



PENGEMBANGAN APLIKASI PENDETEKSI KEMATANGAN BUAH MELON: STUDI KASUS APLIKASI MELONKU

Ihsanul Fikri Abiyyu¹, Hilmi Abidzar Tawakal²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

fikriabiyyu@student.nurulfikri.ac.id, hilmiabidzar@nurulfikri.ac.id

Abstract

There was research in Sekolah Tinggi Teknologi Terpadu Nurul Fikri about how to check the maturity of melon with the skin of melon. That research used image processing to build an algorithm to find the maturity of melon based on melon skin. The algorithm has been created into programming code and only runs in a command prompt that only some people can use. In order algorithm to can be used by many people, this research will continue the previous research to make a mobile application that connects into an algorithm with a web service. The development methodology that is used in this research is the Rational Unified Process. The mobile development using react-native to make a front end and flask to make a backend. The results of this research are expected to help the community to check the maturity of melons more easily.

Keywords: *Maturity of melon, React Native, Flask, Rational Unified Process, web service*

Abstrak

Terdapat penelitian di Sekolah Tinggi Teknologi Terpadu Nurul Fikri tentang bagaimana mengecek kematangan buah melon menggunakan pola pada kulit buah melon. Penelitian tersebut menggunakan *image processing* untuk membentuk algoritma yang dapat mencari pola kematangan buah melon berdasarkan kulit buah melon. Algoritma tersebut dibentuk dalam kode pemrograman dan hanya dapat dijalankan melalui *command prompt* sehingga hanya sebagian orang yang dapat menggunakan algoritma tersebut. Agar algoritma tersebut dapat digunakan lebih banyak orang, penelitian ini akan melanjutkan penelitian sebelumnya dengan membangun aplikasi *mobile* yang terkoneksi dengan algoritma yang dibuat pada penelitian sebelumnya menggunakan *web service*. Pada penelitian ini, metode pengembangan yang digunakan adalah *Rational Unified Process*. Pembangunan aplikasi *mobile* menggunakan *react native* untuk membangun tampilan aplikasi dan *flask* untuk *server* yang digunakan aplikasi. Hasil dari penelitian ini diharapkan dapat membantu masyarakat dalam mengecek kematangan buah melon menjadi lebih mudah.

Kata kunci: *Kematangan Buah Melon, React Native, Flask, Rational Unified Process, web service*

1. PENDAHULUAN

Saat membeli buah melon untuk dikonsumsi, orang akan mencari buah melon yang matang. Salah satu cara untuk mengecek kematangan buah melon adalah dengan melihat ketebalan dan kekasaran jaring pada buah melon. Semakin jaring tebal dan kasar, maka buah melon semakin matang.

Terdapat penelitian pengecekan kematangan buah melon dengan melihat ketebalan dan kekasaran jaring-jaring pada kulit buah melon yang dilakukan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri dengan memanfaatkan algoritma *support vector machine* (SVM) untuk melakukan *image processing* [1].

Penelitian tersebut menghasilkan algoritma yang berbentuk kode pemrograman dan dapat dijalankan melalui *command prompt*. Hal ini membuat algoritma yang telah dibuat pada penelitian sebelumnya belum dapat dipergunakan secara bebas oleh masyarakat dikarenakan tidak semua orang dapat memahami penggunaan *command prompt*. Oleh karena itu perlu adanya penelitian lebih lanjut agar algoritma yang telah dibuat dapat dimanfaatkan oleh masyarakat umum.

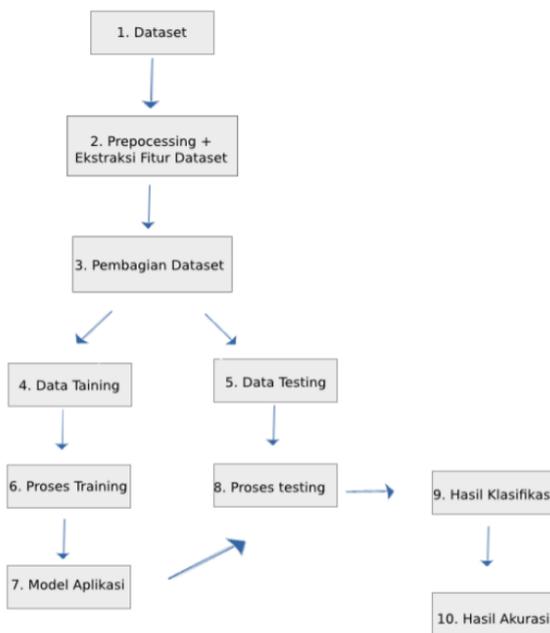
Penelitian ini akan melanjutkan penelitian sebelumnya yaitu dengan membangun aplikasi *mobile* yang terkoneksi dengan algoritma yang dibuat pada penelitian sebelumnya. Algoritma yang dibuat pada penelitian sebelumnya akan dibentuk menjadi *web service* sehingga dapat dikoneksikan dengan aplikasi *mobile* yang akan dibuat. Dengan

