

Sistem Rekomendasi Lokasi Pencacahan Berbasis Konteks

Oleh: **23215131- Aris Prawisudatama**

Pembimbing: **I Gusti Bagus Baskara Nugraha**

Latar Belakang

Pengumpulan data di Badan Pusat Statistik (BPS):

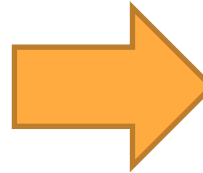
✓ Data Primer

Wawancara:

- Individu
- Rumah tangga
- Perusahaan

✓ Data Sekunder

Kompilasi data yang
dikumpulkan pihak
lain



Bagaimana mengalokasikan
petugas?

- ✓ Domisili dekat dengan lokasi
- ✓ Total biaya dan waktu rendah

**Bagaimana Cara
Alokasinya?**



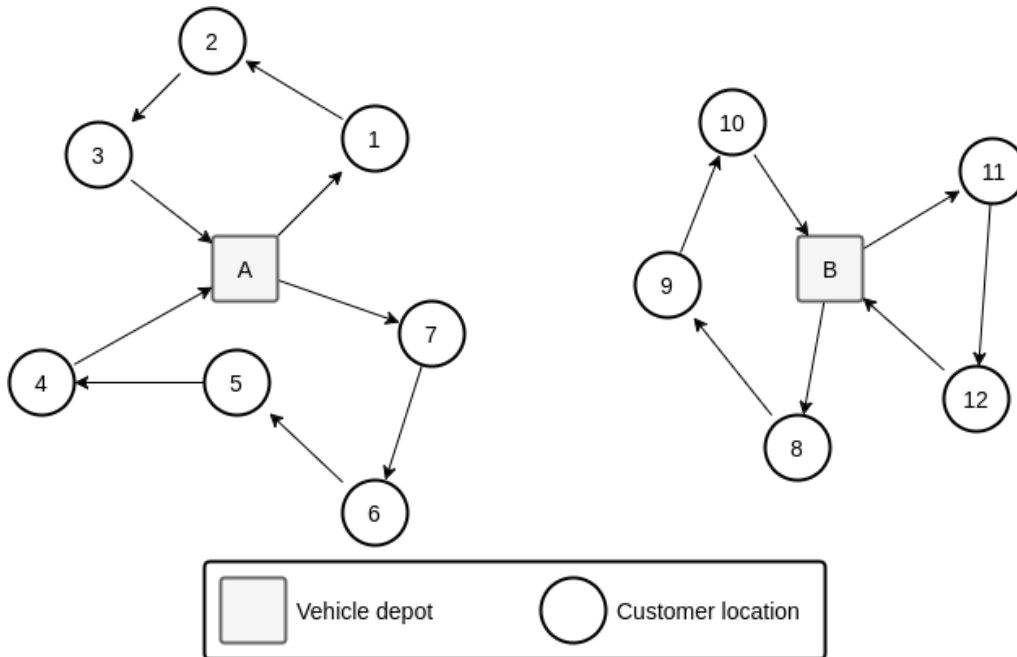
MDVRP

Pendahuluan

Latar Belakang

Apakah **MDVRP** ?

- ✓ Penentuan rute terbaik yang harus ditempuh oleh beberapa kendaraan untuk mengirimkan barang kepada konsumen.
- ✓ Terdapat lebih dari satu depot.



Aturan dalam **MDVRP** ?

- ✓ Setiap konsumen hanya dikunjungi sekali oleh satu kendaraan.
- ✓ Setiap kendaraan memulai dan mengakhiri perjalanan pada depot yang sama.
- ✓ *Demand* dari seluruh konsumen yang dikunjungi tidak melebihi kapasitas kendaraan.

Preliminary Analysis

- ✓ Lama perjalanan ➡ Google Direction API
- ✓ Lama pencacahan ➡ Estimasi Flat?

Pendahuluan

Preliminary Analysis

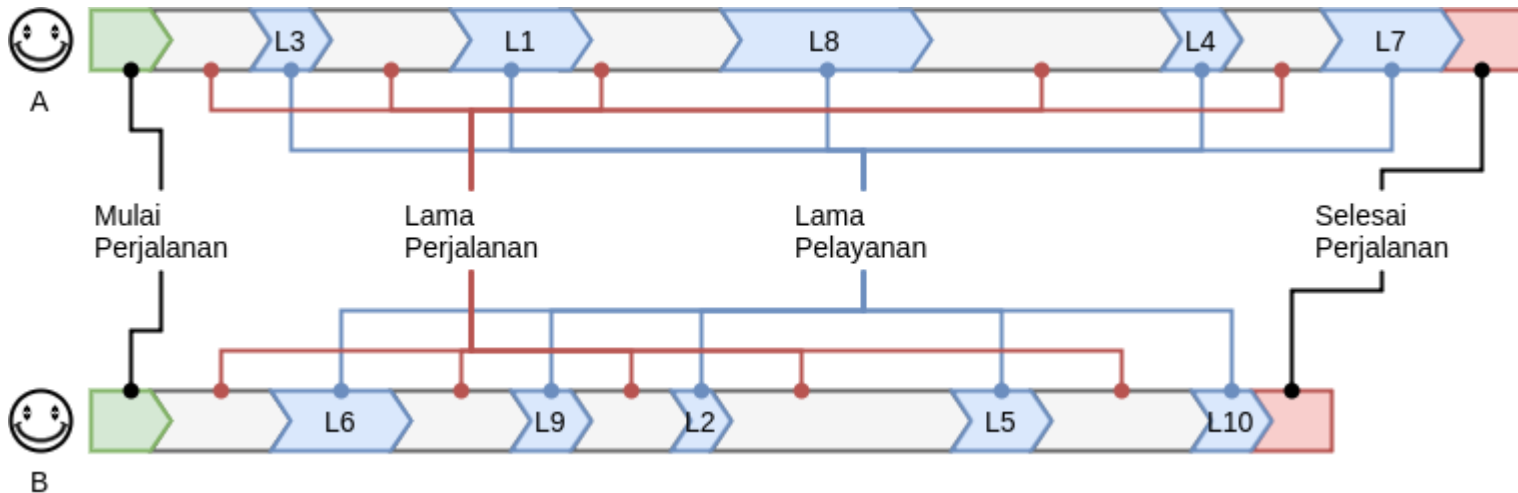
Lama pencacahan sebenarnya



Tidak dapat diestimasi

- ✓ Banyaknya anggota rumah tangga
- ✓ Jarak rumah tangga dalam sebuah Blok Sensus
- ✓ Kelancaran wawancara

Berpotensi terjadi ketidakmerataan waktu penyelesaian

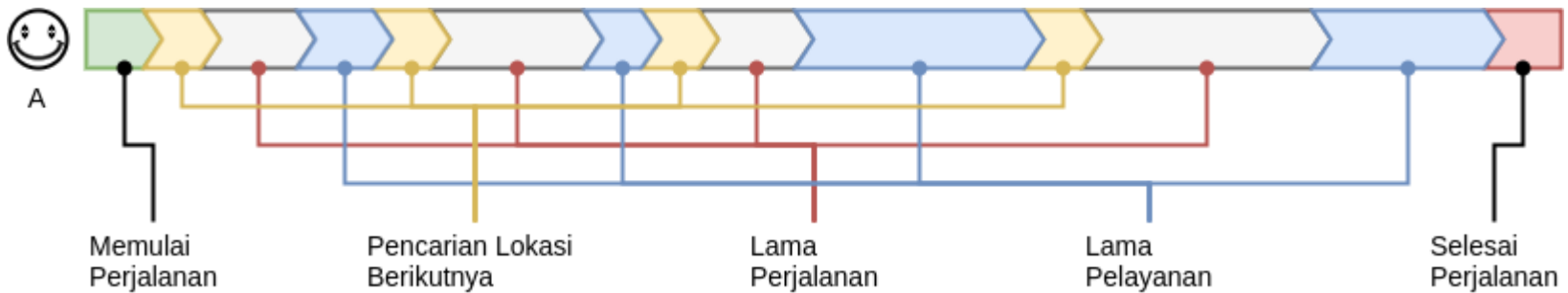


Pendahuluan

Hipotesis

- ✓ Solusi dicari secara bertahap. Lokasi berikutnya ditentukan setelah selesai melakukan pencacahan pada satu lokasi.

➡ **Markov Process**



- ✓ Melibatkan **konteks** dari seluruh pencacah.

➡ **Tersentral**



Publish/Subscribe

Pendahuluan

Rumusan Masalah

- ✓ Bagaimana menentukan **konteks dari petugas pencacahan** yang sesuai digunakan pada permasalahan rekomendasi lokasi pencacahan?
- ✓ Bagaimana menyusun **algoritma pencarian rekomendasi** lokasi pencacahan dengan memanfaatkan konteks dari petugas pencacahan?
- ✓ Bagaimana menyusun **algoritma conflict resolution** agar sistem tidak merekomendasikan lokasi yang sama pada dua atau lebih pencacah?
- ✓ Bagaimana menentukan **mekanisme real-time** yang sesuai digunakan pada permasalahan rekomendasi lokasi pencacahan?

