

Tugas Kecil
IF3230 - Sistem Paralel dan Terdistribusi
Sebulan Bersama Oneesan <3
Paralelisasi dengan Open MPI, OpenMP, dan CUDA

Dipersiapkan oleh
Asisten Lab Sistem Terdistribusi



Waktu Mulai
Jum'at, 10 Februari 2023, 10.00 WIB

Waktu Akhir
Jum'at, 24 Februari 2023, 23.59 WIB
Jum'at, 3 Maret 2023, 23.59 WIB
Jum'at, 17 Maret 2023, 23.59 WIB

I. Latar Belakang

Anda adalah seorang *mad scientist* yang baru saja menemukan sebuah alat bernama D-Mail yang dapat mengirimkan pesan sebesar 36 bytes ke masa lalu. D-Mail mampu mengubah garis waktu dengan mengubah aksi yang akan dilakukan oleh penerima pesan. Sebagai seorang *mad scientist*, tentu saja Anda sangat bangga dan penasaran akan potensi yang dapat dihasilkan oleh penemuan ini. Anda mulai mencari orang-orang dan meminta mereka untuk menguji coba dan mengirimkan pesan ke masa lalu sembari mengumpulkan data terkait perubahan-perubahan yang terjadi setelah pesan dikirimkan ke masa lalu. Saat hendak menuju LabPro, secara tidak sengaja Anda bertemu dengan Qifri Faunal B. Qifri adalah seorang *normies* yang **tidak suka menonton anime**, rajin menabung, senang belajar *programming*, dan rajin nge-gym (setiap Senin, Rabu, dan Jumat). Anda mulai menawarkannya untuk mengirimkan pesan ke masa lalu. Dengan antusias ia menyetujui dan mulai mengirimkan pesan ke masa lalu. Tepat setelah Qifri mengirimkan pesan ke masa lalu, Anda mengalami distorsi visual dan meriang ringan. Mulai terlihat sebuah angka aneh di dalam kepala Anda, awalnya Anda mengira angka tersebut hanyalah imajinasi belaka dan mengabaikannya.



Angka aneh yang terlihat dalam kepala Anda

Seminggu kemudian Anda melihat Qifri memposting fotonya bersama anime, tentu saja ini membuat Anda terkejut karena sejatinya ia bukanlah seorang wibu. Anda mulai menyadari perubahan yang terjadi terhadap Qifri akibat pesan masa lalu yang dikirimkannya. Sekarang ia menjadi wibu halu! Selain itu, dia lebih menyukai mata kuliah sistem terdistribusi dibandingkan PBD. Sebagai teman yang baik Anda merasa prihatin terhadap Qifri dan mulai berusaha untuk mengembalikannya ke jalan yang benar, kita sebut misi ini sebagai *operation saving private Qifri*. Setelah diselidiki lebih lanjut Anda mulai sadar, bahwa angka yang muncul tersebut adalah jarak garis waktu antar dunia (sederhananya itu adalah derajat kewarasannya Qifri) semakin mendekati nilai 1, ia akan semakin waras.



Hahaha difriendzone anime lu

Anda tidak ingin menggunakan D-Mail karena hasilnya yang tidak dapat diprediksi, akhirnya Anda memutuskan untuk belajar sistem terdistribusi dengan lebih giat agar Anda memiliki bahan obrolan yang menarik bersama Qifri, mengalihkan pikirannya dan menghentikannya sebelum ia menjadi wibu yang meresahkan. Kerjakanlah tugas berikut untuk membantu Qifri menjadi *normies!*

DISCLAIMER :

Cerita ini hanya fiktif belaka. Jika ada kesamaan nama tokoh, tempat kejadian ataupun cerita, itu adalah kebetulan semata dan tidak ada unsur kesengajaan (udah izin sama orangnya langsung ya goes)
Merokok dapat menyebabkan kanker, serangan jantung, impotensi dan gangguan kehamilan & janin.

