

# 3

## Perancangan arsitektur sistem untuk komputer multimedia seluler

*Bab ini membahas arsitektur sistem komputer portabel, yang disebut Mobile Digital Companion, yang menyediakan dukungan untuk menangani aplikasi multimedia secara hemat energi. Karena masa pakai baterai terbatas dan bobot baterai merupakan faktor penting untuk ukuran dan bobot Mobile Digital Companion, manajemen energi memainkan peran penting dalam arsitektur. Karena Companion harus tetap dapat digunakan di berbagai lingkungan, Companion harus fleksibel dan dapat beradaptasi dengan berbagai kondisi pengoperasian.*

*Mobile Digital Companion memiliki arsitektur tidak konvensional yang menghemat energi dengan menggunakan dekomposisi sistem pada berbagai tingkat arsitektur dan memanfaatkan lokalitas referensi dengan modul khusus yang dioptimalkan. Pendekatan ini didasarkan pada fungsionalitas khusus dan penggunaan ekstensif teknik pengurangan energi di semua tingkat desain sistem. Sistem ini memiliki arsitektur dengan prosesor tujuan umum yang disertai dengan satu set modul heterogen yang dapat diprogram secara otonom, masing-masing menyediakan implementasi tugas khusus yang hemat energi. Sakelar jaringan komunikasi internal yang dapat dikonfigurasi ulang mengeksplorasi lokalitas referensi dan menghilangkan salinan data yang boros.*

### 3.1 Pendahuluan

Salah satu masalah yang paling menarik dalam komputasi seluler adalah menjaga konsumsi energi ponsel tetap rendah. Bab ini membahas arsitektur sistem komputer portabel, yang disebut *Pendamping Digital Seluler*, yang menyediakan dukungan untuk menangani aplikasi multimedia secara hemat energi. Itu *Pendamping Digital Seluler* dirancang sebagai bagian dari MOBY DICK proyek [62][50]. Proyek ini membahas fundamental

---

<sup>1</sup> Bagian utama dari bab ini telah disajikan dalam dua presentasi di bagian pertama *Sekolah musim panas Euromicro pada komputasi seluler '98*, Agustus 1998 [30][62].

masalah dalam arsitektur, desain dan implementasi komputer genggam berdaya rendah, dengan penekanan khusus pada konservasi energi.

### **3.1.1 Sistem seluler saat ini**

Komunitas riset dan industri telah mengeluarkan banyak upaya untuk komputasi mobile dan desain komputer portabel dan perangkat komunikasi. Gadget murah yang cukup kecil untuk dimasukkan ke dalam saku (seperti PDA, komputer palmtop dan kamera digital) bergabung dengan jajaran komputer notebook, telepon seluler, dan video game. Komputer portabel saat ini menjalankan aplikasi interaktif paling umum seperti pengolah kata dan spreadsheet tanpa penundaan komputasi yang nyata. Perangkat ini sekarang mendukung berbagai fungsi yang terus berkembang, dan beberapa perangkat menyatu menjadi satu unit [37]. Komputer pribadi menjadi bagian integral dari kehidupan sehari-hari, seperti peralatan portabel seperti jam tangan dan telepon seluler selama beberapa tahun terakhir. Munculnya komunikasi nirkabel dan kemajuan besar dalam teknologi yang memungkinkan kita untuk mengintegrasikan banyak fungsi dalam satu chip telah membuka banyak kemungkinan untuk komputasi mobile. Komunikasi, pemrosesan data, dan hiburan akan didukung oleh platform yang sama, ditingkatkan oleh konektivitas seluruh dunia yang disediakan oleh Internet.

Pertama-tama kita akan melihat sekilas berbagai sistem seluler yang ada di pasaran saat ini. Perhatikan bahwa banyak dari sistem ini tidak memiliki kemampuan jaringan nirkabel built-in, melainkan mengandalkan modem nirkabel eksternal untuk konektivitas nirkabel. Modem nirkabel pada umumnya didasarkan pada telepon seluler, atau pada produk WLAN (LAN nirkabel). LAN nirkabel umumnya ditujukan untuk penggunaan dalam ruangan jarak pendek (beberapa ratus meter), sebagai lawan dari sistem seluler dengan jangkauan beberapa kilometer di luar ruangan. LAN nirkabel memiliki kecepatan data yang lebih tinggi daripada telepon seluler, pada urutan megabit per detik, dan ukurannya lebih kecil, konsumsi dayanya sebanding.

Sistem seluler saat ini dapat diklasifikasikan ke dalam kategori berikut berdasarkan fungsi dan faktor bentuknya.

- *Laptop* – Laptop bukanlah sistem yang benar-benar mobile karena terlalu besar dan terlalu berat. Pada dasarnya mereka hanya mesin desktop kecil yang dioperasikan dengan baterai. Komunikasi nirkabel umumnya didasarkan pada produk WLAN yang dapat dicolokkan sebagai kartu PC.
- *tablet pena* – Tablet pena dapat dilihat sebagai laptop tanpa keyboard. Interaksi dengan pen tablet adalah melalui input pena. Dalam kebanyakan kasus, pena menggantikan mouse sebagai perangkat penunjuk. Beberapa tablet memiliki modem radio internal, sedangkan tablet lainnya memerlukan modem radio eksternal. Secara umum, terminal ini tidak berbeda dari desktop rata-rata.
- *Buku virtual* – Baru-baru ini telah diperkenalkan beberapa produk yang menggantikan kertas sebagai media membaca dan browsing berbagai macam bahan [13][59][65]. Sistem ini memiliki kualitas tampilan yang baik, dan arsitektur yang agak konvensional. Masukan pengguna terbatas pada beberapa tombol, dan pena.

- *Komputer Pribadi Genggam (HPC)* – Sistem kategori ini pada dasarnya adalah laptop mini. Mereka dicirikan oleh keyboard faktor bentuk yang diperkecil dan tampilan resolusi setengah VGA. Mereka biasanya menjalankan versi aplikasi Windows yang diperkecil, termasuk pengolah kata, presentasi, dan perangkat lunak penjadwalan. Komunikasi biasanya berupa modem jalur kabel dan port inframerah.
- *Asisten Digital Pribadi (PDA)* – PDA umumnya merupakan perangkat monolitik tanpa keyboard (walaupun beberapa memiliki keyboard berukuran kecil) dan pas di tangan pengguna. Dengan demikian, input pena adalah norma, dan pengenalan tulisan tangan adalah hal biasa. Kemampuan komunikasi melibatkan port dok atau port serial untuk menghubungkan dan menyinkronkan dengan komputer desktop, dan mungkin modem.
- *Ponsel pintar* – Meskipun telepon seluler mungkin memiliki beberapa fungsi periferifal seperti kalkulator, buku tanggal, atau buku telepon, mereka adalah alat komunikasi utama. Perangkat kombinasi seperti Nokia 9000 pada dasarnya adalah perangkat mirip PC yang terpasang pada telepon seluler.
- *Terminal nirkabel* – Sistem ini pada dasarnya tidak lebih dari input dan output nirkabel yang diperluas dari mesin desktop yang bertindak sebagai server. Sistem ini dirancang untuk memanfaatkan jaringan nirkabel berkecepatan tinggi untuk mengurangi jumlah komputasi yang diperlukan pada perangkat portabel.

Akan menjadi jelas bahwa sistem bergerak saat ini terutama merupakan terminal pemrosesan data atau terminal komunikasi. Tren terminal pemrosesan data adalah mengecilkan PC desktop serba guna menjadi paket yang dapat dibawa dengan nyaman. Bahkan PDA tidak jauh dari model tujuan umum, baik arsitektur maupun dari segi model penggunaan.

### 3.1.2 Masa depan: *Mobile Digital Companion*

Topik penelitian ini adalah arsitektur perangkat genggam masa depan, yang disebut *Pendamping Digital Seluler* (dalam tesis ini juga disebut sebagai *Pendamping*). SEBUAH *Pendamping Digital Seluler* akan menjadi mesin pribadi, dan pengguna cenderung menjadi sangat bergantung padanya.

Itu *Pendamping Digital Seluler* adalah komputer portabel pribadi kecil dan perangkat komunikasi nirkabel yang dapat menggantikan uang tunai, buku cek, paspor, kunci, buku harian, telepon, pager, peta dan mungkin juga tas kantor [50]. Ini akan menyerupai PDA, yaitu terlihat seperti PDA biasa, tetapi fungsi dan penggunaan khas sistem sangat berbeda. Aplikasi khas dari *Pendamping Digital Seluler* adalah diary, e-mail, web browsing, note-taking, walkman, video player dan pembayaran elektronik. Itu *Pendamping Digital Seluler* adalah perangkat genggam yang *miskin sumber daya*, yaitu sejumlah kecil memori, masa pakai baterai terbatas, daya pemrosesan rendah, dan terhubung dengan lingkungan melalui jaringan (nirkabel) dengan konektivitas variabel. Tujuan utama kami dalam merancang arsitektur adalah untuk mendukung berbagai macam aplikasi untuk perangkat seluler yang memanfaatkan sumber daya yang tersedia secara efisien. Pendamping tersebut harus memenuhi beberapa persyaratan utama: kinerja tinggi, hemat energi, gagasan Quality of Service (QoS), ukuran kecil, dan kompleksitas desain rendah.

Itu *Pendamping Digital Seluler* lebih dari sekedar mesin kecil untuk digunakan oleh satu orang pada satu waktu seperti penyelenggara tradisional dan asisten desktop. Kami membedakan dua jenis sistem: 'pendamping desktop' dan '*Pendamping Digital Seluler*'. Pendamping desktop adalah mesin genggam yang dirancang untuk memberi pengguna roaming akses ke data bisnis dan aplikasi mereka saat dalam perjalanan. Desktop pendamping dirancang dan dioptimalkan untuk kompatibilitas dan komunikasi dengan mesin desktop pengguna, misalnya melalui modem, inframerah atau stasiun dok. Contoh tipikal pendamping desktop adalah PDA atau (sub) notebook yang menjalankan Windows CE [26].

Itu *Pendamping Digital Seluler* memperluas gagasan pendamping desktop dalam beberapa cara.

- Ini akan menjalankan aplikasi yang biasanya ditemukan di pendamping desktop, tetapi juga akan menjalankan aplikasi lain menggunakan *pelayanan publik eksternal*. SEBUAH *Pendamping Digital Seluler* berinteraksi dengan lingkungan dan merupakan bagian dari sistem terdistribusi terbuka. Ia perlu berkomunikasi dengan - mungkin bermusuhan - layanan eksternal di bawah berbagai kondisi komunikasi dan operasi, dan tidak hanya ke 'master' desktopnya.
- *Komputasi multimedia* juga akan menjadi bagian penting dari *Sahabat Digital Seluler*. Jika komputer mobile harus digunakan untuk pekerjaan sehari-hari, maka perangkat multimedia, seperti audio dan video harus disertakan dalam sistem. Saat ini, ada beberapa perangkat multimedia portabel yang tersedia (kamera digital, MP3man, dll.), tetapi semua sistem ini tidak lebih dari perangkat khusus. Yang kurang adalah integrasi yang baik antara semua perangkat ini.
- Semua pendamping desktop saat ini memiliki fasilitas komunikasi untuk berkomunikasi dengan master desktop. Namun, karena ketergantungan pada penyimpanan dan komputasi informasi yang dapat diakses jaringan meningkat, keinginan untuk mengakses jaringan di mana-mana membutuhkan lebih banyak *jaringan nirkabel canggih sophisticated* kemampuan. Akses jaringan harus mendukung heterogenitas dalam banyak dimensi (media transportasi, protokol, tipe data, dll.).

Faktor yang paling penting, yang akan menentukan keberhasilan *Pendamping Digital Seluler*, adalah utilitas dan kenyamanan sistem. Fitur penting adalah antarmuka dan interaksi dengan pengguna: input dan output suara dan gambar (pengenalan ucapan dan pola) akan menjadi fungsi utama. Penggunaan tipe data multimedia real-time seperti video, pidato, animasi dan musik sangat meningkatkan kegunaan, kualitas, produktivitas, dan kenikmatan sistem ini. Aplikasi multimedia memerlukan pengangkutan beberapa aliran media yang disinkronkan. Beberapa aliran ini (biasanya aliran video) memiliki bandwidth tinggi dan persyaratan waktu nyata yang ketat. Aplikasi ini juga mencakup sejumlah besar interaksi pengguna. Sebagian besar aplikasi yang kami anggap tidak hanya memerlukan Kualitas Layanan tertentu untuk komunikasi (seperti bandwidth tinggi dan latensi rendah), tetapi juga sejumlah besar daya komputasi.

Mobile Digital Companion adalah perangkat yang cukup serbaguna. Namun demikian fungsi-fungsi ini harus disediakan oleh jumlah perangkat keras yang relatif kecil karena

persyaratan untuk Sahabat adalah ukuran dan berat yang kecil. Karena sebagian besar penelitian baterai saat ini tidak memprediksi perubahan substansial dalam energi yang tersedia dalam baterai, efisiensi energi memainkan peran penting dalam arsitektur baterai. *Sahabat Digital Seluler*. Solusi terintegrasi yang mengurangi jumlah chip sangat diinginkan.

### 3.1.3 Pendekatan

Pendekatan untuk mencapai sistem seperti yang dijelaskan di atas adalah memiliki modul otonom yang dapat dikonfigurasi ulang seperti perangkat jaringan, video, dan audio, yang saling terhubung dengan sakelar alih-alih bus, dan melepas beban sebanyak mungkin dari CPU ke modul yang dapat diprogram. yang ditempatkan di aliran data. Dengan demikian, komunikasi antar komponen tidak disiarkan melalui bus tetapi dikirimkan tepat di tempat yang dibutuhkan, pekerjaan dilakukan di mana data melewati, melewati memori. Modul secara otomatis memasuki mode hemat energi dan menyesuaikan diri dengan kondisi sumber daya, lingkungan, dan kebutuhan pengguna saat ini. Jumlah buffer diminimalkan, dan jika diperlukan sama sekali, itu ditempatkan tepat di jalur data, di mana dibutuhkan. Untuk mendukung ini, sistem operasi harus menjadi kecil,

Interkoneksi arsitektur didasarkan pada sakelar, yang disebut *Gurita*, yang menghubungkan prosesor tujuan umum, perangkat (multimedia), dan antarmuka jaringan nirkabel. Itu *Gurita* switch adalah subjek Bab 4. Meskipun tidak secara unik ditujukan pada area meja, pekerjaan kami terkait dengan proyek seperti yang dijelaskan dalam [4][20] dan [32] di mana bus workstation tradisional digantikan oleh jaringan berkecepatan tinggi di untuk menghilangkan hambatan komunikasi yang ada dalam sistem saat ini.

### 3.1.4 Garis Besar

Kami pertama menunjukkan di Bagian 3.2 tantangan utama dalam desain sistem seluler yang akan memberikan motif mengapa ada kebutuhan untuk merevisi arsitektur sistem komputer portabel. Bagian 3.3 kemudian menjelaskan filosofi di balik arsitektur *Pendamping Digital Seluler*, dan memperkenalkan berbagai mekanisme dasar yang digunakan: pendekatan connection-centric, kontrol waktu, kerangka Kualitas Layanan, dan akhirnya menyajikan arsitektur sistem dasar. Kemudian kami akan memberikan gambaran tentang keadaan seni dalam komputasi multimedia seluler di Bagian 3.4. Akhirnya, kami menyajikan ringkasan dan kesimpulan di Bagian 3.5.

## 3.2 Masalah desain sistem seluler

Ada kelas pengguna baru dan berkembang yang kebutuhan komputasi utamanya adalah untuk mengakses infrastruktur informasi, sumber daya komputasi, dan sistem interaktif real-time serta komunikasi langsung dengan orang lain. Aplikasi ini, yaitu

berorientasi komunikasi daripada berorientasi komputasi, adalah motivasi untuk pemeriksaan ulang persyaratan arsitektur sistem dan perangkat keras yang dibutuhkan. Aplikasi ini memerlukan pendamping digital seluler pribadi yang terutama memiliki dukungan untuk komunikasi real-time bandwidth tinggi dan kemampuan multimedia [70]. Komputasi tujuan umum berkinerja tinggi bukanlah persyaratan yang menonjol.

Perbaikan terbaru dalam teknologi sirkuit dan pengembangan perangkat lunak telah memungkinkan penggunaan tipe data waktu nyata seperti video, ucapan, animasi, dan musik. Seiring berkembangnya komputer seluler, dukungan untuk aplikasi kaya multimedia akan menjadi standar. Diharapkan pada tahun 2000 90 persen dari siklus desktop akan dihabiskan untuk aplikasi multimedia [16].

Industri komputer telah membuat kemajuan besar dalam pengembangan komputasi mobile dan desain komputer portabel. Ini sebagian karena kemajuan teknologi baru-baru ini. Sistem ini umumnya didasarkan pada arsitektur komputer pribadi berkinerja tinggi yang memiliki beberapa ketentuan untuk komputasi nirkabel dan bentuk manajemen daya yang belum sempurna. Pada bagian ini kami akan menunjukkan bahwa pendekatan seperti itu tidak cukup jika kita ingin memiliki mesin genggam dengan kemampuan multimedia yang dapat digunakan dengan nyaman di lingkungan nirkabel.

### **3.2.1 Mobilitas**

Munculnya aplikasi dan layanan multimedia baru yang memanfaatkan pertumbuhan komputasi seluler bergantung pada ketersediaan infrastruktur nirkabel pita lebar yang fleksibel. Masalah teknis utama dari infrastruktur ini termasuk kontrol Kualitas Layanan dan integrasi perangkat lunak aplikasi. Sistem seluler akan memiliki serangkaian tantangan yang timbul dari beragam tipe data dengan persyaratan kualitas layanan (QoS) yang berbeda yang akan mereka tangani, sumber daya baterai yang terbatas, kebutuhan mereka untuk beroperasi di lingkungan yang mungkin tidak dapat diprediksi, tidak aman, dan berubah, dan mobilitas mereka menghasilkan perubahan set layanan yang tersedia.

Berikut ini adalah tantangan teknologi utama yang kami yakini perlu diatasi sebelum sistem seluler seperti Mobile Digital Companion menjadi nyata.

- *Efisiensi energi* – Karena komputer portabel saat ini telah terbukti mampu membantu pengguna ponsel dalam pekerjaan sehari-hari mereka, menjadi semakin jelas bahwa hanya meningkatkan kekuatan pemrosesan dan meningkatkan bandwidth jaringan mentah tidak berarti perangkat yang lebih baik. Berat dan masa pakai baterai menjadi lebih penting daripada kecepatan pemrosesan murni. Konsumsi energi menjadi faktor pembatas dalam jumlah fungsionalitas yang dapat ditempatkan di komputer portabel seperti PDA dan laptop.
- *Infrastruktur* – Desain sistem seluler tidak dapat dilakukan secara terpisah. Sistem seluler masa depan kemungkinan akan dirancang untuk beroperasi secara mandiri, tetapi juga sangat mungkin bergantung pada infrastruktur eksternal untuk mengakses informasi dalam bentuk apa pun. Ponsel kemungkinan akan menghadapi banyak, lingkungan yang sangat beragam dan berbagai infrastruktur jaringan. Selain itu, ponsel dapat bervariasi di banyak sumbu, termasuk ukuran layar, kedalaman warna, kekuatan pemrosesan, dan fungsi yang tersedia. Server (atau agen proxy yang ditempatkan di antara ponsel dan server) dapat melakukan

komputasi dan penyimpanan atas nama klien. Pemisahan fungsi antara sistem nirkabel dan server yang berada di jaringan merupakan keputusan arsitektur penting yang menentukan di mana aplikasi dapat berjalan, di mana data dapat disimpan, kompleksitas terminal, dan biaya layanan komunikasi.

