Komputika: Jurnal Sistem Komputer
Volume 8, Nomor 2, Oktober 2019, hlm. 59 - 65
Terakreditasi Peringkat 3, SK No. 28/E/KPT/2019
DOI: 10.34010/komputika.v8i2.2194

ISSN: 2252-9039 (print)
ISSN: 2655-3198 (online)

Implementasi dan Uji Kinerja Algoritma Background Subtraction pada ESP32

Didit Andri Jatmiko [1*], Salita Ulitia Prini[2]

[1)]Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 – 116, Bandung, Indonesia 40132

[2)]Pusat Penelitian Elektronika dan Telekomunikasi, Lembaga Ilmu Pengetahuan Indonesia, Jl.Sangkuriang, Komplek LIPI Gedung 20 Lantai 4, Bandung, Indonesia
**email: didit@email.unikom.ac.id*

ABSTRAK *– Salah satu hal penting pada computer vision adalah ciri (feature) citra. Ciri digunakan sebagai dasar untuk mendeteksi objek, baik itu benda, manusia maupun hewan. Ciri citra yang biasa digunakan dalam penelitian antara lain tepian, sudut, bentuk maupun gradient histogram. Penelitian ini menjelaskan kinerja algoritma background subtraction pada unit pemroses berdaya rendah sebagai salah satu algoritma pada computer vision. Algoritma ini memiliki kompleksitas yang rendah dan dapat digunakan untuk mendeteksi objek sehingga berpotensi diterapkan pada kamera keamanan. Algoritma ini bekerja dengan melakukan pengurangan nilai piksel current frame dengan background model. Penelitian ini telah berhasil menerapkan algoritma dasar pengolahan citra, yaitu algoritma background subtraction pada modul ESP32. Pengujian menggunakan input citra yang memiliki dimensi 80x60 piksel dengan format warna 8bit grayscale. Ukuran frame citra 80 x 60 piksel dipilih sebagai citra uji karena keterbatasan memory DRAM EPS32 sebesar 328 KB (kilobyte). Implementasi pada modul ESP32 yang dilengkapi dengan mikroprosesor Xtensa 32-bit LX6 yang bekerja pada frekuensi 240MHz dapat memproses algoritma background subtraction 10000 kali dalam waktu ±2000ms menggunakan input citra uji tersebut.*

Kata Kunci *– Background Subtraction; ESP32; Image Processing; Microcontroller; Object Detection.*

Implementation and Performance Testing of Background Subtraction Algorithm on ESP32

ABSTRACT *– One important thing of computer vision is the feature. The feature is used for detecting objects like humans or animals. Features may be specific structures in the image such as edges, angles, shapes and gradient histograms. This paper describes the background subtraction algorithm and its performance on low-power processing units as one of the algorithms in computer vision. This algorithm has low complexity and can be used to detect objects so that they can potentially be applied to security cameras. This algorithm works by subtracted the current frame pixel value with the background model. This study has succeeded in applying background subtraction algorithm on ESP32 module as basic image processing algorithm. The test uses image input which has dimensions of 80x60 pixels with 8-bit grayscale color format. The image frame size of 80 x 60 pixels was chosen as a test because of the limited DRAM ESP32 memory of 328 KB (kilobytes). Implementation of the ESP32 module equipped with Xtensa 32-bit LX6 microprocessors running at 240MHz can process background subtraction algorithms 10000 times in ± 2000ms using the test image input.*

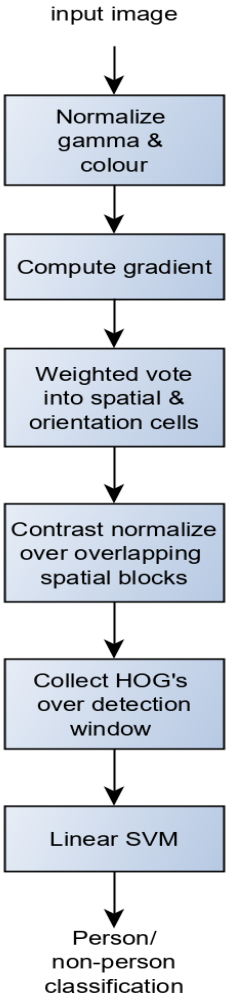
Keywords *– Background Subtraction; ESP32; Image Processing; Microcontroller; Object Detection.*

