

Perancangan Arsitektur Sistem Berbasis Iot Dan *Object Detection* Untuk Pemantauan Dan Klasifikasi Pertumbuhan Jamur Tiram (Studi Kasus Umkm Budidaya Jamur Tiram Barokah)

1st Yohanes Rico Wijaya
Information Systems
Telkom University
Bandung, Indonesia

rcwjys@student.telkomuniversity.ac.id

2nd Ekky Novriza Alam
Information Systems
Telkom University
Bandung, Indonesia

ekkynovrizalam@telkomuniversity.ac.id

3rd Sinung Suakanto
Information Systems
Telkom University
Bandung, Indonesia

sinung@telkomuniversity.ac.id

Budidaya Jamur Tiram Barokah sebagai UMKM di Indonesia menghadapi tantangan dalam pemantauan suhu, kelembapan, dan klasifikasi fase pertumbuhan jamur tiram yang masih dilakukan secara manual, sehingga berisiko menimbulkan ketidakefisienan dan keterbatasan data historis. Penelitian ini bertujuan untuk merancang arsitektur sistem pemantauan dan klasifikasi fase pertumbuhan jamur tiram berbasis *Internet of Things* (IoT) dan *object detection*, menggunakan pendekatan *Waterfall* dan kerangka 4+1 *View* Model. Sistem dikembangkan untuk memantau suhu dan kelembapan kumbung secara *real-time* serta mengklasifikasikan fase pertumbuhan jamur tiram berbasis citra visual. Evaluasi dilakukan dengan mengukur *Mean Time To Recovery* (MTTR), *Mean Time Between Failures* (MTBF), dan *availability*. Hasil pengujian menunjukkan bahwa modul suhu dan kelembapan memiliki MTTR sebesar 31,2 detik dan MTBF sebesar 23,07 jam, sedangkan modul kamera memiliki MTTR sebesar 27,3 detik dan MTBF tak terhingga, dengan nilai *availability* masing-masing sebesar 99,96% dan 100%. Temuan ini menunjukkan bahwa arsitektur sistem yang dirancang mampu memastikan reliabilitas dan ketersediaan tinggi, sehingga mendukung praktik budidaya jamur tiram yang lebih efisien dan berkelanjutan pada skala UMKM. Penelitian ini berkontribusi pada pengembangan arsitektur sistem informasi berbasis IoT di sektor agrikultur.

Kata kunci — Jamur tiram, sistem pemantauan, klasifikasi visual, IoT, arsitektur perangkat lunak.

I. PENDAHULUAN

Budidaya Jamur Tiram Barokah merupakan sebuah Usaha Mikro, Kecil, dan Menengah (UMKM) yang bergerak di bidang industri pertanian jamur serta pengolahan makanan berbasis jamur tiram. Berdasarkan hasil observasi dan wawancara yang telah dilakukan pada divisi pemeliharaan, ditemukan berbagai tantangan yang berpotensi menghambat efektivitas operasional budidaya. tantangan utama yang dihadapi adalah kesulitan dalam memastikan suhu dan kelembapan di dalam kumbung secara akurat dan *real-time*. Selama ini, proses pemantauan suhu dan kelembapan masih dilakukan secara manual dengan frekuensi pengecekan yang

terbatas. Meskipun metode ini dapat memberikan gambaran umum mengenai kondisi lingkungan, metode tersebut belum mampu mendeteksi fluktuasi suhu dan kelembapan yang terjadi dalam interval waktu pendek secara akurat dan menyeluruh. Hal ini didukung oleh temuan Zainuddin et al. [1] yang menyatakan bahwa pemantauan dan pengendalian secara manual dapat menimbulkan ketidakefisienan serta inkonsistensi. Kondisi ini mengakibatkan tindakan preventif dan perawatan lingkungan budidaya sering kali kurang tepat sasaran, sehingga meningkatkan risiko terjadinya kegagalan panen akibat ketidaksesuaian parameter lingkungan dengan kebutuhan optimal pertumbuhan jamur tiram. Hal tersebut juga didukung oleh Desnanjaya dan Sugiartawan [2] yang menyatakan bahwa pertumbuhan jamur tiram sangat dipengaruhi oleh faktor suhu dan kelembapan.

Selain itu, permasalahan lainnya adalah pemantauan dilakukan secara manual melalui observasi visual tanpa adanya pencatatan yang terstruktur maupun bukti citra digital. Kondisi tersebut menyebabkan kurangnya data historis yang dapat digunakan untuk mengevaluasi pola pertumbuhan serta melakukan prediksi waktu panen secara lebih akurat. Permasalahan ini menjadi semakin kompleks ketika frekuensi pemantauan yang dilakukan tidak sesuai dengan kebutuhan optimal proses budidaya. Akibatnya, terdapat risiko kesalahan dalam klasifikasi fase pertumbuhan jamur tiram, sehingga penentuan waktu panen cenderung bersifat subjektif dan hanya didasarkan pada pengalaman semata. Situasi ini berpotensi menimbulkan kerugian ekonomi yang signifikan karena kualitas panen yang dihasilkan tidak dapat dimaksimalkan sesuai dengan potensi optimalnya.

