

Oil Palm Plantation Image Analysis Using Quantum Image processing Approach

Desi Amirullah¹, Lipantri Mashur Gultom²

^{1,3} Program Studi Teknik Informatika, Jurusan Teknik Informatika,
Politeknik Negeri Bengkalis

Jl. Bathin Alam, Sungai Alam, Bengkalis, Riau

andes@polbeng.ac.id, lipantri@polbeng.ac.id

Abstrak

Penelitian ini berfokus pada pengolahan citra/gambar (image processing) dalam pendeteksian tepi pohon sawit dari beberapa kumpulan citra/gambar dengan beberapa variasi resolusi pixel dengan pendekatan quantum image processing dengan tujuan menghasilkan analisis yang akurat sehingga dapat digunakan untuk riset berkelanjutan kedepannya. Quantum Hadamard Edge Detection (QHED) digunakan untuk pendeteksian tepi suatu gambar dimana jumlah qubit yang digunakan ternyata sangat mempengaruhi waktu pemrosesan CPU. Jumlah qubit yang digunakan pada penelitian ini yaitu 2, 4, 6, 8, 10 dan 12 qubit, sedangkan jumlah qubit lebih dari 12 tidak dapat diuji karena keterbatasan memori RAM dari perangkat yang ada dalam penelitian ini. Hasil akhir dari penelitian membuktikan bahwa QHED dapat mendeteksi tepi suatu gambar dimana waktu pemrosesan yang paling cepat pada penggunaan 6 qubit sedangkan hasil proses pendeteksian tepi yang terbaik terletak pada penggunaan 2 qubit. Selain itu penelitian ini juga membandingkan QHED dengan Canny dan Sobel dimana komparasi waktu pemrosesan Canny dan Sobel masih lebih cepat namun kualitas hasil pemrosesannya masih lebih bagus QHED.

Kata kunci— Quantum, Hadamard, Edge Detection

Abstract

This study focuses on image processing (image processing) in detecting the edges of palm trees from several collections of images/images with several variations of pixel resolution with a quantum image processing approach to produce an accurate analysis so that it can be used for future sustainable research. Quantum Hadamard Edge Detection (QHED) is used to detect the edges of an image where the number of qubits used affects CPU processing time. The number of qubits used in this study was 2, 4, 6, 8, 10, and 12 qubits, while the number of qubits more than 12 could not be tested due to the limited RAM of the devices in this study. The final result of the research proves that QHED can detect the edges of an image where the fastest processing time is on the use of 6 qubits while the best edge detection process results are in the use of 2 qubits. In addition, this study also compares QHED with Canny and Sobel where the comparison between Canny and Sobel's processing time is still faster but the quality of the processing results is still better than QHED.

Keywords—Quantum, Hadamard, Edge Detection

1. PENDAHULUAN

Indonesia merupakan negara terbesar di dunia dalam ekspor hasil perkebunan sawit (<https://www.liputan6.com>, <https://www.wartasawit.com>, <https://dunia.tempo.co>). Hal ini tidak dapat dimungkiri karena pada kenyataannya beberapa wilayah di Indonesia terdapat lahan perkebunan sawit yang sangat luas seperti pulau Sumatera dan Kalimantan. Potensi ini diikuti dengan banyaknya implementasi hasil riset pada bidang *image processing*. *Image processing* sangat penting posisinya karena merupakan salah satu metode yang dapat digunakan dalam *Knowledge Discovery in Database* (KDD) baik pada kumpulan data skala kecil maupun skala besar (*Big Data*). Melihat potensi ini seharusnya Indonesia memiliki satu *platform big data* yang baik dan berkualitas dalam pengolahan perkebunan sawit sesuai dengan kerangka industri 4.0 dan juga karena adanya persaingan yang ketat dengan beberapa negara yang memiliki potensi yang sama.