II. Deskripsi Tugas

2.1. Overview

Dokumen ini mencakup tiga tugas kecil terpisah dengan topik yang sama yaitu paralelisasi menggunakan framework yang berbeda-beda. Ketiga framework yang akan digunakan adalah

- Open MPI
- OpenMP
- CUDA

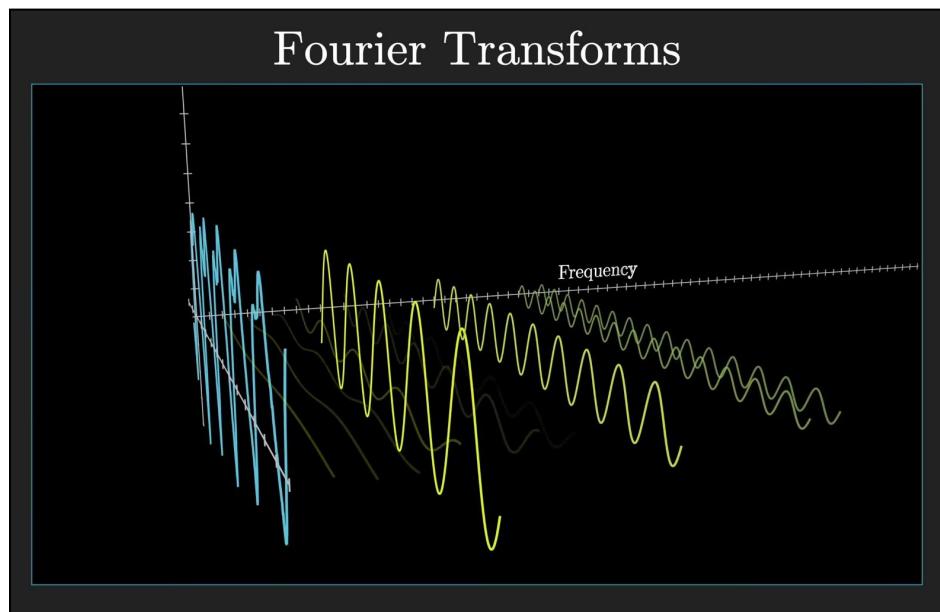
Topik yang akan digunakan sebagai *problem domain* adalah **Discrete Fourier Transform**.

Tugas kecil akan berfokus kepada paralelisasi suatu program menggunakan framework parallel programming.

Kode dasar yang dapat dijalankan secara serial pada CPU akan disediakan pada kit.

2.2. Discrete Fourier Transform

Transformasi Fourier secara singkat merupakan suatu metode untuk mengubah fungsi dari suatu domain ke fungsi domain frekuensi.



3B1B - 1D Fourier Transform

Transformasi Fourier akan membentuk fungsi atau sinyal baru yang membawa informasi dalam domain frekuensi. Pada dunia komputasi, Transformasi Fourier sering digunakan dalam bentuk **Discrete Fourier Transform**. DFT memiliki prinsip yang sama dengan transformasi Fourier, yaitu melakukan transformasi fungsi dari suatu domain ke fungsi dengan domain frekuensi.

Namun DFT menggunakan nilai sampel fungsi dalam jumlah yang terbatas, contohnya rekaman suara yang umumnya diambil dengan 44100hz memiliki arti fungsi suara yang diambil dalam 44100 data dalam satu detik. Atau contoh lain seperti suatu gambar juga memiliki resolusi yang terbatas seperti 1920x1080 pixel.

Tugas kecil akan menggunakan 2D DFT pada matriks $2^N \times 2^N$ sebagai topik utama permasalahan.

2.3. Serial Naive 2D DFT

Algoritma naif 2D DFT dapat diimplementasikan langsung dari definisi DFT berikut

$$F[k, l] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] e^{-2\pi i (\frac{k \cdot m}{M} + \frac{l \cdot n}{N})}$$

Kode serial 2D DFT akan disediakan pada template repository GitHub Classroom.

Untuk operasi bilangan kompleks akan digunakan library yang telah disediakan oleh bahasa C dan C++ dalam header `<complex.h>` atau `<complex>`. Selain kedua *library* tersebut, *framework* tugas, dan *standard library* tidak boleh digunakan. Algoritma serial yang diberikan merupakan *brute force* $O(n^2)$ untuk mencari transformasi DFT.

Tugas utama dari tugas kecil ini adalah mengimplementasikan versi paralel dari DFT sederhana menggunakan beberapa framework paralel. Jika ingin, diperbolehkan menggunakan FFT, **tetapi wajib paralel**.

Semua test case yang akan digunakan pada tugas kecil terdapat pada folder **test_case**.