1. PENDAHULUAN

Perkembangan algoritma pengolahan citra saat ini mengalami peningkatan yang sangat signifikan tidak hanya digunakan untuk memperbaiki kualitas maupun mensegmentasi bagian citra tetapi juga telah mampu mendeteksi dan mengenali bagian citra, kemampuan mendeteksi dan mengenali bagian citra ini merupakan gabungan dari pengolahan citra dan kecerdasan buatan yang biasa disebut dengan

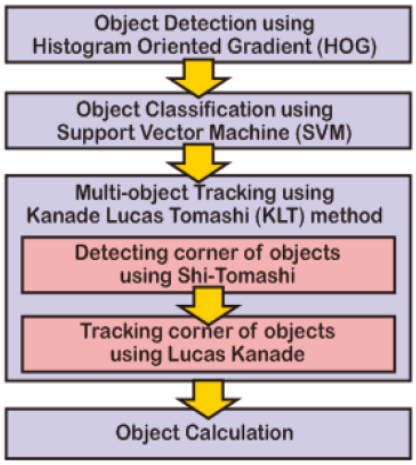
computer vision, implementasi *computer vision* pada kehidupan sehari-hari antara lain deteksi kematangan buah [1], aplikasi deteksi untuk robot [2], deteksi manusia [3] (blok sistem dapat dilihat pada Gambar 1), maupun deteksi dan pengenalan multi objek bergerak [4] (blok sistem dapat dilihat pada Gambar 2).

Salah satu hal penting dari *computer vision* adalah ciri (*feature*) citra, ciri digunakan sebagai dasar untuk mendeteksi objek baik itu benda, manusia maupun hewan, ciri citra yang biasa digunakan antara lain tepian [5], sudut [6], bentuk maupun *gradient histogram* [3], untuk mendapatkan ciri tersebut salah satu teknik yang digunakan adalah *background subtraction* [7], [8], [9], algoritma *background subtraction* adalah salah satu teknik pengolahan citra yang digunakan untuk menghasilkan bagian citra yang tidak terdapat pada pada *background* sehingga mampu menampilkan objek yang sebelumnya tidak terdapat pada *background* pada citra yang dihasilkan oleh kamera statis. Cara kerja algoritma ini adalah dengan melakukan pengurangan nilai piksel *current frame* dengan *background model.*

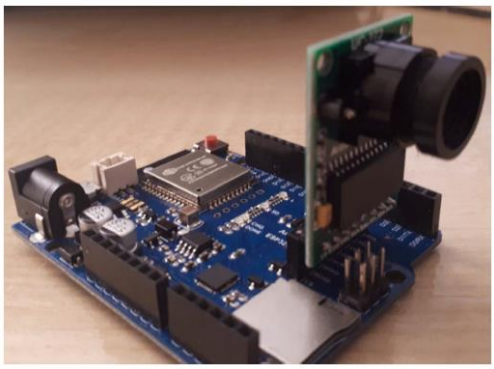


Gambar 1. Blok diagram sistem deteksi manusia menggunakan HOG [3]

ESP32 adalah *microcontroller* yang yang dilengkapi *Wi-Fi* 2.4 GHz dan teknologi *Bluetooth*, modul (*development kit*) ESP32 dapat dengan mudah dijumpai di pasaran dengan harga yang murah (±Rp100.000), keunggulan utama chip ESP32 ini antara lain : berdaya rendah, terintegrasi dengan *TCP/IP* dan *Bluetooth*, dokumentasi yang cukup baik, serta mendukung *compiler* C++ (*arduino* & *ESP-IDF*). Saat ini aplikasi yang menggunakan ESP32 sangat beragam diantaranya *Internet of Think* [10], *VR Hand Controller* [11], *Smart Surveillance System* [12] (dapat dilihat pada Gambar 3), web server berbiaya rendah untuk monitoring sistem *photovoltaic* [13], dan kriptografi [14], berdasarkan informasi ini ESP32 memiliki potensi sebagai unit pemroses algoritma *background subtraction*.



