

Pertemuan ke – 14

Sistem Bus

Riyanto Sigit, ST.
Nur Rosyid, S.kom
Setiawardhana, ST
Hero Yudo M, ST

Politeknik Elektronika Negeri Surabaya

Tujuan

- ⌘ Menjelaskan struktur antar hubungan
- ⌘ Menjelaskan *bus* antar hubungan
- ⌘ Menjelaskan elemen dari desain *bus*
- ⌘ Menjelaskan PCI, SCSI, Fire wire dan USB

Sistem Bus

- ⌘ Penghubung bagi keseluruhan komponen komputer dalam menjalankan tugasnya
- ⌘ Komponen komputer :
 - ☑ CPU
 - ☑ Memori
 - ☑ Perangkat I/O
- ⌘ Transfer data antar komponen komputer.
 - ☑ Data atau program yang tersimpan dalam memori dapat diakses dan dieksekusi CPU melalui perantara *bus*
 - ☑ Melihat hasil eksekusi melalui monitor juga menggunakan sistem *bus*
 - ☑ Kecepatan komponen penyusun komputer harus diimbangi kecepatan dan manajemen *bus* yang baik

Sistem Bus

⌘ Mikroprosesor

- ☑ Melakukan pekerjaan secara paralel
- ☑ Program dijalankan secara multitasking
- ☑ Sistem *bus* tidak hanya lebar tapi juga cepat

⌘ Interkoneksi komponen sistem komputer dalam menjalankan fungsinya

- ☑ Interkoneksi *bus*
- ☑ Pertimbangan–pertimbangan perancangan *bus*

Struktur Interkoneksi

- ⌘ Kumpulan lintasan atau saluran berbagai modul (CPU,Memori,I/O)
- ⌘ Struktur interkoneksi bergantung pada
 - ☒ Jenis data
 - ☒ Karakteristik pertukaran data

Jenis Data

Memori :

Memori umumnya terdiri atas N word memori dengan panjang yang sama. Masing-masing word diberi alamat numerik yang unik (0, 1, 2, ...N-1). *Word* dapat dibaca maupun ditulis pada memori dengan kontrol *Read* dan *Write*. Lokasi bagi operasi dispesifikasikan oleh sebuah alamat.

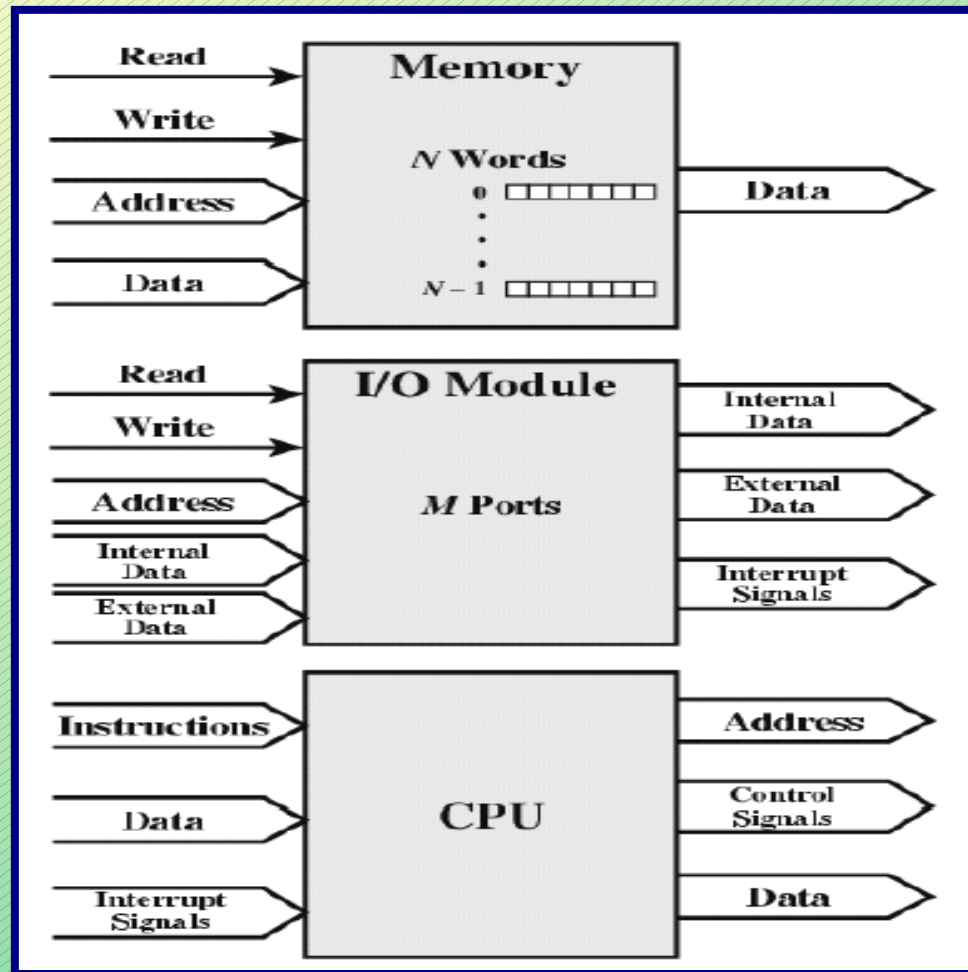
Modul I/O :