Berikut adalah sampel waktu eksekusi kode serial yang dapat digunakan sebagai pembanding. Kode yang digunakan: C dengan **gcc**

Command yang digunakan : `time ./serial_dft < testcase.txt > output.txt`
Dieksekusi pada env: **Win10 - WSL2 Ubuntu 20.04 - Ryzen 4800H (Full TDP chassis)**

N	Waktu	Rasio waktu
32	0.039 detik	-
64	0.628 detik	15.75x
128	12.7 detik	20.16x
256	3 menit 54 detik	18.43x
512	71 menit 6.124 detik	18.23x

III. Tugas Kecil 1 : Open MPI



0.409420 α

Dengan menggunakan Open MPI, paralelisasi algoritma 2D DFT serial tersebut. Skema paralelisasi dibebaskan. Kode contoh untuk MPI telah disediakan pada template repository pada `src/sample/mpi.c`

Untuk tugas kecil Open MPI, kode akan dijalankan pada server dengan 4 node.

Tugas kecil ini akan berfokus pada paralelisasi menggunakan distributed memory & message passing.

Tidak diperbolehkan menggunakan multithreading pada tugas kecil Open MPI.

Gunakan perintah berikut untuk melakukan kompilasi source code C Open MPI

```
mpicc mpi.c -o mpi
```

Gunakan perintah berikut untuk menjalankan hasil kompilasi program dengan dua node

```
mpirun -n 2 ./mpi
```

Untuk menjalankan diserver, buatlah sebuah hostfile yang berisikan IP private node-node yang ingin dijalankan. Gunakan perintah seperti berikut untuk menggunakan hostfile

```
mpirun -hostfile hostfile ./mpi < 128.txt > out.txt
```

Pengumpulan

Simpan **semua kode tugas kecil 1** dan **README.md** untuk tugas kecil 1 pada sebuah folder pada folder src bernama **open-mpi**

Deadline Tugas Kecil 1: Open MPI adalah **Jumat, 24 Februari 2023 Pukul 23.59 WIB**

IV. Tugas Kecil 2 : OpenMP



0.456903 α

Dengan menggunakan OpenMP, paralelisasi algoritma 2D DFT serial tersebut. Skema paralelisasi dibebaskan. Kode contoh untuk OpenMP telah disediakan pada template repository pada `src/sample/mp.c`.

Pada demo OpenMP, program akan dijalankan pada salah satu node server.

Tugas kecil ini akan berfokus pada paralelisasi dengan shared memory.

Tidak diperbolehkan menggunakan multiprocessing pada tugas kecil OpenMP.

Gunakan perintah berikut untuk melakukan kompilasi source code C OpenMP

```
gcc mp.c --openmp -o mp
```

Program dapat dijalankan seperti executable seperti biasa

```
./mp < 128.txt > out.txt
```

Pengumpulan

Simpan **semua kode tugas kecil 2** dan **README.md** untuk tugas kecil 2 pada sebuah folder pada folder src bernama **open-mp**

Deadline Tugas Kecil 2: OpenMP adalah **Jumat, 3 Maret 2023 Pukul 23.59 WIB**

V. Tugas Kecil 3 : CUDA



0.523307 α

Dengan menggunakan CUDA, paralelisasi algoritma 2D DFT serial tersebut. Skema paralelisasi dibebaskan. Kode contoh untuk CUDA telah disediakan pada template repository pada `src/sample/cuda.cu` dan `src/sample/cuda_colab.ipynb`

Kompilasi kode dapat menggunakan perintah berikut

```
nvcc cuda.cu -o cuda
```

Pastikan sistem yang digunakan memiliki **Nvidia GPU** yang telah terkonfigurasi dengan baik. Jika tidak memiliki sistem dengan spesifikasi tersebut, gunakan [Google Colaboratory](#). Contoh notebook sudah disediakan.

Untuk demo tugas kecil 3, diperbolehkan untuk menampilkan hasil eksekusi pada local atau Colab. Tampilkan spesifikasi GPU jika eksekusi di *local* (dapat dari Nvidia Control Panel atau program `nvidia-smi`).