Gambar 2. Blok diagram sistem deteksi dan pengenalan multi objek [4]



Gambar 3. Arducam ESP32 UNO dilengkapi modul kamera [12]

Penelitian ini berusaha mengimplementasikan algoritma *background subtraction* di *microcontroller* ESP32 serta melakukan pengujian kinerja algoritma dengan beberapa variasi perulangan proses, dengan tujuan untuk mengetahui:

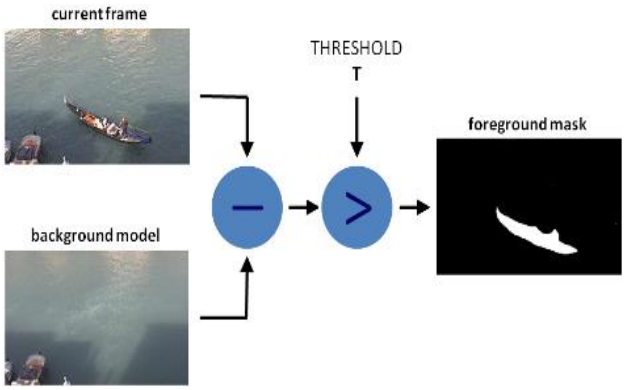
a. Apakah *microcontroller* ESP32 mampu

memproses algoritma *background subtraction*?
b. Dimensi citra yang dapat diproses
c. Jumlah *frame per second* (fps) yang diperoleh dari eksekusi algoritma *background subtraction* di ESP32

2. METODE DAN BAHAN

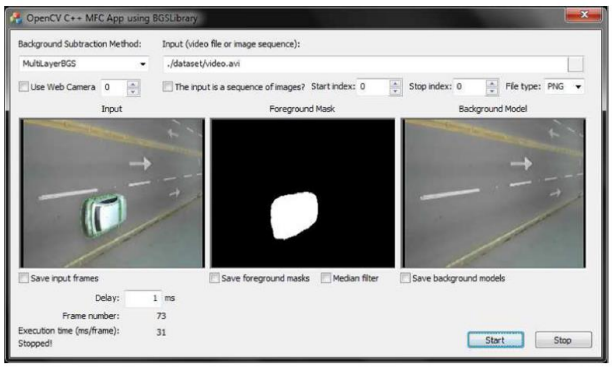
2.1. Algoritma Background Subtraction

Algoritma *background subtraction* adalah salah satu teknik pengolahan citra yang digunakan untuk menghasilkan bagian citra yang tidak terdapat pada pada *background* sehingga mampu menampilkan objek yang sebelumnya tidak terdapat pada *background* pada citra yang dihasilkan oleh kamera statis. Cara kerja algoritma ini adalah dengan melakukan pengurangan nilai piksel *current frame* dengan *background model*, ilustrasi cara kerja algoritma background subtraction dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi algoritma background subtraction [15]

Salah satu contoh penggunaan algoritma ini adalah untuk mendeteksi mobil yang melintasi jalan, contoh aplikasi yang menerapkan algoritma ini dapat dilihat pada Gambar 5.



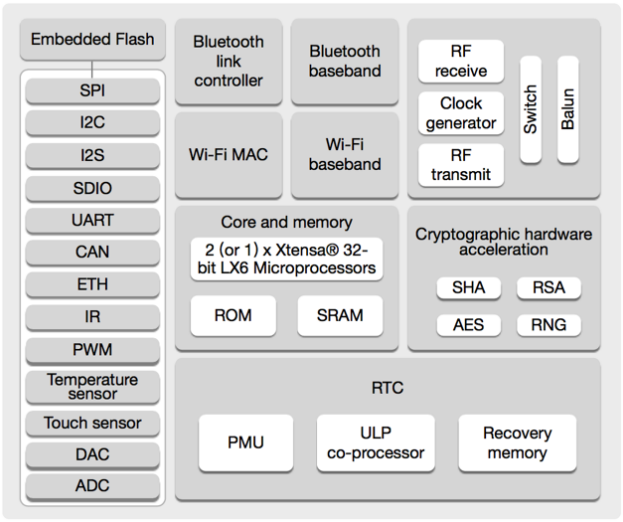
Gambar 5. Contoh software menggunakan algoritma background subtraction [8]