menggunakan *web service*, layanan atau data dapat diakses melalui berbagai bahasa pemrograman berbeda [2].

Diharapkan penelitian ini dapat membantu masyarakat agar dapat mengecek kematangan buah melon melalui aplikasi mobile yang dibuat pada penelitian ini.

1.1 Penelitian Terkait

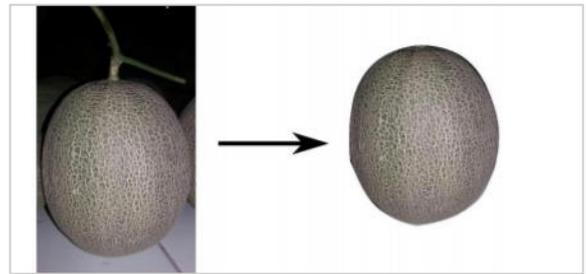
Pada penelitian sebelumnya, pengecekan kematangan buah melon dapat dilakukan dengan melihat ketebalan dan jaring jaring pada kulit buah melon. Tahapan yang dilakukan oleh penelitian sebelum nya untuk melakukan pengecekan kematangan buah melon melalui kulit dapat dilihat pada gambar 1 [1]:



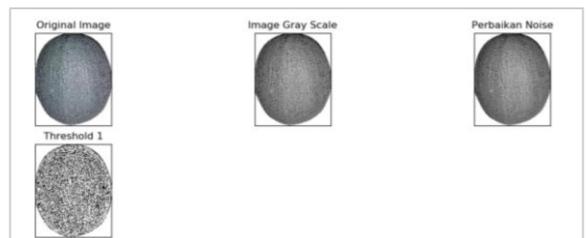
Gambar 1. Tahapan Pendeteksian Kematangan Buah Melon

- Dataset**
Terdapat 150 data foto buah melon yang diperoleh secara sampling. Data tersebut terbagi menjadi 3 kategori buah yaitu melon tidak matang berumur 45-50 hari setelah penanaman, melon setengah matang berumur 51-59 hari setelah penanaman, melon matang berumur 60-70 hari setelah penanaman.
- Preprocessing**
Kerusakan pada citra buah melon di perbaiki pada metode preprocessing yang terbagi menjadi 2 tahap yaitu *image cropping* dan *image enhancement*. *Image cropping* dilakukan untuk menghilangkan area di luar buah melon sehingga yang tampil hanya area buah melon nya saja. Sedangkan, *image enhancement* dilakukan untuk memperbaiki citra buah melon. Pertama, dilakukan konversi dari nilai *rgb* ke bentuk *grayscale*. Lalu, dilakukan *median blur* untuk memperbaiki *noise* pada buah melon. Terakhir, mengambil ciri pada citra buah melon dengan melakukan proses *adaptive threshold*. Contoh proses

image cropping dan *image enhancement* dapat dilihat pada gambar 2 dan gambar 3.

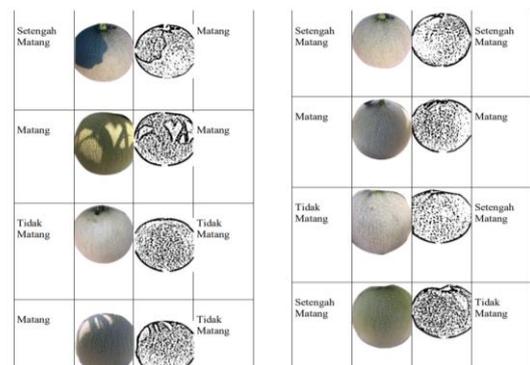


Gambar 2. Image Cropping pada Citra Buah Melon [1]



Gambar 3. Image Enhancement pada Citra Buah Melon [1]

