

DAFTAR PUSTAKA

- Batuhan Hangün. (n.d.). *PERFORMANCE EVALUATION OF A PARALEL IMAGE ENHANCEMENT TECHNIQUE FOR DARK IMAGES ON MULTITHREADED CPU AND GPU ARCHITECTURES*.
- Besari, A.R. A. (n.d.). *24 JAM BELAJAR IOT DENGAN RASPBERRY PI - TINGKAT DASAR - REV 1.Pdf*.
- B.S.,R.T.(R)(MR)(CT) Wm. Faulkner. (1996). *Basic Principles of MRI*. OutSource, Inc.
- Elliott, D. F. (1987). Untitled. Retrieved March 5, 2023, from https://jntukucen.ac.in/ebook_files/134.pdf
- ELSTER LLC. (2017). *k-space - Questions and Answers in MRI*. MRI Questions. Retrieved March 5, 2023, from <https://mriquestions.com/k-space-basic.html>
- Gonzalez-Huitron a, V. (2021). *Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4*. <https://doi.org/10.1016/j.compag.2020.105951>
- Hatsa, M. R. (2017). *Penerapan Komputasi GPGPU untuk Teknik Rekonstruksi Citra Medis Pengisian Nol*. Skripsi Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin.
- Johan Overweg. (n.d.). *MRI main field magnets*. Philips Research.
- Lyons,, R. G. (2011). *Untitled*. Mikrocontroller.net. https://www.mikrocontroller.net/attachment/341426/Understanding_digital_signal_processing.pdf
- Mulya, M. (2013). *Penerapan Multi-threading untuk Meningkatkan Kinerja Pengolahan Citra Digital*.
- Nasir, B. S. M. (2020). *EVALUASI KINERJA MINI PC PADA REKONSTRUKSI CITRA MEDIS MRI (magnetic resonance imaging)*.
- Notosiswoyo, M. (2004). *Pemanfaatan Magnetic Resonance Imaging (MRI Sebagai Sarana Diagnosa Pasien)*. <http://repository.litbang.kemkes.go.id/1125/1/901-986-1-PB.pdf>.

pjbevel. (2010). *Fourier Transform*. Retrieved March 5, 2023, from <http://theFouriertransform.com/>

Rajaraman et al. (2014). *Multithreading Implementation in a Single Core TMS*.

Triwiyanto. (2022). *Embedded Machine Learning Using a Multi-Thread Algorithm on a Raspberry Pi Platform to Improve Prosthetic Hand Performance*.

Lampiran 1 Waktu eksekusi Rekonstruksi Citra MRI menggunakan program serial

Ukuran <i>K-space</i>	Waktu Eksekusi (ms)								
	Percobaan	<i>GetSpecificSlice Data</i>	<i>DoZero Filled</i>	<i>FFT (ReconsData)</i>	<i>FFTSHIFT (Recons)</i>	<i>DORSS (Recons)</i>	<i>DataScalling (Recons)</i>	<i>CPUTimingOf RawData</i>	<i>CPU Timing</i>
512	1	74,640038	54,389954	632,51709	266,023865	1059,643066	13,587952	539,321106	3144,123047
	2	73,562812	53,740978	631,526001	266,513824	1508,013916	13,624191	605,520935	3152,503662
	3	74,289085	53,569077	630,632141	265,996	1509,827148	13,55195	598,127869	3145,643311
	4	73,149918	54,549934	633,037109	267,730011	1510,986816	13,56411	604,982117	3158
	5	73,280098	53,662064	636,35614	268,052094	1514,611938	13,489962	607,146973	3166,599365
	6	74,318886	53,498032	632,138977	270,380981	1515,593994	13,524055	595,647095	3155,102051
	7	73,81105	53,69091	634,335999	261,850128	1507,043091	13,630152	610,11908	3154,480469
	8	73,925972	53,73597	638,047913	638,047913	1513,18103	13,519049	606,391049	3168,945801
	9	73,265076	53,637981	635,906006	267,784119	1504,521118	13,521,118	607,718933	3156,362305
1024	1	73,570969	57,183983	4175,578125	1053,500854	5978,349609	54,078102	620,934937	12013,19727
	2	73,735954	53,593872	4178,128418	1050,694945	5978,18457	53,969147	613,12915	12001,43652
	3	73,342087	53,79081	4178,307617	1058,98877	5944,350098	53,952934	616,542114	11948,27344
	4	75,206993	53,58696	4157,464844	1049,468994	5964,608398	53,853035	613,641968	11967,83106
	5	75,16098	53,482056	4169,626953	1062,536011	5980,541016	53,938866	624,308105	12019,59375
	6	75,222015	53,712128	4131,002441	1050,300781	5949,106445	54,075005	609,592896	11923,01172
	7	73,051926	53,647041	4212,187988	1057,114868	6019,276367	53,879021	600,435974	12069,59277
	8	73,608871	53,488968	4172,783691	1060,895874	5973,869141	53,977013	619,579102	12008,20313
	9	73,358063	53,603172	4178,98877	1054,606934	5967,697266	53,915977	614,720093	11996,88965
2048	1	73,681114	54,105995	20811,6875	4216,59082	23676,71875	214,761017	591,585876	49639,13281
	2	73,059082	54,188965	20758,49023	4268,01123	23751,95312	214,919098	590,044983	49710,66797
	3	73,709015	56,079865	20712,87305	4284,084473	23754,40039	215,427872	593,546875	49690,12109
	4	73,46582	54,11792	20932,07031	4205,783691	23752,38477	215,77597	590,758057	49824,35156
	5	73,870895	55,704117	20723,47852	4237,474121	23622,29883	215,437897	590,528015	49518,79297
	6	72,89505	57,451012	20598,20898	4271,21582	23615,76172	215,639114	592,155945	49423,32031
	7	73,45295	54,111958	20685,17969	4244,955078	23761,07422	214,9712069	591,298096	49625,03906
	8	72,777031	54,151058	20660,93555	4191,313965	23627,5625	215,137955	590,054016	49411,92969
	9	73,414803	58,366058	20117,97684	4165,030762	23584,34766	215,610031	589,217896	48803,96484

Lampiran 2 Data Utilitas CPU *Kspace* ukuran 512 x 512

Percobaan	Utilitas CPU 512 x 512			
	Serial	Paralel (Jumlah <i>Thread</i>)		
		1 <i>Thread</i>	2 <i>Thread</i>	4 <i>Thread</i>
1	77,7	11,6	50,5	21,9
	99,7	100	135,8	148,8
2	98,3	19,2	78,5	43,7
	99,7	99,3		186,1
	72,8	2	32,3	19,9
3	99,7	99,3	119,5	146,8
		100		
4	82,8	30,8	21,3	27,6
	99	99,7	114,6	149
	31,1	5,3	4,6	31,8
5	99,3	100	100	152,3
		99,7		
6	91,1	27,2	27,6	23,8
	100	99,7	114,9	143,7
7	95,4	29,5	28,8	27,6
	99	100	116,6	148
	3	8	25,5	27,9
8	99,7	99	115,6	144,9
	100			
9	31,9	7,3	24,3	27,2
	99,7	100	115,9	143,4
10	100	24,5	18,9	27,9
	99,7	100	113,3	146,5
Rata-Rata	84,74286	61,91364	71,5	89,44

Lampiran 3 Data Utilitas CPU *Kspace* ukuran 1024 x 1024

Percobaan	Utilitas CPU 2048 x 2048			
	Serial	Paralel (Jumlah <i>Thread</i>)		
		1 <i>Thread</i>	2 <i>Thread</i>	4 <i>Thread</i>
1	82,9	70,5	98	98,7
	99,7	99,3	181,1	280,8
		99,3	166,6	
2	91,7	89,7	6	95,4
	100	99,7	99,3	272,8
	100	100	181,7	
3			175,2	
	52,3	55,5	67,8	70,9
	100	100	155	276,7
4				272,8
	99,7	99,7	163,5	
	50,8	74,8	74,1	86
5				
	99,7	100	161,9	281,5
	100	99,7	163,8	
6				
	55,1	11,6	66,1	79,1
	100	100	156	279,4
7	99,7	99,7	162,9	
		100		
	74,4	24,9	85,8	94
8				
	99,3	100	173,4	267,5
	100	99,7	163,2	
9				
		100		
	81,2	87,1	94,4	7
10				
	99,7	99,7	181,4	99,7
	100	99,7	162,9	272,8
11				
	88,4	99,7	60,5	67,4
	100	100	147,4	258,3
12				
	100	99,7	163,5	278,7
	53,8	71,8	97,4	90,4
13				
	100	99,7	182,4	277,1
	99,7	99,7	165,9	
14				
	64,9	38,1	95,7	78,8
	100	100	183,4	278,4
15				
	99,3	100	164,2	
Rata-Rata	89,38966	88,10313	135,5	181,0522