Operasi modul I/O adalah pertukaran data dari dan ke dalam komputer. Berdasarkan pandangan internal, modul I/O dipandang sebagai sebuah memori dengan operasi pembacaan dan penulisan. Seperti telah dijelaskan pada bab 6 bahwa modul I/O dapat mengontrol lebih dari sebuah perangkat peripheral. Modul I/O juga dapat mengirimkan sinyal interrupt.

CPU :

CPU berfungsi sebagai pusat pengolahan dan eksekusi data berdasarkan routine-routine program yang diberikan padanya. CPU mengendalikan seluruh sistem komputer sehingga sebagai konsekuensinya memiliki koneksi ke seluruh modul yang menjadi bagian sistem komputer.

Modul – modul komputer



Struktur interkoneksi

- ⌘ Dari jenis pertukaran data yang diperlukan modul–modul komputer, maka struktur interkoneksi harus mendukung perpindahan data.
 - ☒ *Memori ke CPU* : CPU melakukan pembacaan data maupun instruksi dari memori.
 - ☒ *CPU ke Memori* : CPU melakukan penyimpanan atau penulisan data ke memori.
 - ☒ *I/O ke CPU* : CPU membaca data dari peripheral melalui modul I/O.
 - ☒ *CPU ke I/O* : CPU mengirimkan data ke perangkat peripheral melalui modul I/O.
 - ☒ *I/O ke Memori atau dari Memori* : digunakan pada sistem DMA

Perkembangan Struktur Intekoneksi

⌘ Sampai saat ini terjadi perkembangan struktur interkoneksi, namun yang banyak digunakan saat ini adalah sistem *bus*.

⌘ Sistem *bus*

☑ Digunakan secara tunggal

☑ Digunakan secara jamak,

Hal ini Tergantung karakteristik sistemnya

Interkoneksi *Bus*

⌘ *Bus* ?

- ☑ merupakan lintasan komunikasi yang menghubungkan dua atau lebih komponen komputer

⌘ Sifat penting dan merupakan syarat utama ?

- ☑ *bus* adalah media transmisi yang dapat digunakan bersama oleh sejumlah perangkat yang terhubung padanya

⌘ Digunakan bersama ?

