

*Seminar 1*

**IMPLEMENTASI FILTER SPASIAL LINEAR PADA VIDEO *STREAM*  
MENGUNAKAN *FPGA HARDWARE ACCELERATOR***



Oleh  
SULAEMAN  
H131 16 002

Pembimbing Utama	: Armin Lawi, M.Eng.
Pembimbing Pertama	: Dr. Diaraya, M.Ak.
Penguji	: 1. Dr. Hendra, S.Si., M.Kom. 2. Nur Hilal A Syahrir, S.Si., M.Si.

**PROGRAM STUDI ILMU KOMPUTER  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2020**

# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>ii</b>
<b>DAFTAR GAMBAR</b>	<b>iii</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
<b>BAB II TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 Landasan Teori	5
2.1.1 Citra Digital	5
2.1.2 Pengolahan Citra Digital	6
2.1.3 Filter Spasial	7
2.1.4 Kernel	8
2.1.5 Konvolusi	9
2.1.6 Video Streaming	10
2.1.7 FPGA	11
2.2 Penelitian Terkait	13
2.2.1 Spatial Filtering Based Boundary Extraction in Underwater Images for Pipeline Detection: FPGA Implementation	13
2.2.2 FPGA Implementation of Spatial Filtering techniques for 2D Images	13
2.2.3 Features of Image Spatial Filters Implementation on FPGA	13
2.2.4 An FPGA-Oriented Algorithm for Real-Time Filtering of Poisson Noise in Video Streams, with Application to X-Ray Fluoroscopy	14

2.2.5 A real-time video denoising algorithm with FPGA implementation for Poisson-Gaussian noise . . . . .	14
<b>BAB III METODE PENELITIAN . . . . .</b>	<b>16</b>
3.1 Tahapan Penelitian . . . . .	16
3.2 Waktu dan Lokasi Penelitian . . . . .	17
3.3 Rancangan Sistem . . . . .	17
3.4 Instrumen Penelitian . . . . .	17
<b>DAFTAR PUSTAKA . . . . .</b>	<b>18</b>

## DAFTAR GAMBAR

2.1.1 (a) Contoh citra biner, (b) contoh citra grayscale, (c) contoh citra warna. . . . .	6
2.1.2 Ilustrasi konvolusi pada citra. Sumber: <a href="https://indoml.com">https://indoml.com</a> . . . . .	10
2.1.3 Struktur FPGA. . . . .	12
2.1.4 FPGA Board Xilinx PYNQ-Z2. . . . .	12
3.1.1 Flowchart tahapan penelitian. . . . .	16
3.3.1 Rancangan sistem. . . . .	17

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Citra digital merupakan citra yang dihasilkan dari pengolahan secara digital dengan merepresentasikan citra secara numerik dengan nilai-nilai diskret. Suatu citra digital dapat direpresentasikan dalam bentuk matriks dengan fungsi  $f(x,y)$  yang terdiri dari M kolom dan N baris. Perpotongan antara baris dan kolom disebut sebagai *pixel* (Gonzalez and Woods, 2001). Setiap *pixel* mewakili sebuah warna, pada citra biner sebuah *pixel* hanya berwarna hitam atau putih saja. Pada citra grayscale warna sebuah *pixel* mewakili tingkat keabuannya. Sedangkan pada citra warna (RGB) setiap *pixel* mewakili warna yang merupakan kombinasi dari tiga warna dasar ,yaitu merah, hijau dan biru. Pada umumnya warna dasar dalam citra RGB menggunakan penyimpanan 8 bit untuk menyimpan data warna atau tingkat keabuannya, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna . Dewasa ini, citra digital dapat menggunakan 16 bit untuk menyimpan data warna atau tingkat keabuannya, hal ini menyebabkan semakin banyak gradasi warnanya sehingga citra yang dihasilkan memiliki tingkat warna yang jauh lebih banyak. Namun tentu saja hal ini mengakibatkan ukuran file citra digital yang dihasilkan juga menjadi semakin besar.

Pengolahan citra digital merupakan proses mengolah *pixel* di dalam citra secara digital untuk tujuan tertentu. Berdasarkan tingkat pemrosesannya pengolahan citra digital dikelompokkan menjadi tiga kategori, yaitu: *low-level*, *mid-level* dan pemrosesan *high-level*. Pemrosesan *low-level* dilakukan dengan operasi primitif seperti *image preprocessing* untuk mengurangi derau (*noise*), memperbaiki kontras citra dan mempertajam citra (*sharpening*). Pemrosesan *mid-level* melibatkan tugas-tugas seperti segmentasi atau mempartisi gambar menjadi beberapa bagian atau objek, deskripsi objek untuk dilakukan pemrosesan lanjutan, dan klasifikasi objek yang terdapat dalam citra digital. Pemrosesan *high-level* merupakan proses tingkat lanjut dari dua proses sebelumnya, dilakukan untuk mendapat informasi lebih yang terkandung dalam citra seperti *pattern recognition*, *template matching*, *image analysis* dan sebagainya (Gonzalez and Woods, 2001).