Lampiran 4 Data Utilitas CPU *Kspace* ukuran 2048 x 2048

Percobaan	Utilitas CPU 2048 x 2048			
	Serial	Paralel (Jumlah Thread)		
		1 Thread	2 Thread	4 Thread
1	38,1	8,3	194,1	55,1
	100	100	199,7	99,7
	100	100	113,6	
	79,5	100	130,1	
2			200	228,1
	99,7	6	16,2	35,5
	100	100	99,7	99,7
		100	153,8	191,7
3		100	199,7	256,3
	67,8	81,2	37,2	35,5
	100	99,7	99,7	99,3
	99,3	100	187	394,7
4			200	340,4
	98,3	99,7	32,2	36,2
	100	100	99,3	100
	100	88,7	171,4	335,2
5			199,7	387,5
	87,7	11,3	38,9	5,6
	99,7	100	99,7	99,3
	100	99,7	189,7	242,2
6		100	199,3	295,7
	14,3	9,6	14	39,1
	100	99,7	99,7	100
	99,7	100	160,8	359,6
7	100	100	199,3	393,1
	95,7	28,7	36,5	42,5
	100	99,3	100	99,7
	100	100	177,1	331,9
8		99,7	199,7	398,3
	12,3	19,2	9,3	35,8
	99,7	99,7	100	100
	100	100	153	285
5	99,7	99,7	199,7	371,9
	71,4	28,7	94,1	53,3
	99,7	100	118,6	110,3
	100	99	199	383,2
10		99,7	200	379,6
			199,3	
	91,7	29,9	25,9	18,9
	99,3	99	99,7	99,7
	100	100	165,1	300
		100	200	359,7
Rata-Rata	89,175	81,75	133,6143	189,9921

Lampiran 5 Data Utilitas RAM *Kspace* ukuran 512 x 512

Percobaan	Utilitas RAM 512 x 512			
	Serial	Paralel (Jumlah <i>Thread</i>)		
		1 <i>Thread</i>	2 <i>Thread</i>	4 <i>Thread</i>
1	10,8	2,4	10	5
	1,4	1,3	1,4	1,3
2	9,5	4,3	10,6	10,1
	1,4	1,3		1,4
3	17,1	0,3	7,5	4,5
	1,4	7,8	1,4	1,3
4		1,4		
	10,9	6,6	4,8	6,4
5	1,4	1,3	1,3	2,2
	9,5	1	0,9	7,4
6	1,4	1,3	1,4	1,4
		1,4		
7	6,8	6,3	6,2	5,4
	2,1	1,6	2,2	2,2
8	2,2	6,8	6,7	6,3
	1,4	2	1,4	2,2
9	0,5	1,3	5,8	6,5
	2,5	1,3	2,2	2,2
10	1,4			
	6,9	1,6	5,6	6,3
Rata-Rata	2,2	1,3	2,2	2,2
	9,7	5,2	3,7	6,4
	1,4	1,3	1,3	2,2
Rata-Rata	4,852381	2,686364	4,031579	4,145

Lampiran 6 Data Utilitas RAM *Kspace* ukuran 1024 x 1024

Percobaan	Utilitas RAM 2048 x 2048			
	Serial	Paralel (Jumlah <i>Thread</i>)		
		1 <i>Thread</i>	2 <i>Thread</i>	4 <i>Thread</i>
1	17,8	17	23,7	23,8
	3,8	3,8	4,6	7,1
		7,1	4,2	
2	21	21,1	1,2	22,9
	3,8	3,8	21,4	7,1
	4,8	3,9	6,4	
3			4,2	
	12	12,8	16,1	17,1
	3,8	3,8	3,8	3,8
4	3,8	5	4,2	4,1
	11,6	17,4	17,5	20,1
	3,8	3,8	3,8	6,3
5		3,8	7,1	
	12,9	2,6	16	18,5
	3,8	7,7	3,8	4,6
6		4,2	4,2	
	17,3	5,7	20,4	22,3
	3,8	3,8	3,8	7,1
7		7,1	4,2	
	18,9	20,5	22,5	1,5
	3,8	3,8	3,8	17,7
8	7,1	7,1	4,2	7,1
	21,4	24,1	14,1	15,6
	3,8	3,8	3,8	3,8
9	3,9	4	4,2	4,1
	12,2	17	23,4	21,3
	3,8	3,8	4,3	7,1
10	3,8	7,1	4,2	
	15,2	8,5	22,9	19
	3,8	3,8	3,8	5
	6,5	3,8	4,2	
Rata-Rata	8,389655	7,803125	9,132258	11,6087

Lampiran 7 Data Utilitas RAM *Kspace* ukuran 2048 x 2048

Percobaan	Utilitas RAM 2048 x 2048			
	Serial	Paralel (Jumlah Thread)		
		1 Thread	2 Thread	4 Thread
1	8,4	1,2	13,7	11,5
	31,8	25,2	13,7	34
	13,7	13,7	26,9	13,7
		13,7	14,2	
2			15,4	
	18,8	1,2	3,1	6,5
	13,7	25,3	26,7	29,3
	13,7	13,7	13,7	13,7
3		13,7	13,7	13,7
	15,6	18,9	8,4	8,4
	13,7	13,7	33,9	34
	13,7	13,7	13,7	13,7
4			13,7	13,7
	23,6	20,9	7,4	7,9
	13,7	13,7	30,4	33
	13,7	13,7	13,7	13,7
5			13,7	13,7
	20,6	2	9	1
	13,7	25,6	34,2	25,4
	13,7	13,7	13,7	13,7
6		13,7	13,7	13,7
	3,2	2,1	3,1	8,9
	27,6	25,7	27,8	34,2
	13,7	13,7	13,7	13,7
7		13,7	13,7	13,7
	22,5	6,5	8,6	4,3
	13,7	31,5	31,7	29,3
	13,7	13,7	13,7	13,7
8		13,7	13,7	13,7
	2,6	6,7	2	9,7
	27,6	31,7	26,2	33,9
	13,7	13,7	13,7	13,7
5		13,7	13,7	13,7
	13,7	13,7	13,7	13,7
	16,7	4,4	17	7,7
	13,7	28,8	13,7	32,4
10		13,7	13,7	13,7
	13,7	13,7	13,7	13,7
	21,6	7	6	12,7
	13,7	30,3	28,9	13,7
		13,7	13,7	13,7
		13,7	13,7	13,7
Rata-Rata	15,49355	14,97368	15,455	16,36154

