



## PERBANDINGAN KECEPATAN KOMPUTASI SUDUT FOUR BAR LINKAGE ANTARA CPU DAN GPU

Adimas Wicaksana<sup>1</sup>

<sup>1</sup>Program Studi Teknik Mesin, Universitas Pamulang, Jl. Surya Kencana No. 1, Tangerang Selatan, Indonesia

E-mail : dosen01678@unpam.ac.id

Masuk : 5 Maret 2019

Direvisi : 20 Maret 2019

Disetujui : 27 Maret 2019

**Abstrak:** Four Bar Linkage adalah salah satu mekanisme paling tua yang dipelajari manusia. Kelebihan dari Four Bar Linkage adalah bentuknya yang relatif sederhana sekaligus serba guna. Karena banyaknya kombinasi parameter dari Four Bar Linkage sangat penting untuk menghasilkan sebanyak mungkin posisi untuk mempelajari gerakan dari Four Bar Linkage. Telah dibuat sebuah studi yang membandingkan kecepatan Intel i7-4790 dan AMD Radeon R9-280 dalam melakukan komputasi sudut dari Four Bar Linkage. Penelitian menunjukkan AMD Radeon R9-280 5x lebih cepat dalam melakukan komputasi sudut R9-280 untuk jumlah titik yang relatif lebih kecil (ratusan ribu) dan 8x lebih cepat untuk jumlah titik yang lebih besar (puluhan juta).

**Kata Kunci:** Four Bar Linkage, Komputasi, Mekanisme, OpenCL

**Abstract:** *Four Bar Linkage is one of the oldest mechanism ever studied by humankind. Among its advantages are its relative simplicity while remaining versatile. Due to its large number of parameter combinations it is quite important to compute as many points as possible to study the movement of Four Bar Linkage. This paper attempts to study Intel i7-4790 speed in finding Fourbar's coupler angle as compared to AMD Radeon R9-280. This study found that R9-280 computes coupler angle 5 times faster as compared to Intel i7-4790 when performing relatively low number of points (in hundreds of thousands) while being 8 times faster when computing large number of points (tens of millions).*

**Key Words:** *Four Bar Linkage, Computation, Mechanism, OpenCL*

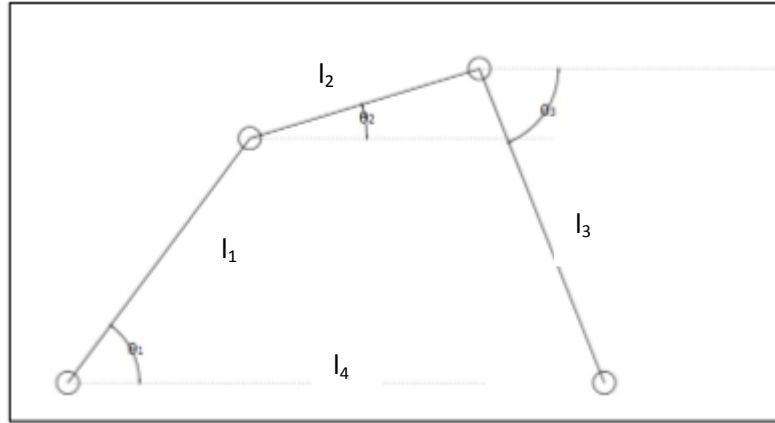
### PENDAHULUAN

Four Bar Linkage adalah sebuah mekanisme yang terdiri dari 3 penghubung yang bisa berputar yang masing-masing dihubungkan dengan sebuah pin atau engsel. Perlu diingat bahwa nama Four Bar Linkage berasal dari keberadaan sambungan ke empat yaitu tanah atau bingkai, dengan kata lain sambungan yang tidak bergerak.

