

Analisis Deteksi Tepi Citra Dengan *Quantum Hadamard Edge Detection (QHED)*

Analysis of Image Edge Detection Using Quantum Hadamard Edge Detection (QHED)

Lipantri Mashur Gultom¹, Desi Amirullah²

^{1,2}Jurusan Teknik Informatika, Politeknik Negeri Bengkalis

E-mail: ¹lipantri@polbeng.ac.id, ²andes@polbeng.ac.id

Abstrak

Fokus penelitian ini pada eksperimen *Quantum Hadamard Edge Detection (QHED)* untuk pendeteksian tepi suatu gambar dimana jumlah qubit yang digunakan ternyata sangat mempengaruhi waktu pemrosesan CPU. Penelitian ini menggunakan *benchmark dataset* gambar yaitu *contour detection and image segmentation* dari *Berkeley Computer Vision Group*. 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.

Kata kunci: Quantum, Hadamard, Edge Detection

Abstract

The focus of this research is on the Quantum Hadamard Edge Detection (QHED) experiment for the detection of the edges of an image where the number of qubits used actually affects CPU processing time. This study uses an image dataset benchmark, namely contour detection and image segmentation from Berkeley Computer Vision Group. The number of qubits used in this study were 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 memory 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.

Keywords: Quantum, Hadamard, Edge Detection

1. PENDAHULUAN

Quantum Image Processing (QIP) merupakan metode lain dalam pengolahan citra yang menggunakan pendekatan komputasi kuantum dalam representasi dan penyimpanan data citra [1][2][3]. Dalam komputasi kuantum semua nilai keadaan disimpan kedalam bentuk *qubit* dimana biasanya hasil akhir pemrosesan bergantung pada rangkaian gerbang logika kuantum [4][5]. *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 $n = \log_2 N$ [6]. 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 [7][8].

QHED menggunakan gerbang *Hadamard* (H) sebagai operasi transformasi *qubit* [9].

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

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} . Selanjutnya penerapan gerbang H ke LSB pada ruang register kuantum menjadi hasil unitari sebagai berikut.

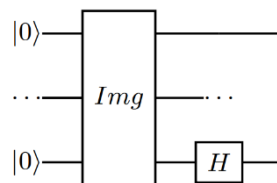
$$I_{2^{n-1}} \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0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

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)

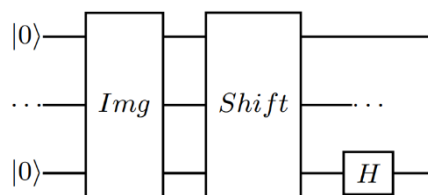
$$|Img\rangle = \sum_{i=0}^{N-1} c_i |i\rangle \quad (3)$$

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

Berikut gambar *circuit* untuk mengekstrak semua informasi tepi dari gambar [10].

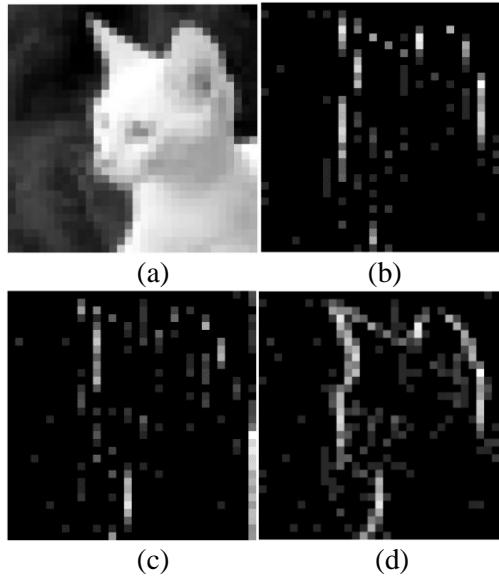


Gambar 1 Gerbang logika untuk mendapatkan nilai $(c_0 - c_1), (c_2 - c_3), \dots$



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

dari 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. Sehingga untuk 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 $|1\rangle$. Secara umum dapat dilihat contoh pemrosesan gambar berikut ini [10].



