

# MODEL *DISTRIBUTED-DATA PROCESSING* BERBASIS *OPEN-SOURCE* UNTUK PENGOLAHAN DATA KEPENDUDUKAN DALAM RANGKA IMPLEMENTASI *E-GOV* INDONESIA

Nonot Harsono, Sritrusta Sukaridhoto, Ahmad Subhan, Amang Sudarsono, [nonot@eepis-its.edu](mailto:nonot@eepis-its.edu)

Laboratorium Jaringan Komputer  
Politeknik Elektronika Negeri Surabaya

## ABSTRAK

Salah satu fungsi yang diharapkan ada di dalam sistem informasi nasional nanti adalah sistem verifikasi penduduk. Manfaatnya luas, misalnya untuk verifikasi nasabah perbankan, melacak pelaku kejahatan dari sidik jari yang ditemukan, dan manfaat yang lain. Sistem verifikasi yang umum adalah memanfaatkan teknologi biometric seperti sidik jari, pengenalan wajah, deteksi DNA, pengenalan suara, dan retina mata. Data penduduk Indonesia yang jumlahnya lebih dari 200 juta tersebar di lebih dari 30 provinsi dan lebih dari 400 kabupaten, merupakan data berukuran besar dan terdistribusi. Dalam paper ini disampaikan sebuah model *distributed data processing* yang difokuskan pada fungsi verifikasi penduduk dengan memanfaatkan teknologi biometric utamanya sidik jari. Disini ditunjukkan bahwa proses verifikasi akan lebih cepat bila digunakan teknologi *parallel-processing*. Sistem verifikasi penduduk secara nasional akan terbangun apabila infrastruktur jaringan komunikasi tersedia dan tertata baik.

**Kata kunci:** *distributed data, parallel processing*

## 1. PENDAHULUAN

### 1.1. Distributed Data

Indonesia dengan jumlah penduduk lebih dari 200 juta jiwa dan tersebar secara tidak merata di wilayah kepulauan yang sangat luas akan memiliki ukuran file data yang besar dan terdistribusi secara tidak merata pula. Andai dalam data base nasional tersimpan foto semua penduduk ukuran passport dengan format JPEG, maka ukuran file foto ini akan sebesar kira-kira 7 TB (satu foto 35 KB); dan untuk data sidik jari yang per orangnya sekitar 100 KB, ukuran data base nasional sidik jari penduduk adalah sekitar 20 TB. Ini belum termasuk data pribadi dan bentuk data yang lain. Untuk menangani data yang demikian besar, perlu dibentuk suatu hirarki pusat data yang nantinya bisa sejalan dan mendukung proses administrasi pemerintahan. Bila mengikuti pola otoritas otonomi daerah, maka pusat data nasional sebaiknya dibuat hingga level kabupaten saja, selanjutnya bisa disebut KDC (Kabupaten Data Center). Kemudian, di atas KDC dibentuk sebuah pusat data provinsi, selanjutnya bisa disebut PDC (Province Data Center). Di atas PDC adalah pusat data nasional yang bisa disebut sebagai IDC (Indonesia Data Center). Data base di KDC harus selalu di update melalui mekanisme administrasi daerah, dari Kelurahan, Kecamatan, hingga Kabupaten.

### 1.2. Parallel Processing

Parallel processing adalah upaya mempercepat proses eksekusi dari suatu program dengan cara membagi program menjadi fragmen-fragmen yang dijalankan di dalam beberapa prosesor yang terpisah. Suatu program yang dieksekusi oleh  $n$  prosesor, secara teoritis akan  $n$  kali lebih cepat. Namun dalam implementasinya, ada beberapa konfigurasi hardware yang bisa disebut sebagai parallel processing. Pertama, dalam satu CPU terdapat lebih dari satu prosesor dengan shared memory; kedua, dalam satu CPU terdapat beberapa/banyak prosesor dengan masing-masing memiliki memory sendiri; dan ketiga adalah beberapa atau banyak CPU terhubung ke jaringan komputer untuk mengeksekusi suatu program secara bersama-sama. Dalam penelitian ini, tim penulis mencoba konfigurasi yang ketiga, yang sering juga disebut sebagai *distributed processing*.