Pengumpulan

Simpan **semua kode tugas kecil 3** dan **README.md** untuk tugas kecil 3 pada sebuah folder pada folder src bernama **cuda**

Deadline Tugas Kecil 3: CUDA adalah **Jumat, 17 Maret 2023 Pukul 23.59 WIB**

VI. Penilaian

Untuk program, ketiga tugas akan dinilai secara terpisah dan dirata-rata sebagai nilai kelompok tugas kecil. Demo untuk ketiga tugas kecil akan dinilai menjadi satu. Nilai demo bersifat individu. Peer assessment akan disatukan pada akhir semester. Teknik penilaian (weighting dengan poin peer) akan sama seperti tugas-tugas laboratorium sistem paralel dan terdistribusi lainnya.

Tugas kecil berfokus kepada paralelisasi, bukan algoritma DFT sendiri. Optimisasi *overhead & speedup* paralel merupakan subjek utama pada tugas kecil ini.

- **Program (60)**
 - a. Implementasi paralelisasi dengan **Open MPI / OpenMP / CUDA (25)**
 - b. Program paralel dapat berjalan dan menghasilkan output yang *reasonable* vs serial **(15)**
 - c. Kinerja program paralel sama atau lebih cepat dibandingkan serial (diperbolehkan untuk membandingkan dengan serial referensi / local / server) **(20)**
 - 1 test case bernilai **(5)**, demo akan membandingkan 4 test case
- **Program - Bonus (up to 5)**
 - Optimisasi hingga speedup 8x pada salah satu test case
 - Serial & paralel FFT
 - Kreativitas & optimisasi lainnya, dapat diperlihatkan ketika demo
- **Demo (40)**

VII. Pengumpulan & Deliverables

1. Pengerajan tugas dilakukan dengan membuat sebuah repository pada [Assignment di Github Classroom "Lab Sister 20"](#). Pastikan bahwa project visibility repository kelompok Anda diatur menjadi private selama pengerajan.
2. Gunakan link assignment diatas untuk membuat repository (Gunakan tombol Can't find your name? Skip to the next step).

Template untuk tugas kecil secara terpisah tersedia pada repository GitHub berikut: [repository template](#)

3. Pengerajan tugas dilakukan berkelompok dengan masing-masing kelompok maksimal 4 orang dari kelas yang sama. Isi anggota kelompok pada [sheets berikut](#). Pengisian anggota kelompok paling lambat **13 Februari 2023 pukul 10.00 WIB**.

Note: Perlu diperhatikan bahwa kelompok akan digunakan untuk pengerajan tugas kecil dan tugas besar

4. Setiap kelompok diwajibkan untuk membuat tim dalam Github Classroom dengan nama yang sama pada spreadsheet kelompok.
5. Pengerajan tugas kecil dapat memanfaatkan bahasa pemrograman **C/C++**
6. Berikut adalah struktur folder dari template:
 - a. *Folder src* yang berisi kode program, letakkan kode setiap tucil pada tempatnya
 - b. *Folder bin* yang berisi hasil kompilasi
 - c. *Folder test_case* yang berisi *file-file test_case* yang akan digunakan.
 - Untuk demo, gunakan minimal **4 test case** yang tersedia
7. Di setiap *folder* tucil (**open-mpi**, **open-mp**, **cuda**) wajib terdapat *file README.md*. Setiap **README.md** akan berisi:
 - a. Penjelasan cara kerja paralelisasi program
 - b. Penjelasan cara program Anda membagikan data antar-proses atau antar-thread dan alasan pemilihan skema pembagian data.
8. Karena untuk ketiga tucil menggunakan repository yang sama, gunakan release tag git (seperti tugas besar sistem operasi) untuk melakukan pengumpulan. **Jika tag tidak ada maka akan dianggap tidak mengerjakan tugas kecil.** Berikut adalah ketentuan tag

Tugas Kecil	Tag	Deadline
Open MPI	tucil-1	24 Februari 2023 – 23.59 WIB
OpenMP	tucil-2	3 Maret 2023 – 23.59 WIB
CUDA	tucil-3	17 Maret 2023 – 23.59 WIB

9. Gunakan server yang disediakan untuk melakukan tugas kecil Open MPI dan OpenMP yang telah dibuat. Server dapat diakses dengan menggunakan kunci SSH yang diberikan.

Kunci SSH dapat di download di [link berikut ini](#). Pastikan mendownload kunci SSH yang sesuai dengan kelas dan kelompok. "kXX-YY" dengan XX diisi 2 digit nomor kelas (01-03) dan YY diisi 2 digit nomor kelompok (pada kolom No Kel) pada sheets kelompok. Perhatikan pula bahwa username pada server juga mengikuti aturan "kXX-YY".

