



**FUNDAMENTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT  
DEPARTMENT OF ELECTRICAL ENGINEERING  
UNIVERSITAS INDONESIA**

**Image Downscaler Using Box Sampling Algorithm**

**GROUP PA06**

<b>Alexander Christian</b>	<b>2306267025</b>
<b>Muhammad Iqbal Alfajri</b>	<b>2306250705</b>
<b>Muhammad Raditya Alif N</b>	<b>2306212745</b>
<b>Zhafira Zahra Alfarisy</b>	<b>2306250636</b>

## **PREFACE**

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa atas segala anugerah-Nya, yang membantu kami menyelesaikan laporan proyek akhir ini dengan judul "Image Downscaler Using Box Sampling Algorithm". Kami juga mengucapkan terima kasih kepada para asisten laboratorium dan rekan-rekan yang telah memberikan dukungan dan kontribusi berharga dalam penyelesaian laporan ini.

Laporan ini disusun sebagai bagian dari pemenuhan modul 10 dalam Proyek Akhir Praktikum Perancangan Sistem Digital Tahun Ajaran 2024/2025. Fokus utama dari proyek ini adalah perancangan dan implementasi sistem downscaling gambar berbasis algoritma box sampling, yang dirancang untuk diterapkan pada FPGA dengan menggunakan bahasa pemrograman HDL. Proyek ini bertujuan untuk mengurangi ukuran gambar, yang akan berimbas pada penghematan bandwidth dalam transmisi data, serta mengurangi penggunaan ruang penyimpanan dan beban pemrosesan pada sistem penerima gambar.

Laporan ini mencakup penjelasan terkait latar belakang permasalahan, deskripsi sistem yang dirancang, serta hasil-hasil yang diperoleh dari implementasi dan analisis sistem tersebut. Kami menyadari adanya keterbatasan dalam pengetahuan dan pengalaman yang kami miliki, sehingga kami sangat mengharapkan kritik dan saran konstruktif yang dapat menjadi bahan evaluasi untuk perbaikan di masa depan. Kami juga mohon maaf atas segala kekurangan yang terdapat dalam laporan ini.

Depok, December 4, 2024

Group PA06

# **TABLE OF CONTENTS**

## **CHAPTER 1: INTRODUCTION**

- 1.1 Background
- 1.2 Project Description
- 1.3 Objectives
- 1.4 Roles and Responsibilities

## **CHAPTER 2: IMPLEMENTATION**

- 2.1 Equipment
- 2.2 Implementation

## **CHAPTER 3: TESTING AND ANALYSIS**

- 3.1 Testing
- 3.2 Result
- 3.3 Analysis

## **CHAPTER 4: CONCLUSION**

## **REFERENCES**

## **APPENDICES**

- Appendix A: Project Schematic
- Appendix B: Documentation

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 BACKGROUND**

Image Downscaler merupakan hardware yang dibuat untuk menurunkan ukuran sebuah gambar dengan menurunkan jumlah pixelnya. Downscaler ini menggunakan algoritma box sampling yang bekerja dengan mengelompokkan beberapa pixel dalam area tertentu dan menghitung nilai rata-rata atau representasi lain untuk menghasilkan pixel yang lebih sedikit tetapi tetap representatif. Teknologi ini sangat berguna dalam aplikasi yang memerlukan pemrosesan gambar efisien, seperti perangkat IoT atau sistem pemantauan dengan kapasitas penyimpanan terbatas.

Proyek kami bertujuan untuk mengimplementasikan fungsi downscaling sebuah gambar menggunakan VHDL. Dengan pendekatan ini, kami dapat menciptakan sistem digital yang mampu melakukan downscaling gambar secara cepat dan akurat menggunakan box sampling algorithm, tanpa bergantung pada daya komputasi CPU. Scaling factor juga akan menjadi input yang menentukan seberapa besar gambar akan di-downscale.