Pendahuluan

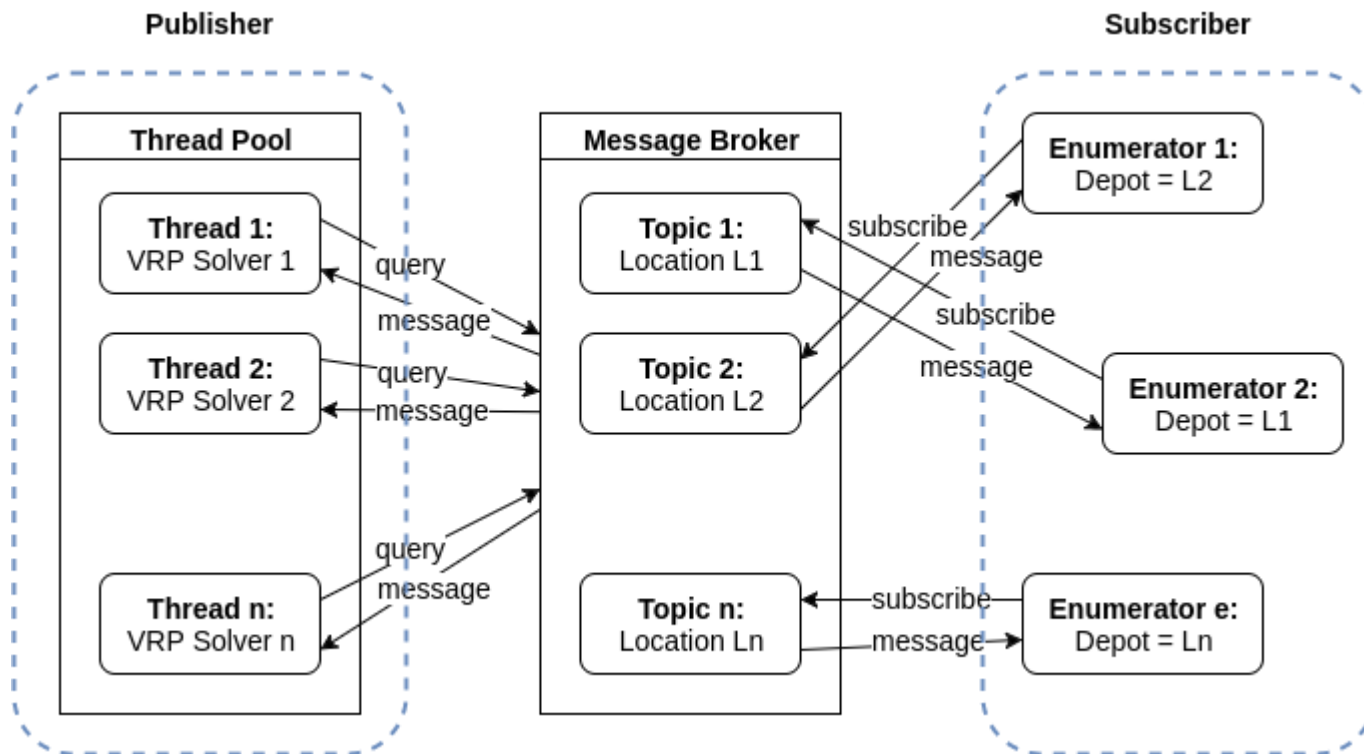
Tujuan Penelitian

- ✓ Menentukan **konteks dari petugas pencacahan** yang sesuai digunakan pada permasalahan rekomendasi lokasi pencacahan.
- ✓ Menyusun **algoritma pencarian rekomendasi** lokasi pencacahan dengan memanfaatkan konteks dari petugas pencacahan.
- ✓ Menyusun **algoritma conflict resolution** agar sistem tidak merekomendasikan lokasi yang sama pada dua atau lebih pencacah.
- ✓ Menentukan **mekanisme real-time** yang sesuai digunakan pada permasalahan rekomendasi lokasi pencacahan.

Perancangan

Garis Besar Sistem

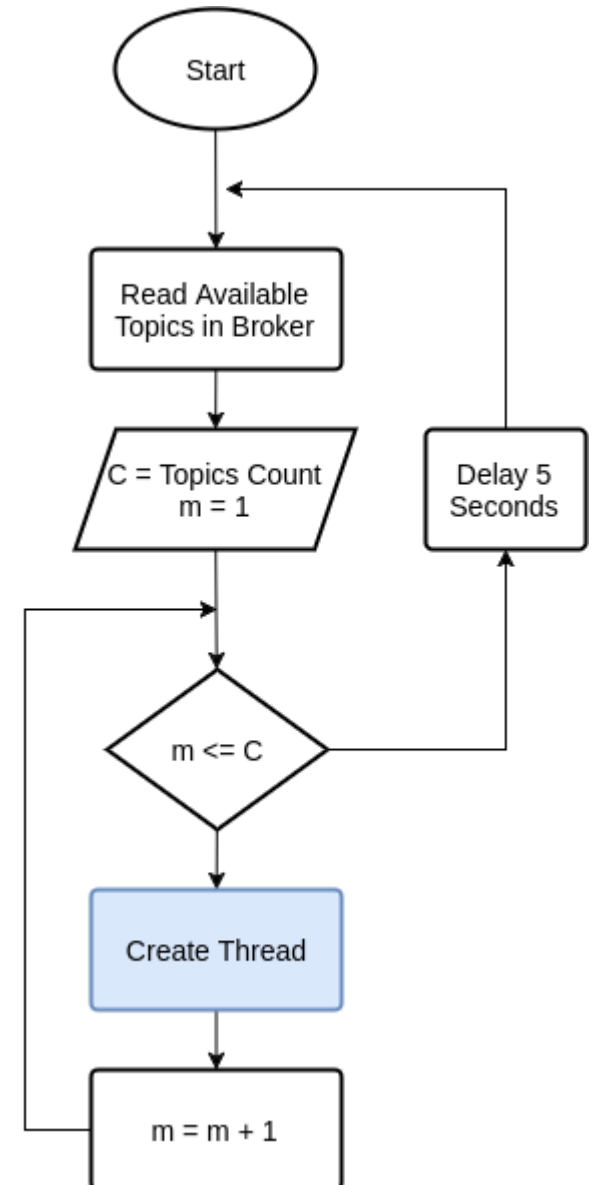
- ✓ Dirancang berdasarkan arsitektur Publish/Subscribe
- ✓ Request dari Subscriber dilakukan dengan menggunakan topik
- ✓ Topik yang digunakan adalah '**Konteks dari Pencacah**' ➡ **Lokasi Terkini Pencacah**



Publisher Rekomendasi

TOPIC WATCHER

- ✓ Melakukan pengecekan request/topik yang di-subscribe melalui message broker secara berkala.
- ✓ Setiap request/topik yang diterima akan disiapkan sebuah thread yang dilengkapi dengan VRP Solver.
- ✓ Seluruh thread akan ditampung pada threadpool.
- ✓ Ketika seluruh lokasi telah dialokasikan, jumlah elemen pada threadpool akan sama dengan jumlah seluruh lokasi pencacahan.



Publisher Rekomendasi

PUBLISHER

- ✓ Setiap thread yang terdapat pada threadpool dieksekusi secara berurutan menurut kedatangan
- ✓ Setiap sesi hanya terdapat satu thread yang dieksekusi



Mencegah sebuah lokasi direkomendasikan kepada lebih dari satu pencacah

Perancangan

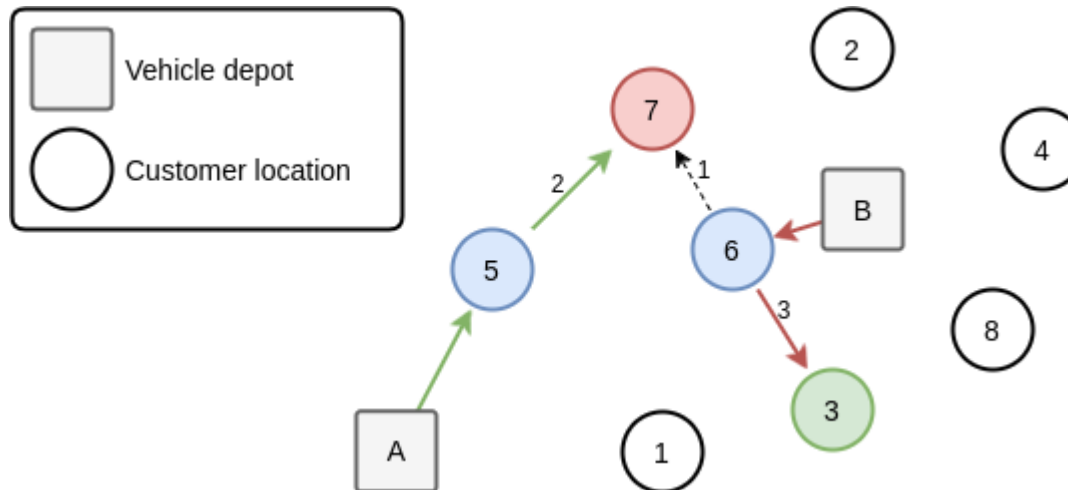
Publisher Rekomendasi

PUBLISHER

- ✓ Setiap eksekusi dari thread akan melibatkan seluruh pencacah dan seluruh lokasi yang belum dialokasikan.