Konsep filter spasial pada pengolahan citra digital berasal dari penerapan transformasi Fourier untuk pemrosesan sinyal pada domain frekuensi. Istilah filter spasial ini digunakan untuk membedakan proses ini dengan filter pada domain frekuensi. Proses filter dilakukan dengan cara menggeser filter kernel dari titik ke titik dalam citra digital. Istilah *mask*, *kernel*, *template*, dan *window* merupakan istilah yang sama dan sering digunakan dalam pengolahan citra digital. Dalam penelitian ini penulis menggunakan istilah kernel untuk istilah tersebut. Konsep filter spasial linear mirip seperti konsep konvolusi pada domain frekuensi, dengan alasan tersebut filter spasial linear biasa disebut juga konvolusi sebuah kernel dengan citra digital (Gonzalez and Woods, 2001). Proses filter dalam pengolahan citra digital dilakukan dengan memanipulasi sebuah citra menggunakan kernel untuk menghasilkan citra yang baru, sehingga dengan kernel yang berbeda maka citra hasil yang didapat juga akan berbeda.

Video stream dapat dipandang sebagai serangkaian citra digital berturut-turut (Zhao, 2015). Berbeda dengan format video lainnya, video stream ini tidak disimpan pada media penyimpanan sebagai file video melainkan setiap *frame* langsung disalurkan dari sumber (*source*) ke penerima, dalam hal ini FPGA. Dengan menganggap Video stream adalah kumpulan citra digital (*frame*) maka dapat dilakukan metode pengolahan seperti pada citra digital, termasuk filter spasial. Setiap citra yang ditangkap dari source disebut sebagai *frame*, setiap *frame* ini dilakukan metode filter spasial kemudian hasilnya ditampilkan secara berkesinambungan sehingga tampak seperti video yang telah difilter.

Frame per second (*fps*) atau *frame rate* adalah banyaknya *frame* yang ditampilkan per detik. Semakin tinggi *fps* sebuah video maka semakin halus pula gerakan yang dapat ditampilkan karena dibentuk dari *frame* yang lebih banyak, namun dengan jumlah *frame* yang lebih besar tentu dibutuhkan juga *resource* yang lebih besar dalam pengolahan video tersebut (Kowalczyk, Przewlocka, and Krvjak, 2018).

Field Programmable Gate Arrays atau FPGA adalah perangkat semikonduktor yang berbasis *matriks configurable logic block* (CLBs) yang terhubung melalui interkoneksi yang dapat diprogram. FPGA dapat diprogram ulang dengan aplikasi atau fungsi yang diinginkan setelah *manufacturing*. Fitur ini yang membedakan FPGA

dengan *Application Specific Integrated Circuits* (ASICs), yang dibuat khusus untuk tugas tertentu saja (Xilinx, 2020).

FPGA Xilinx PYNQ-Z2 yang digunakan pada penelitian ini secara *official* dapat menerima input video stream dengan resolusi 720p. Setiap *frame* yang diterima dari *source* akan dilakukan proses filter spasial, kemudian hasilnya disalurkan melalui HDMI output untuk kemudian ditampilkan. Video hasil filter spasial yang ditampilkan melalui port HDMI Output akan mengalami penurunan *fps*, hal ini disebabkan adanya penambahan jeda waktu komputasi untuk proses filter yang dilakukan di setiap *frame*. Pada kesempatan ini penulis ingin melakukan "Implementasi Filter Spasial Linear pada Video Stream menggunakan FPGA Hardware Accelerator".

## **1.2 Rumusan Masalah**

Adapun rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana cara implementasi filter spasial linear pada video stream menggunakan FPGA?
2. Bagaimana kinerja FPGA dalam mengimplementasikan filter spasial linear pada video stream?

## **1.3 Batasan Masalah**

Berikut ini merupakan beberapa batasan dalam penelitian ini.

1. Filter kernel yang digunakan berukuran 3x3.
2. Video stream yang digunakan dalam penelitian ini beresolusi 720p.
3. Setiap frame dari video stream diubah menjadi citra grayscale sebelum dilakukan penerapan filter spasial.
4. FPGA yang digunakan adalah Xilinx PYNQ-Z2 dengan processor 650MHz dual-core ARM Cortex-A9.

## **1.4 Tujuan Penelitian**

Adapun tujuan dari penelitian ini yaitu:

1. Mampu melakukan implementasi filter spasial linear pada video stream menggunakan FPGA.
2. Mengetahui kinerja FPGA dalam mengimplementasikan filter spasial linear pada video stream.

## **1.5 Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat memberikan pemahaman tentang penerapan filter spasial pada video stream. Selain itu, penelitian ini juga diharapkan dapat menjadi rujukan untuk melihat kinerja FPGA dalam mengimplementasikan filter spasial linear pada video stream.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 Citra Digital**

Citra digital dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dengan x dan y adalah kordinat spasial, dan amplitudo f di titik kordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada citra tersebut (Putra, 2010). Pada umumnya warna dasar dalam citra RGB menggunakan penyimpanan 8 bit untuk menyimpan data warna, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna . Dewasa ini, citra digital dapat menggunakan 16 bit untuk menyimpan data warna dasarnya, hal ini menyebabkan semakin banyak gradasi warnanya sehingga citra yang dihasilkan memiliki tingkat warna yang jauh lebih banyak. Namun tentu saja hal ini mengakibatkan ukuran file citra digital yang dihasilkan juga menjadi semakin besar walaupun dengan resolusi yang sama. Berdasarkan jenis warnanya citra digital dibagi menjadi 3 jenis:

##### **a. Citra Biner (monokrom)**

Banyaknya dua warna, yaitu hitam dan putih. Warna hitam direpresentasikan dengan 1 dan warna putih direpresentasikan dengan 0. Dibutuhkan 1 bit di memori untuk menyimpan warna ini. Contoh citra biner dapat dilihat pada gambar 2.1.1(a).