### **1.2 PROJECT DESCRIPTION**

Proyek ini berfokus pada desain dan implementasi Image Downscaler, sebuah sistem perangkat keras yang dikembangkan khusus untuk mengurangi ukuran gambar dengan menurunkan jumlah pixelnya. Tujuan utama dari downscaling ini adalah untuk mengoptimalkan gambar untuk aplikasi transmisi, seperti mengurangi kebutuhan bandwidth untuk komunikasi, meminimalkan ruang penyimpanan server, dan mengurangi kebutuhan daya pemrosesan selama transmisi dan penerimaan gambar.

Algoritma inti yang digunakan untuk downscaling dalam proyek ini adalah box sampling, yang merupakan teknik pemrosesan gambar yang sederhana dan efisien. Box sampling bekerja dengan mengelompokkan sejumlah pixel dalam area tertentu dan menghitung nilai rata-rata atau nilai representatif lainnya, sehingga mengurangi jumlah pixel sambil tetap mempertahankan detail gambar yang penting. Metode ini sangat berguna dalam aplikasi yang memerlukan minimisasi kompleksitas komputasi dan waktu pemrosesan,

seperti perangkat Internet of Things (IoT), sistem pemantauan, dan platform dengan daya dan penyimpanan terbatas.

Sistem downscaling ini diimplementasikan menggunakan VHDL (VHSIC Hardware Description Language), yang ditujukan untuk FPGA (Field Programmable Gate Array) untuk penerapan hardware. Dengan FPGA, sistem ini dapat melakukan pemrosesan paralel, sehingga dapat melakukan downscaling yang cepat dengan ketergantungan minimal pada CPU, yang cocok untuk sistem embedded. Scaling factor—sebuah parameter input—menentukan sejauh mana downscaling diterapkan pada gambar, sehingga aplikasi yang fleksibel dan dapat diskalakan tergantung pada bandwidth transmisi dan kapasitas penyimpanan.

Melalui proyek ini, kami bertujuan untuk menciptakan solusi dengan kompleksitas rendah dan kinerja tinggi yang ideal untuk aplikasi transmisi gambar, terutama di mana konservasi bandwidth sangat penting. Fokus utama dari proyek ini adalah untuk mencapai proses downscaling gambar yang efisien tanpa mengorbankan integritas visual dari data yang ditransmisikan.

### **1.3 OBJECTIVES**

The objectives of this project are as follows:

1. Sebagai pemenuhan nilai dalam Praktikum Perancangan Sistem Digital.
2. Mengimplementasikan pemrograman VHDL untuk desain sistem downscaling gambar.
3. Merancang Hardware Accelerator untuk menjalankan Image Downscaling menggunakan algoritma Box Sampling.
4. Menyediakan solusi untuk transmisi gambar dengan mengurangi ukuran gambar tanpa mengorbankan kualitas visual.

### **1.4 ROLES AND RESPONSIBILITIES**

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Role 1	Code	Alexander Christhian
Role 2	Laporan dan PPT	Muhammad Iqbal Alfajri
Role 3	README, Laporan, dan PPT	Muhammad Raditya Alif Nugroho
Role 4	Laporan dan PPT	Zhafira Zahra Alfarisy

Table 1. Roles and Responsibilities

## CHAPTER 2

### IMPLEMENTATION

#### 2.1 EQUIPMENT

The tools that are going to be used in this project are as follows:

- VS Code
- Quartus Prime
- ModelSim
- Vivado

#### 2.2 IMPLEMENTATION

Secara keseluruhan, proyek ini bertujuan untuk melakukan *downscale* terhadap sebuah gambar. Langkah yang dilakukan adalah dengan mengurangi jumlah pixel namun tetap mempertahankan representasi aslinya. Tujuan ini bisa tercapai menggunakan algoritma *box sampling* yang menggabungkan beberapa pixel menjadi sebuah box dan menghitung nilai rata-rata setiap boks untuk membuat gambar dengan resolusi lebih kecil.