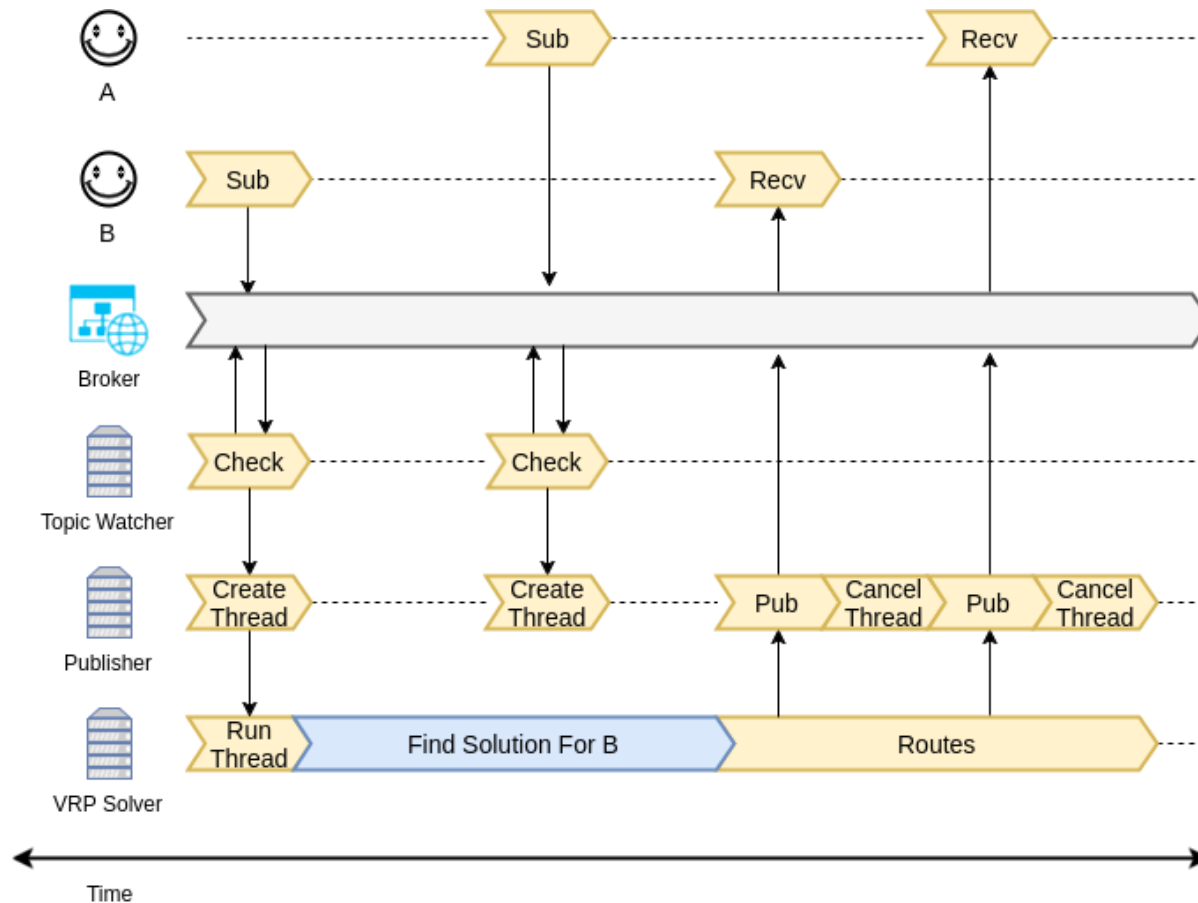
Memastikan solusi terbaik yang diperoleh adalah solusi terbaik secara global



Publisher Rekomendasi

PUBLISHER

- ✓ Menghasilkan rute sejumlah antara 1 s.d M (jumlah pencacah yang diikuti sertakan dalam pencarian).



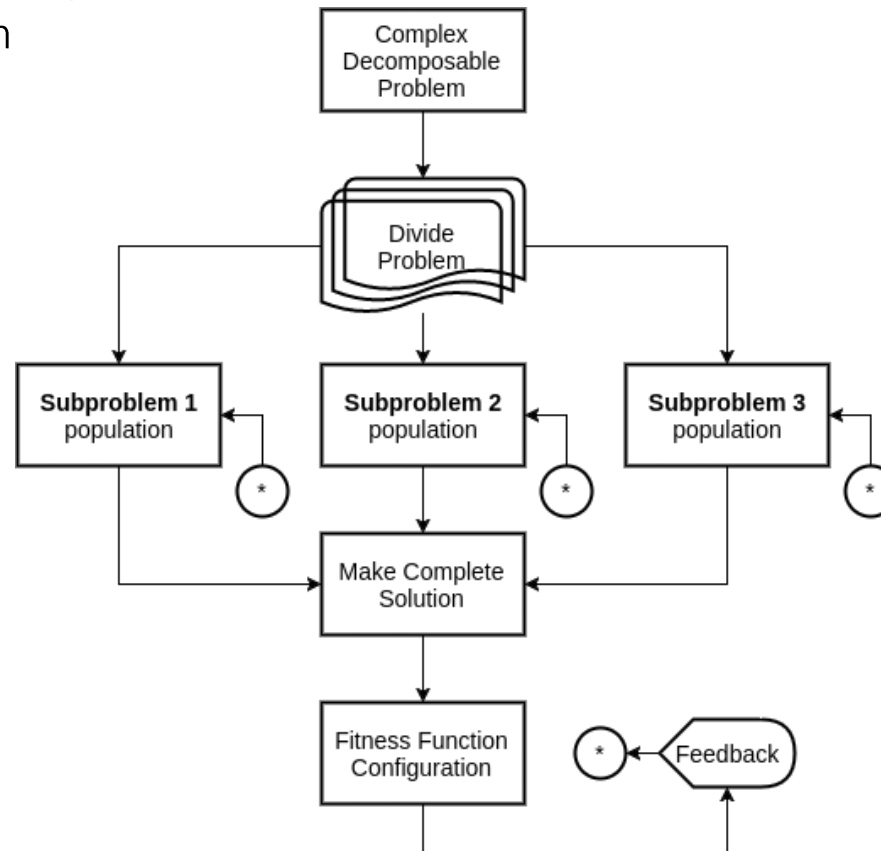
Publisher Rekomendasi

VRP SOLVER

- ✓ Menggunakan algoritma *Cooperative Coevolution Algorithms (CoEAs)*
- ✓ Dijalankan sampai menghasilkan rute yang konvergen



- ✓ Total biaya minimum
- ✓ Waktu pemrosesan relatif singkat dibanding algoritma lain



