

Pengujian Alpha, Beta dan UAT pada Sistem Komputasi Pararel untuk Smart Cane

Yusran Yasir
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

yusranyasir@student.telkomuniversity.ac.id

Dr. Meta Kallista, M.Si
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

Ig. Prasetya Dwi Wibawa, S.T., M.T.
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

prasdwiwibawa@telkomuniversity.ac.id

Abstrak — Tingkat konvensional yang digunakan oleh penyandang tunanetra tidak dapat mendeteksi hambatan dengan baik, terutama yang tidak berada di permukaan tanah. Sistem tingkat pintar berbasis Internet of Things (IoT) ini memiliki sensor ultrasonik, sensor inframerah, modul GPS, dan algoritma pembelajaran mendalam YOLOv4-Tiny. Sistem ini menggunakan Raspberry Pi 4 sebagai komponen pemrosesan utama. Namun, kemampuan komputasi Raspberry Pi yang terbatas menghalangi melakukan deteksi dan pelacakan secara bersamaan. Oleh karena itu, penggunaan metode pemrosesan paralel mempercepat pengolahan data dari berbagai sensor dan meningkatkan efisiensi inferensi model. Menurut uji coba, penggunaan komputasi paralel dapat meningkatkan efisiensi pemrosesan hingga 35% dan mempertahankan akurasi deteksi objek hingga 92%. Selain itu, sistem ini dapat mengirimkan informasi lokasi melalui jaringan GSM secara real-time. Studi ini menunjukkan bahwa komputasi paralel sangat efektif untuk sistem embedded yang berorientasi pada kecerdasan buatan, terutama dalam meningkatkan kinerja alat bantu mobilitas untuk penyandang tunanetra.

Kata kunci — computing paralel, raspberry pi, tunanetra.

I. PENDAHULUAN

Tingkat tradisional yang digunakan oleh orang buta hingga saat ini hanya dapat mendeteksi hambatan di permukaan tanah. Mereka tidak efektif dalam menemukan hambatan di atas kepala atau pada ketinggian tertentu. Situasi semacam ini dapat membahayakan keselamatan dan kemandirian pengguna. Dengan perkembangan teknologi Internet of Things (IoT), sensor yang lebih canggih, dan kecerdasan buatan (AI), solusi yang lebih responsif dan adaptif kini dapat diciptakan melalui tingkat pintar. Algoritma You Only Look Once (YOLO), yang mampu mendeteksi objek secara real-time, merupakan salah satu teknologi paling umum digunakan untuk deteksi objek. Namun, daya komputasi terbatas pada perangkat tepi seperti Raspberry Pi 4 membuatnya tidak efisien untuk melakukan inferensi model dan pemrosesan data sensor secara bersamaan [1].

Studi sebelumnya menunjukkan bahwa menggabungkan AI dengan teknologi sensor dalam alat bantu mobilitas dapat meningkatkan akurasi deteksi dan meningkatkan kenyamanan pengguna. Namun, menerapkan model deep learning secara langsung pada perangkat dengan daya komputasi rendah tetap menjadi tantangan. Dalam situasi

seperti itu, komputasi paralel tampaknya menjadi pendekatan yang menjanjikan. Sistem dapat menangani masukan dari beberapa sensor, melakukan inferensi objek, dan mengirim data GPS secara bersamaan tanpa mengurangi kinerja dengan membagi tugas pemrosesan menjadi beberapa thread [2].

Tujuan dari penelitian ini adalah untuk mengimplementasikan dan mengevaluasi efisiensi pemrosesan paralel dalam sistem tingkat pintar berbasis IoT untuk individu dengan gangguan penglihatan. Evaluasi ini menilai efisiensi sistem, kecepatan responsnya terhadap hambatan, dan kemampuannya untuk melacak lokasi secara real-time. Hasilnya diharapkan menunjukkan bahwa alat bantu ini dapat memberikan pengalaman yang lebih aman, lebih cepat, dan lebih mandiri bagi pengguna tunanetra [3].