##### **Komponen**

a. Image Downscaler:

Komponen utama yang mengendalikan keseluruhan proses *downscaling* dan berperan sebagai kode “main”. Tugasnya menangani gambar input, state machine, dan koordinat yang berkaitan dengan Box Sampling.

b. Box Sampling

Komponen yang benar-benar melakukan *downscaling* menggunakan algoritma box sampling. Tugasnya memproses gambar input dan menghasilkan versi yang sudah di-downscale

c. RGB Package

Komponen berupa package yang mendefinisikan tipe data dan function untuk menangani RGB dari gambar input. Package ini juga menangani dimensi dari image dan memperkirakan sizenya setelah di-downscale.

d. Testbench

Komponen pengujian yang mensimulasikan behavior dari Image Downscaler. Tugasnya memberikan gambar input dan memeriksa output yang diberikan.

## **Implementasi dari Modul**

a. Dataflow

- Digunakan ketika BoxSampling memetakan port-portnya ke sinyal seperti clk, rst, width, height, downscale\_factor, r\_array, downscaled\_image, dan box\_sampling\_done.
- Data mengalir dari sinyal input (input\_width, input\_height, image\_input) ke sinyal internal (width, height, r\_array) dan akhirnya menuju sinyal output (downscaled\_width, downscaled\_height, output\_image).

b. Behavioral

- Pada process utama mendeskripsikan behavior dari komponen ImageDownscaler.
- State machine di dalam proses tersebut mendefinisikan berbagai state seperti IDLE, PROCESS\_INPUT, dan DOWNSCALE, serta transisi antar state berdasarkan kondisi tertentu.

c. Structural

- BoxSampling menggunakan aspek structural dengan menghubungkan komponen BoxSampling ke modul ImageDownscaler.
- Port-port dari BoxSampling dipetakan ke sinyal di dalam modul ImageDownscaler, menggunakan prinsip structural.

d. Testbench

- Testbench digunakan pada secara hierarchy paling atas untuk menyimulasikan downscaling algorithm dari component - component yang dibuat.

e. Looping

- Loop digunakan di state DOWNSCALE untuk menyalin gambar hasil downscale ke output image.

f. Function

- Function seperti calculate\_downscaled\_size didefinisikan di dalam RGBPackage untuk menghitung dimensi berdasarkan faktor downscale.

g. Finite State Machine



- State machine dalam process utama dapat dianggap sebagai bentuk FSM, di mana terdapat setiap state (IDLE, PROCESS\_INPUT, DOWNSCALE) yang mengontrol alur dari komponen ImageDownscaler.