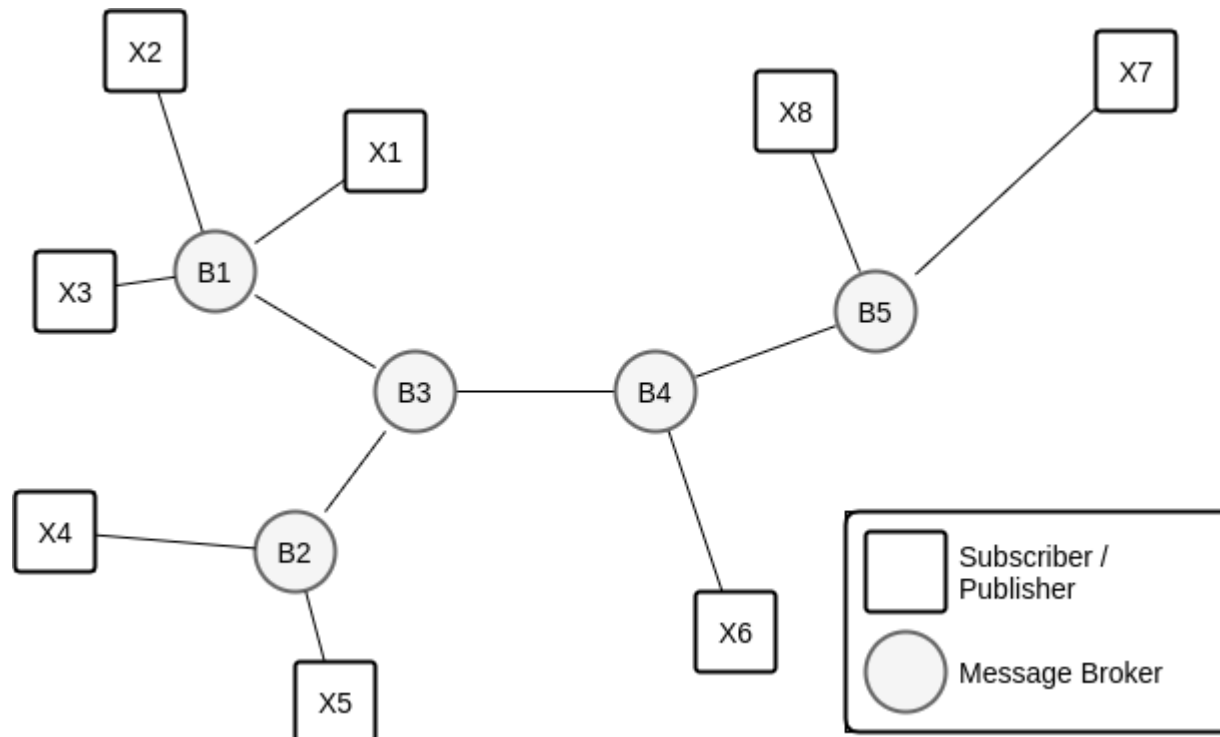
Perancangan

Message Broker

- ✓ Bertanggungjawab dalam meneruskan pesan dari *publisher* ke *subscriber* dan sebaliknya.
- ✓ Menggunakan lebih dari satu broker yang membentuk sebuah *cluster*.



Untuk mengakomodir wilayah pencacahan yang terdistribusi secara geografis



Implementasi

Implementasi

VRP SOLVER

Diimplementasikan dalam bahasa pemrograman C++ pada lingkungan:

- Elementary OS Loki (Berbasis Ubuntu 16.04)
- c++ (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
- Asus TP300L, Quad-Core Intel® Core™ i3-4030U CPU @ 1.90GHz, 3,7 GiB DDRIII, 256GB SSD



Lebih efisien dalam penggunaan CPU dan memory

PUBLISHER

Diimplementasikan dalam bahasa pemrograman Python pada lingkungan:

- Elementary OS Loki (Berbasis Ubuntu 16.04)
- Python 2.7.12
- Asus TP300L, Quad-Core Intel® Core™ i3-4030U CPU @ 1.90GHz, 3,7 GiB DDRIII, 256GB SSD

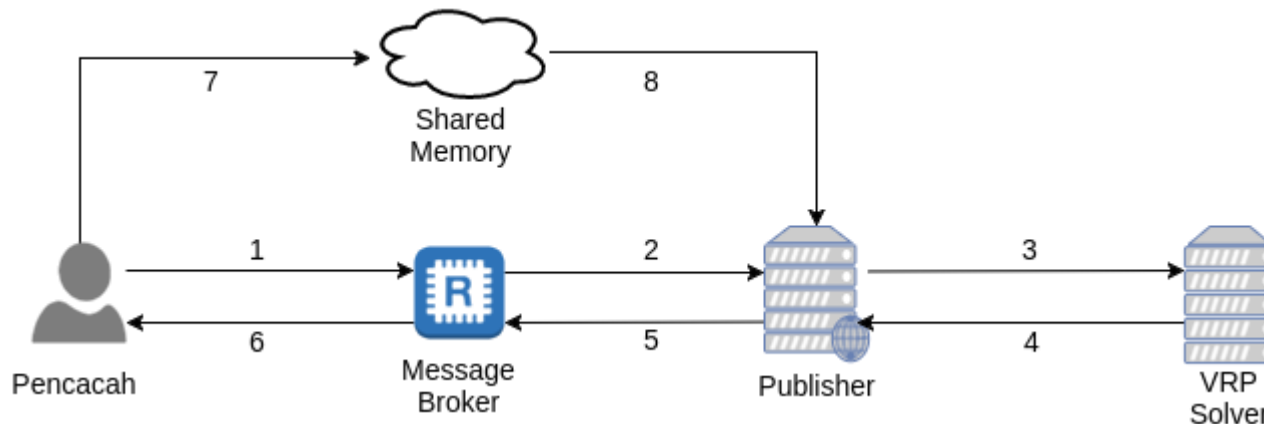


- ✓ **Code simpel**
- ✓ **Mudah digunakan untuk membuat prototipe**

Pengujian

Interaksi Antar Komponen

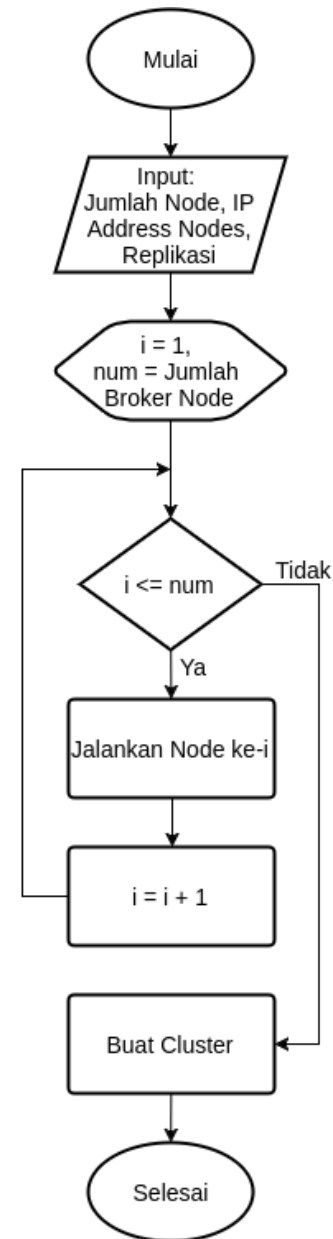
1. Pencacah mengirimkan permintaan lokasi yang akan dicacah melalui message broker
2. TopicWatcher pada publisher akan memantau request yang diterima secara periodik dan menyiapkan thread untuk setiap request
3. VRP Solver akan mengeksekusi setiap thread secara berurutan
4. Solusi yang diperoleh merupakan sejumlah rute
5. Publisher memilah rute dan mengirimkannya kepada tiap-tiap petugas yang bersesuaian melalui message broker
6. Message broker meneruskan pesan kepada tiap-tiap petugas dengan topik yang sesuai
7. Ketika pencacah sampai lokasi, lokasi terkini dari pencacah disimpan pada shared memory
8. Lokasi tersebut akan digunakan sebagai depot pada kalkulasi berikutnya