- Ekstraksi Ciri Statistik**
Setelah dilakukan *preprocessing*, dapat dilakukan ekstraksi dengan cara mengubah citra yang sudah diperbaiki kualitasnya menjadi *histogram* berbentuk *array*. *Histogram* digunakan untuk melakukan proses training dan testing dengan menggunakan *svm*.
- Training Support Vector Machine (SVM)**
Training dilakukan untuk memperoleh model algoritma dengan klasifikasi buah melon matang, setengah matang, dan tidak matang dengan metode *svm*. *Training* dilakukan dengan melibatkan 100 citra buah melon per kategorinya yaitu melon matang, melon setengah matang, dan melon tidak matang.
- Proses Testing**
Model algoritma yang sudah didapat melalui proses *training* dapat digunakan untuk melakukan *testing*. Hasil dari *testing* menghasilkan klasifikasi yang mempunyai akurasi sebesar 76%. Hasil klasifikasi dari tingkat kematangan buah melon dapat dilihat pada gambar 4.



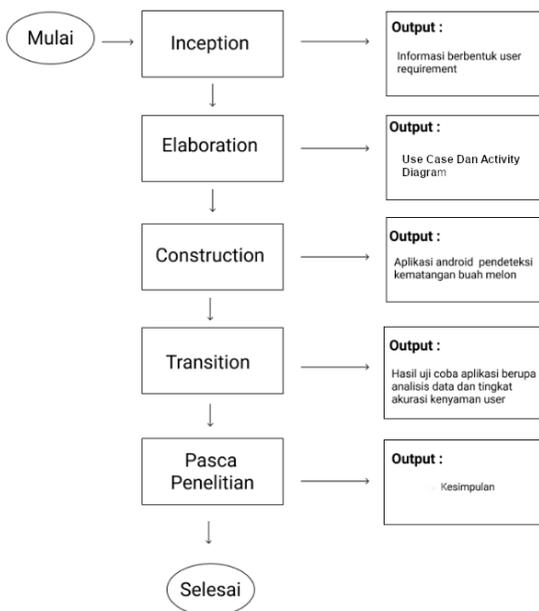
Gambar 4. Hasil Pengklasifikasian Citra Buah Melon [1]

2. METODE PENELITIAN

Metode penelitian yang dilakukan adalah kualitatif. Metode ini digunakan untuk menilai penggunaan aplikasi pada pengguna. Semakin pengguna mengerti bagaimana menggunakan aplikasi yang dibuat pada penelitian ini, maka aplikasi layak digunakan. Metode pengembangan aplikasi yang akan dilakukan adalah *Rational Unified Process (RUP)* dan terdiri dari 4 tahapan, yaitu *inception, elaboration, construction, dan transition* [3].

Tahap *inception* dilakukan untuk mendapatkan informasi melalui observasi atau wawancara. Tahap selanjutnya adalah *elaboration*. Pada tahap ini dilakukan analisis data yang didapatkan berdasarkan informasi pada tahap sebelumnya. Hasil analisis digunakan untuk membangun aplikasi pada tahap *construction*. Setelah aplikasi selesai dibangun, dilakukan pengujian aplikasi pada tahap *transition*. Penarikan kesimpulan dilakukan pada tahap terakhir yaitu pasca penelitian.

Hasil dari masing masing tahapan dapat dilihat pada gambar 5.



Gambar 5. Diagram Tahapan Penelitian

Metode pengujian menggunakan *black box testing* dan *usability testing*. *Black box testing* digunakan untuk pengujian perangkat lunak tanpa perlu tahu alat pembuatannya [4]. Sedangkan, *usability testing* digunakan untuk mengetahui kepuasan pengguna agar bisa dilakukan perbaikan serta penambahan fitur pada perangkat lunak [5].

Pengumpulan data dilakukan dengan observasi dan kuisisioner. Observasi dilakukan untuk pengamatan terhadap algoritma pengecekan buah melon dan melihat bagaimana proses algoritma tersebut dapat berjalan. Kuisisioner dilakukan untuk mendapatkan data yang akan digunakan untuk *usability testing*.

3. HASIL DAN PEMBAHASAN

Hasil yang didapatkan pada penelitian ini terbagi menjadi beberapa tahapan sesuai dengan metode rup yang digunakan.