## 2. UJI COBA SKALA LAB

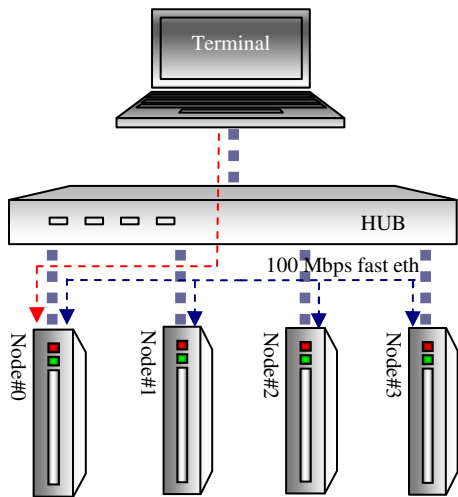
### 2.1. Konfigurasi Parallel Processor

Pada skala laboratorium, tim PENS telah mencoba membangun sebuah sistem parallel processing menggunakan embedded system SH4 yang diujar sebanyak empat buah. Spesifikasi dari masing-masing CPU adalah seperti pada Tabel 2.1 berikut.

**TABEL 2.1. SPECIFIKASI PERANGKAT KERAS**

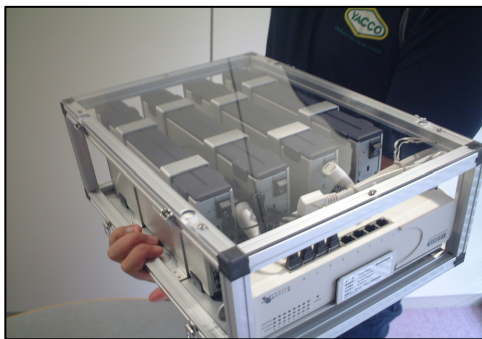
CPU	SH4 (SH7751R, 266MHz)
Memory	64MB SDRAM
HDD	120GB, ATA133, 5400rpm
NIC	10/100 BASE-T (RTL-8139C+)
I/F	USB 2.0x2port
Power	14 Watt

Keempat CPU kecil ini dihubungkan melalui sebuah hub dan terhubung ke PC notebook yang berfungsi sebagai monitor. Konfigurasi hardware-nya adalah seperti pada Gambar 2.1.



**GAMBAR 2.1. KONFIGURASI HARDWARE**

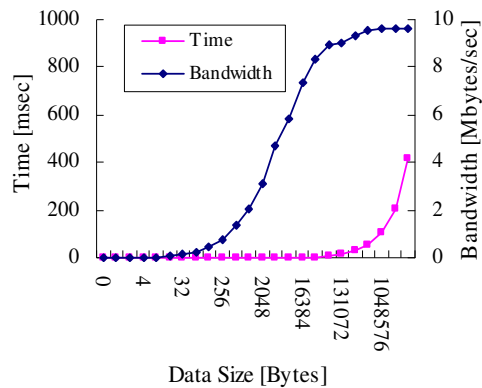
Garis penghubung antar node menandai adanya inter-processor-communication(rsh) melalui fast-ethernet 100 Mbps dimana node#0 juga bekerja sebagai server administrator (NIS,NFS); sedangkan garis antara terminal dan node#0 menandai adanya jalur login dan file transfer (telnet, FTP) untuk keperluan debugging. Disini, computing-node-nya adalah embedded CPU yang cocok untuk membuat prototipe komputer masa depan dan OS-nya menggunakan Linux yang memiliki IPC yang sangat stabil dan open-source sehingga lebih murah.