Pengujian

Message Broker Setup

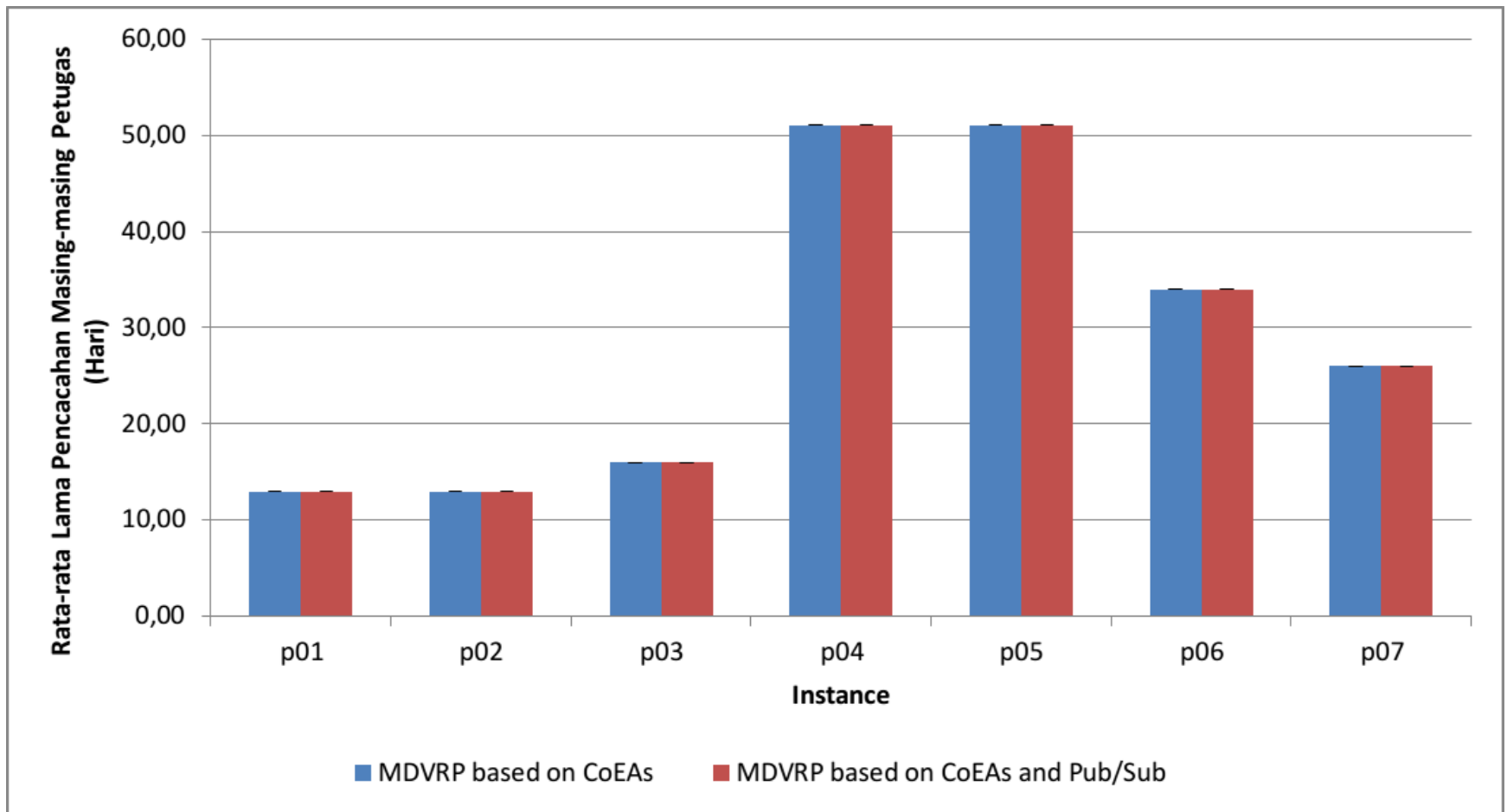
- ❑ Menggunakan software Redis
- ❑ Konfigurasi:
 - ✓ Menggunakan fitur Redis Cluster
 - ✓ Menggunakan 6 Redis Nodes
 - 3 Nodes sebagai Master
 - 3 Nodes sebagai Slave



Pengujian

Hasil Pengujian Kondisi Normal (Cordeau)

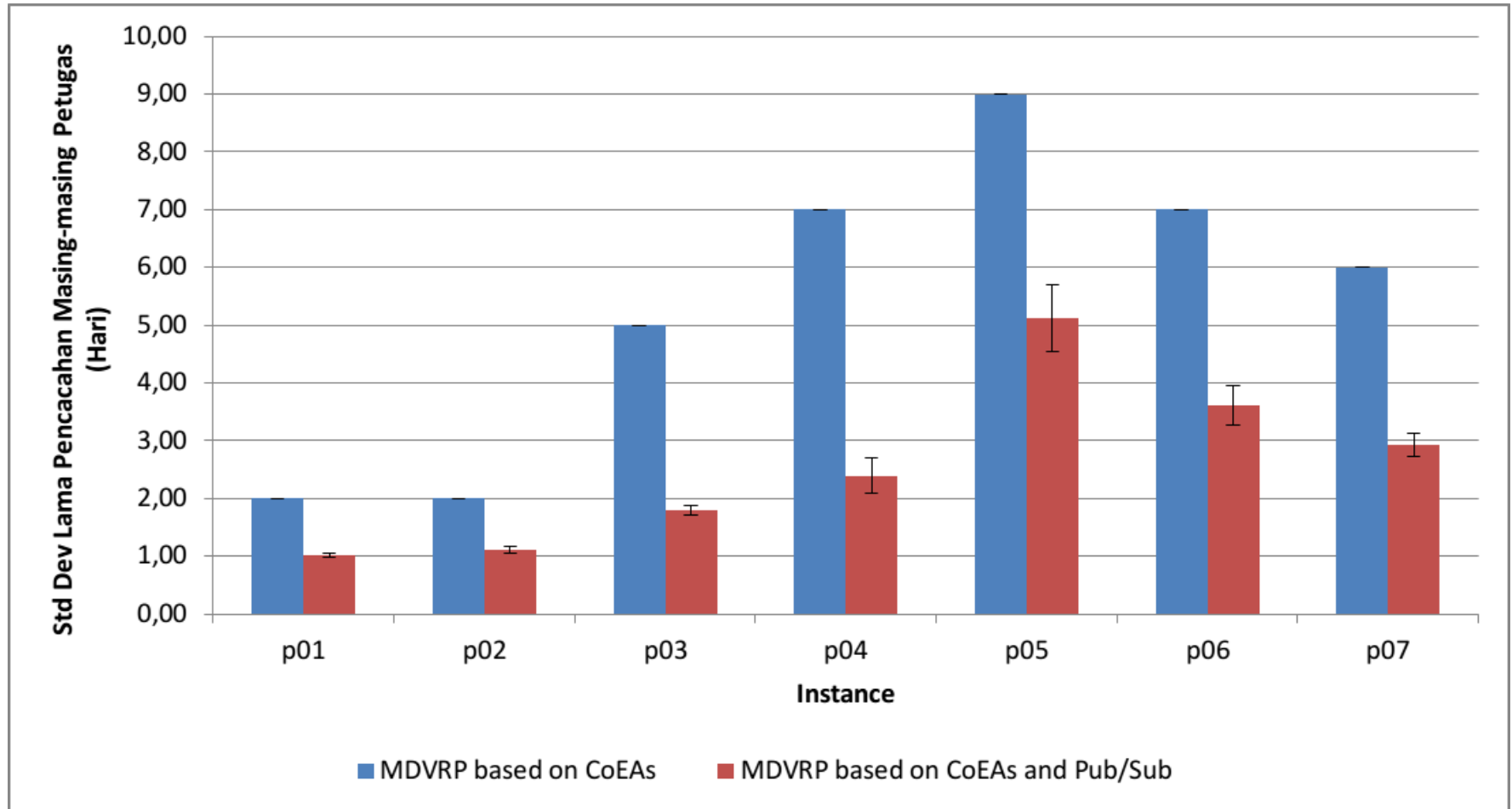
Rata-rata Lama Pencacahan Tiap-tiap Petugas



Pengujian

Hasil Pengujian Kondisi Normal (Cordeau)

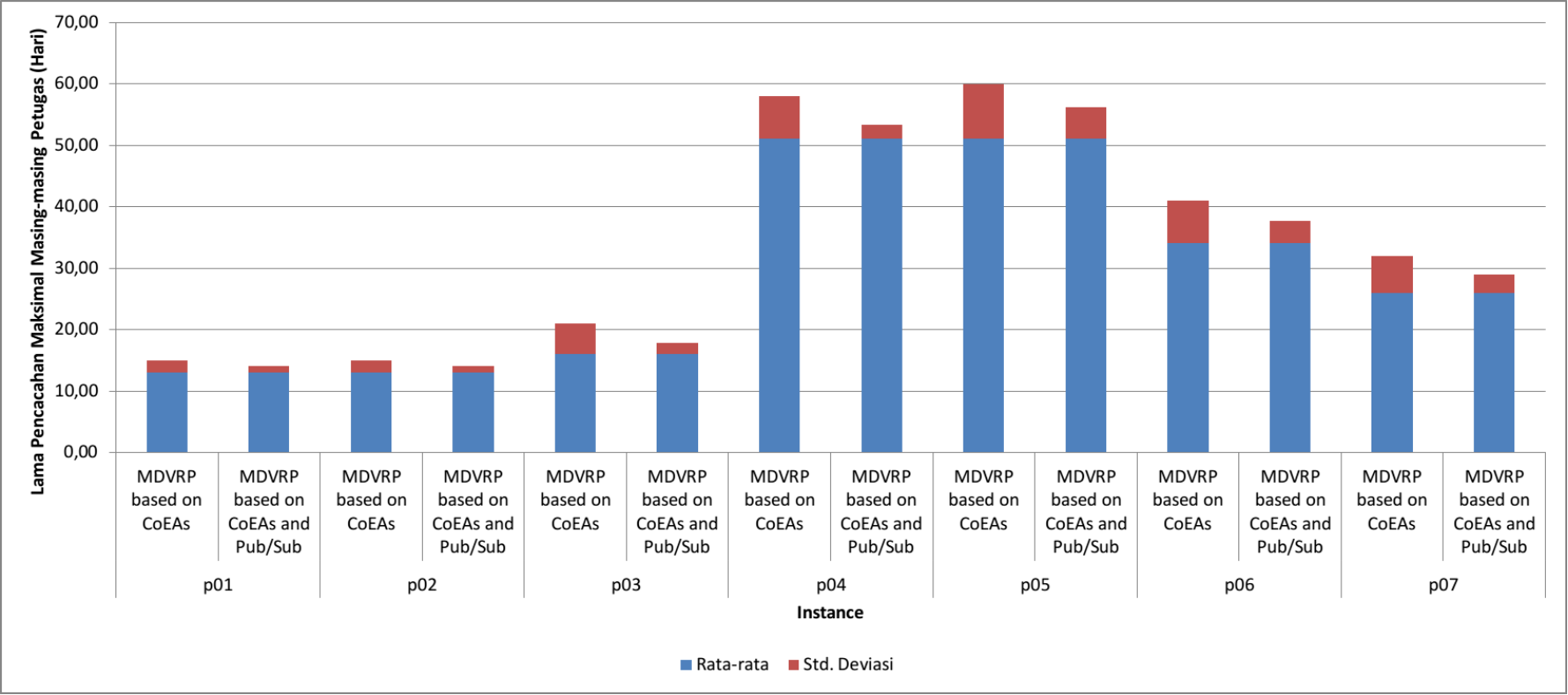
Variasi Lama Pencacahan Tiap-tiap Petugas