- *Kemampuan beradaptasi* – Sistem seluler nirkabel menghadapi berbagai jenis variabilitas di lingkungannya baik dalam jangka pendek maupun jangka panjang. Sistem seluler akan membutuhkan kemampuan untuk beradaptasi dengan kondisi yang berubah ini, dan akan membutuhkan radio adaptif, protokol, codec, dan sebagainya. Kontrol kesalahan adaptif dan kompresi adaptif adalah contoh dari teknik tersebut.
- *Dapat dikonfigurasi ulang* – Untuk memerangi tingkat variasi yang lebih tinggi dalam lingkungan operasional daripada yang dimungkinkan dengan sistem yang dapat disesuaikan, arsitektur yang dapat dikonfigurasi ulang dapat digunakan yang memungkinkan fungsi perangkat lunak dan perangkat keras baru untuk diunduh. Jadi daripada mengubah parameter algoritme ke kondisi saat ini, seperangkat protokol dan algoritme yang sama sekali baru dapat digunakan. Pendekatan alternatif untuk beradaptasi dengan perubahan lingkungan adalah memiliki sistem seluler dengan semua skenario yang mungkin ada di dalamnya. Sistem multimode seperti itu menjadi mahal, dan relatif tidak fleksibel.
- *Keamanan* – Ketika komputer menjadi lebih terlibat dalam aktivitas pribadi dan bisnis seseorang, keamanan seperti kerahasiaan, privasi, keaslian, dan penolakan menjadi perhatian penting. Penerapan kriptografi yang bijaksana dapat memenuhi masalah ini, asalkan sistem menyediakan lingkungan yang aman bagi pengguna di mana algoritme kriptografi yang sesuai dapat melakukan pekerjaan mereka tanpa risiko kompromi atau kehilangan kunci atau data rahasia.
- *Antarmuka pengguna* – Keyboard tradisional dan antarmuka berbasis tampilan tidak memadai untuk sistem seluler masa depan karena ukuran dan berat yang dibutuhkan sistem ini kecil. Sebaliknya, antarmuka yang secara intrinsik lebih sederhana berdasarkan ucapan, sentuhan, pena, dan sebagainya lebih mungkin digunakan dan lebih memadai untuk faktor bentuk kecil dari sistem ini. Karena sistem ini akan menjadi peralatan konsumen yang digunakan oleh non-ahli, lingkungan yang kompleks harus tetap tersembunyi dari pengguna, atau disajikan pada tingkat yang dapat dengan mudah dipahami oleh pengguna.

Selebihnya kami akan fokus pada isu-isu yang terkait dengan konsumsi energi, yaitu efisiensi energi dan kemampuan beradaptasi.

### **3.2.2 Multimedia**

Sistem yang diperlukan untuk aplikasi multimedia dalam lingkungan seluler harus memenuhi persyaratan yang berbeda dari yang dapat ditawarkan oleh workstation saat ini di lingkungan desktop. Karakteristik dasar yang perlu didukung oleh sistem dan aplikasi multimedia adalah [17]:

- *Tipe data media berkelanjutan* – Fungsi media biasanya melibatkan pemrosesan aliran data yang berkelanjutan, yang menyiratkan bahwa lokalitas temporal dalam akses memori data tidak lagi berlaku. Hebatnya, cache data mungkin menjadi penghalang untuk high

kinerja dan efisiensi energi untuk tipe data media berkelanjutan karena prosesor akan mengalami kesalahan cache terus-menerus.

- *Memberikan Kualitas Layanan (QoS)* – Alih-alih memberikan kinerja maksimal, sistem harus menyediakan QoS yang cukup untuk persepsi kualitatif dalam aplikasi seperti video.
- *Paralelisme berbutir halus* – Fungsi multimedia yang umum seperti pemrosesan gambar, suara, dan sinyal memerlukan paralelisme yang halus karena operasi yang sama di seluruh urutan data dilakukan. Operasi dasar relatif kecil.
- *Paralelisme berbutir kasar* – Dalam banyak aplikasi, pipeline fungsi memproses satu aliran data untuk menghasilkan hasil akhir.
- *Lokalitas referensi instruksi tinggi* – Operasi pada data biasanya menunjukkan lokalitas temporal dan spasial yang tinggi untuk instruksi.
- *Bandwidth memori tinggi* – Banyak aplikasi multimedia membutuhkan bandwidth memori yang besar untuk kumpulan data besar yang memiliki lokalitas terbatas.
- *Bandwidth jaringan tinggi* – Streaming data – seperti video dan gambar dari sumber eksternal – membutuhkan bandwidth jaringan dan I/O yang tinggi.

Aplikasi multimedia terdistribusi yang berjalan di lingkungan seluler memiliki sejumlah karakteristik khusus. Banyak sistem seluler nirkabel masa depan akan beroperasi di berbagai lingkungan yang relatif tidak diatur seperti LAN rumah dan tempat kerja dengan tingkat interferensi waktu yang bervariasi. Jaringan telekomunikasi seluler yang ada juga dapat digunakan untuk menyediakan akses nirkabel ke jaringan komputer berkabel. Aplikasi tidak dapat mengandalkan jaringan nirkabel untuk menyediakan throughput yang tinggi atau waktu respons yang cepat. Dua karakteristik yang paling mendasar adalah:

- SEBUAH *lingkungan pemrosesan yang heterogen* (termasuk host seluler berdaya relatif rendah) dan,
- *fluktuasi yang cepat dan masif* dalam kualitas layanan yang disediakan oleh infrastruktur komunikasi yang mendasarinya.

Kontrol QoS adalah fitur utama untuk pemanfaatan sumber daya yang efisien dalam jaringan nirkabel yang mendukung multimedia seluler. Model alokasi sumber daya statis tradisional kurang fleksibel, dan dengan demikian mengatasi interaktivitas multimedia dan mobilitas sesi dengan buruk.

Tantangannya adalah untuk mempertahankan kualitas end-to-end yang dirasakan tinggi tanpa membatasi aplikasi ke titik di mana mereka tidak lagi berguna. Jaringan multimedia membutuhkan setidaknya alokasi bandwidth minimum tertentu untuk kinerja aplikasi yang memuaskan [58]. Persyaratan bandwidth minimum memiliki rentang dinamis yang luas tergantung pada ekspektasi kualitas pengguna, model penggunaan aplikasi, dan toleransi aplikasi terhadap degradasi. Selain itu, beberapa aplikasi dapat dengan anggun beradaptasi dengan degradasi jaringan sporadis sambil tetap memberikan kinerja yang dapat diterima. Misalnya, sementara aplikasi video-on-demand secara umum dapat mentolerir peraturan bit rate dalam rentang dinamis kecil, aplikasi seperti telekonferensi mungkin memiliki rentang dinamis yang lebih besar untuk kontrol bit rate.



### 3.2.3 Keterbatasan sumber daya energi

Meskipun komputer portabel saat ini telah terbukti mampu membantu pengguna ponsel dalam pekerjaan sehari-hari mereka, menjadi semakin jelas bahwa hanya meningkatkan kekuatan pemrosesan dan meningkatkan bandwidth jaringan mentah tidak berarti perangkat yang lebih baik. Berat dan masa pakai baterai menjadi lebih penting daripada kecepatan pemrosesan murni. Kedua faktor ini terkait dengan ukuran baterai: untuk mengoperasikan komputer lebih lama tanpa mengisi ulang, kita membutuhkan baterai yang lebih besar dan lebih berat. Oleh karena itu, batasannya adalah jumlah total energi listrik yang tersimpan dalam baterai yang tersedia untuk operasi. Teknologi baterai telah meningkat dengan kecepatan glasial dibandingkan dengan kecepatan di mana jumlah daya pemrosesan dalam sistem seluler meningkat sementara ukurannya menurun.

Untuk memperpanjang masa pakai baterai, kami harus merancang sistem agar lebih efisien dalam penggunaan energi ini. Namun, hingga saat ini, penelitian masih terfokus pada kinerja dan desain sirkuit. Karena keterbatasan fisik yang mendasar, kemajuan menuju pengurangan energi lebih lanjut harus ditemukan di luar level chip. Kunci efisiensi energi di sistem seluler masa depan adalah tingkat yang lebih tinggi: arsitektur dan protokol hemat energi, aplikasi yang sadar energi, dll.

Sebagian besar produk elektronik yang kritis energi jauh lebih kompleks daripada satu chip. Pada sebagian besar produk elektronik, komponen digital mengkonsumsi sebagian kecil dari energi yang dikonsumsi. Komponen analog, elektro-mekanik dan optik sering bertanggung jawab atas kontribusi besar terhadap anggaran daya komputer portabel [42]. Salah satu teknik paling sukses yang digunakan oleh desainer di tingkat sistem adalah *manajemen daya dinamis*, di mana bagian sistem memiliki mode energi yang berbeda, dan bahkan dapat dimatikan sepenuhnya.

### 3.2.4 Masalah arsitektur sistem

Dalam desain tradisional ponsel, sejumlah area masalah dalam arsitektur perangkat keras dan perangkat lunak dapat diidentifikasi mengenai konsumsi energi [8]. Kami hanya akan menyebutkan beberapa.

- Masalah utama adalah kurangnya interaksi antara fasilitas perangkat keras untuk manajemen energi (mode hemat daya, interupsi perangkat yang 'membangunkan' CPU, dll.) dan sistem operasi dan perangkat lunak aplikasi. Secara khusus, driver perangkat, sistem operasi, dan aplikasi berusaha mengontrol perangkat keras secara mandiri. Kesalahan koordinasi ini menyebabkan perilaku tidak menentu yang tidak dapat dijelaskan. Contoh masalah ini adalah layar kosong yang tidak terduga selama presentasi, atau saat disk berputar ke atas dan ke bawah tanpa pemberitahuan (menyebabkan penundaan yang mengganggu).
- Kedua, peluang untuk menghemat energi tidak dimanfaatkan karena perangkat dikendalikan pada tingkat yang terlalu rendah, mengabaikan informasi tingkat tinggi tentang apa yang sebenarnya dibutuhkan pengguna selama operasi sistem.
- Ketiga, aplikasi mengasumsikan bahwa komputer selalu menyala. Asumsi ini seringkali menyebabkan konsumsi energi yang berlebihan. Misalnya, siklus pemungutan suara, ketika

aplikasi sedang menunggu tanggapan, sangat tidak efisien dari sudut pandang energi.

- Akhirnya, perangkat lunak sistem operasi dan perangkat lunak jaringan saat ini menekankan fleksibilitas dan kinerja, dan dibangun dari komponen yang dikembangkan oleh kelompok independen. Dalam desain sistem, peran kunci terletak pada pengembangan antarmuka. Praktik kerja yang baik adalah mendefinisikan antarmuka secara hierarkis, karena kompleksitas sistem direduksi menjadi proporsi yang dapat dikelola. Pendekatan semacam itu juga diarahkan oleh pendekatan standarisasi seperti struktur lapisan jaringan ISO/OSI. Namun, hasil dari fleksibilitas dan pendekatan pengembangan ini adalah bahwa dalam banyak kasus banyak salinan data yang tidak perlu terjadi di antara modul yang berbeda. Tumpukan protokol non-penyalinan memang ada [67], tetapi tidak banyak digunakan. Operasi seperti penyalinan data, servis interupsi, sakelar konteks, kompresi perangkat lunak, berada dalam sistem saat ini sering bertanggung jawab atas kinerja yang buruk dan konsumsi energi yang tinggi. Protokol jaringan yang tidak dirancang dengan baik yang tidak secara efisien menggunakan salah satu perangkat seluler yang paling menuntut energi, antarmuka nirkabel, juga membuang banyak energi.

Visi kami adalah bahwa ada hubungan penting antara arsitektur perangkat keras, arsitektur sistem operasi, arsitektur aplikasi dan arsitektur antarmuka manusia, di mana masing-masing mendapat manfaat dari yang lain: aplikasi dapat beradaptasi dengan situasi daya jika mereka memiliki API sistem operasi yang sesuai untuk melakukannya; sistem operasi dapat meminimalkan konsumsi energi dengan menjaga sebanyak mungkin komponen dimatikan; arsitektur perangkat keras dapat dirancang untuk merutekan jalur data sedemikian rupa sehingga, untuk fungsi tertentu, hanya komponen minimum yang perlu aktif.

### **3.2.5 Integrasi tingkat sistem**

Alur desain suatu sistem terdiri dari berbagai tingkat abstraksi. Dengan hati-hati merancang semua komponen yang membentuk sistem mobile (yaitu komponen perangkat keras, arsitektur, sistem operasi, protokol, dan aplikasi) secara koheren dan terintegrasi, adalah mungkin untuk meminimalkan overhead yang dihasilkan dari penggunaan operasi ini dan mengurangi konsumsi energi. Setiap aplikasi tunggal, driver perangkat, atau modul perangkat keras tidak memiliki pengetahuan yang cukup tentang status seluruh sistem untuk secara efektif membuat keputusan otonom mengenai manajemen energi.

Sebuah aspek penting dari aliran desain adalah hubungan dan umpan balik antara tingkat. Mengingat spesifikasi desain, seorang desainer dihadapkan dengan beberapa pilihan yang berbeda pada tingkat abstraksi yang berbeda. Perancang harus memilih algoritma tertentu, merancang atau menggunakan arsitektur yang dapat digunakan untuk itu, dan menentukan berbagai parameter seperti tegangan suplai dan frekuensi clock. *Iniruang desain multi-dimensi* menawarkan berbagai kemungkinan trade-off. Keputusan desain yang paling efektif berasal dari memilih dan mengoptimalkan arsitektur dan algoritma pada tingkat tertinggi. Telah ditunjukkan oleh beberapa peneliti [14] bahwa keputusan desain tingkat sistem dan arsitektur dapat memiliki dampak dramatis pada konsumsi daya. Namun, ketika merancang sebuah sistem, merupakan masalah untuk memprediksi konsekuensi dan efektivitas keputusan desain.

karena detail implementasi hanya dapat dimodelkan atau diperkirakan secara akurat pada tingkat teknologi dan bukan pada tingkat abstraksi yang lebih tinggi.

Kemampuan untuk mengintegrasikan beragam fungsi sistem pada chip yang sama memberikan tantangan dan peluang untuk melakukan desain arsitektur sistem dan optimisasi di berbagai lapisan dan fungsi sistem. Khususnya perangkat komputasi bergerak yang menggabungkan komputasi multimedia dan fungsi komunikasi menunjukkan kebutuhan akan integrasi tingkat sistem. Fungsi mulai dari pemrosesan audio dan video, modem radio, antarmuka nirkabel, mekanisme keamanan, dan aplikasi berorientasi antarmuka pengguna harus terintegrasi dalam perangkat portabel kecil dengan jumlah energi yang terbatas. Informasi yang dihasilkan oleh perangkat atau aplikasi harus melintasi dan diproses di semua lapisan ini, memberikan arsitek sistem ruang pertukaran desain yang kaya.

### **3.2.6 Kemampuan program dan kemampuan beradaptasi**

Karena komputer seluler harus tetap dapat digunakan di berbagai lingkungan, mereka harus mendukung skema dan protokol penyandian dan enkripsi yang berbeda agar sesuai dengan standar jaringan yang berbeda, dan untuk beradaptasi dengan berbagai kondisi operasi. Oleh karena itu, komputer seluler akan membutuhkan sejumlah besar sirkuit yang dapat disesuaikan untuk aplikasi tertentu agar tetap fleksibel dan kompetitif. *Keterprograman dan kemampuan beradaptasi* dengan demikian merupakan persyaratan penting untuk sistem seluler, karena ponsel harus cukup fleksibel untuk mengakomodasi berbagai layanan multimedia dan kemampuan komunikasi serta beradaptasi dengan berbagai kondisi operasi dengan cara (energi) yang efisien.

Persyaratan untuk programabilitas dalam sistem pada sebuah chip juga dipicu oleh alasan ekonomis [72]. ASIC yang dirancang dengan baik akan memecahkan masalah spesifik yang dirancang, tetapi mungkin bukan masalah yang sedikit dimodifikasi yang diperkenalkan setelah desain selesai. Selanjutnya, bahkan jika ASIC yang dimodifikasi dapat dikembangkan untuk masalah baru, sirkuit perangkat keras asli mungkin terlalu disesuaikan untuk digunakan kembali dalam generasi berikutnya. Selain itu, biaya tinggi yang terlibat dalam merancang dan mengimplementasikan chip tidak membenarkan desain sistem yang hanya mengimplementasikan satu aplikasi. Selanjutnya, karena persyaratan aplikasi meningkat pesat dan standar baru muncul cukup cepat, chip baru perlu sering dirancang.

### **3.2.7 Diskusi**

Pada dasarnya, ada dua jenis perangkat komputer untuk digunakan di jalan: komputer palm-top dan komputer notebook. Palm tops terutama digunakan untuk membuat catatan, buku janji elektronik, dan buku alamat. Komputer notebook adalah komputer pribadi bertenaga baterai, dan arsitektur terkini untuk komputer bergerak sangat terkait dengan arsitektur stasiun kerja berkinerja tinggi. Baik notebook maupun komputer pribadi umumnya menggunakan sistem operasi PC standar yang sama seperti Windows 98 atau Unix, aplikasi yang sama, menggunakan protokol komunikasi yang sama dan menggunakan arsitektur perangkat keras yang sama. Satu-satunya perbedaan adalah komputer portabel lebih kecil, memiliki baterai, antarmuka nirkabel, dan terkadang menggunakan komponen berdaya rendah. Masalah yang melekat pada komputasi seluler diabaikan (mis

protokol komunikasi masih didasarkan pada TCP/IP, meskipun ini berperilaku buruk di lingkungan nirkabel [6]), atau mencoba untuk memecahkan dengan kekerasan mengabaikan peningkatan konsumsi energi (misalnya kontrol kesalahan ekstensif, atau dekompresi perangkat lunak). Kemampuan beradaptasi dan programabilitas harus menjadi persyaratan utama dalam desain arsitektur komputer bergerak.

Dalam waktu dekat gadget elektronik kecil lainnya seperti Web-ponsel, MP3 man, game dan kamera digital akan terintegrasi dengan komputer portabel di *komputasi seluler pribadi* lingkungan Hidup. Ini tidak hanya menyebabkan permintaan yang lebih besar untuk daya komputasi, tetapi pada saat yang sama ukuran, berat, dan konsumsi energi harus kecil.

Kita memasuki era di mana setiap microchip akan memiliki miliaran transistor. Salah satu cara untuk menggunakan kesempatan ini adalah dengan terus memajukan arsitektur dan teknologi chip kami sebagai hal yang sama: membangun mikroprosesor yang merupakan versi rumit dari jenis yang dibangun saat ini [5]. Namun, hanya mengecilkan terminal pemrosesan data dan modem radio, memasangnya melalui bus, dan mengemasnya bersama-sama tidak mengurangi kemacetan arsitektural. Tantangan desain sebenarnya adalah untuk merekayasa sistem seluler terintegrasi di mana pemrosesan data dan komunikasi memiliki kepentingan yang sama dan dirancang satu sama lain dalam pikiran. Menghubungkan desain PC atau PDA saat ini dengan subsistem komunikasi yang tersedia, bukanlah solusi.