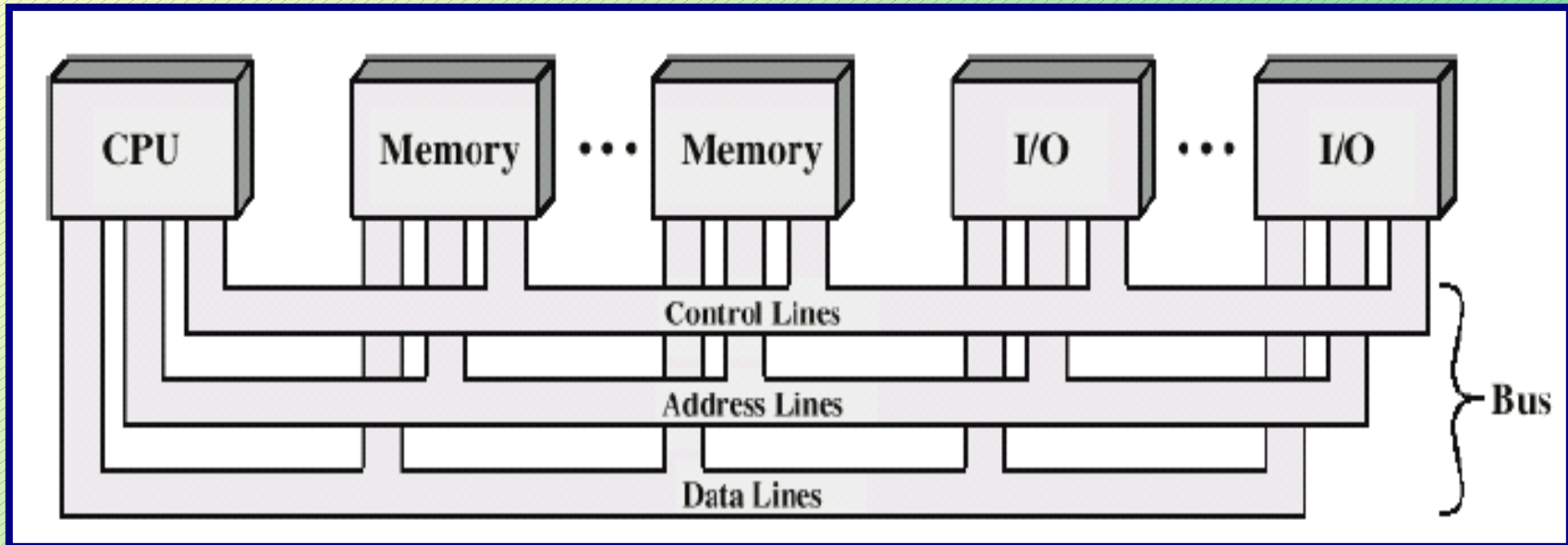
- ☑ Diperlukan aturan main agar tidak terjadi tabrakan data atau kerusakan data yang ditransmisikan.

- ☑ Walaupun digunakan bersama namun dalam satu waktu hanya ada sebuah perangkat yang dapat menggunakan *bus*

Interkoneksi *Bus* - Struktur *Bus*

- ⌘ Sebuah bus biasanya terdiri atas beberapa saluran.
 - ☑ Sebagai contoh *bus* data terdiri atas 8 saluran sehingga dalam satu waktu dapat mentransfer data 8 bit.
- ⌘ Secara umum fungsi saluran *bus* dikategorikan dalam tiga bagian
 - ☑ Saluran data
 - ☑ Saluran alamat
 - ☑ Saluran kontrol

Pola interkoneksi *bus*



Saluran data (*data bus*)

- ⌘ Lintasan bagi perpindahan data antar modul.
- ⌘ Secara kolektif lintasan ini disebut *bus data*.
Umumnya jumlah saluran terkait dengan panjang word, misalnya 8, 16, 32 saluran
- ⌘ Tujuan : agar mentransfer word dalam sekali waktu.
- ⌘ Jumlah saluran dalam *bus* data dikatakan *lebar bus*, dengan satuan bit, misal lebar *bus* 16 bit

Saluran alamat (*address bus*)

- ⌘ Digunakan untuk menspesifikasi sumber dan tujuan data pada *bus* data.
- ⌘ Digunakan untuk mengirim alamat word pada memori yang akan diakses CPU.
- ⌘ Digunakan untuk saluran alamat perangkat modul komputer saat CPU mengakses suatu modul.
- ⌘ Semua peralatan yang terhubung dengan sistem komputer, agar dapat diakses harus memiliki alamat.
 - ☒ Contoh : mengakses port I/O, maka port I/O harus memiliki alamat *hardware*-nya

Saluran kontrol (*control bus*)

- ⌘ Digunakan untuk mengontrol *bus* data, *bus* alamat dan seluruh modul yang ada.
- ⌘ Karena *bus* data dan *bus* alamat digunakan oleh semua komponen maka diperlukan suatu mekanisme kerja yang dikontrol melalui *bus* kontrol ini.
- ⌘ Sinyal – sinyal kontrol terdiri atas
 - ☑ Sinyal pewaktuan
 - ☑ Sinyal–sinyal perintah

Sinyal Saluran kontrol

- ⌘ Sinyal pewaktuan menandakan validitas data dan alamat
- ⌘ Sinyal perintah berfungsi membentuk suatu operasi

Saluran kontrol

Apa saja ?