Berbicara tentang beberapa kajian riset yang berhubungan dengan *image processing* khususnya pada topik perkebunan sawit masih berada pada ruang lingkup metode-metode klasik seperti pada jurnal terdahulu (Eja, 2021), (Irsanti, 2019), (Prasvita, 2021), (Rosalina, 2019), (Srinarta, 2022) yang berfokus pada metode klasifikasi pohon sawit dengan menguraikan kelebihan dan kekurangan hasil risetnya masing-masing. Namun dengan perkembangan teknologi komputasi sekarang ini yang sudah mulai menerapkan komputasi kuantum, peneliti menfokuskan pada penerapan *image processing* dengan pendekatan komputasi kuantum yaitu *quantum image processing* (Yan, 2020), (Yan, 2017). Pada prinsipnya penelitian ini dimulai dari pengolahan citra/gambar (*image processing*) yang sangat sederhana yaitu klasifikasi pohon sawit dari beberapa kumpulan citra/gambar dengan beberapa variasi resolusi *pixel* dengan pendekatan *quantum image processing* dengan tujuan menghasilkan analisis yang akurat sehingga dapat digunakan untuk riset berkelanjutan kedepannya. Dalam menghasilkan produk riset dalam satu platform yang besar sangat dibutuhkan waktu dan biaya yang sangat banyak.

Quantum Image processing (QIP) merupakan metode lain dalam pengolahan citra yang menggunakan pendekatan komputasi kuantum dalam representasi dan penyimpanan data citra (Yan, 2020), (Yan, 2017), (Pramanik, 2021). Dalam komputasi kuantum semua nilai keadaan disimpan kedalam bentuk qubit dimana biasanya hasil akhir pemrosesan bergantung pada rangkaian gerbang logika kuantum (Gan, 2022) (Anand, 2022). *Quantum Hadamard Edge Detection* (QHED) merupakan salah satu algoritma pendeteksian tepi dengan pendekatan komputasi kuantum menggunakan transformasi *hadamard*. Pada QHED memiliki skema pengkodean gambar yang efisien dalam menghemat ruang memori dengan menggunakan pengkodean amplitudo terhadap penurunan eksponensial dalam jumlah *qubit* yang digunakan yaitu $n = \log_2 N$ (Geng, 2022). Namun kompleksitas waktu dalam persiapan awal pada pengkodean gambar sangatlah tinggi. Oleh karena itu, penelitian ini difokuskan pada penganalisisan bagaimana pengaruh jumlah *qubit* yang digunakan pada QHED terhadap waktu dan ruang memori yang digunakan (Ruan, 2021)(Su, 2020).

QHED menggunakan gerbang *Hadamard* (H) sebagai operasi transformasi *qubit* (Yuan, 2019).

$$|0\rangle = \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \quad |1\rangle = \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \quad (1)$$

Kemudian N -*pixel* gambar yang diproses akan diberi nomor dengan menggunakan *binari bit-string* $|b_{n-1}b_{n-2}b_{n-3} \dots b_1b_0\rangle$ dimana $b_i \in \{0,1\}$. Untuk dua *pixel* yang berdekatan, maka dapat dituliskan dalam bentuk pasangan *bit-string* $|b_{n-1}b_{n-2} \dots b_10\rangle$ dan $|b_{n-1}b_{n-2} \dots b_11\rangle$ dimana hanya *least significant bit* (LSB) yang berbeda dari keduanya. Masing – masing nilai intensitas *pixel* yang sesuai (dinormalisasi) dapat ditulis sebagai $c_{b_{n-1}b_{n-2} \dots b_10}$ dan $c_{b_{n-1}b_{n-2} \dots b_11}$. Kemudian penulisan nilai *pixel* tersebut dapat disederhanakan kedalam representasi desimal yaitu c_i dan c_{i+1} . Penerapan gerbang H ke LSB pada ruang register kuantum menjadi hasil unitari sebagai berikut.