Jelas, ada kebutuhan untuk merevisi arsitektur sistem komputer portabel jika kita ingin memiliki mesin yang dapat digunakan dengan nyaman di lingkungan nirkabel. Diperlukan integrasi tingkat sistem dari arsitektur, sistem operasi, dan aplikasi seluler. Sistem harus memberikan solusi dengan keseimbangan yang tepat antara fleksibilitas dan efisiensi dengan menggunakan campuran hibrida dari tujuan umum dan pendekatan khusus aplikasi.

### 3.3 Arsitektur sistem Mobile Digital Companion

Pada bagian ini kami menjelaskan arsitektur dari *Sahabat Digital Seluler*. Sifat-sifat yang ingin dicapai oleh arsitektur adalah:

1. itu *fleksibilitas* untuk menangani berbagai layanan dan standar (multimedia) dan
2. itu *kemampuan beradaptasi* untuk mengakomodasi lingkungan saat ini untuk kondisi yang berubah dalam konektivitas komunikasi, tingkat keamanan yang diperlukan, dan sumber daya yang tersedia.
3. Parameter konfigurasi dapat disesuaikan sesuai dengan *persyaratan QoS*. Komponen-komponen arsitektur harus mampu menyesuaikan perilakunya dengan arus

lingkungan dan persyaratan untuk menangani tugas yang diperlukan secara efisien. Dalam melakukan ini, sistem harus sepenuhnya menyadari *konsumsi energi*.

Kesulitan dalam mencapai semua persyaratan ke dalam satu arsitektur berasal dari trade-off yang melekat antara fleksibilitas dan konsumsi energi, dan juga antara kinerja dan konsumsi energi. Fleksibilitas memerlukan komputasi umum dan struktur komunikasi yang dapat digunakan untuk mengimplementasikan berbagai jenis algoritma. Sementara arsitektur konvensional (seperti yang digunakan pada laptop saat ini) dapat diprogram untuk melakukan hampir semua tugas komputasi, arsitektur ini mencapainya dengan biaya konsumsi energi yang tinggi. Menggunakan dekomposisi sistem pada tingkat arsitektur yang berbeda dan memanfaatkan lokalitas referensi dengan modul khusus yang dioptimalkan, banyak energi yang dapat dihemat.

### 3.3.1 Pendekatan

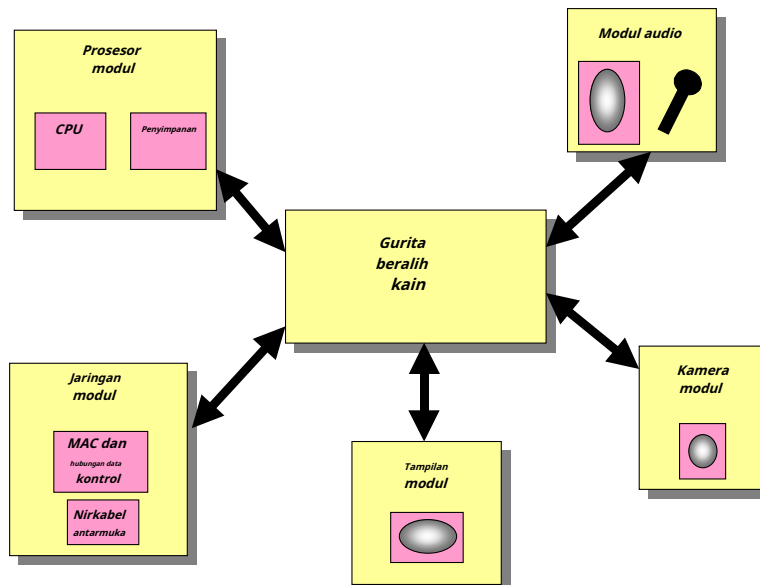
Ruang lingkup bagian ini adalah arsitektur perangkat keras, firmware, dan perangkat lunak sistem secara umum, dan masalah berikut secara khusus:

- Hilangkan sebanyak mungkin CPU sebagai komponen aktif di semua aliran data. Secara khusus kami bertujuan untuk menghilangkan partisipasi aktif CPU dalam transfer media antar komponen seperti jaringan, tampilan dan sistem audio (misalnya ketika pendamping berfungsi sebagai telepon, walk-man, TV, atau surat kabar elektronik). Tidak seperti arsitektur CPU lokal, di mana periferal I/O meningkatkan fungsionalitas prosesor inti, tujuan kami adalah merancang periferal cerdas yang mampu memproses kejadian I/O dan dapat mengelola aliran data tanpa bergantung pada prosesor terpusat.
- Hilangkan sebanyak mungkin memori sebagai stasiun perantara untuk semua transfer data antar perangkat. Energi yang diperlukan untuk mentransfer dan menyimpan data akan terbuang sia-sia jika data hanya menempati memori dalam perjalanan antara dua perangkat (misalnya jaringan dan layar atau jaringan dan audio).
- Gunakan perangkat dinamis yang dapat diprogram dan dapat disesuaikan yang mengubah aliran data masuk atau keluar, khususnya perangkat jaringan, keamanan, tampilan, dan audio. Karena dapat diprogram, mereka dapat menangani standar pengkodean data dan protokol komunikasi yang berbeda secara mandiri. Ini memiliki dua efek. Pertama, perangkat dapat dirancang untuk berkomunikasi secara langsung satu sama lain, alih-alih memerlukan intervensi CPU untuk mengadaptasi aliran data.
  - Perangkat tampilan akan mengkonversi antara, misalnya, data terkompresi MJPEG dan data piksel. Aplikasi multimedia bisa mendapatkan keuntungan dari kompresi sebagai sarana penghematan (pemborosan energi) bandwidth jaringan, tetapi membutuhkan platform daya rendah untuk perhitungan yang diperlukan.
  - Perangkat jaringan akan mengkonversi antara aliran byte yang digunakan secara internal dan, misalnya, aliran paket TCP/IP. Tumpukan protokol jaringan dapat diinstal pada perangkat antarmuka jaringan, atau bahkan di stasiun pangkalan, di mana mereka dapat menangani banyak fungsi komunikasi saat CPU dimatikan.
  - Protokol keamanan dapat dijalankan di lingkungan di luar kendali langsung dari

sistem operasi atau aplikasi. Perangkat lunak biasa rentan terhadap berbagai bentuk serangan (virus, trojan horse, bug).

Efek kedua adalah, untuk sejumlah besar fungsi konversi data ini (atau fungsi filter), prosesor sinyal digital (DSP), perangkat keras yang dapat diprogram di lapangan, atau perangkat keras khusus keduanya lebih cepat dan lebih hemat energi daripada CPU tujuan umum.

Untuk membatasi overhead komunikasi dan buffering yang diperlukan, granularitas tugas pada perangkat agak kasar, dan aplikasi dipartisi dalam blok besar. Kemampuan program setiap perangkat (atau *modul*) lebih halus dan dikendalikan oleh modul otonom individu. Aplikasi modul dapat dipartisi melalui berbagai sumber daya komputasi, berdasarkan granularitas aplikasinya. Arsitektur yang diusulkan *Pendamping Digital Seluler* ditunjukkan pada Gambar 1. Gambar tersebut menunjukkan sistem tipikal dengan modul Prosesor, modul Jaringan, modul Tampilan, modul Kamera, dan modul Audio, semuanya saling berhubungan oleh struktur switching (*Gurita beralih*).



Gambar 1: Arsitektur Mobile Digital Companion yang khas.

Sistem ini memiliki sejumlah premis:

- Arsitektur dengan prosesor tujuan umum yang disertai dengan serangkaian modul yang dapat diprogram heterogen, menyediakan implementasi tugas khusus yang hemat energi.
- Komunikasi antar modul didasarkan pada *koneksi*. Koneksi dikaitkan dengan QoS tertentu. Pengidentifikasi ini menyediakan mekanisme untuk mendukung protokol jaringan yang menyediakan layanan transportasi khusus data.
- Jaringan komunikasi internal yang dapat dikonfigurasi ulang mengeksploitasi lokalitas referensi dan menghilangkan salinan data yang boros. Jalur data melalui sakelar hanya mengkonsumsi energi saat data sedang ditransfer, membuat sebagian besar sakelar dimatikan hampir sepanjang waktu.
- Sebuah desain sistem yang menghindari aktivitas pemborosan: misalnya dengan menggunakan modul otonom yang dapat dimatikan secara individual dan data didorong.
- Sebuah sistem komunikasi nirkabel yang dirancang untuk konsumsi energi yang rendah dengan menggunakan antarmuka jaringan cerdas yang dapat menangani secara efisien dengan lingkungan mobile, dengan menggunakan protokol jaringan sadar energi, dan dengan menggunakan protokol MAC hemat energi yang meminimalkan konsumsi energi antarmuka jaringan [31].
- Kerangka kerja Kualitas Layanan untuk pengelolaan sumber daya terintegrasi *integrated Pendamping Digital Seluler* di mana setiap modul memiliki – khusus – manajemen daya lokal. Sistem operasi akan mengontrol status daya perangkat dalam sistem dan membagikan informasi ini dengan aplikasi dan pengguna.

Mobile Digital Companion adalah perangkat yang cukup serbaguna. Namun demikian fungsi-fungsi ini dapat disediakan oleh perangkat keras yang relatif sedikit. Semua modul dapat diprogram, tetapi dengan pengecualian modul prosesor, tidak semudah atau secara fleksibel dapat diprogram seperti CPU konvensional. Komponen modul dalam prototipe meliputi prosesor (mikro), DSP dan logika yang dapat diprogram (Field Programmable Gate Arrays (FPGA), atau Field Programmable Function Arrays (FPFA) [64]). Pada akhirnya, semua komponen ini harus diintegrasikan ke dalam satu chip VLSI besar (system-on-a-chip).

### 3.3.2 Filsafat

Pendekatan kami didasarkan pada fungsionalitas khusus dan penggunaan ekstensif teknik pengurangan energi di semua tingkat desain sistem. Kami akan menggunakan teknik ini di seluruh desain *Pendamping Digital Seluler*, termasuk tingkat teknologi, tingkat arsitektur, dan tingkat sistem. Untuk melestarikan lokalitas yang melekat dalam aplikasi atau algoritma arsitektur hirarkis-granularitas digunakan yang cocok dengan granularity komputasi untuk operasi yang diperlukan.

Dua tema utama yang dapat digunakan untuk pengurangan energi pada tingkat sistem adalah untuk *menghindari pemborosan*, dan untuk *mengeksploitasi lokalitas referensi*.

Menghindari pemborosan tampak jelas, tetapi dalam desain suatu sistem sulit untuk menghindari pemborosan pada berbagai tingkatan dalam sistem. Alasan untuk ini bukan hanya kecerobohan perancang, tetapi juga karena kompleksitas sistem dan hubungan antara

berbagai tingkatan dalam sistem. Yang dibutuhkan adalah model yang tepat di mana konsekuensi dari keputusan desain untuk bagian lain dalam sistem dapat diprediksi.

Komponen yang memberikan kontribusi signifikan terhadap total konsumsi energi suatu sistem adalah interkoneksi. Eksperimen telah menunjukkan bahwa dalam desain chip, sekitar 10 hingga 40% dari total daya dapat dihaburkan di bus, multiplexer, dan driver [47]. Jumlah ini dapat meningkat secara dramatis untuk sistem dengan banyak chip karena kapasitansi bus off-chip yang besar.

Jumlah energi yang dibutuhkan untuk pengangkutan data dapat dikurangi dengan menggunakan antarmuka memori khusus (misalnya teknologi memori Rambus [56]), atau dengan menggunakan memori on-chip (cache). Namun, seperti yang disebutkan sebelumnya, cache data dalam prosesor untuk aplikasi multimedia tidak banyak digunakan, dan mungkin menjadi hambatan untuk kinerja tinggi dan daya rendah, karena fungsi media ini biasanya melibatkan pemrosesan aliran input yang berkelanjutan [37], sehingga secara efektif mengosongkan cache data prosesor yang berguna. Properti lokalitas temporal dalam akses memori data tidak berlaku untuk lalu lintas data tersebut.

Seperti yang telah dijelaskan di Bab 2, ada dua sifat algoritma yang penting untuk mengurangi konsumsi daya interkoneksi: lokalitas dan keteraturan.

- *Lokalitas* Berkaitan dengan sejauh mana suatu sistem atau algoritma memiliki kluster operasi atau penyimpanan alami yang terisolasi dengan beberapa interkoneksi di antara mereka. Mempartisi sistem atau algoritme ke dalam kluster lokal secara spasial memastikan bahwa sebagian besar transfer data terjadi di dalam kluster dan relatif sedikit antar kluster. Lokalisasi mengurangi overhead komunikasi dalam prosesor dan memungkinkan penggunaan transistor berukuran kecil, yang menghasilkan pengurangan kapasitansi. Hasilnya adalah bus lokal lebih pendek dan lebih sering digunakan daripada bus global yang lebih panjang dan sangat kapasitif.
- *Keteraturan* dalam suatu algoritma mengacu pada kemunculan berulang dari pola komputasi. Pola umum memungkinkan desain arsitektur yang kurang kompleks dan oleh karena itu struktur interkoneksi yang lebih sederhana (bus, multiplexer, buffer) dan perangkat keras kontrol yang lebih sedikit.

Sebagian besar teknik untuk mengurangi konsumsi energi dapat diterapkan pada komputasi tujuan umum. Namun, untuk aplikasi multimedia khususnya, ada kemungkinan pengurangan konsumsi energi yang substansial karena kompleksitas komputasi yang tinggi dan mereka memiliki komputasi lokal yang teratur dan spasial. Juga, komunikasi antar modul sangat penting. Meningkatkan efisiensi energi dengan memanfaatkan lokalitas referensi dan menggunakan modul khusus aplikasi yang efisien karena itu memiliki dampak besar pada sistem seperti *Sahabat Digital Seluler*.

Lokalitas referensi dieksploitasi pada beberapa tingkatan. Filosofi utama yang digunakan adalah bahwa operasi pada data harus dilakukan di tempat yang paling hemat energi dan meminimalkan komunikasi yang diperlukan. Ini dapat dicapai dengan mencocokkan granularitas komputasi dan arsitektur. Dalam sistem kita memiliki *granularitas hierarkis* di mana kami membedakan tiga ukuran butir utama dari operasi:



- *berbutir halus* operasi dalam modul yang melakukan fungsi seperti perkalian dan penambahan,
- *berbutir sedang* operasi adalah fungsi dari modul. Fungsi-fungsi ini didedikasikan untuk fungsionalitas dasar modul, misalnya modul tampilan mendekomposisi aliran video.
- *tentu saja berbutir* operasi adalah tugas-tugas yang tidak spesifik untuk modul dan yang dapat dilakukan oleh modul CPU, atau bahkan pada server komputasi jarak jauh.

Kami menggunakan *distribusi mikro* untuk memigrasi tugas antara modul fungsional dalam *Pendamping Digital Seluler*, dan *distribusi makro* untuk memindahkan tugas antar modul di dunia luar dan pada sistem portabel. Dalam pendekatan terakhir, fungsi-fungsi tertentu dimigrasikan dari sistem portabel ke server jauh yang memiliki banyak sumber daya energi. Server jarak jauh menangani fungsi-fungsi yang tidak dapat ditangani secara efisien pada mesin portabel. Contoh khas dari distribusi makro dapat digunakan untuk penanganan protokol jaringan. Beberapa peneliti telah menunjukkan bahwa beberapa protokol jaringan berkinerja buruk melalui saluran nirkabel [6]. Solusinya adalah dengan membagi koneksi dalam koneksi nirkabel terpisah antara seluler dan stasiun (basis) di jaringan tetap, dan koneksi yang berbeda melalui jaringan kabel. Stasiun pangkalan dapat melakukan bagian dari tumpukan protokol jaringan sebagai pengganti ponsel, dan menggunakan protokol khusus dan hemat energi melalui saluran nirkabel. Dalam sistem seperti itu juga lebih sederhana dan efisien untuk mengadaptasi protokol untuk lingkungan spesifik yang digunakannya. Misalnya, modul jaringan dari *Pendamping Digital Seluler* mampu mengadaptasi kontrol kesalahannya, kontrol alirannya, dan kebijakan penjadwalannya dengan lingkungan saat ini dan persyaratan sistem dan aplikasi [31].

Prinsip-prinsip ini telah mengarah pada arsitektur kami yang mampu menangani aliran media secara efisien.

### 3.3.3 Memori-sentris versus koneksi-sentris

Arsitektur sistem dari *Pendamping Digital Seluler* adalah koneksi (atau media) sentris, yang berarti bahwa jenis media lalu lintas mendorong aliran data dalam sistem menggunakan koneksi. Misalnya, lalu lintas video dari antarmuka jaringan ditransfer langsung ke layar, tanpa gangguan dari CPU. Ini berbeda dengan arsitektur memory-centric (atau CPU-centric) yang berpusat di sekitar prosesor tujuan umum yang mengontrol aliran media di komputer menggunakan pengalaman memori.

#### *Berpusat pada memori*

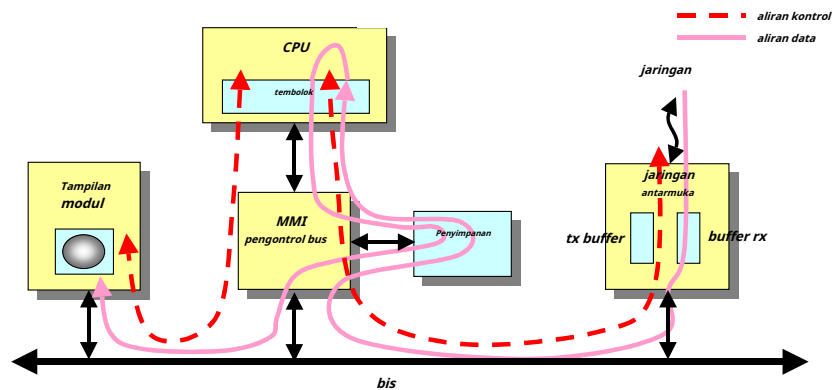
Protokol jaringan berkinerja tinggi modern mengharuskan semua akses jaringan ditangani oleh sistem operasi, yang menambahkan overhead yang signifikan ke jalur transmisi (biasanya panggilan sistem dan salinan data) dan jalur penerimaan (biasanya interupsi, panggilan sistem, dan salinan data). Biaya komunikasi dapat dipecah dalam biaya per paket dan per byte. Biaya per-paket dapat dioptimalkan, dan untuk paket besar, overhead ini diamortisasi untuk banyak data. Namun, biaya operasi per-byte seperti:

karena penyalinan data dan checksumming tidak dikurangi dengan meningkatkan ukuran paket. Pertama-tama mari kita lihat jalur pemrosesan tipikal yang terjadi pada bit informasi di komputer jaringan multimedia. Biasanya, bit informasi dihasilkan pada modul melalui sensor seperti kamera. Pemrosesan seperti pengkodean dan kompresi dapat dilakukan oleh codec pada tahap ini. Aliran kontrol (dan mungkin juga bit informasi) melewati lapisan middleware / sistem operasi. Bit kemudian akan diproses oleh aplikasi untuk transmisi melalui jaringan. Bit-bit tersebut kemudian dikirim oleh aplikasi, sekali lagi melalui middleware / sistem operasi, ke tumpukan protokol jaringan yang terdiri dari protokol transport, network, link and medium access (MAC), dan lapisan fisik. Fungsi khas dalam tumpukan protokol termasuk perutean, kontrol kemacetan, kontrol kesalahan, reservasi sumber daya, penjadwalan, dll. Bit akhirnya dikirim melalui jaringan ke node lain di mana mereka melintasi jalur sebaliknya. Perhatikan bagaimana, alih-alih fungsi aritmatika seperti penambahan dan perkalian, kepentingan utama dalam sistem adalah pemrosesan untuk protokol.