##### **b. Citra Grayscale**

Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk. Pada umumnya citra digital grayscale menggunakan 8 bit memori dengan derajat keabuan dari 0 sampai 255. Contoh citra grayscale dapat dilihat pada gambar 2.1.1(b).

### c. Citra Warna

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RG8 = Red Green Blue). Setiap warna dasar menggunakan penyimpanan 8 bit, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak  $255 \times 255 \times 255 = 16$  juta warna lebih. Dibutuhkan  $3 \times 8 = 24$  bit di memori untuk menyimpan sebuah data warna ini. Contoh citra warna dapat dilihat pada gambar 2.1.1(c).



Gambar 2.1.1: (a) Contoh citra biner, (b) contoh citra grayscale, (c) contoh citra warna.

### 2.1.2 Pengolahan Citra Digital

Pengolahan citra digital merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. Berdasarkan tingkat pemrosesannya pengolahan citra digital dikelompokkan menjadi tiga kategori, yaitu: *low-level*, *mid-level* dan pemrosesan *high-level*. Pemrosesan *low-level* dilakukan dengan operasi primitif seperti *image preprocessing* untuk mengurangi derau (*noise*), memperbaiki kontras citra dan mempertajam citra (*sharpening*). Karakteristik dari pemrosesan *low-level* yaitu keluaran atau hasil dari pemrosesannya berupa citra digital. Pemrosesan *mid-level* melibatkan tugas-tugas seperti segmentasi (mempartisi gambar menjadi beberapa bagian atau objek), deskripsi objek untuk dilakukan pemrosesan lanjutan, dan klasifikasi objek dalam citra digital. Karakteristik dari pemrosesan *mid-level* yaitu keluaran atau hasilnya berupa atribut atau fitur seperti, kontur, tepi, atau objek yang terdapat dalam citra tersebut. Pemrosesan *high-level* merupakan proses tingkat

lanjut dari dua proses sebelumnya, dilakukan untuk mendapat informasi lebih yang terkandung dalam citra (Gonzalez and Woods, 2001).

### 2.1.3 Filter Spasial

Konsep filter spasial pada pengolahan citra digital berasal dari penerapan transformasi Fourier untuk pemrosesan sinyal pada domain frekuensi. Istilah filter spasial ini digunakan untuk membedakan proses ini dengan filter pada domain frekuensi. Proses filter dilakukan dengan cara menggeser filter kernel dari titik ke titik dalam citra digital. Istilah *mask*, *kernel*, *template*, dan *window* merupakan istilah yang sama dan sering digunakan dalam pengolahan citra digital (Gonzalez and Woods, 2001). Dalam penelitian ini penulis menggunakan istilah kernel untuk istilah tersebut.

Proses filter dalam pengolahan citra digital dilakukan dengan memanipulasi sebuah citra menggunakan kernel untuk menghasilkan citra yang baru, sehingga dengan kernel yang berbeda maka citra hasil yang didapat juga akan berbeda.

#### 2.1.3.1 Operator Linear dan Non-linear

Didefinisikan  $H$  sebuah operator dengan input dan output adalah citra digital.  $H$  dikatakan operator linear jika untuk sembarang gambar  $f$  dan  $g$ , dan untuk sembarang skalar  $a$  dan  $b$  berlaku,

$$H(af + bg) = aH(f) + bH(g) \quad (2.1.1)$$

Dengan kata lain hasil dari operator linear dengan jumlahan dua buah citra (yang telah dikali dengan konstanta  $a$  dan  $b$ ) identik dengan hasil operator linear pada masing-masing gambar, dikali dengan konstanta yang sama, kemudian hasilnya dijumlahkan. Sebagai contoh, sebuah operator dengan fungsi yang menjumlahkan  $K$  citra adalah operator linear. Operator yang menghitung nilai absolut dari perbedaan dua gambar adalah tidak linear. Operator yang tidak memenuhi persamaan (2.1.1) dikatakan non-linear (Gonzalez and Woods, 2001).