Oleh karena itu, diperlukan solusi untuk membantu divisi pemeliharaan dalam mengelola kondisi lingkungan kumbung secara lebih efektif, sehingga dapat mendukung peningkatan produktivitas dan kualitas hasil panen jamur tiram secara berkelanjutan. Solusi yang dibutuhkan tidak

hanya terbatas pada kemampuan untuk memantau suhu dan kelembapan secara akurat, melainkan juga mampu menyediakan data historis yang terstruktur serta mudah diakses untuk mendukung proses analisis dan evaluasi.

Sebagai solusi atas permasalahan tersebut, diperlukan pendekatan yang lebih modern untuk mengatasinya, salah satunya melalui penerapan pertanian pintar (*smart farming*). Pendekatan ini memungkinkan proses pemantauan dan pengelolaan lingkungan budidaya dilakukan secara lebih efisien dengan memanfaatkan teknologi informasi, sehingga dapat menghemat sumber daya serta meningkatkan efisiensi dan hasil pertanian melalui pemantauan dan pengelolaan jarak jauh [3], [4]. Selain itu, Inayah et al. [3] juga menyatakan bahwa penerapan *smart farming* memungkinkan proses pemantauan dilakukan secara *real-time* tanpa memerlukan pengecekan langsung ke lokasi budidaya. Dengan demikian, penerapan konsep ini berpotensi untuk mengoptimalkan produktivitas dan efektivitas operasional budidaya jamur tiram secara berkelanjutan.

Untuk mendukung implementasi *smart farming* secara optimal, diperlukan sistem perangkat lunak yang tidak hanya fungsional, tetapi juga memiliki tingkat keandalan (*reliability*) dan ketersediaan (*availability*) yang tinggi. Kompleksitas sistem modern menuntut adanya pendekatan arsitektur perangkat lunak yang mampu mengelola berbagai kebutuhan fungsional dan non-fungsional secara sistematis [5]. Studi sebelumnya menyatakan bahwa arsitektur perangkat lunak yang dirancang secara tepat mampu meningkatkan kualitas sistem dari aspek performa, efisiensi, keamanan, hingga keandalan operasional [6]. Dalam konteks ini, *reliability* mencerminkan kemampuan sistem untuk beroperasi secara konsisten tanpa gangguan dalam periode tertentu [7], sementara *availability* mengindikasikan probabilitas sistem untuk tetap dapat diakses selama jam operasional [8].

Namun demikian, masih terdapat kesenjangan dalam literatur terkait bagaimana rancangan arsitektur sistem berkontribusi langsung terhadap pencapaian *reliability* dan *availability*, khususnya dalam konteks sistem pemantauan berbasis IoT yang diterapkan pada sektor pertanian jamur tiram. Kebutuhan akan sistem yang mampu beroperasi secara konsisten dan selalu tersedia menjadi semakin krusial, mengingat dinamika parameter lingkungan yang memengaruhi keberhasilan produksi jamur tiram. Oleh karena itu, penelitian ini difokuskan untuk mengeksplorasi perancangan arsitektur sistem pemantauan yang tidak hanya mampu memenuhi tuntutan fungsional, tetapi juga menjamin keandalan dan ketersediaan layanan secara berkelanjutan. Pendekatan ini diharapkan dapat memberikan kontribusi nyata terhadap pengembangan sistem pertanian cerdas (*smart farming*) yang efisien, adaptif, dan aplikatif pada skala usaha mikro, kecil, dan menengah (UMKM).

II. KAJIAN TEORI

A. Arsitektur Perangkat Lunak (Software Architecture)

Arsitektur perangkat lunak merupakan representasi konseptual yang berfungsi sebagai cetak biru (*blueprint*)

dalam proses pengembangan sistem perangkat lunak. Menurut Richards dan Ford [9], menyatakan bahwa arsitektur perangkat lunak adalah peta jalan yang memandu proses pengembangan sistem secara menyeluruh. Hal tersebut diperkuat dengan Cervantes dan Kazman [10], yang mendefinisikan arsitektur perangkat lunak sebagai suatu sistem yang terdiri dari kumpulan struktur untuk merepresentasikan sistem, mencakup elemen perangkat lunak, hubungan antar elemen, serta properti masing-masing elemen. Peran penting arsitektur tidak hanya mencakup aspek desain, tetapi juga memengaruhi kualitas perangkat lunak secara keseluruhan, termasuk performa, keandalan, pemeliharaan, dan keberhasilan implementasi [5].

Prinsip-prinsip utama dalam arsitektur perangkat lunak berfokus pada pencapaian sistem yang efisien, aman, serta mudah dikembangkan dan dipelihara. Aspek performa, keamanan, dan keterkelolaan menjadi pilar utama dalam merancang arsitektur sistem [11]. Di samping itu, aspek non-fungsional seperti keandalan, skalabilitas, portabilitas, dan ketersediaan juga harus diperhitungkan. Pandangan ini diperkuat oleh Kruchten [12], yang menyatakan bahwa pemenuhan kebutuhan fungsional dan non-fungsional secara bersamaan merupakan landasan dalam merancang arsitektur yang tangguh dan adaptif. Dengan demikian, desain arsitektur perangkat lunak yang efektif harus mampu menjawab tantangan kompleksitas sistem dan mendukung integrasi komponen secara terpadu.