Untuk mengatasi masalah kinerja ini, beberapa *arsitektur komunikasi tingkat pengguna* telah dikembangkan yang menghapus sistem operasi dari jalur komunikasi kritis [10] dan untuk meminimalkan berapa kali data benar-benar disentuh oleh CPU host pada jalurnya melalui sistem. Skenario yang ideal adalah arsitektur salinan tunggal di mana data disalin tepat satu kali. Pengukuran tumpukan protokol salinan tunggal di Universitas Carnegie Mellon menunjukkan bahwa untuk pembacaan dan penulisan besar, jalur salinan tunggal melalui tumpukan lima hingga tujuh kali lebih efisien daripada implementasi tradisional untuk penulisan besar [67].

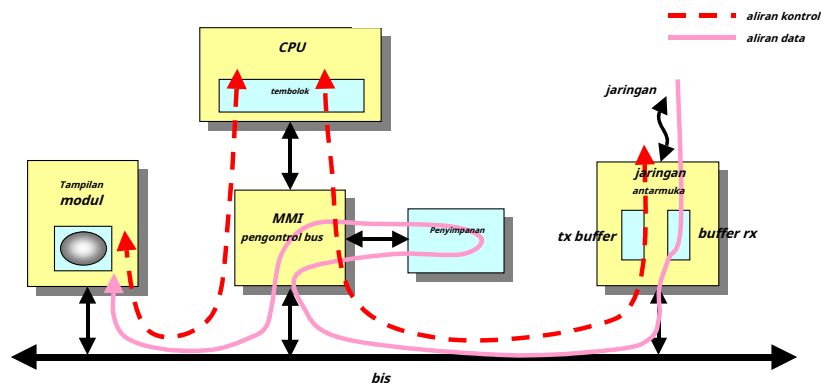
Ada beberapa cara untuk membangun protokol komunikasi yang fleksibel dan berkinerja tinggi. Teknik desain protokol lanjutan meliputi: *pembingkai tingkat aplikasi*, di mana buffering protokol sepenuhnya terintegrasi dengan pemrosesan khusus aplikasi, dan *pemrosesan lapisan terintegrasi*, di mana banyak lapisan protokol diciutkan menjadi jalur kode monolitik yang sangat efisien. Mengintegrasikan manajemen buffer antara aplikasi dan antarmuka jaringan penting dalam menghilangkan salinan data dan mengurangi alokasi dan de-alokasi. Integrasi ini, bagaimanapun, menimbulkan kompleksitas tambahan karena mekanisme dan perlindungan pengalamatan virtual [22] dan mungkin memerlukan sejumlah besar memori [67].

Itu *arsitektur tradisional* dari ponsel, ditunjukkan pada Gambar 2, dipusatkan di sekitar prosesor tujuan umum dengan memori lokal dan bus yang menghubungkan periferal ke CPU. Panah panjang pada gambar menunjukkan aliran data penting melalui sistem ketika data tiba dari jaringan, ditransfer melalui buffer penerima pada antarmuka jaringan, disalin ke memori 'utama', dan kemudian diproses oleh aplikasi. Setelah data diproses oleh aplikasi, data akan melintasi melalui memori 'utama', melalui bus, ke perangkat output (yaitu pada gambar modul tampilan). Secara umum transfer bus tambahan antara CPU dan memori diperkenalkan saat melintasi beberapa lapisan protokol (misalnya untuk konversi data paket seperti Ethernet ke IP, dan selanjutnya IP ke TCP). Sebagian besar waktu sistem dan anggaran daya dengan demikian dikhususkan untuk transaksi bus.



**Gambar 2: Aliran data melalui arsitektur tradisional.**

Seperti dapat dilihat pada gambar, CPU diperlukan tidak hanya untuk menangani jalur kontrol, tetapi juga untuk mentransfer data antar perangkat. Pendekatan yang lebih baik adalah mentransfer sejumlah besar data melalui sistem dengan menggunakan fungsionalitas Direct Memory Access (DMA) dari antarmuka dan modul. DMA digunakan untuk transfer data antara memori utama dan buffer pada antarmuka jaringan. Gambar 3 menunjukkan data terpisah dan aliran kontrol dari arsitektur yang dioptimalkan. CPU sekarang hanya diperlukan untuk melakukan aliran kontrol antar perangkat, misalnya – seperti pada contoh sebelumnya – ke antarmuka jaringan dan ke modul tampilan. Meskipun ini sudah mengurangi tuntutan yang diberikan pada prosesor secara drastis, prosesor masih harus aktif selama transaksi data untuk melakukan aliran kontrol.



**Gambar 3: Data terpisah dan aliran kontrol arsitektur tradisional.**

DMA dapat memperkenalkan beberapa kelemahan. Checksumming perlu dilakukan saat menyalin data, yaitu checksumming perlu dilakukan di perangkat keras. Pada workstation, penggunaan DMA untuk mentransfer data antara buffer jaringan dan memori utama menjadi lebih rumit dengan adanya cache dan memori virtual. DMA dapat membuat

inkonsistensi antara cache dan memori utama. Host dapat menghindari masalah ini dengan mem-flush data ke memori sebelum mentransfernya menggunakan DMA saat mengirim dan membatalkan data dalam cache sebelum DMA saat menerima. Halaman pengguna harus disambungkan ke dalam memori untuk memastikan bahwa halaman tersebut tidak dikeluarkan saat DMA sedang berlangsung. Karena overhead tambahan ini, terkadang mungkin lebih efisien menggunakan CPU untuk menyalin paket antara ruang pengguna dan sistem. Lebih lanjut, dapat diinginkan untuk mengubah aliran data online, seperti dekompresi aliran audio atau video MPEG. Dalam kasus seperti itu, CPU umumnya perlu melakukan konversi.

Dalam beberapa kasus khusus dimungkinkan untuk meneruskan data secara langsung dari sumber ke tujuan. Ini misalnya dapat diterapkan untuk menampilkan data grafik di layar jika memori tampilan dapat diakses oleh sumbernya. Dalam hal ini data hanya perlu melakukan perjalanan sekali melintasi interkoneksi antara sumber dan sink.

Perhatikan bahwa bahkan dalam kasus yang dioptimalkan ini, masih ada satu atau dua salinan data yang diperlukan melalui bus bersama. Bus adalah sumber disipasi daya yang signifikan karena aktivitas switching yang tinggi dan kapasitansi yang besar. Sistem modern biasanya dicirikan oleh bus yang lebar dan berkecepatan tinggi, yang berarti bahwa faktor kapasitansi dan frekuensi dari disipasi daya mendominasi.

Arsitektur shared-bus konvensional yang berpusat pada memori membutuhkan traversal aliran multimedia yang sering melalui bus pusat yang sangat kapasitif dan melalui lapisan-lapisan perangkat lunak sistem operasi untuk operasi yang paling sederhana seperti multiplexing/demultiplexing dan sinkronisasi antar aliran. Memang, pengukuran dengan prototipe terminal multimedia nirkabel di University of California di Los Angeles (UCLA) dengan arsitektur berbasis PC yang tertanam menunjukkan bahwa sejumlah besar waktu dan daya masuk ke memori dan transaksi I/O di bus bersama [41].

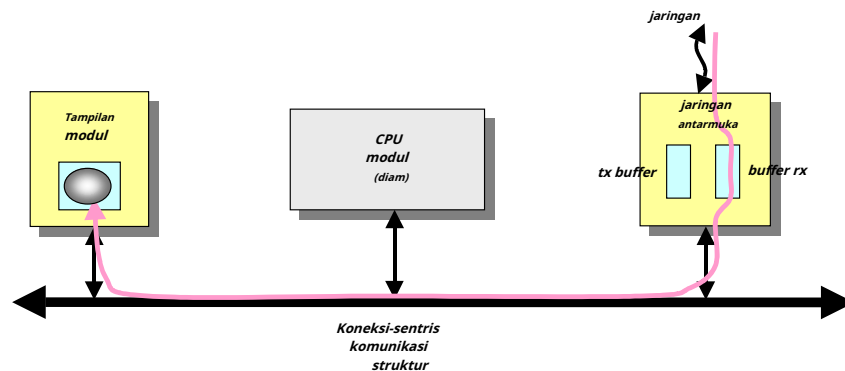
Tren yang sama dapat diamati dalam desain mikroprosesor. *Komputer* menghasilkan masalah khusus pada "arsitektur Miliar transistor" yang membahas masalah dan tren yang akan mempengaruhi desain prosesor masa depan, dan beberapa arsitektur dan implementasi mikroprosesor yang diusulkan [12]. Sebagian besar desain ini fokus pada domain desktop dan server. Mayoritas menggunakan 50 hingga 90 persen dari anggaran transistor mereka pada cache, yang membantu mengurangi latensi tinggi dan bandwidth rendah dari memori eksternal. Dengan kata lain, visi konvensional komputer masa depan menghabiskan sebagian besar anggaran transistor miliaran pada salinan data lokal yang berlebihan yang biasanya ditemukan di tempat lain dalam sistem [37].

Sistem saat ini berdasarkan arsitektur bus bersama mampu memberikan kinerja yang diperlukan untuk berbagai aplikasi multimedia tidak hanya dengan menggunakan kemajuan pesat dalam teknologi, tetapi juga dengan desain yang cermat dan penggunaan modul antarmuka. Proses untuk mencapai hal ini membutuhkan banyak usaha baik dari perancang perangkat keras antarmuka I/O dan perancang sistem. Perancang perangkat keras harus sangat hati-hati menghubungkan perangkat agar sesuai dengan kebutuhan aplikasi, menyesuaikan sirkuit sehingga kabel sependek mungkin dan semua sinyal sampai dari titik asalnya ke tempat yang tepat pada waktu yang tepat. Kemudian, perancang perangkat lunak harus hati-hati menentukan secara rinci apa yang mampu perangkat sebelum merancang sistem multimedia [11]. Ada banyak masalah perangkat lunak yang dapat memengaruhi kinerja I/O keseluruhan sistem. Ketika, setelah banyak penyesuaian, akhirnya sistem berjalan dengan memuaskan, masalah kinerja dapat

timbul dengan mudah ketika konfigurasi (perangkat keras atau lunak) sistem (sedikit) diubah, sistem operasi diperbarui, atau pengguna menggunakan aplikasi baru. Alasan untuk masalah ini sering disebabkan oleh interkoneksi dan protokol interkoneksi. Karena bus bersama tidak dapat memberikan jaminan QoS, satu perangkat atau aplikasi dapat mengurangi throughput yang tersedia untuk semua perangkat.

### Koneksi-sentris

Dengan merancang arsitektur yang menggerakkan kekuatan pemrosesan lebih dekat ke aliran data, dimungkinkan untuk melewati CPU sama sekali. Pendekatan ini sangat cocok untuk data media kontinu (misalnya audio, video, dll.), di mana pemrosesan sebenarnya bersifat sangat khusus (misalnya pemrosesan sinyal, kompresi, enkripsi, dll.) dan perlu dilakukan secara nyata. -waktu. CPU dengan demikian dipindahkan dari jalur data aliran data, meskipun masih berpartisipasi dalam aliran kontrol. Peran CPU dikurangi menjadi pengontrol yang menginisialisasi sistem dan menangani pemrosesan protokol kompleks yang paling mudah diimplementasikan dalam perangkat lunak pada prosesor tujuan umum.



**Gambar 4: Aliran data dalam arsitektur connection-centric.**

Berbeda dengan sistem memori-sentris, sistem koneksi-sentris didekomposisi dari koprosesor khusus aplikasi yang berkomunikasi menggunakan koneksi. Sistem operasi memainkan peran penting dalam arsitektur ini, karena bertanggung jawab untuk mengatur koneksi antar modul. CPU dan sistem operasi tidak berpartisipasi dalam aliran kontrol selama transaksi. Struktur interkoneksi tidak didasarkan pada bus yang menggunakan alamat, tetapi didasarkan pada struktur komunikasi berorientasi koneksi (seperti *Gurita* switching fabric seperti yang digambarkan pada Gambar 1). Dalam sistem seperti itu lalu lintas data berkurang, terutama karena salinan data yang tidak perlu dihapus. Misalnya, dalam sistem di mana aliran data video akan ditampilkan di layar, data dapat disalin langsung ke modul tampilan tanpa melalui prosesor utama. Modul tampilan mungkin mengubah aliran data dan meneruskannya ke memori layarnya. Hasilnya adalah bahwa alih-alih delapan transaksi data (dua di antaranya melalui bus besar) yang diperlukan dalam arsitektur tradisional, dalam arsitektur connection-centric hanya diperlukan dua transaksi lokal (antarmuka jaringan untuk beralih, beralih untuk menampilkan).

modul). Modul CPU terutama digunakan untuk memulai koneksi, dan dapat mengganggu selama transaksi.

Dalam arsitektur connection-centric, setiap koneksi dapat dikaitkan dengan QoS tertentu menggunakan pengenalan koneksi. Pengidentifikasi ini menyediakan mekanisme untuk mendukung protokol ringan yang menyediakan layanan transportasi khusus data yang terkait dengan QoS tertentu. Desain hati-hati dan proses fine-tuning yang dibutuhkan oleh perancang sistem untuk arsitektur memori sentris, tidak diperlukan jika QoS dapat dijamin di seluruh sistem.

### 3.3.4 Modul khusus domain aplikasi

Gambar 1 memberikan gambaran skematis dari *Pendamping Digital Seluler* Arsitektur. Di dalamnya, kami membedakan sakelar yang dikelilingi oleh beberapa modul yang masing-masing dioptimalkan untuk domain aplikasi tertentu. Selain itu, modul-modul ini dapat dikonfigurasi ulang, sehingga dapat disesuaikan saat diperlukan.

Secara umum selalu ada modul dengan prosesor utama tujuan umum yang melakukan operasi tipe kontrol. Modul lain dapat berupa perangkat seperti pengontrol tampilan, antarmuka jaringan, dan penyimpanan yang stabil. Arsitektur terdiri dari banyak perangkat yang biasanya ditemukan di workstation multimedia, tetapi karena target kami adalah komputer portabel, perangkat ini umumnya tidak memiliki kinerja dan ukuran rekan kerja mereka. Target utama kami adalah memiliki *sistem-on-a-chip*, di mana semua fungsionalitas sistem terintegrasi dalam satu chip.

Perhatikan bahwa perangkat kami bukan hanya perangkat I/O khusus dalam pengertian tradisional. Kami lebih suka memanggil perangkat *modul*, atau *subsistem I/O*, untuk menekankan fakta bahwa mereka menyediakan lebih banyak fungsionalitas daripada perangkat sederhana. Modul berdiferensiasi menjadi perangkat I/O dalam berbagai cara. Pertama, setiap modul merupakan subsistem otonom yang dapat beroperasi tanpa intervensi dari CPU utama. Kedua, ia memiliki prosesor kontrol yang melakukan beragam operasi, termasuk manajemen koneksi dan manajemen energi. Akhirnya, sebagian besar modul dapat menyesuaikan perilakunya secara mandiri dengan 'keinginan' klien atau aplikasi, dan mencoba beroperasi dengan cara yang paling efisien.

**Keuntungan** – Seringkali lebih masuk akal untuk mengimplementasikan lapisan adaptasi spesifik perangkat (atau media) di dalam modul, daripada membutuhkan antarmuka jaringan atau CPU utama untuk mengimplementasikan sejumlah besar lapisan adaptasi yang berbeda [7]. Alasan utama untuk ini adalah:

- *Pemrosesan yang efisien* – Modul mampu melakukan tugas spesifik perangkat atau aplikasi secara efisien. Misalnya dapat mendekompresi aliran video, tepat sebelum ditampilkan di layar (ini adalah contoh khas dari prinsip lokalitas referensi). Modul khusus dapat dioptimalkan untuk menjalankan tugas tertentu secara efisien, dengan overhead energi yang minimal. Alih-alih mengeksekusi semua komputasi dalam jalur data tujuan umum, seperti yang biasa dilakukan dalam arsitektur terprogram konvensional, tugas-tugas intensif energi dan komputasi dieksekusi pada modul yang dioptimalkan. Misalnya, bahkan ketika koprosesor khusus aplikasi mengkonsumsi lebih banyak daya daripada prosesor, itu dapat menyelesaikan tugas yang sama dalam waktu yang jauh lebih sedikit, menghasilkan penghematan energi bersih. Prosesor dapat, misalnya, menjadi

diturunkan dengan tugas-tugas seperti decoding JPEG dan MP3, enkripsi, dan beberapa penanganan protokol jaringan. Perancang sistem dapat menerapkan koprocesor khusus aplikasi (misalnya perangkat keras khusus) untuk tugas-tugas yang dapat ditangani modul secara efisien, dan menggunakan prosesor untuk bagian-bagian algoritme yang perangkat kerasnya tidak sesuai (misalnya inisialisasi).

- *Hilangkan salinan data yang tidak berguna* – Ketika data mengalir langsung di antara modul yang perlu memprosesnya, salinan data yang tidak perlu dapat dihilangkan. Menghilangkan salinan data yang tidak perlu dapat mengurangi lalu lintas di bus. Misalnya, dalam sistem di mana aliran data video akan ditampilkan di layar, data dapat disalin langsung dari jaringan ke memori layar, tanpa melalui prosesor utama.
- *Bebaskan CPU tujuan umum* – Dalam sistem connection-centric, data dapat mengalir antar modul tanpa melibatkan CPU utama dan tanpa menggunakan siklus prosesor. CPU utama juga dibebaskan dari keharusan melayani interupsi dan melakukan sakelar konteks setiap kali data baru tiba, atau perlu dikirim dari perangkat lokal. Alih-alih memiliki satu sistem pusat yang perlu mengontrol dan memproses operasi berbutir halus, sistem kontrol terdistribusi kurang kompleks. Lebih mudah untuk memberikan dukungan real-time untuk perangkat jika prosesor khusus mengontrolnya.
- *Adaptasi yang mudah* – Modul dapat dengan mudah menyesuaikan perilakunya. Jika sebuah modul mengadaptasi perilakunya, ia mampu bereaksi terhadap perubahan lingkungan, baik yang dipaksakan oleh pengguna (ketika memulai aplikasi baru atau yang berbeda) atau oleh perubahan sumber daya (misalnya ketika modul jaringan melihat perubahan dalam nirkabel kondisi saluran).
- *Manajemen energi yang memadai* – Setiap modul berisi pengetahuan khusus tentang pola penggunaan dan persyaratan khusus untuk perangkat. Oleh karena itu, setiap modul memiliki tanggung jawab sendiri dan memiliki otonomi dalam memutuskan bagaimana mengelola status operasinya untuk meminimalkan konsumsi energinya tanpa mengurangi kualitas layanannya. Ini berbeda dengan sistem saat ini di mana CPU utama dapat mengontrol status daya perangkat yang terhubung. Kami telah memberikan modul tanggung jawab mereka sendiri dalam memutuskan bagaimana mengelola sumber daya mereka. Modul individu dikendalikan oleh kebijakan energi global yang membuat keputusan tingkat tinggi tentang keadaan seluruh sistem.
- *Fleksibel dan mudah beradaptasi* – Karena modul dapat diprogram, mereka dapat menawarkan fleksibilitas untuk menyediakan dukungan untuk berbagai standar yang mungkin perlu digunakan oleh Pendamping (misalnya skema penyandian dan enkripsi yang berbeda), dan kemampuan beradaptasi untuk mengadaptasi mekanisme, algoritme, dan tekniknya ke berbagai kondisi pengoperasian. Dari semua modul yang dapat diprogram, prosesor serba guna hanyalah yang paling fleksibel. Prosesor akan digunakan untuk semua tugas yang tidak mampu dilakukan oleh modul khusus aplikasi, atau ketika implementasinya tidak efisien. Prosesor tujuan umum akan melakukan semua perhitungan yang terlalu rumit atau akan membutuhkan terlalu banyak area jika diimplementasikan dengan modul perangkat keras. Prosesor tujuan umum dengan demikian juga dapat dilihat sebagai domain aplikasi

modul spesifik: domainnya mencakup semua area yang tidak tercakup oleh modul lain.