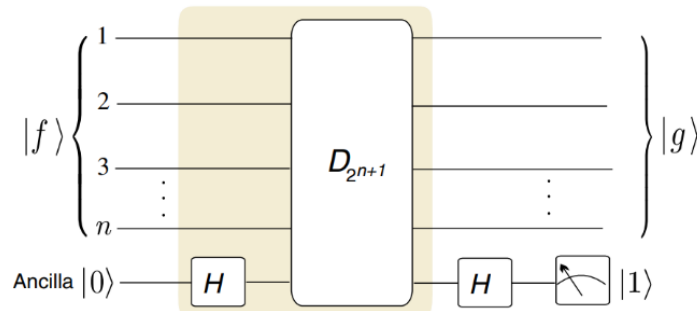
Gambar 3 Contoh (a) sumber gambar, (b) hasil gambar pemrosesan dari fig. 1, (c) hasil gambar pemrosesan dari fig. 2 dan (d) penggabungan fig. 3, fig. 4 dan fig. 5

Dari penelitian [11] dapat divariasikan dengan *qubit* tambahan ke register kuantum sehingga dapat memperluas algoritma QHED dalam melakukan komputasi pada pasangan *pixel* genap dan ganjil secara bersamaan. Misalnya dari tahapan sebelumnya inisialisasi $|Img\rangle = (c_0, c_1, c_2, \dots, c_{N-1})^T$. Selanjutnya gerbang H diterapkan pada *qubit* tambahan dengan inisialisasi $|0\rangle$. Pada akhirnya menghasilkan redudansi $(n+1)$ *qubit* pada gambar yang direpresentasikan dengan persamaan (5). Selanjutnya didefinisikan kesatuan permutasi amplitudo pada persamaan (6) untuk mengubah amplitudo menjadi suatu struktur yang akan memudahkan dalam perhitungan nilai gradien gambar untuk tahapan selanjutnya.

$$|Img\rangle \otimes \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} c_0 \\ c_0 \\ c_1 \\ c_1 \\ c_2 \\ c_2 \\ \vdots \\ c_{N-2} \\ c_{N-2} \\ c_{N-1} \\ c_{N-1} \end{bmatrix} \quad (5)$$

$$D_{2^{n+1}} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (6)$$

Persamaan (6) disebut juga dengan *decrement gate* (D) yang secara efisien dapat menguraikan hasil unitari kedalam satu kelompok gerbang rotasi baik tunggal maupun multi *controlled-X* pada register kuantum. Lebih jelasnya dapat dilihat ada gambar *circuit* berikut ini



Gambar 4 Rangkaian QHED dengan satu *qubit* tambahan (ancilla).

Dengan penerapan gerbang $D_{2^{n+1}}$, akan mentransformasikan $(c_0, c_0, c_1, c_1, c_2, c_2, \dots, c_{N-2}, c_{N-2}, c_{N-1}, c_{N-1})^T$ menjadi $(c_0, c_1, c_1, c_2, c_2, \dots, c_{N-2}, c_{N-1}, c_{N-1}, c_0)^T$. selanjutnya dengan penerapan gerbang H kepada *qubit* tambahan maka didapatkan gradien untuk pasangan *pixel* genap dan ganjil secara bersamaan.

$$(I_{2^n} \otimes H) \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_1 \\ c_2 \\ c_2 \\ c_3 \\ \vdots \\ c_{N-2} \\ c_{N-1} \\ c_{N-1} \\ c_0 \end{bmatrix} \rightarrow \begin{bmatrix} c_0 + c_1 \\ c_0 - c_1 \\ c_1 + c_2 \\ c_1 - c_2 \\ c_2 + c_3 \\ c_2 - c_3 \\ \vdots \\ c_{N-2} + c_{N-1} \\ c_{N-2} - c_{N-1} \\ c_{N-1} + c_0 \\ c_{N-1} - c_0 \end{bmatrix} \quad (7)$$

