

VISUALIZATION 2D MULTIPHASE FLUID WITH LATTICE BOLTZMAN METHOD SIMULATION IN GRAPHICS PROCESSING UNIT

Arifiyanto Hadinegoro

Informatika Universitas AMIKOM Yogyakarta

email : arifiyanto@amikom.ac.id

Abstraksi

LBM adalah salah satu metode pendekatan computational Fluid Dynamics (CFD) yang sangat baik untuk menggambarkan sistem fisik dalam bentuk cell mikro, dimana pada setiap cellnya melakukan interaksi seperti streaming dan tumbukan, hasil yang di dapatkan dari metode ini , di ambil kemudian di baca menggunakan library OpenGL untuk memvisualkan simulasinya, untuk dimulasi 2 fase menggunakan metode shan dan chen yang menambahkan nilai untuk gaya eksternal seperti gravitasi, penerapan paralel computing menggunakan GPU di butuhkan untuk mengurangi beban komputasinya serta pengurangan waktu hingga mendekati kondisi nyata, namun perbedaan arsitektur dan tata cara penulisan program membuat implementasi simulasi cairan 2D dua fase membutuhkan modifikasi formula untuk mendapatkan nilai setiap cellnya agar hasil simulasi mendekati hasil nyata.

Kata Kunci :

Fluida, GPU, lattice Boltzman

Abstract

Fluid is one of the natural ingredients which is quite difficult to make the simulation, apart from the versatility of its shape, great computing is also one obstacle to making it. The development of Graphical Processing Unit (GPU) as an image processor on the computer began to switch functions as a tool for parallel computing, The use of GPU as a computing tool is still very rarely used in conducting research, they tend to be used only for graphic or gameplay. GPU programming is a parallel computing technology to improve performance in a simulation. Because in a simulation that applied computer needs a large processing, and limited CPU capabilities also limit the simulation performance itself. In this case, the author tries to implement a 2-phase fluid simulation that requires quite a lot of computing, That is written in a discrete method using the Boltzmann methods and visualized using the OpenGL framework. The final result of the study is to see the difference in visualization display using OpenGL without GPU programming and using GPU programming.

Keywords :

Fluida, GPU, lattice Boltzman,

Pendahuluan

Visualisasi pada sebuah simulasi computer di buat dengan mengambil hasil nilai yang telah di buat secara numerik, beberapa penelitian memanfaatkan library OpenGL sebagai tool penerjemah kedalam bentuk vector [1], dan beberapa penelitian yang melibatkan cairan seperti gelombang laut atau tetesan cairan pada bidang bebas [2][3], cairan sendiri adalah zat yang mudah berubah-ubah sesuai penampangnya dan terpengaruh pada factor luar seperti tekanan udara, grafitasi membuat fenomena ini cukup rumit untuk membuat pendekatannya, dasar untuk membuat pendekatan ini salah satunya adalah metode Lattice Boltzman yang telah banyak di gunakan dalam berbagai kasus singlephase atau multiphase [3]–[6], untuk model multiphase penulis menggunakan Model Shan –Chen [7] untuk penerapan model multiphase, Graphics Processing Unit pada perkembangannya tidak hanya di gunakan

sebagai alat bantu untuk menampilkan hasil grafis, namu isa di dimanfaatkan untuk keperluan komputasi, seperti yang banyak di gunakan beberapa penelitian lain utnuk meningkatkan performa pada satu simulasi [8], [9], pada GPU penulisan harus menggunakan library yang tersedia seperti CUDA, dari Nvidia corp, [10], dari perbedaan arsitektur antara GPU dan CPU penulis ingin membandingkan hasil visualisasi cairan multiphase yang di tulis pada CPU dan GPU dengan menggunakan metode yang sama yaitu LBM.