## 2.1.4 Kernel

### 2.1.4.1 Averaging Filter

Averaging filter atau biasa juga disebut *box filter* adalah salah satu filter yang digunakan untuk menghaluskan citra dan mengurangi derau. Secara sederhana nilai sebuah piksel yang baru adalah nilai rata-rata dari nilai piksel tersebut dengan nilai piksel tetangganya (Kowalczyk, Przewlocka, and Krvjak, 2018). Berikut ini kernel *averaging* filter yang digunakan dalam penelitian ini:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

### 2.1.4.2 Gaussian Blur

Filter ini juga digunakan untuk menghaluskan citra dan mengurangi derau. Idennya mirip seperti averaging filter, nilai piksel yang baru dibentuk dari nilai pixel tetangganya, tetapi dengan memberikan bobot yang lebih kuat pada nilai pikselnya sendiri diikuti dengan bobot yang lebih rendah pada piksel atas, bawah dan sampingnya (Ustyukov, Efimov, and Kolchaev, 2019). Berikut kernel gaussian blur filter yang digunakan dalam penelitian ini:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

### 2.1.4.3 Filter Sobel

Filter Sobel termasuk *high-pass* filter yang umum digunakan untuk deteksi tepi pada citra. Sobel memiliki dua kernel untuk deteksi tepi yaitu kernel sobel vertikal untuk mendeteksi tepi secara vertikal dan kernel sobel horizontal yang mendeteksi tepi secara horizontal. Berikut ini kernel sobel vertikal dan horizontal yang digunakan

dalam penelitian ini:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

#### 2.1.4.4 Laplacian High-pass

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. C

#### 2.1.4.5 Gaussian High-pass

$$\frac{1}{16} \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bi

### 2.1.5 Konvolusi

Konsep filter spasial linear mirip seperti konsep konvolusi pada domain frekuensi, dengan alasan tersebut filter spasial linear biasa disebut juga konvolusi sebuah kernel dengan citra digital (Gonzalez and Woods, 2001). Konvolusi pada fungsi  $f(x)$  dan  $g(x)$  didefinisikan sebagai berikut:

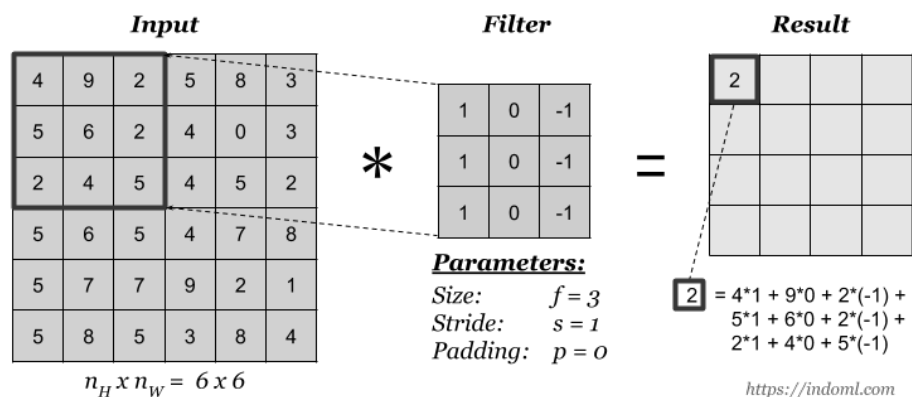
$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a)g(x-a)da \quad (2.1.2)$$

dimana tanda  $*$  menyatakan operator konvolusi, dan peubah  $a$  adalah peubah bantu. Untuk fungsi diskrit, konvolusi didefinisikan sebagai berikut:

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a)g(x - a) \quad (2.1.3)$$

Pada operasi konvolusi diatas,  $g(x)$  disebut kernel konvolusi atau filter kernel. Kernel  $g(x)$  dioperasikan secara bergeser pada sinyal masukan  $f(x)$ . Jumlah perkalian kedua fungsi pada setiap titik merupakan hasil konvolusi yang dinyatakan dengan keluaran  $h(x)$  (Rinaldi, 2004).

Pada gambar (2.1.2) diilustrasikan bagaimana proses konvolusi pada citra digital yang direpresentasikan dalam bentuk matriks. Operasi konvolusi dilakukan pada matriks input berukuran 6x6 dengan filter berukuran 3x3. Hasil konvolusinya ditampilkan pada matriks result.



Gambar 2.1.2: Ilustrasi konvolusi pada citra. Sumber: <https://indoml.com>

Jika hasil konvolusi menghasilkan nilai pixel negatif, maka nilai tersebut dijadikan 0, sebaliknya jika hasil konvolusi menghasilkan nilai pixel yang melebihi nilai keabuan keabuan maksimum, maka nilai tersebut dijadikan ke nilai keabuan maksimum (Sutoyo et al., 2009).

## 2.1.6 Video Streaming

Video stream dapat dipandang sebagai serangkaian citra digital berturut-turut (Zhao, 2015). Berbeda dengan format video lainnya, video stream ini tidak disimpan pada media penyimpanan sebagai file video melainkan langsung disalurkan setiap

framennya dari sumber (source) ke penerima, dalam hal ini FPGA. Dengan menganggap Video stream adalah kumpulan citra digital (frame) maka dapat dilakukan metode pengolahan seperti pada citra digital, termasuk filter spasial.

#### 2.1.6.1 Frame Per Second (FPS)

Frame per second merupakan bla bla ...