Mekanisme four bar linkage ini sudah dikenal dari sejak jaman revolusi industri di tahun 1800-an dan banyak dipakai karena bentuknya yang relatif sederhana dan hanya membutuhkan satu penggerak utama. Selain itu mekanisme sifatnya closed loop sehingga sifatnya relatif lebih stabil daripada mekanisme open loop (misalkan lengan robot). Dari segi gerakan yang dihasilkan four bar ini pun memiliki banyak variasi sehingga sangat fleksibel bagi kita untuk mendesain parameter agar kita bisa menghasilkan gerakan yang kita inginkan. Mekanisme fourbar linkage ini digunakan antara lain untuk pembuatan lutut palsu bagi orang lumpuh [1] dan juga pemodelan dalam cara kerja rahang ikan [2].

Baru-baru ini Disney Research [3] juga mengeluarkan jurnal tentang pemanfaatan Four Bar Linkage sebagai bagian dari pemodelan animatronics. Pada dasarnya seorang animator akan membuat sketsa gerakan yang diinginkan kemudian komputer akan menentukan kombinasi parameter seperti apakah yang akan memenuhi kriteria gerakan yang sudah ditentukan. Studi seperti ini adalah hal yang relatif baru dengan berkembangnya teknologi komputasi. Sebelum itu Acharrya et al. [4] dan Lin et al. [5] juga mengeluarkan studi tentang cara-cara menghasilkan Four Bar Linkage yang menghasilkan kurva yang sudah ditentukan. Metode seperti ini bergantung pada kemampuan prosesor dalam menentukan titik-titik yang dilewati oleh Four Bar Linkage. Semakin banyak

kombinasi parameter yang dicoba maka akan semakin dekat pula kurva yang dihasilkan terhadap kurva yang diinginkan. Pada umumnya pemrosesan akan dilakukan menggunakan CPU yang sudah banyak digunakan sejak tahun 1980-an. Akan tetapi beberapa saat ini telah ditunjukkan bahwa bagi beberapa aplikasi seperti Google TensorFlow [6] GPU atau Graphic Card menunjukkan performa yang kompetitif atau bahkan lebih baik daripada CPU. Dalam bidang teknik mesin GPU juga banyak dipakai dalam simulasi aliran fluida[7]. Program komersial seperti ANSYS dan CATIA juga menggunakan GPU dalam melakukan komputasi.



Gambar 1. Four Bar Linkage

Walaupun demikian setahu penulis belum ada studi yang dilakukan untuk membandingkan kemampuan CPU dan GPU dalam menentukan posisi Four Bar Linkage. Adapun satu paper oleh Chen et al[8] yang mendekati studi ini namun paper tersebut tidak membandingkan performa CPU dan GPU. Selain itu paper tersebut dikeluarkan 13 tahun yang lalu sehingga mungkin tidak lagi relevan karena cepatnya perkembangan teknologi. Hal inilah yang mendorong penulis untuk meneliti kemungkinan penggunaan GPU dalam simulasi Four Bar Linkage.

## METODOLOGI

Untuk komputer yang digunakan adalah HP Pavilion PC 500-330d. Komputer ini mempunyai CPU yang cukup cepat, yaitu Intel i7-4970. Dengan demikian kita bisa langsung menggunakan CPU tersebut sebagai pembanding dari kecepatan GPU.

Karena GPU yang ada pada komputer tersebut cenderung relatif low-end maka digunakanlah GPU tambahan AMD Radeon R9 280. Pemilihan GPU tersebut adalah berdasarkan fakta bahwa GPU ini masih relatif baru (dikeluarkan tahun 2014) dan masih termasuk salah satu GPU paling cepat di pasaran.



Gambar 2. AMD Radeon R9 280

Agar tidak perlu menggantikan GPU original maka digunakanlah PCI/E extender untuk memanjangkan sambungan PCI/E dari komputer menuju ke GPU. Dengan demikian kita juga tidak perlu terlalu memusingkan kurangnya konektor PCI/E-16x pada motherboard komputer asli.



**Gambar 3.** PCI/E Extender

Masalah berikutnya yang kita hadapi adalah kurangnya daya yang bisa disupply oleh power supply asli dari komputer. Power supply pada komputer asli cenderung terbatas, yaitu 300W sementara konsumsi daya R9 280 adalah sekitar 250W padahal perlu diingat bahwa konsumsi daya CPU sendiri adalah sekitar 100W dan GPU asli komputer menggunakan 50W. Oleh karena itu maka digunakanlah power supply ATX tambahan Corsair VS650 yang bisa mensupply daya sebesar 650W.