Pengurangan 2 buah citra (*current frame* dengan *background model)* dilakukan dengan menggunakan operasi aritmetika ke setiap nilai piksel [15], [16]. Nilai piksel output dihasilkan dari persamaan:

$$Q(i,j) = P_1(i,j) - P_2(i,j) \quad (1)$$

2.2. ESP32

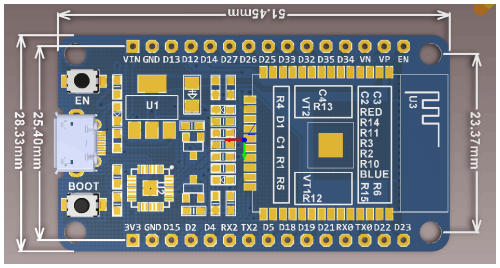
ESP32 adalah *microcontroller* yang yang dilengkapi *Wi-Fi* 2.4 GHz dan teknologi *Bluetooth* berdaya rendah [17]. Berdasarkan blok diagram pada Gambar 6, chip ESP32 terdiri atas *microprocessor* Xtensa 32bit, *cryptographic hardware acceleration*, DAC, ADC, UART, CAN, SPI, dan I2C.



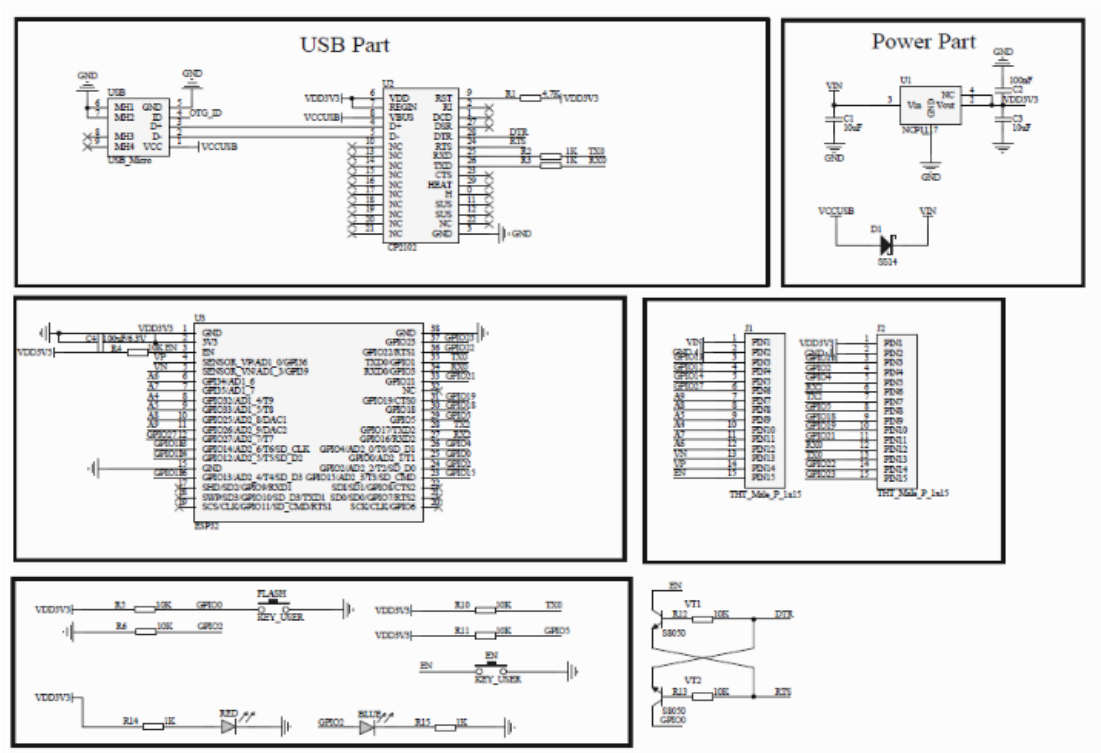
Gambar 6. Blok diagram chip ESP32 [18]

Tampilan modul DOIT ESP32 dapat dilihat pada Gambar 7, sedangkan dimensi fisik dan tata letak pin PCB modul ESP32 dapat dilihat pada Gambar 8.

