

Rekayasa Perangkat Lunak Sistem Penjadwalan Perawatan Nenas Menggunakan Algoritma First In First Out

Putri Betha¹, Kasmawi², Danuri³

Politeknik Negeri Bengkalis

Putribetha9@gmail.com¹, kasmawi@polbeng.ac.id², danuri@polbeng.ac.id³

Abstract

Pineapple care scheduling system uses information technology that can make it easier for farmers to carry out maintenance. Conventional scheduling systems can cause difficulties in determining the treatment schedule, so that pineapple care is not scheduled properly. The purpose of this research is software engineering of pineapple care scheduling system by applying round robin algorithm. System development using the waterfall method. System testing is done by following the black box testing method. The results showed that the scheduling system application can help pineapple farmers in managing pineapple care schedules.

Keywords : scheduling system, pineapple care, round robin algorithm, waterfall.

1. PENDAHULUAN

Sistem adalah sebuah hasil akhir dari serangkaian suatu proses yang kompleks yang selanjutnya disusun untuk mencapai tujuan tertentu. Penjadwalan didefinisikan sebagai proses pengalokasian sumber untuk memilih sekumpulan tugas dalam jangka waktu tertentu. Tujuan penjadwalan untuk mengurangi waktu tunggu sehingga waktu proses dapat berkurang dan menghasilkan sebuah produktivitas. Sistem penjadwalan merupakan proses mengatur rencana kegiatan dengan pembagian waktu pelaksanaan (Hamdani & Ziveria, 2017).

Perawatan tanaman memerlukan sistem penjadwalan yang baik. Hal ini untuk mendukung pertumbuhan tanaman agar menghasilkan hasil panen yang baik. Penjadwalan perawatan yang baik akan mendukung perawatan tanaman, sehingga perawatan lebih terarah, terstruktur dan tepat sasaran (Ainida, 2019).

Desa Penampi dikenal sebagai penghasil nenas di Kabupaten Bengkalis, hal ini disampaikan langsung oleh petani nenas desa tersebut. Saat ini sekitar 45-55 hektar lahan di desa tersebut sudah ditanami nenas. Setiap petani nenas masing-masing memiliki kebun nenas minimal 2 hektar. Dalam proses perawatan nenas, tidak jarang petani mengalami *human error* atau kendala dalam melaksanakan proses perawatan. Kendala yang dihadapi dapat berupa cuaca, *human error*, faktor lingkungan. Akibat dari kendala tersebut, proses perawatan nenas menjadi tidak teratur dan berakibat pada pertumbuhan serta perkembangan nenas itu sendiri.

Sistem penjadwalan yang berlangsung saat ini di Desa Penampi masih menggunakan cara konvensional. Petani nenas hanya memprediksi jangka kapan tanggal perawatan dan bahkan seringkali mengabaikan pembersihan kebun. Kurangnya perhatian petani terhadap proses penjadwalan perawatan nenas dapat memicu terjadinya penundaan proses perawatan nenas. Kemudian untuk mengatasi hal tersebut di perlukan sebuah sistem berupa aplikasi penjadwalan perawatan nenas untuk para petani nenas yang mampu menyusun jadwal perawatan nenas dengan baik. Hal ini untuk membantu petani nenas dalam mengontrol proses perawatan nenas, sehingga proses perawatan nenas dapat dilakukan sesuai jadwal serta mampu mendapatkan hasil panen nenas yang lebih baik. Penelitian ini akan menghasilkan aplikasi sistem penjadwalan perawatan nenas berbasis android sehingga dengan sistem penjadwalan perawatan ini nantinya akan mempermudah petani dalam mengatur jadwal perawatan nenas sehingga hasil panen bisa lebih baik.