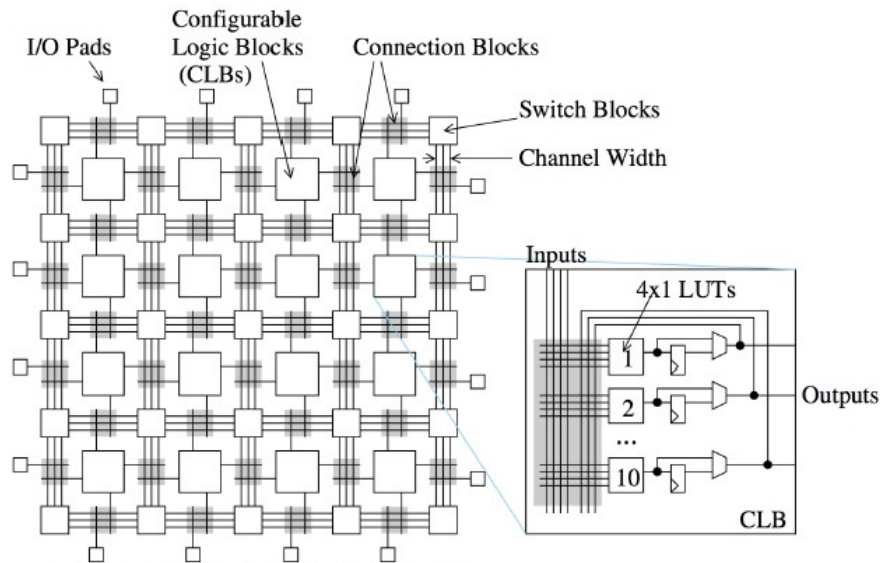
$$fps = \frac{frames}{seconds} = \frac{1}{\frac{seconds}{frames}} \quad (2.1.4)$$

#### 2.1.7 FPGA

Field Programmable Gate Arrays atau FPGA adalah perangkat semikonduktor yang berbasis *matriks configurable logic block* (CLBs) yang terhubung melalui interkoneksi yang dapat diprogram. FPGA dapat diprogram ulang ke aplikasi atau fungsi yang diinginkan setelah *manufacturing*. Fitur ini yang membedakan FPGA dengan *Application Specific Integrated Circuits* (ASICs), yang dibuat khusus untuk tugas tertentu saja (Xilinx, 2020).

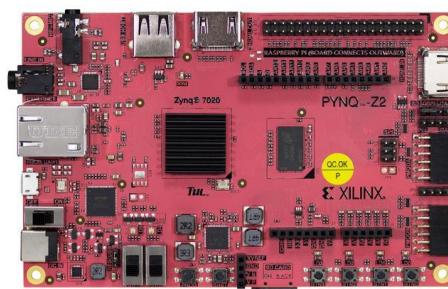
Sebuah *microprocessor* menerima instruksi berupa kode 1 atau 0, kode-kode ini selanjutnya diinterpretasikan oleh komputer untuk menjalankan perintah yang diberikan. *Microprocessor* ini membutuhkan intruksi berupa kode secara terus menerus untuk menjalankan fungsinya. Sedangkan pada FPGA hanya dibutuhkan sekali konfigurasi *chip* setiap kali dinyalakan. Membuat atau mengunduh *bitstream* yang menentukan fungsi logika dilakukan oleh *logic elements* (LEs), sebuah sirkuit dapat dibuat dengan mengabungkan beberapa LEs menjadi satu kesatuan. Setelah *bitstream* dipasang, FPGA tidak perlu lagi membaca instruksi berupa 1 dan 0, berbeda dengan *microprocessor* yang selalu membutuhkan instruksi (Cheung, 2019). Secara tradisional, untuk membuat sebuah desain FPGA, aplikasi dideskripsikan menggunakan Hardware Description Language (HDL) seperti Verilog atau VHDL sehingga menghasilkan sebuah *bitstream* FPGA.

Pada FPGA terdahulu tidak terdapat processor (CPU) untuk menjalankan software apapun, sehingga ketika ingin mengimplementasikan aplikasi haruslah merancang sirkuit dari awal, seperti mengonfigurasi FPGA sesederhana gerbang



Gambar 2.1.3: Struktur FPGA.

logika OR atau serumit multi-core processor (Biswas, 2019). Dewasa ini telah dikembangkan *FPGA Development Board* atau biasa disebut juga *FPGA Board* yaitu teknologi FPGA yang dirangkai dalam sebuah *board* dan dilengkapi dengan *microprocessor* dan beberapa interface *IO* untuk menjangkau tugas tertentu. Umumnya FPGA Board telah dilengkapi dengan interface untuk mengakses dan menerapkan desain sirkuitnya. Xilinx, Altera dan Intel adalah produsen FPGA Board yang terkenal. FPGA Board yang digunakan dalam penelitian ini yaitu Xilinx PYNQ-Z2 dengan Jupyter Notebook sebagai interface untuk mengakses dan menjalankan program pada penelitian ini.



Gambar 2.1.4: FPGA Board Xilinx PYNQ-Z2.