**Gambar 4.** Power Supply Corsair VS650

Karena power supply ini tidak terhubung ke motherboard maka kita perlu menyambungkan kabel hijau pada power supply ke ground (kabel hitam) supaya power supply tetap bisa menyala. Kabel ini mempunyai tegangan 5V dan harus ditarik ke ground (0V) untuk menandakan bahwa motherboard sudah siap.



**Gambar 5** Menghubungkan Kabel Hijau ke Ground menggunakan Paper Clip

Dalam studi ini kita berusaha membandingkan kecepatan CPU dan GPU dalam melakukan perhitungan terhadap sudut yang dibuat oleh sambungan ke 2 terhadap garis horizontal dengan memberikan sudut sambungan

pertama sebagai inputnya. Untuk lebih jelasnya perhatikanlah Gambar 1 di atas. Dalam kasus ini kita menganggap bahwa sambungan pertama dihubungkan langsung dengan penggerak utama. Jika kita mengetahui sudut  $\theta_1$  maka kita bisa mendapatkan sudut  $\theta_2$  dengan cara sebagai berikut:

$$l_1 \sin \theta_1 + l_2 \sin \theta_2 - l_3 \sin \theta_3 = 0 \quad (1)$$

$$l_1 \cos \theta_1 + l_2 \cos \theta_2 + l_3 \cos \theta_3 = l_4 \quad (2)$$

Dengan mengkuadratkan masing masing persamaan di atas dan kemudian menjumlahkannya kita bisa mendapatkan:

$$l_4^2 + l_1^2 + l_2^2 + 2l_1l_2(\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) - 2l_4(l_1 \cos \theta_1 + l_2 \cos \theta_2) = l_3^2 \quad (3)$$

Dengan memindahkan semua suku ke sisi yang sama:

$$l_3^2 - l_4^2 - l_1^2 - l_2^2 + (2l_4l_2 - 2l_1l_2 \cos \theta_1) \cos \theta_2 - 2l_1l_2 \sin \theta_1 \sin \theta_2 = 0 \quad (4)$$

Persamaan ini mempunyai bentuk:

$$A \sin \theta_2 + B \cos \theta_2 + C = 0 \quad (5)$$

Dimana:

$$A = -2l_1l_2 \sin \theta_1 \quad (6)$$

$$B = 2l_4l_2 - 2l_1l_2 \cos \theta_1 \quad (7)$$

$$C = l_3^2 - l_4^2 - l_1^2 - l_2^2 \quad (8)$$

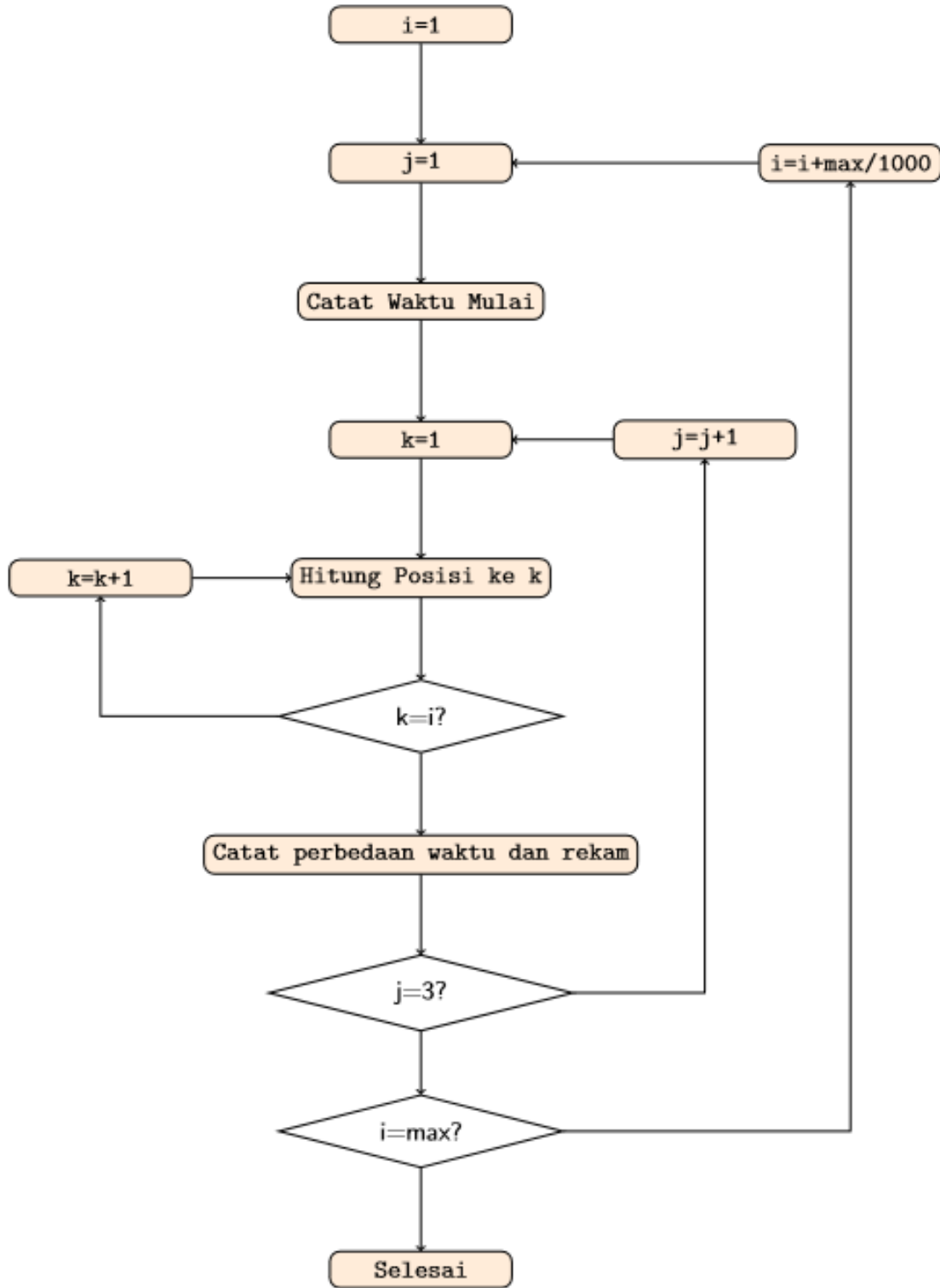
Penyelesaian dari persamaan ini adalah

$$\delta = \arctan \frac{A}{B} \quad (9)$$

$$\theta_2 = \delta \pm \arccos \frac{-C}{\sqrt{A^2 + B^2}} \quad (10)$$

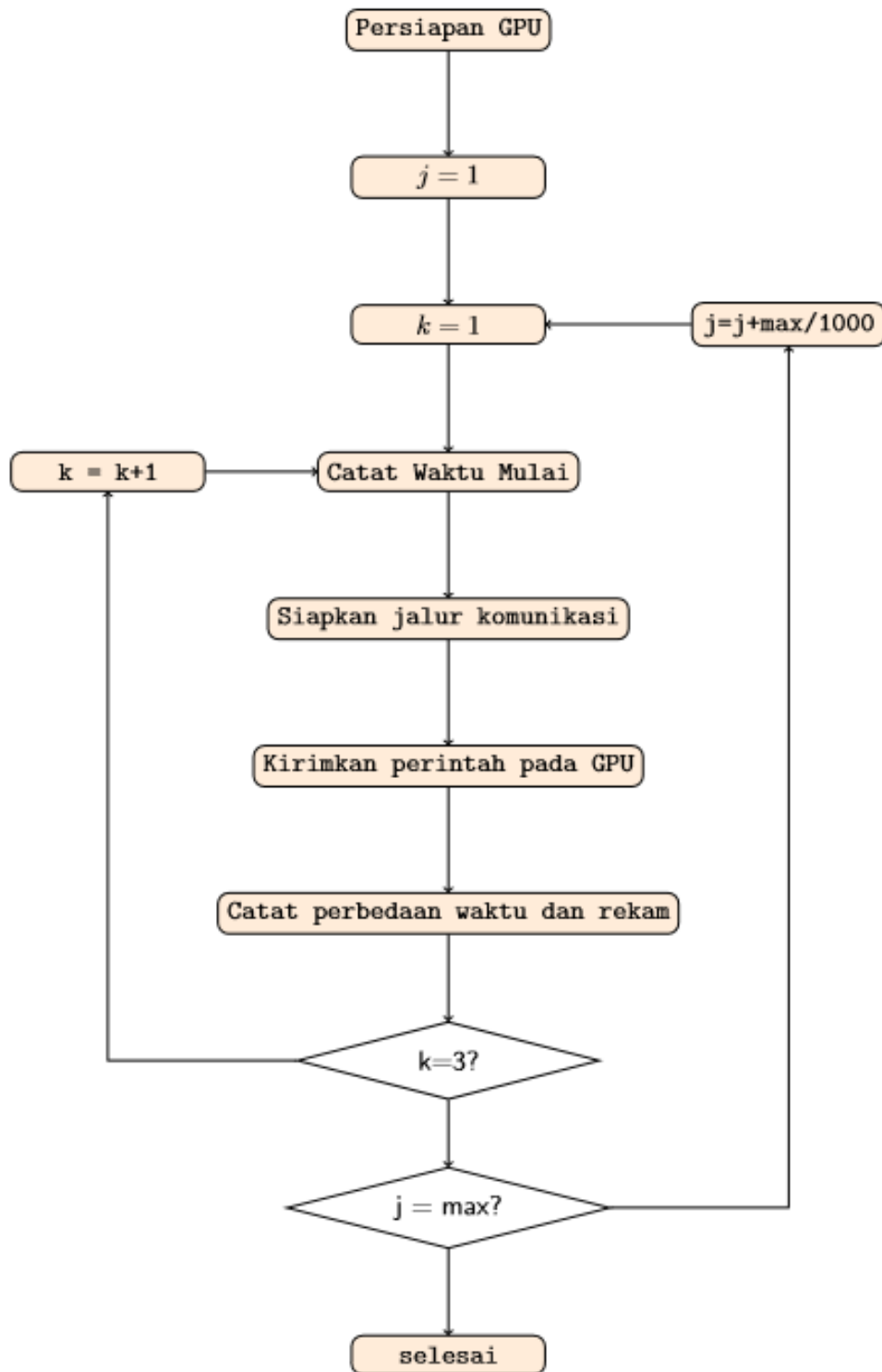
Penelitian ini akan mencoba membandingkan kecepatan perhitungan  $\theta_2$  untuk beberapa titik dengan menggunakan metode yang sudah dijelaskan di atas. Untuk mengukur waktu yang diperlukan studi ini akan menggunakan paket `std::chrono` yang merupakan bagian standar dari C++11. Pada dasarnya kita mencatat waktu sebelum perhitungan dilakukan dan waktu setelah perhitungan selesai. Perbedaan antara kedua waktu tersebut adalah waktu yang dihabiskan untuk perhitungan. Metode ini sudah banyak digunakan untuk keperluan benchmarking seperti pada contohnya studi oleh Ubik et al. tentang pengambilan video dari mobil tanpa awak[9]. Untuk mengeliminir noise maka untuk setiap pengukuran dilakukan 3 kali berturut-turut dan kemudian diambil rata-ratanya.