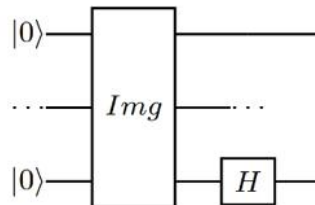
$$I_{2^{n-1}} \otimes H_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \quad (2)$$

Dimana $I_{2^{n-1}}$ adalah *matrix* $2^{n-1} \times 2^{n-1}$. Hasil unitari yang berisikan nilai *pixel* gambar kedalam register kuantum dikodekan menggunakan *Quantum Probability Image Encoding* (QPIE).

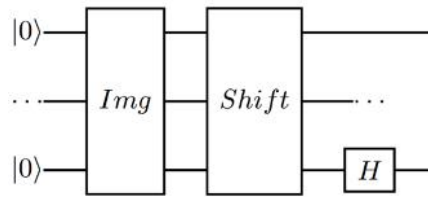
$$|I\rangle = \sum_{i=0}^{N-1} c_i |i\rangle \quad (3)$$

$$(I_{2^{n-1}} \otimes H_0) \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{bmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{bmatrix} c_0 + c_1 \\ c_0 - c_1 \\ c_2 + c_3 \\ c_2 - c_3 \\ \vdots \\ c_{N-2} + c_{N-1} \\ c_{N-2} - c_{N-1} \end{bmatrix} \quad (4)$$

Berikut gambar 1 *circuit* untuk mengekstrak semua informasi tepi dari gambar (Cavalieri, 2020).

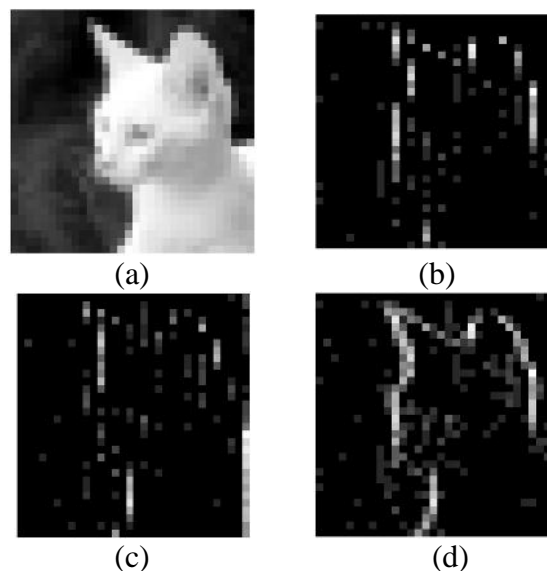


Gambar 1. Gerbang logika untuk mendapatkan nilai $(c_0 - c_1), (c_2 - c_3), \dots$



Gambar 2. Gerbang logika untuk mendapatkan nilai $(c_1 - c_2)$, $(c_3 - c_4)$, ...

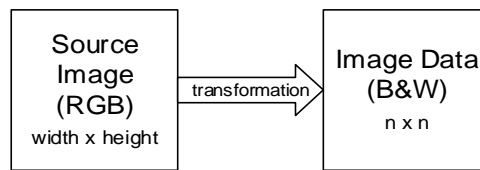
Berdasarkan hasil *matrix* persamaan (4) dapat dihasilkan deteksi tepi secara horizontal antara pasangan *pixel* bilangan genap : 0 & 1, 2 & 3 , dst serta deteksi tepi secara horizontal antara pasangan *pixel* bilangan ganjil : 1 & 2, 2 & 3, dst. Dalam memperoleh nilai permutasi amplitudo pada register kuantum dengan mengkonversi vektor amplitudo $(c_0, c_1, c_2, \dots, c_{N-1})^T$ menjadi $(c_1, c_2, c_3, \dots, c_{N-1})^T$. Setelah itu ditransformasi dengan gerbang H dan dilakukan pengukuran register kuantum pada LSB sampai menjadi |1>. Secara umum dapat dilihat contoh pemrosesan gambar 3 (Cavalieri, 2020).