B. Waterfall

Metode *Waterfall* merupakan salah satu pendekatan klasik dalam pengembangan perangkat lunak yang bersifat sekuensial dan sistematis. Metode ini sangat sesuai untuk proyek berskala kecil hingga menengah karena alur kerjanya yang sederhana, mudah dipahami, dan mudah dipelihara [13]. Model ini terdiri dari enam fase utama yang berlangsung secara berurutan: *Requirement*, *Design*, *Development*, *Testing*, *Deployment*, dan *Maintenance*.

Fase *Requirement* berperan sebagai tahap awal yang mendefinisikan kebutuhan fungsional dan non-fungsional melalui analisis sistem dan bisnis. Tahap ini menghasilkan dokumen *Software Requirements Specification* (SRS) sebagai dasar pengembangan. Selanjutnya, fase *Design* bertujuan untuk menyusun solusi teknis yang mencakup perancangan algoritma, arsitektur sistem, antarmuka pengguna, serta struktur basis data.

Setelah desain sistem dirancang, fase *Development* melibatkan proses implementasi dalam bentuk kode program berdasarkan spesifikasi yang telah ditetapkan. Kemudian, fase *Testing* dilakukan untuk memastikan sistem berfungsi sesuai dengan kebutuhan melalui proses verifikasi, validasi, dan debugging. Setelah sistem bebas dari kesalahan, fase *Deployment* bertugas untuk mendistribusikan solusi kepada pengguna. Fase terakhir, *Maintenance*, mencakup pemeliharaan sistem agar tetap relevan terhadap kebutuhan pengguna, memperbaiki *bug*, serta meningkatkan kualitas dan kinerja perangkat lunak [14].

C. 4 + 1 View Model

Model 4+1 *View* merupakan pendekatan arsitektur perangkat lunak yang dikembangkan oleh Kruchten, bertujuan untuk memberikan representasi komprehensif terhadap sistem melalui lima pandangan utama: *Logical View*, *Process View*, *Development View*, *Physical View*, dan *Scenario View*. Keempat pandangan pertama menangani aspek teknis sistem, sedangkan *Scenario View* menjembatani kebutuhan pengguna melalui ilustrasi *use case* [12].

Logical View menekankan aspek kebutuhan fungsional, mengorganisasi struktur sistem berdasarkan interaksi antar elemen utama. *Process View* berfokus pada aspek non-fungsional, seperti performa, konkurensi, distribusi, dan toleransi kesalahan. Arsitektur ini memungkinkan partisi perangkat lunak ke dalam tasks yang dapat dikendalikan secara mandiri dan dikomunikasikan melalui berbagai metode seperti *remote procedure calls* dan *message-based services*.

Sementara itu, *Development View* menjelaskan struktur pengembangan perangkat lunak dalam bentuk modul dan subsistem, serta mendukung manajemen pengembangan, pembagian kerja tim, dan pengelolaan sumber daya. *Physical View*, di sisi lain, memetakan perangkat lunak ke dalam infrastruktur fisik seperti nodes dan jaringan, serta memperhatikan aspek *availability*, *reliability*, dan *scalability*. Akhirnya, *Scenario View* menggambarkan interaksi sistem dalam konteks fungsional menggunakan skenario untuk memvalidasi kesesuaian rancangan arsitektur.

D. Mean Time To Recover (MTTR)

MTTR merupakan parameter yang menunjukkan waktu rata-rata yang dibutuhkan sistem untuk pulih dari kegagalan. MTTR digunakan untuk mengukur efektivitas respons sistem terhadap insiden, dari awal gangguan hingga kembali normal [15]. Mollajan dan Iranmanesh [16], menegaskan bahwa MTTR menggambarkan kecepatan sistem dalam memulihkan fungsinya.

Nilai MTTR dihitung menggunakan rumus:

$$MTTR = \frac{\text{Total Time Of Down Time}}{\text{Number Of Incident}} \quad (1)$$

Parameter ini penting dalam evaluasi performa sistem, khususnya dalam konteks pertanian berbasis IoT, di mana kontinuitas layanan menjadi faktor utama dalam keberhasilan implementasi teknologi secara nyata.

E. Mean Time Between Failure (MTBF)

MTBF adalah indikator penting dalam menilai keandalan sistem yang dapat diperbaiki. MTBF dapat menggambarkan rata-rata waktu antar kegagalan dalam suatu sistem. Meskipun memiliki kesamaan dengan MTTF (*Mean Time To Failure*), MTBF relevan digunakan untuk komponen repairable, sedangkan MTTF untuk *non-repairable* [17].