Lampiran 8 *Source Code* Serial Program Rekonstruksi Citra MRI

```

#include <iostream>
#include <fstream>
#include <string>
#include <stdio.h>
#include <vector>
#include <cmath>
#include <sys/time.h>
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>
#include <fftw3.h>

typedef struct float2 {
    float x;
    float y;
} cplx;

#define RANK 1
#define M_PI 3.14159265358979323846
#define fftshift(out, in, x, y, bitdim, numSlice) circshift(out, in, x, y, bitdim, (x/2), (y/2), numSlice)

float cuCrealf(cplx x) {
    return x.x;
}

float cuCimagf(cplx x) {
    return x.y;
}

double start_timer();
double lap_time(const char *msg);
void readHeaderFile(char *filen, int * dims);
int readCflFile(char *file, cplx* data);
void getSpecificSliceData(cplx* input, cplx* out, cplx* rawOut, int spec_slice, int spec_bit, const int *dim, int x_new, int y_new, float* times);
void dataScaling(float* data, int size);
int DoDisplayImage2CV(float* dataRaw, float* dataRecons, int xdim, int ydim, int spec_slice, int zipxdim, int numSlice);
void circshift(cplx *out, cplx *in, int xdim, int ydim, int bitdim, int xshift, int yshift, int numSlice);
void DoRSSCPU(cplx* input, float* out, int N, int x, int y, int bitdim, int numSlice);
int DoRECONSOOperation(cplx* DatafftOneSlice, float* out, int *dim, int numSlice, float* times);

float cuCabsf(cplx x){
    float a = cuCrealf(x);
    float b = cuCimagf(x);
    float v, w, t;
    a = fabsf(a);
    b = fabsf(b);
    if (a > b) {
        v = a;
        w = b;
    } else {
        v = b;
        w = a;
    }
    t = w / v;
    t = 1.0f + t * t;
    t = v * sqrtf(t);
    if ((v == 0.0f) || (v > 3.402823466e38f) || (w > 3.402823466e38f)) {
        t = v + w;
    }
    return t;
}

using namespace std;
using namespace cv;

int main(int argc, char* argv[])
{
    if(argc < 5)
    {
        cout << "<Usage> <path/filename> <Start Slice> <Number Slice Per Operatio> <New X dimension  
(Zero Filled Dimension)>" << endl;
        return 1;
    }
}

```

```

    }
    float CPU_start1, CPU_end1;
    char *filename = (char*)malloc(100);
    int *dim = (int*)malloc(sizeof(int)*4);
    int *Newdim = (int*)malloc(sizeof(int)*4);
    float *timesRecons = (float*)malloc(sizeof(float)*4);
    float *timesRaw = (float*)malloc(sizeof(float)*2);
    sprintf(filename,"%s",argv[1]);
    int slice = atoi(argv[2]);
    int numSlice = atoi(argv[3]);
    int new_x = atoi(argv[4]);
    if (new_x < 512)
    {
        cout << "ZIP Dimension Less than 512" << endl;
        new_x = 512;
    }
    else
    {
        int x_comp = 1;
        while(x_comp < new_x)
        {
            x_comp <<= 1;
        }
        if(x_comp > new_x)
        {
            cout << "Zero Filled Dimension Must be Power of TWO" << endl;
            return 1;
        }
    }
    readHeaderFile(filename, dim);
    int sizeCfl = 1;
    for(int i = 0; i < 4; i++)
    {
        sizeCfl *= dim[i];
        Newdim[i] = dim[i];
    }
    cplx *data = (cplx*)malloc(sizeof(cplx)*sizeCfl);
    int ret = readCflFile(filename, data);
    if(ret == 1)
    {
        cout << "Error on Reading CFL File" << endl;
        return 1;
    }
    int xdim = dim[0];
    int ydim = dim[1];
    int bitdim = dim[3];
    int xdim_new = new_x;
    int ydim_new = xdim_new;
    int sizeImageNew = xdim_new*ydim_new * numSlice;
    int sizeImage = xdim*ydim*numSlice;
    int sizeImageNewSlice = sizeImageNew* bitdim;
    int sizeImageNewSliceRaw = xdim*ydim* bitdim * numSlice;
    size_t nBytes_C = sizeof(cplx)*sizeImageNewSlice;
    size_t nBytes_CRaw = sizeof(cplx)*sizeImageNewSliceRaw;
    size_t nBytes_F = sizeof(float)*sizeImageNew;
    size_t nBytes_FRSS = sizeof(float)*sizeImage;
    cplx* dataManySlice = (cplx*)malloc(nBytes_C);
    cplx* dataManySliceRaw = (cplx*)malloc(nBytes_CRaw);
    memset(dataManySlice, 0, nBytes_C);
    getSpecificSliceData(data, dataManySlice, dataManySliceRaw, slice, numSlice, dim, xdim_new, ydim_new,
timesRaw);
    free(data);
    Newdim[0] = xdim_new;
    Newdim[1] = ydim_new;
    xdim = dim[0];
    ydim = dim[1];
    bitdim = dim[3];
    int sizeManyImage = xdim*ydim*numSlice;
    float CPUTimer_getspec = timesRaw[0] + timesRaw[1];
    float* dataFFT_F = (float*)malloc(nBytes_F);
    int retCUFFT = DoRECONSOOperation(dataManySlice, dataFFT_F, Newdim, numSlice, timesRecons);
    if(retCUFFT == 1)
    {

```

```

        cout << "CPU RECONSTRUCTION Operation is FAILED" << endl;
        return 1;
    }
    float CPUTimer_fft = timesRecons[0]+timesRecons[1]+timesRecons[2]+timesRecons[3];
    CPU_start1 = (float)start_timer();
    float *rss = (float*)malloc(nBytes_FRSS);
    DoRSSCPU(dataManySliceRaw, rss, sizeManyImage, xdim, ydim, bitdim, numSlice);
    float CPUTimer_RSS = (float)lap_time("");
    float CPUTimer = CPUTimer_getspec + CPUTimer_fft + CPUTimer_RSS;
    cout <<
    "=====
    =====>" << endl;
    cout << "<path/filename> <Start Slice> <Number Slice Per Operatio> <New X dimension (Zero Filled
    Dimension)>" << endl;
    cout << "<kspace> " << std::to_string(slice) << " " << std::to_string(numSlice) << " " << std::to_string(new_x)
    << endl;
    cout << " " << endl;
    cout << "Timing - Get Specific Slice Data on CPU Processor : " << std::to_string(timesRaw[0]*1000) << " ms"
    << endl;
    cout << "Timing - Zero Filling Interpolation on CPU Processor : " << std::to_string(timesRaw[1]*1000) << "
    ms" << endl;
    cout << "Timing - RECONSTRUCTION on CPU Processor : " << std::to_string(CPUTimer_fft*1000) << " ms"
    << endl;
    cout << "    Timing - FFT on CPU Processor : " << std::to_string(timesRecons[0]*1000) << " ms" << endl;
    cout << "    Timing - FFTSHIFT on CPU Processor : " << std::to_string(timesRecons[1]*1000) << " ms" <<
    endl;
    cout << "    Timing - RSS on CPU Processor : " << std::to_string(timesRecons[2]*1000) << " ms" << endl;
    cout << "    Timing - Data Scaling on CPU Processor : " << std::to_string(timesRecons[3]*1000) << " ms"
    << endl;
    cout << "Timing - RAW Operation on CPU Processor : " << std::to_string(CPUTimer_RSS*1000) << " ms" <<
    endl;
    cout << "CPU Timing using CPU Timer : " << std::to_string(CPUTimer*1000) << " ms" << endl;
    cout <<
    "=====
    =====>" << endl;
    int retCV = DoDisplayImage2CV(rss, dataFFT_F, xdim, ydim, slice, xdim_new, numSlice);
    if(retCV == 1)
    {
        cout << "Display Image is Failed" << endl;
        return 1;
    }
    free(filename);
    free(dim);
    free(Newdim);
    free(timesRecons);
    free(timesRaw);
    free(rss);
    free(dataManySlice);
    free(dataManySliceRaw);
    free(dataFFT_F);
    return 0;
}
void readHeaderFile(char *filen, int * dims)
{
    char path[20];
    sprintf(path, "%s.hdr", filen);
    FILE *myFile;
    myFile = fopen(path, "r");
    string line;
    streampos size;
    fseek(myFile, 13, SEEK_SET);
    for(int i = 0; i < 4; i++)
    {
        fscanf(myFile, "%d", &dims[i]);
    }
}
int readCflFile(char *filen, cplx* data)
{
    streampos size;
    char path[20];
    sprintf(path, "%s.cfl", filen);
    ifstream file(path, ios::in | ios::binary | ios::ate);
    if(file.is_open())