Gambar 3. Contoh (a) sumber gambar, (b) hasil gambar pemrosesan dari fig. 1, (c) hasil gambar pemrosesan dari fig. 2 dan (d) penggabungan fig. 3, fig. 4 dan fig. 5

2. METODE PENELITIAN

Fokus pada pengimplementasian seluruh tahapan yang sudah diuraikan pada bagian sebelumnya. *Dataset* gambar yang diuji menggunakan gambar perkebunan sawit yang diperoleh dari *drone* merek DJI phantom 4. Kode program dibangun dengan menggunakan *Qiskit* dan disimulasikan dengan *Qiskit backend statevector simulator*. Perangkat keras yang digunakan yaitu Dell inspiron 3881 dengan RAM *memory* 16 GB (DDR4 SDRAM), *Processor* Intel(R) Core(TM) i7-10700F CPU @ 2.90 GHz (8 Core) dan sistem operasi Windows 10 *Home Single Language* 64 Bit. Pertama sekali dilakukan tahapan pre-pemrosesan data gambar yang akan diproses kedalam QHED (Pramanik, 2021).



Gambar 4. Pre-pemrosesan data gambar

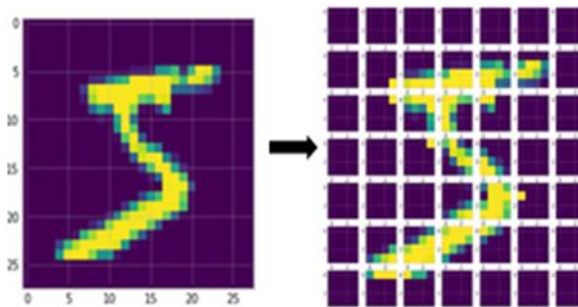


(a)

(b)

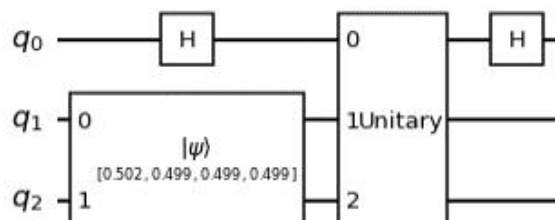
Gambar 5. Contoh pre-pemrosesan data gambar : (a) gambar RGB dengan 481 x 321 *pixel* dan (b) gambar B&W image dengan 321 x 321 *pixel*

Kemudian dilakukan pemilihan pasangan jumlah *qubit* (*N-qubit*) dan ukuran *chunk* 2^{cp} untuk membagi $n \times n$ *pixel* gambar menjadi $2^{cp} \times 2^{cp}$ (Anand, 2022).