- ⌘ *Memory Write*, memerintahkan data pada *bus* akan dituliskan ke dalam lokasi alamat.
- ⌘ *Memory Read* memerintahkan data dari lokasi alamat ditempatkan pada *bus* data.
- ⌘ *I/O Write*, memerintahkan data pada *bus* dikirim ke lokasi port I/O.
- ⌘ *I/O Read*, memerintahkan data dari port I/O ditempatkan pada *bus* data.
- ⌘ *Transfer ACK*, menunjukkan data telah diterima dari *bus* atau data telah ditempatkan pada *bus*.
- ⌘ *Bus Request*, menunjukkan bahwa modul memerlukan kontrol bus.
- ⌘ *Bus Grant*, menunjukkan modul yang melakukan request telah diberi hak mengontrol *bus*.
- ⌘ *Interrupt Request*, menandakan adanya penangguhan interupsi dari modul.
- ⌘ *Interrupt ACK*, menunjukkan penangguhan interupsi telah diketahui CPU.
- ⌘ *Clock*, kontrol untuk sinkronisasi operasi antar modul.
- ⌘ *Reset*, digunakan untuk menginisialisasi seluruh modu

Sinyal kontrol secara fisik

- ⌘ Konduktor listrik paralel yang menghubungkan modul – modul.
- ⌘ Konduktor adalah saluran utama pada PCB motherboard dengan layout tertentu sehingga didapat fleksibilitas penggunaan.
- ⌘ Untuk modul I/O biasanya dibuat slot *bus* yang mudah dipasang dan dilepas
 - ☑ Slot PCI
 - ☑ Slot ISA.
- ⌘ Untuk chips akan terhubung melalui pinnya

Prinsip operasi *bus*

Operasi pengiriman data ke modul

1. Meminta penggunaan *bus*.
2. Apabila telah disetujui, modul akan memindahkan data yang diinginkan ke modul yang dituju

Prinsip operasi *bus*

Operasi meminta data dari modul lainnya

1. Meminta penggunaan *bus*.
2. Mengirim *request* ke modul yang dituju melalui saluran kontrol dan alamat yang sesuai.
3. Menunggu modul yang dituju mengirimkan data yang diinginkan

