Jurnal Otomasi Kontrol dan Instrumentasi Vol 16 (2), 2024 ISSN: 2085-2517
https://doi.org/10.5614/joki.2024.16.2.8 E-ISSN: 2460-6340

Rancang Bangun Sistem Deteksi Hama Tanaman *Whiteflies* Berbasis Jaringan Sensor Nirkabel dan Aplikasi Web

Design and Development of a Whiteflies Plant Pest Detection System Based on Wireless Sensor Networks and Web Applications

[1]Tua Agustinus Tamba*), [1]Stevanus Darwin, [1]Gabriela Solaiman, dan [1]Jason Reysan

[1]Program Studi Sarjana Teknik Elektro, Fakultas Teknologi Industri, Universitas Katolik Parahyangan, Bandung, Indonesia

*) *corresponding email: ttamba@unpar.ac.id*

Abstrak

Salah satu tantangan dalam kegiatan pertanian adalah terkait upaya deteksi dan pemantauan hama yang dapat dijadikan basis untuk penentuan tindak penanganan yang tepat. *Greenhouse whiteflies* (GWF) adalah salah satu hama berbahaya yang menyerap getah tanaman sehingga membuat tanaman melemah atau bahkan mati akibat kekurangan nutrisi. Upaya deteksi hama GWF umumnya masih dilakukan secara manual menggunakan *yellow sticky trap* (YST) yang membutuhkan pengecekan oleh petani secara berulang dengan selang waktu lama antar setiap aktivitas pengecekan. Penelitian ini ditujukan untuk merancang suatu sistem otomasi pemantauan hama GWF berbasis gambar YST. Metode perancangan yang diajukan adalah melalui penggunaan suatu sistem jaringan sensor nirkabel berbasis web yang dilengkapi dengan algoritma pengolahan citra dan dioperasikan secara *real-time*. Pada metode yang diajukan, sensor kamera digunakan untuk menangkap citra YST yang kemudian diproses dengan aplikasi berbasis *website*. Pengujian purwarupa hasil penelitian menunjukkan bahwa rancangan sistem deteksi hama berfungsi dengan baik dan tanpa kendala. Secara khusus pada aspek perangkat keras, pengujian menunjukkan sensor kamera di lapangan dapat diakses dan mengirimkan data secara *real-time* ke aplikasi website yang dirancang. Sementara pada aspek algoritma pengolahan citra, teknik deteksi citra berbasis analisis histogram yang digunakan mampu mendeteksi hama *whiteflies* pada citra daun yang ditinjau dengan tingkat akurasi mencapai rentang 94%-100%.

Kata Kunci: pemrosesan citra; histogram equalization; otomasi pemantauan hama; hama whiteflies; jaringan sensor nirkabel.

Abstract

One of the main challenges in farming activities is regarding efforts to detect and monitor pests, which can be used as a basis for determining correct pest control efforts. Greenhouse whitefish (GWF) is a dangerous pest that absorbs plant sap, causing plants to weaken or even die due to lack of nutrition. Efforts to detect GWF pests are generally carried out manually using yellow sticky traps (YST), which require routine and repeated checking efforts from farmers with a long time interval between each checking activity. This research is aimed to design an automatic monitoring solution of GWF pests based on the camera image of YST paper. The proposed design method uses a web-based wireless sensor network system equipped with image processing algorithms that operate in a real-time manner. In the proposed method, the camera sensor is used to capture YST images, which are then processed using a website-based application system. The evaluation of the prototype testing suggests that the designed pest detection system functions well and without problems. With regard to the developed hardware, evaluation results show that the camera sensor can be accessed and send data to the developed website application in a real-time manner. Regarding the image processing algorithm, evaluation results show that the developed histogram based image processing technique can detect GWF on plant leaf being considered with accuracy levels up to a range of 94%-100%.

Keywords: image processing; histogram equalization; pest monitoring automation; whiteflies pest; wireless sensor network.

Makalah diterima 18 Juli 2024 – makalah direvisi 13 September 2024 – disetujui 18 September 2024

Karya ini adalah naskah akses terbuka dengan lisensi CC BY-SA.

1 Pendahuluan

Indonesia merupakan salah satu negara agraris dimana pertanian menjadi salah satu sektor penting dalam pembangunan perekonomian nasional. Pembangunan sektor pertanian diharapkan dapat memberi kontribusi dalam perluasan lapangan kerja dan pemanfaatan peluang ekonomi yang ditimbulkan globalisasi dan

liberalisasi perekonomian dunia. Untuk mencapai harapan ini, ketersediaan sumber daya manusia berkualitas dan handal serta penerapan teknologi di bidang pertanian menjadi kebutuhan yang sangat penting. Terlepas dari peluang tersebut, Indonesia saat ini masih mengalami krisis petani. Berdasarkan data Badan Pusat Statistik pada Survei Antar Sensus 2018 versi A2, diketahui bahwa pertanian Indonesia didominasi petani dengan usia di atas 45 tahun sebanyak 65,82 % atau berjumlah 18,2 juta dari total 33,48 juta petani [1]. Sementara jumlah petani berusia dibawah 25 tahun hanya 191.000 atau 0,69 % dari jumlah total petani. Minimnya jumlah petani produktif tersebut dapat memicu permasalahan di bidang ketahanan pangan nasional. Salah satu alternatif solusi untuk menjaga ketahanan pangan adalah melalui peningkatan penggunaan dan penerapan teknologi pertanian mulai dari tahap pembenihan, pemeliharaan dan pemupukan, penanganan hama/wabah penyakit, hingga proses/pasca panen [2].

Greenhouse whiteflies (GWF) adalah hama berbahaya yang menyerap getah tanaman yang membuat tanaman melemah atau bahkan mati akibat kekurangan nutrisi. Agar dapat hinggap dan tinggal di daun tanaman, GWF menghasilkan zat lengket *honey dew* yang dapat menempelkan pada daun [3],[4]. Jika dibiarkan, *honey dew* dapat menutupi permukaan daun sehingga cahaya matahari tidak dapat menyinari daun dan perlahan akan membunuh tanaman karena tidak dapat melakukan fotosintesis [5-6]. Pada tahun 2015, lebih dari 80% tanaman suatu perkebunan kentang di Sulawesi Selatan mati karena GWF [7]. GWF cenderung sulit dideteksi karena biasanya tinggal di sisi bawah daun. Selain itu, ukuran badan GWF juga sangat kecil dengan GWF dewasa hanya memiliki panjang sekitar 1-2 mm [3]-[6].

Untuk mempermudah deteksi GWF, *yellow sticky trap* (YST) sering digunakan karena latar belakangnya yang berwarna kuning dapat menarik perhatian GWF [8],[9]. Jumlah GWF yang tertangkap pada YST kemudian akan digunakan untuk mengestimasi jumlah populasi GWF pada sebuah daerah untuk penentuan langkah penanggulangan yang diperlukan [10],[11]. Di banyak negara termasuk Indonesia, deteksi GWF masih dilakukan secara manual menggunakan YST dan membutuhkan selang waktu yang lama antar setiap pengecekan. Dengan ukuran tubuh yang sangat kecil dan warna hampir transparan, perhitungan GWF secara manual memiliki akurasi rendah [12],[13]. Hal ini mengindikasikan perlunya metode deteksi GWF secara cepat untuk mengurangi potensi timbulnya kerusakan.