```

```

    {
        size = file.tellg();
        cout << "Contains Size : "<< std::to_string(size) << endl;
    }
    else
    {
        cout << "Unable to open file";
        return 1;
    }
    if(file.is_open())
    {
        file.seekg(0, ios::beg);
        file.read((char*)data, size);
        file.close();
    }
    return 0;
}
void dataScaling(float* data, int size)
{
    float max = 0;
    for(int i = 0; i < size; i++)
    {
        if(data[i] > max)
        {
            max = data[i];
        }
    }
    for(int j = 0; j < size; j++)
    {
        data[j] = data[j]/max;
    }
}
void getSpecificSliceData(cplx* input, cplx* out, cplx* rawOut, int spec_slice, int numSlice, const int* dim, int xdim_new,
int ydim_new, float* times )
{
    clock_t start,end;
    int xdim = dim[0];
    int ydim = dim[1];
    int slicedim = dim[2];
    int bitdim = dim[3];
    int sizeOneImage = xdim*ydim;
    int residue = (xdim_new-xdim)/2;
    int sidx = (xdim_new*residue)+residue;
    int offset = 2*residue;
    int offsetPerBitmask = numSlice*xdim_new*ydim_new;
    int offsetPerSlice = xdim_new*ydim_new;
    int index = 0;
    start_timer();
    for(int m = 0; m < bitdim; m++)
    {
        for(int l = spec_slice; l < (spec_slice+numSlice); l++)
        {
            for(int k = 0; k < ydim; k++)
            {
                for(int j = 0; j < xdim; j++)
                {
                    rawOut[index] = input[j + (k*xdim) + (l*sizeOneImage) +
(m*slicedim*sizeOneImage)];
                    index++;
                }
            }
        }
    }
    times[0] = (float)lap_time("get specific slice time[0] : ");
    start_timer();
    for(int m = 0; m < bitdim; m++)
    {
        for(int l = 0; l < numSlice; l++)
        {
            for(int i = 0; i < ydim ; i++)
            {
                for(int j = 0; j < xdim; j++)
                {

```

```

        out[sidx + j + (i*xdim) + (i*offset) + (i*offsetPerSlice)
+(m*offsetPerBitmask)] = rawOut[j + (i*xdim) + (i*sizeOneImage) + (m*numSlice*sizeOneImage)];
    }
}
}
times[1] = (float)lap_time("get specific slice time[1] ");
}
void DoRSSCPU(cplx* input, float* out, int N, int x, int y, int bitdim, int numSlice)
{
    float *temp = (float*)malloc(sizeof(float)*N);
    for(int i = 0; i < numSlice; i++)
    {
        for(int m = 0; m < bitdim ; m++)
        {
            for(int j = 0; j < y; j++)
            {
                for(int k = 0; k < x; k++)
                {
                    out[k + (j*x) + (i*y*x)] = cuCabsf(input[k + (j*x) + (i*y*x) +
(m*x*y*numSlice)]);
                    out[k + (j*x) + (i*y*x)] = (float)pow((out[k + (j*x) + (i*y*x)]),2);
                    temp[k + (j*x) + (i*y*x)] += out[k + (j*x) + (i*y*x)];
                }
            }
        }
    }
    for(int i = 0; i < numSlice ; i++)
    {
        for(int j = 0; j < y; j++)
        {
            for(int k = 0; k < x; k++)
            {
                out[k + (j*x) + (i*y*x)] = 0;
                out[k + (j*x) + (i*y*x)] = sqrt(temp[k + (j*x) + (i*y*x)]);
            }
        }
    }
    free(temp);
}
int DoDisplayImage2CV(float* dataRaw, float* dataRecons, int xdim, int ydim, int spec_slice, int zipdimx, int numSlice)
{
    char winNameRaw[100];
    char winNameRecons[100];
    cv::Mat imgRaw;
    cv::Mat imgRecons;
    int oneDimRaw = xdim*ydim;
    int oneDim = zipdimx*zipdimx;
    size_t nBytes_One = sizeof(float)*oneDim;
    size_t nBytes_OneRaw = sizeof(float)*oneDimRaw;
    float* oneImage = (float*)malloc(nBytes_One);
    float* oneImageRaw = (float*)malloc(nBytes_OneRaw);
    int offset = 0;
    int offsetRaw = 0;
    for(int a = 0; a < numSlice; a++)
    {
        offset = a*oneDim;
        offsetRaw = a*oneDimRaw;
        memcpy(oneImage, dataRecons + offset, nBytes_One);
        memcpy(oneImageRaw, dataRaw + offsetRaw, nBytes_OneRaw);
        imgRaw = cv::Mat(xdim, ydim, CV_32F, oneImageRaw);
        imgRecons = cv::Mat(zipdimx, zipdimx, CV_32F, oneImage);
        if(imgRaw.rows == 0 || imgRaw.cols == 0)
            return 1;
        if(imgRecons.rows == 0 || imgRecons.cols == 0)
            return 1;
        sprintf(winNameRecons, "Reconstructed Image on CV - Slice %d", spec_slice+a );
        sprintf(winNameRaw, "Raw Image on CV - Slice %d", spec_slice+a );
        cv::namedWindow(winNameRaw, WINDOW_KEEPRATIO | WINDOW_NORMAL);
        cv::namedWindow(winNameRecons, WINDOW_KEEPRATIO | WINDOW_NORMAL);
        cv::imshow(winNameRaw, imgRaw);
        cv::waitKey(500);
        cv::imshow(winNameRecons, imgRecons);
    }
}

```

```

        cv::waitKey(2000);
    }
    cv::waitKey();
    free(oneImage);
    free(oneImageRaw);
    return 0;
}
int DoRECONSOOperation( cplx* DatafftManySlice, float* out, int *dim, int numSlice, float* times)
{
    int xdim = dim[0];
    int ydim = dim[1];
    int bitdim = dim[3];
    int sizeOneSlice = xdim*ydim*bitdim;
    int sizeManySlice = sizeOneSlice*numSlice;
    int sizeManyImage = xdim*ydim*numSlice;
    int sizeOneImage = xdim*ydim;
    size_t nBytes_C = sizeof(cplx)*sizeManySlice;
    cplx* temp_in = (cplx*)malloc(nBytes_C);
    clock_t start, end;
    start_timer();
    fftwf_plan pfftw;
    int n[] = {sizeOneImage};
    int howmany = bitdim*numSlice;
    int idist = sizeOneImage;
    int odist = sizeOneImage;
    int istride = 1;
    int ostride = 1;
    int *onembed = NULL;
    int *inembed = NULL;
    pfftw = fftwf_plan_many_dft(RANK, n, howmany,

    reinterpret_cast<fftwf_complex*>(DatafftManySlice),

    inembed, istride, idist,

    reinterpret_cast<fftwf_complex*>(DatafftManySlice),

    onembed, ostride, odist,
    FFTW_BACKWARD,
    FFTW_ESTIMATE);

    fftwf_execute(pfftw);
    fftwf_destroy_plan(pfftw);
    fftwf_cleanup();
    times[0] = (float)lap_time("fftw time[0]");
    start_timer();
    memcpy(temp_in, DatafftManySlice, nBytes_C);
    fftshift(DatafftManySlice, temp_in, xdim, ydim, bitdim, numSlice);
    free(temp_in);
    times[1] = (float)lap_time("fft time[1] ");
    start_timer();
    DoRSSCPU(DatafftManySlice, out, sizeManyImage, xdim, ydim, bitdim, numSlice);
    times[2] = (float)lap_time("DoRSSCPU time :");
    start_timer();
    dataScaling(out, sizeManyImage);
    times[3] = (float)lap_time("Data scaling time ");
    return 0;
}
void circshift(cplx *out, cplx *in, int xdim, int ydim, int bitdim, int xshift, int yshift, int numSlice)
{
    int N = xdim*ydim;
    int Bitm = N*numSlice;
    for(int m = 0; m < bitdim; m++)
    {
        for(int l = 0; l < numSlice ; l++)
        {
            for (int i =0; i < xdim; i++) {
                int ii = (i + xshift) % xdim;
                for (int j = 0; j < ydim; j++) {
                    int jj = (j + yshift) % ydim;
                    out[ii * ydim + jj + (l*N) + (m*Bitm)] = in[i * ydim + j + (l*N) + (m*Bitm)];
                }
            }
        }
    }
}

