Nama = Muhammad Wafiq Kamaluddin

NRP = 2210181042

A. Tujuan

1. mahasiswa mampu memahami dan menerapkan prinsip transformasi fourir dalam pengolahan citra.

B. Dasar Teori.

Metode fourir adalah metode untuk mengubah sebuah fungsi dalam domanin waktu menuju kedomain frequensi. Sebuah gambar juga merupakan fungsi yang tersusun atas pixel dalam koordinat x dan y sehingga memungkinkan dilakukanya transformasi fourir pada gambar tersebut.

Keunggulan dari domain frekuensi iyalah waktu eksekusi pada gambar yang lebih cepat, jika dibanding dengan domain waktu, dengan syarat operasi yang dilakukan merupakan operasi kompleks. Akan tetapi, jika hanya operasi sederhana lebih dianjurkan untuk melakukan operasi pada domain waktu.

C. Kode program.

#include "opencv2/core.hpp"

#include "opencv2/imgproc.hpp"

#include "opencv2/imgcodecs.hpp"

#include "opencv2/highgui.hpp"

#include <iostream>

using namespace cv;

using namespace std;

int main(int argc, char \*\* argv)

{

Mat I = imread( "eth.jpeg" , IMREAD\_GRAYSCALE);

Mat I2 = imread( "ethku.jpeg" , IMREAD\_GRAYSCALE);

if( I.empty()){

cout << "Error opening image" << endl;

return EXIT\_FAILURE;

}

Mat padded,padded2; //expand input image to optimal size

int m = getOptimalDFTSize( I.rows );

int n = getOptimalDFTSize( I.cols ); // on the border add zero values

int m2 = getOptimalDFTSize( I2.rows );

int n2 = getOptimalDFTSize( I2.cols ); // on the border add zero values

copyMakeBorder(I, padded, 0, m - I.rows, 0, n - I.cols, BORDER\_CONSTANT, Scalar::all(0));

copyMakeBorder(I2, padded2, 0, m2 - I2.rows, 0, n2 - I2.cols, BORDER\_CONSTANT, Scalar::all(0));

Mat planes[] = {Mat\_<float>(padded), Mat::zeros(padded.size(), CV\_8U)};

Mat planes2[] = {Mat\_<float>(padded2), Mat::zeros(padded2.size(), CV\_8U)};

Mat complexI;

merge(planes , 1,complexI); // Add to the expanded another plane with zeros

dft(complexI, complexI); // this way the result may fit in the source matrix

split(complexI, planes); // planes[0] = Re(DFT(I), planes[1] = Im(DFT(I))

Mat complexI2;

merge(planes2 , 1,complexI2); // Add to the expanded another plane with zeros

dft(complexI2, complexI2); // this way the result may fit in the source matrix

split(complexI2, planes2); // planes[0] = Re(DFT(I), planes[1] = Im(DFT(I))

//untuk domain waktu

Mat Add\_time, Sub\_time;

bitwise\_or(I,I2, Add\_time);

bitwise\_and(I,I2,Sub\_time);

bitwise\_not(Sub\_time,Sub\_time);

// untuk domain frekuensi

Mat Add\_freq, Sub\_freq;

Add\_freq = planes[0] + planes2[0];

Sub\_freq = planes[0] - planes2[0];

Mat back\_to\_time[4],awal,awal2;

dft(Add\_freq, back\_to\_time[0], cv::DFT\_INVERSE);

normalize(back\_to\_time[0], back\_to\_time[0], 0, 1, CV\_MINMAX);

dft(Sub\_freq, back\_to\_time[1], cv::DFT\_INVERSE);

normalize( back\_to\_time[1], back\_to\_time[1], 0, 1, CV\_MINMAX);

dft(planes[0], awal, cv::DFT\_INVERSE);

normalize( awal, awal, 0, 1, CV\_MINMAX);

dft(planes2[0], awal2, cv::DFT\_INVERSE);

normalize( awal2, awal2, 0, 1, CV\_MINMAX);