Tentu saja ada juga beberapa kelemahan. Sebagian besar trade-off yang terlibat telah dibahas dalam Bagian 3.2.6. Kerugian yang paling terlihat adalah memiliki modul khusus domain aplikasi membutuhkan lebih banyak perangkat keras. Alih-alih memproses semua tugas pada satu prosesor tujuan umum, tugas-tugas ini didistribusikan ke beberapa modul. Namun, diharapkan kemajuan teknologi memberikan kemungkinan yang cukup untuk memanfaatkan peningkatan area chip yang efektif dan menyediakan lebih banyak fungsionalitas sambil menjaga konsumsi energi tetap rendah.

### *Contoh modul*

Modul bisa sangat beragam dan memiliki karakteristik dan persyaratan yang berbeda. Modul harus mampu menangani koneksi dengan modul lain. Mereka biasanya berisi beberapa kecerdasan untuk pengaturan koneksi dan manajemen energi. Contoh umum modul adalah:

- *CPU-modul* – Ini adalah modul yang dapat melakukan aplikasi tujuan umum, dan menyediakan berbagai layanan. Salah satu tugas penting dari modul ini adalah bertanggung jawab untuk manajemen koneksi. Jika koneksi harus dibuat yang memerlukan beberapa jaminan kualitas layanan, maka modul CPU bernegosiasi dengan modul yang mengambil bagian dalam koneksi. Modul-CPU adalah tempat sentral di mana semua koneksi terkait QoS dikelola.
- *Modul jaringan* – Modul ini menyediakan akses ke dan dari jaringan eksternal (nirkabel). Dalam penelitian kami, kami telah mengembangkan modul jaringan hemat energi yang dapat menangani lalu lintas multimedia. Modul ini adalah topik utama Bab 5. Modul dapat mengarahkan lalu lintas menurut VCI sel ATM langsung ke modul tujuan. Stasiun pangkalan memainkan peran penting, karena menangani sebagian besar lapisan tumpukan protokol jaringan dalam komunikasi melalui jaringan kabel sebagai pengganti seluler. Dengan cara ini kekhasan saluran nirkabel dipisahkan dari lapisan protokol jaringan, dan dapat memberikan komunikasi yang lebih efisien [6]. Stasiun pangkalan juga dapat bertindak sebagai server proxy yang menyesuaikan data ke format yang dapat digunakan ponsel dengan cara yang efisien. Misalnya, aliran video dari jaringan tetap dikonversi di stasiun pangkalan ke format yang dapat dengan mudah diinterpretasikan oleh layar [74]. Komunikasi antara modul jaringan dan base station menggunakan protokol MAC yang hemat energi (EzMaC) yang mampu memberikan jaminan QoS melalui saluran nirkabel [31]. Arsitektur modul jaringan menggunakan kontrol kesalahan dinamis yang disesuaikan dengan QoS dan jenis lalu lintas koneksi, dan memiliki antrian koneksi khusus dan kontrol aliran untuk setiap koneksi.

Perhatikan bahwa sementara kami mengusulkan untuk menghilangkan ketergantungan pada tumpukan protokol berbasis perangkat lunak dari ponsel, tidak ada alasan untuk secara dogmatis menghalangi keterlibatan prosesor tujuan umum. Misalnya, untuk tujuan penelitian, diinginkan memiliki kemampuan untuk mengembangkan dan menguji algoritma dan protokol baru. Dalam arti, tumpukan tetap merupakan tumpukan perangkat lunak: perangkat dapat diprogram. Kerugian tradisional dari tumpukan perangkat keras (tidak fleksibel, biaya beton) tidak berlaku di sini.



- *Modul tampilan* – Pada komputer portabel, tampilan umumnya akan kecil dan memiliki resolusi rendah. Pendekatan terbaik adalah dengan mengadaptasi data yang ditransmisikan ke modul tampilan sehingga dapat dengan mudah menafsirkan data dan menampilkannya langsung di layar. Jika modul tampilan berisi mesin dekompresi, maka energi dan bandwidth dapat dihemat karena tidak ada data yang tidak terkompresi yang harus melintasi interkoneksi, dan mungkin juga melintasi jaringan nirkabel.
- *Modul komputasi yang dapat dikonfigurasi ulang* – Modul ini pada dasarnya adalah perangkat yang mampu menangani berbagai macam layanan. Modul berisi logika yang dapat dikonfigurasi ulang yang dapat (kembali) dikonfigurasi secara dinamis sesuai kebutuhan. Contoh tipikal di mana modul semacam itu dapat memberikan fleksibilitas yang besar adalah ketika digunakan sebagai mesin enkripsi/dekripsi. Data yang ditujukan atau berasal dari jaringan tetap dapat dibuat untuk 'melewati' perangkat enkripsi dalam perjalanan ke modul tujuan.

Arsitekturnya bersifat modular dan dapat diperluas dengan modul-modul yang memiliki fungsi berbeda.

### **3.3.5 Jaringan interkoneksi**

Jaringan interkoneksi adalah komponen kunci untuk memberikan fleksibilitas dalam sistem yang dapat dikonfigurasi ulang [76]. Semua modul dalam sistem berkomunikasi melalui jaringan komunikasi yang dapat dikonfigurasi ulang yang diatur sebagai sakelar. Secara konseptual, arsitektur ini analog dengan switch paket self-routing. Implementasi yang tepat dari interkoneksi bukanlah masalah vital dalam arsitektur *Sahabat Digital Seluler*.

Sama seperti ring, palang dan bus semuanya telah digunakan di switch ATM [7], sehingga mereka dapat digunakan di Companion (walaupun mereka berbeda dalam kompleksitas dan konsumsi energi). Ini adalah pendekatan berorientasi koneksi menggunakan sel berukuran tetap dan multiplexing asinkron yang merupakan faktor kunci. Seperti pada jaringan switching, penggunaan topologi multi-jalur akan memungkinkan aliran data paralel antara pasangan modul yang berbeda dan dengan demikian akan meningkatkan kinerja.

Sakelar menghubungkan modul dan menyediakan jalur yang andal untuk komunikasi antar modul. Pengalamatan didasarkan pada koneksi daripada alamat memori. Ini tidak hanya menghilangkan kebutuhan untuk mentransfer sejumlah besar bit alamat per akses, tetapi juga memberi sistem kemungkinan untuk mengontrol QoS tugas ke infrastruktur komunikasi. Ini merupakan persyaratan penting karena dalam arsitektur QoS semua komponen sistem, perangkat keras serta perangkat lunak, harus dicakup ujung ke ujung sepanjang jalan dari sumber ke tujuan.

QoS adalah tema umum dalam penelitian kami. Ini digunakan sebagai sarana untuk menangani perilaku dinamis saluran komunikasi, dan untuk dapat menyediakan berbagai aliran di komputer seluler multimedia dengan kualitas yang memuaskan dengan biaya terendah. Seluruh sistem didasarkan pada koneksi antar modul. Setiap koneksi dikaitkan dengan QoS tertentu. Aplikasi harus menunjukkan QoS yang diharapkan dari sistem, dan kemampuan serta kemauannya untuk mengubah persyaratan QoS ini. Semua modul berkomunikasi hanya menggunakan koneksi ini. Modul jaringan menggunakan mekanisme yang sama untuk berkomunikasi melalui saluran nirkabel dengan stasiun pangkalan yang terhubung ke kabel

lingkungan Hidup. Jika lingkungan kabel juga menyediakan mekanisme yang mampu menangani QoS (seperti jaringan ATM), maka pada prinsipnya kita dapat membangun sebuah *QoS ujung ke ujung*. Dengan cara ini kita dapat membuat koneksi dari aplikasi di jaringan kabel, melalui jaringan nirkabel, langsung ke tujuan di mana data akan diproses. Di jalur dari sumber ke tujuan, kami menerapkan tuntutan QoS ke berbagai sumber daya yang terlibat termasuk kontrol kesalahan pada saluran nirkabel, lapisan tautan komunikasi, dan protokol akses media. Tujuannya adalah untuk memenuhi persyaratan QoS koneksi pada konsumsi energi terendah. Konsekuensi untuk sistem operasi digambarkan dalam Bagian 3.3.8.

Dalam infrastruktur kami semua koneksi diidentifikasi dengan pengenalan koneksi yang digunakan untuk mengidentifikasi jenis data, dan alamat tujuan modul. Pengidentifikasi ini menyediakan mekanisme untuk mendukung protokol ringan yang menyediakan layanan transportasi khusus data yang terkait dengan QoS tertentu.

Arsitektur di mana interkoneksi packet-switched umum digunakan untuk menghubungkan prosesor, memori, dan perangkat secara luas dikenal sebagai 'jaringan area meja' (DAN). Leslie, McAuley dan Tennenhouse pertama kali memperkenalkan konsep DAN [40]. Kami telah mengadopsi konsep ini, dan akan menunjukkan bahwa arsitektur seperti itu juga cocok untuk komputer portabel berdaya rendah. Oleh karena itu, arsitektur kami memiliki beberapa kesamaan dengan misalnya: *Jaringan Area Meja* dari Cambridge [32], *VuNet* dari MIT [34] dan *APIC* arsitektur dari University of Washington [18]. Namun, motivasi utama mereka adalah kinerja dan interoperabilitas antara (ATM [54]) jaringan dan perangkat. Motivasi utama kami adalah mengurangi konsumsi energi dan bukan hanya kinerja.

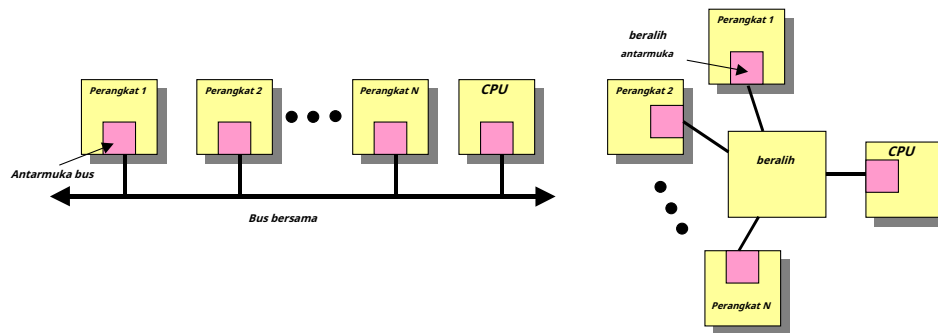
Pada akhirnya, arsitektur harus diimplementasikan hanya dalam satu chip. Oleh karena itu, kami tidak akan menyebut arsitektur sebagai *Meja tulis*-jaringan area, tetapi hanya sebagai *chip*-jaringan daerah. Dalam *Ular derik* Switch ATM kami telah menunjukkan bahwa cukup layak untuk membangun sistem switching ATM hemat biaya yang memenuhi persyaratan multimedia [27][61]. Beberapa ide yang diperkenalkan dalam arsitektur sakelar Rattlesnake telah digunakan dalam desain *Gurita* sakelar yang menjadi subjek Bab 4.

Setiap perangkat dilengkapi dengan antarmuka ke interkoneksi ATM. Lampiran ini tidak lebih rumit dari antarmuka bus yang setara, dan bahkan lebih sederhana daripada bus kinerja tinggi yang rumit (Bab 4). Header sel ATM standar dibagi menjadi jalur virtual dan saluran virtual. Dalam *Pendamping Digital Seluler*, subdivisi ini tidak signifikan. Jika kita akan menggeneralisasi arsitektur kita sehingga menjadi jaringan ATM lengkap yang dapat berinteraksi dengan perangkat ATM yang berbeda, maka kita sampai pada arsitektur berbasis DAN yang lebih futuristik. Sementara sistem seperti itu kemungkinan akan memberikan kinerja yang lebih tinggi daripada arsitektur *Pendamping Digital Seluler*, itu tidak datang tanpa beberapa kekurangan. Di antaranya adalah konsumsi energi yang tinggi yang diperlukan untuk mengimplementasikan tumpukan jaringan ATM yang lengkap. Selain itu, biayanya akan lebih tinggi karena pertimbangan penting ketika merancang arsitektur jaringan adalah skalabilitas dan toleransi terhadap tautan dan node yang tidak berfungsi. Dalam jaringan area chip, batas kepercayaan dan kondisi operasi jauh berbeda dari jaringan area meja dan jaringan area lokal atau luas. Jaringan yang lebih besar membutuhkan perlindungan dari klien yang bermusuhan atau salah dan sejumlah besar

kekuatan pemrosesan harus dimasukkan ke dalam perangkat untuk mengelola fungsi kontrol dan keamanan. Arsitektur interkoneksi yang dirancang untuk menghubungkan komponen sistem seluler harus memenuhi persyaratan yang tidak terlalu ketat. Dalam arsitektur kami, node akhir didedikasikan untuk satu lingkungan kerja, mereka dapat dikontrol dengan lebih mudah, mereka dapat dipercaya untuk bersikap adil dan tidak melebihi alokasi sumber daya mereka.

### 3.3.6 Analisis energi

Pada bagian ini kita akan mengevaluasi dampak energi dari arsitektur connection-centric berdasarkan switch, dibandingkan dengan arsitektur memory-centric berdasarkan shared bus. Gambar 5 menunjukkan dua arsitektur. Dalam analisis ini kita hanya akan membahas efek energi dari komunikasi. Perhatikan bahwa konsumsi energi yang diperlukan untuk komunikasi hanya sebagian kecil dari total konsumsi energi suatu sistem menggunakan aplikasi multimedia biasa (yaitu  $1/10^6$ , lihat Bab 6<sup>2</sup>).



Gambar 5: Arsitektur bus bersama dan arsitektur yang diaktifkan.

Dari Bab 2 kita tahu bahwa perkiraan orde pertama dari konsumsi energi dinamis sirkuit CMOS diberikan oleh rumus:

$$P_d = C_{eff} V_2 f \quad (1)$$

dimana  $P_d$  adalah konsumsi energi dalam Watt,  $C_{eff}$  adalah kapasitansi switching efektif dalam Farad,  $V$  adalah tegangan suplai dalam Volt, dan  $f$  adalah frekuensi operasi dalam Hertz.  $C_{eff}$  menggabungkan dua faktor, kapasitansi  $C$  sedang diisi/dikosongkan, dan *pembobotan aktivitas*, yang merupakan peluang terjadinya transisi.

$$C_{eff} = \alpha C \quad (2)$$

Sebagian besar parameter dalam persamaan ini dipengaruhi oleh pilihan arsitektur. Dalam analisis kami, kami membuat beberapa asumsi.

<sup>2</sup> Dalam Bab 6 kita akan mengevaluasi konsumsi energi sistem bergerak untuk aplikasi tipikal.

1. Elemen yang akan kami pertimbangkan untuk berkontribusi pada konsumsi energi di keduanya arsitektur adalah konsumsi energi antarmuka koneksi ( $P_{dua}$  untuk antarmuka bus dan  $P_{si}$  untuk antarmuka sakelar) dan konsumsi energi yang disebabkan oleh kabel ( $P_{bw}$  untuk setiap antarmuka ke bus, dan  $P_{sw}$  untuk setiap antarmuka ke sakelar). Konsumsi energi yang disebabkan oleh kain switching adalah  $P_{beralih}$  per antarmuka perangkat.
2. Arsitektur bus bersama adalah memori sentris, yang menyiratkan bahwa untuk mentransfer paket alamat harus disediakan. Kami akan berasumsi bahwa pengontrol antarmuka bus menyediakan mode burst, sehingga hanya satu alamat yang diperlukan untuk seluruh paket. Arsitektur yang diaktifkan adalah koneksi sentris, yang menyiratkan bahwa pengenalan koneksi harus disediakan per paket. Dalam analisis kami, kami akan mengabaikan perbedaan yang disebabkan oleh skema pengalamatan ini.
3. Kita asumsikan bahwa *tidak* perangkat (modul) memiliki koneksi setengah dupleks. Ini menyiratkan bahwa throughput agregat dengan demikian maksimal  $T/2$  kali throughput dari satu koneksi. Misalnya, dalam arsitektur Octopus ada empat koneksi simultan ketika sumber dan tujuan koneksi ini terputus. Namun, karena koneksi dalam suatu sistem tidak selalu terputus-putus, kita akan mengasumsikan throughput agregat rata-rata sebesar  $T/4$ .
4. Kami berasumsi dalam analisis kami bahwa kompleksitas logika antarmuka bus sama dengan antarmuka sakelar. Perhatikan bahwa, bagaimanapun, lebih mungkin bahwa antarmuka bus akan lebih kompleks. Ini karena antarmuka bus harus fleksibel (karena harus mampu menangani berbagai macam perangkat) dan memiliki kinerja tinggi (karena bus digunakan bersama, misalnya, perlu menerapkan mode transfer data burst untuk mencapai data yang diperlukan. tarif). Karena antarmuka bus harus beroperasi pada frekuensi yang lebih tinggi, konsumsi energi antarmuka bus akan lebih tinggi. Menggunakan Asumsi 3 kita akan mengasumsikan bahwa konsumsi energi dari antarmuka bus adalah  $T/4$  kali konsumsi energi antarmuka sakelar, dengan demikian  $P_{bi} = T/4 \cdot P_{si}$ .
5. Dari Persamaan (1) kita tahu bahwa ada ketergantungan linier kapasitansi pada konsumsi energi. beban kapasitif  $C_{di\ luar}$  dari gerbang logika CMOS G terutama terdiri dari a) kapasitansi gerbang  $C_{untuk}$  transistor di gerbang yang digerakkan oleh G, b) kapasitansi  $C_w$  dari kabel yang menghubungkan gerbang dan c) kapasitansi parasit  $C_p$  transistor di gerbang G [8]. Dalam simbol:

$$C_{keluar} = C_{untuk} + C_w + C_p \quad (3)$$

Kapasitansi fan-out tergantung pada jumlah gerbang logika yang digerakkan oleh G dan dimensi transistornya. Dalam arsitektur bus, jumlah ini setara dengan jumlah perangkat yang terhubung (termasuk CPU)  $N$  Ukuran transistor yang dibutuhkan untuk menggerakkan bus berkecepatan tinggi juga lebih besar dari transistor yang hanya perlu dikendarai satu gerbang. Sangat sulit untuk memperkirakan  $C_w$  akurat karena tergantung pada topologi dan perutean kabel dan ukurannya. Kopling antar kabel adalah menjadi faktor yang paling penting untuk kapasitansi kabel. Kapasitansi kabel mendominasi  $C_{di\ luar}$  untuk bus. kapasitansi parasit  $C_p$  mungkin adalah

komponen yang paling sedikit menimbulkan kekhawatiran, karena relatif kecil dibandingkan dengan dua kontribusi lainnya.