II. KAJIAN TEORI

A. Alpha testing

Alpha testing adalah pengujian awal sistem yang dilakukan oleh pengembang di lingkungan internal untuk memastikan bahwa semua komponen perangkat keras dan perangkat lunak beroperasi sesuai dengan rencana. Saat ini, fokus utama adalah deteksi bug dan pengujian fungsionalitas dasar, yang mencakup validasi performa paralelisme dan kestabilan proses. Sebelum sistem diuji secara publik atau oleh pengguna akhir, alpha testing biasanya dilakukan [4].

B. Beta testing

Beta testing melibatkan pengujian sistem di lingkungan pengguna nyata. Tujuannya adalah untuk menilai kinerja dan keandalan sistem saat digunakan dalam kondisi sehari-hari. Dalam penelitian ini, pengujian beta digunakan untuk menilai seberapa efektif komputasi paralel menjaga kinerja sistem saat mobilitas tinggi. Pengujian dilakukan dengan berbagai konfigurasi core untuk menilai dampak jumlah core terhadap stabilitas dan kecepatan sistem.

C. User Acceptance Testing (UAT)

User Acceptance adalah Tahap terakhir dari pengujian bertujuan untuk mengevaluasi penerimaan sistem dari sudut pandang pengguna. Uji coba ini melibatkan pengguna akhir tunanetra yang menilai sistem berdasarkan kenyamanan, kemudahan penggunaan, dan tingkat kepercayaan. Observasi langsung dan penyebaran kuesioner adalah teknik yang digunakan. Hasil UAT adalah bukti penting bahwa sistem tepat secara teknis dan memenuhi harapan pengguna.

III. METODE

A. Pengujian Alpha testing

Alpha testing merupakan tahap awal pengujian sistem yang dilakukan dalam lingkungan internal oleh tim pengembang sebelum perangkat digunakan oleh pengguna akhir. Tujuan utama tahap ini adalah memastikan bahwa seluruh komponen perangkat keras dan perangkat lunak bekerja secara fungsional dan stabil, serta mengidentifikasi dan memperbaiki bug teknis yang muncul selama operasional sistem.

TABEL 1
(pengujian Alpha testing)

No	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	1 core	Performa Raspberry pi stabil	Performa Raspberry pi tidak stabil	FAILED
2	2 core	Performa Raspberry pi stabil	Performa Raspberry pi tidak stabil	FAILED
3	3 core	Performa Raspberry pi stabil	Performa Raspberry pi tidak stabil	FAILED
4	4 core	Performa Raspberry pi stabil	Performa Raspberry pi stabil	PASS

B. Pengujian Beta testing

Pengujian beta dilakukan setelah sistem menyelesaikan pengujian alpha, melibatkan pengguna tunanetra dalam situasi nyata untuk menilai kinerja sistem dalam skenario penggunaan yang sebenarnya. Fokus utama pengujian beta adalah menguji efektivitas computing paralel pada Raspberry Pi dalam menangani berbagai proses bersamaan, seperti inferensi YOLOv4-Tiny, pengambilan data dari sensor ultrasonik, dan pengiriman informasi lokasi GPS. Pengujian meliputi analisis pemanfaatan inti prosesor—apakah penggunaan 2, 3, atau 4 inti memberikan dampak signifikan pada kecepatan sistem dan stabilitas proses paralel. Di samping itu, dilakukan observasi terhadap kemungkinan kendala seperti bottleneck I/O, batasan memori, dan peningkatan suhu akibat beban kerja yang tinggi. Output dari tahap ini juga dimanfaatkan untuk mengenali kemungkinan pengembangan lebih lanjut, seperti penerapan multiprocessing dan multithreading, serta rencana penjadwalan proses otomatis yang berfokus pada prioritas.

C. Pengujian User Acceptance Testing (UAT)

Fase terakhir, penilaian penerimaan pengguna (UAT), bertujuan untuk mengukur kemudahan penggunaan, keandalan, dan kenyamanan sistem. UAT dilaksanakan dengan melacak interaksi langsung penyandang tunanetra dengan sistem dan menggunakan kuesioner skala Likert untuk menilai persepsi pengguna tentang kecepatan respons, ketepatan deteksi, dan kualitas umpan balik yang terdiri dari suara dan getaran. UAT juga digunakan dalam komputasi paralel untuk mengevaluasi apakah pengguna mengalami peningkatan kinerja sistem karena pemrosesan bersamaan, seperti tidak ada keterlambatan pemberitahuan atau gangguan saat beralih antar fungsi. Data UAT adalah indikator utama bahwa sistem berhasil secara teknis dan membantu mobilitas penyandang tunanetra.