Penelitian ini mengajukan solusi otomasi proses pemantauan hama GWF menggunakan sebuah sistem jaringan sensor nirkabel berbasis web yang dilengkapi algoritma pengolahan citra (*image processing*) yang dioperasikan secara *real-time.* Jaringan sensor nirkabel merupakan sebuah sistem dimana sensor mengirim data ke komponen *sink* yang akan menampung data dari sensor tersebut menggunakan teknologi internet tanpa membutuhkan kabel data [14],[15]. Pada penelitian ini, sensor yang digunakan adalah kamera untuk menangkap citra YST, sementara *sink* akan digunakan untuk melakukan pemrosesan citra yang diperoleh sensor kamera. Keuntungan sistem ini antara lain instalasinya lebih mudah karena tidak membutuhkan kabel yang panjang, cakupan wilayah yang luas sesuai area pertanian tertentu, serta pemeliharaan sistem lebih mudah dilakukan [15]. Kelebihan utama yang ditawarkan sistem yang dirancang adalah komponen yang digunakan terbilang umum dan mudah diperoleh di pasaran dengan biaya yang relatif rendah. Selain itu, algoritma pengolahan citra berbasis teknik histogram yang digunakan juga membutuhkan sumber daya komputasi yang rendah. Analisis gambar YST juga dapat dilakukan pada perangkat dengan kapasitas yang lebih kuat untuk pemantauan secara paralel di beberapa daerah. Hasil analisis ditampilkan pada website agar dapat diakses petani dari area perkebunan yang berbeda secara daring. Sistem pemantauan otomatis tersebut juga dapat digunakan untuk tujuan manajemen/kontrol hama di area perkebunan yang ditinjau.

2 Metode

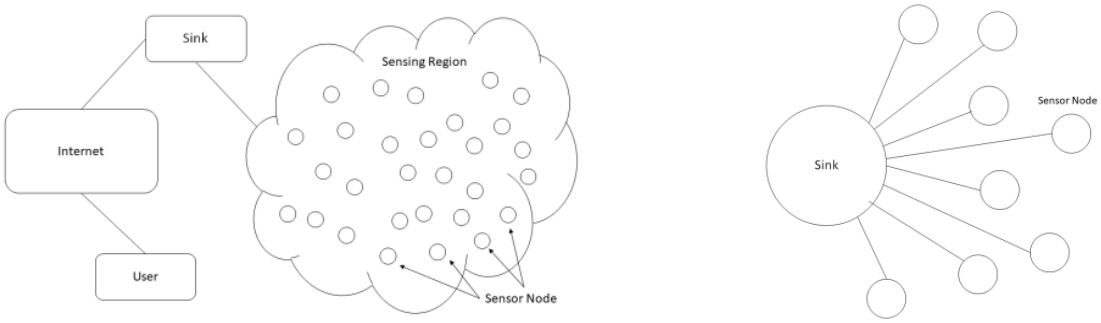
2.1 Jaringan Sensor Nirkabel

Jaringan sensor nirkabel atau *wireless sensor network* (WSN) adalah jaringan sistem komunikasi yang terdiri dari beberapa perangkat sensor sebagai komponen *node* yang dilengkapi kemampuan penginderaan, komunikasi nirkabel, dan kapasitas komputasi tertentu [16],[17]. *Node* sensor terdistribusi secara spasial, baik di dalam maupun di luar ruangan. Sebuah perangkat WSN ditujukan untuk mengumpulkan data lingkungan yang penempatan perangkat nodenya dapat diketahui atau tidak diketahui. Setiap *node* pada WSN dapat berkomunikasi secara *real-time* dengan perangkat lain untuk melakukan transfer/ fusi data/informasi. Model komunikasi didefinisikan oleh topologi WSN sesuai aplikasian WSN dapat memiliki satu atau dua topologi yang sama dalam suatu jaringan yang umumnya didefinisikan berdasarkan peran *node* [17]. Topologi dapat berdasarkan model yang ada atau sumber daya jaringan yang tersedia.

Gambar 1(a) menunjukkan model WSN dan hubungan antar komponen *user*, *internet*, *sink* dan *node* sensor. *Sink* berfungsi mengirimkan perintah ke *node* sensor, dan setiap *node* mengikuti perintah tersebut dan bekerjasama dengan *node* sensor lainnya. Ketika perintah selesai dilakukan dan setiap *node* telah

mengumpulkan data, maka *node* sensor mengirimkan data tersebut kembali ke *sink*. *Sink* juga berfungsi sebagai gerbang untuk memberikan koneksi terhadap jaringan lainnya melalui internet. Setelah seluruh data dari *node* sensor diterima oleh *sink*, proses data dilakukan dan informasi dikirim ke *user* melalui internet.

Node sensor pada penelitian ini memiliki empat (4) komponen utama, yaitu (i) suplai daya listrik (power supply) untuk mengaktifkan node, (ii) unit pengolahan data berupa mikrokontroler, (iii) elemen sensor sesuai variabel fisis yang diamati, dan (iv) sistem komunikasi untuk transfer data dengan sink atau sensor lain. Arsitektur jaringan yang digunakan pada penelitian ini adalah single-hop network di mana setiap node sensor terhubung langsung ke sink seperti ditunjukkan di Gambar 1 (b). Pada arsitektur ini tidak ada komunikasi data antar node sehingga memungkinkan node sensor untuk berada pada jarak berjauhan sehingga transmisi jarak jauh memungkinkan dilakukan. Akibatnya, arsitektur ini juga membutuhkan daya besar untuk kebutuhan bagian sistem komunikasinya.



(a) Konfigurasi WSN (b) Arsitektur single-hop

Gambar 1. Ilustrasi jaringan sensor nirkabel atau WSN

2.1.1 Web Server

Web server merupakan sebuah sistem yang menyampaikan konten atau layanan kepada pengguna melalui internet. *Web server* umumnya terdiri dari *server* fisik, sistem operasi (*operating system*, OS) dari *server*, dan perangkat lunak untuk protokol komunikasi HTTP yang diperlukan. *Web server* merupakan *internet server* yang merespon permintaan HTTP untuk menyampaikan konten/servis [18]. *Web server* berbeda dari *website* di mana *web server* adalah perangkat yang menjalankan *website* sedangkan *website* adalah tampilan yang dilihat pengguna. Untuk *web server* pada penelitian ini, Raspberry Pi digunakan sebagai server fisik, Linux digunakan untuk OS, dan Apache digunakan sebagai perangkat lunak. Dalam perancangan *web server*, beberapa konsep yang juga ditinjau mencakup *flask framework*, *relational database management system*, dan Apache *server*.

2.1.2 Flask Framework

Framework adalah suatu paket berisi fungsi umum yang membantu penyusunan *website* untuk menghindari *input* kode yang tidak diperlukan. *Frameworks website* dibagi menjadi dua, yaitu *frontend* dan *backend*. *Frontend* dikenal juga dengan CSS *frameworks* dan merupakan bagian *website* yang dapat dilihat/berinteraksi dengan pengguna. *Backend* merupakan bagian di balik layar yang terjadi saat penyusunan *website* [18]. Pada penelitian ini, *flask framework* berbasis pemrograman Python dipilih dengan pertimbangan berikut:

- Python bersifat *open source* dan dilengkapi dengan berbagai *library* yang mendukung pengolahan data gambar yang dibutuhkan di penelitian ini.
- Mempermudah integrasi *server* dengan *website* dengan menggunakan bahasa pemrograman yang sama yaitu Python.
- Kesamaan bahasa pemrograman berbasis Python yang digunakan di berbagai *website* umum seperti Google, Youtube, Instagram, dll.

Flask dipilih sebagai *framework* Python karena tergolong *micro-framework* dan berisi fungsionalitas dasar yang tidak membatasi berbagai pengembangan yang mungkin dilakukan. *Flask* juga bekerja sebagai perekat berbagai *library* atau *toolbox* pada bahasa pemrograman yang mungkin digunakan/diperlukan.