Mempertimbangkan pertimbangan ini, kami akan mengasumsikan dalam analisis kami bahwa: kapasitas per perangkat pada bus bersama ( $C_{bw}$ ) adalah dua kali dari kapasitas kabel pada arsitektur sakelar ( $C_{barat}$ ), jadi  $C_{bw} = 2 C_{barat}$

6. Seperti dijelaskan dalam Bagian 3.3.3, arsitektur bus bersama membutuhkan setidaknya satu, dan dalam kebanyakan situasi dua transfer data melalui bus bersama untuk aliran antara dua perangkat. Kami akan berasumsi bahwa transfer data melalui interkoneksi didasarkan pada Direct Memory Access (DMA) yang dilakukan oleh pengontrol DMA perangkat. Jumlah salinan DMA adalah  $D$ . Kompleksitas pengontrol DMA yang diperlukan untuk arsitektur bus bersama dan untuk arsitektur yang diaktifkan diasumsikan sama.
7. Kami mengabaikan dalam analisis ini konsumsi energi yang disebabkan oleh CPU yang mengendalikan aliran data. Namun, perhatikan bahwa ini dapat menjadi bagian penting dari total konsumsi energi karena dalam arsitektur bus bersama di mana CPU harus mengelola aliran data di mesin, CPU tidak dapat memasuki mode tidur untuk waktu yang wajar. Dalam arsitektur yang diaktifkan, CPU keluar dari jalur data dan dapat memasuki mode tidur setiap kali koneksi telah diatur. Kami selanjutnya mengabaikan konsumsi energi yang disebabkan oleh buffering perantara di CPU-node.
8. Penskalaan tegangan adalah cara yang efektif untuk mengurangi konsumsi energi. Arsitektur yang diaktifkan dapat beroperasi pada tegangan yang lebih rendah daripada arsitektur bus bersama karena dapat beroperasi pada kecepatan yang lebih rendah. Dalam arsitektur bus bersama, semua aliran data harus dilakukan secara berurutan menggunakan satu sumber daya bersama (bus). Untuk mencapai throughput yang dibutuhkan, bus harus cepat. Arsitektur yang diaktifkan memungkinkan beberapa aliran data paralel. Karena itu, bandwidth koneksi individu dalam arsitektur yang diaktifkan tidak harus setinggi bandwidth dalam arsitektur bus bersama. Karena bandwidth yang lebih rendah yang dibutuhkan untuk koneksi pada arsitektur yang diaktifkan, tegangan dapat dikurangi. Delay sebuah inverter CMOS dapat digambarkan dengan rumus berikut [8]:

$$T_d = \frac{C_{V_{dd}}}{saya} \frac{C_{di \text{ luar } V_{DD}}}{(W/L)(V_{DD} - V_t)^2} \quad (4)$$

dimana  $\eta$  adalah konstanta yang bergantung pada teknologi,  $W$  dan  $L$  masing-masing adalah transistor lebar dan panjang, dan  $V_{untuk}$  adalah tegangan ambang. Banyak asumsi penyederhanaan dibuat dalam derivasi Persamaan (4). Namun demikian, persamaan tersebut mengandung variabel di mana penundaan gerbang benar-benar tergantung, dan sifat efeknya diwakili dengan benar.

Kami akan mengasumsikan bahwa bandwidth yang dibutuhkan lebih rendah dari arsitektur yang diaktifkan (Asumsi 3), bersama dengan kapasitas kabel sakelar yang lebih rendah, sesuai dengan potensi pengurangan tegangan 50% ( $V_{bis} = 2 V_{beralih}$ ), sesuai dengan konsumsi energi empat kali lebih rendah.

Konsumsi energi  $P_{ba}$  diperlukan untuk mentransfer sejumlah data (paket) melalui arsitektur bus bersama dengan *tidak* perangkat diberikan oleh:

$$P_{ba} = N \cdot D \cdot (P_{bi} + P_{bw}) \quad (5)$$

Demikian pula, konsumsi energi  $P_{sa}$  diperlukan untuk mentransfer paket melalui arsitektur yang diaktifkan (dari perangkat sumber ke sakelar, dan dari perangkat sakelar ke perangkat sink) diberikan oleh:

$$P_{sa} = 2 \cdot (P_{si} + P_{sw} + P_{beralih}) \quad (6)$$

di mana  $P_{beralih}$  adalah konsumsi energi dari kain switching per koneksi antarmuka. Ini mengarah ke rasio berikut:  $R$  antara konsumsi energi arsitektur bus bersama dan konsumsi energi arsitektur yang diaktifkan.

$$R = \frac{P_{ba}}{P_{sa}} = \frac{N \cdot D \cdot (P_{bi} + P_{bw})}{2 \cdot (P_{si} + P_{sw} + P_{beralih})} \quad (7)$$

$P_{bw}$  dan  $P_{sw}$  bergantung pada kapasitas dan tegangan kabel. Ketika kita memasukkan efek dari Asumsi 5 ( $C_{bw} = 2 \cdot C_{barat}$ ), dan 8 ( $V_{bi} = 2 \cdot V_{beralih}$ ), kita dapat menyimpulkan bahwa  $P_{bw} = 8 \cdot P_{sw}$ . Menggunakan ini dan Asumsi 3 ( $P_{bi} = T/4 \cdot P_{si}$ ) kita dapat menulis ulang Persamaan (7) sebagai:

$$R = \frac{P_{ba}}{P_{sa}} = \frac{N \cdot D \cdot (N/4 \cdot P_{si} + 8 \cdot P_{barat})}{2 \cdot (P_{si} + P_{sw} + P_{beralih})} \quad (8)$$

Untuk memberikan perasaan efek dari bagaimana persamaan ini bekerja dalam sistem nyata, kami akan menggunakan nilai-nilai yang diperoleh dengan implementasi testbed kami dari arsitektur yang diaktifkan (the *Gurita* sakelar yang dijelaskan dalam Bab 4), dan bandingkan dengan sistem berbasis bus bersama dengan kinerja yang setara.

Dalam pengujian arsitektur Octopus kami menggunakan mikrokontroler kecil berdaya rendah yang beroperasi sebagai pengontrol antarmuka. Konsumsi daya ini mikrokontroler sekitar 26 mW (pada 3.3V, 33 MHz)<sup>3</sup>. Jadi nilai yang wajar dari  $P_{si}$  demikian 26 mW. Konsumsi energi  $P_{beralih}$  dari kain switching sekitar 15 mW per koneksi antarmuka implementasi testbed kami <sup>4</sup>.

Menghitung  $P_{sw}$  kami mengasumsikan antarmuka lebar 8 bit yang beroperasi pada 3,3 V dengan kapasitansi 5 pF per kawat. Ketika kita mengasumsikan frekuensi operasi menjadi 33 MHz dan probabilitas toggling 0,25 pada setiap siklus clock, maka disipasi daya  $P_{sw}$  akan menjadi:  $(8 \times 5 \text{ pF}) \times (0,25 \times 33 \text{ MHz}) \times 3,3^2 \text{ V}_2$

Ketika kita memiliki jumlah perangkat  $N=8$ , dan jumlah transfer DMA pada bus bersama  $D=2$ , maka rasionya  $R = 1344/90$

<sup>3</sup> Pengontrol khusus akan mengkonsumsi lebih sedikit energi. Perhatikan bahwa pengontrol bus PCI memiliki konsumsi daya yang jauh lebih tinggi, misalnya master bus PCI9060 PCI dari teknologi PLX memerlukan 680 mW untuk beroperasi [54].

<sup>4</sup> Testbed dirancang agar fleksibel. Ini hemat energi, dan tidak berdaya rendah (lihat Bab .)

4). Implementasi khusus akan mengkonsumsi lebih sedikit energi.

asumsi yang kami buat sangat konservatif dan kami telah menggunakan konsumsi daya a of *ranjang percobaan* arsitektur yang diaktifkan, arsitektur yang diaktifkan jauh lebih hemat energi daripada arsitektur bus bersama.

Dalam diskusi sebelumnya kami tidak mempertimbangkan kontrol daya dinamis, tetapi mengasumsikan koneksi antara dua perangkat dengan aliran data kontinu. Dalam situasi di mana tidak ada aktivitas pada interkoneksi, arsitektur sakelar yang dirancang dengan baik mampu beroperasi pada mode tidur berdaya rendah, sedangkan dalam arsitektur bus bersama antarmuka bus harus aktif sepanjang waktu.

### 3.3.7 Kontrol waktu

Itu *Pendamping Digital Seluler* memiliki pendekatan koneksi-sentris karena beberapa alasan seperti kinerja, ketentuan QoS, efisiensi energi, dan kompleksitas. Masih ada alasan lain untuk memilih arsitektur connection-centric. Semua alasan ini berasal dari mekanisme kontrol waktu dalam arsitektur.

#### Struktur kontrol waktu dasar

Pilihan dan desain struktur kontrol waktu yang tepat untuk suatu sistem adalah masalah vital dan juga sangat praktis. Skema waktu sinkron sering menjadi pilihan pertama dalam desain sistem karena kompleksitas perangkat keras yang rendah dan kesederhanaan desain logika. Di sebuah *sistem sinkron* sinyal clock melayani dua tujuan: sebagai referensi urutan, dan sebagai referensi waktu. Sebagai referensi urutan, transisi jam mendefinisikan contoh di mana sistem dapat mengubah keadaan (sehingga perubahan keadaan acak dan interferensi dapat dihilangkan). Sebagai referensi waktu, interval antara transisi tingkat jam mendefinisikan wilayah waktu di mana data dapat berpindah antara tahap pemrosesan yang berurutan atau diproses dalam tahap yang terisolasi dari yang lain. Sinyal clock dengan demikian dapat dilihat sebagai penjaga yang mengontrol kapan dan apa yang harus dilakukan atau tidak dilakukan. Untuk memastikan operasi sistem yang benar, skema distribusi jam harus digunakan untuk menghasilkan sinyal jam yang setara secara logis di seluruh sistem. Namun, kemiringan jam dalam sistem tidak dapat dihindari dan disebabkan oleh banyak faktor acak seperti berbagai penundaan propagasi sinyal pada kabel dan logika, variasi pemuatan kapasitif pada titik yang berbeda, dan variasi dalam parameter perangkat dan proses. Sinyal clock biasanya menggerakkan beban besar karena harus menjangkau banyak elemen berurutan yang didistribusikan ke seluruh chip. Oleh karena itu, sinyal clock telah menjadi sumber disipasi energi yang terkenal karena frekuensi dan beban yang tinggi. Telah diamati bahwa distribusi jam dapat memakan waktu hingga 45% dari total disipasi energi mikroprosesor kinerja tinggi [75]. Oleh karena itu, sinyal clock telah menjadi sumber disipasi energi yang terkenal karena frekuensi dan beban yang tinggi. Telah diamati bahwa distribusi jam dapat memakan waktu hingga 45% dari total disipasi energi mikroprosesor kinerja tinggi [75]. Oleh karena itu, sinyal clock telah menjadi sumber disipasi energi yang terkenal karena frekuensi dan beban yang tinggi. Telah diamati bahwa distribusi jam dapat memakan waktu hingga 45% dari total disipasi energi mikroprosesor kinerja tinggi [75].

Dalam sistem adaptif dan (kembali) dikonfigurasi, metode waktu sinkron menderita lebih banyak masalah. Dalam sistem seperti itu, karakteristik penundaan (baik dalam penundaan komunikasi dan penundaan komputasi) mungkin sangat berbeda dengan konfigurasi yang berbeda, dan tidak dapat, atau sangat terbatas, diperkirakan sebelumnya. Oleh karena itu, akan sangat sulit untuk menentukan kecepatan clock yang sesuai untuk sistem.

Karena semakin sulit untuk mendistribusikan jaringan jam global yang tepat melalui area silikon yang luas, dan semakin mahal untuk merancang jadwal yang efisien untuk sistem sinkron dengan jutaan transistor, *asinkron*

*metode desain* mungkin memberikan solusi. Sistem self-timed semacam itu dibangun dengan menguraikan sistem menjadi satu set blok logika kombinatorial dan memasukkan kontrol jabat tangan asinkron di antara setiap pasangan blok yang terhubung. Karena kompleksitas sirkuit jabat tangan meningkat secara drastis dengan jumlah input dan output, blok bangunan melakukan fungsi yang relatif sederhana. Karena tidak ada jam global dalam sistem seperti itu, kinerja sistem dan konsumsi energi bergantung pada data saat run-time. Energi dihaburkan hanya ketika sirkuit aktif. Akibatnya, sirkuit asinkron dapat memiliki kinerja energi yang luar biasa [9][46]. Namun, kompleksitas rangkaian untuk menerapkan logika kontrol handshaking dan area yang diperlukan untuk mengimplementasikan sistem semacam itu relatif tinggi jika ukuran logika yang terkait kecil. Juga, ada penundaan tambahan yang disebabkan oleh protokol jabat tangan dan logika yang diperlukan untuk mengimplementasikannya. Logika asinkron telah gagal untuk mendapatkan penerimaan di tingkat sirkuit, terutama berdasarkan kriteria area dan kinerja, tetapi juga karena kesulitan desain.

#### *Kontrol waktu dalam arsitektur connection-centric*

Suatu sistem secara umum – dan khususnya sistem connection-centric – terdiri dari dua bagian penting: satu set modul fungsional dan jaringan komunikasi yang menghubungkan modul-modul ini. Sistem yang paling efisien dengan kinerja tertinggi dapat dicapai jika modul dan jaringan komunikasi berjalan pada kinerja tertinggi dan/atau paling efisien, dan kinerja ini cocok satu sama lain. Jika semua modul fungsional dan jaringan komunikasi sistem diatur waktunya secara terpisah, maka ada peluang lebih baik untuk mencapai tujuan ini. Kelayakan memenuhi persyaratan seperti itu tidak hanya bergantung pada skema waktu, tetapi juga pada arsitektur sistem.

Pendekatan desain sinkron dan asinkron mewakili dua ekstrem, dan ada banyak varian di antaranya. Dalam sistem connection-centric, kombinasi yang menarik adalah menggunakan jam lokal ke modul logika individual untuk operasi sinkron di setiap modul, dan protokol asinkron antara modul fungsional untuk komunikasi asinkron di jaringan interkoneksi. Baru-baru ini beberapa penelitian (misalnya [2][25]) menunjukkan bahwa akan bermanfaat untuk mempertimbangkan pendekatan tersebut untuk menghilangkan kebutuhan mendistribusikan jam global antara blok granularity yang lebih besar. Dengan cara ini, sirkuit antarmuka akan mewakili komponen overhead yang sangat kecil, dan aspek yang paling memakan energi dari sirkuit sinkron (yaitu distribusi jam global) akan dihindari. Kesulitan waktu dalam sistem sinkron dilokalisasi ke logika di dalam modul, dan tidak mempengaruhi transfer data yang benar. Antarmuka antara modul dan jaringan antar-komunikasi menyinkronkan kejadian dalam protokol handshaking pada input modul sinkron dengan jam lokal di modul.

### **3.3.8 Kerangka Kualitas Layanan**

Aplikasi harus beradaptasi dengan lingkungan yang terus berubah dan mereka membutuhkan bantuan sistem operasi untuk menyediakan informasi untuk itu. Sistem operasi tradisional tidak memberi tahu aplikasi ketika jaringan sedang down, berapa banyak biaya komunikasi (dalam hal biaya per bit atau konsumsi energi), atau berapa banyak sumber daya CPU yang tersedia.



Adaptasi terhadap bandwidth jaringan yang tersedia sudah ada dalam konteks komunikasi multimedia. Ini bisa sangat berguna untuk aplikasi komputasi seluler juga untuk mengetahui pemadaman jaringan dan biaya komunikasi jaringan. SEBUAH *Pendamping Digital Seluler* mungkin berada di lokasi di mana komunikasi melalui jaringan mahal dan bandwidth rendah. Jika hal ini terjadi, sistem file (untuk menyebutkan satu contoh saja) mungkin dapat menyesuaikan perilakunya dengan baik dan berhenti mengambil terlebih dahulu untuk meningkatkan kinerja.

Jika seseorang menyelidiki dengan metode apa aplikasi dapat mengadaptasi Quality of Service (QoS), orang memperhatikan bahwa, untuk menjembatani perubahan substansial dalam alokasi sumber daya (CPU, konsumsi energi, dan bandwidth jaringan adalah sumber daya yang paling terpengaruh), hanya mengubah parameter tidak cukup. Dalam lingkungan seluler yang dinamis diperlukan perubahan yang lebih drastis, misalnya dengan mengubah algoritme. Di *MOBY DICK* arsitektur, Quality of Service adalah kerangka kerja untuk model integrasi dan manajemen terintegrasi dari semua layanan sistem dan aplikasi di *Sahabat Digital Seluler*. Konsumsi sumber daya oleh satu aplikasi dapat memengaruhi aplikasi lain, dan ketika sumber daya habis, semua aplikasi terpengaruh. Jika ketersediaan sumber daya berubah, apakah itu file, siklus CPU, atau konsumsi energi, aplikasi yang menggunakannya akan diberi tahu, dan mereka dapat menyesuaikan perilakunya. Misalnya, aplikasi yang memelihara buku harian terdistribusi akan meminta, untuk QoS tertingginya, untuk menggunakan tampilan file yang konsisten, tetapi, jika ini tidak dapat tersedia karena partisi jaringan, aplikasi akan menerima versi yang tidak konsisten sebagai hal terbaik berikutnya. Karena bandwidth komunikasi, konsumsi energi, dan perilaku aplikasi terkait erat, kami percaya bahwa kerangka kerja QoS adalah dasar yang kuat untuk manajemen sumber daya yang terintegrasi. *Sahabat Digital Seluler*.

Kerangka kerja QoS mempengaruhi sejumlah besar parameter dari berbagai komponen dalam sistem. Sebagian besar parameter ini juga memiliki dampak yang signifikan terhadap konsumsi energi, secara umum kualitas yang lebih tinggi juga membutuhkan lebih banyak energi. Konsumsi energi dengan demikian merupakan parameter penting dalam kerangka QoS. Untuk mengintegrasikan kesadaran daya dalam kerangka QoS, perubahan harus dilakukan pada perangkat keras, driver, firmware, sistem operasi, dan aplikasi. Sistem harus fleksibel, dan memiliki beberapa implementasi fungsi yang salah satunya dapat dipilih tergantung pada QoS dan sumber daya yang tersedia. Sistem operasi akan mengontrol status daya perangkat dalam sistem dan membagikan informasi ini dengan aplikasi dan pengguna.

Salah satu aspek kunci dari pendekatan QoS kami adalah memindahkan keputusan kebijakan manajemen daya kepada pengguna dan koordinasi operasi ke dalam sistem operasi. Sistem operasi akan mengontrol status daya perangkat dalam sistem dan membagikan informasi ini dengan aplikasi dan pengguna. Namun, ini tidak berarti bahwa modul tidak bertanggung jawab. Setiap modul memiliki – khusus – manajemen daya lokal. Hanya modul yang mampu, dan memiliki pengetahuan untuk menerapkan penyesuaian manajemen daya yang diperlukan untuk fungsi internal. Namun, kontrol manajemen daya modul secara keseluruhan dilakukan oleh sistem operasi dan pengguna. Untuk memanfaatkan mode daya rendah dari modul sistem,

### 3.4 Pekerjaan terkait

Pada bagian ini kami akan memberikan gambaran tentang pekerjaan terkait dalam berbagai topik yang dicakup oleh arsitektur Mobile Digital Companion: yaitu arsitektur multimedia, arsitektur heterogen, perangkat yang terhubung ke jaringan, dan manajemen energi.