2. TINJAUAN PUSTAKA

FIFO atau *first in first out* merupakan metode dengan alur flow dimana proses yang lebih dahulu masuk akan lebih dahulu dilayani (Firgia & Nurcahyo, 2021). Jika ada proses yang datang pada waktu yang bersamaan, maka akan di eksekusi berdasarkan urutan kedatangan. Jika proses di antrian belakang harus menunggu setiap proses yang ada di depannya selesai di eksekusi. Setiap proses yang masuk kedalam stuktur data antrian harus sesuai dengan waktu tiba (Parinduri & hutalung, 2019).

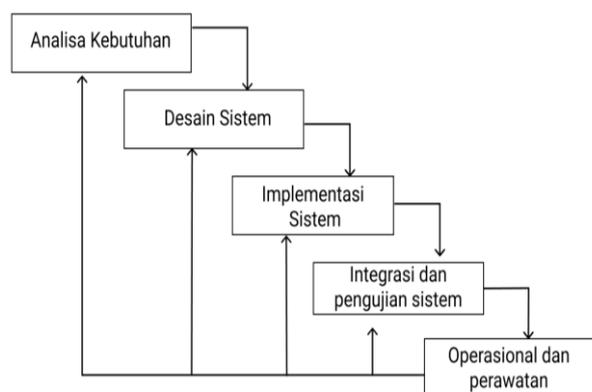
Berdasarkan penelitian Baskoro dan S.P (2017) dengan judul Aplikasi Reminder Jadwal Kerja Untuk Pekerja *Casual* Berbasis Android (Studi kasus: Grand City Convention & Exhibition Surabaya). Penelitian ini menghasilkan sebuah aplikasi *reminder* untuk para pekerja *casual*. Aplikasi ini membantu para pekerja dalam mengingat setiap jadwal kerja. Metode yang digunakan dalam penelitian ini yaitu *waterfall*. Pada metode penelitian *waterfall* terdapat beberapa tahapan yaitu pengumpulan data, analisa kebutuhan, implementasi, desain sistem dan uji coba. Berdasarkan penelitian Havaso, dkk (2019) dengan judul penelitian Aplikasi Pengingat Jadwal Imunisasi pada Puskesmas Kebun Handil Kota Jambi Berbasis Android. Hasil dari penelitian ini adalah sebuah aplikasi pengingat jadwal imunisasi untuk puskesmas Kebun Handil kota Jambi. Aplikasi ini juga menampilkan informasi tumbuh kembang anak dan informasi vaksin KIPI (Kejadian Ikutan Paksa Imunisasi) juga dapat memberikan petunjuk tentang praktek dokter yang terdekat dengan lokasi orang tua.

Selanjutnya pada penelitian yang dilakukan oleh Hartati dan Sintawati (2020) dengan judul penelitian Implementasi Metode *Waterfall* Pada Perancangan Aplikasi SIPSIBA. Hasil dari penelitian ini merupakan sebuah aplikasi berbasis *website* yang berfungsi sebagai sistem informasi penerimaan mahasiswa baru. Dengan aplikasi SIPSIBA ini diharapkan proses dalam proses pendaftaran calon siswa baru diharapkan dapat menjadi lebih efektif dan efisien.

Berdasarkan penelitian Destingrum dkk (2017) dengan judul penelitian Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan *Framework Codeigniter* (Studi Kasus : Rumah Sakit Yukum Medical Centre). Hasil dari penelitian ini adalah sebuah aplikasi yang menjadi solusi untuk pengolahan data serta dapat memberikan informasi tentang kesehatan dan jadwal dokter kepada pasien yang lebih akurat. Aplikasi ini dikembangkan menggunakan metode *waterfall*. Penggunaan model *waterfall* dalam pengembangan sistem diharapkan mampu memudahkan pembuatan sehingga pembangunan sistem bisa terstruktur.