Dalam konteks sistem yang menyediakan layanan fungsional seperti modul pemantauan atau kamera visual, MTBF mengukur sejauh mana sistem dapat beroperasi secara

konsisten sebelum mengalami gangguan. Nilainya dihitung dengan:

$$MTBF = \frac{\text{Total Operational Hours}}{\text{Number of Failures}} \quad (2)$$

Dengan memahami MTBF, pengembang sistem dapat mengevaluasi frekuensi kegagalan serta kebutuhan pemulihan yang dibutuhkan untuk menjaga stabilitas layanan.

F. Availability

Availability merujuk pada tingkat kesiapan sistem dalam menyediakan layanan yang berfungsi dengan benar pada waktu tertentu. *Availability* merepresentasikan probabilitas sistem untuk berada dalam kondisi operasional sebagaimana mestinya [18]. Pendekatan ini menekankan pentingnya kesinambungan layanan dalam sistem yang bergantung pada konektivitas dan waktu nyata.

Untuk mengukur *availability* secara kuantitatif, digunakan rumus:

$$\text{Availability} = \frac{MTBF}{MTBF + MTTR} \quad (3)$$

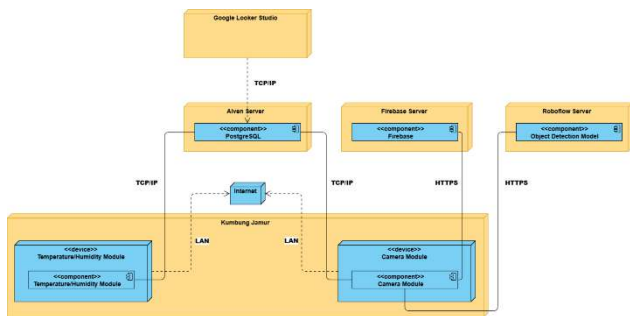
Rumus ini didukung oleh beberapa penelitian [19], [20], yang menegaskan akurasi pendekatan tersebut dalam menilai ketersediaan sistem. Tingginya tingkat *availability* sangat penting dalam sistem pemantauan berbasis IoT yang harus bekerja secara real-time dan tanpa gangguan.

III. METODE

Penelitian ini menggunakan pendekatan eksperimental untuk mengembangkan dan mengevaluasi sistem pemantauan budidaya jamur tiram berbasis *Internet of Things* (IoT) dan teknologi deteksi objek. Fokus utama dari metode ini adalah merancang arsitektur perangkat lunak sistem pemantauan, melakukan instalasi perangkat keras IoT di lapangan, serta menguji *reliability* dan *availability* sistem melalui parameter MTTR, MTBF, dan *Availability*.

A. Perancangan Arsitektur Sistem Pemantauan

Langkah awal dalam penelitian ini adalah merancang arsitektur sistem pemantauan yang mengintegrasikan sensor IoT dan modul klasifikasi visual dengan menggunakan pendekatan 4+1 *View* Model, yang terdiri atas *Logical View*, *Development View*, *Process View*, *Physical View*, dan *Scenario View*. Identifikasi kebutuhan sistem dilakukan melalui wawancara dengan perwakilan divisi pemeliharaan yang memiliki pengalaman selama lima tahun dalam budidaya jamur tiram, serta melalui observasi lapangan secara langsung. Pendekatan ini dilakukan untuk memastikan bahwa rancangan sistem disusun secara kontekstual, relevan, dan sesuai dengan kondisi aktual di lapangan, sehingga hasil pengembangan dapat menjawab kebutuhan operasional secara efektif dan aplikatif.



GAMBAR 1
Arsitektur Sistem Pemantauan

Pada Gambar. 1. struktur sistem dirancang secara modular, mencakup penggunaan sensor suhu dan kelembapan (DHT11), kamera Raspberry Pi Cam 3 untuk pengambilan citra visual, serta mikrokontroler Raspberry Pi Zero 2W sebagai unit pengolah utama. Data dari sensor dikirim otomatis setiap sepuluh menit ke PostgreSQL melalui jaringan *Local Area Network* (LAN), sedangkan citra dikirim harian untuk diproses oleh model deteksi objek berbasis melalui *Application Programming Interface* (API) Roboflow. Hasil klasifikasi disimpan kembali dan divisualisasikan secara *real-time* melalui *dashboard digital twin* menggunakan platform Google Looker Studio.

B. Instalasi Perangkat IoT

Setelah tahapan perancangan sistem diselesaikan, proses berikutnya adalah mengimplementasikan instalasi perangkat keras IoT pada lingkungan nyata, yakni unit budidaya jamur tiram yang divisualisasikan Pada Gambar 2.



GAMBAR 2
Instalasi Modul Suhu Kelembapan

Instalasi dilakukan di dalam kumbung jamur yang terdiri atas tiga rak utama yang berisi baglog. Sensor DHT11 ditempatkan pada bagian tengah atas kumbung berdasarkan hasil uji awal yang menunjukkan distribusi suhu dan kelembapan yang relatif merata di berbagai titik, sehingga penggunaan satu sensor DHT11 dianggap memadai untuk merepresentasikan kondisi dalam satu kumbung.