```

```
using namespace std;
double second() {
    double t;
    struct timeval tv;
    gettimeofday(&tv,NULL);
    t = (double)(tv.tv_sec) + ((double)(tv.tv_usec))/1.0e6;
    return t;
}

static double start_time;

double start_timer() {
    start_time = second();
    return start_time;
}

double lap_time(const char *msg) {
    double t, lap_t;
    t = second();
    lap_t = t - start_time;
    cout << msg << " : " << (double)lap_t << " Sec";
    cout << endl;
    return lap_t;
}
```


Lampiran 9 Source Code Program Paralel Rekonstruksi Citra MRI

```

#include <iostream>
#include <fstream>
#include <string>
#include <stdio.h>
#include <vector>
#include <cmath>
#include <sys/time.h>
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>
#include <fftw3.h>
#include <omp.h>
typedef struct float2 {
    float x;
    float y;
}cplx;

#define RANK 1
#define M_PI 3.14159265358979323846
#define fftshift(out, in, x, y, bitdim, numSlice) circshift(out, in, x, y, bitdim, (x/2), (y/2), numSlice)

float cuCrealF(cplx x) {
    return x.x;
}

float cuCimagf(cplx x) {
    return x.y;
}

double lap_time(const char *);
double start_timer();
void readHeaderFile(char *file, int * dims);
int readCflFile(char *file, cplx* data);
void getSpecificSliceData(cplx* input, cplx* out, cplx* rawOut, int spec_slice, int spec_bit, const int *dim, int x_new, int y_new, float* times);
void dataScaling(float* data, int size);
int DoDisplayImage2CV(float* dataRaw, float* dataRecons, int xdim, int ydim, int spec_slice, int zipxdim, int numSlice);
void circshift(cplx *out, cplx *in, int xdim, int ydim, int bitdim, int xshift, int yshift, int numSlice);
void DoRSSCPU(cplx* input, float* out, int N, int x, int y, int bitdim, int numSlice);
int DoRECONOperation( cplx* DatafftOneSlice, float* out, int *dim, int numSlice, float* times);
int nthreads = 1;

float cuCabsf(cplx x){
    float a = cuCrealF(x);
    float b = cuCimagf(x);
    float v, w, t;
    a = fabsf(a);
    b = fabsf(b);
    if (a > b) {
        v = a;
        w = b;
    } else {
        v = b;
        w = a;
    }
    t = w / v;
    t = 1.0f + t * t;
    t = v * sqrtf(t);
    if ((v == 0.0f) || (v > 3.402823466e38f) || (w > 3.402823466e38f)) {
        t = v + w;
    }
    return t;
}

using namespace std;
using namespace cv;

int main(int argc, char* argv[])
{
    if(argc < 5)
    {

```

```

        cout << "<Usage> <path/filename> <Start Slice> <Number Slice Per Operatio> <New X dimension  
(Zero Filled Dimension)>" << endl;
        return 1;
    }
    float CPU_start1, CPU_end1;
    char *filename = (char*)malloc(100);
    int *dim = (int*)malloc(sizeof(int)*4);
    int *Newdim = (int*)malloc(sizeof(int)*4);
    float *timesRecons = (float*)malloc(sizeof(float)*4);
    float *timesRaw = (float*)malloc(sizeof(float)*2);
    sprintf(filename,"%s",argv[1]);
    int slice = atoi(argv[2]);
    int numSlice = atoi(argv[3]);
    int new_x = atoi(argv[4]);
    nthreads = atoi(argv[5]);
    if (new_x < 512)
    {
        cout << "ZIP Dimension Less than 512" << endl;
        new_x = 512;
    }
    else
    {
        int x_comp = 1;
        while(x_comp < new_x)
        {
            x_comp <= 1;
        }
        if(x_comp > new_x)
        {
            cout << "Zero Filled Dimension Must be Power of TWO" << endl;
            return 1;
        }
    }
    readHeaderFile(filename, dim);
    int sizeCfl = 1;
    for(int i = 0; i < 4; i++)
    {
        sizeCfl *= dim[i];
        Newdim[i] = dim[i];
    }
    cplx *data = (cplx*)malloc(sizeof(cplx)*sizeCfl);
    int ret = readCflFile(filename, data);
    if(ret == 1)
    {
        cout << "Error on Reading CFL File" << endl;
        return 1;
    }
    // CPU_start1 = (float)start_timer();
    int xdim = dim[0];
    int ydim = dim[1];
    int bitdim = dim[3];
    int xdim_new = new_x;
    int ydim_new = xdim_new;
    int sizeImageNew = xdim_new*ydim_new * numSlice;
    int sizeImage = xdim*ydim*numSlice;
    int sizeImageNewSlice = sizeImageNew* bitdim;
    int sizeImageNewSliceRaw = xdim*ydim* bitdim * numSlice;
    size_t nBytes_C = sizeof(cplx)*sizeImageNewSlice;
    size_t nBytes_CRaw = sizeof(cplx)*sizeImageNewSliceRaw;
    size_t nBytes_F = sizeof(float)*sizeImageNew;
    size_t nBytes_FRSS = sizeof(float)*sizeImage;
    cplx* dataManySlice = (cplx*)malloc(nBytes_C);
    cplx* dataManySliceRaw = (cplx*)malloc(nBytes_CRaw);
    memset(dataManySlice, 0, nBytes_C);
    getSpecificSliceData(data, dataManySlice, dataManySliceRaw, slice, numSlice, dim, xdim_new, ydim_new,
timesRaw);
    free(data);
    Newdim[0] = xdim_new;
    Newdim[1] = ydim_new;
    xdim = dim[0];
    ydim = dim[1];
    bitdim = dim[3];
    int sizeManyImage = xdim*ydim*numSlice;

```

```

//      CPU_end1 = clock();
//      float CPUTimer_getspec = timesRaw[0] + timesRaw[1];
//      CPU_start1 = (float)start_timer();
//      float* dataFFT_F = (float*)malloc(nBytes_F);
//      int retCUFFT = DoRECONSOOperation(dataManySlice, dataFFT_F, Newdim, numSlice, timesRecons);
//      if(retCUFFT == 1)
//      {
//          cout << "CPU RECONSTRUCTION Operation is FAILED" << endl;
//          return 1;
//      }
//      CPU_end1 = (float)lap_time("DoRECONSOPs time : ");
//      float CPUTimer_fft = timesRecons[0]+timesRecons[1]+timesRecons[2]+timesRecons[3];
//      CPU_start1 = (float)start_timer();
//      float *rss = (float*)malloc(nBytes_FRSS);
//      DoRSSCPU(dataManySliceRaw, rss, sizeManyImage, xdim, ydim, bitdim, numSlice);
//      float CPUTimer_RSS = (float)lap_time("DoRSSCPU raw data : ");
//      float CPUTimer = CPUTimer_getspec + CPUTimer_fft + CPUTimer_RSS;
//      cout <<
//      "=====
//      =====>" << endl;
//      cout << "<path/filename> <Start Slice> <Number Slice Per Operatio> <New X dimension (Zero Filled
//      Dimension)>" << endl;
//      cout << "<kspace> " << std::to_string(slice) << " " << std::to_string(numSlice) << " " << std::to_string(new_x)
//      << endl;
//      cout << " " << endl;
//      cout << "Timing - Get Specific Slice Data on CPU Processor : " << std::to_string(timesRaw[0]*1000) << " ms"
//      << endl;
//      cout << "Timing - Zero Filling Interpolation on CPU Processor : " << std::to_string(timesRaw[1]*1000) << "
//      ms" << endl;
//      cout << "Timing - RECONSTRUCTION on CPU Processor : " << std::to_string(CPUTimer_fft*1000) << " ms"
//      << endl;
//      cout << "      Timing - FFT on CPU Processor : " << std::to_string(timesRecons[0]*1000) << " ms" << endl;
//      cout << "      Timing - FFTSHIFT on CPU Processor : " << std::to_string(timesRecons[1]*1000) << " ms" <<
//      endl;
//      cout << "      Timing - RSS on CPU Processor : " << std::to_string(timesRecons[2]*1000) << " ms" << endl;
//      cout << "      Timing - Data Scaling on CPU Processor : " << std::to_string(timesRecons[3]*1000) << " ms"
//      << endl;
//      cout << "Timing - RAW Operation on CPU Processor : " << std::to_string(CPUTimer_RSS*1000) << " ms" <<
//      endl;
//      cout << "CPU Timing using CPU Timer : " << std::to_string(CPUTimer*1000) << " ms" << endl;
//      cout <<
//      "=====
//      =====>" << endl;
//      int retCV = DoDisplayImage2CV(rss, dataFFT_F, xdim, ydim, slice, xdim_new, numSlice);
//      if(retCV == 1)
//      {
//          cout << "Display Image is Failed" << endl;
//          return 1;
//      }
//      free(filename);
//      free(dim);
//      free(Newdim);
//      free(timesRecons);
//      free(timesRaw);
//      free(rss);
//      free(dataManySlice);
//      free(dataManySliceRaw);
//      free(dataFFT_F);
//      return 0;
}
void readHeaderFile(char *filen, int * dims)
{
    char path[20];
    sprintf(path,"%s.hdr",filen);
    FILE *myFile;
    myFile = fopen(path,"r");
    string line;
    streampos size;

    fseek(myFile, 13, SEEK_SET);
    for(int i = 0; i < 4; i++)
    {
        fscanf(myFile,"%d",&dims[i]);
    }
}