#### 3.4.1 Arsitektur multimedia

Masalah desain arsitektur perangkat keras untuk prosesor berkinerja tinggi adalah topik yang dibahas secara luas dalam literatur. Berbagai arsitektur telah diusulkan untuk mengatasi masalah yang terkait dengan komputasi multimedia. Pendekatan ini didasarkan pada teknologi kinerja tinggi dan sebagian besar merupakan ekstensi sederhana untuk arsitektur saat ini. Sistem ini gagal memanfaatkan peluang pengurangan energi yang ditawarkan oleh multimedia.

Sistem seperti *InfoPad* [60][70] dan *ParcTab* [36] dirancang untuk memanfaatkan jaringan nirkabel berkecepatan tinggi untuk mengurangi jumlah komputasi yang diperlukan pada perangkat portabel. Sistem ini adalah semacam terminal portabel dan memanfaatkan kekuatan pemrosesan server komputasi jarak jauh. Pendekatan ini menyederhanakan desain dan mengurangi konsumsi daya untuk komponen pemrosesan, tetapi secara signifikan meningkatkan penggunaan jaringan dan dengan demikian berpotensi meningkatkan konsumsi energi karena antarmuka jaringan membutuhkan energi yang mahal. Sistem ini juga mengandalkan ketersediaan konektivitas jaringan bandwidth tinggi dan tidak dapat digunakan saat tidak terhubung.

UCLA telah membangun testbed jaringan [43] yang menggunakan arsitektur perangkat keras untuk melokalisasi data untuk komunikasi dan video. Dengan cara ini aliran data berkurang dan secara efisien ditransfer langsung ke tujuan mereka. Granularity sistem ini jauh lebih besar dari sistem sebelumnya. Evaluasi kinerja menggunakan testbed telah mengungkapkan kepentingan relatif dari overhead yang dikeluarkan oleh aplikasi dan protokol jaringan serta pemrosesan sinyal dalam perangkat keras video dan radio [15]. Untuk node berkinerja tinggi, overhead karena transfer bus, salinan memori, dan pemrosesan jaringan tinggi. Transfer bus adalah sumber utama keterbatasan throughput sistem untuk aplikasi yang membutuhkan pergerakan blok data yang besar melintasi bus sistem.

Beberapa tahun terakhir menunjukkan peningkatan penggunaan arsitektur khusus aplikasi di dunia tujuan umum. Dalam pendekatan ini operasi yang sering digunakan yang mahal dalam waktu komputasi diimplementasikan dalam perangkat keras khusus di dalam mikroprosesor. Unit perangkat keras sering disebut *akselerator perangkat keras*. Contoh tipikal dapat ditemukan di Intel's *MMX*™ arsitektur [39]. Untuk lebih meningkatkan kinerja beberapa instruksi dapat dilakukan secara paralel, contohnya dapat ditemukan di arsitektur VLIW (Very Large Instruction Word). Istilah prosesor media sering digunakan untuk kelas prosesor multimedia, terutama ditujukan untuk pasar multi-media-PC. Misalnya, *TriMedia* prosesor menggunakan arsitektur VLIW dengan akselerator perangkat keras dan jalan raya data untuk dapat menangani aplikasi seperti dekompresi audio dan video real-time [57]. Meskipun akselerator perangkat keras memungkinkan perancang untuk mengimplementasikan operasi tingkat yang lebih tinggi, tingkat ini masih dibatasi oleh persyaratan yang berlaku secara umum

instruksi untuk mendukung programabilitas tingkat tinggi. Jumlah paralelisme yang dapat diperoleh pada tingkat seperti itu agak terbatas, biasanya faktor 3 sampai 5 [38]. Jumlah konsumsi energi yang dibutuhkan umumnya tidak menjadi perhatian para desainer, dan tinggi.

### **3.4.2 Arsitektur paralel heterogen**

Dengan menambahkan koprosesor khusus di sebelah prosesor tujuan umum, butir operasi ditingkatkan ke tingkat fungsi lengkap yang dijalankan pada perangkat keras khusus. Namun, koprosesor tidak dapat beroperasi secara independen dari prosesor tujuan umum yang melakukan sinkronisasi tugas. Hal ini menyebabkan overhead yang signifikan dalam waktu eksekusi dan membatasi peningkatan konkurensi yang dapat diperoleh. Selain itu, kemacetan komunikasi dapat dengan mudah terjadi karena dalam aplikasi multimedia yang membutuhkan data dalam jumlah besar, bandwidth yang ditawarkan sangat tidak mencukupi karena semua prosesor harus berkomunikasi melalui bus yang sama. Memanfaatkan paralelisme tingkat fungsi dapat meningkatkan kinerja dan efisiensi pemrosesan.

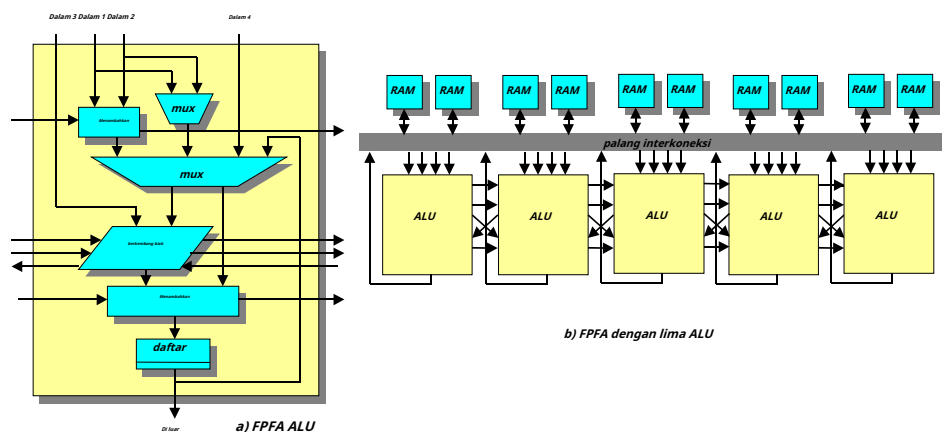
Abnous dan Rabaey mengusulkan arsitektur untuk aplikasi pemrosesan sinyal yang fleksibel dan menggunakan daya rendah [1]. Arsitekturnya terdiri dari prosesor kontrol yang dikelilingi oleh susunan heterogen prosesor satelit tujuan khusus yang otonom. Permintaan komputasi pada prosesor kontrol minimal, tugas utamanya adalah mengkonfigurasi sistem dan mengelola aliran kontrol keseluruhan dari algoritma pemrosesan sinyal yang diberikan. Prosesor satelit melakukan tugas komputasi algoritma yang dominan dan intensif energi. Perincian tugas-tugas ini relatif kecil. Beberapa contoh termasuk generator alamat, prosesor multi-akumulasi untuk menghitung produk titik vektor, dll. Arsitekturnya tidak mengizinkan multiplexing tugas yang berbeda pada prosesor yang sama. Ini membatasi tingkat efisiensi,

Nieuwland dan Lippens [52] mengusulkan arsitektur multiprosesor heterogen yang mendukung model memori global. Model seperti itu memungkinkan pemetaan ulang program tipikal saat ini dengan mudah pada elemen pemrosesan yang heterogen. Bus menghubungkan elemen pemrosesan yang heterogen. Memori lokal pada elemen pemrosesan diposisikan dalam satu pemetaan global aplikasi dan dapat diakses oleh semua elemen pemrosesan lainnya. Karena antarmuka komunikasi yang terdefinisi dengan baik, mengalokasikan tugas ke elemen pemrosesan lain tidak memerlukan perubahan pada perangkat lunak aplikasi yang tersisa. Eksperimen dalam perangkat lunak menunjukkan bahwa meskipun protokol komunikasi berjalan agak efisien, sebagian besar kecepatan hilang dalam komunikasi karena ukuran butir komunikasi yang kecil dengan tugas koprosesor.

Leijten telah mengusulkan template multiprosesor heterogen untuk dapat memperoleh kinerja pemrosesan [38]. Ini diperoleh dengan mengganti koprosesor dengan prosesor yang memiliki utas kontrolnya sendiri, yaitu, prosesor otonom dapat menjalankan tugas sepenuhnya secara independen dari mikroprosesor. Dalam solusi multiprosesor yang dihasilkan, mikroprosesor serba guna menjalankan tugas-tugas berkinerja rendah yang membutuhkan tingkat kemampuan program yang tinggi, sementara prosesor lain menjalankan tugas-tugas berkinerja tinggi yang hanya membutuhkan kemampuan pemrograman terbatas. Ini tinggi-

prosesor kinerja adalah *khusus domain aplikasi* (Prosesor ADS) dioptimalkan dalam hal kecepatan, area, dan daya, dan disetel ke arah serangkaian tugas yang terdefinisi dengan baik. Granularitas operasi prosesor relatif kecil, target utama dari sistem ini adalah untuk mengimplementasikan prosesor multimedia.

University of Twente telah mengembangkan arsitektur yang cocok untuk algoritme mirip DSP berdaya rendah yang dapat dikonfigurasi ulang. *Dapat Diprogram di Lapangan Fungsi Himpunan* (FPFA) perangkat mengingatkan pada FPGA, tetapi memiliki matriks ALU dan tabel pencarian, bukan CLB (Blok Logika yang Dapat Dikonfigurasi). Konstruksi ALU dari beberapa tabel pencarian selebar 1-bit tidak efisien energi [64]. Untuk berbagai fungsi multimedia yang menggunakan algoritme penyaringan digital pada data paralel: kompresi video (de), enkripsi data, dan tanda tangan digital, perangkat ini tidak memiliki kekuatan pemrosesan yang diperlukan. Untuk fungsi ini diperlukan perhitungan 16/32 bit (kalikan, tambah). Arsitektur yang lebih baru didasarkan pada unit fungsi 'chunky' seperti ALU lengkap dan pengganda. Misalnya, kumpulan pengganda mungkin tersedia bersama dengan interkoneksi palang untuk secara efisien mendukung berbagai filter respons impuls tak terbatas (IIR).



**Gambar 6: Arsitektur FPFA.**

Set instruksi dari FPFA-ALU dapat dianggap sebagai set instruksi ALU biasa, dengan pengecualian bahwa tidak ada operasi beban dan penyimpanan yang beroperasi pada memori. Sebaliknya, mereka beroperasi pada interkoneksi yang dapat diprogram; yaitu, ALU memuat operannya dari output ALU tetangga, atau dari nilai (input) yang disimpan dalam tabel pencarian atau register lokal. Eksekusi FPFA berbasis grafik digunakan untuk mengeksekusi loop dalam aplikasi. Struktur perangkat tujuan umum yang teratur memungkinkan peralihan konteks yang cepat dari satu loop dalam ke loop lainnya, sehingga konfigurasi ulang dapat dilakukan dengan cepat. Ini adalah bagaimana kelas yang luas dari algoritma komputasi intensif dapat diimplementasikan pada FPFA [63].

Di Laboratorium MIT untuk Ilmu Komputer, arsitektur baru sedang dikembangkan yang menghilangkan antarmuka set instruksi tradisional dan sebagai gantinya menggunakan direplikasi

arsitektur langsung ke compiler [73]. Hal ini memungkinkan compiler untuk menentukan dan mengimplementasikan alokasi sumber daya terbaik untuk setiap aplikasi. Mereka menyebut sistem berdasarkan pendekatan itu *Arsitektur mentah* karena mereka hanya menerapkan mekanisme minimal dalam perangkat keras. Arsitektur didasarkan pada satu set ubin yang saling berhubungan, yang masing-masing berisi instruksi dan memori data, unit logika aritmatika, register, logika yang dapat dikonfigurasi, dan sakelar yang dapat diprogram yang mendukung perutean statis yang diatur oleh compiler dan dinamis. Arsitektur mentah paling cocok untuk komputasi pemrosesan sinyal berbasis aliran.

Perangkat TMS320C80 adalah chip tunggal, prosesor paralel yang ditujukan untuk aplikasi seperti pemrosesan audio/video real-time, komunikasi data kelas atas, dan pemrosesan gambar [69]. Chip ini membuktikan kemungkinan menempatkan beberapa prosesor yang saling berhubungan pada satu chip. Perangkat kompleks ini berisi empat DSP pemrosesan paralel (PP) dengan instruksi 64-bit dan data titik-tetap 32-bit; prosesor master RISC (MP) dengan unit titik mengambang; 50 Kbytes RAM on-chip; Pengontrol Video; dan pengontrol transfer yang melayani permintaan data dan cache yang terlewat oleh MP dan PP. Sakelar paling menyediakan akses MP dan PP ke memori on-chip.

#### 3.4.3 Perangkat yang terhubung ke jaringan

Network-attached peripheral (NAP) adalah perangkat komputer yang berkomunikasi melalui jaringan daripada bus I/O tradisional, seperti SCSI [20]. Beberapa proyek penelitian menggunakan periferal terpasang jaringan di workstation multimedia sedang berlangsung di berbagai universitas. Contoh kanonik penggunaan NAP dalam multimedia adalah keinginan untuk mengirimkan data secara langsung dari kamera ke buffer bingkai tanpa melewati bidang belakang sistem, di mana ia menghabiskan bandwidth secara tidak produktif. Pengambilan video ke disk dan pemutaran dari disk serupa. Kami sekarang hanya akan menyebutkan beberapa contoh tipikal.

*Jaringan Area Meja* – Salah satu cara untuk memberikan jaminan real-time saat mentransfer data di dalam workstation adalah dengan juga menggunakan switch jaringan ATM untuk menghubungkan komponen sistem workstation. Pekerjaan ini telah dilakukan di Cambridge [32] dan beberapa dari pekerjaan ini sekarang telah dikomersialkan oleh Nemesys. Desk-Area Network (DAN) membawa ide ini secara ekstrim dimana switch ATM juga digunakan untuk menghubungkan memori.

*VuNet* – Arsitektur VuNet dirancang sebagai bagian dari proyek multimedia ViewStation di MIT [4]. VuNet adalah jaringan gigabit per detik menggunakan ATM, yang menghubungkan stasiun kerja serba guna dan perangkat multimedia. VuNet dimaksudkan untuk digunakan sebagai jaringan area meja dan area lokal. Dalam pendekatan mereka, informasi multimedia disalurkan semua ke prosesor workstation daripada melewatinya dengan perangkat keras khusus. Mereka berharap bahwa dengan tingkat kemajuan kinerja workstation saat ini, tingkat kinerja yang memungkinkan tugas-tugas multimedia untuk dieksekusi secara paralel dengan tugas-tugas lain, akan segera tercapai. Karena pendekatan intensif perangkat lunak untuk multimedia, VuNet dan perangkat keras video khusus dirancang untuk memberikan dukungan yang efisien untuk penanganan aliran multimedia yang digerakkan oleh perangkat lunak.

FIFO yang dapat menyangga 64 sel pada arah transmisi dan 256 sel pada arah penerimaan.

*Switcherland* – Arsitektur komunikasi scalable ini didasarkan pada crossbar switch yang memberikan jaminan QoS untuk cluster workstation dalam bentuk bandwidth yang dicadangkan dan penundaan transmisi terbatas [21]. Mirip dengan teknologi ATM, Switcherland memberikan jaminan QoS dengan bantuan kelas layanan. Target utama mereka adalah untuk memberikan kinerja tinggi dan ketersediaan prosesor dan perangkat I/O yang baik dengan mengizinkan topologi sembarang. Sakelar dapat digunakan sebagai jalinan interkoneksi I/O dari sebuah stasiun kerja serta sebagai jalinan interkoneksi jaringan dari kluster stasiun kerja.

### **3.4.4 Manajemen energi**

Salah satu teknik paling sukses yang digunakan oleh perancang komputer saat ini di tingkat sistem adalah manajemen daya dinamis [8]. Namun, ada beberapa sistem operasi yang dirancang khusus untuk peralatan komputasi portabel. Microsoft Windows CE [26] adalah satu, USRobotics PalmOS [71] adalah yang lain. Manajemen daya dalam sistem ini hampir secara eksklusif terdiri dari mematikan CPU dan perangkat lain ketika sistem menjadi tidak aktif dan mematikan layar setelah beberapa menit pengguna tidak aktif.

Saat ini beberapa pengembang dan vendor sistem sedang mengejar strategi jangka panjang dan cakupan luas untuk sangat menyederhanakan tugas sistem yang dikelola daya yang besar dan kompleks. Strategi ini didasarkan pada inisiatif standarisasi yang dikenal sebagai *Konfigurasi Lanjutan dan Antarmuka Daya (ACPI)*. Itu *Sekarang* inisiatif menargetkan migrasi algoritma dan kebijakan manajemen daya ke dalam sistem operasi komputer [49]. OnNow dan ACPI menyediakan kerangka kerja bagi desainer untuk menerapkan strategi manajemen daya. Pilihan kebijakan diserahkan kepada perancang. OnNow adalah inisiatif yang diusulkan oleh satu perusahaan perangkat lunak, dan terikat erat dengan model abstrak komputer pribadi. Meskipun OnNow membutuhkan ACPI sebagai antarmuka antara sistem operasi dan perangkat keras, ACPI lebih umum dalam cakupan dan tidak bergantung pada sistem operasi atau model perangkat keras. Namun, baik ACPI dan OnNow mengasumsikan sistem sentris CPU dan sistem operasi, di mana aktivitas sistem dikelola oleh satu entitas. ACPI dan OnNow dikembangkan untuk mendukung implementasi sistem komputer yang dikelola daya,

Pendekatan pemodelan yang ditujukan untuk memberikan dukungan untuk eksplorasi arsitektur tingkat sistem dari sistem yang dikelola daya dijelaskan dalam [8]. Dalam model mereka, sistem didefinisikan oleh satu set komponen dan pola komunikasi antar komponen. Komunikasi dimodelkan oleh peristiwa-peristiwa abstrak. Abstraksi model jauh lebih tinggi daripada ACPI, dan tidak ada detail yang ditentukan tentang perilaku fungsional penyedia layanan (seperti unit driver disk atau driver video).

### 3.5 Ringkasan dan kesimpulan and

Dalam bab ini kami mempertimbangkan masalah merancang arsitektur untuk komputer multimedia seluler genggam. Arsitektur dari *Pendamping Digital Seluler* adalah connection-centric di mana modul berkomunikasi menggunakan antarmuka handshaking asinkron. Modul-modul ini dapat kombinatorial atau dikendalikan oleh jam lokal untuk setiap modul. CPU dipindahkan dari aliran data, meskipun masih berpartisipasi dalam aliran kontrol. Pendekatan desain semacam itu menawarkan solusi dalam desain multimedia, terminal nirkabel berdaya rendah. Arsitektur menyajikan beberapa keunggulan dibandingkan model memori-sentris tradisional.