GAMBAR 3
Instalasi Modul Kamera

Dalam rangka mendukung proses klasifikasi fase pertumbuhan jamur, kamera Raspberry Pi Cam 3 diposisikan di sisi rak dengan sudut pandang yang diarahkan ke rak di depannya. Penempatan ini dirancang sedemikian rupa agar memungkinkan pengambilan citra visual dari sejumlah besar baglog secara simultan, sehingga menghasilkan sampel gambar yang representatif, sebagaimana ditunjukkan pada Gambar 3. Citra yang dihasilkan selanjutnya dikirim secara otomatis ke sistem penyimpanan berbasis *cloud* untuk kemudian diproses menggunakan *platform* Roboflow. Melalui pemanfaatan model deteksi objek, fase pertumbuhan jamur dapat diidentifikasi secara otomatis. Informasi hasil klasifikasi tersebut kemudian disimpan ke dalam basis data PostgreSQL guna mendukung integrasi data yang terstruktur dan mempermudah akses untuk analisis lanjutan.

C. Pengujian *Reliability* dan *Availability* Sistem

Setelah sistem berhasil dioperasikan secara menyeluruh, dilakukan proses evaluasi untuk mengukur tingkat keandalan dan ketersediaan sistem melalui tiga parameter utama, yaitu MTTR, MTBF, dan *availability*. Evaluasi ini dilaksanakan dengan mengoperasikan sistem secara kontinu tanpa intervensi manual selama periode tertentu untuk mensimulasikan kondisi operasional yang mendekati situasi nyata di lapangan. Selama pengamatan, setiap kejadian kegagalan dicatat secara manual dengan merujuk pada *log* aktivitas sistem yang mencatat waktu terjadinya gangguan serta durasi yang dibutuhkan untuk memulihkan fungsionalitas sistem. Dalam hal ini, MTTR digunakan untuk menghitung rata rata waktu pemulihan sistem dari kondisi tidak operasional hingga kembali berfungsi secara normal, sebagaimana ditunjukkan dalam (1). Selanjutnya, parameter MTBF digunakan untuk mengukur rata rata interval waktu antar dua kegagalan yang berurutan, dengan perhitungan yang didasarkan pada total durasi operasional sistem dibagi jumlah kegagalan yang terjadi selama masa evaluasi, sebagaimana dirumuskan dalam (2). Pengujian terhadap MTBF dilakukan selama kurang lebih 25 hari atau setara dengan 599,9980555555555 jam waktu operasional sistem secara akumulatif.

Berdasarkan nilai MTTR dan MTBF yang telah diperoleh, dilakukan penghitungan terhadap *availability* sistem, yang mengacu pada probabilitas sistem untuk tetap berada dalam kondisi operasional pada waktu tertentu. Perhitungan ini dilakukan menggunakan (3), yang mengintegrasikan dua parameter sebelumnya untuk memberikan gambaran menyeluruh mengenai kemampuan sistem dalam mempertahankan ketersediaannya selama beroperasi.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian MTTR dan MTBF

Hasil pengujian MTTR terhadap modul pemantauan suhu dan kelembapan menunjukkan bahwa sistem mampu melakukan pemulihan dari kondisi gangguan dengan rata rata waktu sebesar 31,2 detik. Nilai tersebut diperoleh melalui sepuluh kali simulasi pemutusan aliran daya listrik yang dilakukan secara berulang untuk merepresentasikan skenario gangguan yang mungkin terjadi dalam kondisi operasional. Rincian hasil pengujian ini disajikan pada Tabel 1.

TABEL 1
Hasil Pengujian MTTR Modul Suhu Kelembapan

Percobaan Ke	Modul Suhu Kelembapan	Modul Kamera
	Selisih (Detik)	Selisih (detik)
1	36	39
2	25	18
3	19	12
4	20	29
5	39	23
6	30	47
7	38	31
8	33	22
9	22	21
10	50	31
Mean	31,2	27,3

Pada pengujian terhadap modul kamera, nilai MTTR tercatat sebesar 27,3 detik, yang menunjukkan kemampuan pemulihan yang lebih baik dibandingkan dengan modul pemantauan suhu dan kelembapan, sebagaimana ditunjukkan pada Tabel 1. Nilai tersebut diperoleh dari akumulasi total waktu pemulihan sebesar 273 detik yang dihasilkan melalui sepuluh kali simulasi pemutusan aliran daya.

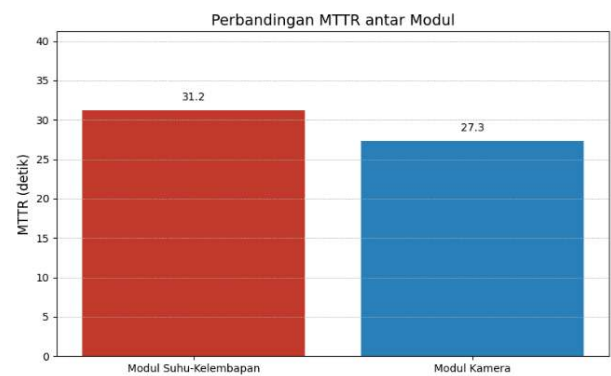
Selanjutnya, untuk pengujian MTBF, sistem suhu dan kelembapan diuji selama total 599,998 jam, dengan mencatat 26 kejadian kegagalan. Berdasarkan persamaan (2), diperoleh nilai MTBF sebesar 23,07 jam sebagaimana ditunjukkan pada Tabel 2.