## **2.2 Penelitian Terkait**

### **2.2.1 Spatial Filtering Based Boundary Extraction in Underwater Images for Pipeline Detection: FPGA Implementation**

Pipa bawah air diletakkan di dasar laut untuk tujuan pengangkutan minyak bumi dan gas menyebrangi lautan. Pipa perlu terus dipantau untuk menghindari gangguan dalam proses transportasi. Gambar dasar laut dapat diperoleh dengan menggunakan kamera dan dengan memproses gambar yang diperoleh dapat membantu dalam mendeteksi pipa. Penelitian ini membahas tentang metode pemrosesan citra untuk deteksi pipa bawah laut dari gambar bawah laut yang diambil oleh kendaraan bawah laut yang dapat digunakan sebagai langkah awal untuk melacak saluran pipa. Implementasinya berhasil dilakukan pada Field Programmable Gate Array (FPGA) berbasis development board (Raj., Abraham, and Supriya, 2016).

### **2.2.2 FPGA Implementation of Spatial Filtering techniques for 2D Images**

Berbagai teknik filter telah menjadi inti dari pemrosesan citra sejak awal teknik peningkatan citra (*image enhancement*). Filter spasial domain pada pemrosesan citra digital digunakan dalam banyak kepentingan seperti mempertajam citra, menghaluskan citra, menghilangkan derau dan sebagainya. Fleksibilitas dari metode filter spasial sering dibandingkan dengan domain transformasi karena dapat digunakan untuk filter linear dan filter non-linear. Penghalusan citra dilakukan dengan langsung memanipulasi nilai intensitas dari citra asli dengan sebuah kernel filter. Hasilnya yaitu berkurangnya detail kecil dan derau pada citra. Penelitian ini tentang penerapan berbagai macam operator filter spasial. Hasilnya didasarkan pada konsumsi perangkat keras, kecepatan desain masing-masing arsitektur. Ukuran kualitas citra didapat dengan membandingkan output dari Matlab dan output dari Xilinx FPGA dan dengan menghitung MSE (Sadangi et al., 2017).

### **2.2.3 Features of Image Spatial Filters Implementation on FPGA**

Penelitian ini menyajikan fitur-fitur implementasi filter spasial pada citra dengan Programmable Logic Integrated Circuits (FPGA). Solusi yang disajikan

memungkinkan untuk membuat arsitektur kristal dengan performa tinggi untuk algoritma filter spasial. Hasilnya menunjukkan kelebihan menggunakan *programmable logic* dalam tugas pemrosesan gambar (Ustyukov, Efimov, and Kolchaev, 2019).

#### **2.2.4 An FPGA-Oriented Algorithm for Real-Time Filtering of Poisson Noise in Video Streams, with Application to X-Ray Fluoroscopy**

Pada penelitian ini dibahas tentang algoritma baru untuk *real-time filtering* pada video yang rusak karena *poison noise*. Algoritma yang disajikan efektif dalam penanganan derau, dan ini secara ideal cocok dengan implementasi hardware, dan dapat diimplementasikan pada FPGA kecil yang memiliki hardware resource yang terbatas. Pada penelitian ini penerapan algoritma menggunakan hasil X-ray fluoroscopy sebagai studi kasus. Hasil implementasi menggunakan yang StratixIV FPGA menunjukkan bahwa sistem hanya menggunakan, paling banyak, 22% dari sumber daya perangkat, dalam implementasi real-time filtering pada video stream 1024x1024 @49fps. Sebagai perbandingan, implementasi filter berbasis FIR pada FPGA yang sama dan dengan video stream yang serupa, dibutuhkan 80% resource logic pada FPGA (Castellano et al., 2019).

#### **2.2.5 A real-time video denoising algorithm with FPGA implementation for Poisson-Gaussian noise**

Fungsi denoising dalam perangkat citra digital harus memperhatikan penggunaan sumber daya dan kemampuan real-time pengurangan derau. Pada penggunaan umum metode denoising yaitu *Pixel Similarity Weighted Frame Average* (PSWFA). Pada penelitian ini, dilakukan peningkatan kemampuan denoising dari PSWFA menggunakan pre-filter yang mengandung operator *downsampling* dan small Gaussian filter. Bahkan, derau pada perangkat citra digital adalah *signal dependent* dan biasanya dimodelkan sebagai distribusi Poisson-Gaussian, dikenalkan generalisasi transformasi Asncombe untuk menghilangkan signal dependency dengan merender *noise variance constant*. Transformasi citra dapat mengalami gangguan oleh derau Gaussian. Untuk memasang algoritma ini pada perangkat keras, sebelumnya diimplementasikan algoritma ini pada Spartan-6 FPGA untuk evaluasi. Dilakukan