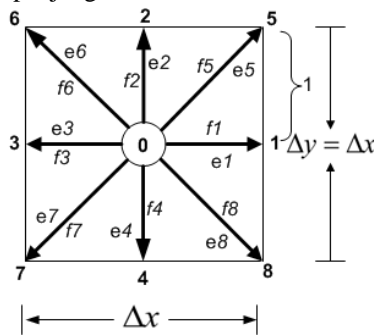
Tinjauan Pustaka

A. Lattice Boltzmann Method (LBM)

Metode Lattice boltzman atau di singkat LBM, metode yang di kembangkan dari cellular automata, dimana setiap benda terdiri dari banyak sel sejenis, kemudian setiap sel mempunyai nilai tersendiri dan mempengaruhi terhadap sel tetangganya , LBM memiliki pola sel sederhana dimana setaip sel

memiliki arah gerak statis untuk hubungan antara tiap sel masing masing, gerak sel ini dihitung secara matematik dengan aturan memperhitungkan nilai perubahan pada setiap sel sekitarnya.

DnQm adalah lambang model dari LBM, n menyatakan jumlah dimensi yang di gunakan dan m menyatakan jumlah arah lattice yang di gunakan, model yang umum digunakan adalah D1Q2 dan D1Q3, D1Q5, D2Q5 dan D2Q4, D2Q9, D3Q15, dan D3Q19, model ini gunakan sesuai kebutuhan, Hasil visual dari peneltian ini menggunakan 2 dimensi maka model D2Q9, 2 dimensi dan 9 arah lattice, yang di gunakan sebagai dasarnya pada Gambar 1 Arah variable LBM dibawah setiap titik atau di wakili dengan ea, memiliki nilai nol jika tidak bergerak tau dalam kondisi diam, setiap arah sel memiliki panjang satu.



Gambar 1 Arah variable LBM

Arah Vektor pada LBM seluruhnya bernilai 1, pada gambar 1 titik tengah pada model gambar memiliki nilai nol (0) dan panjang Nol (0) dan sebgai penyimpan jumlah partake yang berjalan atau berhenti di sel berikutnya, susunan variable arahnya adalah f_i yaitu $f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8$. Untuk perhitungan desitas $\rho(\rho)$ (1) dan kecepatan atau velocity (2) di hitung dengan:

$$\rho = \sum_{i=1}^{19} f_i \quad (1)$$

$$u = \frac{1}{\rho} \sum_{i=1}^{19} f_i e_i \quad (2)$$

Proses aliran (streaming) (3), tahap tabrakan (collision) (4) di tambahkan F sebagai gaya luar (5), merupakan proses yang terjadi pada LBM.

$$f_i(\bar{x}, t) = f_i'(\bar{x} - \bar{e}_i, t + 1) \quad (3)$$

$$f_i'(\bar{x}, t) = f_i(\bar{x}, t) - \frac{1}{\tau} (f_i(\bar{x}, t) - f_i^{eq}(\bar{x}, t)) + F_i \quad (4)$$

$$f_i(\bar{x} + \bar{e}_i \Delta t, t + \Delta t) - f_i'(\bar{x}, t) = -\frac{1}{\tau} (f_i - f_i^{eq}) + F_i \quad (5)$$

Setiap fungsi distribusi tetap di perlukan tahapan aliran atau streaming pada sel lain, secara numerik LBM bisa di tulis dalam persamaa aliran dalam

satuan waktu, Proses tumbukan di lakukan untuk mendapatkan nilai keseimbangan (fieq).

B. Lattice Boltzmann untuk multiphase

Persamaan untuk dua fase atau lebih di kenal multiphase Shan & Chen [7], disebutkan cukup stabil dalam hal menghitung gaya luar.

$$F(x, t) = -G\psi(x, t) \sum_{a=1}^9 w_a \psi(X + e_a \Delta t, t) e_a \quad (6)$$

G adalah kekuatan interaksi, w_a adalah $1/9$ $a = \{1, 2, 3, 4\}$, di mana $1/36$ for $a = \psi$ adalah potensi interaksi persamaannya di tulis pada persamaan (7) di bawah:

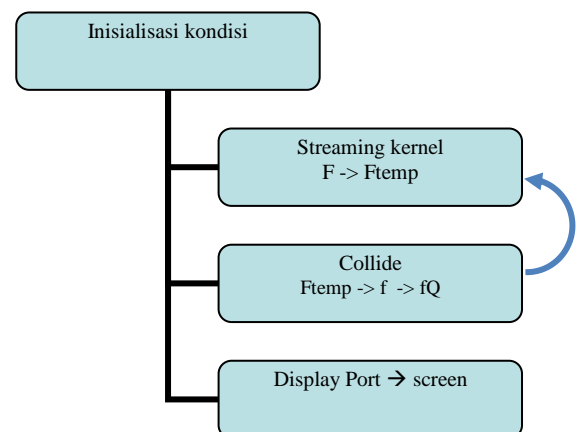
$$\psi(\rho) = \psi_0 \exp(-\rho_0 / \rho) \quad (7)$$

ψ_0 dan ρ_0 adalah nilai konstan sembarang.