// Show the result

imshow("Input Image 1 " , I);

imshow("Input Image 2 " , I2);

imshow("image1 sesudah inverse fourir", awal);

imshow("image2 sesudah inverse fourir", awal2);

// imshow("spectrum magnitude", magI);

imshow("image1 saat fourir transform",planes[0]);

imshow("image2 saat fourir transform",planes2[0]);

imshow("image adding domain waktu", Add\_time);

imshow("image subracting domain waktu", Sub\_time);

imshow("image adding domain frekuensi", Add\_freq);

imshow("image subtracting domain frekuensi", Sub\_freq);

imshow("adding fourir in time", back\_to\_time[0]);

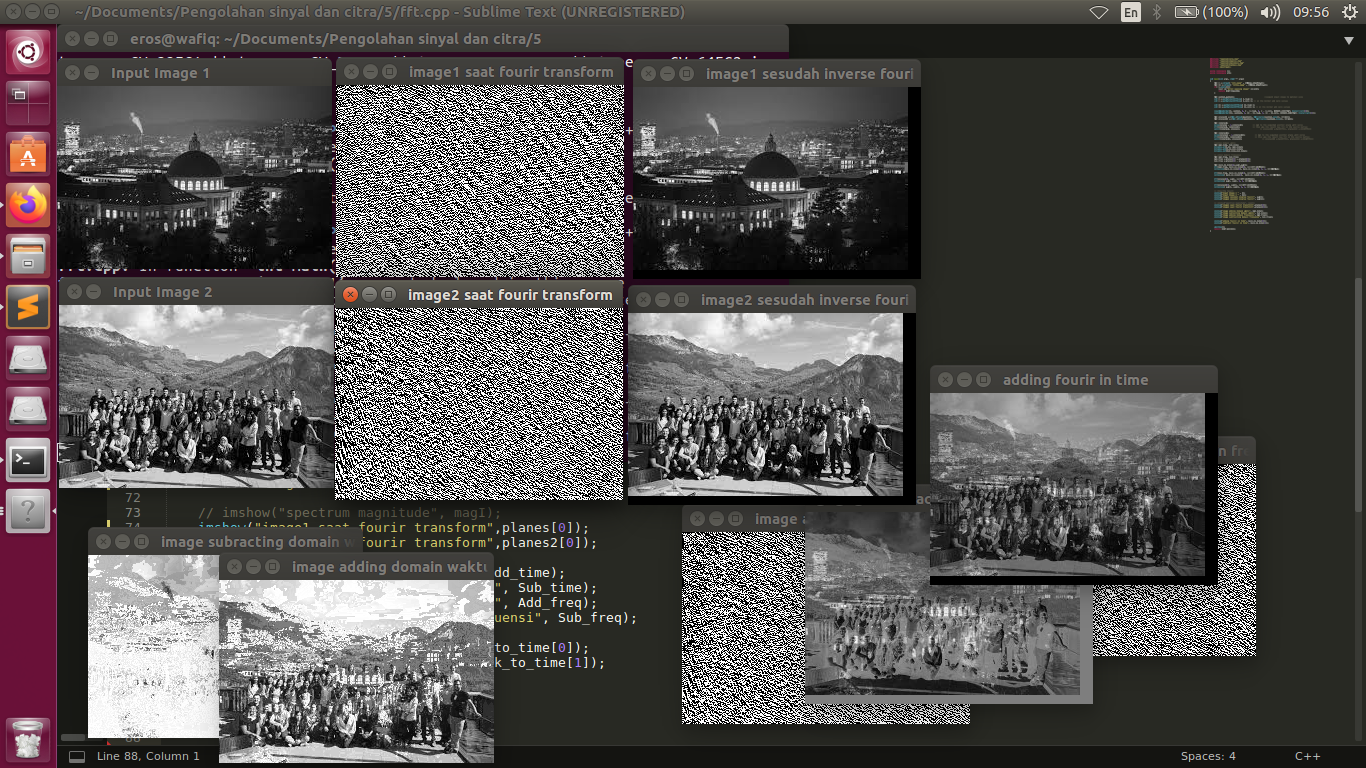
imshow("subtract fourir in time", back\_to\_time[1]);

waitKey();