2.1.3 Relational Database Management System

Database management system (DMS) atau atau sistem manajemen basis data adalah sistem penyimpanan data terkomputerisasi [18]. Beberapa operasi dapat dilakukan dengan DMS, antara lain manipulasi data yang ada di dalam DMS atau manajemen struktur DMS itu sendiri. DMS dapat dikelompokkan berdasarkan tipe

atau struktur data yang digunakan. *Relational DMS* (RDMS) dapat digunakan untuk berbagai hal seperti mengatur data/info suatu musik hingga mengatur data/informasi finansial institusional dengan semua transaksi yang ada. RDMS tergolong sistem yang fleksibel dan relatif mudah digunakan untuk mengatur berbagai informasi [18]. Pada RDMS, terdapat beberapa relasi penghubung:

- *One-to-one*, menggambarkan satu baris data pada suatu tabel yang terhubung dengan satu baris data lainnya pada tabel yang berbeda.
- *One-to-many*, menggambarkan satu baris data pada suatu tabel yang terhubung dengan satu atau lebih baris data pada tabel yang berbeda
- *Many-to-many*, menggambarkan satu atau lebih baris data pada suatu tabel yang terhubung dengan satu atau lebih baris data di tabel lain yang berbeda.

Pada penelitian ini, basis data yang digunakan untuk menyimpan informasi pada sistem yang dirancang adalah MySQL yang juga bersifat *open-source*.

2.1.4 Apache Server

Apache adalah web *server* tipe *open-source* yang umum digunakan pada sistem dengan OS berbasis Linux, Microsoft Windows, atau OpenVMS yang mengizinkan pengguna untuk meluncurkan *website* di internet. Apache merupakan aplikasi modular, sehingga strukturnya dapat memberi fungsi tambahan pada aplikasi utamanya. Apache umumnya di kombinasikan dengan basis data SQL, seperti MySQL atau MariaDB, dan bahasa pemrograman dari sisi *server*, seperti PHP [17],[18].

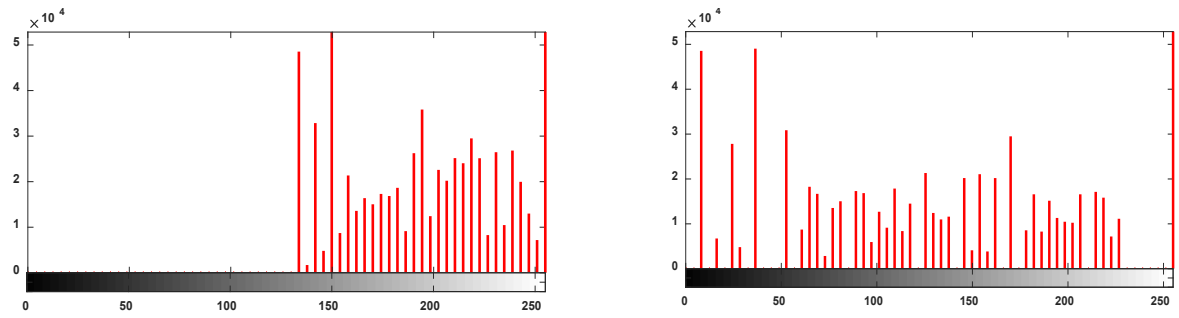
2.2 Algoritma Pengolahan Citra

Pengolahan citra pada prinsipnya merupakan suatu proses yang digunakan untuk mengubah suatu gambar menjadi bentuk data digital yang kemudian dijadikan dasar bagi implementasi berbagai operasi matematis dengan tujuan untuk memperoleh informasi/data yang diinginkan. Pada penelitian ini, metode pengolahan citra digunakan untuk mendeteksi adanya hama GWF pada YST.

2.2.1 Metode Histogram Equalization

Pada metode pengolahan citra, histogram digunakan untuk memodelkan jumlah setiap nilai intensitas cahaya pada gambar yang ditinjau dan disajikan dalam bentuk grafik batang. Dari histogram dapat diketahui distribusi nilai intensitas cahaya pada suatu gambar yang mempermudah analisis komposisi gambar. Salah satu aspek analisis tersebut adalah terkait tingkat *contrast* pada gambar serta metode yang diperlukan untuk memodifikasinya.

Seperti diilustrasikan di Gambar 2(a), suatu gambar seringkali memiliki komponen pixel yang terpusat di rentang nilai tertentu karena diambil dengan pencahayaan yang kurang memadai. Akibatnya gambar menjadi terlalu gelap/terang dan tidak memberikan informasi yang diinginkan. Untuk mengatasi ini, *histogram equalization* (HE) dapat digunakan agar sedemikian rupa mengubah distribusi intensitas cahaya gambar lebih merata seperti diilustrasikan di Gambar 2(b).



(a) Sebelum *histogram equalization* (b) Setelah *histogram equalization*

Gambar 2. Ilustrasi proses *histogram equalization*

Pada prinsipnya, HE menggunakan suatu fungsi non-linier untuk memetakan ulang nilai intensitas terang/gelap pada gambar yang ditinjau demi menghasilkan gambar yang memiliki intensitas dengan distribusi serba sama (*uniform*). Dalam hal ini, HE akan menghasilkan gambar dengan histogram distribusi intensitas berbentuk datar (*flat*) [19],[20]. Operasi fungsi pemetaan pada metode HE dapat dilakukan dengan teknik *lookup table*. Dalam hal ini, metode HE secara umum mencakup empat (4) tahap berikut:

- Penentuan histogram intensitas gambar *input* yang ditinjau
- Normalisasi histogram yang diperoleh pada rentang nilai [0,1]
- Transformasi histogram ternormalisasi menjadi tabel warna
- Pemetaan gambar input menggunakan tabel warna yang diperoleh

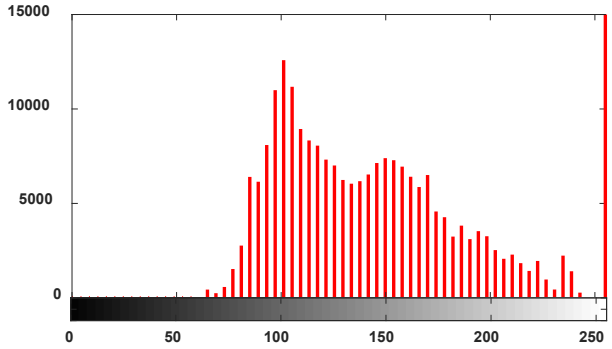
Pada penelitian ini, implementasi HE dimulai dengan penentuan *probability mass function* (PMF) intensitas cahaya gambar yang kemudian dipetakan menjadi histogram. PMF tersebut kemudian dinormalisasi pada rentang nilai [0,1] untuk menghasilkan grafik *cumulative distribution function* (CDF). Untuk melakukan *equalization*, setiap nilai CDF dikalikan dengan level intensitas abu-abu dikurang 1. Sebagai contoh jika gambar memiliki nilai intensitas 8-bit atau 256, maka nilai CDF dikalikan oleh 256-1 = 255. Hasil dari perkalian kemudian dibulatkan ke bawah untuk memperoleh nilai CDF yang lebih datar (*uniform*).

Walaupun HE dapat mendistribusikan nilai intensitas pada gambar, terkadang distribusi tersebut tidak memperbaiki namun justru memperburuk nilai *contrast* yang dihasilkan. Hal ini dapat terjadi jika gambar *input* memiliki nilai intensitas yang ekstrim pada rentang nilai tertentu [20],[21]. Karena HE dilakukan dalam skala global, maka ketika ada kesalahan pada rentang kecil tertentu, distribusi tetap dilakukan di semua rentang pixel gambar dan merusak keseimbangan *contrast* pada rentang pixel lain. Untuk mengatasi ini, metode *contrast limited adaptive histogram equalization* (CLAHE) dapat digunakan.

2.2.2 Metode CLAHE

CLAHE merupakan sebuah variasi algoritma HE yang digunakan untuk memperbaiki keseimbangan tingkat kontras (*contrast*) dari level pencahayaan (gelap/terang) suatu gambar/citra [22]. Prinsip kerja CLAHE secara umum dikembangkan dari algoritma *adaptive histogram equalization* (AHE) yang menerapkan HE dengan menggeser sebuah *window* dengan ukuran yang dapat ditentukan sehingga penentuan distribusi pixel tidak dilakukan di skala global namun pada skala lokal. Dalam hal ini, perubahan distribusi pixel pada area tertentu tidak mempengaruhi area lain. AHE memiliki kelemahan dalam penentuan nilai distribusi karena berpotensi menggeser nilai intensitas terlalu jauh dan mengakibatkan gambar menjadi terlalu terang/gelap.