**GAMBAR 2.2. EMPAT CPU TERHUBUNG HUB**

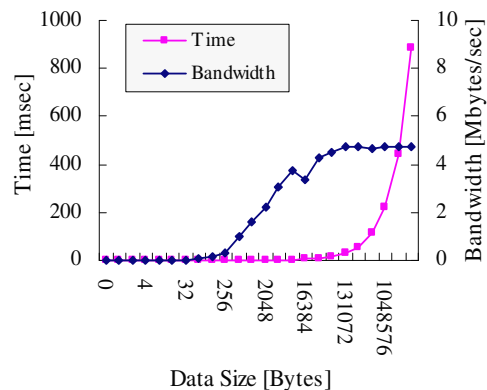
## 2.2. Evaluasi Kinerja

Sistem parallel processing di atas di-evaluasi kinerjanya menggunakan Pallas MPI Benchmark utamanya pada kemampuan komunikasi antar CPU itu untuk melakukan kerja parallel. Dua item yang di-evaluasi adalah ping-pong dan broadcast test. Ping-pong test dilakukan untuk mengukur delay-time pada saat transfer data diantara dua prosesor dan broadcast test adalah untuk mengetahui delay-time terbesar pada saat node#0 mentransfer data kepada node yang lain. Hasil dari kedua test tersebut adalah sebagai berikut.



**GAMBAR 2.3. HASIL PING-PONG TEST**

Pada uji komunikasi ping-pong, diperoleh hasil bahwa kemampuan transfer paling tinggi adalah sekitar 70 Mbps yang mana ini dianggap cukup baik bekerja pada ethernet 100 Mbps.



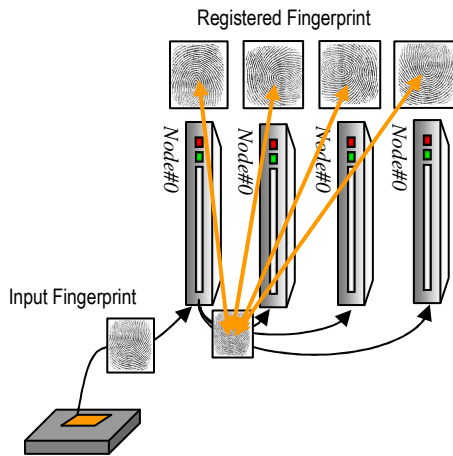
**GAMBAR 2.4. HASIL BROADCAST TEST**

Dari broadcast-test ini diperoleh hasil bahwa kemampuan komunikasi broadcast tertinggi adalah 36 Mbps yang mana hal ini juga dianggap cukup baik untuk sistem yang menggunakan HUB yang biasa. Dari kedua hasil test di atas, parallel processing system yang tersusun dari empat CPU yang dihubungkan melalui fast ethernet 100 Mbps ini siap diaplikasikan untuk menjalankan program verifikasi

biometric sidik jari, pengenalan suara, pengenalan wajah, atau pun yang lain.

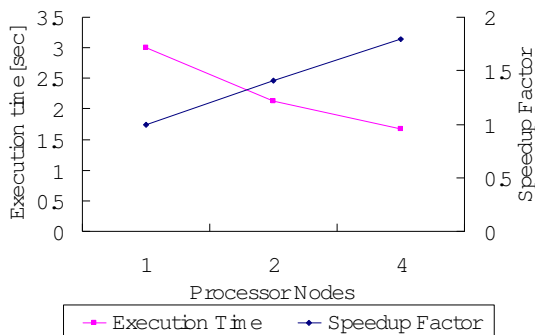
### 3. UJI COBA VERIFIKASI SIDIK JARI

Tim PENS telah melakukan uji aplikasi verifikasi sidik jari menggunakan model distributed/parallel-processing di atas dengan menggunakan algoritma phase-based correlation. Alur logika proses verifikasi sidik jari ditunjukkan pada Gambar 3.1 berikut.