Gambar 7. DOIT ESP32 development board [18]



Gambar 8. Tata letak pin dan dimensi modul DOIT ESP32 [18]

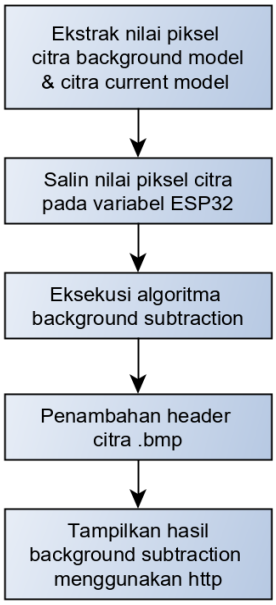


Gambar 9. Skematik dan konfigurasi pin pada modul DOIT ESP32 [18]

Skematik dari modul ESP32 yang digunakan pada penelitian ini dapat dilihat pada Gambar 9, komponen utama pada skematik adalah *chip* ESP32, chip *USB to Serial* CP2102 dan *regulator* NCP1117.

2.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak untuk menerapkan algoritma *background subtraction* pada ESP32 memiliki tahapan yang secara garis besar dapat dilihat pada Gambar 10.



Gambar 10. Tahapan implementasi background subtraction di ESP32

Keterangan tahapan implementasi adalah sebagai berikut:

a. **Ektrak nilai piksel citra *background model* dan citra *current model***, pada tahap ini citra yang digunakan memiliki format 8 bit *grayscale,* citra yang digunakan terdiri dari *background model* dan *current frame,* 2 buah input citra ini dapat dilihat pada Gambar 11.

(a) (b)

Gambar 11. (a) Background model, (b) dan current frame

Secara umum citra input memiliki dimensi *m* x *n* seperti matriks pada persamaan (2a) dan (2b).

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{1,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \quad (2a)$$

$$\begin{bmatrix} pv_{[n_x.(n_y-1)]} & pv_{\left[\left(n_x.(n_y-1)\right)+1\right]} & \cdots & pv_{[(n_x.n_y)-1]} \\ \vdots & \vdots & \ddots & \vdots \\ pv_{[n_x]} & pv_{[n_x+1]} & \cdots & pv_{[(n_x.2)-1]} \\ pv_{[0]} & pv_{[1]} & \cdots & pv_{[n_x-1]} \end{bmatrix} \quad (2b)$$

Matriks citra yang berisi nilai tiap piksel diubah dalam format array 1 dimensi, konstruksi nilai piksel dalam array 1 dimensi dapat dilihat pada Tabel 1.

Tabel 1. Nilai piksel pada array 1 dimensi

Indeks	Nilai piksel	Indeks array
0	220	Nilai piksel[0]
1	185	Nilai piksel[1]
⋮	⋮	⋮
$(n_x.n_y)-1$	240	Nilai piksel $[(n_x.n_y)-1]$

b. **Salin nilai piksel citra pada variable ESP32**, pada tahap ini array 1 dimensi disalin kedalam variabel program ESP32.
c. **Eksekusi algoritma *background subtraction***, pada tahap ini nilai piksel citra *current model* dikurangi nilai piksel citra *background model* sesuai Persamaan 1, hasil pengurangan ini disimpan sebagai citra *output*.
d. **Penambahan header citra .bmp**, pada tahap ini nilai piksel hasil *background subtraction* dari tahap sebelumnya ditambahkan dengan header bitmap, file bitmap terdiri dari beberapa bagian yang dapat dilihat pada Tabel 2. Detil tiap bagian pada header file bitmap dapat dilihat pada Tabel 3.

Tabel 2. Bagian file bitmap [19]