CLAHE mengatasi keterbatasan AHE dengan membatasi nilai intensitas cahaya gambar, sehingga ketika didistribusikan hanya dapat bergeser hingga batas yang ditentukan [22]. Gambar 3 mengilustrasikan CLAHE yang membatasi pergeseran hingga nilai 60 pada batas bawah. Dengan metode CLAHE, perubahan *contrast* dapat dilakukan untuk mencegah adanya kehilangan informasi yang dibutuhkan pada gambar. CLAHE dapat dilakukan tidak hanya pada intensitas cahaya, namun juga pada komponen gambar lain selama nilainya memiliki vektor 1 dimensi. Salah satu aplikasi tersebut adalah HSV CLAHE dimana algoritma CLAHE digunakan pada gambar dengan skema warna HSV (*Hue*, *Saturation*, *Value/Brightness*). Karena skema warna HSV merupakan gambar dengan vektor 3 dimensi, maka vektor tersebut perlu dipecah menjadi vektor setiap komposisinya yaitu H, S, dan V. HSV CLAHE bekerja sama seperti CLAHE, namun data yang dimanipulasi bukan intensitas, melainkan komponen warna H, S, dan V.



Gambar 3. Ilustrasi hasil algoritma CLAHE pada histogram di Gambar 2(b)

2.2.3 Teknik Thresholding

Thresholding adalah metode segmentasi yang memisahkan objek berdasarkan nilai intensitas cahaya di setiap pixelnya dan mengubahnya menjadi *grayscale* [23]. Data gambar kemudian dikonversi menjadi vektor 1 dimensi terkait informasi nilai intensitas cahaya. Hasil ini berbeda dari skema lain seperti RGB yang memiliki tiga (3) nilai, yaitu merah (R), hijau (G), biru (B).

Pada metode *thresholding*, nilai intensitas cahaya setiap pixel dibandingkan dengan suatu nilai batas (*threshold*) tertentu [6]. Jika intensitas melebihi batas yang ditentukan, algoritma menentukan *output* bernilai

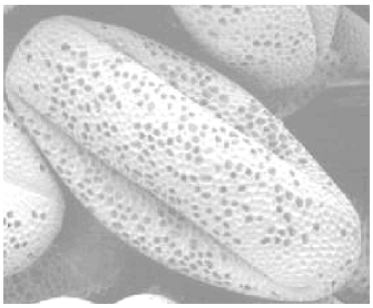
1; sebaliknya algoritma akan menentukan *output* bernilai 0. Fungsi pemetaan intensitas pada teknik *thresholding* dalam hal ini memenuhi persamaan (1).

$$q(x,y) = \{1 \quad \text{jika } q(x,y) > T\,, \qquad 0 \quad \text{jika } q(x,y) \leq T\} \tag{1}$$

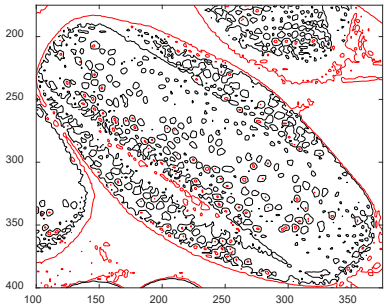
di mana $q(x,y)$ adalah *output* hasil *thresholding* dengan (x,y) menyatakan koordinat dua dimensi pixel yang dibandingkan, p adalah nilai intensitas cahaya, dan T adalah nilai batas (*threshold*) yang digunakan. Sesuai (1), *output* dari metode *thresholding* adalah gambar biner dengan warna putih/hitam di setiap pixelnya.

2.2.4 Deteksi Kontur Citra

Kontur gambar merupakan suatu garis yang menyambungkan seluruh titik di sepanjang batas objek yang memiliki warna/intensitas yang sama [23]. Kontur digunakan untuk misalnya memisahkan objek gambar dari latar belakang untuk memperoleh informasi objek seperti bentuk, dimensi, dan posisi (lihat Gambar 4). Untuk meningkatkan akurasi deteksi, gambar *input* dapat dikonversi terlebih dahulu menjadi gambar biner/*gray*.



(a) Objek gambar *input*.

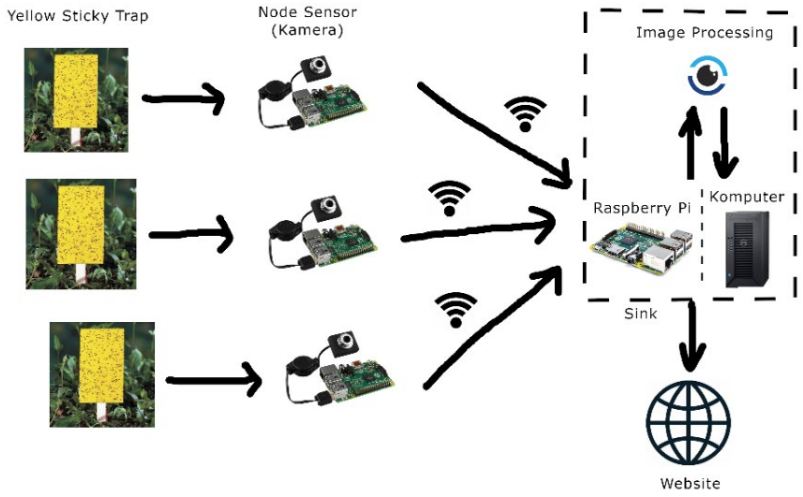


(b) Hasil deteksi kontur.

Gambar 4. Ilustrasi proses deteksi kontur

3 Perancangan Sistem

Hasil Gambar 5 menunjukkan skematika sistem deteksi hama GWF berbasis WSN dan teknik pengolahan citra yang dikembangkan pada penelitian ini. *Node* pada WSN terdiri dari sebuah kamera yang tersambung ke modul Raspberry Pi [24] dan digunakan untuk menangkap gambar YST. Pengambilan gambar oleh kamera dilakukan secara otomatis pada waktu cacah yang diatur dengan Raspberry Pi. Gambar dari kamera kemudian dikirim dari Raspberry Pi menuju *sink* yang merupakan perangkat Raspberry Pi yang di dalamnya menjalankan algoritma pengolahan citra. Data hasil pengolahan citra terkait tingkat kerusakan tanaman secara detail (tempat, waktu, populasi GWF) lalu diunggah ke *webserver* untuk ditampilkan di *website* agar dapat dilihat/dimanfaatkan pengguna.

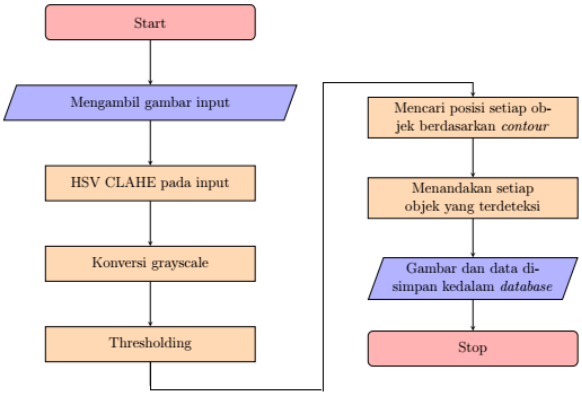


Gambar 5. Skematika sistem deteksi hama GWF berbasis *website*

Perangkat keras pada rancangan WSN adalah modul kamera ESP32-CAM yang terkoneksi dengan komputer mini Raspberry Pi 3b menggunakan kabel USB. Koneksi tersebut memungkinkan kamera di program langsung dari Raspberry Pi menggunakan script *crontab* berbasis UNIX yang juga merupakan basis OS Raspberry Pi. Raspberry Pi 3b memiliki performa tinggi dan dilengkapi fitur cukup lengkap seperti modul *wifi* dan *bluetooth*.