```

```

    }
}
int readCflFile(char *filen, cplx* data)
{
    streampos size;
    char path[20];
    sprintf(path,"%s.cfl",filen);
    ifstream file(path, ios::in | ios::binary | ios::ate);
    if(file.is_open())
    {
        size = file.tellg();
        cout << "Contains Size : "<< std::to_string(size) << endl;
    }
    else
    {
        cout << "Unable to open file";
        return 1;
    }

    if(file.is_open())
    {
        file.seekg(0, ios::beg);
        file.read((char*)data, size);
        file.close();
    }
    return 0;
}
void dataScaling(float* data, int size)
{
    float max = 0;
    for(int i = 0; i < size; i++)
    {
        if(data[i] > max)
        {
            max = data[i];
        }
    }

    for(int j = 0; j < size; j++)
    {
        data[j] = data[j]/max;
    }
}
void getSpecificSliceData(cplx* input, cplx* out, cplx* rawOut, int spec_slice, int numSlice, const int* dim, int xdim_new,
int ydim_new, float* times )
{
    clock_t start,end;
    int xdim = dim[0];
    int ydim = dim[1];
    int slicedim = dim[2];
    int bitdim = dim[3];
    int sizeOneImage = xdim*ydim;
    int residue = (xdim_new-xdim)/2;
    int sidx = (xdim_new*residue)+residue;
    int offset = 2*residue;
    int offsetPerBitmask = numSlice*xdim_new*ydim_new;
    int offsetPerSlice = xdim_new*ydim_new;
    int index = 0;
    start_timer();
    for(int m = 0; m < bitdim; m++)
    {
        for(int l = spec_slice; l < (spec_slice+numSlice); l++)
        {
            for(int k = 0; k < ydim; k++)
            {
                for(int j = 0; j < xdim; j++)
                {
                    rawOut[index] = input[j + (k*xdim) + (l*sizeOneImage) +
(m*slicedim*sizeOneImage)];
                    index++;
                }
            }
        }
    }
}

```

```

    }
    times[0] = (float)lap_time("");
    start_timer();
    for(int m = 0; m < bitdim; m++)
    {
        for(int l = 0; l < numSlice; l++)
        {
            for(int i = 0; i < ydim ; i++)
            {
                for(int j = 0; j < xdim; j++)
                {
                    out[sidx + j + (i*xdim) + (i*offset) + (l*offsetPerSlice)
+ (m*offsetPerBitmask)] = rawOut[j + (i*xdim) + (l*sizeOneImage) + (m*numSlice*sizeOneImage)];
                }
            }
        }
    }
    times[1] = (float)lap_time("");
}

void DoRSSCPU(cplx* input, float* out, int N, int x, int y, int bitdim, int numSlice)
{
    float *temp = (float*)malloc(sizeof(float)*N);
    double start,end;
    std::cout << "bitdim " << bitdim << endl;
    start = omp_get_wtime();
    #pragma omp parallel for schedule(static,1) num_threads(nthreads)
    for(int m = 0; m < bitdim; m++)
    {
        for(int i = 0; i < numSlice ; i++)
        {
            for(int j = 0; j < y; j++)
            {
                for(int k = 0; k < x; k++)
                {
                    out[k + (j*x) + (i*y*x)] = cuCabsf(input[k + (j*x) + (i*y*x) +
(m*x*y*numSlice)]);

                    out[k + (j*x) + (i*y*x)] = (float)pow((out[k + (j*x) + (i*y*x)]),2);
                    temp[k + (j*x) + (i*y*x)] += out[k + (j*x) + (i*y*x)];
                }
            }
        }
    }
    #pragma omp parallel for schedule(static,1) num_threads(nthreads)
    for(int i = 0; i < numSlice ; i++)
    {
        for(int j = 0; j < y; j++)
        {
            for(int k = 0; k < x; k++)
            {
                out[k + (j*x) + (i*y*x)] = 0;
                out[k + (j*x) + (i*y*x)] = sqrt(temp[k + (j*x) + (i*y*x)]);
            }
        }
    }
    printf("DoRSS wall time : %lf\n",omp_get_wtime()-start);
    free(temp);
}

int DoDisplayImage2CV(float* dataRaw, float* dataRecons, int xdim, int ydim, int spec_slice, int zipdimx, int numSlice)
{
    char winNameRaw[100];
    char winNameRecons[100];
    cv::Mat imgRaw;
    cv::Mat imgRecons;
    int oneDimRaw = xdim*ydim;
    int oneDim = zipdimx*zipdimx;
    size_t nBytes_One = sizeof(float)*oneDim;
    size_t nBytes_OneRaw = sizeof(float)*oneDimRaw;
    float* oneImage = (float*)malloc(nBytes_One);
    float* oneImageRaw = (float*)malloc(nBytes_OneRaw);
    int offset = 0;
    int offsetRaw = 0;
    for(int a = 0; a < numSlice; a++)
    {

```

```

        offset = a*oneDim;
        offsetRaw = a*oneDimRaw;
        memcpy(oneImage, dataRecons + offset, nBytes_One);
        memcpy(oneImageRaw, dataRaw + offsetRaw, nBytes_OneRaw);
        imgRaw = cv::Mat(xdim, ydim, CV_32F, oneImageRaw);
        imgRecons = cv::Mat(zipdimx, zipdimx, CV_32F, oneImage);
        if(imgRaw.rows == 0 || imgRaw.cols == 0)
            return 1;
        if(imgRecons.rows == 0 || imgRecons.cols == 0)
            return 1;
        sprintf(winNameRecons, "Reconstructed Image on CV - Slice %d", spec_slice+a );
        sprintf(winNameRaw, "Raw Image on CV - Slice %d", spec_slice+a );
        cv::namedWindow(winNameRaw, WINDOW_KEEPRATIO | WINDOW_NORMAL);
        cv::namedWindow(winNameRecons, WINDOW_KEEPRATIO | WINDOW_NORMAL);
        cv::imshow(winNameRaw, imgRaw);
        cv::waitKey(500);
        cv::imshow(winNameRecons, imgRecons);
        cv::waitKey(2000);
    }
    cv::waitKey();
    free(oneImage);
    free(oneImageRaw);
    return 0;
}

int DoRECONSOOperation( cplx* DatafftManySlice, float* out, int *dim, int numSlice, float* times)
{
    int xdim = dim[0];
    int ydim = dim[1];
    int bitdim = dim[3];
    int sizeOneSlice = xdim*ydim*bitdim;
    int sizeManySlice = sizeOneSlice*numSlice;
    int sizeManyImage = xdim*ydim*numSlice;
    int sizeOneImage = xdim*ydim;
    size_t nBytes_C = sizeof(cplx)*sizeManySlice;
    cplx* temp_in = (cplx*)malloc(nBytes_C);
    clock_t start, end;
    start_timer();
    fftwf_plan pfftw;
    int n[] = {sizeOneImage};
    int howmany = bitdim*numSlice;
    int idist = sizeOneImage;
    int odist = sizeOneImage;
    int istride = 1;
    int ostride = 1;
    int *onembed = NULL;
    int *inembed = NULL;
    fftwf_init_threads();
    fftwf_plan_with_nthreads(nthreads);
    pfftw = fftwf_plan_many_dft(RANK, n, howmany,

        reinterpret_cast<fftwf_complex*>(DatafftManySlice),

        inembed, istride, idist,

        reinterpret_cast<fftwf_complex*>(DatafftManySlice),

        onembed, ostride, odist,
        FFTW_BACKWARD,
        FFTW_ESTIMATE);

    fftwf_execute(pfftw);
    fftwf_destroy_plan(pfftw);
    fftwf_cleanup_threads();
    times[0] = (float)lap_time("waktu eksekusi fft : ");
    start_timer();
    memcpy(temp_in, DatafftManySlice, nBytes_C);
    fftshift(DatafftManySlice, temp_in, xdim, ydim, bitdim, numSlice);
    free(temp_in);
    times[1] = (float)lap_time("waktu eksekusi fftshift ");
    start_timer();
    DoRSSCPU(DatafftManySlice, out, sizeManyImage, xdim, ydim, bitdim, numSlice);
    times[2] = (float)lap_time("waktu eksekusi DoRSSCPU :");
    start_timer();
    dataScaling(out, sizeManyImage);
    end = clock();
    times[3] = (float)lap_time("data scaling");
}