3.1 Analisis dan Perancangan Sistem (*Inception and Elaboration*)

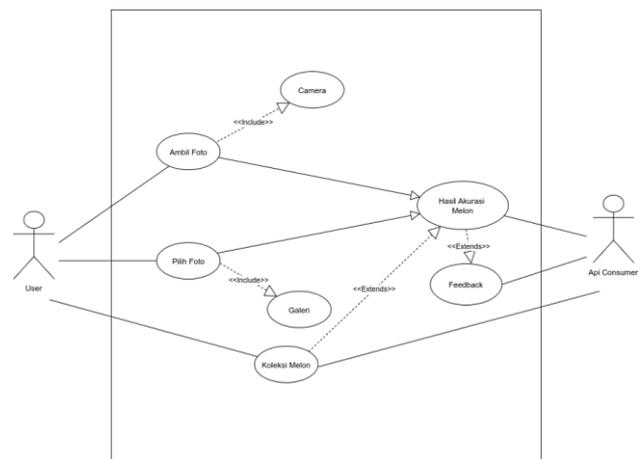
Tahapan ini menghasilkan analisis dan perancangan sistem dari penelitian yang dilakukan. Analisis yang dilakukan merupakan tahapan *inception* dan menghasilkan 2 kebutuhan sistem, yaitu kebutuhan fungsional dan non fungsional.

Kebutuhan fungsional yang dibutuhkan adalah aplikasi dapat mengambil gambar melon baik melalui kamera atau galeri, hasil dari gambar yang diambil dapat di cek prediksi kematangan nya, hasil kematangan dapat dilihat kembali tanpa melakukan pengambilan gambar ulang.

Kebutuhan non fungsional yang dibutuhkan adalah aplikasi dapat dijalankan pada *smartphone*, aplikasi dibuat menggunakan *react native*, server dibuat menggunakan *flask*, dan *database* menggunakan *sqlite*.

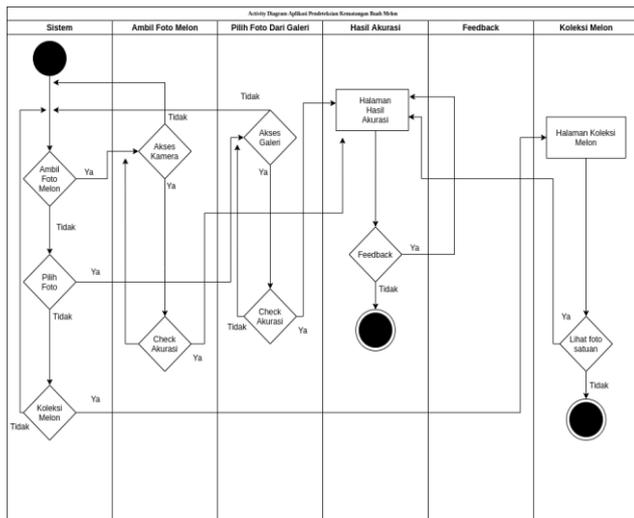
Hasil dari kebutuhan fungsional dan non fungsional dilanjut ke tahap *elaboration*. Tahap *elaboration* dilakukan untuk membuat *use case* dan *activity diagram* berdasarkan tahapan sebelumnya. *Use case* digunakan untuk mendeskripsikan bagaimana *user* bisa melakukan interaksi dengan sistem, sedangkan *activity diagram* digunakan untuk menggambarkan bagaimana aktivitas yang bekerja dalam suatu sistem [6].

Use case dan *activity diagram* yang telah dihasilkan, dapat dilihat pada gambar 6 dan 7.



Gambar 6. Use Case Aplikasi

Pada *use case* di gambar 6, *user* dapat mengecek akurasi melon melalui fitur ambil foto dan pilih foto. Selain itu *user* juga dapat melihat kembali semua hasil akurasi melon melalui fitur koleksi melon.



Gambar 7. Activity Diagram Aplikasi

Pada *activity diagram* di gambar 7, saat membuka aplikasi, user akan dihadapkan pada 3 menu yaitu ambil foto, pilih foto dari galeri, dan koleksi melon. Menu ambil foto digunakan untuk mengecek akurasi buah melon dengan menggunakan kamera. Sedangkan, menu pilih foto dari galeri digunakan untuk mengecek akurasi buah melon melalui galeri. Setelah foto di cek akurasi nya, maka hasil akurasi nya akan muncul pada menu hasil akurasi. Dalam menu hasil akurasi terdapat menu opsional yaitu *feedback*. *Feedback* digunakan untuk memberikan *feedback* jika hasil akurasi yang diberikan salah.