3.1 Rancangan Algoritma Pengolahan Citra

Gambar 6 menunjukkan diagram alir algoritma pengolahan citra pada penelitian ini. Berdasarkan diagram ini, data gambar dari kamera diproses terlebih dahulu dengan metode HSV CLAHE untuk mengubah komposisi warna gambar. Pengubahan komposisi dilakukan melalui manipulasi nilai *saturation* dan *brightness* dari gambar untuk meningkatkan nilai intensitas cahaya dari objek GWF berwarna putih yang akan dideteksi. Gambar yang telah diproses dengan algoritma HSV CLAHE kemudian dikonversi menjadi tipe *grayscale*.



Gambar 6. Diagram alir rancangan algoritma pengolahan citra

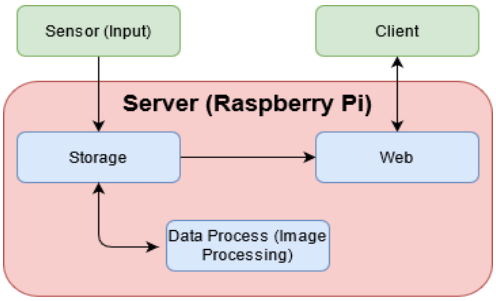
Hasil citra *grayscale* selanjutnya diproses dengan algoritma *thresholding* untuk memisahkan objek utama (GWF) dari latar belakang gambar. Pemisahan dilakukan dengan konversi komponen setiap pixel berdasarkan nilai intensitas cahaya. Setelah proses *thresholding*, posisi setiap objek pada gambar kemudian ditentukan menggunakan algoritma deteksi *contour* untuk penghitungan jumlah hama GWF di gambar. Data citra dengan informasi posisi/jumlah GWF tersebut kemudian disimpan pada *database* untuk selanjutnya diunggah ke *website*.

3.2 Rancangan *Website*

Perancangan *web server* secara umum dilakukan dalam dua tahap, yaitu perancangan alir data dan perancangan fungsi pada *website*.

3.2.1 Perancangan Alir Data

Gambar 7 menunjukkan diagram alir dari data yang masuk ke dalam WSN. Pertama, data citra yang diperoleh dari sensor kamera dikirimkan ke *server* untuk disimpan pada penyimpanan data lokal. Data pada Raspberry Pi *server* tersebut kemudian dienkripsi untuk selanjutkan dikirimkan ke basis data yang digunakan. Proses pengolahan citra dilakukan dengan mengakses data citra pada penyimpanan data lokal tersebut. Setelah di proses, data citra tersebut kemudian dikembalikan ke penyimpanan data lokal yang lokasinya diatur berbeda dengan lokasi data awal yang diterima dari sensor kamera. Data pada basis data kemudian ditampilkan pada *website* agar dapat diakses klien atau pengguna (*user*) *website*.

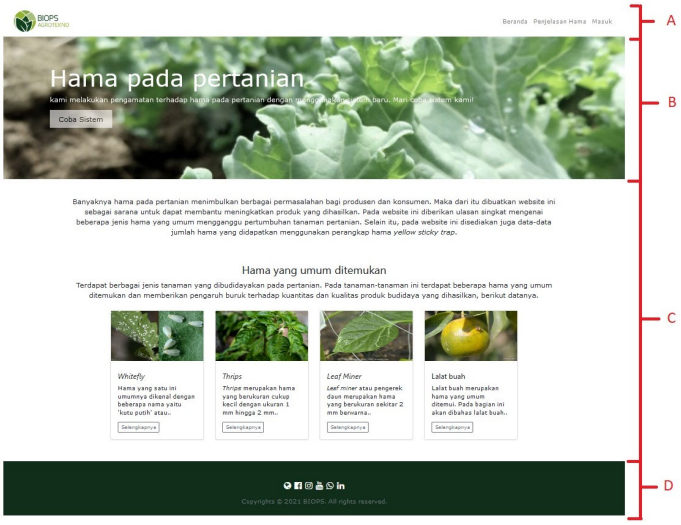


Gambar 7. Skematika perancangan aliran data pada sistem

3.2.2 Perancangan Alir Fungsi

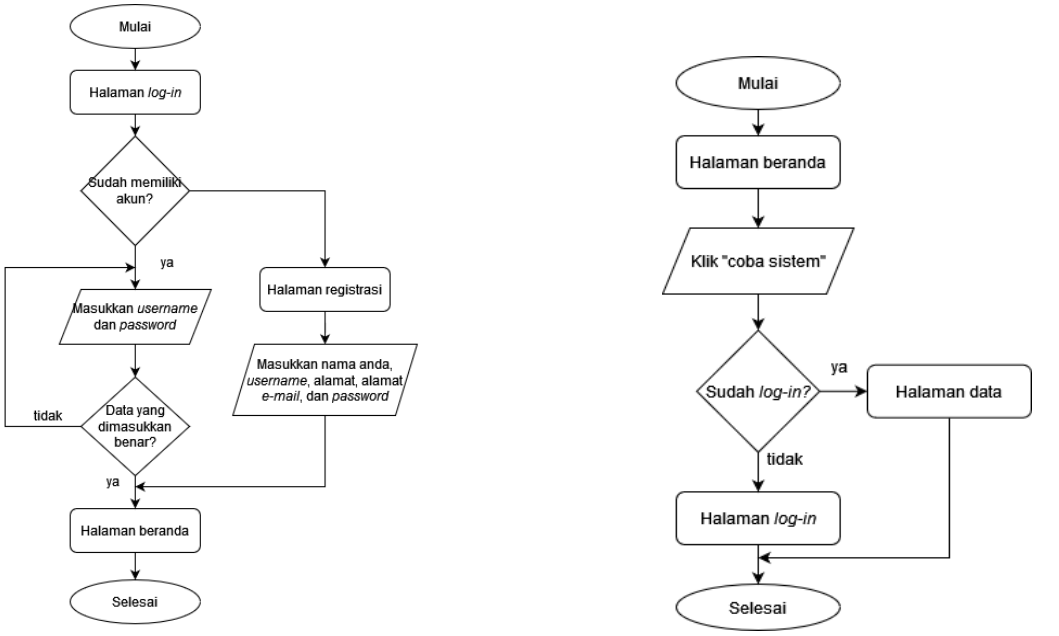
Gambar 8 menunjukkan tampilan *website* yang dirancang pada penelitian ini serta fungsionalitas yang tersedia, diantaranya fungsi *log-in* dan registrasi. Tampilan pada *website* memiliki beberapa bagian dengan fungsional sebagai berikut.

- Bagian **A**: *navigation bar* yang merupakan posisi fungsi untuk mengakses web dengan pilihan *"Beranda"*, *"Penjelasan Hama"*, dan *"Masuk"*.
- Bagian **B**: *web header* yang dirancang untuk menarik perhatian pengguna. Pada bagian ini juga terdapat tombol *"Coba Sistem"* untuk menampilkan data hasil pengolahan citra yang dapat diakses menggunakan akun pengguna yang telah masuk/melakukan *log-in*.
- Bagian **C**: *web body* yang merupakan lokasi peletakan konten. Pada bagian ini terdapat empat pilihan penjelasan hama yang ditinjau pada penelitian ini.
- Bagian **D**: *web footer* yang menampilkan informasi kontak pemilik *website*.



Gambar 8. Skematika perancangan aliran data pada sistem

Sebagai ilustrasi, diagram alir fungsi *"Log-In"* dan *"Coba Sistem"* ditunjukkan di Gambar 9(a) dan 9(b). Untuk memulai akses, pengguna dapat memasukkan data *log-in* atau membuat akun baru jika belum memiliki akun. Jika berhasil *log-in*, *navigation bar* akan berubah menjadi *"Beranda"*, *"Penjelasan Hama"*, *"Data"* dan *"Keluar"*. Pemilihan tombol *"Coba Sistem"* pada *header* akan menampilkan halaman data yang tersedia. Informasi data yang tersedia meliputi data dan waktu pengambilan citra dari kamera, hasil pengolahan citra yang dilakukan dalam bentuk persen total jumlah hama GWF yang terdeteksi, serta rekomendasi bagi pengguna berdasarkan data dan hasil pengolahannya. Informasi dan berbagai data terbaru akan ditampilkan pada bagian bawah halaman *website*.