3. METODE PENELITIAN

Metode penelitian ini mengikuti tahapan dari alur metode *waterfall* sebagai metode pengembangan sistem. Dalam proses perancangan terdapat 4 (empat) tahapan yaitu : Gambar 1 menunjukkan tahapan yang dilakukan dalam metode *waterfall* Rosa dan Shalauddin, 2016



Gambar 1. Tahapan Metode *waterfall*
(Sumber : Rosa dan shalauddin, 2016)

- 1) Analisis Kebutuhan Perangkat Lunak
Pada analisis kebutuhan, dilakukan proses pengumpulan data secara intensif agar diperoleh kebutuhan perangkat lunak seperti apa yang dibutuhkan *user*.
- 2) Desain Sistem
Tahapan ini merupakan proses yang fokus pada desain representasi antar muka, arsitektur perangkat lunak, dan prosedur pengkodean dengan mengimplementasikan hasil rancangan desain.
- 3) Implementasi Sistem
Pengkodean dilakukan berdasarkan implementasi kebutuhan sistem pada perancangan desain. Hasil dari tahapan ini adalah program komputer yang merupakan hasil representative proses desain yang telah terjadi.
- 4) Integrasi dan Pengujian Sistem
Tahapan pengujian pada perangkat lunak dilakukan secara logik dan fungsional. Hal ini dilakukan untuk menekan angka kesalahan dan memastikan bahwa keluaran yang dihasilkan sesuai dengan yang diinginkan.
- 5) Operasional dan Perawatan
Tahapan pemeliharaan saat terjadi sebuah kesalahan yang tidak terdeteksi saat proses pengujian berlangsung atau sistem yang rusak saat beradaptasi dengan lingkungan atau sistem yang baru. Tahapan akhir ini dapat mengulang proses pengembangan mulai dari analisis spesifikasi untuk perubahan pada perangkat lunak yang ada.

4. HASIL PENELITIAN DAN PEMBAHASAN

Hasil penelitian ini adalah rekayasa perangkat lunak sistem penjadwalan perawatan nenas menggunakan algoritma FIFO. Aplikasi ini dapat membantu petani nenas dalam mengatur jadwal perawatan nenas dalam masa tanam 1 periode. Dengan perawatan nenas yang baik dan tepat waktu diharapkan bisa meningkatkan hasil panen nenas petani. Berikut pembahasan penerapan Algoritma FIFO dengan metode pengembangan *waterfall* sebagai berikut :

- 1) Analisa kebutuhan
Pada tahapan awal ini dilakukan identifikasi masalah pada kelompok petani nenas di desa Penampi. Identifikasi masalah difokuskan pada sistem penjadwalan perawatan nenas, mulai dari tahap penanaman sampai tahap panen. Dari identifikasi masalah yang dilakukan, diketahui para petani nenas sering melewatkan tahapan-tahapan dalam perawatan nenas, jadwal pemupukan nenas yang tidak tepat, gangguan cuaca yang sebenarnya sangat berpengaruh pada pertumbuhann nenas dan hasil panen nenas.

1.1 Analisa Perawatan Nenas

Perawatan nenas yang berlangsung di desa Penampi dimulai sejak penanaman nenas. Bibit nenas berasal dari anakan nenas, kemudian ditanam di lahan yang sudah dibersihkan dari tungkul-tungkul kayu sampai tanah dalam kondisi gembur. Setelah 30 hari penanaman akan dilakukan pemupukan. Hal tersebut dilakukan selama 3 bulan berturut-turut. Setelah itu, nenas akan dipupuk setiap dua bulan sekali. Sampai usia nenas mencapai 8 bulan, dilakukan pemupukan untuk merangsang buah nenas agar lekas dipanen.

1.2 Sistem Perawatan Nenas

Sistem perawatan nenas dirancang berbasis android untuk mempermudah proses penjadwalan perawatan nenas. Sistem tersebut akan mempermudah petani nenas dalam sistem penjadwalan perawatan nenas.