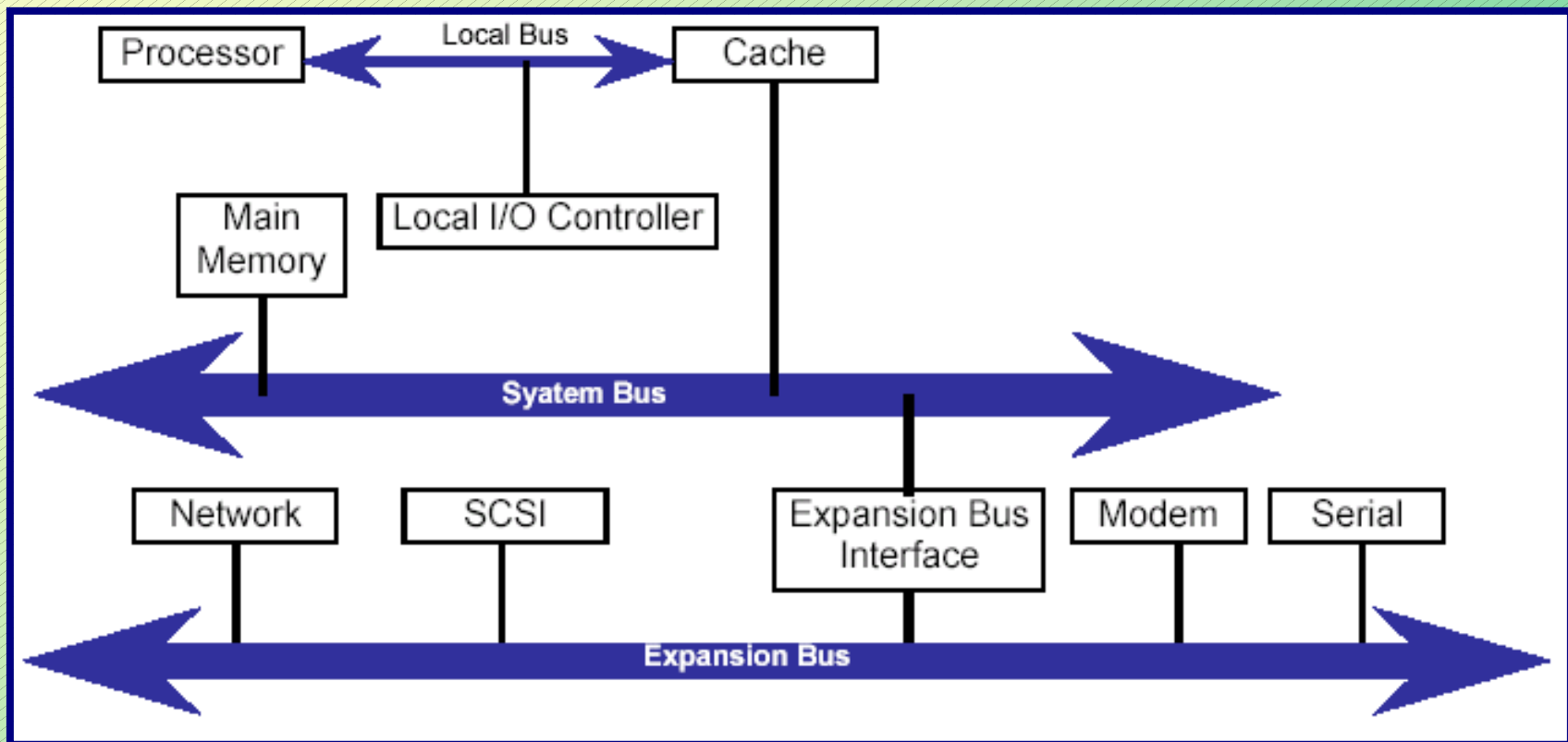
Hierarki Multiple Bus

⌘ Bila terlalu banyak modul atau perangkat dihubungkan pada *bus* maka akan terjadi penurunan kinerja

Faktor – faktor :

- ☒ Semakin besar delay propagasi untuk mengkoordinasikan penggunaan *bus*.
- ☒ Antrian penggunaan *bus* semakin panjang.
- ☒ Dimungkinkan habisnya kapasitas transfer *bus* sehingga memperlambat data.