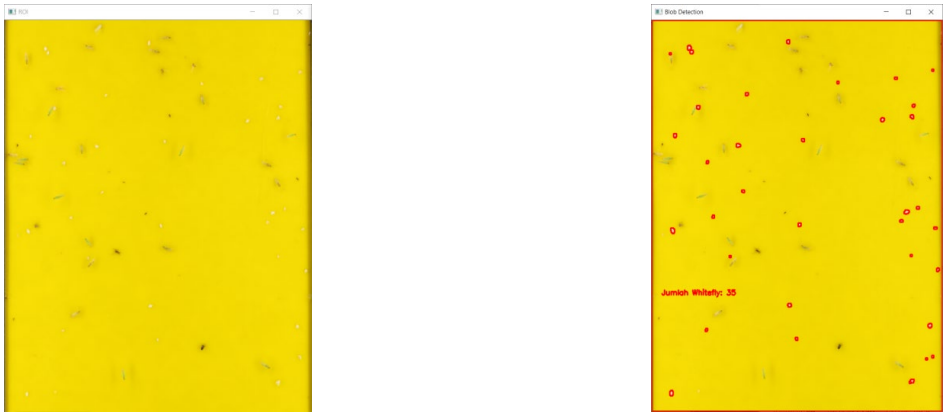


(a) Diagram alir fungsi "*Log-in*" (b) Diagram alir fungsi "*Coba Sistem*"

Gambar 9. Diagram alir fungsi "*Log-in*" dan "*Coba Sistem*"

4 Hasil dan Analisis

4.1 Hasil dan Analisis Metode Pengolahan Citra



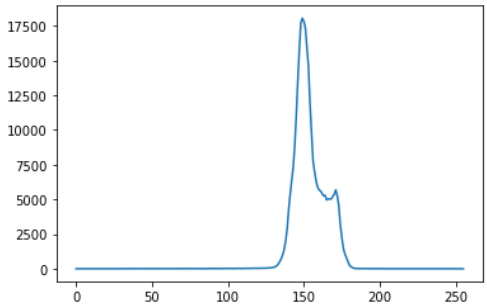
(a) Citra *input* (b) *Output* pengolahan citra

Gambar 10. Data citra percobaan I dengan metode *thresholding*

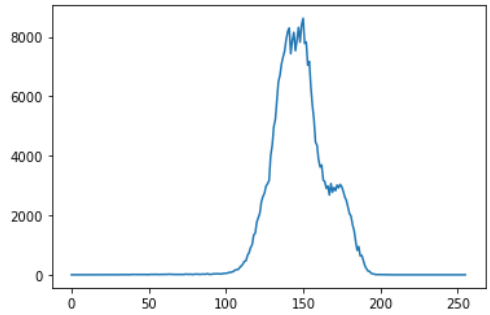
Pengujian I dilakukan pada data citra ideal seperti ditunjukkan di Gambar 10(a) yang kemudian diproses dengan metode *thresholding* seperti ditunjukkan pada Gambar 10(b) dengan nilai batas (*threshold*) 210. Algoritma yang digunakan berhasil mendeteksi sebanyak 35 dari jumlah aktual 33 GWF seperti ditunjukkan Gambar 11. Berdasarkan evaluasi hasil deteksi metode *thresholding* yang digunakan, diketahui bahwa tambahan 2 buah objek yang dideteksi merupakan hasil *false positive*, di mana data gangguan (*noise*) pada citra yang pada dasarnya merupakan debu ataupun objek lain dideteksi sebagai GWF. Pada penelitian ini, jumlah hasil deteksi yang merupakan *false positive* ditentukan sebagai *error* deteksi yang secara persentase mengakibatkan pengurangan tingkat akurasi deteksi yang dihasilkan. Dalam hal pengujian ini, jika persentase eror deteksi didefinisikan sebagai (2/33)×100%=6%, maka algoritma deteksi yang digunakan disimpulkan memiliki kinerja cukup baik dengan rentang nilai akurasi sebesar 94%-100%.

(a) Citra *input*

(b) *Output* pengolahan citra



(c) Histogram pixel pada citra *input.*

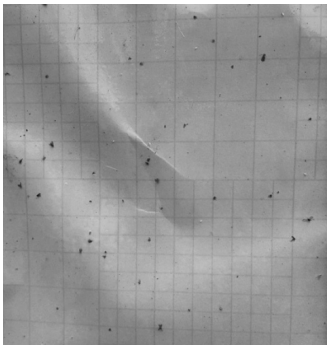


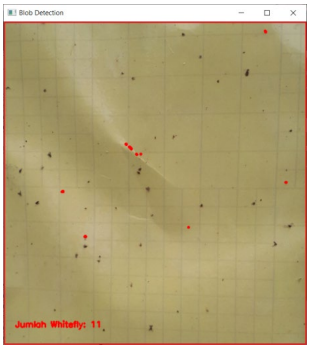
(d) Histogram pixel pada *output* HSV CLAHE

Gambar 11. Hasil implementasi HSV CLAVE percobaan II

Pengujian II dilakukan pada data citra YST asli yang diperoleh dari perkebunan (Gambar 11(a) dengan menggunakan algoritma *thresholding* standar. Algoritma standar tersebut sama sekali tidak dapat mendeteksi GWF pada citra YST yang ditinjau (Gambar 11(b)). Metode HSV CLAHE kemudian digunakan untuk mengatasi masalah ini. Gambar 12 menunjukkan hasil pemrosesan dan tipe *grayscale* dari citra yang diperoleh dengan metode HSV CLAHE. Warna citra yang diperoleh pada Gambar 12(a) menjadi lebih tajam dengan tingkat gelap/terang yang meningkat dibanding citra awal di Gambar 11(a). Histogram citra sebelum penggunaan metode HSV CLAHE di Gambar 11(c) memiliki nilai puncak 17500 pixel pada nilai sekitar 160 saja, sedangkan histogram setelah implementasi HSV CLAHE pada Gambar 11(d) memiliki nilai puncak lebih rendah dengan nilai 8000 pada rentang yang lebih lebar yaitu 140 hingga 150.

Citra dengan tingkat kontras yang lebih baik tersebut kemudian dikonversi menjadi tipe *grayscale* (Gambar 12(a)) dan selanjutnya diproses dengan metode *thresholding* untuk mendeteksi adanya GWF pada YST. Pada revisi percobaan II, nilai batas (*threshold*) diturunkan dari 210 menjadi 200 sesuai dengan karakteristik cahaya citra gambar yang ditinjau. Hasil pengujian mencapai akurasi 75% ketika sinyal gangguan *false positive* diabaikan (Gambar 12(b)). Berdasarkan observasi, nilai akurasi dapat ditingkatkan dengan konsekuensi jumlah gangguan terdeteksi juga ikut meningkat. Parameter deteksi yang diuji tidak dapat digunakan serba sama untuk seluruh citra karena masing-masing memiliki kondisi penerangan yang berubah pada tahapan pengambilan citra. Parameter uji yang berbeda untuk citra dengan kondisi penerangan yang berbeda akan menghasilkan sinyal gangguan signifikan.





(a) Konversi citra menjadi *grayscale* (b) Hasil deteksi GWF

Gambar 12. Hasil deteksi GWF menggunakan HSV CLAHE dan *thresholding*

4.2 Hasil dan Analisis Perangkat Lunak WSN

Pengujian *website* dilakukan dengan dua cara, yaitu lokal dan nonlokal. Uji lokal dilakukan dengan menjalankan *website* pada komputer untuk melihat apakah tampilan yang dihasilkan sesuai dengan rencana dan rancangan. Uji lokal terkait fitur mencakup melakukan register, akses data, *log-in*, dan *log-out*.