IP Publik	IP Privat	DNS
34.126.112.192	10.148.0.38	sister1.frederon.com
34.142.238.239	10.148.0.39	sister2.frederon.com
34.126.64.121	10.148.0.37	sister3.frederon.com
35.240.228.227	10.148.0.36	sister4.frederon.com

IP publik digunakan untuk menyambungkan dari komputer masing-masing ke server. Sedangkan IP private digunakan untuk komunikasi antara server (digunakan pada MPI hostfile).

Sebelum melakukan eksekusi program, lakukan langkah-langkah berikut:

1. Hubungkan dengan salah satu server menggunakan SSH melewati IP publik
2. Dalam kondisi masih terhubung dengan salah satu server, lakukan SSH ke semua server dengan IP private
3. Putuskan koneksi SSH ke server dan kembali ke local. Buat [file hostname](#) dengan isi semua IP private dipisahkan dengan *newline*
4. *Compile & salin executable* dan test case ke semua server (menggunakan utilitas [scp](#))
5. Lakukan SSH lagi ke server utama dan jalankan program dengan [mpirun](#) untuk MPI

Note: Server akan di-provision untuk digunakan mulai **Jumat, 17 Februari 2023 pukul 13.00**. Sebelum tanggal tersebut, Anda dapat menggunakan *environment* lokal untuk menguji kode yang dibuat.

Apabila Anda menggunakan WSL untuk melakukan SSH ke server, edit *permission* SSH key dengan chmod dengan *permission*:

- 600 untuk **id_rsa**
- 644 untuk **id_rsa.pub**

Selain itu, harap untuk menyimpan SSH key anda pada *directory* WSL Anda (`/home/[username]/`). Bukan pada *directory* Windows (`/mnt/[drive letter]/[any]**/*`). Hal ini disebabkan karena *permission* ala Linux hanya bisa dilakukan pada *filesystem* Linux (dalam direktori WSL).

10. Hasil pekerjaan yang dinilai adalah hasil dari commit terakhir sebelum deadline. Terdapat 3 deadline berbeda untuk setiap jenis tugas kecil, yaitu:
 - a. Tugas Kecil 1: **Jumat, 24 Februari 2023 Pukul 23.59 WIB**
 - b. Tugas Kecil 2: **Jumat, 3 Maret 2023 Pukul 23.59 WIB**
 - c. Tugas Kecil 3: **Jumat, 17 Maret 2023 Pukul 23.59 WIB**
11. Apabila ada pertanyaan lebih lanjut, jangan lupa untuk selalu kunjungi [sheet QnA](#).
12. Asisten menyediakan asistensi terkait tugas kecil yang sifatnya opsional. Asistensi akan bersifat seperti QnA dalam skala kelompok, maka persiapkan pertanyaan terlebih dahulu sebelum melakukan asistensi. Setelah mengisi form, asisten akan mengontak perwakilan kelompok **paling lambat H-1 asistensi**. Apabila asisten belum mengontak sampai hari-H, dapat diasumsikan tidak ada asisten yang sedang tersedia. Gunakan form yang sama untuk men-request asistensi lagi, jika masih dibutuhkan asistensi. Apabila Kelompok Anda ingin melakukan asistensi dapat mengisi [Form Asistensi berikut](#).
13. **Catatan penting:** Jika diketahui terdapat kode yang sama dengan repository di internet, **maka akan dianggap melakukan kecurangan**. Alasan menggunakan fitur kode autocomplete seperti Github Copilot yang melakukan copas akan diabaikan.
14. Segala kecurangan baik sengaja dan tidak disengaja akan ditindaklanjuti oleh pihak asisten, yang akan berakibat sanksi akademik ke setiap pihak yang terlibat.
15. Akan diadakan demo setelah semua deadline tugas kecil berakhir. Sheet untuk demo akan dirilis paling lambat **6 Maret 2023**.
16. Demo tugas akan dapat dilakukan secara sinkron atau asinkron bergantung pada asisten yang Anda pilih di sheet demo (**TBD**). Identitas dan keterangan asisten untuk demo akan dirilis di sheet demo (**TBD**). Berikut merupakan keterangan jenis demo yang ditawarkan setiap asisten:
 - a. Apabila mendapatkan asisten dengan skema demo **sinkron**, maka Anda **wajib** melakukan demo secara sinkron dengan mengisi jadwal di sheet asisten tersebut. Pelaksanaan demo kurang lebih akan sama seperti guideline video demo asinkron. Sinkron *by default* menggunakan *meeting* secara *online*. Jika ingin *offline*, diskusikan dengan asisten terkait.
 - b. Apabila mendapatkan asisten dengan skema demo **asinkron**, maka Anda **wajib** melakukan demo dengan membuat video dengan tata cara pembuatan demo di