1.3 Kebutuhan Fungsional

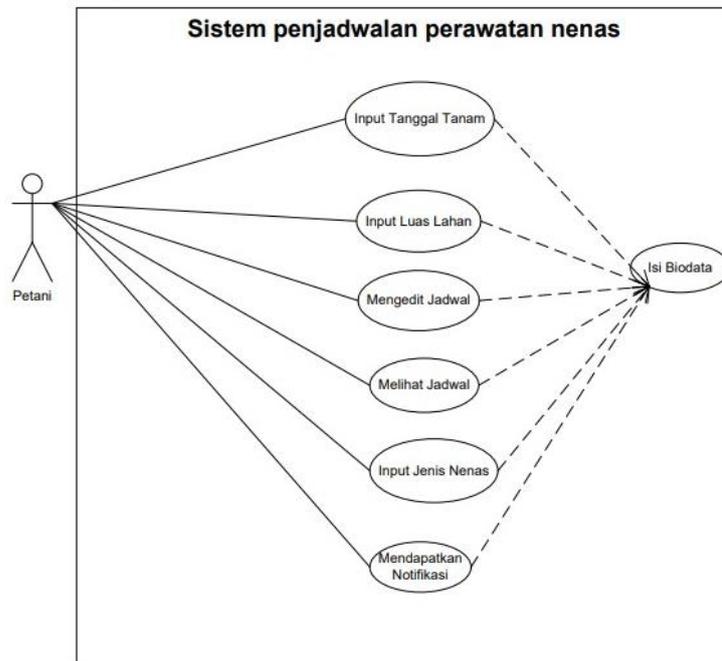
- a. Fungsi membuat jadwal
- b. Fungsi menampilkan jadwal
- c. Fungsi Mengubah jadwal

- d. Fungsi menambahkan jadwal
- e. Fungsi Notifikasi jadwal
- 1.4 Kebutuhan Non Fungsional
 - a. Sistem dijalankan si *smartphone*
 - b. Waktu yang dibutuhkan sistem untuk memproses perintah dari user
 - c. Memiliki tampilan antarmuka yang menarik sehingga *user* bisa menggunakan sistem dengan mudah

2) Desain Sistem

Tampilan sistem dirancang menggunakan *usecase diagram*, *activity diagram*. Tampilan pada sistem perlu diperhatikan dari penggunaan warna dan tata letak menu untuk mencapai tingkat kepuasan *user* dalam menggunakan sistem.