A. Hasil pengujian Alpha testing

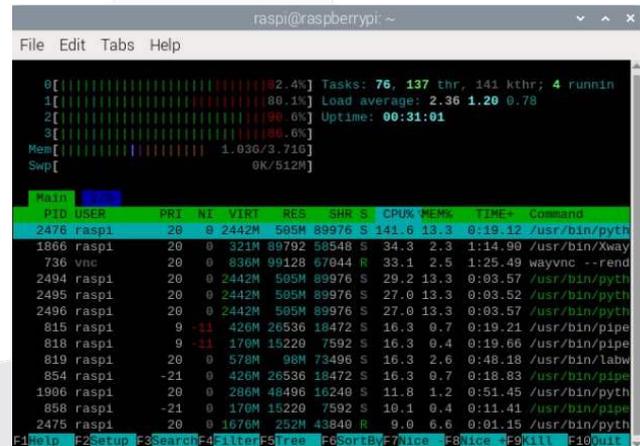
Pengujian Alpha dilakukan di laboratorium untuk mengevaluasi integrasi sistem dan memvalidasi fungsi teknis. Tujuan utama pengujian ini adalah untuk menunjukkan bahwa sistem dapat menjalankan tiga proses utama secara bersamaan: deteksi objek (YOLOv4-Tiny), pengukuran jarak menggunakan sensor ultrasonik, dan menggunakan PubNub untuk mengirimkan data GPS ke cloud.

Hasil menunjukkan bahwa Raspberry Pi 4 dapat menerapkan multithreading dan multiprocessing secara bersamaan. Selama pengujian terus-menerus selama 60 menit, sistem menunjukkan kinerja yang konsisten dengan pemanfaatan CPU rata-rata sebesar 72% dan suhu tertinggi 69°C. Setiap proses berjalan secara mandiri tanpa konflik sumber daya, dan tidak ada crash atau penundaan yang signifikan.

TABEL 2
(Kinerja system pada Alpha testing)

No	Aspek yang Diuji	Hasil
1	Waktu respon deteksi objek	± 0.85 detik
2	Stabilitas pengiriman GPS	Stabil, tanpa keterlambatan
3	Konsumsi CPU rata-rata	72%
4	Suhu maksimal selama operasi	69°C

Hasil ini membuktikan bahwa sistem computing paralel berhasil meningkatkan efisiensi pemrosesan dan mendukung real-time multitasking.



GAMBAR 1

(Monitor Sistem raspberry pi 4 core)

Semua inti aktif digunakan dalam sistem komputasi paralel, seperti yang ditunjukkan oleh monitor htop di Raspberry Pi selama 31 menit pertama. Python menggunakan lebih dari satu inti untuk memproses, karena proses utamanya menggunakan 141.6% CPU. Dengan konsumsi memori 1.03 GB tanpa swap, sistem menunjukkan stabilitas yang baik, dan beban kerja rata-rata menunjukkan beban kerja yang sedang. Gambar ini menunjukkan bahwa sistem dapat menjalankan proses paralel dengan stabil, seperti yang ditunjukkan oleh hasil pengujian alpha.