3.2 Implementasi (Construction)

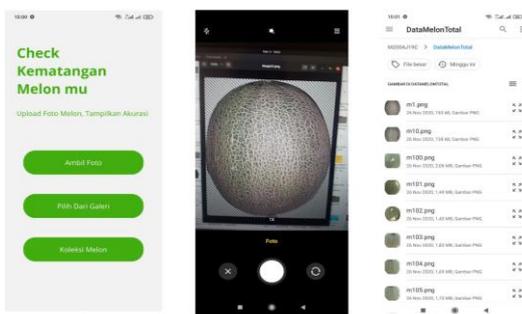
Tahapan implementasi dilakukan untuk membangun aplikasi. Tahapan ini merupakan tahapan *construction*. Implementasi yang dilakukan terbagi menjadi 2 yaitu implementasi antar muka dan *api*.

3.2.1 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan membangun aplikasi *mobile* menggunakan *react native* dan *expo*. *React native* berfungsi sebagai framework untuk membangun aplikasi berbasis *mobile* dengan menggunakan bahasa pemrograman *javascript* [7]. Sedangkan *expo* berfungsi sebagai *library* yang mendukung fitur *native* pada *mobile* seperti kamera, galeri, dan lain sebagainya [8].

Berikut adalah implementasi antarmuka yang diterapkan:

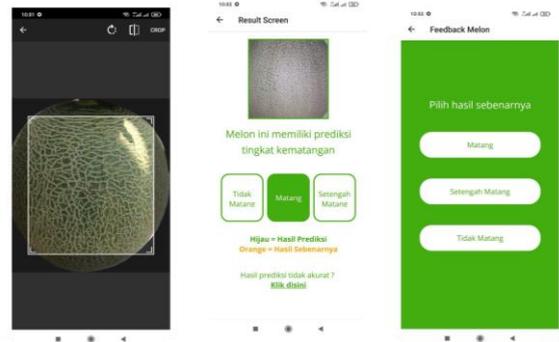
1. Start Screen



Gambar 8. Antarmuka Start Screen

Pada gambar 8, saat *user* membuka aplikasi, *user* akan diarahkan ke *start screen*. Terdapat 3 tombol yang dapat dipilih pada *start screen*, yaitu ambil foto, pilih dari galeri, dan koleksi melon. Ambil foto akan mengarahkan ke kamera dan *user* dapat mengambil gambar melon melalui kamera. Pilih dari galeri akan mengarahkan ke galeri, dan *user* dapat mengambil gambar melon melalui galeri. Koleksi melon akan mengarahkan ke koleksi melon *screen*.

2. Crop, Result Screen, dan Feedback Screen



Gambar 9. Antarmuka Result Screen

Setelah foto melon diambil melalui kamera atau galeri, foto tersebut dapat di *cropping* seperti pada gambar 9. Hasil dari *cropping* akan diprediksi kematangan nya dan *user* akan diarahkan ke *result screen* untuk melihat hasil prediksi foto yang telah diambil. Hasil prediksi dibagi menjadi 3 yaitu matang, setengah matang, tidak matang. Kotak yang berwarna hijau pada *result screen* merupakan hasil prediksi kematangan buah melon.

Jika prediksi yang diberikan tidak tepat, *user* dapat memberikan *feedback* akurasi yang tepat dengan menekan tulisan klik disini pada *result screen* dan akan diarahkan ke *feedback screen*. *User* dapat memilih hasil yang tepat berdasarkan pilihan pada 3 tombol yang tersedia pada *feedback screen* yaitu matang, tidak matang, dan setengah matang. Setelah *user* menekan salah satu tombol, maka akan diarahkan kembali ke *result screen* dan hasil *feedback* ditandai dengan kotak berwarna *orange* pada *result screen*.

3. Koleksi Melon Screen



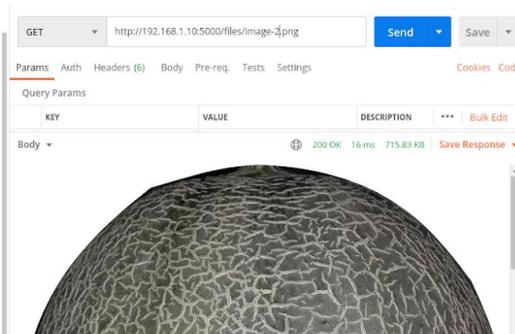
Gambar 10. Antarmuka Koleksi Melon Screen

Saat *user* menekan tombol koleksi melon pada *start screen*, maka *user* akan diarahkan ke *screen* koleksi melon yang terlihat pada gambar 10. *Screen* koleksi melon berisi kumpulan foto melon yang telah diprediksi kematangan nya. Jika salah satu foto melon pada koleksi melon di tekan, maka *user* akan diarahkan ke *result screen* dan hasil prediksi kematangan melon tersebut akan ditampilkan.