```

```

        return 0;
    }
}
void circshift(cplx *out, cplx *in, int xdim, int ydim, int bitdim, int xshift, int yshift, int numSlice)
{
    int N = xdim*ydim;
    int Bitm = N*numSlice;
    for(int m = 0; m < bitdim; m++)
    {
        for(int l = 0; l < numSlice ; l++)
        {
            for (int i =0; i < xdim; i++) {
                int ii = (i + xshift) % xdim;
                for (int j = 0; j < ydim; j++) {
                    int jj = (j + yshift) % ydim;
                    out[ii * ydim + jj + (l*N) + (m*Bitm)] = in[i * ydim + j + (l*N) + (m*Bitm)];
                }
            }
        }
    }
}

using namespace std;
double second() {
    double t;
    struct timeval tv;
    gettimeofday(&tv,NULL);
    t = (double)(tv.tv_sec) + ((double)(tv.tv_usec))/1.0e6;
    return t;
}
static double start_time;

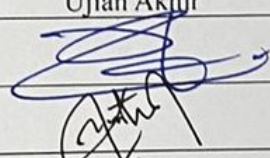
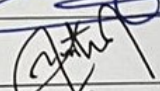
double start_timer() {
    start_time = second();
    return start_time;
}
double lap_time(const char *msg) {
    double t, lap_t;
    t = second();
    lap_t = t - start_time;
    cout << msg << " : " << (double)lap_t << " Sec";
    cout << endl;
    return lap_t;
}

```

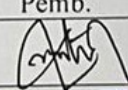
KARTU BIMBINGAN SKRIPSI

Prodi S1 Teknik Informatika Universitas Hasanuddin

Stb.	Nama Mahasiswa
D121181317	PUTRI ALISSYAH SABRINA

Pembimbing	Nama Pembimbing	Paraf & Tgl. Persetujuan Ujian Akhir
I	Dr. Adnan, S.T., M.T.	
II	Prof. Dr. Eng. Intan Sari Areni, S.T., M.T.	
No. SK Pemb		

Judul Skripsi	Evaluasi Kinerja Program Multithreading pada Penerapan rekonstruksi Citra
---------------	---

No	Tanggal Bimbingan	Uraian Kegiatan Bimbingan	Paraf Pemb.
1	30 November 2023	Konsultasi terkait penulisan hasil penelitian	
2	10 Juni 2024	Konsultasi hasil revisi seminar hasil	
3	13 Juni 2024	Konsultasi hasil revisi seminar hasil	
4	19 Juni 2024	Latihan presentasi hasil revisi seminar hasil	
5	19 Juni 2024	Latihan presentasi hasil revisi seminar hasil	
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

LEMBAR PERBAIKAN SKRIPSI

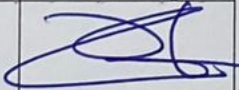
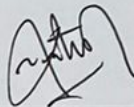
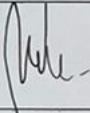
“EVALUASI KINERJA PROGRAM MULTITHREADING PADA PENERAPAN REKONSTRUKSI CITRA MRI MENGGUNAKAN SINGLE BOARD COMPUTER”

OLEH:

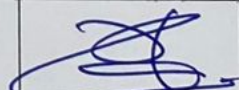
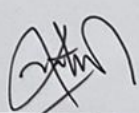
PUTRI ALISSYAH SABRINA
D121181317

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 24 Juni 2024.
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan
pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Adnan, S.T., M.T., P.hD.	
Sekretaris	Prof. Dr. Eng. Intan Sari Areni, S.T., M.T.	
Anggota	Dr-Eng.Ir. Muhammad Niswar, ST, M.InfoTech	
	Dr. Eng. Zulkifli Tahir, S.T., M.Sc.	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Adnan, S.T., M.T., P.hD.	
II	Prof. Dr. Eng. Intan Sari Areni, S.T., M.T.	



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Pores Maimo Km.6 Binamaramma 92172, Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://www.unhas.ac.id>, Email: teknika@unhas.ac.id

SURAT PENUGASAN

No. 17113/UN4.7.1/TD.06/2022

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : 1. Adnan, ST., M.T., Ph.D Pemb. I
2. Dr.Eng. Intan Sari Areni, ST., M.T Pemb. II

Isi : 1. Berdasarkan Surat Ketua Departemen Teknik Informatika Fakultas Teknik Nomor. 435-UN4.7.7/TD.06/2022 tanggal 16 Agustus 2022 tentang usul DOSEN PEMBIMBING MAHASISWA, maka dengan ini kami menugaskan Saudara untuk membimbing penulisan Skripsi/Tugas Akhir mahasiswa Teknik Informatika Fakultas Teknik Universitas Hasanuddin di bawah ini :

Nama : No. Stambuk :
Putri Alissyah Sabrina D121 18 1317

Judul Skripsi Tugas Akhir :

“ Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra MRI Menggunakan Single Board Computer “

2. Surat penugasan pembimbing ini mulai berlaku sejak tanggal ditetapkannya dan berakhir sampai selesainya penulisan Skripsi/Tugas Akhir mahasiswa tersebut.
3. Agar penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

Ditetapkan di Gowa
Pada tanggal 16 Agustus 2022
a.n. Dekan,
Wakil Dekan Bidang Akademik, Riset
dan Inovasi Fakultas Teknik Unhas

Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :

1. Dekan FT-UH,
2. Ketua Departemen Teknik Informatika FT-UH,
3. Mahasiswa yang bersangkutan





KEMENTERIAN PENDIDIKAN KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

Jalan Perintis Kemerdekaan Km. 10, Makassar 90245
Telepon (0411) 586200, (6 Saluran), 584200, Fax (0411) 585188

Laman: www.unhas.ac.id

SURAT IZIN UJIAN SKRIPSI
Nomor 36262/UN4.1.1.1/PK.03.02/2023

Berdasarkan Peraturan Rektor Universitas Hasanuddin tentang Penyelenggaraan Program Sarjana Nomor 2781/UN4.1/KEP/2018 tanggal 16 Juli 2018, dengan ini menerangkan bahwa:

Nama : PUTRI ALISSYAH SABRINA
NIM : D121181317
Tempat/Tanggal Lahir : MAKASSAR, 6 JULI 2000
Fakultas : TEKNIK
Program Studi : TEKNIK INFORMATIKA

Telah memenuhi syarat untuk Ujian Skripsi Strata I (S1). Demikian Surat Persetujuan ini dibuat untuk digunakan dalam proses pelaksanaan ujian skripsi, dengan ketentuan dapat mengikuti wisuda jika persyaratan kelulusan/wisuda telah dipenuhi. Terima Kasih.