TABEL 2
Hasil Pengujian MTBF Modul Suhu Kelembapan

Modul Suhu Kelembapan		Modul Kamera	
Hasil yang Diharapkan	Hasil Aktual	Hasil yang Diharapkan	Hasil Aktual
3600	3574	25	25

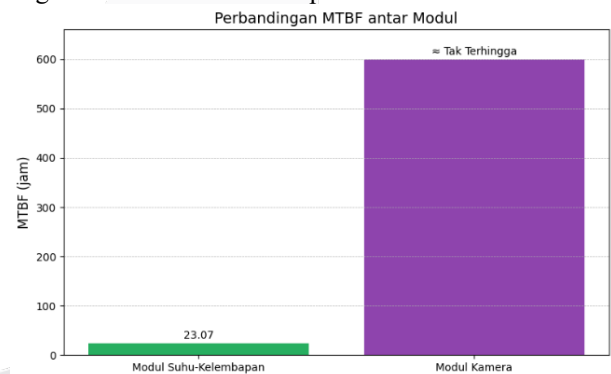
Untuk MTBF pada modul kamera, selama periode pengujian 599,998 jam, tidak ditemukan satu pun insiden kegagalan dalam penyimpanan data citra visual. Hal ini menunjukkan bahwa nilai MTBF kamera mendekati tak terhingga.

Berdasarkan hasil pengujian yang telah dilakukan, nilai MTTR yang dicapai oleh kedua modul menunjukkan bahwa sistem memiliki kemampuan pemulihan yang cukup cepat terhadap gangguan operasional. Rata rata waktu pemulihan yang dicapai, yaitu 31,2 detik pada modul suhu dan kelembapan serta 27,3 detik pada modul kamera, mengindikasikan bahwa mekanisme otomatisasi sistem dalam menangani gangguan telah berfungsi dengan baik, sebagaimana yang ditunjukkan pada Gambar 4 berdasarkan hasil perhitungan pada Tabel 1.



GAMBAR 4
Komparasi Hasil Pengujian MTTR

Di sisi lain, nilai MTBF sebesar 23,07 jam pada modul pemantauan suhu dan kelembapan menunjukkan bahwa sistem memiliki kemampuan untuk beroperasi dalam jangka waktu yang cukup panjang sebelum mengalami gangguan. Hasil ini mengindikasikan bahwa performa modul suhu dan kelembapan telah berada pada kategori baik serta mampu mendukung kebutuhan pemantauan harian secara efektif. Adapun pada modul kamera, tidak ditemukan satu pun insiden kegagalan selama periode pengujian berlangsung. Kondisi ini mencerminkan tingkat kestabilan operasional yang sangat tinggi dan memperkuat keyakinan bahwa modul tersebut memiliki karakteristik yang andal dalam mendukung kontinuitas sistem pemantauan secara berkelanjutan, sebagaimana divisualisasikan pada Gambar 5.



GAMBAR 5
Komparasi Hasil Pengujian MTBF

Secara keseluruhan, hasil pengujian ini memberikan gambaran positif terhadap performa sistem yang telah dikembangkan, sekaligus menjadi dasar yang kuat untuk langkah pengembangan berikutnya dalam rangka meningkatkan keandalan dan efisiensi pada implementasi di lingkungan nyata.

B. Hasil Pengujian Availability

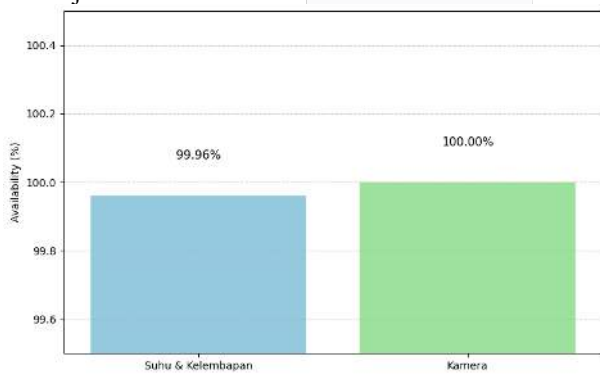
Berdasarkan hasil pengujian terhadap parameter *reliability* dan *availability*, diperoleh MTBF sebesar 23,0768 jam pada modul suhu dan kelembapan. Sementara itu, waktu pemulihan rata rata pada modul ini tercatat sebesar 0,0075833 jam, atau setara dengan sekitar 27 detik. Adapun pada modul kamera, hasil pengujian menunjukkan tidak adanya insiden kegagalan selama durasi pengamatan, yang secara teoritis menghasilkan nilai MTBF mendekati tak terhingga. Dengan nilai MTTR sebesar 0,0075833 jam dan tanpa adanya gangguan, tingkat *availability* pada modul

kamera tercatat mencapai nilai maksimum sebesar 1. Informasi lengkap mengenai hasil pengukuran ini disajikan pada Tabel 3