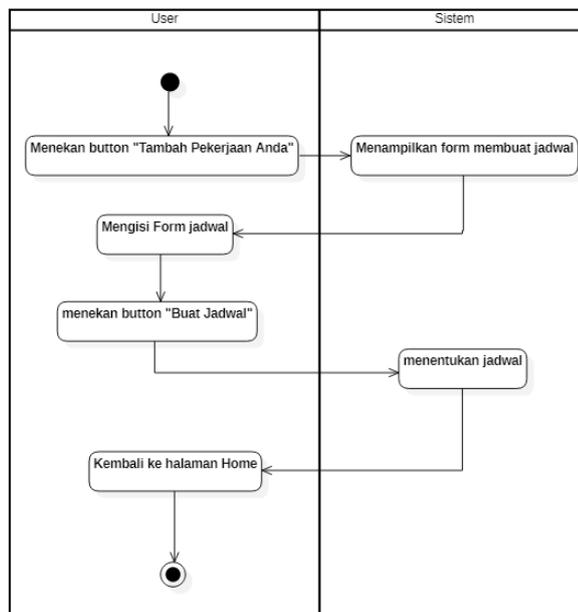
a. UseCase Diagram



Gambar 4. 1 Usecase Diagram

b. Activity Diagram

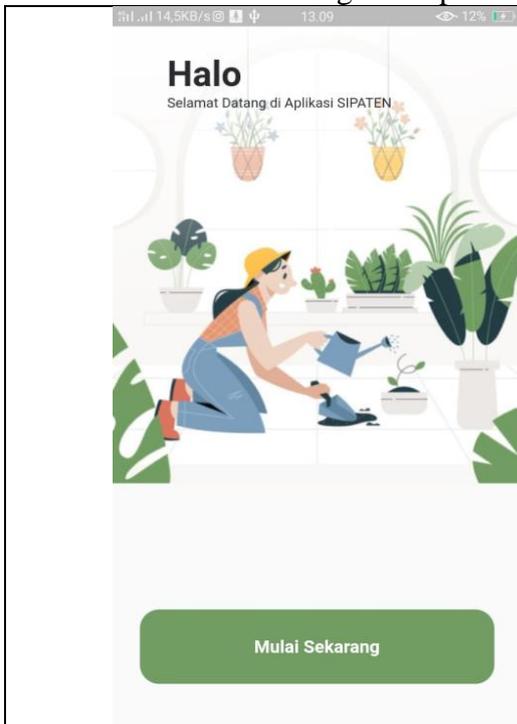
Activity Membuat Jadwal



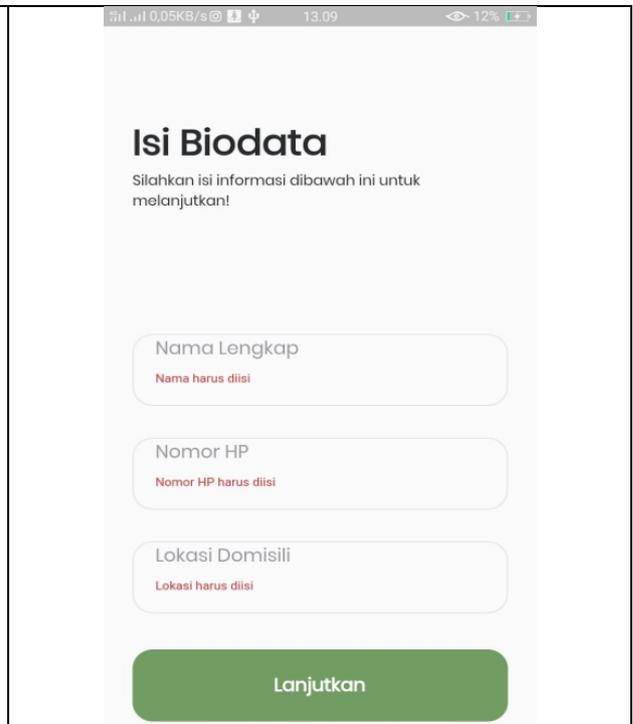
3) Implementasi Sistem

Implementasi sistem berupa pengkodean yang merujuk pada hasil dari perancangan desain sistem. Berikut hasil implementasi dari perancangan yang dilakukan :