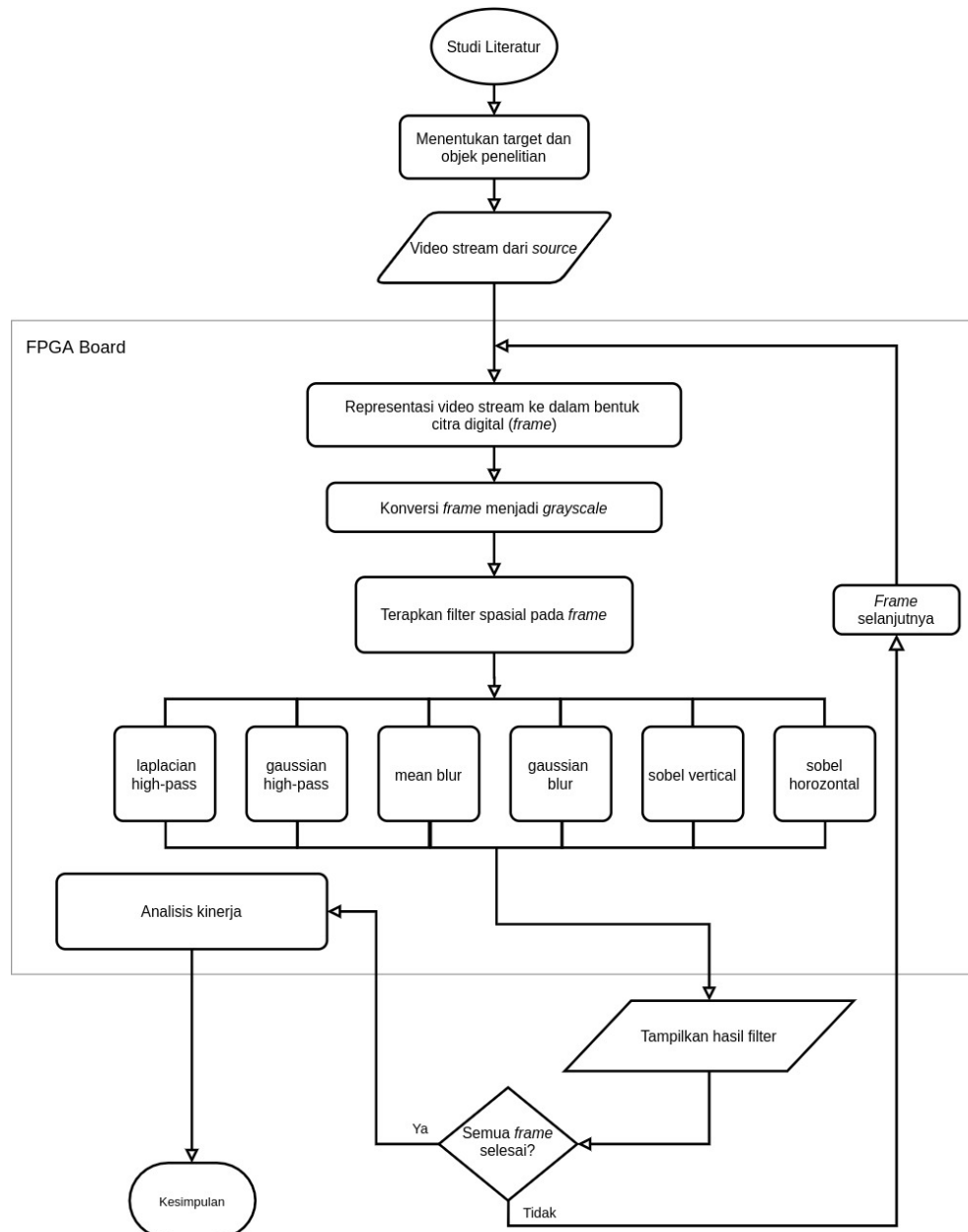
juga perbandingan dengan beberapa metode denoising yang sudah ada. Evaluasi selanjutnya untuk kemampuan denoising, algoritma ini dibandingkan dengan beberapa algoritma state-of-art yang tidak diimplementasikan pada FPGA tetapi memiliki performa yang baik pada personal komputer. Hasil eksperimen pada kedua simulasi video berderau dan video yang ditangkap pada pencahayaan yang kurang menunjukkan tingkat efektifitas pada algoritma ini, terkhusus pada pemrosesan derau berskala besar (Tan et al., 2014).

## BAB III

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian

Tahapan dalam penelitian ini dapat dilihat pada gambar 3.1.1.



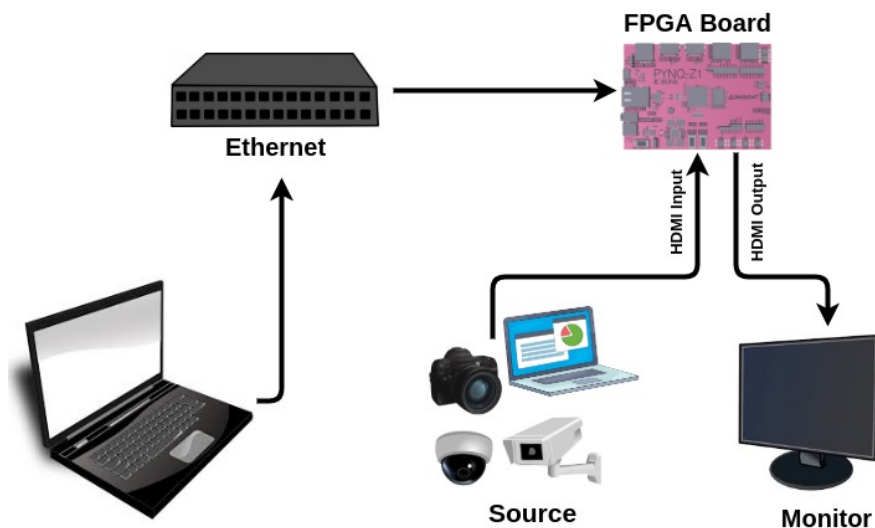
Gambar 3.1.1: Flowchart tahapan penelitian.