## CHAPTER 3

### TESTING AND ANALYSIS

#### 3.1 TESTING

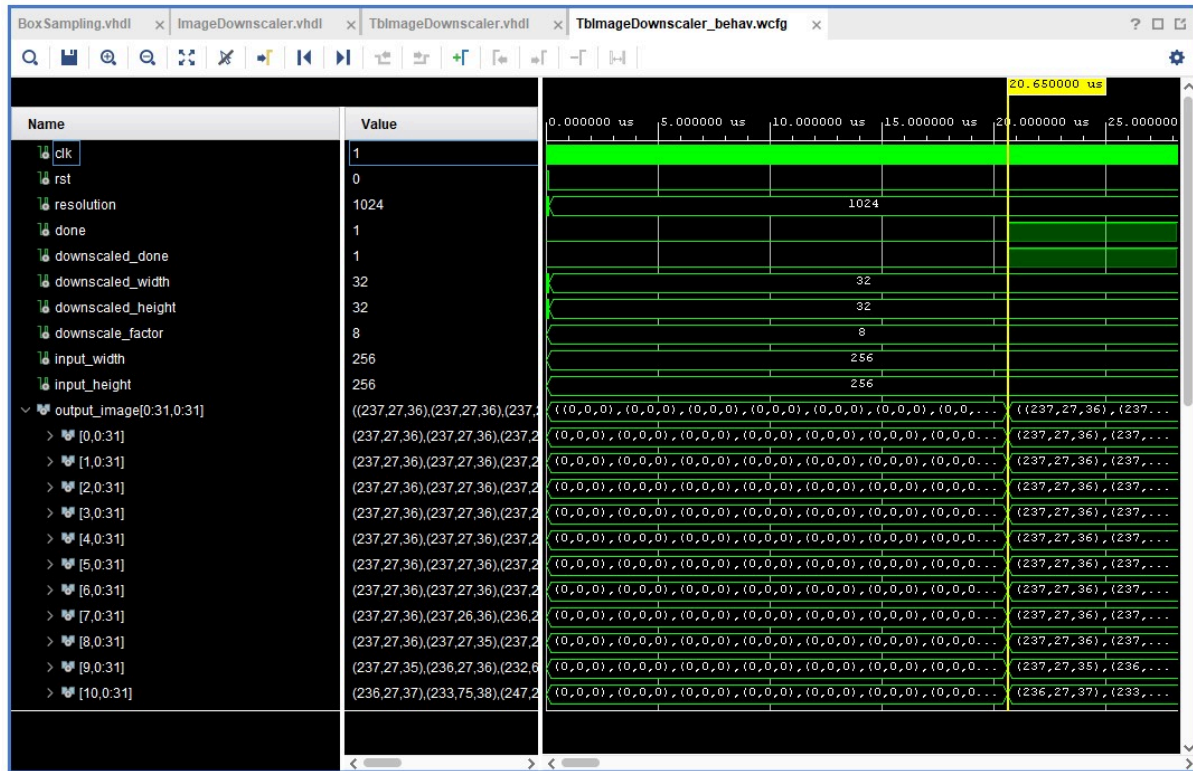


Fig.1 Test Wave

Proses testing dilakukan dengan menggunakan gambar input berdimensi 256x256 pixel yang memiliki data RGB untuk setiap pixel. Faktor downscale yang diterapkan pada gambar adalah 8, yang berarti gambar tersebut akan diperkecil menjadi 32x32 pixel setelah melalui proses downscaling. Pada tahap ini, sistem BoxSampling berfungsi untuk mengurangi jumlah pixel dengan cara menghitung rata-rata nilai warna dari blok pixel yang lebih besar. Proses ini dilakukan dengan mengiterasi setiap blok berukuran 8x8 pixel dan menghitung rata-rata nilai warna RGB untuk masing-masing blok, yang kemudian akan di-copy ke output image.

Komponen ImageDownscaler berfungsi untuk mengatur keseluruhan proses downscaling, dengan menggunakan finite state machine untuk mengelola tahapan-tahapan dalam proses tersebut. Dimulai dari keadaan IDLE untuk menyiapkan gambar input, kemudian memasuki keadaan PROCESS\_INPUT saat gambar sedang diproses. Setelah itu,

sistem masuk ke keadaan DOWNSCALE, di mana BoxSampling akan dipanggil untuk melakukan proses downscale resolusi gambar. Setelah proses downscaling selesai, hasil gambar yang telah diperkecil di-copy ke output image, dan status done serta downscaled\_done diatur menjadi '1' untuk menandakan bahwa proses selesai.

### 3.2 RESULT



Fig.2 Test Image



Fig.3 Downscaled Test Image

Setelah proses downscaling selesai, gambar output yang dihasilkan memiliki resolusi 32x32 pixel, yang dihitung berdasarkan faktor downscale 8. Hasil gambar output tersebut adalah hasil perhitungan rata-rata warna dari blok pixel input yang berukuran 8x8 pixel. Jadi jika blok pixel input memiliki nilai RGB yang sama, warna dominan dari blok tersebut akan identical dalam pixel output. Secara visual, gambar output tetap mempertahankan pola dominan warna yang ada pada gambar input, meskipun ukurannya jauh lebih kecil. Namun, beberapa detail pada gambar input mungkin hilang dalam proses ini, karena metode Box Sampling mengurangi *image sharpness* dengan cara menghitung rata-rata nilai warna dalam blok pixel besar. Namun, gambar output yang lebih kecil tetap mempertahankan *quality* warna dari gambar input, meskipun dengan mengorbankan *sharpness* dan detailnya.