Pengujian

Hasil Pengujian Kondisi Normal (Cordeau)

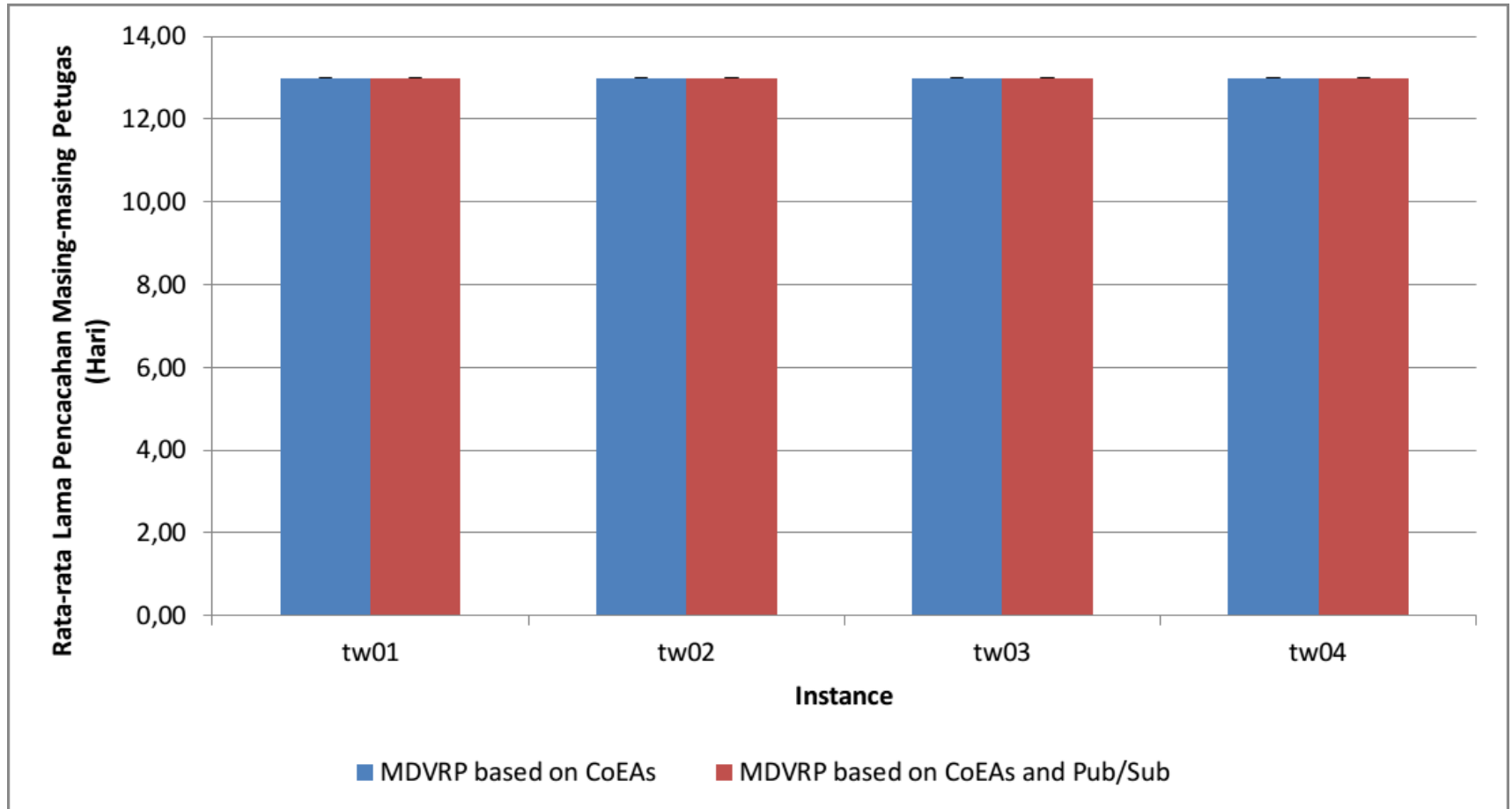
Waktu Tunggu Seluruh Pencacah Selesai



Pengujian

Hasil Pengujian Kondisi Normal (Lapangan)

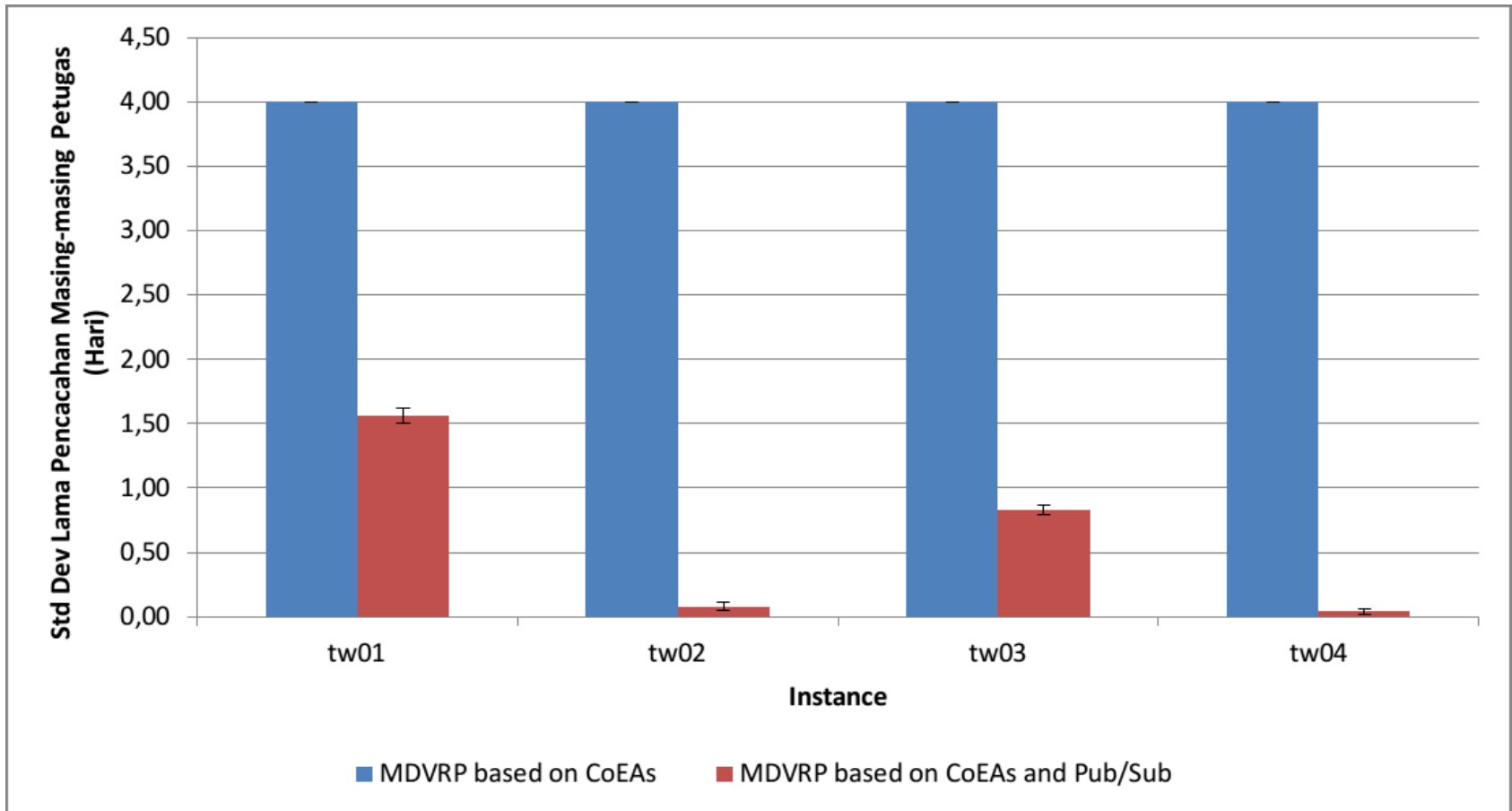
Rata-rata Lama Pencacahan Tiap-tiap Petugas