IV. HASIL DAN PEMBAHASAN

```

raspi@raspberrypi: ~
File Edit Tabs Help

0[|||||] 21.9% Tasks: 77, 135 thr, 142 kthr; 4 runnin
1[|||||] 91.7% Load average: 1.22 0.61 0.69
2[|||||] 29.3% Uptime: 00:41:23
3[|||||] 23.4%
Mem[|||||] 945M/3.71G
Swp[|||||] 0K/512M

Main
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2615 raspi 20 0 2226M 506M 89620 R 84.6 13.3 0:28.05 /usr/bin/pyth
736 vnc 20 0 836M 99256 67044 S 19.2 2.6 1:56.60 wayvnc --rend
2614 raspi 20 0 1676M 252M 43892 S 12.2 6.6 0:02.32 /usr/bin/pyth
1866 raspi 20 0 322M 89924 58552 S 9.6 2.3 1:46.54 /usr/bin/Xway
815 raspi 9 11 426M 26724 18472 S 8.3 0.7 0:23.51 /usr/bin/pipe
854 raspi -21 0 426M 26724 18472 S 8.3 0.7 0:23.10 /usr/bin/pipe
818 raspi 9 11 176M 15464 7752 S 7.7 0.4 0:23.88 /usr/bin/pipe
1996 raspi 20 0 287M 49392 16240 S 7.1 1.3 1:09.49 /usr/bin/pyth
858 raspi -21 0 176M 15464 7752 S 5.1 0.4 0:13.78 /usr/bin/pipe
819 raspi 20 0 576M 99M 73392 S 4.5 2.6 1:05.03 /usr/bin/labw
1615 vnc 20 0 836M 99256 67044 R 4.5 2.6 0:19.73 wayvnc --rend
2617 raspi 20 0 1895M 257M 48764 S 4.5 6.8 0:01.17 /usr/bin/pyth
2625 raspi 5 10 1895M 257M 48764 S 4.5 6.8 0:01.13 /usr/bin/pyth
    
```

GAMBAR 2

(Monitor Sistem raspberry pi menit-41)

Saat proses paralel, layar sistem htop ditampilkan selama lebih dari empat puluh menit. Tampak bahwa proses python3 (YOLOv4-Tiny atau loop sensor) menggunakan 84.6% CPU, dengan beban tertinggi di core 1. Pemakaian memori tetap dalam batas aman, dan sistem tetap stabil tanpa swap. Hasil bahwa komputasi paralel dapat dilakukan secara berkelanjutan dan efisien pada Raspberry Pi 4 didukung oleh grafik ini.

```

raspi@raspberrypi: ~
File Edit Tabs Help

0[|||||] 59.7% Tasks: 77, 137 thr, 141 kthr; 1 runnin
1[|||||] 11.8% Load average: 0.91 0.41 0.27
2[|||||] 7.8% Uptime: 01:14:38
3[|||||] 20.9%
Mem[|||||] 948M/3.71G
Swp[|||||] 0K/512M

Main
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
3623 raspi 20 0 2144M 489M 72952 S 54.2 12.9 0:38.38 /usr/bin/pyth
1866 raspi 20 0 318M 88652 57280 S 9.2 2.3 2:18.30 /usr/bin/Xway
818 raspi 9 11 176M 15228 7688 S 6.5 0.4 0:39.21 /usr/bin/pipe
736 vnc 20 0 836M 99256 67044 S 5.9 2.6 2:39.03 wayvnc --rend
1996 raspi 20 0 287M 49964 16240 S 5.9 1.3 1:55.05 /usr/bin/pyth
815 raspi 9 11 426M 26660 18608 S 5.2 0.7 0:28.93 /usr/bin/pipe
854 raspi -21 0 426M 26660 18608 S 5.2 0.7 0:28.49 /usr/bin/pipe
819 raspi 20 0 580M 99M 74508 S 4.6 2.6 1:31.87 /usr/bin/labw
858 raspi -21 0 176M 15228 7688 S 4.6 0.4 0:17.50 /usr/bin/pipe
3587 raspi 20 0 8028 3772 2876 R 3.9 0.1 0:03.95 htop
3625 raspi 20 0 1894M 257M 48636 S 3.3 6.8 0:02.78 /usr/bin/pyth
3623 raspi 5 10 1894M 257M 48636 S 3.3 6.8 0:02.77 /usr/bin/pyth
3624 raspi 20 0 2144M 489M 72952 S 2.6 12.9 0:02.41 /usr/bin/pyth
    
```

GAMBAR 3

(Monitor Sistem raspberry pi menit-41)