GAMBAR 3.1. VERIFIKASI SIDIK JARI DENGAN 4-CPU

Hasil evaluasi verifikasi sidik jari menggunakan distributed processor 4-node di atas ditunjukkan pada Gambar 3.2 di bawah ini.



GAMBAR 3.2. HASIL UJI COBA VERIFIKASI

Dari uji coba aplikasi di atas diperoleh data bahwa lama waktu verifikasi sidik jari menjadi 2 kali lebih cepat dibandingkan apabila hanya menggunakan satu buah CPU. Dari uji coba ini pula didapat bahwa kinerja embedded system untuk melakukan fungsi verifikasi biometric cukup memuaskan.

Model di atas dibangun menggunakan prosesor yang mempunyai kecepatan setara pentium II yaitu dengan kecepatan 266 MHz dan RAM hanya 64 MB. Untuk

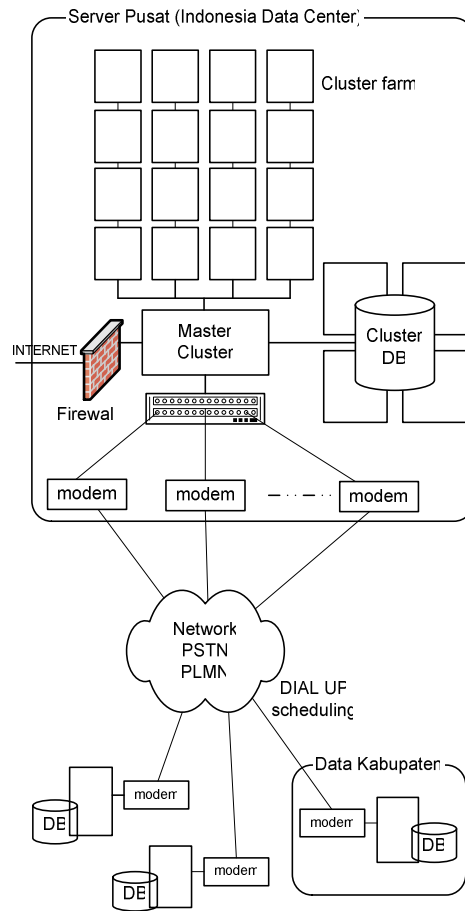
aplikasi yang sebenarnya menggunakan komputer Pentium IV yang berkecepatan lebih dari 3 GHz, diyakini bahwa distributed/parallel processing akan meningkatkan kualitas layanan server utamanya pada kecepatan eksekusinya. Dengan menggunakan CPU yang lebih banyak lagi, bisa dibangun sistem parallel-processing di setiap kota kabupaten dan propinsi sehingga membentuk jaringan verifikasi nasional (makin padat penduduk, makin banyak jumlah CPU).

### 4. KONFIGURASI JARINGAN NASIONAL

Model jaringan nasional untuk sistem verifikasi penduduk yang menjangkau seluruh kabupaten di Indonesia disampaikan dua macam dengan mempertimbangkan bahwa ada kabupaten yang belum mempunyai koneksi internet atau kanal data.