TABEL 3
Hasil Pengujian *Availability*

Modul Suhu Kelembapan		Modul Kamera	
MTTR (jam)	MTBF (jam)	MTTR (jam)	MTBF (jam)
0,00866666667	23,0768	0,0075833	∞
Availability: 0,9996		Availability: 1	

Hasil pengujian terhadap parameter *availability* menunjukkan bahwa modul suhu dan kelembapan memiliki tingkat ketersediaan sebesar 0,9996 atau setara dengan 99,96 persen. Nilai ini diperoleh melalui perhitungan yang menggabungkan rata-rata waktu antar kegagalan dengan waktu pemulihan, sehingga mencerminkan kemampuan sistem dalam mempertahankan keberlangsungan operasional hampir sepenuhnya selama periode pengamatan. Tingginya tingkat ketersediaan tersebut mengindikasikan bahwa modul suhu dan kelembapan memiliki performa yang dapat diandalkan untuk mendukung operasional sistem pemantauan dalam skala menengah hingga jangka panjang. Selain itu, nilai *availability* yang mendekati sempurna ini juga menunjukkan bahwa interupsi layanan yang terjadi selama proses operasional berada pada tingkat yang sangat minimal, sehingga tidak berdampak signifikan terhadap kontinuitas fungsi pemantauan. Dengan performa seperti ini, sistem dapat memberikan kepercayaan lebih kepada pengguna dalam hal konsistensi kinerja, terutama pada konteks pemantauan lingkungan budidaya jamur tiram yang memerlukan pembaruan data secara real-time dan berkelanjutan.



GAMBAR 6
Komparasi Hasil Pengujian *Availability*

Sementara itu, modul kamera menunjukkan performa yang lebih stabil selama pengujian, dengan nilai *availability* yang tercatat sebesar 1. Hasil ini diperoleh dari tidak ditemukannya satu pun insiden kegagalan selama periode pengamatan, yang dikombinasikan dengan waktu pemulihan yang singkat. Kondisi ini mencerminkan bahwa modul kamera mampu mempertahankan operasionalnya secara konsisten tanpa gangguan, sehingga menjadi komponen yang sangat potensial untuk digunakan dalam sistem pemantauan yang membutuhkan keandalan tinggi. Perbandingan *availability* antara kedua modul, sebagaimana divisualisasikan pada Gambar 6, memberikan gambaran menyeluruh mengenai efektivitas masing-masing komponen dalam menjaga kontinuitas sistem.

V. KESIMPULAN

Arsitektur sistem pemantauan dan klasifikasi jamur tiram dikembangkan menggunakan pendekatan metode *Waterfall* yang dikombinasikan dengan kerangka *4+1 View Model*. Pendekatan ini diterapkan untuk memastikan bahwa kebutuhan pengguna yang telah didefinisikan secara rinci dapat dipenuhi secara menyeluruh dan sistematis. Arsitektur yang dihasilkan mampu mengakomodasi pengumpulan dan pemrosesan data secara terjadwal, sehingga memberikan dukungan operasional yang lebih efektif dalam kegiatan budidaya jamur tiram Barokah. Berdasarkan hasil pengujian MTTR, modul suhu dan kelembapan menunjukkan waktu pemulihan rata-rata sebesar 31,2 detik, sedangkan modul kamera mencatat waktu pemulihan rata-rata yang lebih cepat, yaitu sebesar 27,3 detik. Sementara itu, pengujian MTBF pada modul suhu dan kelembapan menunjukkan nilai sebesar 23,07 jam. Hasil ini menunjukkan bahwa performa modul suhu dan kelembapan telah berada pada kategori baik dan mendukung kebutuhan pemantauan harian secara efektif. Adapun pada modul kamera, tidak ditemukan satu pun kegagalan selama periode pengujian, yang menunjukkan tingkat kestabilan operasional yang sangat tinggi dan memperkuat keandalannya dalam mendukung kontinuitas sistem pemantauan secara berkelanjutan. Dari sisi *availability*, hasil perhitungan menunjukkan bahwa modul suhu dan kelembapan memiliki tingkat ketersediaan sebesar 99,96%, sedangkan modul kamera mencapai tingkat ketersediaan sempurna sebesar 100%. Nilai-nilai ini menunjukkan bahwa kedua modul mampu menjaga kontinuitas layanan selama masa pengujian dengan sangat baik, meskipun terdapat perbedaan karakteristik performa antar modul. Tingkat *availability* yang tinggi, dikombinasikan dengan hasil MTTR dan MTBF, memperkuat kesimpulan bahwa sistem yang telah diimplementasikan memiliki performa yang andal dan layak untuk diterapkan dalam konteks pertanian berbasis teknologi. Dengan demikian, dapat disimpulkan bahwa arsitektur sistem yang dirancang dan diuji dalam penelitian ini telah menunjukkan kinerja yang menjanjikan, khususnya dari aspek keandalan dan ketersediaan layanan. Meskipun masih terdapat ruang untuk optimalisasi pada modul suhu dan kelembapan agar mendukung penggunaan jangka panjang secara lebih optimal, hasil pengujian secara keseluruhan membuktikan bahwa sistem ini mampu beroperasi dengan stabil dan mendukung kegiatan pemantauan jamur tiram secara efisien dan berkelanjutan.