Pengujian

Hasil Pengujian Kondisi Normal (Lapangan)

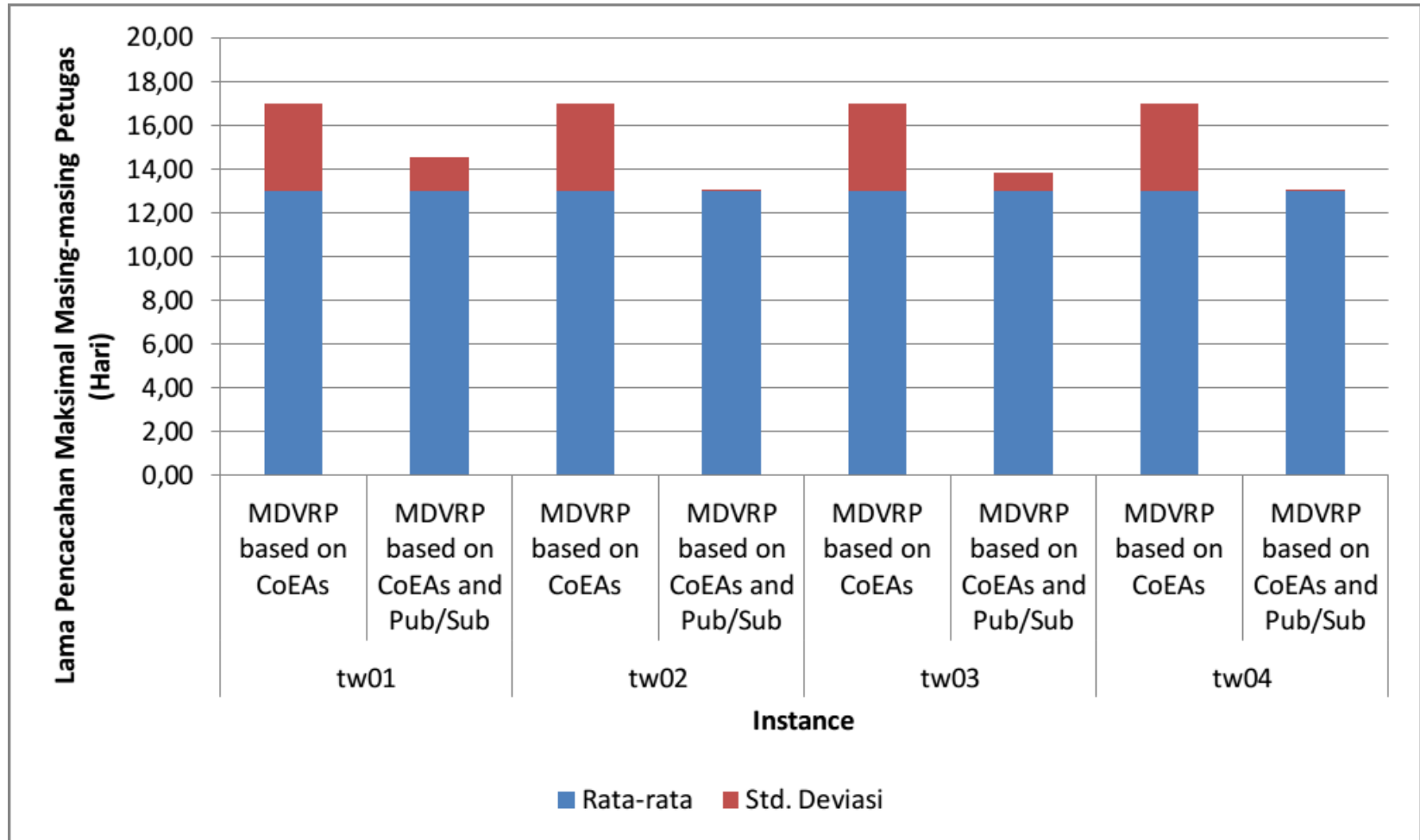
Variasi Lama Pencacahan Tiap-tiap Petugas



Pengujian

Hasil Pengujian Kondisi Normal (Lapangan)