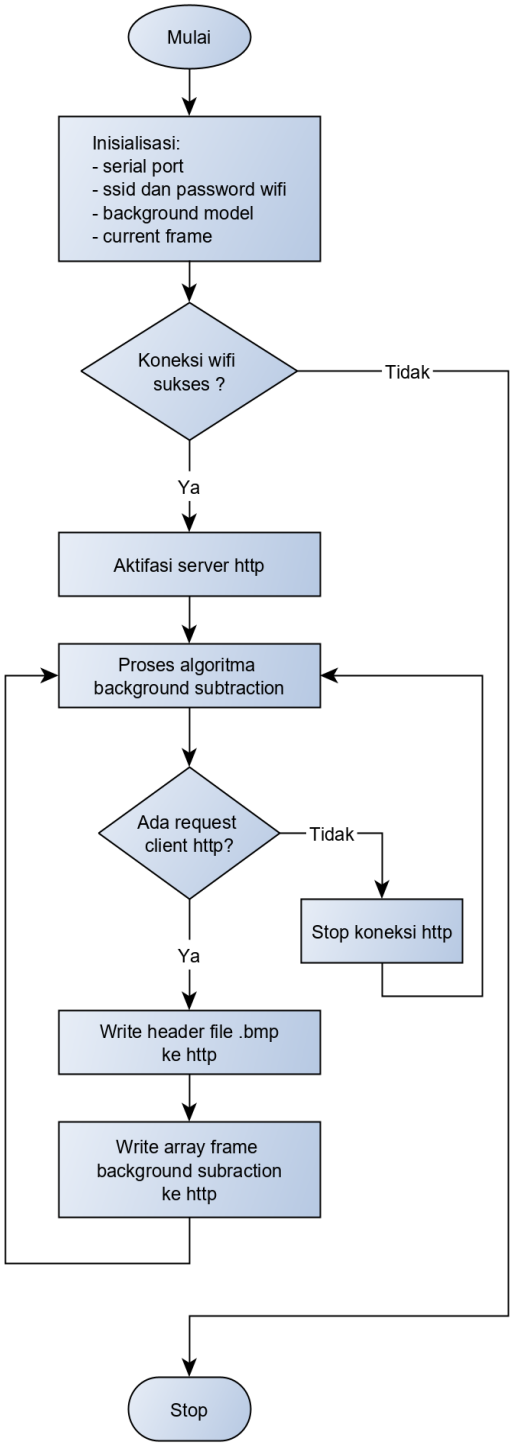
Bagian	Deskripsi
Header	*Basic file information, 14bytes*
Image Information Header	*Information about the image, 40 bytes*
Color Information (optional)	*Information about how the image encodes colors, a variable number of bytes if it's present*
Image data	*The actual image, a variable number of bytes*

Tabel 3. Detil header file bitmap [19]

Bagian	Deskripsi
BM	*2 bytes*
File size, bytes	*4 bytes*
Two "reserved values" that are not needed	*2 bytes each*
Offset to beginning of image data	*4 bytes*
Header size, bytes (should be 40)	*4 bytes*
Image width, pixels	*4 bytes*
Image height, pixels	*4 bytes*
Number of color planes	*2 bytes*
Bits per pixel, 1 to 24	*2 bytes*
Compression, bytes (assumed 0)	*4 bytes*
Image size, bytes	*4 bytes*
X-resolution & Y-resolution, pixel per meter	*4 bytes each*
Number of colors & "important colors", bytes	*4 bytes each*

e. **Tampilkan hasil *background subtraction* menggunakan http**, pada tahap ini ESP32 dijadikan *server http* lalu header .bmp beserta seluruh nilai piksel citra hasil *background subtraction* dikirimkan melalui protokol http.

Diagram alir program ESP32 untuk memproses algoritma *background subtraction* dapat dilihat pada Gambar 12.



Gambar 12. Diagram alir pengujian algoritma background subtraction di ESP32

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan menggunakan input citra yang terdiri dari *background model* dan *current frame*, kemudian memproses *background subtraction* sesuai dengan Persamaan 1 pada semua piksel. Citra yang digunakan untuk pengujian memiliki dimensi 80 x 60 piksel, dimensi pengujian ini dipilih karena jumlah memory DRAM ESP32 hanya sebesar 328KB(kilobyte), format warna yang digunakan adalah 8bit *grayscale*, hasil implementasi dapat dilihat pada Gambar 13.

(a)

(b)

(c)

Gambar 13. (a, b) Contoh citra grayscale sebagai input pengujian algoritma background subtraction, (c) citra output tanpa proses thresholding

Untuk menguji *throughput* yang dihasilkan oleh ESP32, algoritma *background subtracton* dengan input citra *grayscale* diulangi dengan beberapa variasi perulangan proses *background subtraction*, detil *throughput* yang dihasilkan dapat dilihat pada Tabel 4.