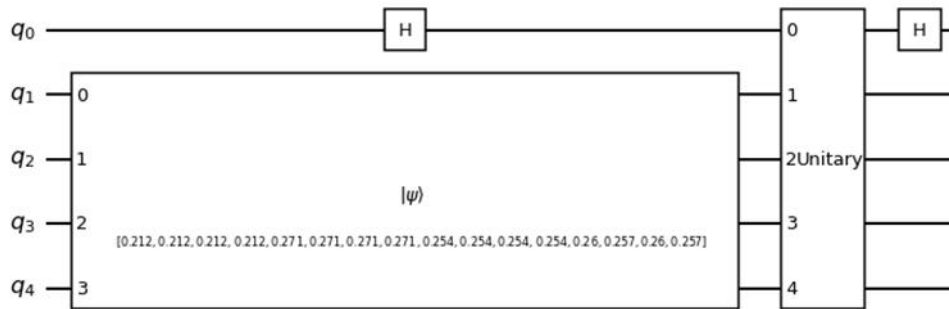


Gambar 6. Pembagian $n \times n$ gambar kedalam $2^{cp} \times 2^{cp}$

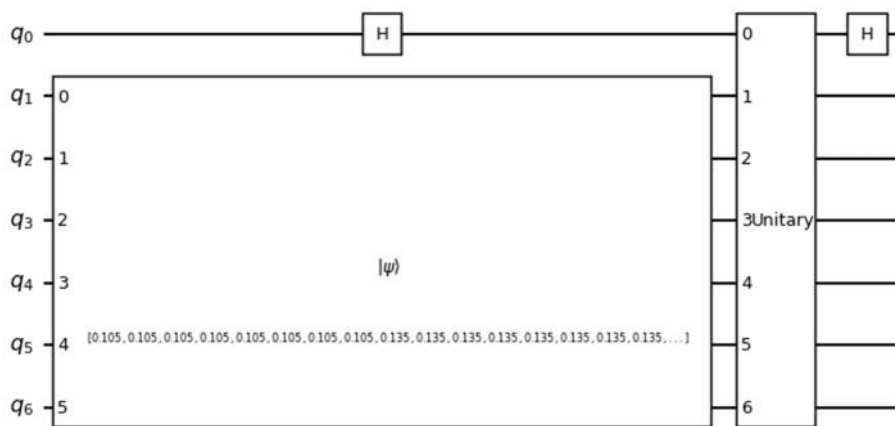
Jumlah *qubit* di sini digunakan untuk merancang quantum *circuit* dari gambar 7. Misalkan dipilih pasangan (2, 1) *N-qubit* = 2 dan $cp = 1$ maka ukuran *chunk* yang sesuai yaitu $2^{cp} = 2^1 = 2$, dan seterusnya dimana *N-qubit* bertambah 2 dan cp bertambah 1 menjadi (4,2), (6,3), (8, 4) dan seterusnya.



Gambar 7. Contoh rangkaian QHED dengan 2 *qubit*

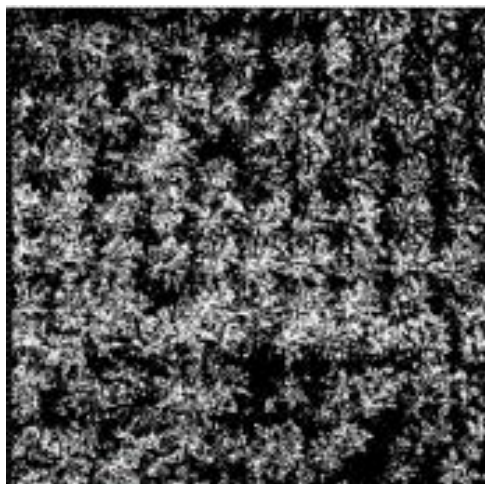


Gambar 8. Contoh rangkaian QHED dengan 4 qubit



Gambar 9. Contoh rangkaian QHED dengan 6 qubit

Dengan contoh hasil akhir dari proses QHED dapat dilihat pada gambar 10 berikut ini.

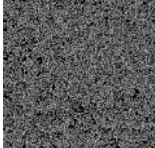
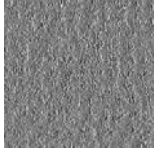
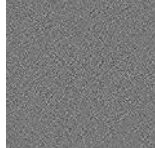
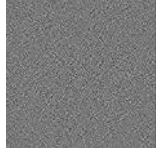
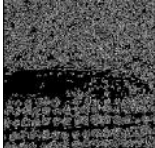
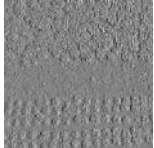
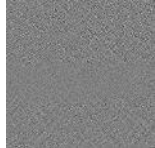
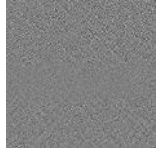
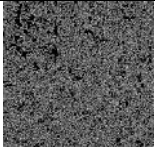
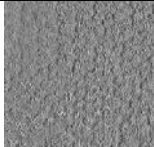
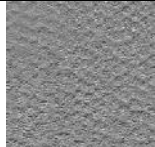
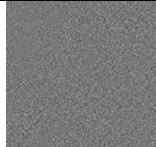
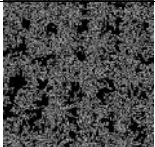
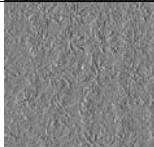
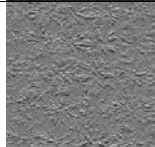
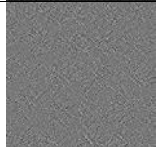