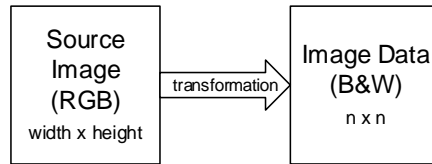
pengukuran dari kondisi ini terhadap *qubit* tambahan dalam keadaan $|1\rangle$ dapat menghasilkan nilai horizontal gradien $c_i - c_{i+1}$ untuk semua nilai kemungkinan dari pasangan *qubit* yang berdekatan. Sedangkan untuk mendapatkan nilai deteksi tepi gambar secara vertikal dapat menggunakan *matrix transpose* dari gambar kemudian mengikuti seluruh tahapan yang sudah diuraikan sebelumnya. Untuk tahapan akhir dalam menghasilkan merode deteksi tepi secara utuh dari proses ini maka hasil pengolahan secara horizontal dan vertikal harus dipadukan.

2. METODE PENELITIAN

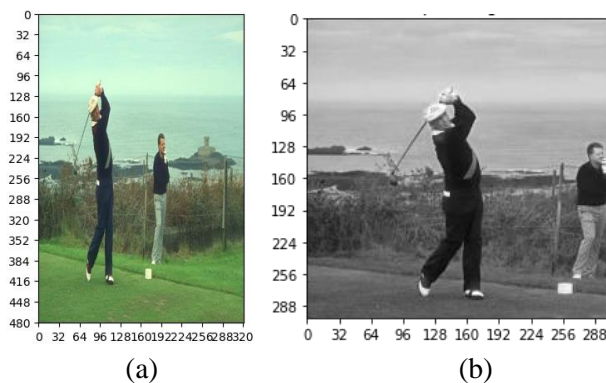
Bab ini fokus pada pengimplementasian seluruh tahapan yang sudah diuraikan pada bab sebelumnya. sedangkan *dataset* gambar yang diuji menggunakan *benchmark dataset* dari *contour detection and image segmentation resources* dari *berkeley computer vision group* [12]. Kode

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 [13].

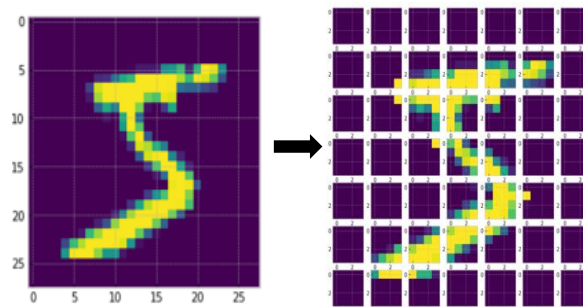


Gambar 5 Pre-pemrosesan data gambar



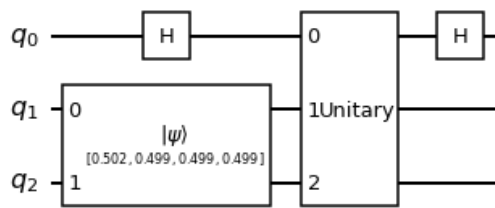
Gambar 6 Contoh pre-pemrosesan data gambar : (a) gambar RGB dengan 481 x 321 pixel dan (b) gambar B&W image dengan 321 x 321 pixel

kemudian dilakukan pemilihan pasangan jumlah *qubit* (N -*qubit*) dan ukuran *chunk* 2^{cp} untuk membagi $n \times n$ pixel gambar menjadi $2^{cp} \times 2^{cp}$ [5].