Makassar, 18 Oktober 2023
a.n. Direktur Pendidikan
Kepala Subdirektorat Administrasi
Pendidikan,



Susy Asteria Irafany, S.T., M.Si.
NIP 197403132009102001

Keterangan online wisuda:

User : D121181317
Password : 2168842
Alamat : <http://wisuda.unhas.ac.id>
Web



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
Poros Malino Km.6 Bontomarannu (92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 21184/UN4.7.1/TD.06/2023

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : Mereka yang tercantum namanya dibawah ini

Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 18 (SK.Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Adnan, ST., M.T., Ph.D
Pembimbing II / Sekretaris : 2. Prof. Dr.Eng. Intan Sari Areni, ST., M.T
Anggota : 3. Dr.Eng. Muhammad Niswar, ST., M.IT
4. Dr.Eng. Zulkifli Tahir, S.T., M.Sc

Untuk menguji bagi mahasiswa tersebut dibawah ini :

Nama/NIM : Putri Alissyah Sabrina D121181317
Program Studi : Teknik Informatika
Judul thesis/Skripsi : Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra MRI Menggunakan Single Board Computer

2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa

Pada tanggal 22 September 2023

a.n. Dekan,

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Nomor : 1070/UN4.7.7/TD.06/2023
Lamp : -
Hal : Penerbitan Surat Penugasan Panitia
Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,

Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk	: Putri Alissyah Sabrina	D121181317
Judul TA	: Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra MRI Menggunakan Single Board Computer	

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua	: 1. Adnan, ST., M.T., Ph.D
Pembimbing II / Sekretaris	: 2. Prof. Dr.Eng. Intan Sari Areni, ST., M.T
Anggota	: 3. Dr.Eng. Muhammad Niswar, ST., M.IT
	: 4. Dr.Eng. Zulkifli Tahir, S.T., M.Sc

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 22 September 2023
Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.19750716 200212 1 004

Tembusan :
1. Arsip



KEMENTERIAN PENDIDIKAN , KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA SEMINAR HASIL

Pada hari ini **Rabu**, tanggal 27 September 2023 Pukul 11.00 WITA - Selesai bertempat di Ruang Lab. IOT Departemen Teknik Informatika Gowa, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Putri Alissya Sabrina

No. Stambuk : D121181317

Fakultas/Departemen : Teknik/Teknik Informatika

Judul Skripsi : "Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra MRI Menggunakan Single Board Computer"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Adnan,ST.,M.T.,Ph.D	Pemb I/Ketua	
2.	Prof. Dr.Eng.Intan Sari Areni,ST.,M.T	Pemb II/Sekretaris	2.....
3.	Dr.Eng.Muhammad Niswar,ST.,M.IT	Anggota	3.....
4.	Dr. Eng. Zulkifli Tahir,ST.,M.Sc	Anggota	4.....

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus** / ~~Tidak lulus~~ dengan nilai angka ... **4,67** dan huruf **A**.....

Gowa, 27 September 2023
Ketua/Sekretaris Panitia Ujian,

Adnan,ST.,M.T.,Ph.D



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : 1. Putri Alissya Sabrina D121181317

Judul Skripsi/T.A : "Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra MRI Menggunakan Single Board Computer"

Hari/Tanggal : Rabu, 27 September 2023

Jam : 11- 00 Wita – Selesai

Tempat : Ruang Lab. IOT Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Adnan, ST., M.T., Ph.D	1.
	Pembimbing II	2. Prof. Dr. Eng. Intan Sari Areni, ST., M.T	2.
II.	Anggota Penguji	3. Dr. Eng. Muhammad Niswar, ST., M.IT	3.
		4. Dr. Eng. Zulkifli Tahir, ST., M.Sc	4.

PANITIA UJIAN

Ketua,

Adnan, ST., M.T., Ph.D

Sekretaris,

Prof. Dr. Eng. Intan Sari Areni, ST., M.T



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id , <https://eng.unhas.ac.id>

SURAT PENUGASAN

No. 14240/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin Nomor : **29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin**, dengan ini menugaskan Saudara sebagai **PENGUJI/PANITIA UJIAN SARJANA** Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Adnan, ST., M.T., Ph.D
Pembimbing II / Sekretaris : 2. Prof. Dr.Eng. Intan Sari Areni, ST., M.T.
Anggota : 3. Dr.Eng. Ir. Muhammad Niswar, ST., M.IT.
4. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc.

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Putri Alissyah Sabrina D121 18 1317

Program Studi : Teknik Informatika

Judul Thesis/Skripsi : Evaluasi Kinerja Program Multithreading Pada
Penerapan Rekonstruksi Citra Mri Menggunakan
Single Board Computer

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).

3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 20 Juni 2024
a.n. Dekan

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP.197310101998021001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH



Balai
Sertifikasi
Elektronik

- Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
- UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1

"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Gowa, 20 Juni 2024

Nomor : 809/UN4.7.7.1/TD.06/2024
Lamp : -
Hal : Usulan Susunan Panitia/Penguji Ujian Sarjana

Yth. : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas
Di
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia/Penguji Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas nama :

Pembimbing I / Ketua : 1. Adnan, ST., M.T., Ph.D
Pembimbing II / Sekretaris : 2. Prof. Dr.Eng. Intan Sari Areni, ST., M.T.
Anggota : 3. Dr.Eng. Ir. Muhammad Niswar, ST., M.IT.
4. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc.

Untuk Bertugas sebagai Penguji/ Penanggung Ujian Sarjana bagi Mahasiswa :

Nama : Putri Alissyah Sabrina
Stambuk : D121 18 1317

Dengan Judul Skripsi :

“ Evaluasi Kinerja Program Multithreading Pada Penerapan Rekonstruksi Citra Mri
Menggunakan Single Board Computer “

Pada :
Hari/Tanggal : Senin, 24 Juni 2024
Jam : 08.30 Wita - Selesai
Tempat : Ruang Sidang Lab. CBS

Demikian penyampaian kami, atas perhatiannya diucapkan terimah kasih.

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.197507016 200212 1 004

Tembusan :
1. Arsip



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SKRIPSI

Pada hari ini Senin, tanggal 24 Juni 2024 Pukul 08.30 WITA - Selesai bertempat di Lab.CBS Departemen Teknik Informatika Gowa, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Putri Alissyah Sabrina
No. Stambuk : D121181317
Fakultas/Departemen : Teknik /Teknik Informatika
Judul Skripsi : "Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra Mri Menggunakan Single Board Computer"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Adnan,ST.,M.T.,Ph.D	Pemb I/Ketua	
2.	Prof. Dr.Eng. Intan Sari Areni,ST.,M.T	Pemb II/Sekretaris	2.
3.	Dr. Eng.Ir. Muhammad Niswar,ST.,M.IT	Anggota	3.
4.	Dr. Eng. Ir. Zulkifli Tahir,ST.,M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : **Lulus / Tidak lulus** dengan nilai angka 85.5 dan huruf A

Gowa, 24 Juni 2024

Ketua/Sekretaris Panitia Ujian,

Adnan,ST.,M.T.,Ph.D



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : 1. Putri Alissyah Sabrina D121181317

Judul Skripsi/T.A : “Evaluasi Kinerja Program Multithreading pada Penerapan Rekonstruksi Citra Mri Menggunakan Single Board Computer”

Hari/Tanggal : Senin, 24 Juni 2024

Jam : 08- 30 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
I.	Pembimbing I	1. Adnan,ST.,M.T.,Ph.D	1.....
	Pembimbing II	2. Prof. Dr.Eng. Intan Sari Areni,ST.,M.T	2.....
II.	Anggota Penguji	3. Dr. Eng.Ir. Muhammad Niswar,ST.,M.IT	3.....
		4. Dr. Eng. Ir. Zulkifli Tahir,ST.,M.Sc	4.....

PANITIA UJIAN

Ketua,

Adnan,ST.,M.T.,Ph.D

Sekretaris,

Prof. Dr.Eng. Intan Sari Areni,ST.,M.T