Walaupun secara garis besar pengukuran yang dilakukan terhadap baik CPU maupun GPU adalah sama namun tetap ada sedikit perbedaan dalam detail implementasi. Hal ini dikarenakan eksekusi dalam GPU sedikit lebih rumit daripada eksekusi pada CPU. Untuk lebih jelasnya perhatikanlah bagan yang menunjukkan pengukuran pada CPU berikut



Gambar 6. Diagram Alir Pencatatan Waktu CPU

Bandingkanlah dengan diagram alur perhitungan waktu pada GPU pada halaman berikut :



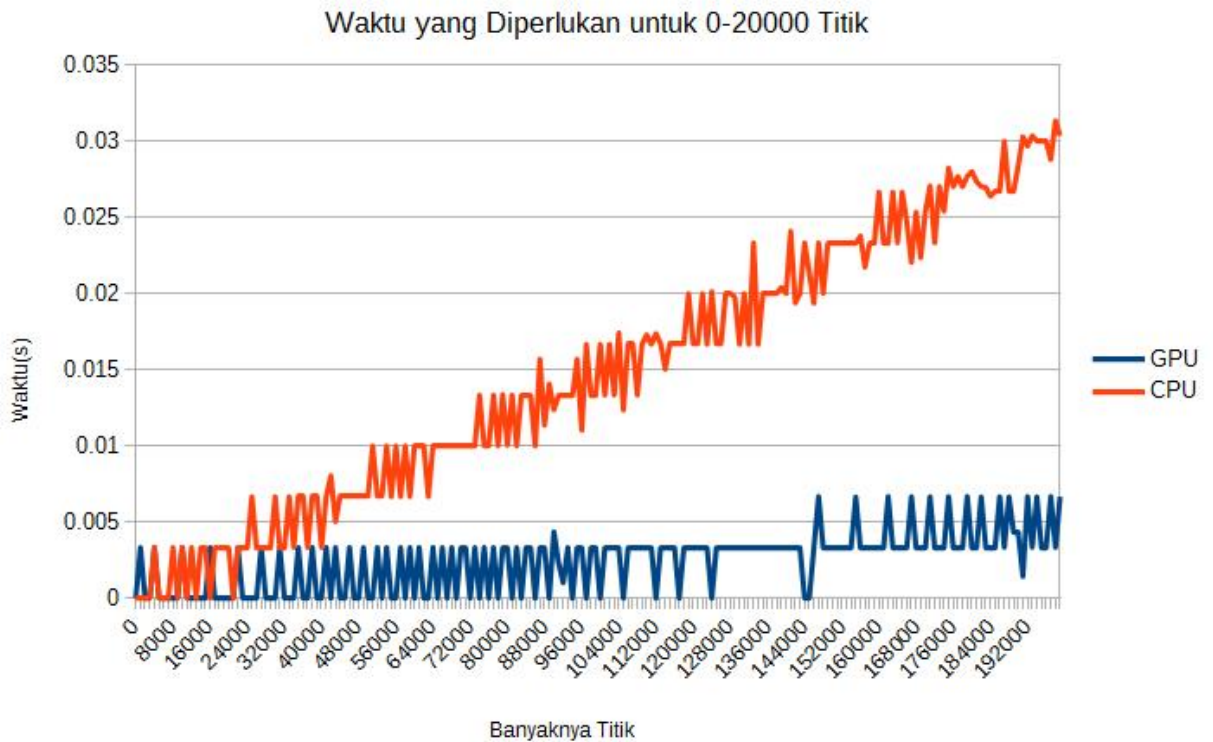
**Gambar 7.** Diagram Alir Pencatatan Waktu GPU

Bisa kita lihat bahwa terdapat perbedaan antara skema pengukuran sebagai berikut :

- a. Adanya fasa persiapan untuk GPU. Fasa ini termasuk permintaan ijin penggunaan GPU ke Operating System, pembentukan Context dan Command Queue, kompilasi terhadap software kernel (program ini perlu dikompilasi terpisah dari program utama), dsb. Fasa ini tidak dimasukkan ke dalam perhitungan waktu karena cukup dilakukan sekali saja.
- b. Perlunya membentuk jalur komunikasi ke GPU, hal ini dikarenakan GPU adalah program eksternal. Jalur komunikasi ini sifatnya sekali pakai sehingga perlu kita masukkan ke dalam perhitungan waktu
- c. Kita tidak perlu melakukan loop untuk setiap titik pada GPU, hal ini dikarenakan GPU mempunyai banyak processing unit, sehingga masing masing perhitungan bisa dilakukan secara independen

### HASIL DAN PEMBAHASAN

Berikut adalah perbandingan antara waktu yang diperlukan bagi CPU dan GPU untuk menyelesaikan komputasi antara 0-200 ribu titik.