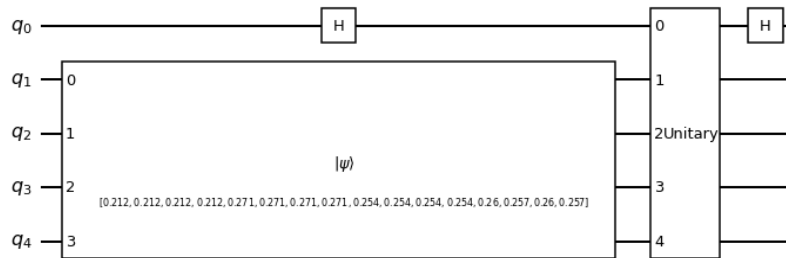


Gambar 7 Pembagian $n \times n$ gambar kedalam $2^{cp} \times 2^{cp}$

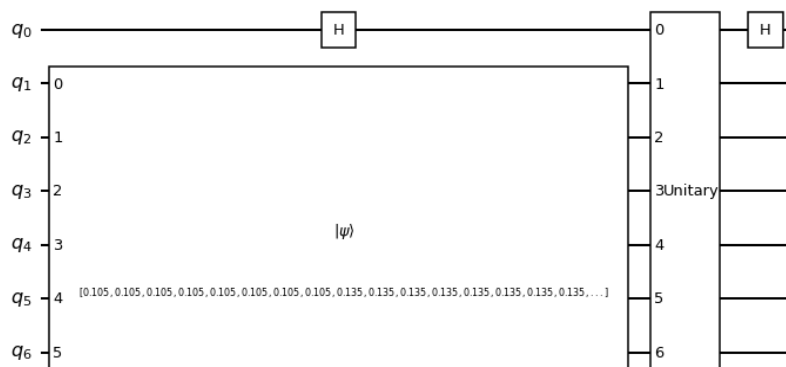
jumlah *qubit* disini digunakan untuk merancang quantum *circuit* dari fig.7. Jadi misalkan dipilih pasangan (2, 1) N -*qubit* = 2 dan $cp = 1$ maka ukuran *chunk* yang sesuai yaitu $2^{cp} = 2^1 = 2$. dan seterusnya dimana N -*qubit* bertambah 2 dan cp bertambah 1 menjadi (4,2), (6,3), (8, 4) dan seterusnya.



Gambar 8 Contoh rangkaian QHED dengan 2 qubit



Gambar 9 Contoh rangkaian QHED dengan 4 qubit



Gambar 10 Contoh rangkaian QHED dengan 6 qubit

dengan contoh hasil akhir dari proses QHED dapat dilihat pada gambar berikut



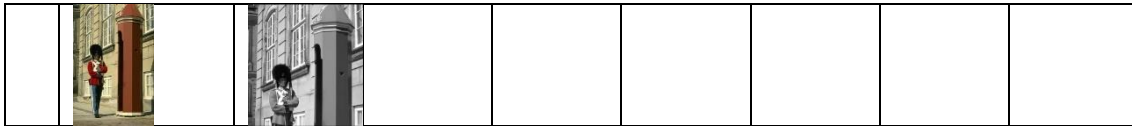
Gambar 11 Contoh hasil dari QHED

3. HASIL DAN PEMBAHASAN

Rangkuman hasil pengujian dengan beberapa dataset gambar yang digunakan dapat dilihat pada tabel 1 berikut ini.

Tabel 1 Hasil QHED dengan jumlah *qubit*

N o.	Gambar Asli (<i>w x h</i>)	Gambar Pre- pemrosesa n (<i>n x n</i>)	Jumlah <i>qubit</i>					
			2	4	6	8	10	12
1	(321 x 481) 	(321 x 321) 						
2	(481 x 321) 	(321 x 321) 						
3	(321 x 481) 	(321 x 321) 						
4	(481 x 321) 	(321 x 321) 						
5	(481 x 321) 	(321 x 321) 						
6	(481 x 321) 	(321 x 321) 						
7	(321 x 481) 	(321 x 321) 						
8	(321 x 481) 	(321 x 321) 						
9	(481 x 321) 	(321 x 321) 						
10	(321 x 481) 	(321 x 321) 						



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* bahkan pada pengujian gambar no. 3 tidak terlihat sama sekali hasil segmentasinya. 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 [14] dan sobel [15] dapat dilihat tabel 2 berikut ini.

Tabel 2 Hasil deteksi tepi dengan metode canny dan sobel

No.	Canny	Sobel X	Sobel Y	Sobel XY
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

dapat dibandingkan 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 3 berikut ini.

