antara memori dan alat I/O.

          DMAC adalah prosesor khusus yang memutuskan hubungan atau mengisolasi MPU dari bus-bus dan mengatur pengiriman yang diperlukan antara memori dan alat I/O. Gambar 4 menunjukkan diagram blok kerja DMAC.

[](http://2.bp.blogspot.com/-yGj2MXa73Bg/UXY6q_rlTQI/AAAAAAAABP4/iSiycqqfezI/s1600/gambar9.4.jpg)

Gambar 4. Diagram Blog Kerja DMAC

          Pada saat sistem bekerja, saklar pada posisi atas sehingga saluran terhubung dari mikroprosesor ke sistem memori dan peripheral. Untuk membaca file ke disk diperlukan sejumlah perintah ke disk controller, memerintahkan untuk mencari dan membaca blok data yang dari disk. Jika disk controller telah menemukan byte pertama dari blok data, disk controller mengirim sinyal DMA request (DREQ) ke DMAC. Jika DMAC tidak dalam terhalang maka DMAC mengirim sinyal hold request (HRQ) ke mikroprosesor melalui pin HOLD. Mikroprosesor menanggapi masukan ini dengan mengambangkan saluran/bus dan mengirim

sinyal hold acknowledge (HLDA) ke DMAC. Jika DMAC menerima sinyal HLDA, akan mengirim sinyal untuk menghubungkan bus/saluran ke posisi DMAC.

          Pada saat DMAC mengontrol saluran, ia mengirim alamat memori dimana byte pertama dari disk controller di tulis. Selanjutnya DMAC mengirim sinyal DMA acknowledge (DACK) ke disk controller untuk memberitahukan kesiapan mengeluarkan byte. Akhirnya DMAC mengaktifkan saluran MEMW\* dan IOR\* pada saluran kontrol.