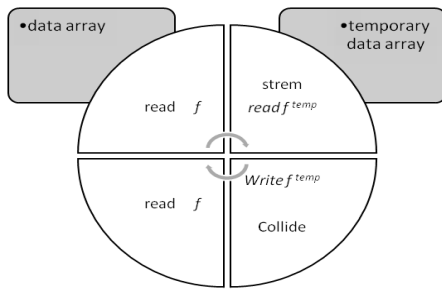
C. Implemntasi pada GPU

Pada bagian ini penerapan secara numerik 2d LBM pada GPU dengan menggunakan Compute Unified Device Architecture (CUDA), Cuda sendiri di rancang sebagai bahasa pemograman yang di rancang dengan menggunakan konsep single Intruction Multiple Data (SMID), dan memungkinkan para pengguna GPU NVIDIA untuk memanfaatkan komputasi ilmiah, walau menggunakan GPU Nvidia dan CUDA pada penelitian ini Metode SMID juga di terapkan pada bahasa pemograman lain seperti open CL, karena dengan menggunakan selain CUDA maka tidak tergantung pada GPU Nvidia saja.

Umumnya tahapan LBM terlihat pada gambar di bawah, pada bagian tertentu di butuhkan komputasi yang besar, makan tahap tersebut yang di pindahkan pada GPU dengan memesan variable pada memori GPU setelah selesai maka nilai hasil akan di ikirinkan kembali ke CPU sebagai nilai variable yang akan di gunakan sebagai nilai visualisasi, alur terebut bisa di lihat pada Gambar 2.



Gambar 2 Struktur utama LBM

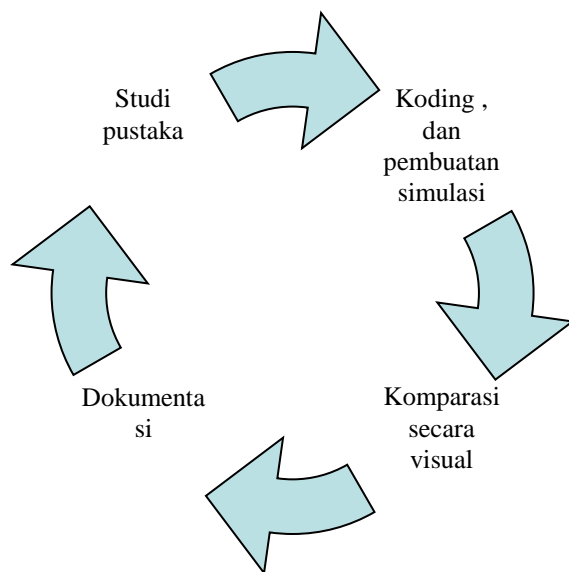


Gambar 3 Proses implementasi pada GPU

Pada dasarnya hasil dari perhitungan f^{temp} di dalam GPU akan di kirimkan ke variable f yang kemudian nilai variable f inilah yang akan di ambil sebagai nilai visualnya.

Metode Penelitian

Penelitian ini di lakukan dengan eksperimen, di mana penulis melakukan studi pustaka untuk algoritma numeriknya , kemudian membuat model 2D dengan Open GL, kemudian barisan kode di ubah agar bisa di jalankan menggunakan paralel computing Cuda, setelah itu di bandingkan hasil dari visual nya secara langsung dan komparasi nilai pixelnya, secara ringkas bisa di lihat pada Gambar 4.



Gambar 4 alur langkah penelitian

Code program yang di gunakan adalah pengembangan dari code program fluida satu fase yang di di kembangkan oleh Pullan pada tahun 2008[11] secara kode penambahan yang di berikan ada pada formula multiphase yang di buat shan dan chen [7] serta nilai densitasnya yang di gunakan, serta para meter eksternal lainnya detailnya bisa di lihat pada Table 1 parameter simulasi kemudian implementasi pada kode program di sesuaikan dengan kebutuhan bahasa pemograman yang di gunakan potongan kode program ada pada Gambar 5.