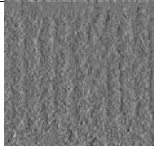
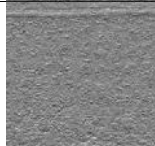
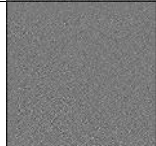


Gambar 10. Contoh hasil dari QHED

3. HASIL DAN PEMBAHASAN

Secara keseluruhan dari pengujian beberapa gambar tersebut dapat dilihat hasil segmentasi terbanyak dan terbaik terletak pada proses deteksi tepi gambar jika menggunakan 2 *qubit*. Sementara itu hasil yang terburuk jika menggunakan 12 *qubit*. Rangkuman hasil pengujian dengan beberapa dataset gambar yang digunakan dapat dilihat pada tabel 3 pada lampiran. Pada penelitian ini juga dilakukan pengujian dengan 14 *qubit* namun dengan spesifikasi memori (RAM 16 GB) tidak mampu mengalokasikan semua data yang dibutuhkan untuk hasil pemrosesan. Disini juga ditampilkan hasil deteksi tepi dengan metode yang tidak menggunakan pendekatan komputasi kuantum yaitu metode canny (Canny, 1986) dan sobel (Irwin, 1968) dapat dilihat tabel 1 berikut ini.

Tabel 1. Hasil deteksi tepi dengan metode canny dan sobel

No.	Canny	Sobel X	Sobel Y	Sobel XY
1				
2				
3				
4				
5				

Perbandingan data pada tabel 1 menunjukkan bahwa hasil dengan metode canny lebih mendekati dengan hasil QHED daripada dengan metode sobel. Berikutnya hasil pengujian dari pengaruh jumlah *qubit* terhadap waktu pemrosesan CPU dapat dilihat pada tabel 2 berikut ini.

Tabel 2. Pengaruh jumlah *qubit* terhadap waktu pemrosesan CPU

No.	QHED dengan jumlah <i>qubit</i>						Canny (detik)	Sobel (detik)
	2 (detik)	4 (detik)	6 (detik)	8 (detik)	10 (detik)	12 (detik)		
1	1193.88	156.85	49.87	71.56	508.38	7093.00	3.43	10.54
2	1163.39	158.28	50.04	71.93	504.19	7085.41	3.41	10.49
3	1163.97	163.21	49.86	74.27	505.60	7060.95	3.53	10.52
4	1199.65	159.29	53.03	73.02	508.01	7134.62	3.47	10.56
5	1213.50	150.59	49.46	71.64	505.57	7065.30	3.43	10.49

Secara keseluruhan waktu pemrosesan QHED yang paling cepat dengan penggunaan 6 *qubit* dan yang paling lama dengan penggunaan 12 *qubit*. Jika diperhatikan secara keseluruhan ternyata semakin banyak jumlah *qubit* yang digunakan maka semakin lama juga waktu pemrosesannya namun pada tabel 2 ditemukan ada sesuatu yang berbeda yaitu waktu pemrosesan dengan menggunakan 2 *qubit* relatif lebih lama dibandingkan dengan jumlah *qubit* sebanyak 4, 6, 8 dan 10. Hal ini bisa terjadi demikian karena pada tahapan QHED terdapat proses pembagian satu gambar menjadi beberapa kelompok gambar yang lebih kecil dengan pemotongan gambar menjadi $2^{cp} \times 2^{cp}$ seperti uraian pada gambar 7. Selain itu, metode canny dan sobel masih lebih cepat dalam waktu pemrosesan dibandingkan dengan QHED.

4. KESIMPULAN

Hasil penelitian ini membuktikan bahwa secara teoritis QHED dapat melakukan proses deteksi tepi pada gambar namun dari pengujian yang dilakukan ternyata penggunaan jumlah *qubit* sangat berpengaruh pada waktu pemrosesan CPU dimana waktu pemrosesan tercepat terletak pada penggunaan jumlah *qubit* sebanyak 6 *qubit* bukan terletak pada jumlah *qubit* yang paling sedikit yaitu 2 *qubit*. Sedangkan hasil deteksi tepi yang terbaik pada penggunaan jumlah 2 *qubit*.