3.2.2 Implementasi Api

Implementasi *api* dilakukan pada server yang dibuat menggunakan *flask*. *Flask* digunakan untuk membangun *api* sederhana dan dapat diintegrasikan dengan komponen yang dibuat pihak ke 3 secara langsung [9].

Api dapat langsung dijalankan melalui *postman* dan dapat langsung di uji secara langsung meskipun permintaan *HTTP Request* yang diberikan tidaklah sederhana [10]. Contoh penggunaan *postman* terdapat pada gambar 11.



Gambar 11. Contoh Penggunaan Postman

Hasil implementasi *api* yang telah dilakukan dapat dilihat pada tabel 1.

Tabel 1. Implementasi Api

URL	Method	Implementasi
/melon	/post	Menambahkan data gambar melon dan hasil akurasi melon
/melons	/get	Menampilkan semua data melon
/melon/id	/get	Menampilkan data melon berdasarkan id
/melon/id	/patch	Menambahkan <i>feedback</i> akurasi pada data melon
/files/namafile	/get	Mengunduh gambar melon berdasarkan nama filenya

3.3 Evaluasi

Tahapan Evaluasi atau *transition* dilakukan untuk menguji aplikasi yang telah dibuat. Pengujian meliputi *black box testing* pada *api*, *blackbox testing* pada *aplikasi*, dan *usability testing*.

3.3.1 Black Box Testing pada Api

Terdapat 5 *endpoint* yang dijadikan item uji untuk pengujian *black box api*. Dari total 10 pengujian, 9 pengujian berhasil,

1 pengujian gagal. Persentase skor pengujian *api* adalah 90% berhasil dan 10% gagal. Detail pengujian yang dilakukan dapat dilihat pada tabel 2.

Tabel 2. Black Box Testing pada Api

No	Item uji	URL	Method	Total Pengujian	Total Pengujian Berhasil	Total Pengujian Gagal
1	Menambahkan data melon	/melon	/post	2	2	0
2	Menampilkan semua data melon	/melon	/get	1	1	0
3	Menampilkan data melon berdasarkan id	/melon/id	/get	2	2	0
4	Menambahkan <i>feedback</i> data melon	/melon/id	/patch	3	3	0
5	Mengunduh image melon berdasarkan nama filenya	/files/namafile	/get	2	1	1

3.3.2 Black Box Testing pada Aplikasi

Terdapat 5 item uji untuk pengujian *black box* pada aplikasi dengan total pengujian sebanyak 9 dan pengujian berhasil sesuai dengan yang diharapkan. Persentase skor keberhasilan adalah 100%. Detail pengujian yang dilakukan dapat dilihat pada tabel 3.

Tabel 3. Black Box Testing pada Aplikasi

No	Pengujian	Total Pengujian	Total Pengujian Berhasil	Total Pengujian Gagal
1	Dapat Mengambil foto melalui kamera dan melakukan pengecekan akurasi	2	2	0
2	Dapat mengambil foto melalui galeri dan melakukan pengecekan akurasi	2	2	0
3	Dapat melihat koleksi melon	2	2	0
4	Melihat salah satu hasil foto prediksi buah melon yang telah diprediksi sebelumnya pada koleksi melon screen	1	1	0
5	Menambahkan <i>feedback</i> akurasi data melon	2	2	0

3.3.3 Usability Testing

Terdapat 8 pengujian *usability testing* yang dilakukan. Hasil *usability testing* yang diujikan menghasilkan rata rata persentase skor 85,78%. Interpretasi skor yang diberikan pada pengujian adalah sangat mudah. Detail pengujian yang dilakukan dapat dilihat pada tabel 4.