**Gambar 8.** Hasil Analisis Perbandingan Kecepatan GPU vs CPU untuk Jumlah Titik < 200.000

Bisa kita lihat bahwa untuk jumlah titik yang relatif kecil relatif tidak ada perubahan pada waktu yang diperlukan GPU untuk memproses titik-titik tersebut. Hal ini dikarenakan GPU mempunyai banyak processing unit sehingga masing-masing titik bisa dikerjakan secara independen. Bisa kita lihat bahwa setelah sekitar 24.000 titik GPU lebih cepat daripada CPU dimana GPU sekitar 5 kali lebih cepat pada akhir grafik. Bandingkanlah dengan grafik yang menunjukkan komputasi antara 0-50 juta titik di bawah ini



**Gambar 9.** Hasil Analisis Perbandingan Kecepatan GPU vs CPU untuk Jumlah Titik > 200.000

Pada jumlah titik yang sangat besar processing unit pada GPU sudah mulai jenuh sehingga waktu pemrosesan menjadi linear terhadap banyaknya titik. Pada rentang ini GPU sekitar 8 kali lebih cepat daripada CPU. Melihat linearnya grafik yang dihasilkan tampaknya pada perbandingan ini adalah hasil maksimal GPU dibandingkan dengan CPU.

## KESIMPULAN

Telah dibuat sebuah studi yang membandingkan kecepatan Intel i7-4790 dan AMD Radeon R9-280 dalam melakukan komputasi sudut dari Four Bar Linkage. Penelitian menunjukkan AMD Radeon R9-280 5x lebih cepat dalam melakukan komputasi sudut R9-280 untuk jumlah titik yang relatif lebih kecil (ratusan ribu) dan 8x lebih cepat untuk jumlah titik yang lebih besar (puluhan juta).

## DAFTAR PUSTAKA

- [1] CW Radcliffe. *Four-bar linkage prosthetic knee mechanisms: kinematics, alignment and prescription criteria*. *Prosthetics and orthotics international*, 18(3):159–173, 1994.
- [2] Mark W Westneat. *Feeding mechanics of teleost fishes (labridae; perciformes): A test of four-bar linkage models*. *Journal of morphology*, 205(3):269–295, 1990.
- [3] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W Sumner, Wojciech Matusik, and Bernd Bickel. *Computational design of mechanical characters*. *ACM Transactions on Graphics (TOG)*, 32(4):83, 2013.
- [4] SK Acharyya and M Mandal. *Performance of eas for four-bar linkage synthesis*. *Mechanism and Machine Theory*, 44(9):1784–1794, 2009.
- [5] Wen-Yi Lin. *A ga-de hybrid evolutionary algorithm for path synthesis of four-bar linkage*. *Mechanism and Machine Theory*, 45(8):1096–1107, 2010.
- [6] Mehdi Goli, Luke Iwanski, and Andrew Richards. *Accelerated machine learning using tensorflow and sycl on opencl devices*. In *Proceedings of the 5th International Workshop on OpenCL*, page 8. ACM, 2017.
- [7] Dana A Jacobsen, Julien C Thibault, and Inanc Senocak. *An mpi-cuda implementation for massively parallel incompressible flow computations on multi-gpu clusters*. In *48th AIAA aerospace sciences meeting and exhibit*, volume 16, page 2, 2010.



[8] Jie-ping CHEN and Chang-peng GUO. *Simulation software development for four-rod mechanism*. 2004.

[9] Sven Ubik, Jiří Melnikov, and Zdeněk Trávníček. *On acquisition of high quality video from unmanned vehicles*. In *Internet of Things (WF-IoT)*, 2015 IEEE 2nd World Forum on, pages 75–80. IEEE, 2015.