TABLE 1 PARAMETER SIMULASI

Variable	Real	Simulasi
Ukuran sel	0.0005(m)	1
Selang waktu	0.00001(s)	1
Gravitasi	9.81(ms ⁻²)	4.0e-4
Kekentalan	10 ⁻⁶ (m ² s ⁻¹)	4.0e-4
Massa jenis udara	1.2 kg/m ³	0.0734
Massa Jenis Air	1000 kg/m ³	2.6429

```

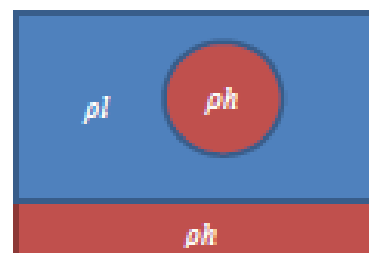
ux[i0] += tau*body_force_x;
uy[i0] += tau*body_force_y;
phi[i0] = 1.0-exp(-rho[i0]);
//.....//
ux[i0] += tau*(-G*phi[i0]*grad_phi_x)/rho[i0];
uy[i0] += tau*(-G*phi[i0]*grad_phi_y)/rho[i0];
    
```

Gambar 5 Potongan kode program

Kode program yang berjalan akan menghasilkan sebuah nilai yang di tapung pada variable f yang kemudian di terjemahkan kedalam gambar visual melalui table warna yang di sediakan, table warna ini digunakan untuk memberikan efek pembeda pada tampilan hasil simulasi yang di lakukan, yaitu membedakan warna massa jenis satu dengan massa jenis yang lainnya, dalam penelitian ini yang memiliki nilai massa jenis tinggi di umpakan cairan, dalam kasus ini simulasi akan di buat kasus di mana cairan akan jatuh ke penampang cairan yang di bawahnya, gambaran kondisi simulasi di gambarkan pada Gambar 5 sketsa simulasi di bawah.

TABLE 2 POTONGAN VARIABLE NIALI CMAP.DAT

0.0000	0.0000	1.0000
0.0000	0.0157	1.0000
.....		
0.0000	1.0000	0.9961
.....		
0.0078	1.0000	0.0000
.....		
1.0000	0.0157	0.0000
1.0000	0.0000	0.0000



Gambar 5 Sketsa simulasi

Karena simulasi cairan 2 fase yang memiliki nilai density yang berbeda maka Nilai yang akan di ambil untuk di gunakan sebagai nilai visual adalah nilai densitas atau nilai ρ (1) yang di dapat pada hasil lattice nilai lattice di dapat dari niali penjumlah seluruh titik arah (f)di LBM untuk menyesuaikan dengan output yang di gunakan nilai Density nya di akarkan agar mendapatkan nilai visual yang sesuai, penulisan kode bisa di lihat pada Gambar 6.

```

macroscopic flow props:
ro = f0now + f1now + f2now + f3now + f4now + f5now + f6now + f7now + f8now;
vx = (f1now - f3now + f5now - f6now - f7now + f8now)/ro;
vy = (f2now - f4now + f5now + f6now - f7now - f8now)/ro;

rho_data[i2d] = ro;

phi_data[i2d] = 1.0-exp(-rho_data[i2d]);
// Set plotting variable to velocity magnitude
plot_data[i2d] = sqrt(rho_data[i2d]);
    
```

Gambar 6 Potongan kode pengambilan nilai densitas

Hasil Simulasi dan Visualisasi

Penerapan formula simulasi akan di buat dalam format CPU dan menggunakan GPU hasil pertama akan di tampilkan menggunakan CPU dan kedua dengan CPU + GPU , Visualisasi dibuat dengan beberapa parameter Jumlah pixel = 640x640pxl dan parameter yang ada table Table 1 parameter simulasi di atas, pengambilan nilai untuk Visualisasi satu fase merupakan nilai dari hasil akar kecepatan pada tiap arah nya , penulisan kodenya terlihat pada Gambar 7.