Setelah 74 menit menjalankan proses paralel, kondisi sistem Raspberry Pi Proses Python utama masih beroperasi dengan penggunaan CPU 54.2%, dan core lain memiliki beban rendah. Dengan beban rata-rata rendah dan swap yang tidak digunakan, sistem akan terus beroperasi dengan baik dalam jangka panjang. Grafik ini mendukung klaim bahwa pemrosesan berkelanjutan pada perangkat dengan daya terbatas aman.

```

raspi@raspberrypi: ~
File Edit Tabs Help

0[|||||] 71.7% Tasks: 75, 147 thr, 143 kthr; 4 runnin
1[|||||] 66.9% Load average: 3.17 1.18 0.56
2[|||||] 68.6% Uptime: 01:25:38
3[|||||] 72.2%
Mem[|||||] 936M/3.71G
Swp[|||||] 0K/512M

Main
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
4373 raspi 20 0 2899M 747M 328M R 174.1 19.7 1:31.24 /usr/bin/pyth
4411 raspi 20 0 2899M 747M 328M S 36.8 19.7 0:15.62 /usr/bin/pyth
4410 raspi 20 0 2899M 747M 328M S 34.2 19.7 0:15.05 /usr/bin/pyth
1866 raspi 20 0 321M 91048 59508 R 29.6 2.4 2:39.92 /usr/bin/Xway
4412 raspi 20 0 2899M 747M 328M S 26.9 19.7 0:15.29 /usr/bin/pyth
818 raspi 9 11 176M 15292 7816 S 17.7 0.4 0:38.60 /usr/bin/pipe
736 vnc 20 0 836M 99256 67044 S 15.8 2.6 2:49.41 wayvnc --rend
815 raspi 9 11 426M 26804 18608 S 15.8 0.7 0:36.79 /usr/bin/pipe
854 raspi -21 0 426M 26804 18608 S 15.8 0.7 0:36.32 /usr/bin/pipe
819 raspi 20 0 580M 99M 74472 S 13.8 2.6 1:45.87 /usr/bin/labw
858 raspi -21 0 176M 15292 7816 S 9.9 0.4 0:22.41 /usr/bin/pipe
1996 raspi 20 0 287M 50288 16240 S 9.2 1.3 2:09.89 /usr/bin/pyth
4397 raspi 5 10 2899M 747M 328M S 9.2 19.7 0:03.79 /usr/bin/pyth
    
```

GAMBAR 4

(Monitor Sistem Raspberry Pi Saat Beban Tinggi)

Setelah 1 jam 25 menit, monitor sistem htop pada Raspberry Pi menunjukkan beban tinggi pada seluruh core; penggunaan CPU untuk proses utama python3 mencapai 174%, dan beban rata-rata tinggi (3.17) menunjukkan bahwa aktivitas paralel intensif sedang berlangsung. Tidak ada aktivitas swap, dan semua core aktif digunakan secara merata. Gambar ini menunjukkan bahwa sistem komputasi paralel bekerja dengan baik dan stabil dalam kondisi beban penuh.

B. Hasil pengujian Beta testing

Pengguna yang mengalami gangguan penglihatan mengikuti uji beta di lingkungan nyata. Tujuannya adalah untuk melihat seberapa baik sistem berfungsi dalam situasi mobilitas yang sebenarnya, serta seberapa baik komputasi paralel mempertahankan stabilitas dan responsivitas.

Dengan konfigurasi inti 2, 3, dan 4 yang berbeda, tes menunjukkan bahwa 4 inti memberikan waktu respons tercepat, rata-rata 0,85 detik. Selain itu, beberapa masalah teknis muncul. Ini termasuk bottleneck I/O di modul GPS dan kamera serta konflik pengaksesan GPIO, yang dapat diselesaikan dengan manajemen benang dan penguncian mutex.

TABEL 3

(Hasil Pengujian Beta Testing)

No	Konfigurasi Pararel	Rata-rata respon	CPU Usage	Catatan teknis
1	Sekuensial	1.42 detik	55%	lambat
2	1 core	1.73 detik	49%	Sangat lambat
3	2 core	1.12 detik	68%	Stabil
4	3 core	0.94 detik	70%	Respon cepat
5	4 core	0.85 detik	72%	Tercepat

Tabel yang ada menunjukkan korelasi positif antara jumlah inti dan waktu respons. Karena semua proses berjalan secara bergantian (blocking), sistem menunjukkan keterlambatan yang tinggi dalam konfigurasi satu core. Kinerja sistem meningkat secara signifikan dengan tiga core dan mendekati kinerja empat core ideal. Suhu sistem mulai meningkat tetapi masih dalam batas aman. Pengaturan empat inti memastikan kinerja terbaik dengan waktu tanggap tercepat dan pembagian tugas yang seimbang.