1. Hasil Perancangan tampilan



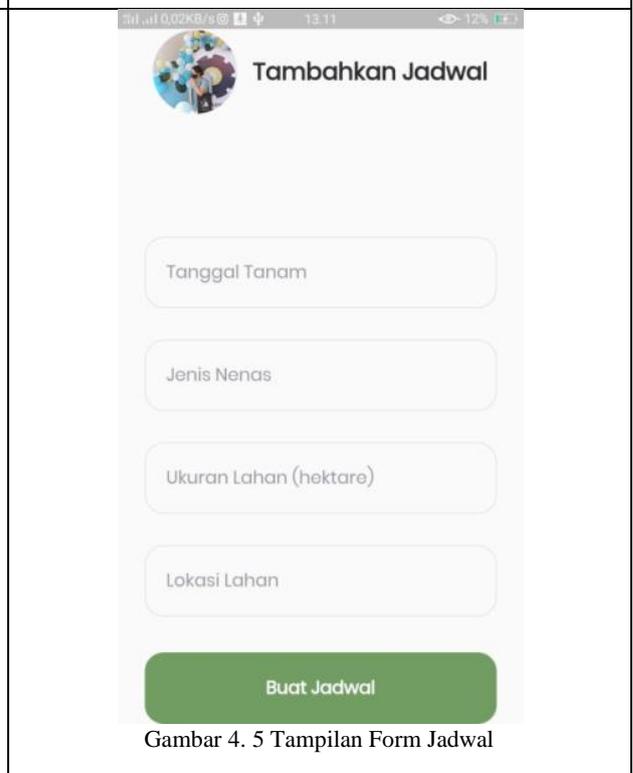
Gambar 4. 2 Tampilan awal sistem



Gambar 4. 3 Tampilan Form Biodata



Gambar 4. 4 Tampilan Home

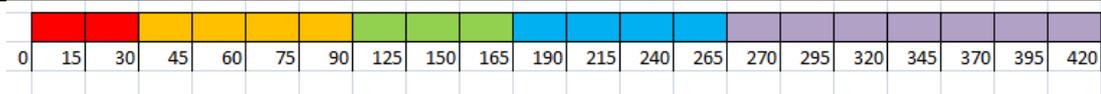


Gambar 4. 5 Tampilan Form Jadwal

2. Pengkodean dengan penerapan FIFO

Tabel 4. 1 Proses Penerapan algoritma FIFO

Proses	Kegiatan	Arrival Time	Burst Time	Start Time	Finish	Respon Time (ft-at)
P1	Pemupukan 1	0	30	0	30	30
P2	Pemupukan 2	31	60	30	90	60
P3	Pembersihan Lahan	91	75	90	165	74
P4	Pemupukan 3	167	105	190	265	98
P5	Pemupukan 4	267	150	270	420	153
TOTAL			420			415



Gambar 4. 6 Gantchart

Berikut potongan kode pada proses penjadwalan dengan penerapan algoritma FIFO :

```
// Ini adalah implemmentasi FIFO
DbHelper().insert('detailjob', DetailJob(value, DateFormat('yyyy-MM-dd')
    .format(DateTime.parse(tanggal).add(Duration(days: 30+ukuran))), 'Pemupukan 1', 'false')).the
n((value) async => {
    await AndroidAlarmManager.oneShotAt(DateTime.parse(tanggal).add(Duration(days: 30+ukuran)), val
ue, BackgroundService.callback, exact: true, wakeup: true)
    }),
    await DbHelper().insert('detailjob', DetailJob(value, DateFormat('yyyy-MM-dd')
    .format(DateTime.parse(tanggal).add(Duration(days: 60+ukuran))), 'Pemupukan 2', 'false')).the
n((value) async => {
    await AndroidAlarmManager.oneShotAt((DateTime.parse(tanggal).add(Duration(days: 60+ukuran))), v
alue, BackgroundService.callback, exact: true, wakeup: true)
    }),
    await DbHelper().insert('detailjob', DetailJob(value, DateFormat('yyyy-MM-dd')
    .format(DateTime.parse(tanggal).add(Duration(days: 60+ukuran))), 'Bersih Kebun 1', 'false')).
then((value) async => {
    await AndroidAlarmManager.oneShotAt((DateTime.parse(tanggal).add(Duration(days: 60+ukuran))), v
alue, BackgroundService.callback, exact: true, wakeup: true)
    }),
    await DbHelper().insert('detailjob', DetailJob(value, DateFormat('yyyy-MM-dd')
    .format(DateTime.parse(tanggal).add(Duration(days: 150+ukuran))), 'Pemupukan 3', 'false')).th
en((value) async => {
    await AndroidAlarmManager.oneShotAt((DateTime.parse(tanggal).add(Duration(days: 150+ukuran))),
value, BackgroundService.callback, exact: true, wakeup: true)
    }),
    await DbHelper().insert('detailjob', DetailJob(value, DateFormat('yyyy-MM-dd')
    .format(DateTime.parse(tanggal).add(Duration(days: 180+ukuran))), 'Bersih Kebun 2', 'false'))
.then((value) async => {
    await AndroidAlarmManager.oneShotAt((DateTime.parse(tanggal).add(Duration(days: 180+ukuran))),
value, BackgroundService.callback, exact: true, wakeup: true)
    }),
    await DbHelper().insert('detailjob', DetailJob(value, DateFormat('yyyy-MM-dd')
    .format(DateTime.parse(tanggal).add(Duration(days: 210+ukuran))), 'Pemupukan 4', 'false')).th
en((value) async => {
    await AndroidAlarmManager.oneShotAt((DateTime.parse(tanggal).add(Duration(days: 210+ukuran))),
value, BackgroundService.callback, exact: true, wakeup: true)
    }),
    setState(() {
        isLoading = false;
    }),
    BotToast.showSimpleNotification(title: "Berhasil menambahkan pekerjaan."),
});
```