#### 4.1. Untuk Daerah Tanpa Kanal Data

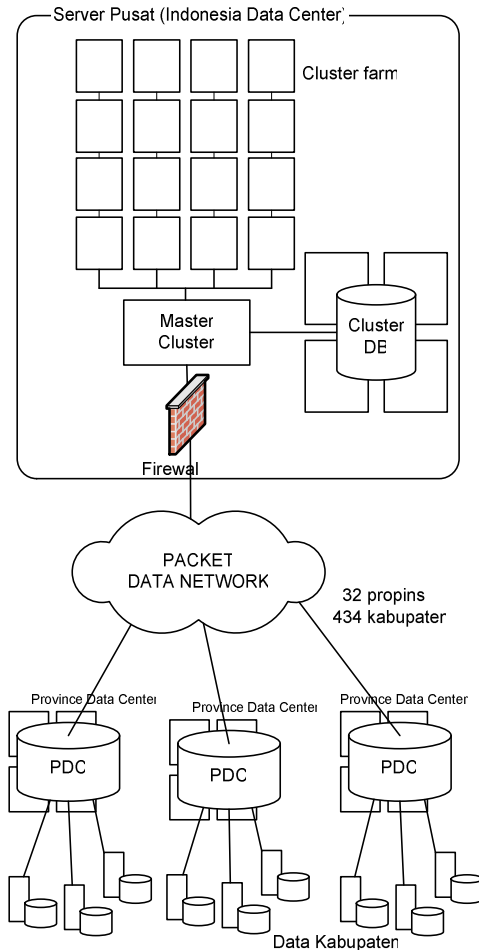
Untuk daerah-daerah yang belum terjangkau internet, sambungan antara KDC dan IDC dilakukan melalui jaringan telepon dengan memanfaatkan modem baik yang analog maupun yang digital (bila ada)



GAMBAR 4.1. KONFIGURASI VIA DIAL-UP

#### 4.2. Untuk Daerah Dengan Kanal Data

Sedangkan untuk daerah-daerah yang sudah lebih maju dan terjangkau kanal data, maka bisa digunakan konfigurasi jaringan seperti pada Gambar 4.2. Kanal data jaringan yang berkapasitas lebih besar, akan memiliki kecepatan proses verifikasi yang lebih tinggi dan lebih memuaskan.



GAMBAR 4.2. KONFIGURASI VIA KANAL DATA

#### 5. PENUTUP

Distributed processing dengan model seperti pada paper ini akan lebih baik lagi kecepatan eksekusinya apabila saluran penghubungnya lebih cepat lagi, misalnya Gbit Ethernet. Untuk aplikasi nasional, distributed processing ini bisa diterapkan di komputer pusat, propinsi, maupun kabupaten, tentunya dengan jumlah CPU yang disesuaikan. Pada konfigurasi nasional, sistem akan berjalan dengan baik apabila Indonesia memiliki backbone jaringan komunikasi yang cepat, misalnya fiber optik yang mampu mem-

bawa data hingga lebih dari Gbps. Namun demikian, jaringan xDSL (yang 2 Mbps ke atas) dan seluler 3G (semisal UTRA) sudah baik dan memuaskan untuk dimanfaatkan (asalkan semua operator membuka layanan-layanan ini).

#### 6. REFERENSI

- [1]. Andrew S Tanenbaum, Maarten van Steen, "Distributed Systems: Principles & Paradigms", Prentice Hall, USA, 2002.
- [2]. T. Aoki, T. Kenji, T. Higuchi, and K. Koji, (2002), "Phase-base image matching and its application to intelligent vision system", Proc. Int. Symp. New Paradigm VLSI Computing, pp. 95-100, Dec. 2002.
- [3]. Fujitsu, (2004), "MBF200, Solid State Fingerprint Sensor", Fujitsu Corp.
- [4]. K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, and T. Higuchi, (2004), "A Fingerprint matching algorithm using phase-only correlation.", IEICE Trans. Fundamentals, vol. E87-A, no. 3, Mar. 2004.
- [5]. A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, (1997), "An identify-authentication system using fingerprints", Proc. IEEE, vol. 85, no.9, pp. 1365-1388, Sept. 1997.
- [6]. Renesas, (2003), "Hitachi SuperH RISC Engine, SH7751 Series", Renesas hardware manual.
- [7]. S. Sukaridhoto, Y. Sasaki, K. Ito, and T. Aoki, (2004), "Development of a Compact Cluster with Embedded CPUs.", Prosiding IES EEPIS-ITS, Desember 2004.
- [8]. S. Sukaridhoto, Y. Sasaki, K. Ito, and T. Aoki, (2004), "Development of a Compact Cluster with Embedded CPUs.", Jurnal EEPIS-ITS, Maret 2005.
- [9]. K. Yaghmour, (2003), "Building Embedded Linux Systems", O Reilly.