Tabel 3 Pengaruh jumlah *qubit* terhadap waktu pemrosesan CPU (dalam detik).

No.	QHED dengan Jumlah <i>qubit</i>						canny	sobel
	2	4	6	8	10	12		
1	698.869	93.043	31.874	48.078	301.377	3633.855	2.649	7.955
2	666.437	91.956	31.818	49.541	302.319	3627.856	2.670	7.917
3	808.029	106.345	36.336	52.119	365.829	5211.040	2.878	8.718
4	793.461	106.059	36.137	52.227	362.643	5205.506	2.956	8.573
5	787.224	105.873	36.090	52.094	362.132	5208.052	2.896	8.575
6	809.801	106.476	36.114	52.243	365.277	5204.094	2.856	8.647
7	759.835	102.422	35.539	51.958	366.631	5219.901	2.838	8.519
8	803.510	106.217	36.246	52.145	365.894	5184.337	2.895	8.384
9	774.315	106.544	36.049	52.121	366.783	5214.482	2.850	8.615
10	797.606	106.517	35.982	52.031	364.099	5200.643	2.936	8.602

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 3 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 DAN SARAN

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*. Kemudian 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

- [1] F. Yan and S. E. Venegas-Andraca, "Quantum image processing," *Quantum Image Process.*, pp. 1–171, 2020, doi: 10.1007/978-981-32-9331-1.
- [2] F. Yan, A. M. Ilyasu, and P. Q. Le, "Quantum image processing: a review of advances in its security technologies," *Int. J. Quantum Inf.*, vol. 15, no. 03, p. 1730001, 2017.
- [3] S. Pramanik *et al.*, "Quantum image processing," *Phys. Rev. X*, vol. 7, no. 3, pp. 1–171, 2020, doi: 10.1103/PhysRevX.7.031041.
- [4] W. S. Gan, "Quantum Image Processing," in *Quantum Acoustical Imaging*, Singapore: Springer Singapore, 2022, pp. 83–86.
- [5] A. Anand, M. Lyu, P. S. Baweja, and V. Patil, "Quantum image processing," 2022, [Online]. Available: <http://arxiv.org/abs/2203.01831>.
- [6] A. Geng, A. Moghiseh, C. Redenbach, and K. Schladitz, "A hybrid quantum image edge detector for the NISQ era," *Quantum Mach. Intell.*, vol. 4, no. 2, pp. 1–19, 2022, doi: 10.1007/s42484-022-00071-3.
- [7] Y. Ruan, X. Xue, and Y. Shen, "Quantum Image Processing: Opportunities and Challenges," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/6671613.
- [8] J. Su, X. Guo, C. Liu, and L. Li, "A new trend of quantum image representations," *IEEE Access*, vol. 8, pp. 214520–214537, 2020.
- [9] S. Yuan, S. E. Venegas-Andraca, Y. Wang, Y. Luo, and X. Mao, "Quantum Image Edge Detection Algorithm," *Int. J. Theor. Phys.*, vol. 58, no. 9, pp. 2823–2833, 2019, doi: 10.1007/s10773-019-04166-9.
- [10] G. Cavalieri and D. Maio, "A Quantum Edge Detection Algorithm," 2020, [Online]. Available: <http://arxiv.org/abs/2012.11036>.
- [11] X. W. Yao *et al.*, "Quantum image processing and its application to edge detection: Theory and experiment," *Phys. Rev. X*, vol. 7, no. 3, 2017, doi: 10.1103/PhysRevX.7.031041.
- [12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011, doi: 10.1109/TPAMI.2010.161.
- [13] S. Pramanik *et al.*, "A Quantum-Classical Hybrid Method for Image Classification and Segmentation," 2021, [Online]. Available: <http://arxiv.org/abs/2109.14431>.
- [14] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.
- [15] F. G. Irwin and others, "An isotropic 3x3 image gradient operator," *Present. Stanford AI Proj.*, vol. 2014, no. 02, 1968.