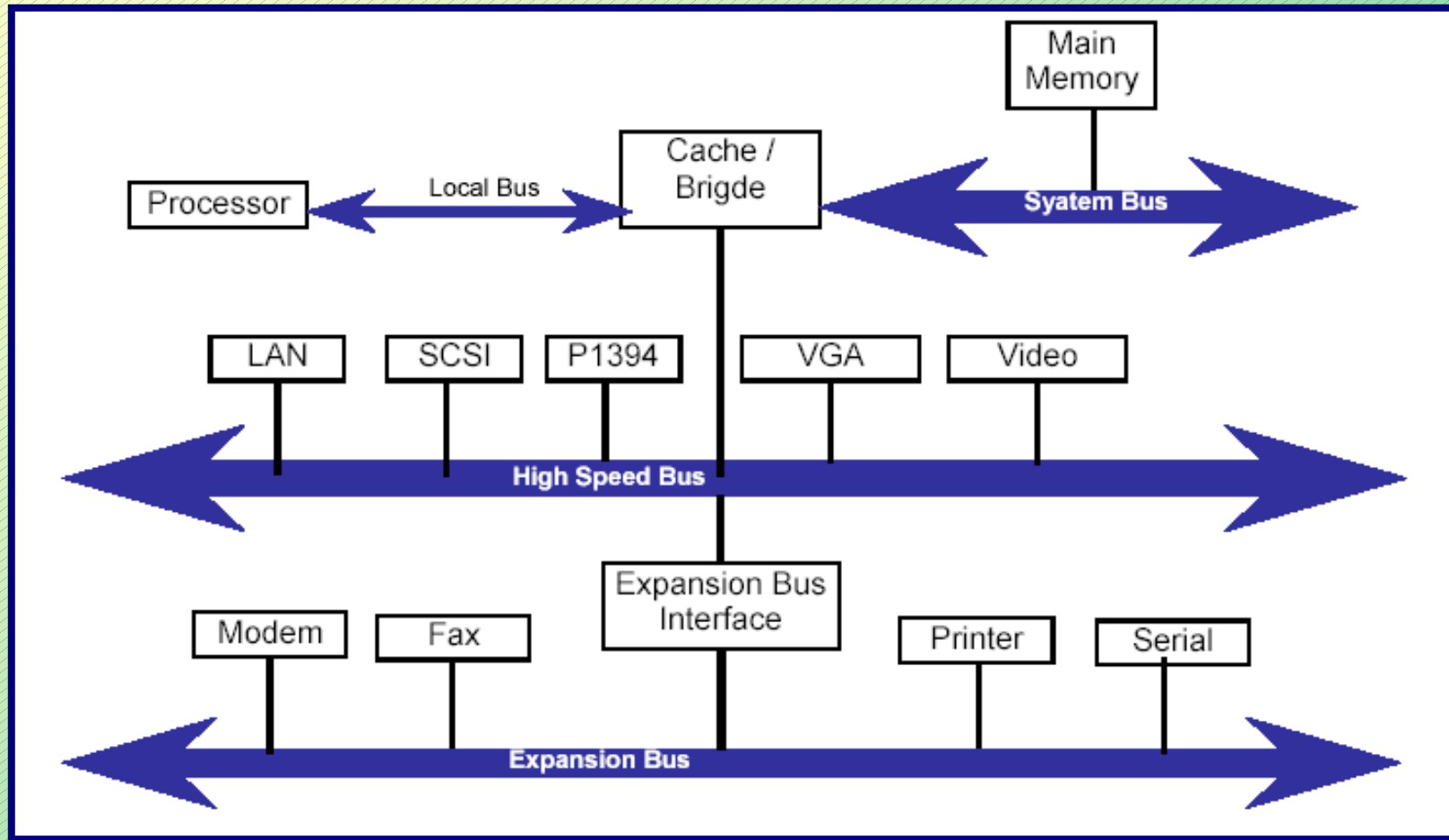
Arsitektur bus jamak tradisional



Arsitektur bus jamak

- ⌘ Prosesor, cache memori dan memori utama terletak pada bus tersendiri pada level tertinggi karena modul – modul tersebut memiliki karakteristik pertukaran data yang tinggi.
- ⌘ Pada arsitektur berkinerja tinggi, modul – modul I/O diklasifikasikan menjadi dua,
 - ☑ Memerlukan transfer data berkecepatan tinggi
 - ☑ Memerlukan transfer data berkecepatan rendah.
- ⌘ Modul dengan transfer data berkecepatan tinggi disambungkan dengan bus berkecepatan tinggi pula,
- ⌘ Modul yang tidak memerlukan transfer data cepat disambungkan pada bus ekspansi

Arsitektur bus jamak kinerja tinggi



Arsitektur bus jamak kinerja tinggi

⌘ Keuntungan hierarki bus jamak kinerja tinggi

- ☑ Bus berkecepatan tinggi lebih terintegrasi dengan prosesor.
- ☑ Perubahan pada arsitektur prosesor tidak begitu mempengaruhi kinerja bus

Kesimpulan

1. Komputer tersusun atas beberapa komponen penting seperti CPU, memori, perangkat I/O. Sistem bus adalah penghubung bagi keseluruhan komponen komputer dalam menjalankan tugasnya.
2. Kumpulan lintasan atau saluran berbagai modul disebut struktur interkoneksi. Rancangan struktur interkoneksi sangat bergantung pada jenis dan karakteristik pertukaran datanya.
3. Secara umum fungsi saluran bus dikategorikan dalam tiga bagian, yaitu saluran data, saluran alamat dan saluran kontrol.

Soal-soal

1. Jelaskan struktur antar hubungan dan beri contohnya.
2. Bila terlalu banyak modul atau perangkat dihubungkan pada bus maka akan terjadi penurunan kinerja, sebutkan penyebabnya?
3. Umumnya perangkat berprioritas paling rendah memiliki waktu tunggu rata-rata yang paling singkat. Dengan dasar ini biasanya CPU diberi prioritas tertinggi pada SBI. Sebutkan alasan perangkat berprioritas 16 memiliki waktu tunggu rata-rata paling rendah? Dibawah kondisi seperti apa keadaan diatas tidak berlaku?