Manajemen energi adalah tema umum dalam desain arsitektur sistem karena masa pakai baterai terbatas dan bobot baterai merupakan faktor penting. Kami telah menunjukkan bahwa ada hubungan penting antara arsitektur perangkat keras, arsitektur sistem operasi dan arsitektur aplikasi, di mana masing-masing mendapat manfaat dari yang lain. Dalam arsitektur kami, kami telah menerapkan beberapa teknik pengurangan energi tambahan pada semua tingkat sistem. Mencapai efisiensi energi yang tinggi pertama-tama membutuhkan penghapusan limbah yang biasanya mendominasi konsumsi energi pada prosesor serba guna. Prinsip utama kedua yang digunakan adalah memiliki locality of reference yang tinggi. Filosofinya adalah bahwa semua operasi yang diperlukan pada data harus dilakukan di tempat yang paling efisien,

Karena Mobile Digital Companion harus tetap dapat digunakan di berbagai lingkungan, ia harus cukup fleksibel untuk mengakomodasi berbagai layanan multimedia dan kemampuan komunikasi serta beradaptasi dengan berbagai kondisi pengoperasian dengan cara (energi) yang efisien. Pendekatan yang dilakukan untuk mencapai sistem seperti itu adalah dengan menggunakan komponen yang otonom dan dapat disesuaikan, saling berhubungan dengan sakelar daripada dengan bus, dan menurunkan beban sebanyak mungkin dari CPU ke modul yang dapat diprogram yang ditempatkan di aliran data. Dengan demikian, komunikasi antar modul dikirimkan tepat ke tempat yang dibutuhkan, pekerjaan dilakukan di mana data melewati, melewati memori 'utama', modul secara otonom memasuki mode hemat energi dan menyesuaikan diri dengan kondisi sumber daya saat ini, dan kebutuhan pengguna. Jika buffering diperlukan sama sekali, itu ditempatkan tepat di jalur data, di mana dibutuhkan. Modul khusus domain aplikasi menawarkan fleksibilitas yang cukup untuk dapat mengimplementasikan serangkaian aplikasi serupa (biasanya) yang telah ditentukan sebelumnya, sambil menjaga biaya dalam hal area dan konsumsi energi ke tingkat rendah yang dapat diterima.

Memiliki arsitektur hemat energi yang mampu menangani kemampuan beradaptasi dan fleksibilitas dalam lingkungan multimedia seluler membutuhkan lebih dari sekadar platform perangkat keras yang sesuai. Pertama-tama kita perlu memiliki arsitektur sistem operasi yang dapat menangani platform perangkat keras dan kemampuan beradaptasi serta fleksibilitas perangkatnya. Pengoptimalan di berbagai lapisan dan fungsi, tidak hanya di tingkat sistem operasi, sangat penting. Mengelola dan mengeksplorasi keragaman ini adalah masalah desain sistem utama. Sebuah model yang mencakup berbagai tingkat perincian sistem sangat penting dalam desain sistem manajemen energi dan dalam membantu perancang sistem dalam membuat keputusan yang tepat dalam banyak trade-off yang dapat dibuat dalam

desain sistem. Akhirnya, untuk sepenuhnya memanfaatkan kemungkinan yang ditawarkan oleh perangkat keras yang dapat dikonfigurasi ulang, kita perlu memiliki dukungan sistem operasi yang tepat untuk komputasi yang dapat dikonfigurasi ulang, sehingga komponen ini dapat diprogram ulang secara memadai ketika sistem atau aplikasi dapat mengambil manfaat darinya.

Meskipun desain kami mengasumsikan komputer multimedia nirkabel berdaya rendah, sebagian besar ide kami dapat diterapkan (mungkin dengan beberapa modifikasi) untuk banyak jenis komputer (sub) sistem lainnya, termasuk workstation kinerja tinggi dan antarmuka jaringan.



## Referensi

- [1] Abnous A., Seno K., Ichikawa Y., Wan M., Rabaey J.: "Evaluasi arsitektur DSP yang dapat dikonfigurasi ulang dengan daya rendah", *proses 5<sup>ini</sup> Lokakarya Arsitektur yang Dapat Dikonfigurasi Ulang (RAW'98)*, 30 Maret 1998, Orlando, AS. (URL:[http://xputers.informatik.uni-kl.de/RAW/RAW98/adv\\_prq\\_RAW98.html](http://xputers.informatik.uni-kl.de/RAW/RAW98/adv_prq_RAW98.html))
- [2] Abnous A., Rabaey J.: "Prosesor multimedia khusus domain berdaya ultra rendah", *Pemrosesan sinyal VLSI IX*, ed. W. Burleson dkk., IEEE Press, hlm. 459-468, November 1996.
- [3] Adam J.: "Multimedia interaktif – aplikasi, implikasi", *Spektrum IEEE*, hlm. 24-29 Maret 1993.
- [4] Adam JF, Houh HH, Tennenhouse DL: "Pengalaman dengan VuNet: arsitektur jaringan untuk sistem multimedia terdistribusi", *Prosiding Konferensi ke-18 IEEE tentang Jaringan Komputer Lokal*, hlm. 70-76, Minneapolis MN, September 1993.
- [5] Agarwal A.: "Perhitungan mentah", *Amerika ilmiah*, hlm. 44-47, Agustus 1999.
- [6] Balakrishnan H., et al.: "Perbandingan mekanisme untuk meningkatkan kinerja TCP melalui tautan nirkabel", *Prosiding ACM SIGCOMM'96*, Stanford, CA, AS, Agustus 1996.
- [7] Barham P., Hayter M., McAuley D., Pratt I.: "Perangkat di Jaringan Area Meja", Maret 1994.
- [8] Benini L., De Micheli G.: "Manajemen Daya Dinamis, teknik desain, dan alat CAD", *Penerbit Akademik Kluwer*, ISBN 0-7923-8086-X, 1998.
- [9] Berkel K., dkk.: "Pengoreksi kesalahan daya rendah yang sepenuhnya asinkron untuk pemutar DCC", *Intisari Makalah Teknis, Konferensi Sirkuit Solid-State Internasional*, hal.88-89, 1994.
- [10] Bhoedjang, RAF, Rühl T., Bal HE: "Protokol antarmuka jaringan tingkat pengguna", *Komputer*, November 1998, hlm. 53-60.
- [11] Bosch P.: "Sistem file media campuran", *Ph.D. tesis Universitas Twente*, Juni 1999.
- [12] Burger D., Goodman J.: "Arsitektur Miliar transistor", *Komputer*, September 1997, hlm. 46-47.
- [13] Chaiken D., Hayter M., Kistler J., Redell D.: "Buku Virtual", *Laporan Penelitian SRC 157*, Pusat Penelitian Sistem Digital, November 1998.
- [14] Chandrakasan A., Brodersen RW: "Terminal Multimedia Portabel", *Majalah Komunikasi IEEE*, hal.64-75, jilid. 30, tidak. 12 Desember 1992.
- [15] Chien C., et al.: "Sebuah testbed terintegrasi untuk komputasi multimedia nirkabel", *Jurnal Sistem Pemrosesan VLSI* 13, hlm. 105-124, 1996.
- [16] Dally W.: "Mesin Komputasi Besar", pidato utama, *Simposium Internasional Keempat Arsitektur Komputer Berkinerja Tinggi*, Februari 1998.
- [17] Diependorff K., Dubey P.: "Bagaimana beban kerja multimedia akan mengubah desain prosesor", *Komputer*, September 1997, hlm.43-45.
- [18] Ditta ZD, Cox RC, Parulkar GM: "Mengejar jaringan: host I/O dengan kecepatan gigabit", *Laporan teknis WUCS-94-11*, Universitas Washington di St. Louis, April 1994.

- [19] Dorward S., Pike R., Presotto D., Ritchie D., Trickey H., Winterbottom P.: "Inferno", *Prosiding COMPCON Spring'97*, 42<sup>dan</sup> Konferensi Komputer Internasional IEEE, 1997, URL: <http://www.lucent.com/inferno>.
- [20] Doyle van Meter, R.: "Survei singkat tentang pekerjaan saat ini pada periferal yang terhubung ke jaringan", *Tinjauan Sistem Operasi ACM*, Januari 1996.
- [21] Eberle H., Oertli E.: "Switcherland: arsitektur komunikasi QoS untuk cluster workstation", *Prosiding ISCA '98 – 25 tahunan Int. Symposium Arsitektur Komputer*, Barcelona, Juni 1998.
- [22] von Eicken, T., Vogels, W.: "Evolusi Arsitektur Antarmuka Virtual", *Komputer*, hlm. 61-68, November 1998
- [23] Estrin G.: "Organisasi Sistem Komputer: Komputer Struktur Variabel Tetap-plus", *Prosiding Konferensi Komputer Bersama Barat*, hlm. 33-40, 1960.
- [24] Flynn MJ: "Apa yang ada di depan dalam desain komputer?", *prosiding Euromicro 97*, hlm. 4-9, September 1997.
- [25] Gao B., Rees DR: "Mengkomunikasikan modul logika sinkron", *konferensi Euromicro ke-21*, September 1995.
- [26] O'Hara, R.: "Microsoft Windows CE: Sejarah dan Desain", *Sistem genggam 5.1*, Jan./Feb. 1997, tersedia di <http://www.cdpubs.com/Excerpts.html>.
- [27] Memiliki PJM, Smit GJM: "Rattlesnake – sakelar ATM kinerja tinggi satu chip", *prosiding Konferensi internasional tentang jaringan multimedia (MmNet'95)*, hlm. 208-217, Aizu, Jepang, 26-29 September 1996.
- [28] Havea, PJM, Smit, GJM: "Meminimalkan konsumsi energi untuk komputer nirkabel di Moby Dick", *prosiding Konferensi Internasional IEEE tentang Komunikasi Nirkabel Pribadi ICPWC'97*, Desember 1997.
- [29] Memiliki PJM, Smit GJM: "Teknik desain sistem daya rendah untuk komputer bergerak", *Seri laporan teknis CTIT 97-32*, Enschede, Belanda, 1997
- [30] Memiliki PJM, Smit GJM: "Arsitektur Sahabat Saku", *Sekolah musim panas Euromicro pada komputasi seluler '98*, Oulu, hlm. 25-34, Agustus 1998
- [31] Memiliki PJM, Smit GJM, Bos M.: "Desain ATM nirkabel hemat energi", *prosiding wmATM'99*, Juni 1999.
- [32] Hayter MD, McAuley DR: "Jaringan area meja", *Tinjauan sistem operasi ACM*, Jil. 25 No 4, hlm. 14-21, Oktober 1991.
- [33] Helm A.: "Sistem untuk transaksi elektronik yang dikendalikan pengguna secara aman", *PhD. tesis Universitas Twente*, Agustus 1997.
- [34] HH Houh, Adam JF, Ismert M., Lindblad CJ, Tennenhouse DL: "Jaringan area meja VuNet: arsitektur, implementasi, dan pengalaman", *Jurnal IEEE Area Terpilih dalam Komunikasi (JSAC)*, 13(4):710-121, Mei 1995 (lihat juga: <http://www.tns.lcs.mit.edu/ViewStation/src/html/publications/JSAC95.html>)
- [35] Hui J.: "Teori switching dan lalu lintas untuk jaringan broadband terintegrasi", *Pers Akademik Kluwer*, 1990.
- [36] C. Kantarjiev dkk.: "Pengalaman dengan X dalam lingkungan nirkabel", *Simposium komputasi seluler dan lokasi independen*, Cambridge MA, Agustus 1993.

- [37] Kozyrakis CE, Patterson DA: "Arah baru untuk penelitian arsitektur komputer", *Komputer*, November 1998, hlm. 24-32,
- [38] Leijten JA: "Komunikasi yang dapat dikonfigurasi ulang secara real-time antara prosesor tertanam", *Ph.D. tesis, Universitas Teknologi Eindhoven*, Nopember 1998.
- [39] Lempel, O., Peleg A., Weiser U.: "Teknologi MMX™ Intel – ekstensi set instruksi baru", *Prosiding IEEE COMPCON*, hal.255-259, 1997.
- [40] Leslie I., D. McAuley, DL Tennenhouse: "ATM Di Mana Saja?", *Jaringan IEEE*, Maret 1993.
- [41] Lettieri P., Srivastava MB: "Kemajuan dalam terminal nirkabel", *Komunikasi Pribadi IEEE*, hlm. 6-19 Februari 1999.
- [42] Lorch JR: "Gambaran lengkap tentang konsumsi energi komputer portabel", *Tesis master, Ilmu Komputer, University of California di Berkeley*, 1995.
- [43] Mangione-Smith, B. et al.: "Arsitektur berdaya rendah untuk sistem multimedia nirkabel: pelajaran yang didapat dari membangun power hog", *prosiding simposium internasional tentang elektronika daya rendah dan desain (ISLPED) 1996*, Monterey CA, AS, hlm. 23-28, Agustus 1996.
- [44] Mangione-Smith WH, dkk.: "Mencari solusi dalam komputasi yang dapat dikonfigurasi", *Komputer IEEE*, hlm. 38-43, Desember 1997.
- [45] Mangione-Smith WH, Hutchings BL: "Komputasi yang dapat dikonfigurasi: jalan di depan", *1997 lokakarya arsitektur yang dapat dikonfigurasi ulang*, 1997.
- [46] Martin AJ, Burns SM, Lee TK, Borkovic D., Hazewindus PJ: "Mikroprosesor asinkron pertama: hasil pengujian", *Berita Arsitektur Komputer*, 17(4):95-110, Juni 1989.
- [47] Mehra R., Rabaey J.: "Mengeksploitasi keteraturan untuk desain berdaya rendah", *prosiding Konferensi internasional tentang desain berbantuan komputer*, 1996.
- [48] Mehra R., Lidsky DB, Abnous A., Landman PE, Rabaey JM: "Algoritma dan metodologi tingkat arsitektur untuk daya rendah", Bagian 11 dalam "*Metodologi desain daya rendah*"; editor J. Rabaey, M. Pedram, Penerbit Akademik Kluwer, 1996.
- [49] Microsoft: "OnNow dan Manajemen Daya", <http://microsoft.com/hwdev/onnow.htm>.
- [50] Mullender SJ, Corsini P., Hartvigsen G. "Moby Dick – The *Pendamping Digital Seluler*", LTR 20422, Lampiran I – Program Proyek, Desember 1995 (lihat juga <http://www.cs.utwente.nl/~havea/pp.html>).
- [51] Mullender SJ, Smit GJM, Havea PJM, Helme A., Hartvigsen G., Fallmur T., Stabellkulo T., Bartoli A., Dini G., Rizzo L., Avvenuti M.: "Arsitektur Moby Dick", *Seri laporan Teknis CTIT*, 98-18, Enschede, Belanda, 1998.
- [52] Nieuwland AK, Lippens PER: "Arsitektur HW-SW yang heterogen untuk terminal multi-media genggam", *prosiding lokakarya IEEE tentang Sistem Pemrosesan Sinyal*, SiPS'98, hlm. 113-122.
- [53] PedramM.: "Minimisasi daya dalam desain IC: prinsip dan aplikasi", *Transaksi ACM pada Otomasi Desain*, Jil. 1, tidak. 1, hlm. 3-56, Januari 1996.
- [54] Teknologi PLX: "PCI9060, chip antarmuka master PCI Bus untuk adaptor dan sistem tertanam", lembar data, 1995, <http://www.plxtech.com/download/9060/datasheets/9060-12.pdf>.
- [55] Prycker: "Mode Transfer Asinkron", 1991.
- [56] Rambus Inc.: "Pengungkapan Teknologi Rambus Langsung", <http://www.rambus.com>.

- [57] Rathnam S., Slavenburg G.: "Ikhtisar arsitektur prosesor multimedia yang dapat diprogram, TM-1", *Prosiding IEEE COMPCON*, hal. 319-326, 1996.
- [58] Reiniger D., Izmailov R., Rajagopalan B., Ott M., Raychaudhuri D.: "Kontrol QoS lunak dalam sistem nirkabel broadband WATMnet", *Komunikasi Pribadi IEEE*, hlm. 34-43, Februari 1999.
- [59] Buku elektronik roket, <http://www.rocket-ebook.com>.
- [60] Sheng S., Chandrakasan A., Brodersen RW: "Terminal Multimedia Portabel", *Majalah Komunikasi IEEE*, hal.64-75, jilid. 30, tidak. 12 Desember 1992.
- [61] Smit GJM: "Desain sistem komunikasi sakelar pusat untuk aplikasi multimedia", *Ph.D. tesis, Universitas Twente*, 1994.
- [62] Smit GJM, Havea PJM, dkk.: "Ikhtisar proyek Moby Dick", *1st Sekolah musim panas Euromicro tentang komputasi seluler*, hlm. 159-168, Oulu, Agustus 1998.
- [63] Smit J., Bosma M.: "Algoritme grafik pada Array Fungsi yang Dapat Diprogram Bidang", *proses 11<sup>ini</sup> Lokakarya EuroGraphics tentang perangkat keras grafis*, Ed. BO Schneider dan A. Schilling, hal.103-108, 1996.
- [64] Smit J., Stekelenburg M., Klaassen CE, Mullender S., Smit G., Havea PJM: "Biaya rendah & perputaran cepat: unit eksekusi berbasis grafik yang dapat dikonfigurasi ulang", *proses 7<sup>ini</sup> bengkel BELSIGN*, Enschede, Belanda, 7-8 Mei 1998.
- [65] Pembaca Buku Lunak, <http://www.softbook.com>.
- [66] Srivastava M.: "Desain dan optimalisasi sistem informasi nirkabel jaringan", *bengkel IEEE VLSI*, April 1998.
- [67] Steenkiste PA Zill BD, Kung HT, Schlick SJ, Hughes J., Kowalski B., Mullaney J.: "Arsitektur antarmuka host untuk jaringan kecepatan tinggi", *Prosiding 4<sup>ini</sup> Konferensi IFIP tentang jaringan kinerja tinggi*, hlm. A3-1 A3-16, Desember 1992.
- [68] Steenkiste P.: "Desain, implementasi, dan evaluasi tumpukan protokol salinan tunggal", *Perangkat lunak – latihan dan pengalaman*, Januari 1998.
- [69] Instrumen Texas, Prosesor Sinyal Digital SMJ320C80, <http://www.ti.com/sc/docs/products/sm320C80.html>.
- [70] Truman TE, Pering T., Doering R., Brodersen RW: Terminal multimedia InfoPad: perangkat portabel untuk akses informasi nirkabel", *transaksi IEEE di komputer*, Jil. 47, No. 10, hlm. 1073-1087, Oktober 1998.
- [71] USRobotics PalmOS, URL: <http://palmpilot.3com.com>.
- [72] Villasenor J., Mangione-Smith WH: "Komputasi yang Dapat Dikonfigurasi", *Amerika ilmiah*, Juni 1997.
- [73] Waingold E, Michael Taylor, Devabhaktuni Srikrishna, Vivek Sarkar, Walter Lee, Victor Lee, Jang Kim, Matthew Frank, Peter Finch, Rajeev Barua, Jonathan Babb, Saman Amarasinghe, dan Anant Agarwal: "Menunjukkan semuanya ke Perangkat Lunak: Mesin Mentah", *Komputer IEEE*, September 1997, hlm. 86-93.
- [74] Forum Protokol Aplikasi Nirkabel Ltd.: "Protokol Aplikasi Nirkabel Resmi", *Penerbitan Komputer Wiley*, 1999, <http://www.wapforum.org>.
- [75] Ya GK: "Desain VLSI digital berdaya rendah yang praktis", *Penerbit Akademik Kluwer*, ISBN 0-7923-80.

- [76] Zhang H., Wan M., George V., Rabaey J.: "Eksplorasi arsitektur interkoneksi untuk DSP chip tunggal yang dapat dikonfigurasi ulang dengan energi rendah", *Prosiding WVLSI*, Orlando, Fl, April 1999.