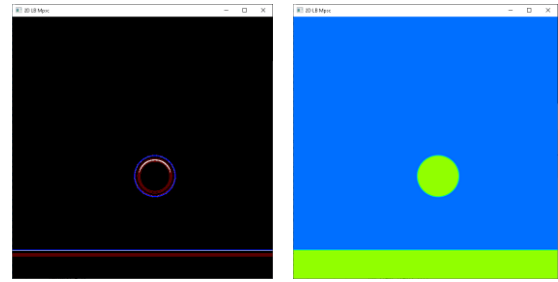
```

// Macroscopic flow props:
ro = f0now + f1now + f2now + f3now + f4now + f5now + f6now + f7now + f8now;
vx = (f1now - f3now + f5now - f6now - f7now + f8now)/ro;
vy = (f2now - f4now + f5now + f6now - f7now - f8now)/ro;

// Set plotting variable to velocity magnitude
plot_data[i2d] = sqrt(vx*vx + vy*vy);
    
```

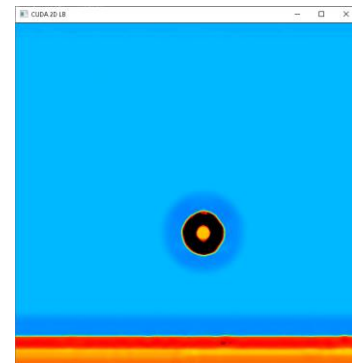
Gambar 1 Potongan gambar nilai arah

Namun jika formula ini di terapkan pada simulasi multi fase dengan CPU maka hasilnya akan seperti pada gambar 8 di bawah , formula pengganti utk menampilkan visualisasi yang lebih sebagai nila penyesuaian pada table warna adalah seperti pada Gambar 6 Potongan kode pengambilan nilai densitas, niali yang di hasilkan lebih menarik dalam hal visual, hasilnya bisa dilihat pada Gambar 8.

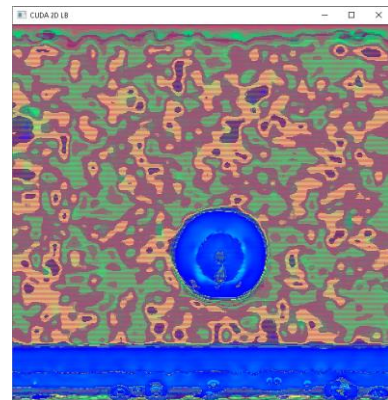


Gambar 2 Hasil Menggunakan Formula Awal Hasil dengan Nilai Density yang Diambil

Namun hasil berbeda jika formula yang di tuliskan menggunakan GPU hasil nya terlihat pada Gambar 9.

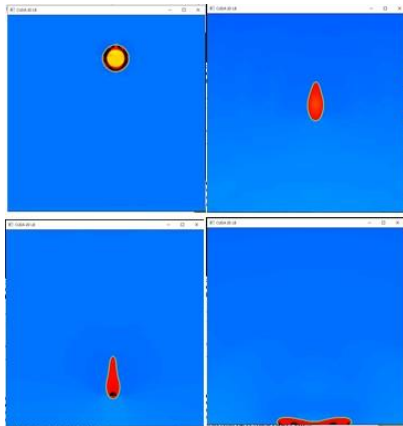


Gambar 9 Perbandingan Visual di GPU Hasil Nilai Densitas

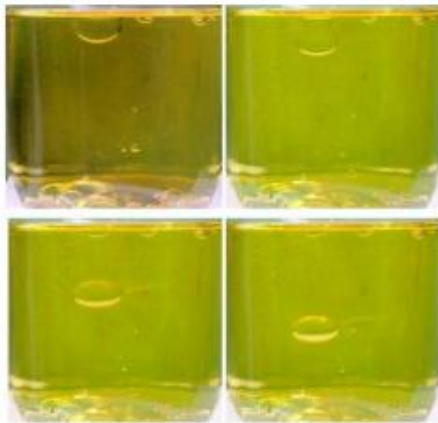


Gambar 10 Visual di GPU dengan Nilai-nilai Kecepatan

Secara numerik Simulasi bisa berjalan dengan baik ini di tunjukkan pada pergerakan cairan yang sudah mirip keadaan nyata, dimana jika cairan yang bertubrukan denagn cairan lain akan menyebabkan gelombang lebih jelasnya bisa di perhatikan pada Gambar 11 yang melihatkan potongan hasil simulasi.