5. SARAN

Pengujian dengan jumlah *qubit* lebih besar dari 12 *qubit* dengan memori (RAM) 16 GB tidak sanggup dilakukan dengan kata lain jika ingin melakukan pemrosesan dengan jumlah *qubit* yang lebih besar dari seluruh tahapan dan parameter yang ada pada penelitian ini harus membutuhkan kapasitas memori (RAM) yang lebih besar juga.

UCAPAN TERIMA KASIH

Penelitian ini didanai oleh Politeknik Negeri Bengkalis dari dana hibah penelitian internal (PNBP 2022) dalam skema penelitian terapan.



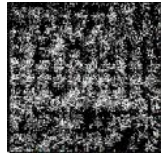
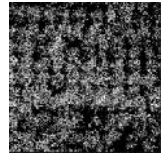
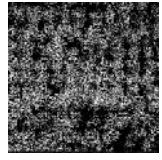
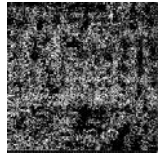
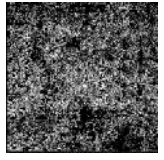
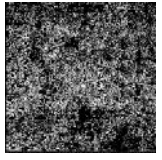

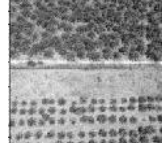
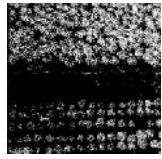
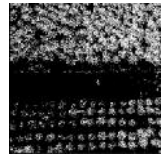
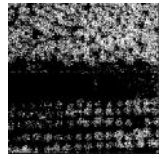
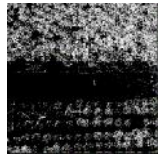
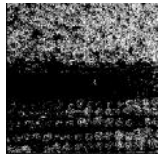
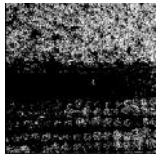

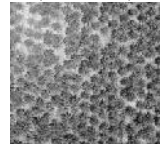
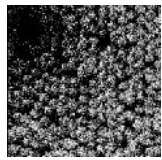
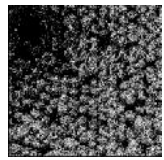
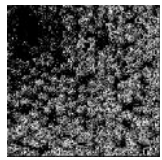
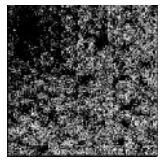
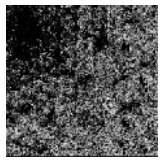
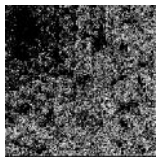


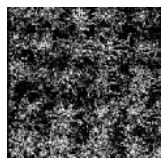
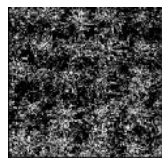
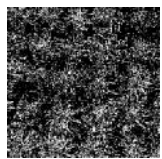
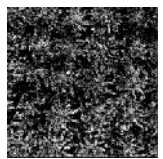
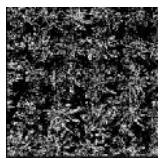
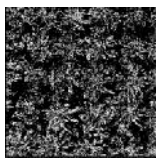

DAFTAR PUSTAKA

- Anand, A., Lyu, M., Baweja, P. S., & Patil, V. (2022). Quantum Image processing. arXiv preprint arXiv:2203.01831.
- Cavalieri, G., & Maio, D. (2020). A quantum edge detection algorithm. arXiv preprint arXiv:2012.11036.

- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.
- Eja, I. R., & Basyid, M. (2021). Kajian Perhitungan Pohon Kelapa Sawit Menggunakan Foto Udara Dan Citra Satelit Pleiades Dengan Metode Obia (Object Based Image Analysis)(Studi Kasus Pulau Sebatik). *FTSP*, 518-524.
- F. G. Irwin and others. (1968). "An isotropic 3x3 image gradient operator," *Present. Stanford AI Proj.*, vol. 2014, no. 02.
- Gan, W. S. (2022). Quantum image processing. In *Quantum Acoustical Imaging* (pp. 83-86). Springer, Singapore.
- Geng, A., Moghiseh, A., Redenbach, C., & Schladitz, K. (2022). A hybrid quantum image edge detector for the NISQ era. *arXiv preprint arXiv:2203.12072*.
- Irsanti, D., Sasmito, B., & Bashit, N. (2019). Kajian Pengaruh Penajaman Citra Untuk Penghitungan Jumlah Pohon Kelapa Sawit Secara Otomatis Menggunakan Foto Udara (Studi Kasus: KHG Bentayan Sumatra Selatan). *Jurnal Geodesi Undip*, 8(1), 428-434.
- Pramanik, S., Chandra, M. G., Sridhar, C. V., Kulkarni, A., Sahoo, P., DV, V. C., ... & Nambiar, M. (2021). A quantum-classical hybrid method for image classification and segmentation. *arXiv preprint arXiv:2109.14431*.
- Prasvita, D. S., Santoni, M. M., Wirawan, R., & Trihastuti, N. (2021). Klasifikasi Pohon Kelapa Sawit pada Data Fusi Citra Lidar dan Foto Udara Menggunakan Convolutional Neural Network. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 6(2), 406-415.
- Elin, R., & Agustin, S. (2019). Klasifikasi Umur Lahan Perkebunan Kelapa Sawit Pada Citra Foto Udara Berdasarkan Tekstur Menggunakan Metode Naïve Bayes. *Indexia: Informatics and Computational Intelligent Journal*, 1(1), 6-13.
- Srinarta, K. (2022). Analisis Perhitungan Jumlah Pohon Kelapa Sawit Berdasarkan Algoritma Canopy Height Model (Chm) Dan Local Maxima (Lm). *Jurnal Geodesi UNDIP*, 11(1), 51-60.
- Su, J., Guo, X., Liu, C., & Li, L. (2020). A new trend of quantum image representations. *IEEE Access*, 8, 214520-214537.
- Yan, F., Venegas-Andraca, S.E. (2020). Quantum Image Operations. In: *Quantum Image processing*. Springer, Singapore. https://doi.org/10.1007/978-981-32-9331-1_3
- Yan, F., Iliyasu, A. M., & Le, P. Q. (2017). Quantum image processing: a review of advances in its security technologies. *International Journal of Quantum Information*, 15(03), 1730001. *Int. J. Quantum Inf.*, vol. 15, no. 03, p. 1730001, 2017.
- Yuan, S., Venegas-Andraca, S. E., Wang, Y., Luo, Y., & Mao, X. (2019). Quantum image edge detection algorithm. *International Journal of Theoretical Physics*, 58(9), 2823-2833.
- <https://www.liputan6.com/bisnis/read/4575604/indonesia-jadi-eksportir-minyak-sawit-terbesar-di-dunia> (diakses tanggal 10 Mei 2022)
- <https://www.wartasawit.com/read/991/indonesia-kembali-jadi-eksportir-terbesar-minyak-sawit-dunia.html> (diakses tanggal 10 Mei 2022)
- <https://bisnis.tempo.co/read/1469857/ri-ekspor-373-juta-ton-minyak-sawit-pada-2020-terbesar-di-dunia> (diakses tanggal 10 Mei 2022)
- <https://www.ibm.com/quantum-computing/>
- <https://qiskit.org/>

LAMPIRAN

Tabel 3. Hasil QHED dengan jumlah *qubit*

No.	Gambar Asli (<i>w x h</i>)	Gambar Pre- pemrosesan (<i>n x n</i>)	Jumlah <i>qubit</i>					
			2	4	6	8	10	12
1	(682 x 384) 	(384 x 384) 						
2	(682 x 384) 	(384 x 384) 						
3	(682 x 384) 	(384 x 384) 						
4	(682 x 384) 	(384 x 384) 						
5	(682 x 384) 	(384 x 384) 