Keterangan:

SS : Sangat Sulit S : Sulit
 M : Mudah SM : Sangat Mudah

Interpretasi Skor:

0% - 25% : Sangat Sulit 26% - 50% : Sulit
 51% - 75% : Mudah 76% - 100% : Sangat Mudah

Tabel 4. *Usability Testing*

No	Skenario	SS	S	M	SM	Skor
1	Membuka aplikasi kematangan buah melon			5	8	90,30%
2	Membuka menu ambil foto melon dan mengambil foto melalui kamera			5	8	90,30%
3	Membuka menu pilih foto dari galeri dan mengambil foto melalui galeri			6	7	88,46%
4	Melakukan cropping pada foto yang dipilih baik melalui kamera maupun galeri			6	7	88,46%
5	Menampilkan halaman hasil akurasi buah melon yang berisi foto melon dan hasil akurasi nya.			10	3	80,70%
6	Memberikan feedback hasil akurasi buah melon dan menampilkan hasil feedback pada halaman hasil akurasi	1	1	7	4	75%
7	Membuka menu koleksi melon dan menampilkan koleksi foto kematangan buah melon			7	6	86,53%
8	Memilih salah satu foto pada koleksi buah melon dan menampilkan hasil akurasi foto buah melon yang dipilih			7	6	86,53%
Rata-rata Persentase Skor				85,78%		

4. KESIMPULAN

Dari hasil penelitian yang penulis lakukan, dapat disimpulkan bahwa penelitian ini berhasil menjawab permasalahan pada penelitian yang dilakukan sebelumnya.

Di bawah ini adalah beberapa kesimpulan yang telah didapatkan:

1. Hasil kode algoritma yang telah dibuat pada penelitian sebelumnya, berhasil diterapkan ke dalam *server* dan dibentuk menjadi *web service*.
2. Aplikasi *mobile* berhasil dibuat menggunakan *react native* dan *server* menggunakan *flask*. *Database* yang digunakan adalah *sqlite3*.
3. Aplikasi *mobile* yang dibangun menggunakan metode pengembangan *rational unified process* dengan beberapa tahapan yaitu *inception*, *elaboration*, *construction*, *transition*, dan pasca penelitian.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada pihak yang membantu penulis dalam menyelesaikan jurnal ini khususnya kepada STT Terpadu Nurul Fikri.

DAFTAR PUSTAKA

- [1] A. Prayoga, H. A. Tawakal, and R. Aldiansyah, "Pengembangan Metode Deteksi Tingkat Kematangan Buah Melon Berdasarkan Tekstur Kulit Buah Dengan Menggunakan Metode Ekstraksi Ciri Statistik Dan *Support Vector Machine* (SVM)," vol. 4, no. 1, Art. no. 1, 2018.
- [2] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, Art. no. 01, Mar. 2019, doi: 10.33884/jif.v7i01.1004.
- [3] A. Fitria and H. Widowati, "Implementasi Metode *Rational Unified Process* Dalam Pengembangan Sistem Administrasi Kependudukan," vol. 22, p. 10, 2017.
- [4] F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya, and A. Saifudin, "Pengujian *Black Box* pada Aplikasi Sistem Seleksi *Sales* Terbaik menggunakan Teknik *Equivalence Partitions*," *J. Inform. Univ. Pamulang*, vol. 4, no. 4, Art. no. 4, Dec. 2019, doi: 10.32493/informatika.v4i4.3782.
- [5] M. Y. Alfiqie, I. Aknuranda, and N. H. Wardani, "Evaluasi *Usability* pada Aplikasi UBER menggunakan Pengujian *Usability*," p. 8.
- [6] D. W. T. Putra and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," vol. 7, no. 1, p. 8, 2019.
- [7] P. Wiguna, I. Swastika, and I. Satwika, "Rancang Bangun Aplikasi *Point of Sales* Distro *Management System* dengan Menggunakan *Framework React Native*," *J. Nas. Teknol. Dan Sist. Inf.*, vol. 4, pp. 149–159, Jan. 2019, doi: 10.25077/TEKNOSI.v4i3.2018.149-159.
- [8] R. T. Yunandar, "Pengujian *Usability System Framework React Native* dengan *Expo* untuk Pengembang Aplikasi Android Menggunakan *Use Questionnaire*," vol. 3, no. 1, p. 8, 2018.
- [9] R. Irsyad, "Penggunaan Python *Web Framework* Flask Untuk Pemula." OSF Preprints, Dec. 19, 2018, doi: 10.31219/osf.io/t7u5r.
- [10] U. M. Qibtiyah and S. Rahayu, "Implementasi *JSON Web Service* pada Aplikasi *Digital Library* Politeknik Sukabumi," p. 8.