Pada uji non-lokal, data penyusun *website* dikirim ke *server* (Raspberry Pi) yang akan digunakan, dan alamat untuk membuka *website* yang telah disusun tersebut disisipkan pada *website* utama milik perusahaan. Setelah itu dilakukan penyesuaian lingkungan pada *server*, seperti instalasi aplikasi yang dibutuhkan, aktivasi basis data, dan membangun dan menyesuaikan basis data yang akan digunakan. Setelah penyesuaian lingkungan *server* selesai dilakukan, maka *website* siap dijalankan dan dapat diakses pengguna selama *server* terhubung dengan listrik dan koneksi internet yang lancar.

Rancangan perangkat lunak diintegrasikan dengan perangkat keras yang digunakan yang dalam hal ini adalah *node* sensor kamera dan *sink* Raspberry Pi. Data pengukuran dari sensor diproses oleh perangkat lunak di *node* sensor untuk kemudian dikirimkan ke *sink*. Pengguna akan menerima data pengukuran setiap *node* sensor langsung dari *sink* tersebut melalui aplikasi *desktop-based*. Secara teknis, flow data pengukuran yang dibaca sensor hingga ditampilkan kepada pengguna ditunjukkan pada Gambar 7. Perangkat Lunak yang dirancang pada penelitian ini memiliki beberapa fungsi berikut:

- Melakukan pengambilan data sesuai interval waktu yang ditentukan.
- Mengirimkan data digital dari *node* sensor Raspberry Pi ke *sink*.
- Menerima data dari *node* sensor lalu mengirimkan ke penyimpanan data lokal.
- Memasukkan alamat penyimpanan data lokal ke basis data di *sink*.
- Melakukan proses pengambilan data dari data gambar.
- Menampilkan data hasil pengukuran/pengolahan ke pengguna.
- Menambahkan pengguna baru.

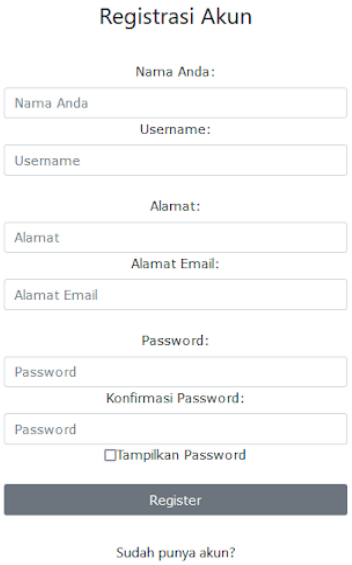
4.2.1 Analisis Perangkat Lunak pada *Sink*

Perangkat lunak *sink* berhubungan langsung dengan pengguna. Sistem pada *sink* menampung data pengukuran berupa data gambar dari *node* sensor, menyimpannya di penyimpanan lokal, dan kemudian menyimpan lokasinya di basis data yang berada pada *sink*. Aplikasi pada *sink* adalah *web-based* dan dibangun untuk pengguna dengan berbasis web. Pengguna dapat melakukan log-in dengan mengisi data *username* dan kata sandi pada halaman utama.

Jika proses login berhasil, maka pengguna akan diarahkan menuju halaman beranda dengan karakteristik *landing page* yang memiliki *shortcut* menuju tujuan halaman lainnya, seperti halaman penjelasan hama, tampilan data, atau unggah data. Hasil proses data berada pada halaman tersendiri yang menampilkan hasil pengambilan gambar dari setiap *node* dalam bentuk objek kartu. Skema yang sama digunakan untuk pengaturan penjelasan fitur hama, unggah, dan registrasi pengguna yang seluruhnya disediakan sebuah halaman tersendiri.

Pada pengujian yang dilakukan, *website* pada *server* dapat di akses dengan baik dan tanpa kendala. Selain itu, tampilan *website* dan basis data juga berfungsi dengan baik tanpa kendala. Tampilan **Beranda** *website* ditunjukkan di Gambar 8 dengan beberapa fitur seperti *navigation bar*, *header*, *body*, dan *footer*. Untuk mengakses *website*, pengguna dapat mengakses fitur **Log-in** seperti ditunjukkan pada Gambar 13 yang juga memungkinkan pengguna baru untuk mendaftar. Untuk pendaftaran akun pertama kali, data yang dibutuhkan mencakup nama, nomor telepon, alamat, kata sandi, dan e-mail.

(a) Fitur untuk pengguna pemilik akun



(b) Fitur pendaftaran akun baru

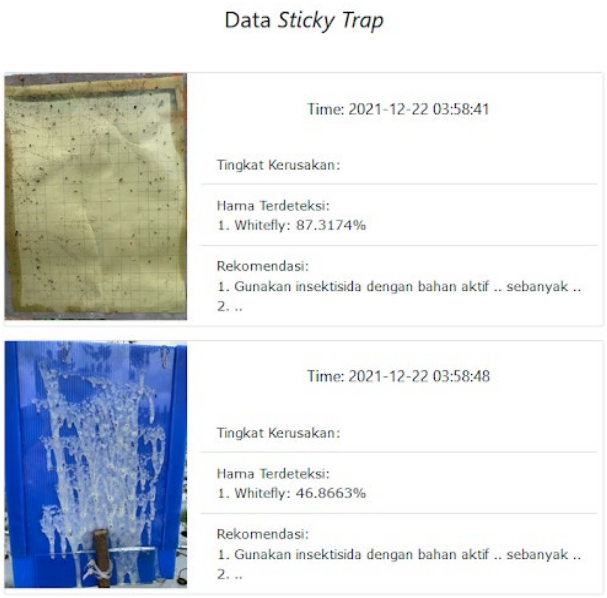
Gambar 13. Fitur login dan registrasi untuk akses *website*

Fitur **Penjelasan Hama** pada navigasi akan membawa pengguna ke laman penjelasan hama pada halaman Beranda. Untuk mengakses penjelasan hama yang diinginkan, pengguna dapat memilih tombol "*Selanjutnya*" yang berada di sisi bawah dari kartu penjelasan hama yang ingin diketahui. Halaman penjelasan hama ditunjukkan pada Gambar 14 dan berisi penjelasan singkat tentang hama, gejala pada tanaman yang terinfeksi hama tersebut, serta cara penanganan hamanya. Sementara itu, fitur **Data** pada navigasi dan tombol "*Coba Sistem*" pada komponen *header website* di bagian Beranda akan membawa pengguna ke data YST yang dikirimkan kamera serta hasil pengolahan citra yang diperoleh sebagaimana diilustrasikan pada Gambar 15.

(a) Penjelasan hama

(b) Metode penanganan hama

Gambar 14. Tampilan penjelasan dan penanganan hama



Gambar 15. Ilustrasi tampilan fitur **Data**

4.2.2 Analisis Basis Data

Website yang dirancang dilengkapi dengan basis data yang digunakan untuk menampung seluruh data hasil pengukuran setiap node sensor dan data seluruh pengguna *website*. Basis data tersebut diimplementasikan pada platform MySQL pada sink Raspberry Pi. Dalam perancangan basis data, sebuah Entity Relationship Diagram (ERD) yang bertujuan untuk menjelaskan hubungan antar objek entitas telah dirancang terlebih dahulu. ERD tersebut diperlukan untuk menyusun struktur data dan hubungan antar data dalam basis data yang digunakan [25].