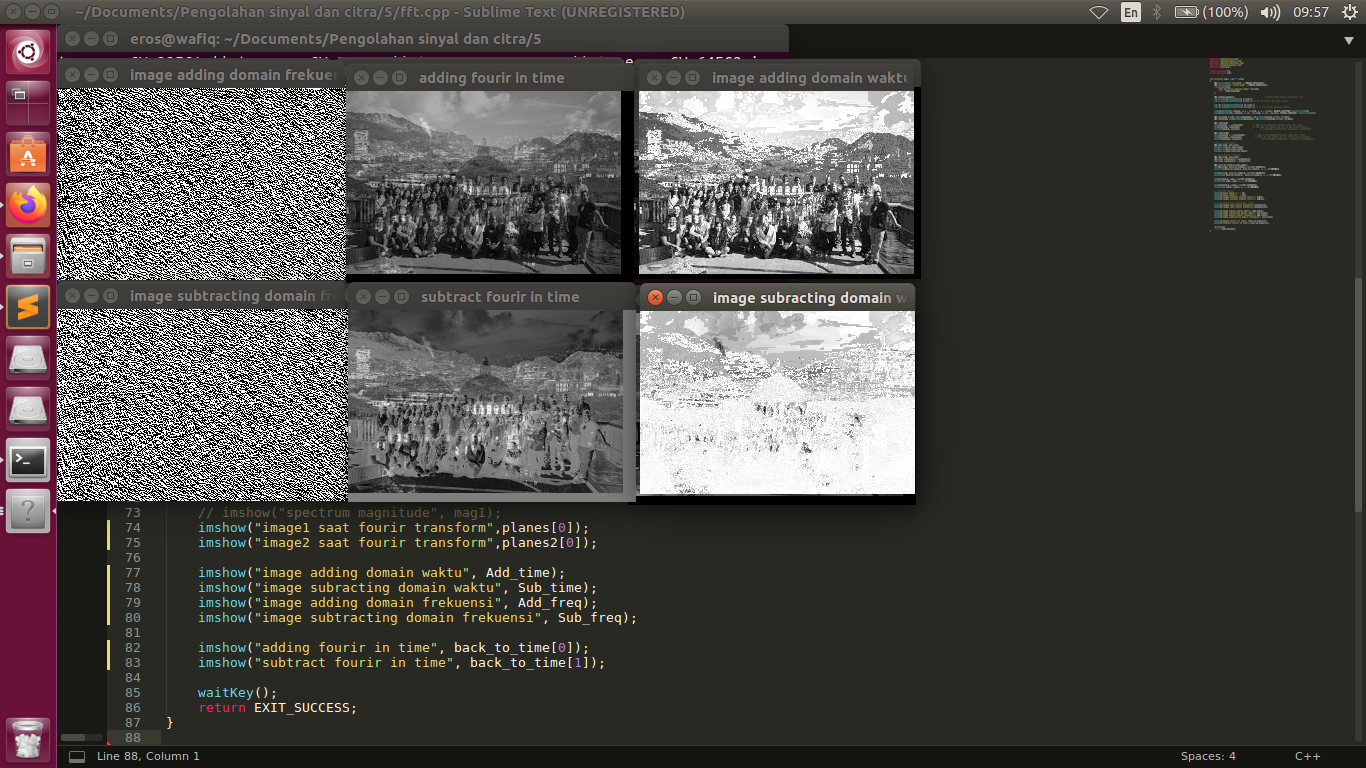
return EXIT\_SUCCESS;

}

D. Hasil.



Dua gambar yang saya ubah ke domain frekuensi dan saya kembalikan lagi ke domain waktu.



Saya lalukan operasi penjumlahan bit pada masing masing domain. Bagian tengah adalah hasil dari pengembalian operasi fourir ke domain waktu.