Tabel 4. Throughput ESP32

No	Ukuran frame citra *(pixel)*	Jumlah perulangan	Waktu pemrosesan *(ms)*
1.	80 x 60	10000	±2000
2.	80 x 60	20000	±3600
3.	80 x 60	30000	±5500

Berdasarkan Tabel 4, pada pengujian pertama mampu memproses *algoritma background* subtraction 10000 kali dalam 2000ms (±5000fps), pada pengujian kedua dan ketiga waktu pemrosesan menjadi lebih cepat karena *overhead* kode perulangan pada pemrograman lebih sedikit dari pengujian pertama.

4. KESIMPULAN

Berdasarkan hasil pengujian dan penelitian ini, dapat disimpulkan bahwa algoritma *background subtraction* dapat diimplementasikan di ESP32 dan mampu memproses *frame* citra dengan ukuran 80 x 60 piksel sebanyak 10000 kali dalam 2000ms (±5000fps), ukuran *frame* citra 80 x 60 piksel dipilih sebagai pengujian karena keterbatasan *memory* DRAM EPS32 sebesar 328 KB (kilobyte).

UCAPAN TERIMA KASIH

Terimakasih kepada Universitas Komputer Indonesia yang telah menyediakan sumber daya penelitian baik infrastruktur maupun peralatan penunjang sehingga memudahkan penulis dalam melakukan penelitian.

DAFTAR PUSTAKA

[1] M. Teixidó, D. Font, T. Pallejà, M. Tresanchez, M. Nogués, and J. Palacín, "An Embedded Real-Time Red Peach Detection System Based on an OV7670 Camera, ARM Cortex-M4 Processor and 3D Look-Up Tables," *Sensors*, vol. 12, no. 10, pp. 14129–14143, Oct. 2012.

[2] A. Rowe, A. Goode, I. Nourbakhsh, and D. Goel, "CMUcam3: An Open Programmable Embedded Vision Sensor," p. 17.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893 vol. 1.

[4] T. N. Nizar, N. Anbarsanti, and A. S. Prihatmanto, "Multi-object tracking and detection system based on feature detection of

the intelligent transportation system," in *2014 IEEE 4th International Conference on System Engineering and Technology (ICSET)*, 2014, vol. 4, pp. 1–6.

[5] J. Canny, "A computational approach to edge detection," *Ieee Trans. Pattern Anal. Mach. In⁷⁴⁷ligence*, 1986.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[7] T. Bouwmans, F. Porikli, B. Hferlin, and A. Vacavant, *Background Modeling and Foreground Detection for Video Surveillance*, 1st ed. Chapman & Hall/CRC, 2014.

[8] "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos - ScienceDirect." [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314213002361. [Accessed: 25-Apr-2019].

[9] O. Barnich and M. V. Droogenbroeck, "ViBe: A Universal Background Subtraction Algorithm for Video Sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[10] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*, 2017, pp. 143–148.

[11] M. Hilman, D. K. Basuki, and S. Sukaridhoto, "Virtual Hand: VR Hand Controller Using IMU and Flex Sensor," in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2018, pp. 310–314.

[12] P. Rai and M. Rehman, "ESP32 Based Smart Surveillance System," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–3.

[13] I. Allafi and T. Iqbal, "Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, 2017, pp. 1–5.

[14] M. Suárez-Albela, P. Fraga-Lamas, L. Castedo, and T. M. Fernández-Caramés, "Clock Frequency Impact on the Performance of High-Security Cryptographic Cipher Suites for Energy-Efficient Resource-Constrained IoT Devices," *Sensors*, vol. 19, no. 1, Dec. 2018.

[15] "OpenCV: How to Use Background Subtraction Methods." [Online]. Available: https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html. [Accessed: 25-Apr-2019].

[16] "Image Arithmetic - Pixel Subtraction." [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm. [Accessed: 25-Apr-2019].

[17] "ESP32 Thing Hookup Guide - learn.sparkfun.com." [Online]. Available: https://learn.sparkfun.com/tutorials/esp32-thing-hookup-guide. [Accessed: 25-Apr-2019].

[18] "GitHub - Nicholas3388/LuaNode: Esp32/esp8266 lua sdk." [Online]. Available: https://github.com/Nicholas3388/LuaNode. [Accessed: 27-Apr-2019].

[19] "Reading .bmp files." [Online]. Available: http://www.instesre.org/howto/BW_image/ReadingBitmaps.htm. [Accessed: 25-Apr-2019].