Gambar 31 Potongan Simulasi



Gambar 12 potongan simulasi uji nyata [12]

Sebagai pembanding hasil pengujian secara nyata bisa di lihat pada gambar 12 , penelitian yang di lakukan oleh peneliti sebelumnya, dari hasil pengamatan terlihat hasil simulasi memiliki gambaran menyerupai hasil uji nyata.

Kesimpulan dan Saran

Studi ini menghasilkan kesimpulan tentang pengambilan nilai untuk keperluan visual , di karena ada perbedaan arsitektur , membuat hasil dari perhitungan Metode LBM menjadi berbeda , jika simulasi yang akan di buat adalah multi fase maka nilai yang idela untuk di ambil sebagai nilai visualisasi adalah akar dari Densitasnya, dan ini terbukti dari hasil visualisasi pada Gambar 11 dan Gambar 12, secara visualisasi hasil pada simulasi di CPU lebih baik , karena mendapatkan nilai lebih stabil antar table warna yang digunakan , hasil visual menggunakan GPU tidak terlalu baik jika di bandingkan hasil pada CPU. Jika di lihat dalam hal bentuk yang di buat terlihat sangat tidak semetris pada GPU.

Daftar Pustaka

[1] M. Park, H. Shin, Y. Ho, and S. Ha, “Open-Source-based Visualization of Flight Waypoint Tracking Using Flight Manipulation System,” vol. 2, no. 2, pp. 1–10, 2011.
[2] D. Zhang, K. Papadikis, and S. Gu, “Three-

dimensional multi-relaxation time lattice-Boltzmann model for the drop impact on a dry surface at large density ratio,” *Int. J. Multiph. Flow*, vol. 64, pp. 11–18, 2014.

[3] X. Zhang, M. Huang, Q. Ji, and X. Luo, “Lattice Boltzmann modeling and experimental study of water droplet spreading on wedge-shaped pattern surface,” *Int. J. Heat Mass Transf.*, vol. 130, pp. 857–861, 2019.
[4] S. An, H. Whitney Yu, Z. Wang, B. Kapadia, and J. Yao, “Unified mesoscopic modeling and GPU-accelerated computational method for image-based pore-scale porous media flows,” *Int. J. Heat Mass Transf.*, vol. 115, pp. 1192–1202, 2017.
[5] Q. Chang and T. Yang, “A Lattice Boltzmann method for image denoising.,” *IEEE Trans. Image Process.*, vol. 18, no. 12, pp. 2797–802, 2009.
[6] W. Wang, Z. Liu, Y. Jin, and Y. Cheng, “LBM simulation of droplet formation in micro-channels,” *Chem. Eng. J.*, vol. 173, no. 3, pp. 828–836, 2011.
[7] X. Shan and H. Chen, “Lattice Boltzmann model for simulating flows with multi phases and componentsShan, X., & Chen, H. (1993). Lattice Boltzmann model for simulating flows with multi phases and components. *Physical Review E*, 47(3), 1815–1819. <https://doi.org/10.1103/PhysRevE.>,” *Phys. Rev. E*, vol. 47, no. 3, pp. 1815–1819, 1993.
[8] N. Delbosc, J. L. Summers, A. I. Khan, N. Kapur, and C. J. Noakes, “Optimized implementation of the Lattice Boltzmann Method on a graphics processing unit towards real-time fluid simulation,” *Comput. Math. with Appl.*, vol. 67, no. 2, pp. 462–475, 2014.
[9] M. Bauer and H. Cook, “CudaDMA : Optimizing GPU Memory Bandwidth via Warp Specialization.”
[10] Nvidia, “CUDA,” 2012. [Online]. Available: <https://developer.nvidia.com/about-cuda>.
[11] G. Pullan, “A 2D Lattice Boltzmann flow solver demo,” 2008. [Online]. Available: <http://www.many-core.group.cam.ac.uk/projects/LBdemo.shtml>.
[12] U. N. A. Qohar, F. D. E. Latief, and U. Fauzi, “Pemodelan Fluida Berbeda Fasa dengan Metode Lattice Boltzmann.”