REFERENSI

- [1] N. A. Zainuddin, K. M. Nordin, A. Ishak, M. S. Nuriman, and T. I. D. T. Azhar, "IoT-Based Oyster Mushroom Farming Monitoring System," *International Journal for Multidisciplinary Research (IJFMR)*, 2021.
- [2] I. G. M. N. Desnanjaya and P. Sugiartawan, "Controlling and Monitoring of Temperature and Humidity of Oyster Mushrooms in Tropical Climates," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 12, no. 1, pp. 69–80, 2022.
- [3] I. Inayah, B. Agustirandi, M. Budiman, M. Djamal, and A. Faizal, "Experimental design: Implementation of IoT-based drip irrigation to enhance the

- productivity of Cilembu sweet potato (*Ipomoea batatas*) cultivation,” *Results in Engineering*, vol. 25, p. 103600, 2025, doi: <https://doi.org/10.1016/j.rineng.2024.103600>.
- [4] A. Elewi, A. Hajhamed, R. Khankan, S. Duman, A. Souag, and A. Ahmed, “Design and implementation of a cost-aware and smart oyster mushroom cultivation system,” *Smart Agricultural Technology*, vol. 8, p. 100439, 2024, doi: <https://doi.org/10.1016/j.atech.2024.100439>.
- [5] M. Jaiswal, “Software Architecture and Software Design,” *SSRN Electronic Journal*, Jan. 2019, doi: 10.2139/ssrn.3772387.
- [6] A. Mishra, Y. I. Alzoubi, and N. Gavrilovic, “Quality attributes of software architecture in IoT-based agricultural systems,” *Smart Agricultural Technology*, vol. 8, p. 100523, 2024, doi: <https://doi.org/10.1016/j.atech.2024.100523>.
- [7] C. Rivera and A. Martinez, “Enhancing Reliability Through Effective System Monitoring,” *Science and Technology*, vol. 8, Jul. 2024.
- [8] T. Pathirana and G. Nencioni, “Availability Model of a 5G-MEC System,” in *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*, 2023, pp. 1–10. doi: 10.1109/ICCCN58024.2023.10230201.
- [9] M. Richards and N. Ford, *Fundamentals of software architecture: an engineering approach*. O’Reilly Media, 2020.
- [10] H. Cervantes and R. Kazman, *Designing software architectures: a practical approach*. Addison-Wesley Professional, 2024.
- [11] Y. Kibru, N. Ayele, T. Brihun, and H. Kefie, “The Principle of Architecture First in Software Project Management Minimizes the Cost of Software Development Process: A Review,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 10, Nov. 2020, doi: 10.35940/ijitee.A8154.1110120.
- [12] P. Kruchten, “The 4+1 View Model of Architecture,” *IEEE Softw*, vol. 12, pp. 45–50, Nov. 1995, doi: 10.1109/52.469759.
- [13] A. K. Wardana, M. W. Sari, and K. Kusumaningtyas, “Developing martial art championship scheduling system using the waterfall model,” in *AIP Conference Proceedings*, 2023. doi: 10.1063/5.0105697.
- [14] U. S. Senarath, “Waterfall methodology, prototyping and agile development,” *Tech. Rep.*, pp. 1–16, 2021.
- [15] F. Auer, V. Lenarduzzi, M. Felderer, and D. Taibi, “From monolithic systems to Microservices: An assessment framework,” *Inf Softw Technol*, vol. 137, p. 106600, 2021, doi: <https://doi.org/10.1016/j.infsof.2021.106600>.
- [16] A. Mollajan and S. H. and Iranmanesh, “Modularisation of system architecture to improve system recoverability: a unique application of design structure matrix,” *Journal of Engineering Design*, vol. 32, no. 12, pp. 703–750, Dec. 2021, doi: 10.1080/09544828.2021.1971634.
- [17] M. Krasich, “How to estimate and use MTTF/MTBF would the real MTBF please stand up?,” in *2009 Annual Reliability and Maintainability Symposium*, 2009, pp. 353–359. doi: 10.1109/RAMS.2009.4914702.
- [18] V. Cortellessa, R. Eramo, and M. Tucci, “From software architecture to analysis models and back: Model-driven refactoring aimed at availability improvement,” *Inf Softw Technol*, vol. 127, p. 106362, 2020, doi: <https://doi.org/10.1016/j.infsof.2020.106362>.
- [19] K. Diatte, B. O’Halloran, and D. L. Van Bossuyt, “The Integration of Reliability, Availability, and Maintainability into Model-Based Systems Engineering,” *Systems*, vol. 10, no. 4, 2022, doi: 10.3390/systems10040101.
- [20] M. Danjuma, B. Yusuf, and I. Yusuf, “Reliability, Availability, Maintainability, and Dependability Analysis of Cold Standby Series-Parallel System,” *Journal of Computational and Cognitive Engineering*, vol. 1, Jun. 2022, doi: 10.47852/bonviewJCCE2202144.