[bagian ini](#). Video yang dibuat berdurasi maksimal 30 menit (percepat video jika dibutuhkan). Lokasi pengumpulan demo akan diumumkan di kemudian hari.

VIII. Tata Cara Pembuatan Demo

Berikut merupakan tata cara demo untuk semua tugas kecil:

1. Sebelum mulai, jalankan perintah `git status` dan `git log`. Tidak menjalankan kedua perintah ini dapat berakibat pada tugas tidak dinilai.
2. Lakukan kompilasi program, direkomendasikan pada server
3. Uji test case yang digunakan terdapat pada *repository* di *folder* `test_case`. Pengujian dilakukan dengan mengeksekusi kode baseline (serial) dan yang dibuat (paralel). Test case yang diuji terdiri atas **4 test case** yang berbeda.

Untuk serial, direkomendasikan untuk menyimpan output sebelum demo. Dan waktu serial dapat menggunakan referensi yang diberikan pada spesifikasi tugas kecil.

4. Jalankan program pada environment sesuai dengan tugas kecil
 - a. Serial - Server / Local
 - b. Open MPI - Server
 - c. OpenMP - Server
 - d. CUDA - Colab / Local
5. Sembari menunggu eksekusi program, tunjukkan bagian kode program Anda yang memanfaatkan paralelisasi. Jelaskan mekanisme paralelisasi program serta cara program membagikan data antar-proses (seperti yang di `README.md`)
6. Tunjukkan hasil average eksekusi program paralel dan bandingkan dengan hasil eksekusi dari program serial. Tunjukkan juga sebagian elemen hasil matriks serial dan paralel. Jika ada perbedaan pada hasil, jelaskan alasannya
7. Bandingkan waktu eksekusi program Anda (paralel) dengan waktu eksekusi program serial. Untuk menguji waktu eksekusi dapat digunakan command `time` (Contohnya: `time ./mp < 128.txt > out.txt`)

IX. Tips

1. [Cara instalasi OpenMP dan Open MPI di Mac](#)
2. **Hati-hati dengan penggunaan stack space yang berlebihan**

X. Referensi

DFT

1. [Rinaldi Munir - Transformasi Citra 2022 - Transformasi Fourier](#)
2. [Rinaldi Munir - IF4073 Interpretasi dan Pengolahan Citra 22/23](#)
3. [UNIVR - 2D DFT](#)
4. [3Blue1Brown - But what is the Fourier Transform?](#)
5. [Reducible - Discrete Fourier Transform](#)

MPI

6. [Open MPI - Docs](#)

MP

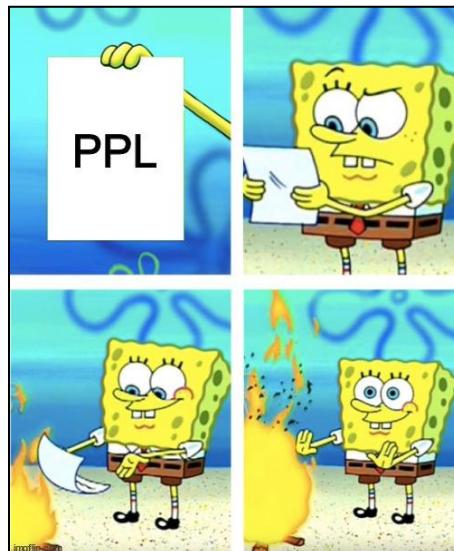
7. [OpenMP - Resources - Tutorials](#)

CUDA

8. [CUDA-tutorial-readthedocs - Tutorial 1 - Hello World](#)

~~🐴🐴🐴🐴 Selamat kerja lembur bagai CUDA🐴🐴🐴🐴~~

Extras



Qifri - Grayscale - Ideal Low Pass Filter in Frequency Domain
Grayscale conversion → DFT → ILPF → Inverse DFT