4) Integrasi dan Pengujian Sistem

Integrasi dan pengujian dilakukan untuk memastikan apakah sistem berjalan dengan baik dan sesuai dengan kebutuhan user. Hasil integrasi memperlihatkan bahwa sistem dapat berjalan dengan baik, sistem dapat memroses jadwal perawatan sesuai dengan kebutuhan serta fitur *reschedule* dapat mengatur ulang jadwal dengan baik. Sementara pengujian sistem dilakukan menggunakan metode *black box* untuk memastikan semua tampilan dapat berfungsi dengan baik tanpa error.

1. Pengujian Integrasi



Gambar 4. 7 Input jadwal



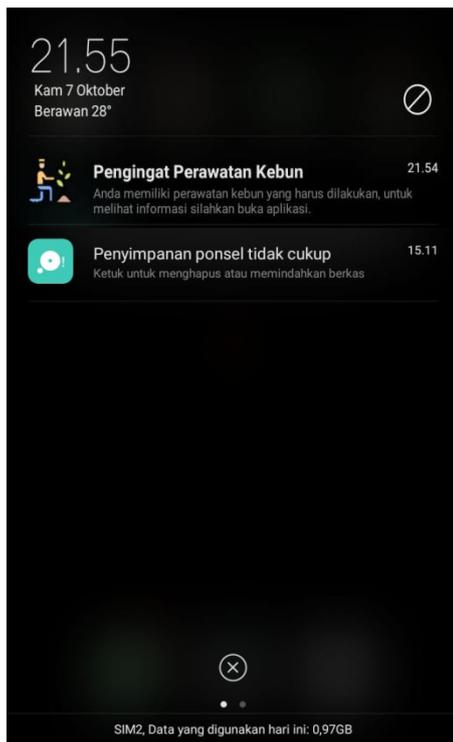
Gambar 4. 8 Hasil Inputan



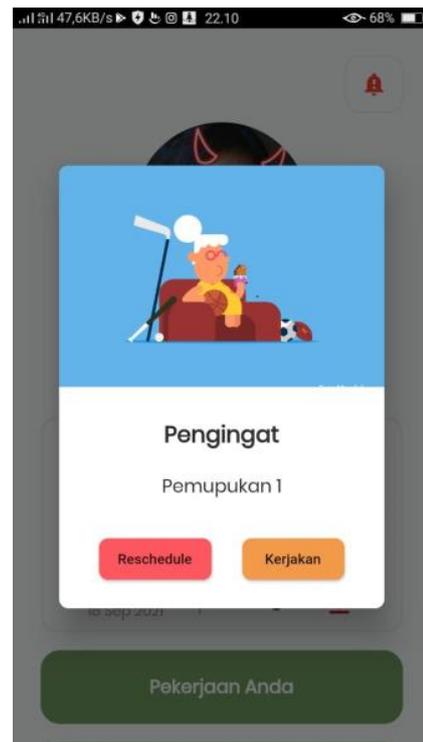
Gambar 4. 9 Jadwal Perawatan



Gambar 4. 10 Pekerjaan yang sudah di Kerjakan



Gambar 4. 11 Notifikasi



Gambar 4. 12 Notifikasi pada aplikasi

2. Pengujian *black box*

Tabel 4. 2 Tabel Pengujian

Aksi	Yang Diharapkan	Hasil yang didapat	Kesimpulan
Menekan tombol “mulai” pada tampilan awal sistem	Menampilkan form biodata pengguna	Setelah menekan tombol “mulai” menampilkan form biodata pengguna	Berhasil
Mengisi seluruh form biodata dengan benar dan klik button “lanjutkan”	Seluruh data terinput dan dapat ditmpilkan di halaman profil	Setelah mengisi seluruh form data petani dan menekan tombol lanjutkan,	Berhasil