### 3.2 Waktu dan Lokasi Penelitian

Penelitian ini dilaksanakan dari bulan Juni 2020 sampai dengan bulan Agustus 2020. Lokasi penelitian dilakukan di Laboratorium Rekayasa Perangkat Lunak Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin Makassar.

### 3.3 Rancangan Sistem

Pada penelitian ini akan dibangun suatu sistem untuk mengimplementasikan filter spasial linear pada FPGA, dapat dilihat pada gambar 3.3.1.



Gambar 3.3.1: Rancangan sistem.

Video stream dari *source* akan disalurkan melalui port HDMI Input pada FPGA Board, kemudian video stream tersebut akan diolah dengan menerapkan filter spasial linear pada setiap framenya. Setiap frame yang telah dilakukan filter spasial akan dialirkan ke monitor untuk kemudian ditampilkan. Selanjutnya dilakukan analisis kinerja pada FPGA. FPGA board yang digunakan dalam penelitian ini dapat diakses dengan *ssh* pada port 22 atau dengan Jupyter Notebook melalui browser.

### 3.4 Instrumen Penelitian

Instrumen penelitian ini yaitu:

1. Kebutuhan perangkat lunak:

- a. Ubuntu 16, sebagai OS pada FPGA Board
  - b. Python 3.6 (dengan modul OpenCV dan Numpy)
  - c. Jupyter Notebook, sebagai interface untuk mengakses FPGA Board
2. Kebutuhan perangkat keras:
- a. FPGA Board Xilinx PYNQ-Z2
  - b. Micro SD Card 8Gb (sebagai media penyimpanan OS pada FPGA Board)
  - c. Monitor Eksternal (untuk menampilkan hasil filter dari FPGA)
  - d. Laptop Lenovo Ideapad 320 (sebagai source video stream)
  - e. 2 buah kabel HDMI (untuk HDMI input dan HDMI output pada FPGA)

## DAFTAR PUSTAKA

- Biswas, Priyabrata (2019). *Introduction to FPGA and its Architecture*. <https://towardsdatascience.com/introduction-to-fpga-and-its-architecture-20a62c14421c>. Accessed on 2020-06-18.
- Castellano, G. et al. (Jan. 2019). “An FPGA-Oriented Algorithm for Real-Time Filtering of Poisson Noise in Video Streams, with Application to X-Ray Fluoroscopy”. In: *Circuits, Systems, and Signal Processing*. DOI: 10.1007/s00034-018-01020-x.
- Cheung, Peter (2019). *Introduction to FPGAs*. [http://www.ee.ic.ac.uk/pcheung/teaching/ee2\\_digital/Lecture2-IntroductiontoFPGAs.pdf](http://www.ee.ic.ac.uk/pcheung/teaching/ee2_digital/Lecture2-IntroductiontoFPGAs.pdf). Accessed on 2020-04-19.
- Gonzalez, Rafael C. and Richard E. Woods (2001). *Digital Image Processing*. 2nd. ISBN-13: 978-0201180756. Upper Saddle River, New Jersey 07458: Prentice Hall.
- Kowalczyk, Marcin, Dominika Przewlocka, and Tomasz Krvjak (Oct. 2018). “Real-Time Implementation of Contextual Image Processing Operations for 4K Video Stream in Zynq UltraScale+ MPSoC”. In: *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*. DOI: 10.1109/DASIP.2018.8597105.
- Putra, Darma (2010). *Pengolahan Citra Digital*. ISBN-13: 978-979-29-1443-6. Jl. Beo 38-40, Yogyakarta 55281: Penerbit Andi.
- Raj., S.M. Alex, Rita Maria Abraham, and M.H. Supriya (Sept. 2016). “Spatial filtering based Boundary Extraction in Underwater Images for Pipeline Detection: FPGA Implementation”. In: *International Journal of Computer Science and Information Security (IJCSIS)*. Vol. 14, No. 9.
- Rinaldi, Munir (2004). *Pengolahan citra digital dengan pendekatan algoritmik*. ISBN: 979-3338296. Bandung: Penerbit Informatika.
- Sadangi, Sushant et al. (May 2017). “FPGA Implementation of Spatial Filtering techniques for 2D Images”. In: *IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT)*.

- Sutoyo, T. et al. (2009). *Teori Pengolahan Citra Digital*. ISBN-13: 978-979-29-0974-6. Jl. Beo 38-40, Yogyakarta 55281: Penerbit Andi.
- Tan, Xin et al. (Feb. 2014). “A real-time video denoising algorithm with FPGA implementation for Poisson-Gaussian noise”. In: *J Real-Time Image Proc.* DOI: 10.1007/s11554-014-0405-2.
- Ustyukov, Dmitry I., Alex I. Efimov, and Dmitry A. Kolchaev (June 2019). “Features of Image Spatial Filters Implementation on FPGA”. In: *MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO)*.
- Xilinx (2020). *Field Programmable Gate Array (FPGA)*. <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>. Accessed on 2020-04-17.
- Zhao, Jin (Apr. 2015). “Video/Image Processing on FPGA”. Master thesis. Worcester Polytechnic Institute.