Basis data yang dibangun pada penelitian ini terdiri dari tiga (3) entitas utama, yaitu:

- *Image*: berfungsi menampung data YST dari sensor yang belum diproses
- *Result*: berfungsi menampung data hasil pemrosesan citra YST
- *User*: berfungsi menampung data pengguna *website*

Gambar 16 menunjukkan desain fisik dari basis data yang dirancang. Terdapat dua buah relasi dalam dua tabel berbeda yang dirancang, yaitu relasi antara entitas *Result* dan *Image* dengan tipe relasi *one-to-one* serta relasi antara entitas *User* dan *Image* dengan tipe relasi *one-to-many*. Tipe *one-to-one* dipilih untuk relasi pertama karena proses yang dilakukan setelah gambar diperoleh hanya akan memiliki suatu hasil. Sedangkan tipe *one-to-many* dipilih untuk relasi kedua karena satu kamera yang terpasang pada lahan pertanian dapat melakukan beberapa kali pengambilan gambar.

Nama Tabel	Atribut	Length	Type
Result	idHasil	100	int
	idImage	100	int
	total	-	int
	damage	-	float
	whitefly	-	int
	waktu_upload	-	datetime
Image	idImage	-	int
	idUser	-	int
	image	300	str
	lokasi	100	str
	waktu_dibuat	-	datetime
	waktu_upload	-	datetime
User	idUser	-	int
	namaUser	100	string
	username	100	string
	alamat	100	string
	email_address	100	string
	password	100	string

Gambar 16. Desain fisik rancangan basis data

5 Kesimpulan

Makalah ini telah memaparkan rancangan sistem pemantauan hama GWF berbasis citra YST dengan menggunakan jaringan sensor nirkabel berbasis web yang dilengkapi dengan algoritma pengolahan citra dan dioperasikan secara *real-time*. Hasil pengujian menunjukkan bahwa sistem deteksi hama yang dirancang dapat berfungsi dengan baik dan tanpa kendala. Penelitian lanjutan akan ditujukan untuk meningkatkan performansi dan kinerja dari sistem yang dirancang, baik secara komponen individual maupun terintegrasi secara keseluruhan. Penelitian lanjutan akan ditujukan untuk meningkatkan performansi dan kinerja dari sistem yang dirancang, baik secara komponen individual maupun terintegrasi secara keseluruhan.

Ucapan Terima kasih

Tim penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM), Universitas Katolik Parahyangan serta P.T. Biops Agrotekno Indonesia atas bantuan serta dukungan yang diberikan dalam pelaksanaan kegiatan penelitian ini.

Referensi

[1] B. P. Statistik, *Hasil Survei Pertanian antar Sensus (SUTAS) 2018 Seri-A2*. [Online] Available: : https://www. bps. go. id/publication/2019/10/31/9567dfb 39bd984 aa45124b40/ hasil-survei-pertani an-antar-sensus--sutas--2018- seri-a2. html. [Accessed 17 July 2024]
[2] G. Idoje, T Dagiuklas, & M. Iqbal, "Survey for smart farming technologies: Challenges and issues," *Computers & Electrical Engineering*, vol. 92, pp. 107104, 2021.
[3] M. C. F. Lima, M. E. D. de Almeida Leandro, C. Valero, L. C. P. Coronel, & C. O. G. Bazzo, "Automatic detection and monitoring of insect pests—A review," *Agriculture*, vol. 10, no. 5, pp. 161, 2020.
[4] W. Li, D. Wang, M. Li, Y. Gao, J. Wu, & X. Yang, "Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse," *Computers and Electronics in Agriculture*, vol. 183, pp. 106048, 2021.
[5] R. M. Saleem, R. Kazmi, I. S. Bajwa, A. Ashraf, S. Ramzan, & W. Anwar, "IOT-based cotton whitefly prediction using deep learning," *Scientific Programming*, vol. 2021.1, pp. 8824601, 2021.
[6] S. Bhat & S. Kumar. "Conventional and recent approaches of integrated pest management in greenhouse cultivation." in *Protected Cultivation,* 1st ed, New York, Apple Academic Press, 2024, pp. 255-274. 2024.
[7] A. Nasruddin, J. Jumardi, & M. Melina, "Population dynamics of Trialeurodes vaporariorum (Westwood) (Hemiptera: Aleyrodidae) and its populations on different planting dates and host plant species," *Annals of Agricultural Sciences*, vol. 66, no. 2, pp. 109-114, 2021.
[8] H. Rehman, A. Bukero, A. G. Lanjar, L. Bashir, Z. Lanja, & S. A. Nahiyoon, " Comparison of different mechanical traps to screening and control of whitefly (Aleyrodidea: Hemiptera) population in tomato crop," *Pure and Applied Biology*, vol. 9, no. 4 pp. 2151-2157, 2020.
[9] E. Böckmann, A. Pfaff, M. Schirrmann, & M. Pflanz, "Rapid and low-cost insect detection for analysing species trapped on yellow sticky traps," *Scientific Reports*, vol. 11, no. 1, pp. 10419, 2021.
[10] M. K. G. Elsherbeni & Y. E. Afia, "Evaluation efficiency of sticky traps on attraction of greenhouse whitefly, Trialeurodes Vaporariorum (Westwood) infesting carnation flowers under glasshouse conditions," *Egyptian Academic Journal of Biological Sciences - A: Entomology*, vol. 14, no. 2, 2021.
[11] L.-Y. Chiu, D. J. A. Rustia, C.-Y. Lu, & T.-T. Lin, "Modelling and forecasting of greenhouse whitefly incidence using time-series and ARIMAX analysis," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 196-201, 2019.
[12] H. Rehman, A. Bukero, A.G. Lanjar, L. Bashir, Z. Lanja, & S. A. Nahiyoon, "Comparison of different mechanical traps to screening and control of whitefly (Aleyrodidea: Hemiptera) population in tomato crop," *Pure and Applied Biology*, vol. 9, no. 4, pp. 2151-2157, 2020.
[13] D. M. Pinto-Zevallos & I. Vänninen, "Yellow sticky traps for decision-making in whitefly management: What has been achieved?," *Crop Protection*, vol. 47, pp. 74-84, 2013.
[14] P. Sanjeevi, S. Prasanna, B. Kumar, G. Gunasekaran, I. Alagiri, & R. Anand, "Precision agriculture and farming using Internet of Things based on wireless sensor network," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12 pp. e3978, 2020.
[15] M. Mahbub, "A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network," *Internet of Things*, vol. 9, pp. 100161, 2020.
[16] M. S. BenSaleh, R. Saida, Y. H. Kacem, & M. Abid, "Wireless sensor network design methodologies: A survey," *Journal of Sensors*, vol. 2020, no. 1 pp. 9592836, 2020.
[17] W. Dargie & C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons, 2010.
[18] E. Setyawati, H. Wijoyo, & N. Soeharmoko. *Relational Database Management System* (*RDBMS*). CV Pena Persada, 2020.

[19] K. G. Dhal, A. Das, S. Ray, J. Gálvez, & S. Das, "Histogram equalization variants as optimization problems: a review," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1471-1496, 2021.
[20] K. Jha, A. Sakhare, N. Chavhan, & P. P. Lokulwar, "A review on image enhancement techniques using histogram equalization," *Grenze International Journal of Engineering & Technology*, vol. 10, no. 1, 2024.
[21] Y. Xie, L. Ning, M. Wang, & C. Li, "Image enhancement based on histogram equalization," *Journal of Physics: Conference Series*, vol. 1314, no. 1, pp. 012161, 2019.
[22] P. Musa, F. A. Rafi, & M. Lamsani, "A review: contrast-limited adaptive histogram equalization (CLAHE) methods to help the application of face recognition," *Proc. International Conference on Informatics & Computing*, pp. 1-6, 2018.
[23] M. Sezgin, Mehmet & B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146-168, 2004.
[24] R. Kamath, M. Balachandra, & S. Prabhu, "Raspberry Pi as visual sensor nodes in precision agriculture: A Study," *IEEE Access*, vol. 7, pp. 45110- 45122, 2019.
[25] Z, F. Azzahra & A. D. Anggoro, "Analisis teknik entity-relationship diagram dalam perancangan database sebuah literature review," *INTECH (Informatika dan Teknologi)*, vol. 3, no. 1, pp. 8-11, 2022.