### 3.3 ANALYSIS

Rangkaian berhasil melakukan Downscaling menggunakan algoritma Box Sampling dengan baik. Namun, ada beberapa kendala saat program di run, seperti “freezing” saat program mencoba melakukan downscale pada JPG yang berukuran besar, ini dapat dikarenakan semakin besarnya ukuran JPG maka semakin banyaknya pixel yang harus di “downscale” oleh algoritma box sampling. bukan berarti rangkaian ini tidak bisa melakukan downscale JPG berukuran besar, tetapi downscaling pada JPG yang berukuran besar hanya akan memakan waktu yang lebih banyak.

Selain permasalahan “freezing”, proyek Image Downscaler Using Box Sampling Algorithm telah berhasil diselesaikan dengan baik. proyek ini telah memenuhi tujuan desain dan implementasinya, yaitu yaitu menggunakan algoritma box sampling untuk menurunkan skala dari gambar dan menyediakan solusi transmisi gambar dengan mengurangi ukuran gambar tanpa mengorbankan kualitas visual.

Secara keseluruhan, hampir semua modul telah diimplementasikan dengan baik dalam proyek ini. Proyek Image Downscaler Using Box Sampling Algorithm memiliki potensi untuk diterapkan dalam berbagai aplikasi yang membutuhkan fungsi pengecilan ukuran gambar tanpa mengubah kualitas visualnya.

## **CHAPTER 4**

### **CONCLUSION**

Proyek Image Downscaler Using Box Sampling Algorithm berhasil mencapai tujuannya dengan baik, yaitu membuat sistem untuk mengecilkan ukuran gambar menggunakan algoritma Box Sampling. Sistem ini mengelompokkan beberapa pixel menjadi satu blok, lalu menghitung rata-rata warna blok tersebut untuk menghasilkan gambar yang lebih kecil. Dengan pendekatan berbasis VHDL, proses downscaling dapat dilakukan lebih cepat dan efisien. Sistem ini juga dirancang untuk dapat diimplementasikan pada aplikasi seperti device IoT, transmisi data, dan platform dengan keterbatasan daya ataupun penyimpanan.

Proses pengujian berhasil menunjukkan bahwa sistem dapat menghasilkan gambar yang tetap mempertahankan warna utama dari gambar asli, walau pasti detailnya berkurang. Tantangan utama dalam menjalankannya adalah waktu pemrosesan yang lebih lama untuk gambar dengan ukuran besar, namun secara keseluruhan proyek ini telah bekerja sesuai dengan harapan. Proyek ini membuktikan bahwa algoritma Box Sampling efektif untuk mengecilkan ukuran gambar tanpa mengorbankan kualitas visual secara signifikan.

## REFERENCES

- [1] R. A. Jordhie, R. Fauzan, F. R. Permana, and N. Febriyanto, “Bicubic Interpolation Image Upscaling Hardware Accelerator on VHDL,” *GitHub*, Dec. 10, 2022. <https://github.com/Jordinia/Bicubic-Interpolation> (accessed Nov. 23, 2024).
- [2] V. Parthipan , “Image Down-Scaler Using the Box Filter Algorithm,” repository.rit.edu, Dec. 07, 2024. <https://repository.rit.edu/cgi/viewcontent.cgi?article=10864&context=theses> (accessed Nov. 23, 2024).
- [3] Chaubey, Aashish , “Downsampling and Upsampling of Images — Demystifying the Theory,” medium.com, Jan. 14, 2020. <https://medium.com/analytics-vidhya/downsampling-and-upsampling-of-images-demystifying-the-theory-4ca7e21db24a> (accessed Nov. 23, 2024).
- [4] Adam Finkelstein , “Sampling, Resampling, and Warping,” princeton.edu, 2018. <https://www.cs.princeton.edu/courses/archive/spring18/cos426/lectures/03-warp.pdf> (accessed Nov. 23, 2024).
- [5] Render Pool , “Image Scaling: Introduction to Scaling, Resizing & Resampling,” renderpool.net, Aug. 24, 2020. <https://renderpool.net/blog/image-scaling/> (accessed Nov. 23, 2024).

## **APPENDICES**

### **Appendix A: Project Schematic**





## Dokumentasi diskusi 2 :