		diharapkan sistem menampilkan biodata petani	
Menambahkan foto profil dengan mengklik foto profil	Dapat mengganti foto profil sesuai dengan yang diharapkan	Foto Profil dapat diganti dengan memilih 1 foto dari perangkat pribadi	Berhasil
Mengklik tombol “Tambahkan Pekerjaan”	Menampilkan form isian untuk data perawatan	Menampilkan form inputan dengan beberapa field isian seperti, tanggal tanam, jenis nenas, luas lahan dan lokasi lahan	Berhasil
Mengisi seluruh field isian dari halaman “Tambahkan Pekerjaan”	Form dapat di isi dengan baik	Sistem mampu menerima inputan dan menyimpan dengan baik	Berhasil
Tidak mengisi form isian “Tambahkan Pekerjaan” dan mengklik tombol “Buat Jadwal”	Form menampilkan pesan <i>requiret</i> bahwa field tidak boleh kosong	Sistem mengeluarkan pesan <i>requiret</i>	Berhasil
Mengisi form isian “Tambahkan pekerjaan” dengan benar. Lalu mengklik tombol “buat jadwal”	Sistem menerima inputan dengan baik dan mampu menampilkan jadwal 1 tahun masa perawatan nenas	Sistem memberikan pesan “Berhasil menambahkan pekerjaan” dan menampilkan jadwal 1 tahun masa perawatan nenas	Berhasil
Pada tampilan pengingat menekan tombol “ <i>reschedule</i> ”	Jadwal perawatan berubah dan bertambah 1 hari	Sesuai keinginan	Berhasil
Pada Tampilan pengingat, menekan tombol “Kerjakan”	Pada tampilan jadwal kegiatan terdapat centang biru	Sesuai keinginan	Berhasil

5. KESIMPULAN DAN SARAN

Sistem penjadwalan perawatan nenas dapat membantu petani dalam nenas dalam proses penjadwalan perawatan nenas dalam 1 periode. Fitur *reschedule* pada sistem dapat mengatur ulang jadwal perawatan nenas jika petani nenas mengalami kendala dalam jadwal perawatan. Berdasarkan pengujian yang dilakukan menggunakan metode *black box* sistem penjadwalan perawatan nenas dapat berfungsi dengan baik sesuai dengan fungsionalitas sistem. Saran untuk pengembangan sistem penjadwalan nenas dapat dilakukan dengan beberapa algoritma penjadwalan selanjutnya bisa dengan menambahkan fitur edukasi perawatan nenas dan batas waktu *reschedule* yang aman bagi perawatan nenas.

6. DAFTAR PUSTAKA

- Hamdani, F., & Ziveria, M. (2017). Aplikasi Pengingat Vaksinasi Hewan Berbasis Desktop. *Jurnal Sains dan Teknologi* , 131.
- Haryanto, E. V. (2012). *Sistem Operasi Konsep & Teori*. Yogyakarta: ANDI.
- Havaso, A. T., Mulyadi, & Suratno, E. (2019). Aplikasi Pengingat Jadwal Imunisasi pada Puskesmas Kebun Handil Kota Jambi Berbasis Android. *Journal of Computer and Information Technology* .

- Jaya, T. S. (2018). Pengujian Aplikasi dengan. Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi kasus :Kantor Digital Politeknik Negeri Lampung .
- Jayanto, A. D., Niswatin, R. K., & Kasih, P. (2021). Sistem Informasi Dan Pelayanan E-Tiket Berbasis Website Menggunakan Algoritma FIFO Pada Kawasan Wisata Trenggalek.
- Kosidin, & Farizah, R. N. (2016). PEMODELAN APLIKASI MOBILE REMINDER BERBASIS ANDROID.
- Putri, R. A. (2021). APLIKASI SIMULASI ALGORITMA PENJADWALAN SISTEM OPERASI . Jurnal Teknologi Informasi .
- S, R. A., & Shalauddin, M. (2016). Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika Bandung.