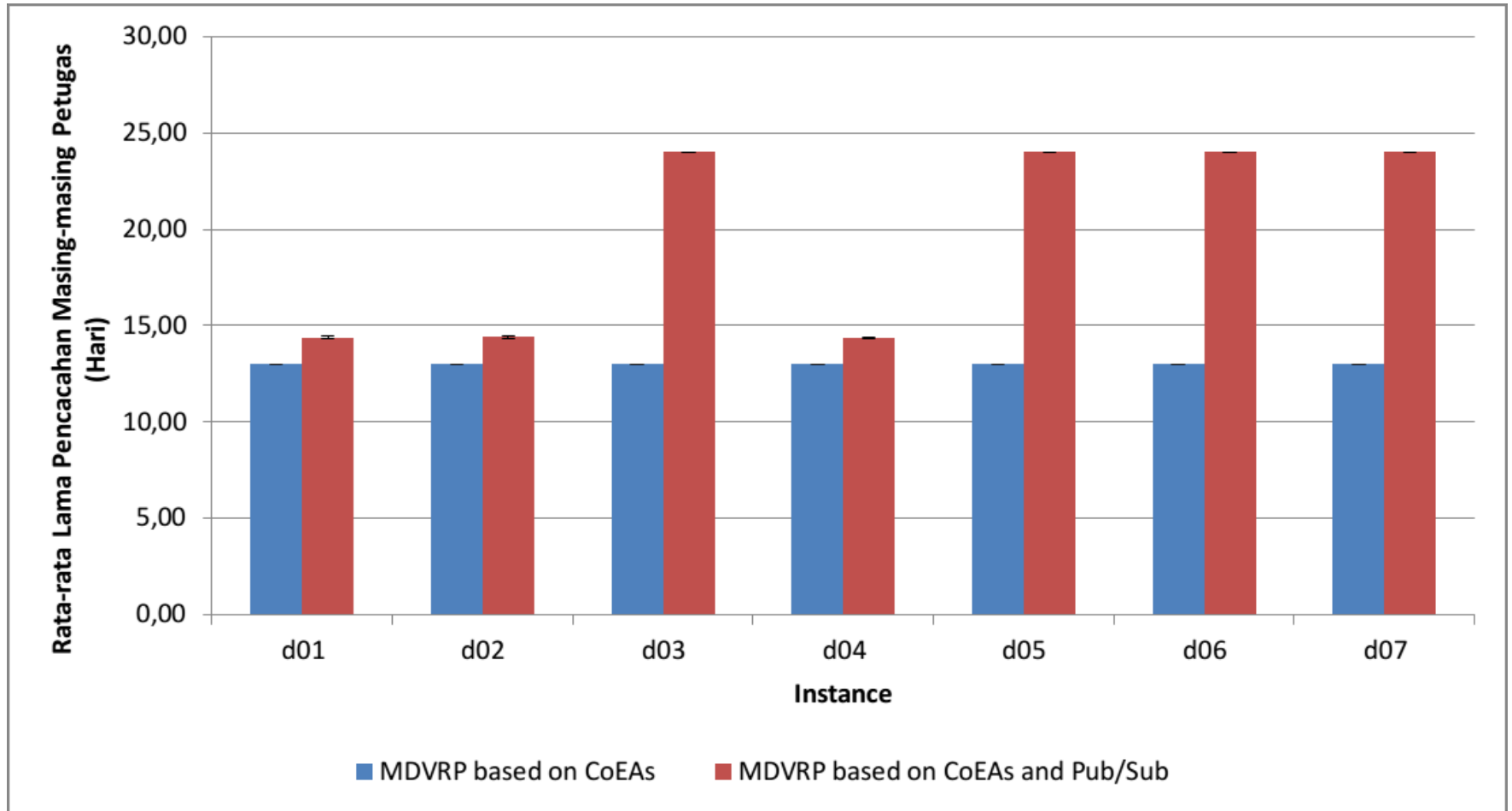
Waktu Tunggu Seluruh Pencacah Selesai



Pengujian

Hasil Pengujian Kondisi Delay

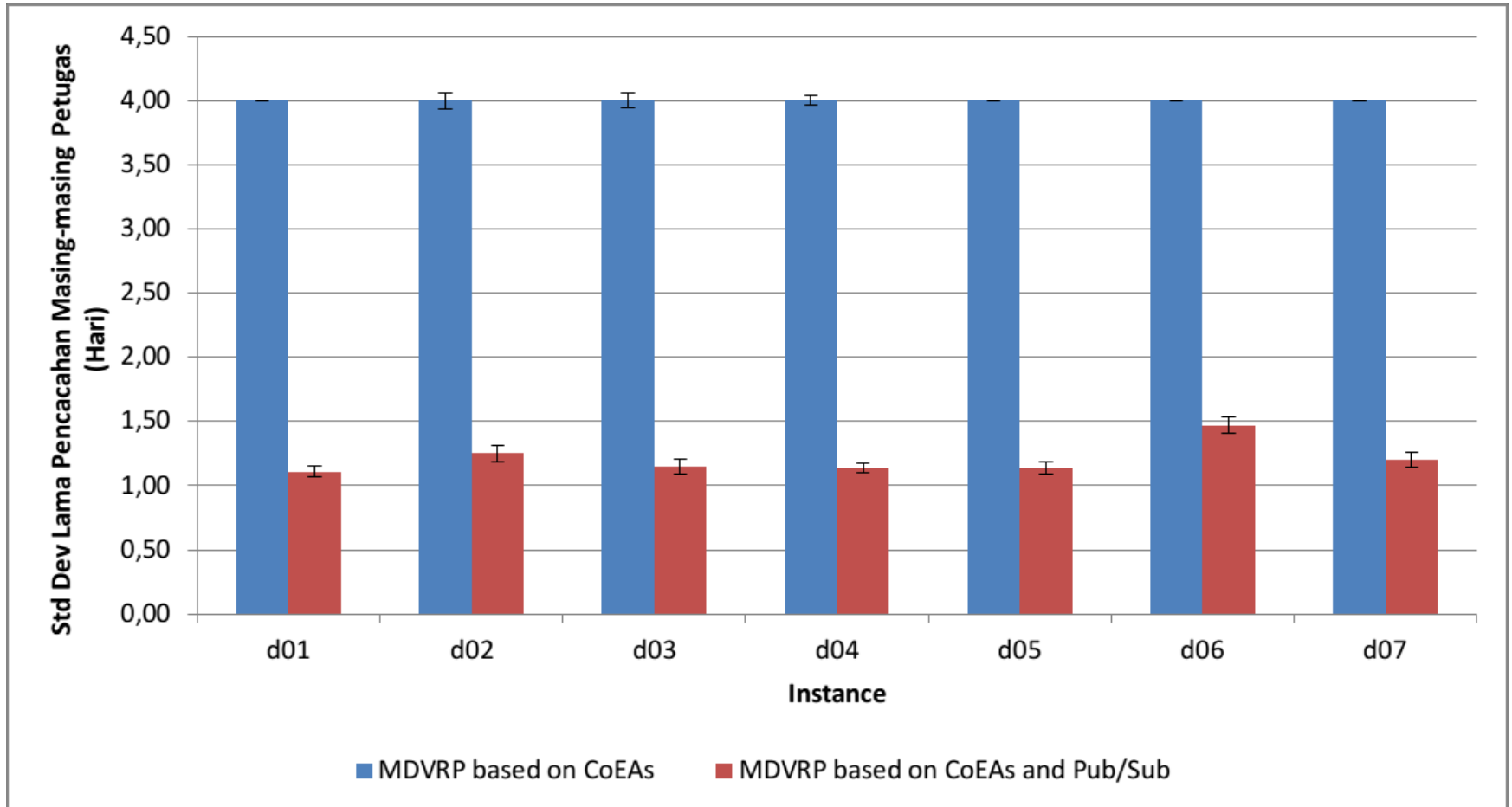
Rata-rata Lama Pencacahan Tiap-tiap Petugas



Pengujian

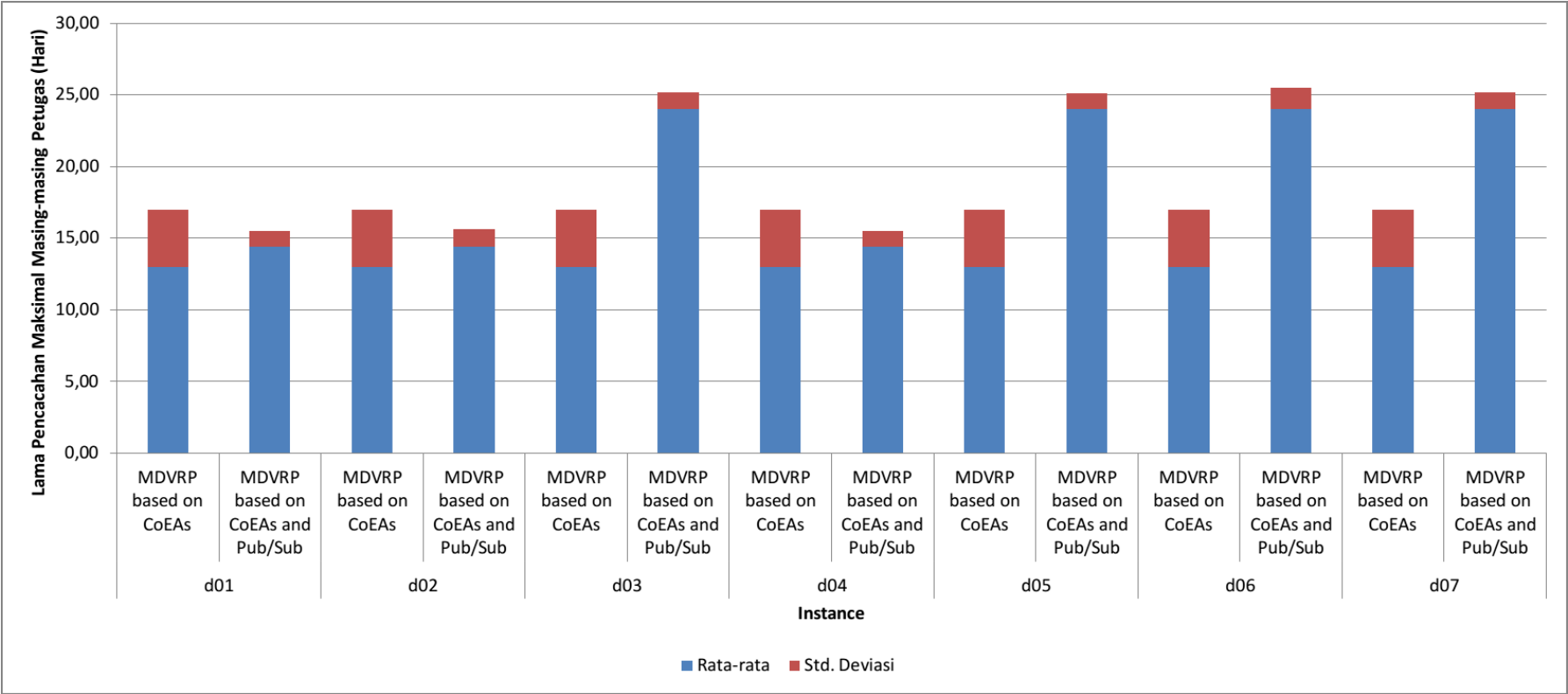
Hasil Pengujian Kondisi Delay

Variasi Lama Pencacahan Tiap-tiap Petugas



Hasil Pengujian Kondisi Delay

Waktu Tunggu Seluruh Pencacah Selesai



Kesimpulan

1. Penelitian ini telah menghasilkan sebuah rancangan sistem yang dapat digunakan untuk membuat rekomendasi lokasi pencacahan sesuai dengan konteks dari tiap-tiap petugas pencacahan.
2. Konteks yang paling sesuai digunakan pada sistem yang telah dirancang adalah lokasi terkini dari tiap-tiap pencacah.
3. Berdasarkan hasil pengujian, sistem usulan menghasilkan rekomendasi lokasi yang lebih efisien dibandingkan dengan sistem pembanding, baik dari segi rata-rata hari pencacahan, variasi lama hari pencacahan, maupun waktu tunggu sampai seluruh pencacahan menyelesaikan pencacahannya.
4. Berdasarkan hasil pengujian, sistem usulan dapat memberikan rekomendasi lokasi yang lebih baik pada kondisi tidak ada delay maupun delay yang rendah. Akan tetapi, sistem usulan memberikan hasil yang lebih buruk pada kondisi terdapat delay yang lama.

Saran

1. Mengkaji penggunaan mekanisme komunikasi yang lain, seperti mekanisme Push/Pull dan Request/Reply, untuk mendapatkan mekanisme komunikasi yang paling sesuai digunakan pada berbagai kondisi jaringan komunikasi
2. Mengkaji penggunaan algoritma penyelesaian MDVRP yang lain, seperti Tabu Search, Artificial Bee Colony, dan Particle Swarm Optimization, sehingga diperoleh algoritma yang menghasilkan rekomendasi paling akurat