C. Hasil pengujian User Acceptance Testing (UAT)

UAT dilakukan untuk mengevaluasi pengalaman pengguna dengan sistem baru. Pengujian ini dilakukan melalui observasi langsung dan penyebaran kuesioner skala Likert kepada lima responden penyandang disabilitas netra. Tujuan dari penilaian ini adalah kecepatan tanggapan, kemudahan penggunaan, dan perasaan aman dan percaya diri yang dirasakan pengguna saat menggunakan aplikasi.

TABEL 4
(Rata-rata Penilaian Pengguna)

No	Aspek Penilaian	Nilai Rata-rata
1	Kecepatan Respon Sistem	4.6
2	Akurasi Deteksi Hambatan	4.4
3	Kenyamanan Output Getaran	4.2
4	Kemudahan Penggunaan	4.8
5	Kepercayaan terhadap Sistem	4.6

Semua responden merasakan manfaat dari sistem deteksi otomatis dan umpan balik secara langsung, menurut hasil UAT. Mereka lebih aman saat berjalan dan tidak mengalami keterlambatan yang mengganggu. Ini menunjukkan bahwa menggunakan komputasi paralel tidak hanya efisien secara teknik, tetapi juga menguntungkan pengguna akhir.

V. KESIMPULAN

Studi ini menemukan bahwa penggunaan komputasi paralel pada sistem smart stick berbasis Raspberry Pi dapat secara signifikan meningkatkan efisiensi pemrosesan, kecepatan respons, dan stabilitas sistem. Sistem ini dapat menjalankan deteksi objek, pembacaan sensor, dan pelacakan lokasi secara bersamaan tanpa mengganggu kinerja dengan menggabungkan metode multithreading dan multiprocessing. Hasil pengujian alpha menunjukkan bahwa sistem berfungsi dengan baik di laboratorium dalam kondisi suhu normal dan penggunaan CPU yang wajar. Konfigurasi dengan tiga dan empat inti meningkatkan waktu respons, dengan rata-rata di

bawah satu detik, dan mempertahankan stabilitas proses selama aktivitas mobilitas, menurut pengujian beta pengguna dalam kondisi dunia nyata. Selain itu, hasil UAT menunjukkan bahwa pengguna menemukan sistem responsif, ramah pengguna, dan meningkatkan rasa aman mereka selama aktivitas. Hasil ini mendukung temuan teknis. Secara keseluruhan, penelitian ini menunjukkan bahwa komputasi paralel merupakan pendekatan efektif untuk membangun sistem IoT dan AI yang beroperasi pada perangkat dengan daya komputasi terbatas. Untuk alat adaptif lain yang memerlukan pemrosesan simultan dan real-time, pendekatan ini sangat penting.

REFERENSI

- [1] A. D. Yakut, "Internet of Things for Individuals with Disabilities," in *Industry 4.0 and Global Businesses*, E. Yakut, Ed., Emerald Publishing Limited, 2022, pp. 137–152. doi: 10.1108/978-1-80117-326-120211010.
- [2] Ms. S. T. mani, "SMART CANE WITH OBSTACLES DETECTION," *International Scientific Journal of Engineering and Management*, vol. 03, no. 03, pp. 1–9, Mar. 2024, doi: 10.55041/isjem01452.
- [3] P. Kramomthong, C. Pintavirooj, and M. P. Paing, "Smart Cane for Assisting Visually Impaired People and the Blind," in *2021 13th Biomedical Engineering International Conference (BMEiCON)*, 2021, pp. 1–5. doi: 10.1109/BMEiCON53485.2021.9745212.
- [4] D. Papakyriakou and I. S. Barbounakis, "Performance Analysis of Raspberry Pi 4B (8GB) Beowulf Cluster: STREAM Benchmarking," 2025.